

Trigger Control Module

9U-Version

2007 Firmware

B. Arnold, H. Bergauer, M. Eichberger, K. Kastner, B. Neuherz, M. Padrta, T. Schreiner, A.
Taurok



Versions

TCS V1023: March 2010

TCSM V0003 March 2008

Table of contents:

1	Description	7
1.1	Front Panel	8
1.2	Input/Output overview	10
1.3	Power.....	11
2	Interfaces	11
2.1	Interface to TTC Partitions (sTTS)	11
2.1.1	Input: Status Signals (Fast Signals).....	11
2.1.2	Output to TTCci - L1A and BGo-Commands.....	12
2.1.3	Output to Emulators	14
2.2	Interface to aTTS (DAQ)	14
2.2.1	Input: aTTS Status signals.....	14
2.2.2	Output: TCS-status signals.....	15
2.3	Interface to GT processor	16
2.3.1	Final_OR and Tech_OR.....	16
2.3.2	GT status signals	16
2.3.3	TCS $\leftarrow \rightarrow$ FDL	16
2.3.4	TCS $\leftarrow \rightarrow$ TIM.....	17
3	Trigger Control logic.....	18
3.1	Synchronization to LHC orbit.....	18
3.2	DAQ-Partition controllers (PTC) overview	18
3.3	Time Slice distribution	19
3.4	Trigger Throttle logic	19
3.5	STATUS bits - Decoder and Merger.....	20
3.6	DAQ-Partition Controller - details	20
3.6.1	State Machine of PTC	21
3.6.1.1	Internal STATES of PTC	22
3.6.1.2	Response to TTS signals	23
3.6.1.3	RESYNC and HARDWARE RESET procedure	24
3.6.2	Calibration circuit.....	24
3.6.3	BC Table	25
3.6.4	Trigger merging.....	26
3.7	MONITORING COUNTERS	28
3.8	BGo command codes	28
3.9	OUTPUT to TTCci boards.....	28
3.10	Event Record for Event Manager (EVM)	28
4	TCS chip.....	29

4.1	Firmware Versions	29
4.1.1	V1023	29
4.1.2	V1022	30
4.1.3	V1021	30
4.1.4	V1020	30
4.1.5	V101F	31
4.1.6	V001E	31
4.1.7	V001D	31
4.1.8	V001C	31
4.1.9	V001B	31
4.1.10	V001A	31
4.1.11	V0019	32
4.1.12	V0018	32
4.1.13	V0017	32
4.1.14	V0016	32
4.1.15	V0015	32
4.1.16	V0014	33
4.1.17	V000D	33
4.1.18	V000C	33
4.1.19	V000B	33
4.1.20	V000A	33
4.2	VME ADDRESSES	33
4.2.1	Overview	33
4.2.2	Common addresses	34
4.2.3	PTC0...7 addresses	37
4.3	Power-Up values	40
4.4	COMMON Registers	43
4.4.1	THROTTLE RULES	43
4.4.2	TIME_SLOTS	45
4.4.3	ASSIGN 32 Detector Partitions to 8 DAQ partitions	46
4.4.4	EN_IO & ASSIGN_LUM_GT	47
4.4.5	ORBIT_LENGTH-1	48
4.4.6	GAP_LIMITH and GAP_LIMITL	48
4.4.7	SETTLE_TIME and RECOVER_TIME	49
4.4.8	BCRES_DELAY	49
4.4.9	GT-STATUS	50
4.4.10	COMMON_STATUS	50
4.4.11	COMMON_CMD pulses	51
4.4.12	IDENTIFIER registers	52

4.4.13	STATUS of DAQ- & Subdetector-Partitions.....	53
4.4.14	Combined STATUS for PTC	53
4.4.15	ORBIT NUMBER.....	54
4.4.16	PARTITION RUN NUMBERS	54
4.4.17	LUMINOSITY_PERIOD.....	55
4.4.18	BOARD_ID.....	55
4.4.19	SIM_EMU_STATUS3128, 2724.....	55
4.4.20	SIM_EMU_CTRL.....	55
4.4.21	SWITCH_TP2PAN.....	56
4.4.22	TESTMASK0 ... 8.....	56
4.4.23	LUMINOSITY NUMBER.....	58
4.4.24	BC_ERRORS	58
4.4.25	ROP_BCR_DELAY.....	58
4.5	PTC0... PTC7 addresses.....	59
4.5.1	PTC0...7_BC_TABLE0 .. 4kx16 DPRAM	59
4.5.2	PTC0...7_BC_TABLE1 .. 4kx4 DPRAM	60
4.5.3	BGO default values	60
4.6	PTC0... PTC7 REGISTER.....	61
4.6.1	PTC0...7_CMD_REG register	61
4.6.2	PTC0...7_RANDOM_FREQ.....	63
4.6.3	PTC0...7_RANDOM_PRESCALE_FACTOR.....	63
4.6.4	PTC0...7_TRIG_TYPE_A and - TRIG_TYPE_B	63
4.6.5	PTC0...7_BGO_PERIOD_L and - BGO_PERIOD_S	64
4.6.6	PTC0...7_EMU_DELAY	65
4.6.7	PTC0...7_P_STATUS register	65
4.6.8	PTC0...7_FSM_STATES	67
4.6.9	PTC0...7_EMU_TRIG_DELAY	68
4.6.10	PTC0...7 MONITORING COUNTERS.....	68
4.6.10.1	Overview	69
4.6.10.2	Description	70
4.6.11	PTC0...7_CMD pulse signals.....	72
4.6.11.1	UNFREEZE_PTC (not implemented in V000A).....	73
4.6.11.2	Restart the RANDOM TRIGGER.....	73
4.6.11.3	SOFTWARE induced BGO instructions.....	73
4.6.11.4	Start a PTC_PTC (Partition Controller).....	73
4.6.11.5	RESET_PTC	74
5	TCSMON_CHIP	74
5.1	Firmware Versions	74
5.1.1	V0003	74

5.1.2	V0002 ... from 2004	74
5.2	Description (to be updated)	74
5.3	VME_addresses	75
5.3.1	Register Overview	76
5.3.2	COMMAND REGISTER	77
5.3.3	COMMAND PULSES	78
5.3.4	STATUS_GT_CRATE	79
5.3.5	SIMULATION Control Registers	79
5.3.6	Partition Status Registers	80
5.3.7	SIMULATION Value Registers	81
5.3.8	Identifier and Version Registers	82
5.3.9	MONITORING COUNTERS	82
5.3.10	Name of Detector Partitions	85
6	VME chip	88
7	Other hardware chips	88
8	Check of Schematics against Pin-Layout	88
9	TEST Procedures	88
9.1	TCS Status Test	88
9.1.1	Setup for EVM in simulation mode	88
9.1.2	Setup for EVM in real mode	89
9.2	STATUS TEST in GT crate	90
9.2.1	EVM Status	90
9.2.2	DAQ Status	90
9.2.3	SET BOARDS to READY	90
9.3	TEST POINTS for internal TCS signals	93
9.4	TEST POINTS for internal TCSM signals	94
9.5	JTAG TEST board for external STATUS bits	95
9.6	Internal STATUS bits from TCSM chip	97
9.7	Test Program with external status bits	98
9.8	Test Program with internal status bits	98
9.9	Status Signal Decoder	98
9.10	Periodic Generator	99
9.11	Random Trigger	99
9.12	Partition Assignment	99
9.13	State Machine	100
9.14	Calibration Triggers and Throttle Logic	100
9.15	Default values for DAQ_PTC0 registers	100
9.16	Default values for Common Registers	100
9.17	BC-Table 0 and 1 with default content	101

1 Description

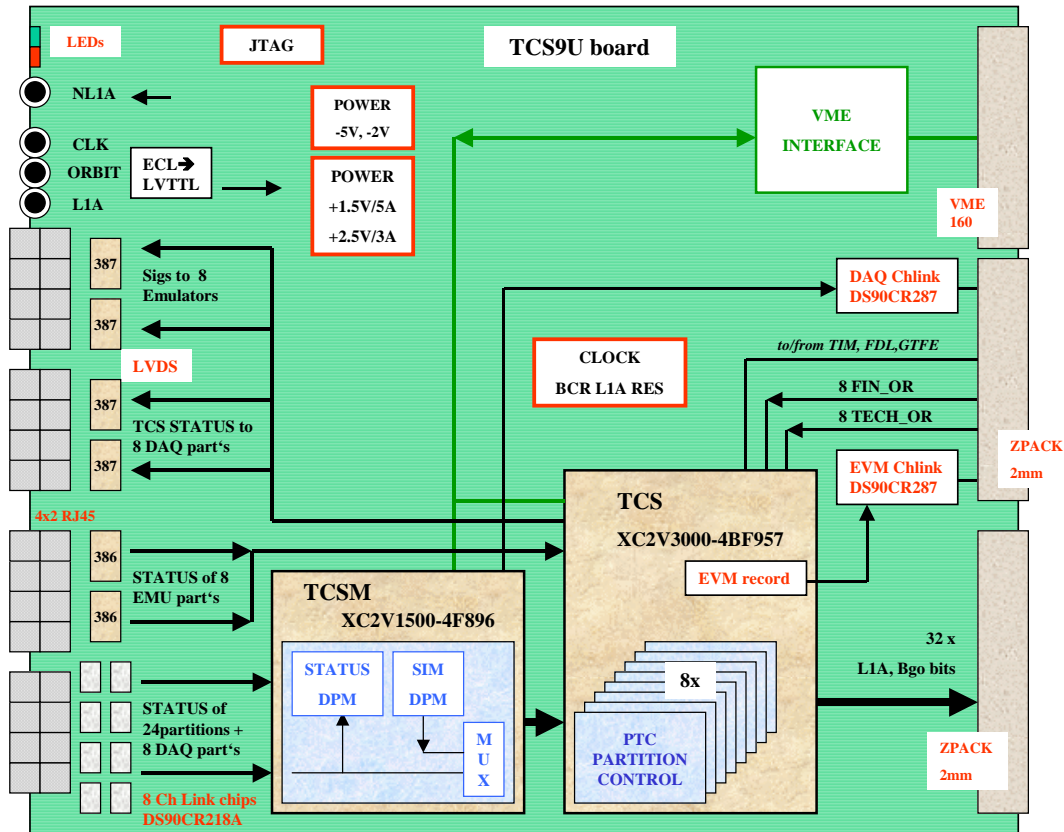


Figure 1 TCS9U board

The central trigger control is mounted in the Global Trigger rack. It consists of the central Trigger Controller module (TCS) in the Global Trigger main 9U VME crate, the CONV6U conversion modules and the Emulator Modules, for example the APVE boards emulating the behavior of the readout circuits for the Derandomizing Buffers in the Tracker electronics. The CONV6U conversion and emulator pcbs (=printed circuit board) are housed in 6U VME crates.

The central TCS

- controls the instantaneous and average rates of L1A triggers according to
 - programmed Trigger Rules,
 - emulation of the front-end buffers and
 - status of connected sub-detector partitions,
- generates the fast commands to be distributed to the sub-detectors by the TTC network (BCRES, L1RESET and other BGO commands)

- collects the status of connected sub-detector partitions and of the DAQ-partition and generates the appropriate action when necessary like inhibit triggers,
- generates calibration and test trigger sequences and
- monitors the experiment dead-time by dead-time counters

1.1 Front Panel

TCS9U front panel

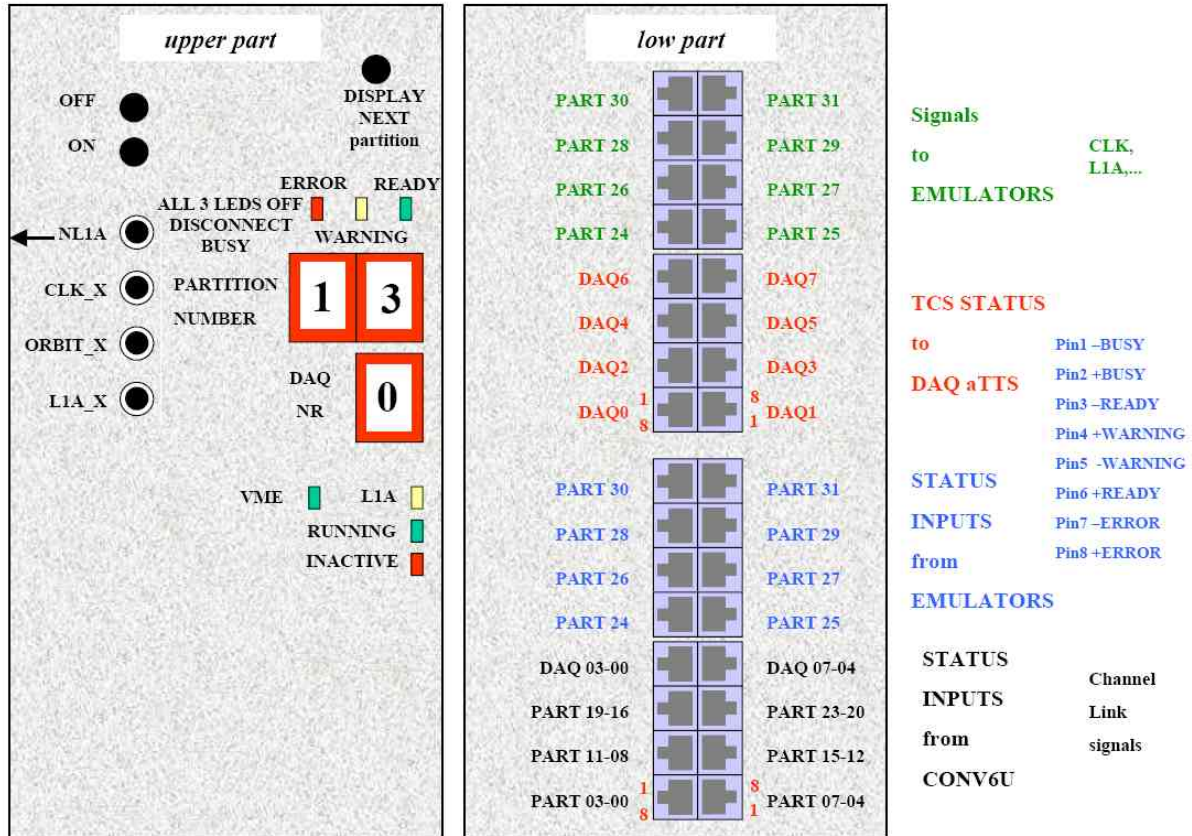


Figure 2 Front Panel

The uppermost 2 buttons are used to switch the on-board voltages. When RUNNING is on then all in- and outputs are enabled and all are disabled if INACTIVE is on.

The L1A LED flashes in case of a L1A sent to all partitions connected to DAQ0.

The VME flag shows on-board VME access cycles.

STATUS DISPLAY:

The DISPLAY button is used to advance the status display to the next partition.

Three LEDs below show either the status of the partitions and the PTCs.

red	yellow	green	status
off	off	ON	ready
off	ON	off	Warning overflow, busy
ON	off	off	error, out_of_sync
off	off	off	Disconnected

PARTITION NUMBER: (7-Segment Display)

0 ...31 detector partitions

40 ... 47 merged status input to PTC0 ...7

50 = GT_status 51 = GTFE(EVM+DAQ) status 52 = tcs_rop status

DAQ_NR: (7-Segment Display) 0 ... 7 status of PTC0 ...7

LEMO connectors:

The Lemo output NL1A is used as TESTPOINT to display internal signals of the TCS chip on an oscilloscope. See TESTMASK registers.

The LEMO input connectors CLK_X, ORBIT_X and L1A_X are not used.

1.2 Input/Output overview

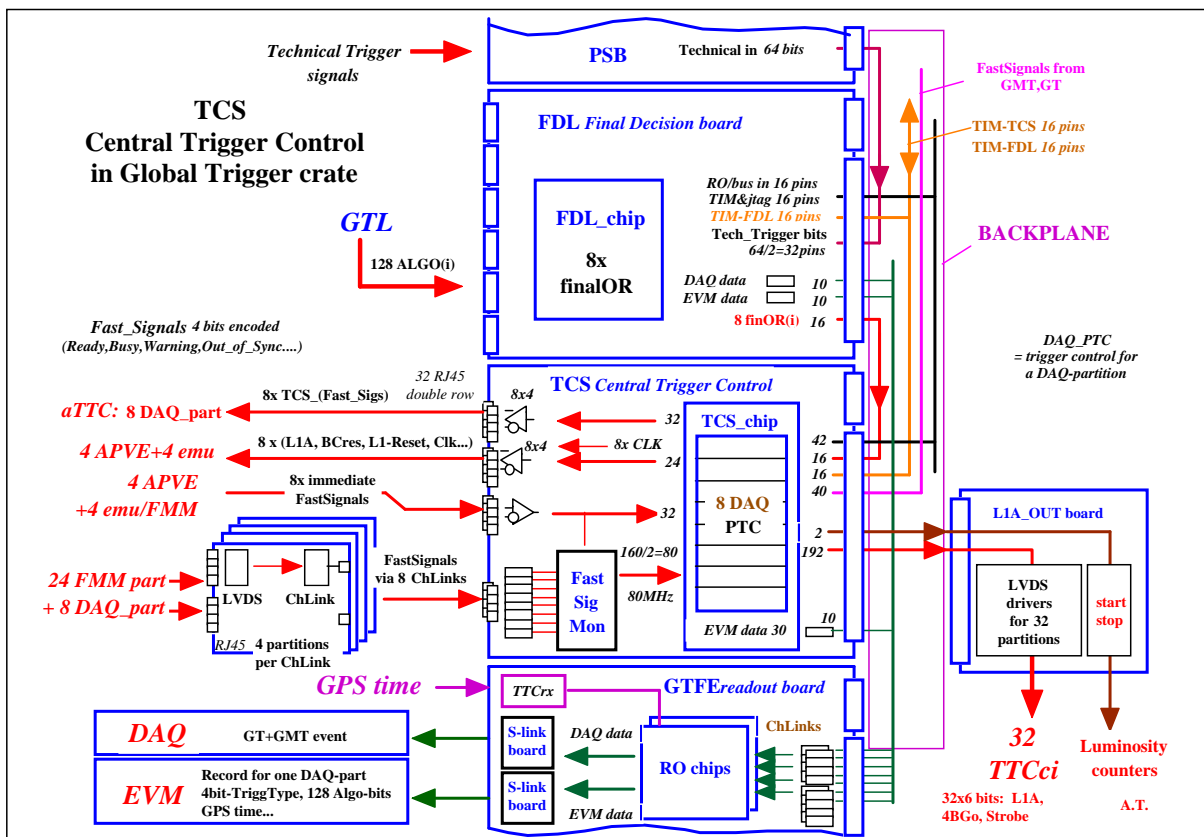


Figure 3 TCS board in the Global Trigger Crate

The central Trigger Control board TCS is connected to the following subsystems

- Global Trigger
 - 8 Final-OR signals (LVDS) transmitted via the crate backplane.
- LHC machine
 - 40 MHz Clock and the LHC-Orbit signal (two cables, ECL) are received via the TIM board of the GT crate. For test purposes, internal generation of clock and orbit signals may be selected.
- 32 TTCci modules, to broadcast L1A and control commands to the sub-detector partitions.
 - 32 cables with 8 bits sent as LVDS signals via two L1A_OUT boards
L1A (=L1Accept) starts extraction of data from pipeline- and ring buffers,
Bgo3...0, Strobe signals address 16 B-Go-circuits in the TTCci, to send broadcast commands via TTC B-channel to partitions,
Clock and **BCRES** for tests without LHC signals.
- sTTS synchronous Trigger Throttling System
 - that collects information on the status of the front-end electronics:
 - 24 cables from detector partitions and

- 8 cables from partition emulators (4 for Tracker APV-Emulators; 2 for Preshower emulators; 2 free)
A cable carries 4 encoded bits as parallel LVD-Signals using RJ45 connectors and an Ethernet cable, unshielded UTP5 or shielded STP6.
- Transmitted messages: *Disconnected, Overflow_Warning, Out_of_Sync, Busy, Ready, Error*
- Emulators:
8 cables to emulators, each sending Clock, L1A, BC0, L1Reset, Reset Event Counter and Reset Orbit Counter using 4 encoded bits, sent as LVD-Signals via RJ45 connectors and Ethernet cables.
- DAQ Event Manager
The TCS board sends event records via a Channel Link chip to the Global Trigger Front-End module (GTFE) where they are integrated into event records and sent via an S64-link to the CMS Event Manager.
The overall trigger rate is limited to 100 kHz even in multi-partition mode.
- aTTS asynchronous Trigger Throttling System
that collects information on the status of trigger electronics:
 - Input: 8 TTS channels (1 per DAQ partition)
 - Output: 8 TTS channels (1 per DAQ partition)*A cable carries 4 encoded bits as parallel LVD-Signals using RJ45 connectors and an Ethernet cable, unshielded UTP5 or shielded STP6.*
- BST system
On the GTFE board a TTCrx receives GPS time and beam information to be appended to event record for the Event Manager.
- Interface to Run Control/Detector Control Systems (VME interface).

1.3 Power

3.3 V / xx A

+ 5 V / xx A

2 Interfaces

2.1 Interface to TTC Partitions (sTTS)

2.1.1 Input: Status Signals (Fast Signals)

The list of input signals to central TCS is described in Table 1. The four bits are encoded and are sampled by the same local clock edge.

Table 1 Input signals to central TCS

2^3	2^2	2^1	2^0	PARTITION STATUS
0	0	0	0	DISCONNECTED*

0	0	0	1	WARNING OVERFLOW
0	0	1	0	OUT_OF_SYNC
0	0	1	1	bad code
0	1	0	0	BUSY
0	1	0	1	bad code
0	1	1	0	bad code
0	1	1	1	bad code
1	0	0	0	READY
1	0	0	1	bad code
1	0	1	0	bad code
1	0	1	1	Partitions 24-31: Trigger/Inhibit ** Partitions 0 - 23: bad code
1	1	0	0	ERROR
1	1	0	1	bad code
1	1	1	0	bad code
1	1	1	1	DISCONNECTED*

Notes:

*) Depends on the output level of unconnected LVDS receiver chips. The 75LVDT386 chips go into high level state if unconnected.

***) The Tracker Emulators APVE send a signal that can be used either as trigger signal or as trigger reject signal. The signal will help identifying problems in the Tracker electronics.

2.1.2 Output to TTCci - L1A and BGo-Commands

The TCS sends B-Go signals to each TTCci board to start the transmission of broadcast commands, stored in the TTCci FIFOs. The broadcast command consists of 8 bits as defined in the TTCrx Manual. Four bits (5-2) are foreseen for System Messages as shown in Table 3. By default the TTCci will forward the same code for System Messages as received from the TCS board, but could change it or add additional messages if required by the connected readout electronics.

The B-Go signals will be used for BC0, to reset event and orbit counters, to run calibration cycles (WTE, TE), to send start/stop commands, to reset hardware errors and to resynchronize the readout and trigger electronics of connected subdetectors. Four B-Go bits (3,2,1,0) address 16 B-Go circuits in the TTCci. The Strobe starts the addressed B-Go circuit. Table 2 shows the list of B-Go commands.

In addition TCS sends also the L1A and for test purposes the 40 MHz clock and a BCRES signal to each connected TTCci board. All signals go via the back-plane to L1AOUT pcbs, are converted to LVDS signals and transmitted via Hirose HR25 connectors and shielded TP cables

to the TTCci boards.

Table 2 B-Go commands sent as System Messages

Bgo channel	B-Go bits System Message code	Command	Comment
1	0001	BC0	Reset Bunch crossing counters to begin a new LHC orbit
2	0010	TE	Test Enable as 2 nd signal starting a calibration procedure, sent before the calibration trigger signal
3	0011	Private Gap	not implemented
4	0100	Private Orbit	Time for private activity
5	0101	RESYNC	Resynchronize the readout and trigger electronic: clears buffers and pipelines, counters...etc. // RESYNC=L1Reset
6	0110	HardReset	Reset Hardware errors
7	0111	EC0	Reset Event Counter sent during resynchronization procedure
8	1000	OC0	Reset Orbit Counter is sent at begin of a new data taking run
9	1001	Start	starts data taking with next orbit
10	1010	Stop	stops data taking with next orbit
11	1011	Start_of_Gap	sent every orbit
13	1101	WTE	Warning Test Enable as first signal starting a calibration procedure, sent before TE
12, 14-16			Reserved for 'private' applications

Table 3 TTC-Broadcast Commands 8bits

Bit #	Signal name	Internal action	TTCrx pin
7:6	User message	Execute 'private' messages	Brcst(7:6)
5:2	System message	Execute system messages	Brcst(5:2)
1	Event counter reset	Reset internal Event counters	EVentRes
0	Bunch counter reset	Reset internal BC-counters	BCntRES

2.1.3 Output to Emulators

Beside the CLOCK signal the TCS sends 3 encoded bits to 8 Emulators.

Actually 4 Tracker-EMULATORS (=APVE) are connected.

The other 4 emulator partitions could also used by normal detector partitions.

Table 4 Signals to Emulator

BC0	RESET	L1A	messages and trigger signal	abbrev.
2^2	2^1	2^0		
0	0	0	inactive	
0	0	1	L1A	L1A
0	1	0	RESYNC (L1RESET)	
0	1	1	RES_EVENT_COUNTER	EC0
1	0	0	BCRES	BC0
1	0	1	concurrent L1A + BCRES	
1	1	0	RESET ORBIT COUNTER	OC0
1	1	1	inactive	

Remark1: PTC inhibits L1A signals when sending L1reset or ResEvnr or ResOrbitnr.

REMARK2: Consider the delay of the BCRES (Bgo) signal when sending a Bgo command (resync, EC0, OC0) to avoid any overlap.

2.2 Interface to aTTS (DAQ)

The asynchronous Trigger Throttling System (aTTS) runs under control of the CMS data acquisition (DAQ) software and monitors the behavior of the readout and trigger electronics checking the status bits of the central partition controllers (PTC) on the TCS board. It sends also status bits back to the PTCs to reduce the trigger rate or to inhibit triggers if required.

2.2.1 Input: aTTS Status signals

The aTTS system sends the status of the DAQ partition to the central TCS board according to Table 5 Status signals from aTTS to TCS below. The codes correspond to the status signals of detector partitions 0 - 23.

Table 5 Status signals from aTTS to TCS

2^3	2^2	2^1	2^0	PARTITION STATUS
0	0	0	0	DISCONNECTED*
0	0	0	1	WARNING OVERFLOW
0	0	1	0	OUT_OF_SYNC
0	1	0	0	BUSY
1	0	0	0	READY
1	1	0	0	ERROR

1	1	1	1	DISCONNECTED*
other codes				bad code

Notes:

*) Depends on the output level of unconnected LVDS receiver chips.

2.2.2 Output: TCS-status signals

Each of the 8 partition controllers (PTC) on the TCS board sends 4 LVDS signals to aTTS encoded as shown in Table 5. The signals show the status of central TCS and are defined like Status input signals. The four signals are applied as voltage levels as long as the corresponding status is valid. The inactive level corresponds to a disconnected cable or a not initialized

TCS resp. PTC status bits:

- **IDLE**: PTC has been set-up and waits for a START of the run.
- **READY**: PTC and all connected partitions are ready. TCS allows L1A at the normal rate.
- **BUSY**: PTC or one of the connected partitions is busy (=OR of input BUSY signals). PTC inhibits L1A until end of busy status.
- **WARNING**: PTC either inhibits L1A or is running with reduced trigger rate because one of the connected partitions is applying a WARNING.
- **DISCONNECTED**: One of the connected partitions has applied DISCONNECTED. The PTC waits until the partition becomes reconnected to continue broadcasting L1A signals. If the partition does not become reconnected again the software has to send a STOP_RUN command to stop data taking.
- **OUT_OF_SYNC**: One of the connected partitions has applied an Out_of_Sync (=OR of Out_of_Sync signals). The PTC is waiting for software interaction to run a RESYNC procedure.
- **ERROR**: One of the connected partitions has applied an Hardware ERROR. The PTC is waiting for software interaction to run a HARD-RESET procedure.

Table 5 Status signals from central TCS to aTTS (DAQ)

Ready	Busy	OutofSync	Warning	TCS STATUS
0	0	0	0	DISCONNECTED*
0	0	0	1	WARNING
0	0	1	0	OUT_OF_SYNC
0	1	0	0	BUSY
1	0	0	0	READY
1	0	1	0	IDLE
1	1	0	0	ERROR

1	1	1	1	DISCONNECTED*
other codes				bad codes

Notes:

*) Depends on the output level of unconnected LVDS receiver chips.

2.3 Interface to GT processor

The TCS board sends L1A and the BGO signals for the GT crate also via the back-plane to the TIM board to run the GT without any TTCrx connection. The CLOCK and BCRES signals are then taken from the ECL inputs of the TIM module.

2.3.1 Final_OR and Tech_OR

The FDL board sends 8 Final_OR signals as differential signals via the back-plane to the TCS board.

2.3.2 GT status signals

The Global Trigger Crate runs as one partition. The status of the GTFE-read-out-board is sent directly to the TCS-MON chip. The status bits of the other boards in the GT crate are combined in the FDL board to a 4-bit status nibble and go as 'GT-STATUS' to the TCSM chip. The TCSM chip monitors the status signals and sends them to the control chip TCS. It could also simulate states of the partitions for tests.

By default the GTFE status (EVM & DAQ chip) is connected to all 8 Partition Controllers to inhibit the L1A signals.

2^3	2^2	2^1	2^0	GT/GMT STATUS
0	0	0	0	DISCONNECTED*
0	0	0	1	WARNING OVERFLOW
0	0	1	0	OUT_OF_SYNC
0	1	0	0	BUSY
1	0	0	0	READY
1	1	0	0	ERROR
1	1	1	1	DISCONNECTED*
other codes				bad codes

Table 6 Status signals of GT boards to the central TCS board

2.3.3 TCS \leftrightarrow FDL

16 FIN-OR pins for 8 differential signals. TCS \leftarrow FDL

16 TECH TRIG pins for 8 differential signals are not used. TCS \leftarrow FDL

GT_STATUS[3:0] GT crate status bits TCS (TCSM chip) \leftarrow FDL

Used by FDL:

```

FDLUTCS[3] := FREEZE_MON // TCS → FDL freeze trigger counters (not implemented)
FDLUTCS[2] := NEW_LUM_SEG // TCS → FDL 'new luminosity segment' pulse to reset
trigger counters.
FDLUTCS[1] := 0 // FDL → TCS not used
FDLUTCS[0] := 0 // FDL → TCS not used

```

Not used by FDL:

FPGA and board layout done for a differential signal

```

FDLTCS[3] := FREEZE_MON // TCS → FDL freeze monitoring (not implemented)

```

```

FDLTCS[2] := NEW_LUM_SEG // TCS → FDL 'new luminosity segment' pulse to reset
trigger counters.

```

```

FDLTCS[1] := 0 // FDL → TCS not used

```

```

FDLTCS[0] := 0 // FDL → TCS not used

```

2.3.4 TCS ↔ TIM

```

TIMTCS[15:8] TCS → TIM // see TTC_OUT in TCS chip

```

The signals are used to run the GT crate without the TTCrx (=default option).

CLK and BCRES are then taken from ECL inputs on TIM board.

The GT_BGo[] and L1A come from the DAQ-partition to which the GT has been assigned by the ASSIGNMENT register ASSIGN_LUM_GT_ENIO at address 480030 .

See also in 4.4.3.

```

15 L1A_FROM_TCS // L1A for GT crate to run without TTCrx

```

```

14 0 // wired to GND in TCS chip

```

```

13 0 // wired to GND in TCS chip

```

```

12 GT_BGO4 // BGO-STROBE signal

```

```

11 GT_BGO3 // BGO command bits for GT crate

```

```

10 GT_BGO2

```

```

9 GT_BGO1

```

```

8 GT_BGO0

```

```

TIMTCS[7:0] TCS ← TIM // seen as COMMON STATUS[7:0]

```

Not used by TCS. The TIM sends encoded status bits to FDL where they are merged with other GT boards.

```

7 TIM_ERR // TIM Status bit

```

```

6 TIM_OUT_OF_SYNC // TIM Status bit

```

```

5 TIM_WARNING_OVFLO // TIM Status bit

```

```

4 TIM_READY // TIM Status bit

```

```

3 TIM_BUSY // TIM Status bit

```

```

2 L1_RES //also sent as LVDS

```

```
1 TI_INHIBIT_PHYS_L1A // undefined utilization
0 TI_INHIBIT_ALL_L1A // undefined utilization
```

3 Trigger Control logic

3.1 Synchronization to LHC orbit

The TIM board sends the 40 MHz clock and the BCRES signal via the GT-backplane to the TCS board. A programmable delay on the TIM board with an adjustment range of 3563 bunch crossings allows even to apply a negative delay to the BCRES signal. In the TCS chip one delay for the BCRES signal resets the address counter of the BC-Tables, where all the control commands, calibration and test trigger signals are stored to be sent at defined bunch crossings to the connected subdetector electronics.

A second delay in the TCS chip for the BCRES signal provides the reset signal for a bunch crossing counter. The content of the bc-counter is appended to the event record and defines the BC number of an event.

The same delayed BCRES signal increments the Orbit Counter and after a preselected number of orbits also the Luminosity Segment Number counter.

The 12 bit BC number, the 32 bit Orbit number and the 16 bit Luminosity Segment Number identify an event for data analysis.

Remark: The actual orbit counter length of 32 bits covers a period of 106 hours = 4.4 days. If required the orbit counter could be extended to more bits to cover a very long period.

3.2 DAQ-Partition controllers (PTC) overview

The Trigger Control logic reflects the optional segmentation of the CMS readout system into 8 **DAQ-partitions**. The Global Trigger Processor therefore generates up to 8 Final-OR signals in parallel and the TCS chip contains 8 DAQ-Partition Controllers (DAQ-PTC, shorter abbreviation PTC) running independently from each other. The Run Control Software can start and stop the PTCs without any restrictions except that only one DAQ partition is allowed to trigger at a given bunch crossing.

The front-end and trigger electronics is configured in up to 32 **detector partitions** according to 32 TTC-trees. All crates connected by TTC-fibres to the same TTCi board belong to the same detector partition. Each detector partition can be connected only to one DAQ-partition at a time. The corresponding DAQ-Partition Controller (=PTC) accepts the Status Signals of the connected partitions and provides the L1A signal and BGo-commands (calibration, ...).

For example the RPC-pos-endcap and pos-CSC partition could be connected to PTC2 to run an alignment procedure and all 4 Tracker and all 6 ECAL partitions could be connected to PTC3 to do calibration measurements.

For a normal physics run PTC0 controls all partitions and uses the Fin-OR0 signal as the common physics trigger.

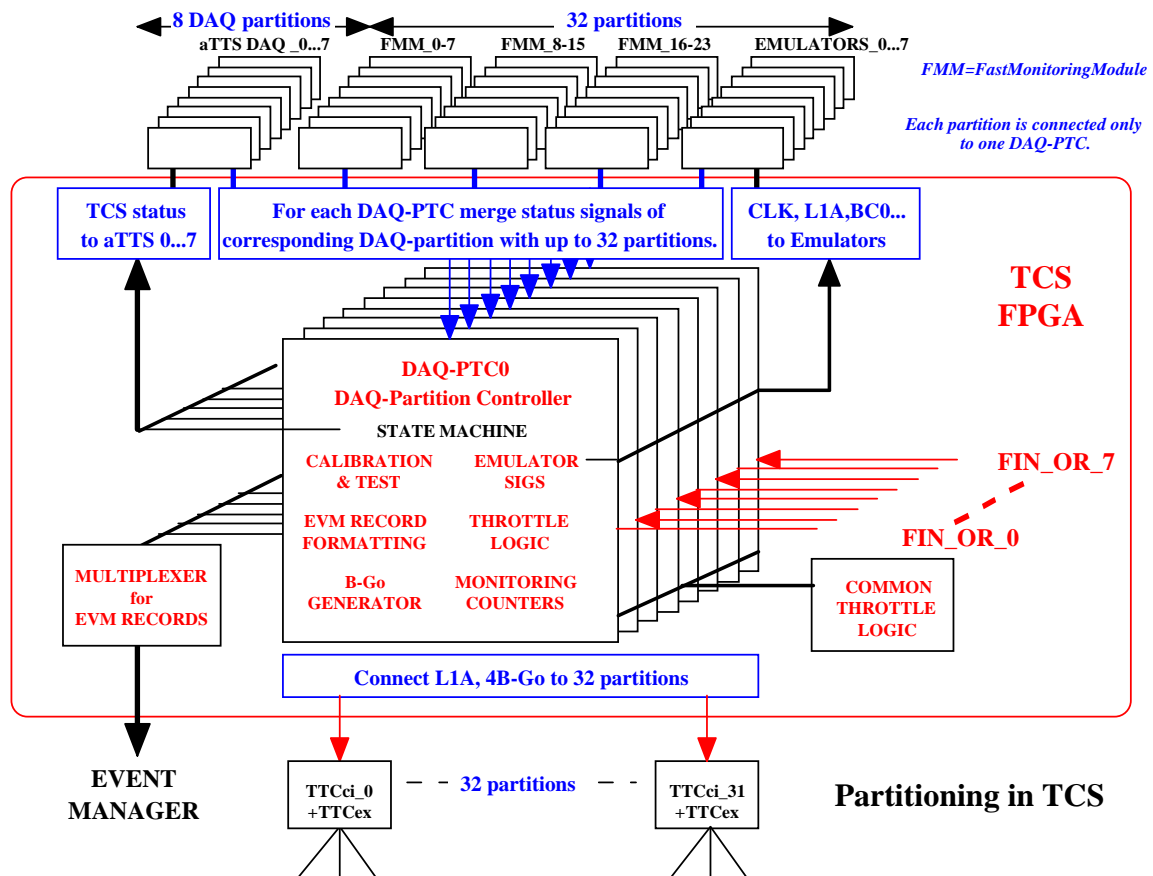


Figure 4 Partitioning of the Trigger Control System

3.3 Time Slice distribution

The Trigger Controller distributes the beam time between the active DAQ-partitions activating them consecutively for programmable periods of time. It distributes time slices in round robin mode with a precision of about 0.5%. The shortest period lasts 10 orbits and the longest 2550 orbits. During inactive periods a PTC inhibits L1As and calibration cycles but still sends control commands to the partitions and monitors and accepts also the status signals as usual.

Remark: A priority scheme for L1A has not been implemented to avoid data analysis problems.

3.4 Trigger Throttle logic

A common TTS (trigger throttle) circuit cuts excessive instantaneous trigger rates for all partitions to avoid problems in the readout system. The throttle circuit accepts L1A signals from all partitions and sends the inhibit signal also to all partitions.

The rate reduction follows two sets of rules, each set consisting of 4 rules. The 1st rule defines the minimum time between 2 consecutive triggers. The other rules allow ‘n’ triggers within a programmable period. The default rules for normal data taking as shown in last but one column of Table 7 will introduce less than 1% dead time. The rules for the ‘Overflow

Warning' state allow fewer triggers within the same periods. Alternatively the 'Overflow Warning' state could suppress all triggers completely (programmable option).

Table 7 Throttle rules

Rule - number	Δt (in nr of BC)	Max of # L1A	Default values (used in CMS)		
			Δt (in nr of BC)	# L1A normal rate	# L1A low rate
1	2 ... 7 *	1	2	1	1
2	1 ... 63	15	25	2	1
3	64 ... 176	63	100	3	2
4	1 ... 2047	511	240	4	2

*) Nr of BC without trigger

3.5 STATUS bits - Decoder and Merger

The TCS chip accepts encoded status bits, often called 'Fast Signals', from 32 subdetectors also called 'detector partitions' or simply 'partitions' as described in the "Interface" chapter.

Encoded input states (4 bits):

0 = F=disconnected 1= warn
 2=out_of_sync 4=busy
 8=ready C=error **B= trigger (partitions 24-31 only)**

First the decoder waits that a new signal state becomes steady for 50 ns to cancel spurious pulses. Then it decodes the 4 bits into 7 states shown in Table 8.

Table 8 Decoded status bits to PTC

7	6	5	4	3	2	1	0
trigger	bad code	disconnected	error	out_of_sync	busy	warning	ready

Then states of the partitions, which are connected to a partition controller (PTC), are combined to one status, the ready states by an AND function and all other states by OR functions. The error and warning status of the Event Manager interface from the GTFE readout board is connected to all Partition Controllers (=PTC). Then priority logic forwards the status of the highest priority according to the following priority chain:

Disconnected > bad_code > error > out_of_sync > busy > warning > ready

'Bad_code' and 'disconnected' can be ignored optionally.

3.6 DAQ-Partition Controller - details

The PTC provides all functions to run a group of detector partitions independently and consists of the following modules:

- **A PTC State machine** runs the control procedures according to the states of the connected partitions,
- **A BGO Generator** defines the time behavior when a broadcast command has to be sent to the readout and trigger electronics,
- **A BC Table** defines the BC numbers when BGO commands and calibration triggers will be sent. It also defines the gaps within a LHC orbit without particle collisions.
- **A Calibration logic** runs calibration cycles
- **A Random Trigger Generator** delivers Poisson-like distributed trigger signals.
- **A Trigger Merger** combines all trigger sources (Fin_OR, Random Trigger, Calibration Trigger, Test Trigger) into a L1A signal,
- **Periodic Signal Generator** for Private Orbit commands and periodic Calibration cycles.

3.6.1 State Machine of PTC

The central TCS handles all the status signals received from the subdetector partitions and from the aTTS with a state machine programmed either to stop L1A signals or to deliver Reset and other BGo commands signals, if needed.

The diagram below explains the behavior of the state machine.

3.6.1.1 Internal STATES of PTC

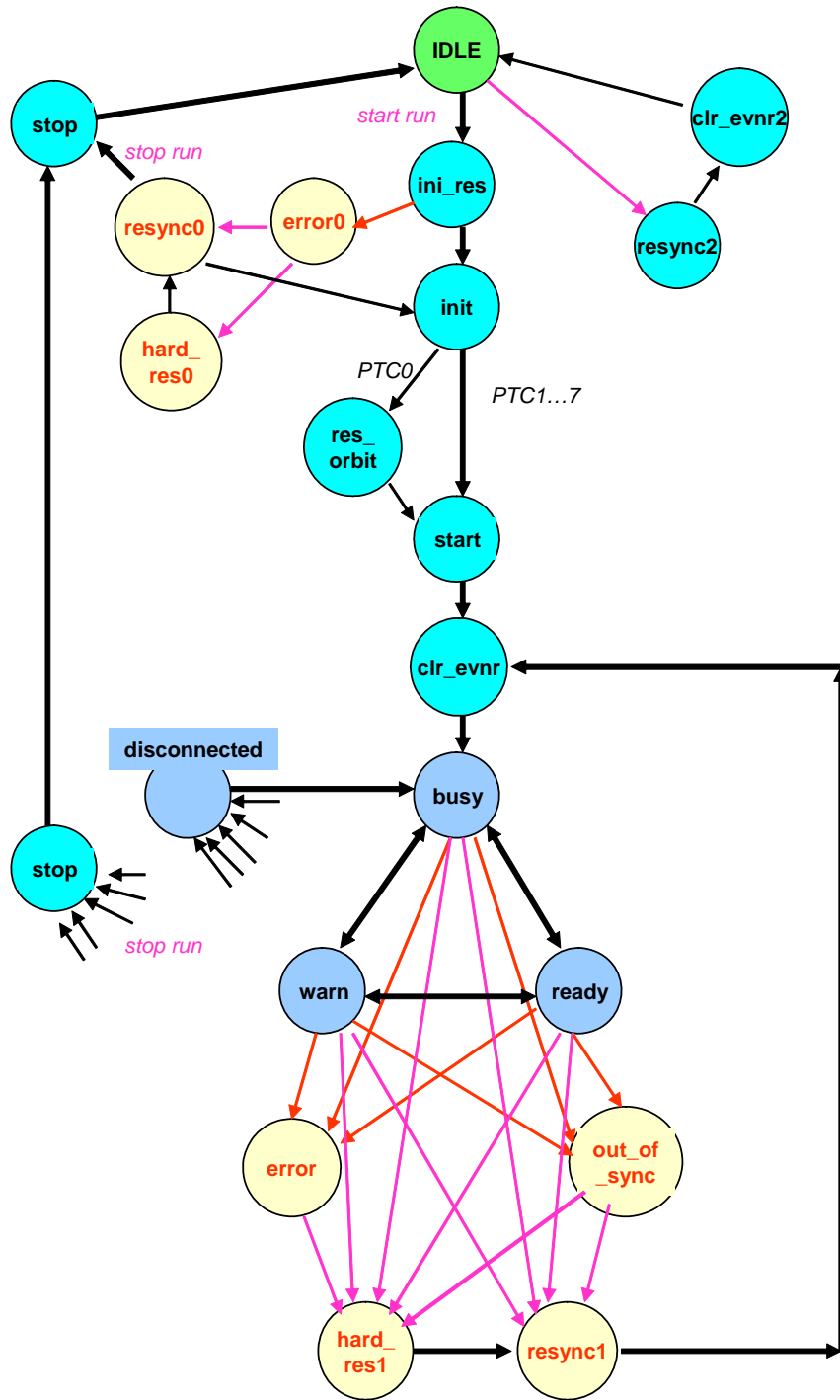


Figure 5 TCS PTC-State Diagram.

- **IDLE STATE:** Initial state after configuration. PTC waits for Run Control Software
- **INI_RES:** PTC sends BGo=Resync signal to initialize the partitions.
- **RESET ORBIT COUNTER:** PTC0 sends BGo= OC0 ('Reset Orbit Counter'); PTC1...7 skip this state.
- **START:** sends BGo= 'Start' starting a data taking run
- **RESET EVENT COUNTER:** PTC sends BGo= EC0 ('Reset Event Counter')
- **BUSY:** PTC inhibits L1A and waits until all partitions are ready or at least in warning state.
- **READY:** PTC runs with normal trigger rate.
- **WARN:** PTC runs with low trigger rate.
- **DISCONNECTED:** PTC waits until partition(s) are reconnected and returns then to the BUSY status.
- **ERROR:** When a partition is in 'error' then the PTC waits for a Trigger Supervisor command either to stop the run or to send BGo= 'HardwareReset' to remove the hardware error.
- **OUT_OF_SYNC:** When a partition is in 'out_of_sync' then the PTC waits for a Trigger Supervisor command either to stop the run or to send BGo= 'HardwareReset' or to send BGo= 'Resync' to remove the out_of_sync error.
- **HARDRES0, 1:** PTC sends BGo= 'HardwareReset' to partitions.
- **RESYNC0, 1:** PTC sends BGo= 'Resync' to partitions.
- **ERROR0:** If the initial 'Resync' command is not successful then the PTC waits for a Trigger Supervisor command either to stop the run or to send BGo= 'HardwareReset' or to send BGo= 'Resync' to remove the hardware error.
- **STOP:** Whenever the Trigger Supervisor sends a 'stop run' the PTC sends a BGo= 'StopRun' to the partitions and returns to the IDLE state.

3.6.1.2 *Response to TTS signals*

- Partition sends READY (subsystem accepts L1A)
 - o PTC runs normally broadcasting trigger (L1A) signals
- Partition sends BUSY (subsystem cannot accept L1A)
 - o PTC sets BUSY flag for DAQ and inhibits L1A.
 - o If BUSY disappears and either READY or WARNING becomes active then PTC allows L1A again.
- Partition sends WARNING
 - o PTC sets WARNING flag and either reduces L1A rate or inhibits trigger signals

- If WARNING disappears then PTC allows normal L1A rate again
- Partition is DISCONNECTED (setup not done, cables removed...)
 - TCS-hardware inhibits L1A for this partition group and sets Disconnected flag and waits until DISCONNECTED has been removed or until the TS (Trigger Supervisor software) sends a 'stop_run' signal .
- Partition sends OUT OF SYNC (sync loss in subsystem)
 - TCS-hardware inhibits L1A for this partition group and sets the OUT OF SYNC flag and waits for a TS-command to start a Resync or HardRes procedure or to stop the run.
 - If the Resync procedure is not successful TCS returns via the BUSY status into the OUT_OF_SYNC status. Then the TS can retry or stop the run.
- Partition sends ERROR (error in subsystem)
 - TCS-hardware inhibits L1A for this partition group and sets the ERROR flag and waits for a TS-command to start a HardRes procedure or to stop the run.

3.6.1.3 RESYNC and HARDWARE RESET procedure

When the PTC receives an "OUT_OF_SYNC" status from a connected partition it enters its "OUT_OF_SYNC" state and waits until a "Resync" command from the Trigger Supervisor software.

First the PTC waits a programmable time (nn orbits = called 'settle time') to allow the subdetectors to forward event data as requested by the CMS Data Acquisition (DAQ).

Then during the next orbit as defined in the BC-table the PTC sends the RESYNC (Bgo=0101) command to the connected subdetectors.

Now the PTC waits again a programmable time (nn orbits = called 'recover time') to allow the subdetectors to finish their resynchronization tasks.

Then the PTC sends the ECO (Bgo=0111) command as defined in the BC-table to reset the event numbers in all connected subdetectors.

Finally the PTC enters the BUSY and then the READY state to continue data taking but only if all connected subdetectors are also ready.

In case of a ERROR the PTC also waits until the Trigger Supervisor software sends a "Hardware Reset" command. Then during the next orbit the PTC sends a HARDRES (Bgo=0110) command to all connected subdetectors and enters the afterwards the RESYNC procedure as described above.

3.6.2 Calibration circuit

As defined in a BC-Table one or several calibration cycles can be done during an orbit and are executed either every orbit or inserted periodically every n-th orbit. First the calibration controller sends

- a WTE ('WARN_TEST_ENABLE') for example to prepare Lasers.

- Then a TE ('TEST_ENABLE') command starts the calibration procedure in the sub-detectors.
- Finally the following L1A reads the calibration data.

The time periods between the BGo_commands and the L1A are defined in the BC-Table. Between the TEST_ENABLE command and the following calibration trigger other triggers are inhibited and corresponding dead-time counters are incremented.

3.6.3 BC Table

The BC table consists of two 4kx16bit dual port memories. Each bit defines a Bgo function or a trigger signal.

Bit 0:	valid_bc	// defines valid bunch crossing with data
Bit 1:	bcres	// sent every orbit
Bit 2:	test_trigger	// period test trigger, useful to find local trigger latencies
Bit 3:	private_gap	...not used
Bit 4:	private_orbit	// foreseen for private activity of subdetectors
Bit 5:	Resync	// does a resynchronization procedure
Bit 6:	Hardware Reset	// rests the hardware
Bit 7:	EC0 Reset Event number	// is sent a begin of a run and after a resync procedure
Bit 8:	OC0 Reset Orbit number	// is sent at begin of a data taking run
Bit 9:	Start run	// starts data taking
Bit 10:	Stop run	// stops data taking and PTC goes to idle status
Bit 11:	Private Bgo – strobe	// foreseen for special tests
Bit 12-15:	Private Bgo code	// foreseen for special tests
Bit 16:	Calibration Trigger	(calibration cycle)
Bit 17:	Test_Enable	(calibration cycle)
Bit 18:	not used	
Bit 19:	not used	
Bit 20:	Warning_Test_Enable	(calibration cycle)
Bit 21:	Start_of_Gap	// defines the begin of the big gap to suppress triggers
Bit 22:	End_of_Cal_Cycle	(calibration cycle)
Bit 23 – 31:	...	not used

Side B that is accessed by software to load the bits for Bgo commands and triggers as required.

The A-side of the BC-table memory is accessed permanently in read mode by an address counter, that is reset by a delayed BCRES signal so that the address corresponds always to the LHC Bunch Crossing Number. *See above the first paragraph in chapter "Synchronization to LHC orbit".*

Therefore all BGO commands and the calibration- and test triggers are generated at defined BC numbers synchronously to the LHC orbit. Whenever a BGo command should be sent

during an orbit the BC-table bit is used to apply the corresponding BGo code to the outputs or to send a calibration- or test trigger signal.

3.6.4 Trigger merging

The periodic processor contains a priority logic that gives the test trigger periods the highest priority. In case of competing periodic BGo commands or periodic triggers than the priority chain below is valid. If an activity is generated every orbit then the lower priority activities are completely suppressed.

Priority chain:

Test_trigger > private_bgo > private_orbit > calibration_trigger > traced events

Actually test triggers are not included in the priority chain and can be interlaced with all other trigger sources except the calibration trigger.

The VHDL code below shows the behavior of the merging logic:

```

-- INHIBIT signals
-- inh_low_rate    <--throttle logic
-- inh_norm_rate   <--throttle logic
-- inh_l1a_sm      <--ptc_sm (busy,...)
-- inh_l1a_cal     <--bgo_ctrl : during calibration cycle
-- inh_l1a_priv    <--bgo_ctrl : during private orbit (=1)
-- inh_l1a_test    <--bgo_ctrl : during test-trigger orbit (=1)
-- low_rate        <--ptc_sm (warning)
-- norm_rate       <--ptc_sm (ready)
-- disable_finor   <--VME
-- active_bc       <-- valid_bc of bc-table or active_beam signal from beam counters

-- Priorities:
-- 1.) Testtrigger(periodic or VME) runs exklusively ///Actually set to =0
-- 2.) Calibr-trigger stops fin_or and random_trig.
-- 3.) Finor has priority over random: trig_typ
-- 4.) random triggers can be mixed with finors

--From BGO_ctrl module:
inh_l1a_test <= inh_l1a(0); -- Test trigger cycle inhibits other triggers //Actually set to =0
inh_l1a_priv <= inh_l1a(1); -- Private orbit inhibits triggers
inh_l1a_cal  <= inh_l1a(2); -- Calibration Cycle inhibits other triggers
inh_l1a_stop <= inh_l1a(3); -- No trigger when not running
inh_l1a_bcr  <= inh_l1a(4); -- Inhibit triggers +/-2bx around Bgo //Actually set to =0

-- Suppress triggers when en_time_slot = 0.
inhibit_all  <= inh_l1a_sm or (not en_time_slot) -- StateMachine or TimeSlotGenerator
              or inh_l1a_stop                -- Stop RUN inhibits all triggers.

```

```

        or inh_l1a_bcr;          -- inhibit L1A +/-2bx around BGo commands BC0, Start_of_Gap
throttle_inhibit <= (inh_low_rate and low_rate)
        or (inh_norm_rate and norm_rate);  -- Throttle Logic
tp_thro_inh <= throttle_inhibit;        -- to test point
inhibit_testtrig <= inhibit_all or throttle_inhibit -- ptc+timeslot+throttle
        or (not en_test_trig)          -- not enabled by VME
        or inh_l1a_cal;                -- during calibration cycle

inhibit_caltrig <= inhibit_all          -- ptc+timeslot;ignores throttle rules
        or inh_l1a_priv;              -- during Private Orbit

inhibit_finor <= inhibit_all or throttle_inhibit
        or (not active_bc)            -- BC is not valid resp. no beam collision
        or disable_finor              -- disabled by VME
        or inh_l1a_priv                -- during Private Orbit
        or inh_l1a_test                -- during test-trigger Orbit
        or inh_l1a_cal;                -- during calibration cycle

inhibit_random <= inhibit_all or throttle_inhibit -- ptc+timeslot+throttle
        or (not active_bc)            -- V0017: BC is not valid resp. no beam collision
        or (not en_rndm_trig)         -- not enabled by VME
        or inh_l1a_priv                -- during Private Orbit
        or inh_l1a_test                -- during test-trigger Orbit
        or inh_l1a_cal;                -- during calibration cycle

inhibit_errtrig <= inhibit_all or throttle_inhibit -- ptc+timeslot+throttle
        or (not en_err_trig);         -- not enabled by VME

test_trig_p <= test_trig and (not inhibit_testtrig);
cal_trig_p <= cal_trig and (not inhibit_caltrig);
fin_or_p <= fin_or and (not inhibit_finor);
rndm_trig_p <= rndm_trig and (not inhibit_random);
err_trig_p <= err_trig and (not inhibit_errtrig);
l1a_p <= test_trig_p or cal_trig_p or fin_or_p or rndm_trig_p or err_trig_p;

l1a <= l1a_p; -- output to L1AOUT board --> TTC system
l1a2emu <= l1a_p; -- output to EMULATORS
l1a2throttle <= l1a_p; -- output to THROTTLE LOGIC

```

3.7 MONITORING COUNTERS

After every luminosity-segment-period the contents of the trigger counters are stored in read only registers and the counters are then reset. The software reads therefore the **number of triggers per luminosity-segment**.

The dead-time-counters are cleared only at begin of a run and their contents are saved in read-only registers at the end of every luminosity-segment-period. Therefore software reads the **accumulated dead-time** from start of the run until the end of the previous luminosity-segment-period.

The trigger number and event number show the actual value independently from luminosity-segment-periods since start of run respectively since the last ‘resynchronization’ procedure.

3.8 BGo command codes

1=0001	BC0 / BCRES	8=1000	OC0 (reset orbitnr)
2=0010	TE (test enable)	9=1001	START
3=0011	PRIVATE GAP	A=1010	STOP
4=0100	PRIVATE ORBIT	B=1011	START_OF_GAP
5=0101	RESYNC	D=1101	WTE (warn_test_enable)
6=0110	HARD RESET		
7=0111	EC0 (reset eventnr)		

3.9 OUTPUT to TTCci boards

The assignment register bits are used to send the L1A and the BGO signals to the connected sub-detector partitions. The bits are sent to the L1AOUT boards, converted to differential signals and sent to the TTCci boards using HR25 connectors from Hirose.

3.10 Event Record for Event Manager (EVM)

Bits 27 to 24 of the Channel Link are used for control information. The first 4 words show hex ‘A’ to ‘D’ and the last 4 words ‘E’ in the most significant bits. Bits 23 to 16 are unused and bits 15 to 0 carry event data.

Table 9 EVM record via Channel Link from TCS to GTFE

27 -24	23 -20	19 -16	15-0	Remark	
A	0	0	LUMINOSITY_SEGMENT_NR(15:0)	Word0	
B	0	0	X"0",DAQNR(3:0),TTYPER(3:0),STATUS(3:0)	Word1	
C	0	0	X"0",BCNR 11:0	Word2	

D	0	0	Board-ID (15:0)	Word3	
1	0	0	ASSIGNED PARTITIONS (15:0)	Word4	
1	0	0	ASSIGNED PARTITIONS (31:16)	Word5	
1	0	0	PART_RUN_NR(15:0)	Word6	
1	0	0	PART_RUN_NR(31:16)	Word7	
1	0	0	EVENTNR (15:0)	Word8	
1	0	0	EVENTNR (31:16)	Word9	
1	0	0	PART_TRIGNR(15:0)	Word10	
1	0	0	PART_TRIGNR(31:16)	Word11	
1	0	0	ORBITNR 15:0	Word12	
1	0	0	ORBITNR 31:16	Word13	
1	0	0	(ORBITNR 47:32)	Word14	optional
1	0	0	FREE	Word15	
E	0	0	<i>RESERVED</i>	Word16	'E' = last W64
E	0	0	<i>for</i>	Word17	'E' = last W64
E	0	0	<i>other</i>	Word18	'E' = last W64
E	0	0	<i>Data words</i>	Word19	'E' = last W64
F	0	0	X"FFFF"		'F' = end of record
5	5	5	X"5555"	IDLE	Between records

DAQNR = PTC number that has sent this L1A

TTYPER = trigger type according to TRIG_TYPE_A, _B registers.

STATUS = actual status of PTC from which this L1A has been sent.

ASSIGNED_PARTITIONS(i) = 1= connected, 0= not connected to this PTC

4 TCS chip

The TCSMON chip has to run synchronously to the TCS chip. It accepts the same VME commands (RESET, HARDRES, FREEZE...), the same Partitioning Assignment, the same Orbit Length etc. The TCSMON also receives the same synchronization signals from the Backplane. Therefore some registers are duplicated in the TCSMON chip but are read back only from the TCS chip.

The additional registers of the TCSMON chip are described in chapter 5.

4.1 Firmware Versions

4.1.1 V1023

- The backplane signals BCRES_from_TIM and L1RES_from TIM are decoded to get

the BCRES signal as in other chips.

- This Version receives Trigger/Inhibit Signals from the Tracker Emulators. A status code of "1011"="X"B" is expected from the APVE Emulator boards. With the new registers "PTC0...7_EMU_TRIG_DELAY" one can delay the Trigger/Inhibit signals for up to 48 bc as described in the register description below.

4x 00C0 PTC0...7_EMU_TRIG_DELAY ...new address

Also a new "trigger_type" has been defined in registers PTCx_TRIG_TYPE_B(15:12) with a default value X"8" = B"1000".

PTC0...7_TRIG_TYPE_B(15:12): EMULATOR Trigger type= "1000"

The new Trigger/Inhibit signal also generates dead time which is monitored by a 48 bit counter. The corresponding registers for PTC0 are defined in the table below:

40 00B0	PTC0_EMU_DEADTIME_47_32	bits 47 - 32	PTC0 only	
40 00B2	PTC0_EMU_DEADTIME_31_16	bits 31 - 16	PTC0 only	
40 00B4	PTC0_EMU_DEADTIME_15_0	bits 15 - 0	PTC0 only	

In the CMD_REG one can define how to apply the Trigger/Inhibit signal to the TCS trigger logic in the VHDL-Module "trigger".

PTC0...7_CMD_REG(5:4):

- = 00 ...Emulator Trigger Signal is ignored
- = 01 ...Emulator Trigger Signal used as TRIGGER
- = 10 ...Emulator Trigger Signal inhibits FINOR signal
- = 11 ...Emulator Trigger Signal inhibits FINOR signal

4.1.2 V1022

```
NEW Mentor and Xilinx SOFTWARE VERSIONS ### FPGAAdv8.2 & ISE10.1 ###
-- All memories and fifo redone with ISE10.1
-- common_cmd_pulse(5) = 'res_orbit_vme' removed
-- ptc_cntrs: Trigger & eventnr. return to 0 after a counter overflow!!
-- ptc_cntrs: accept trigger also at begin of luminosity segment.
-- trigger : 'inhibit fdl rate counters' 3bx before inhibiting the finor signal
```

4.1.3 V1021

```
BC_TABLE1(16) =END_OF_CALCYC // 1 ...removes suppression of FinOR trigger signals
```

TCS sends a new signal to the FDL board inhibiting the Rate Counters during a calibration cycle to run the trigger counters on both boards synchronously.

4.1.4 V1020

"tim_signals2009" : bcnr_rop & orb_nr_rop & lum_seg_nr switch concurrently to get correct values in event records generated at the end of LHC orbits.

4.1.5 V101F

- TEST_MASK0...7: bit13:= bgo_strobe
- "bgo_ctrl": STATE MACHINE for >>> Hardres, L1Reset <<<<
HardReset Recover time is extended to 8*256 orbits ~181 ms by increasing the counting unit to 8 orbits.
- "calibr_sm": caltrig is sent immediately at same time as test triggers(as in previous versions)
- vme_registers_tcs: Power-up values are taken from tcs_types.tcs_init_val_pkg.vhd and are defined in ... \tcs_chip\ucf_files\tcs_reg_list.xls

4.1.6 V001E

Calibration cycle ends either 255 BC after caltrig or as defined by

BC-Table1 bit 22 := 'end_of_cal_cycle' and inhibits other triggers meanwhile.

HardReset Recover time is extended to 4*256 orbits ~91 ms by increasing the counting unit to 4 orbits.

Calibration Trigger is never inhibited when cal_cycle has been started.

4.1.7 V001D

ERROR TRIGGER is not inhibited by ptc_sm.

4.1.8 V001C

bgo_ctrl:

New calibr_sm (state machine with timeout=3563 BC)) generates calibration cycles.

Cal_cycle is started when PTC is ready but completes regardless of PTC status.

Other L1A's are allowed between WTE ('warn_test_en') and TE ('test_en') commands, but inhibited between TE ('test_en') and the calibration trigger signal.

tim_signal_decoder: res_orbit_cntr simplified as in FDL chip.

VME logic: vme_enn with vme_en1,2,3

4.1.9 V001B

*** No change in software required. ***

- FinOR, Random and Test Triggers are allowed simultaneously. The Event type of FinOR prevails over both other types. The Random Trigger type prevails over the Test trigger type.
- FinOR and RANDOM Triggers are allowed as defined in the BC-table by bit0 (= valid_bc) which is programmed to allow long lived particles with only one gap to send the BC0 Bgo command.
- For dead-time counters the ACTIV_BEAM signal is defined alternatively either by the BC-table(bit0 = valid_bc) or by signals from the beam pick up detector BPTX. These signals come from the PSB board in VME-slot 9 from the parallel input bit 0.

4.1.10 V001A

- Lost_trigger equation corrected.
- 100 Hz rate corrected ('periodic_proc').

4.1.11 V0019

lumi_cntr: ld_new_lumsegm...resets when a new value is loaded

New initial/power-up values: lum_segm_period=X"4000"=1,46s;
trigger types..all assigned with values >0

4.1.12 V0018

ROP_BCR_DELAY: New register to be added to the address table!

Trigger counters checked and revised

4.1.13 V0017

PTC_SM: 2 new state codes: 0012, 0013....to be interpreted by software !!!

bgo_ctrl: L1A not suppressed anymore around Bo=BCR, Begin_of_Gap

trigger: Random trigger is sent during valid BCs only

'Stop_Mon' pin carries 'new_lum_segment' signal to TCSM chip

PTC0_OTHER_DEADTIMEA_xx_xx registers removed

4.1.14 V0016

Text for Throttle Rules updated.

PTC register: 'rndm_prescale_fact' replaces 'rndom_start_value'

PTCx_CMD_REG(2)=EN_ERR_TRIG

PTCx_TRIG_TYPE_B(11-8)=ERROR Trigger type = 0111

4.1.15 V0015

COMMON_CMD pulse

bit 5: **RESET_ORBIT_NR** resets the common orbit counter.

ENIO_ASSIGN_LUMGT:

Bits 15: PTC0_CLEARS_ORBITNR

1= PTC0 resets the orbit counter when starting a new run.

0= only the new VME command pulse resets the orbit counter

PTC0...7_CMD pulse

Bit 0: **PANIC_BUTTON** // Sets the PTC_SM to IDLE mode

PTC_SM...additional possibilities for 'stop run'

-- PTC_SM:

-- Starts independently from input status.

-- PANIC BUTTON returns ptc_sm to IDLE status if 'done' bit is missing

-- because the BC-table does not contain the bit for this BGo command.

-- ptc_clears_orbitnr = '1' ..optional reset of orbit counter

-- IDLE:stop_run_fsm_cld <= '1' ; ' =ignore old stop request and keep run_ff=0

-- 'hard_res' is possible when in out_of_sync status

PTCx_FSM_STATES

New states PANIC1, PANIC2, PANIC3

4.1.16 V0014

→ TESTMASKS0..7: sigs2emu added

4.1.17 V000D

→ TESTMASK8: new and moved bits!!

4.1.18 V000C

→ TESTMASKS0..7: throttle_inhibit as bit 9

→ TESTMASK8: end_of_record

→ New BGO Period values: see table

PTC: with Resync, HardRes included

4.1.19 V000B

PTCx_FSM_STATES new register, PTC_SM changed, counter readout corrected

4.1.20 V000A

EN_IO & ASSIGN_LUM_GT :

- IGNORE_GTFE-, .._GT-, ..._TCS_ROP_STATUS bits moved to PTC0...7_CMD_REG to set them differently for each PTC.

- ASS_GT: New code to run Physics and Test modes

4.2 VME ADDRESSES

4.2.1 Overview

VME bus specification 1987:

Tables 2 and 4 combined:

Access	DS1*	DS0*	A01	LWORD*	D31-24	D23-16	D15-8	D7-0
BYTE0-1	low	low	0	high			Byte0	Byte1
BYTE2-3	low	low	1	high			Byte2	Byte3
BYTE0-3	low	low	0	low	Byte0	Byte1	Byte2	Byte3
BYTE0	low	high	0	high			Byte0	
BYTE1	high	low	0	high				Byte1
BYTE2	low	high	1	high			Byte2	
BYTE3	high	low	1	high				Byte3

Address bits A19-1 are connected to the TCS chip.

A31	A30	A29	A28	A27	A26	A25	A24	A23	A22	A21	A20
TCS-base-address								Chip Name			
TCS-base-address								TCS chip=0100			

		A 19-16	A 15 -12	A 11-8	A 7 -4	A 3-0
DAQ0	Registers and Monitoring-Counters	0000	0000	0000	DAQ0-registers 00 ... 5E	
DAQ0	BC-TABLE0	0000	010	4k address		
DAQ0	BC-TABLE1	0000	011	4 k address		
DAQ1	Registers and Monitoring-Counters	0001	0000	0000	DAQ0-registers 00 ... 5E	
DAQ1	BC-TABLE0	0001	010	4k address		
DAQ1	BC-TABLE1	0001	011	4 k address		
Same for DAQ2...DAQ6						
DAQ7	Registers and Monitoring-Counters	0111	0000	0000	DAQ0-registers 00 ... 5E	
DAQ7	BC-TABLE0	0111	010	4k address		
DAQ7	BC-TABLE1	0111	011	4 k address		
TCS	Common Registers	1000	0000	0000	Common registers	

4.2.2 Common addresses

Address	Name	D15 - 0			
bb48 0000	RULE4_DLY	RULE4_DLY[15:0]			
bb48 0002	RULE4_LIM	15 – 9 0000 000	8 -0 RULE4_LIMIT		
bb48 0004	RULE4_LOW-LIM	15 – 9 0000 000	8 -0 RULE4 LOW LIMIT		
bb48 0006	RULE3_DLYH	0000	dly3_(27:24)	dly3_(23:20)	dly3_(19:16)
bb48 0008	RULE3_DLYL	dly3_(15:12)	dly3_(11:8)	dly3_(7:4)	dly3_(3:0)
bb48 000A	RULE3_LIM	15 –8 “00” & RULE3_LIMIT	7 -0 “00” & RULE3 LOW LIMIT		
bb48 000C	RULE2_DLY	dly2_(15:12)	dly2_ (11:8)	dly2_ (7:4)	dly2_ (3:0)
bb48 000E	RULE1_2	15 – 12 RULE2_ LIMIT	11 – 8 RULE2_ LOW LIMIT	7 - 3 0000 0	2 – 0 RULE1_ MIN DT

bb48 0010	TIMESLOT0	00	FF	8 bits used
bb48 0012	TIMESLOT1	00	00	8 bits used
bb48 0014	TIMESLOT2	00	00	8 bits used
bb48 0016	TIMESLOT3	00	00	8 bits used

bb48 0018	TIMESLOT4	00	00	8 bits used
bb48 001A	TIMESLOT5	00	00	8 bits used
bb48 001C	TIMESLOT6	00	00	8 bits used
bb48 001E	TIMESLOT7	00	00	8 bits used

	D15 - 12		D11 - 8		D7 - 4		D3 - 0	
bb48 +xx	E N	DAQ_NR [2:0]	E N	DAQ_NR [2:0]	E N	DAQ_NR [2:0]	E N	DAQ_NR [2:0]
+20		ASSIGN_PART3		ASSIGN_PART2		ASSIGN_PART1		ASSIGN_PART0
+22		ASSIGN_PART7		ASSIGN_PART6		ASSIGN_PART5		ASSIGN_PART4
+24		ASSIGN_PART11		ASSIGN_PART10		ASSIGN_PART9		ASSIGN_PART8
+26		ASSIGN_PART15		ASSIGN_PART14		ASSIGN_PART13		ASSIGN_PART12
+28		ASSIGN_PART19		ASSIGN_PART18		ASSIGN_PART17		ASSIGN_PART16
+2A		ASSIGN_PART23		ASSIGN_PART22		ASSIGN_PART21		ASSIGN_PART20
+2C		ASSIGN_PART27		ASSIGN_PART26		ASSIGN_PART25		ASSIGN_PART24
+2E		ASSIGN_PART31		ASSIGN_PART30		ASSIGN_PART29		ASSIGN_PART28
+30	ENABLE IO				ASSIGN LUM		ASSIGN GT	

bb48 0032	ORBIT_LENGTH_1	default: 3563=0DEB	
bb48 0034	GAP_LIMITH	default: 3563=0DEB	
bb48 0036	GAP_LIMITL	default: 3444=0D74	
bb48 0038	SETTLE_TIME	15 - 8 RESYNC SETTLE TIME Default*: C (orbits)	7 - 0 HARD_RESET SETTLE TIME Default*: C(orbits)
b48 003A	RECOVER_TIME	15 - 8 RESYNC RECOVER TIME Default*: 1(orbit)	7 - 0 HARD_RESET RECOVER TIME Default*: 1(orbit)

bb48 003C	BCRES_DELAY	w/r	delay up to 1 orbit; unit =1bx
bb48 003E	GT_STATUS	-/r	0 3 x x

bb48 0040	COMMONSTATUS	-/r	EVM_FIFO	FDL_TCS	TIM_TCS[7:0]
------------------	--------------	-----	----------	---------	--------------

bb48 0042	COMMON_CMD	w/-	15:0	command pulses
------------------	------------	-----	------	-----------------------

bb48 0044	CHIP_ID_H	-/r	31:16	0001
bb48 0046	CHIP_ID_L	-/r	15:0	5131
bb48 0048	Version_Nr_H	-/r	31:16	0000
bb48 004A	Version_Nr_L	-/r	15:0	nnnn

bb48 004C	STATUS_DAQ_0300	-/r	STAT_DAQ3	STAT_DAQ2	STAT_DAQ1	STAT_DAQ0
bb48 004E	STATUS_DAQ_0704	-/r	STAT_DAQ7	STAT_DAQ6	STAT_DAQ5	STAT_DAQ4
bb48 0050	STATUS_P0300	-/r	STAT_P3	STAT_P2	STAT_P1	STAT_P0
bb48 0052	STATUS_P0704	-/r	STAT_P7	STAT_P6	STAT_P5	STAT_P4
bb48 0054	STATUS_P1108	-/r	STAT_P11	STAT_P10	STAT_P9	STAT_P8
bb48 0056	STATUS_P1512	-/r	STAT_P15	STAT_P14	STAT_P13	STAT_P12
bb48 0058	STATUS_P1916	-/r	STAT_P19	STAT_P18	STAT_P17	STAT_P16
bb48 005A	STATUS_P2320	-/r	STAT_P23	STAT_P22	STAT_P21	STAT_P20
bb48 005C	STATUS_P2724	-/r	STAT_P27	STAT_P26	STAT_P25	STAT_P24
bb48 005E	STATUS_P3128	-/r	STAT_P31	STAT_P30	STAT_P29	STAT_P28

bb48 0060	STAT to PTC1_0	-/r	bits 15:8: Status to PTC1	bits 7:0: Status to PTC0
bb48 0062	STAT to PTC3_2	-/r	bits 15:8: Status to PTC3	bits 7:0: Status to PTC2
bb48 0064	STAT to PTC5_4	-/r	bits 15:8: Status to PTC5	bits 7:0: Status to PTC4
bb48 0066	STAT to PTC7_6	-/r	bits 15:8: Status to PTC7	bits 7:0: Status to PTC6
bb48 0068	free			
bb48 006A	ORBITNRHH	-/r	bits [47:32]	Reserved for Orbit number
bb48 006C	ORBITNRH	-/r	bits [31:16]	Orbit number
bb48 006E	ORBITNRL	-/r	bits [15:0]	(32 bits → 106 hours)

bb48 0070	PART0_RUN_NR_H		31:16	r/w	inserted into EVENT_RECORD
bb48 0072	PART0_RUN_NR_L		15:0	r/w	
bb48 0074	PART1_RUN_NR_H		31:16	r/w	
bb48 0076	PART1_RUN_NR_L		15:0	r/w	
bb48 0078	PART2_RUN_NR_H		31:16	r/w	
bb48 007A	PART2_RUN_NR_L		15:0	r/w	
bb48 007C	PART3_RUN_NR_H		31:16	r/w	
bb48 007E	PART3_RUN_NR_L		15:0	r/w	
bb48 0080	PART4_RUN_NR_H		31:16	r/w	

bb48 0082	PART4_RUN_NR_L	15:0	r/w	
bb48 0084	PART5_RUN_NR_H	31:16	r/w	
bb48 0086	PART5_RUN_NR_L	15:0	r/w	
bb48 0088	PART6_RUN_NR_H	31:16	r/w	
bb48 008A	PART6_RUN_NR_L	15:0	r/w	
bb48 008C	PART7_RUN_NR_H	31:16	r/w	
bb48 008E	PART7_RUN_NR_L	15:0	r/w	
bb48 0090	LUMINOSITY_PERIOD_H	31-16	r/w	2**24 orbits →24,9 min max
bb48 0092	LUMINOSITY_PERIOD_L	15-0	r/w	2**24 orbits →24,9 min max
bb48 0094	BOARD_ID	15:0	r/w	inserted into EVENT_RECORD
bb48 0096	SIM_EMU_STATUS3128	15:0	r/w	simulated status of emulators
bb48 0098	SIM_EMU_STATUS2724	15:0	r/w	simulated status of emulators
bb48 009A	SIM_EMU_CTRL	15:0	r/w	applies either real or simulated status of emulators to PTC's
bb48 009C	SWITCH_TP2PAN	15:0	r/w	switches testpoints to front panel
bb48 009E	TESTMASK0	15:0	r/w	switches signals to testpoint 0
bb48 00A0	TESTMASK1	15:0	r/w	
bb48 00A2	TESTMASK2	15:0	r/w	
bb48 00A4	TESTMASK3	15:0	r/w	
bb48 00A6	TESTMASK4	15:0	r/w	
bb48 00A8	TESTMASK5	15:0	r/w	
bb48 00AA	TESTMASK6	15:0	r/w	
bb48 00AC	TESTMASK7	15:0	r/w	switches signals to testpoint 8
bb48 00AE	TESTMASK8	15:0	r/w	common signals

bb48 00B0	LUMINOSITY_NR_H	31:16	-/r	Reserved for Luminosity number 31:16
bb48 00B2	LUMINOSITY_NR_L	15:0	-/r	Luminosity number 15:0
bb48 00B4	BC_ERRORS	15:0	-/r	# of bc errors since last read access or power-up

bb48 00B6	ROP_BCR_DELAY	15:0	w/r	BCRES delay for ROP; unit =1bx
-----------	---------------	------	-----	--------------------------------

4.2.3 PTC0...7 addresses

PTCx controls DAQx partition.

Old register names: DAQx_... ↔ New register names: PTCx_

X = "0"..."7" hex

Addresses with "bb40" exist only in DAQ0 partition.

Addresses with "bb4x" exist in all DAQ partitions.

Address	Name	D15 - 0
bb4x 4000 until bb4x 5FFE	PTCx_ BC_TABLE0	See bit descriptions below
bb4x 6000 until bb4x 7FFE	PTCx_ BC_TABLE1	See bit descriptions below.

bb4x 0000	PTCx_CMD_REG		See description below.		
bb4x 0002	PTCx_RANDOM_FREQ		'NN'		
bb4x 0004	PTCx_RANDOM_ PRESCALE_FACTOR		not used by Poisson Generator		
bb4x 0006	PTCx_ TRIG_ TYPE_ A	15 - 12 TECHNICAL Trigger type 0100	11 - 8 RANDOM Trigger type 0011	7 - 4 CALIBRATION Trigger type 0010	3 - 0 PHYSICS Trigger type 0001
bb4x 0008	PTCx_ TRIG_ TYPE_B	15 - 12 EMULATOR Trigger type 1000	11 - 8 ERROR Trigger type 0111	7 - 4 TEST Trigger type 0110	3 - 0 TRACED physics Trigger type 0101
bb4x 000A	PTCx_ BGO_ PERIOD L	15 - 12 not used	11 - 8 PRIVATE_ BGO_CMD_ PERIOD	7 - 4 PRIVATE_ ORBIT_ PERIOD	3 - 0 PRIVATE_ GAP_ PERIOD
bb4x 000C	PTCx_ BGO_ PERIOD S	15 - 12 TEST_ TRIGGER_ PERIOD	11 - 8 CALIBR_ PERIOD	7 - 4 TRACE_ PERIOD	3 - 0 BCRES_ PERIOD
bb4x 000E	PTCx_ EMU_ DELAY	EMU_RES_DLY		EMU_BCR_DELAY	

Address	Register name			Remark
bb4x 0010	PTCx_P_STATUS	-/r	15 : 0	See description below.
bb4x 0012	PTCx_FSM_STATES	-/r	15 : 0	PTC-state machine status codes.
bb4x 0014				
bb4x 0016				

bb4x 0018				
bb4x 001A				
bb4x 001C				
bb4x 001E				

PTCx Trigger Counters: New addresses - read only registers

4x 0020	PTCx_TRIGGER_NR_31_16	bits 31 - 16	all PTC	Full after 11.9h at 100kHz
4x 0022	PTCx_TRIGGER_NR_15_0	bits 15 - 0	all PTC	
4x 0024	PTCx_EVENT_NR_31_16	bits 31 - 16	all PTC	
4x 0026	PTCx_EVENT_NR_15_0	bits 15 - 0	all PTC	
4x 0028	PTCx_FINOR_DISTRIBUTED_31_16	bits 31 - 16	all PTC	
4x 002A	PTCx_FINOR_DISTRIBUTED_15_0	bits 15 - 0	all PTC	
4x 002C	PTCx_CAL_TRIGGER_31_16	bits 31 - 16	all PTC	
4x 002E	PTCx_CAL_TRIGGER_15_0	bits 15 - 0	all PTC	
40 0030	PTCx_RANDOM_TRIGGER_31_16	bits 31 - 16	all PTC	
40 0032	PTCx_RANDOM_TRIGGER_15_0	bits 15 - 0	all PTC	
40 0034	PTC0_TEST_TRIGGER_31_16	bits 31 - 16	PTC0 only	
40 0036	PTC0_TEST_TRIGGER_15_0	bits 15 - 0	PTC0 only	
40 0038	PTC0_FINOR_GENERATED_31_16	bits 31 - 16	PTC0 only	
40 003A	PTC0_FINOR_GENERATED_15_0	bits 15 - 0	PTC0 only	
40 003C	PTC0_FINOR_IN_INVALID_BC_31_16	bits 31 - 16	PTC0 only	
40 003E	PTC0_FINOR_IN_INVALID_BC_15_0	bits 15 - 0	PTC0 only	

bb4x 0040				<i>old Version V7: trig & deadtime cntrs</i>
.....	not used			
bb4x 005E				

-- Write only Command pulse at old address.

bb4x 0060	PTCx_CMD(PULSE)	w/-	[15:0]	old address
bb4x 0062	not used			
.....	not used			
bb4x 006E	not used			

DAQ0 Deadtime Counters - read only registers

40 0070	PTCx_DEADTIME_47_32	bits 47 - 32	all PTC	48bit→81d
40 0072	PTCx_DEADTIME_31_16	bits 31 - 16	all PTC	44b →121h
40 0074	PTCx_DEADTIME_15_0	bits 15 - 0	all PTC	40bit→7.6h

40 0076	PTCx <u>LOST FINOR</u> 47 32	bits 47 - 32	all PTC	
40 0078	PTCx <u>LOST FINOR</u> 31 16	bits 31 - 16	all PTC	
40 007A	PTCx <u>LOST FINOR</u> 15 0	bits 15 - 0	all PTC	
40 007C	PTC0 <u>DEADTIMEA</u> 47 32	bits 47 - 32	PTC0 only	
40 007E	PTC0 <u>DEADTIMEA</u> 31 16	bits 31 - 16	PTC0 only	
40 0080	PTC0 <u>DEADTIMEA</u> 15 0	bits 15 - 0	PTC0 only	
40 0082	PTC0 <u>LOST FINORA</u> 47 32	bits 47 - 32	PTC0 only	
40 0084	PTC0 <u>LOST FINORA</u> 31 16	bits 31 - 16	PTC0 only	
40 0086	PTC0 <u>LOST FINORA</u> 15 0	bits 15 - 0	PTC0 only	
40 0088	PTC0 <u>PRIV DEADTIMEA</u> 47 32	bits 47 - 32	PTC0 only	
40 008A	PTC0 <u>PRIV DEADTIMEA</u> 31 16	bits 31 - 16	PTC0 only	
40 008C	PTC0 <u>PRIV DEADTIMEA</u> 15 0	bits 15 - 0	PTC0 only	
40 008E	PTC0 <u>PTCSTATUS DEADTIMEA</u> 47 32	bits 47 - 32	PTC0 only	
40 0090	PTC0 <u>PTCSTATUS DEADTIMEA</u> 31 16	bits 31 - 16	PTC0 only	
40 0092	PTC0 <u>PTCSTATUS DEADTIMEA</u> 15 0	bits 15 - 0	PTC0 only	
40 0094	PTC0 <u>THROTTLE DEADTIMEA</u> 47 32	bits 47 - 32	PTC0 only	
40 0096	PTC0 <u>THROTTLE DEADTIMEA</u> 31 16	bits 31 - 16	PTC0 only	
40 0098	PTC0 <u>THROTTLE DEADTIMEA</u> 15 0	bits 15 - 0	PTC0 only	
40 009A	PTC0 <u>CALIBRATION DEADTIMEA</u> 47 32	bits 47 - 32	PTC0 only	
40 009C	PTC0 <u>CALIBRATION DEADTIMEA</u> 31 16	bits 31 - 16	PTC0 only	
40 009E	PTC0 <u>CALIBRATION DEADTIMEA</u> 15 0	bits 15 - 0	PTC0 only	
40 00A0	PTC0 <u>TIMESLOT DEADTIMEA</u> 47 32	bits 47 - 32	PTC0 only	
40 00A2	PTC0 <u>TIMESLOT DEADTIMEA</u> 31 16	bits 31 - 16	PTC0 only	
40 00A4	PTC0 <u>TIMESLOT DEADTIMEA</u> 15 0	bits 15 - 0	PTC0 only	
40 00A6	PTC0 <u>OTHER DEADTIMEA</u> 47 32	bits 47 - 32	PTC0 only	removed
40 00A8	PTC0 <u>OTHER DEADTIMEA</u> 31 16	bits 31 - 16	PTC0 only	removed
40 00AA	PTC0 <u>OTHER DEADTIMEA</u> 15 0	bits 15 - 0	PTC0 only	removed
40 00AC	PTC0 <u>NR OF RESETS</u> 31 16	bits 31 - 16	All PTC	
40 00AE	PTC0 <u>NR OF RESETS</u> 15 0	bits 15 - 0	All PTC	
40 00B0	PTC0 <u>EMU TRIG DEADTIME</u> 47 32	bits 47 - 32	PTC0 only	
40 00B2	PTC0 <u>EMU TRIG DEADTIME</u> 31 16	bits 31 - 16	PTC0 only	
40 00B4	PTC0 <u>EMU TRIG DEADTIME</u> 15 0	bits 15 - 0	PTC0 only	

4x 00C0	PTCx <u>EMU TRIG DELAY</u>	w/r	15 : 0	Synchronizing Emulator-trigger to Finor
----------------	---------------------------------------	-----	--------	--

4.3 Power-Up values

Version V001A: 5 jun 2008

-- Table for initial values

```

-- STANDARD RULES of July 2007:
--No more than 1 Level 1 Accept per 75 nsec (minimum 2 bx between L1As),
--No more than 2 Level 1 Accepts per 625 nsec (25 bx),
--No more than 3 Level 1 Accepts per 2.5 microsec (100 bx),
--No more than 4 Level 1 Accepts per 6 microsec (240 bx).

CONSTANT init_values003e : vec16_array (0 to 31) :=
-- value---index--name
(X"00f0", --0 RULE4_DLY // X"00F0" = 240 bx
X"0004", --1 RULE4_LIM // norm=4
X"0002", --2 RULE4_LOW_LIM // low =2
X"0000", --3 RULE3_DLYH // 64(constant)+(1+0)+(1+0)+(1+0)+(1+0)
X"00dd", --4 RULE3_DLYL // +(1+0)+(1+0)+(1+d)+(1+d) = 100 bx
X"0301", --5 RULE3_LIM // norm=3, low=1
X"00ab", --6 RULE2_DLY // (1+0)+(1+0)+(1+a)+(1+b)=25bx
X"2102", --7 RULE1_2LIM // norm=2, low=1, min_dt=2
-- 10-1e
X"0004", --8 TIMESLOT0 //PTC0 only for physics run def: FF
//For simulation: X"0004"//
X"0000", --9 TIMESLOT1 // max value = FF --> = 255*10 orbits
X"0000", --10 TIMESLOT2
X"0000", --11 TIMESLOT3
X"0000", --12 TIMESLOT4
X"0000", --13 TIMESLOT5
X"0000", --14 TIMESLOT6
X"0000", --15 TIMESLOT7
-- 20-2e
X"8888", --16 ASS_PART0300 // All to PTC0 ..physics run
X"8888", --17 ASS_PART0704
X"8888", --18 ASS_PART1108
X"8888", --19 ASS_PART1512
X"8888", --20 ASS_PART1916
X"8888", --21 ASS_PART2320
X"8888", --22 ASS_PART2724 //Emulators to PTC0
X"8888", --23 ASS_PART3128 //Emulators to PTC0
-- 30-3e
X"8000", --24 ASS_LUM_GT_ENIO --//11-8:=0 all enabled
-- 15: PTC0_CLEARS_ORBITNR (V0015)
-- 14: all PTCs ignore_gt_status
-- 13: all PTCs ignore_gtfe_status
-- 12: all PTCs ignore_tcs_rop_status
-- 11: dis_bussw(TCS<->TIM), 10: dis_chlink_gtfe
-- 9: dis_chlinks23_0(detector part's(23-0) status -
>PTCi)
-- 8: dis_out_daq (PTCi status ->DAQi_aTTS)
-- 7-4: ass_lum (0= not connected to PTC0)
-- 3-0: ass_gt_crate (0=BGo from PTC0, L1A from all
PTC)

X"0DEB", --25 MAX_BCNR // 0deb = 3563
X"0000", --26 GAP_LIM_H not used **** <--
X"0000", --27 GAP_LIM_L not used **** <--
X"0C0C", --28 SETTLE_TIME //DTTF=1,052ms=11,8orbits
X"0101", --29 RECOVER_TIME // after Resync: all detectors ready
<lorbit
X"0000", --30 BCRES_DELAY // depends from synchronisation in CMS
X"0000" --31 GT_STATUS = -/r
);
--New Registers instead of old TP1+2 regs
CONSTANT init_values70ae : vec16_array (0 to 31) :=
(
-- value---index--name
X"0000", --0 PART_RUN_NR(0) H
X"0000", --1 PART_RUN_NR(1) L
X"0000", --2 PART_RUN_NR(2)
X"0000", --3 PART_RUN_NR(3)
X"0000", --4 PART_RUN_NR(4)
X"0000", --5 PART_RUN_NR(5)
X"0000", --6 PART_RUN_NR(6)
X"0000", --7 PART_RUN_NR(7)
-- 80-8e

```

```

X"0000", --8 PART_RUN_NR(8) H
X"0000", --9 PART_RUN_NR(9) L
X"0000", --10 PART_RUN_NR(10)
X"0000", --11 PART_RUN_NR(11)
X"0000", --12 PART_RUN_NR(12)
X"0000", --13 PART_RUN_NR(13)
X"0000", --14 PART_RUN_NR(14)
X"0000", --15 PART_RUN_NR(15)
-- 90-9e
X"0000", --16 LUM_SEGMENT_PERIOD_H
X"4000", --17 LUM_SEGMENT_PERIOD_L : 4000hex--> 16384 *89,lus ~1,46
s
X"CC07", --18 BOARD_ID cc=central_control, 07=vme_slot
X"8888", --19 SIM_EMU_STATUS3128
X"8888", --20 SIM_EMU_STATUS2724
X"0000", --21 SIM_EMU_CTRL // run in real mode
X"0000", --22 SWITCH_TP2PAN
X"8000", --23 TESTMASK(0) // 8000=11a2front(0)
---a0-ae
X"8000", --24 TESTMASK(1) // 8000=11a2front(1)
X"8000", --25 TESTMASK(2) // 8000=11a2front(2)
X"8000", --26 TESTMASK(3) // 8000=11a2front(3)
X"8000", --27 TESTMASK(4) // 8000=11a2front(4)
X"8000", --28 TESTMASK(5) // 8000=11a2front(5)
X"8000", --29 TESTMASK(6) // 8000=11a2front(6)
X"8000", --30 TESTMASK(7) // 8000=11a2front(7)
X"8000", --31 TESTMASK(8) // 8000=bcores
);
CONSTANT init_value_b6 : std_logic_vector (15 DOWNT0 0) :=
(X"0002"); -- ROP_BCR_DELAY
CONSTANT init_values_ptc0 : vec16_array (0 to 7) :=
(X"4200", -- CMD_REG // dis_bad_code, use_valid_bc
X"0001", -- RNDM_PERIOD // should be ~ 10 Hz //400 ~600kHz
for simulation
X"0001", -- RNDM_PRESCALE_FACT // default =1
X"4321", -- TRIG_TYPE_A // tech & rndm & cal & phys
X"8765", -- TRIG_TYPE_B // 0 & error & test & trace
X"0000", -- BGO_PERIOD_L // 0 & 0 & priv_orb & 0
X"0008", -- BGO_PERIOD_S // test & cal & trace & bcr //bit3=en,
X"0000" -- PTC_EMU_DELAY
);
CONSTANT init_values_ptc1 : vec16_array (0 to 7) :=
(X"4200", -- CMD_REG // dis_bad_code, use_valid_bc
X"0001", -- RNDM_PERIOD // should be ~ 10 Hz
X"0001", -- RNDM_PRESCALE_FACT // default =1
X"4321", -- TRIG_TYPE_A // tech & rndm & cal & phys
X"0765", -- TRIG_TYPE_B // 0 & error & test & trace
X"0000", -- BGO_PERIOD_L // 0 & 0 & priv_orb & 0
X"0008", -- BGO_PERIOD_S // test & cal & trace & bcr //bit3=en,
X"0000" -- PTC_EMU_DELAY
);
CONSTANT init_values_ptc2 : vec16_array (0 to 7) :=
(X"4200", -- CMD_REG // dis_bad_code, use_valid_bc
X"0001", -- RNDM_PERIOD // should be ~ 10 Hz
X"0001", -- RNDM_PRESCALE_FACT // default =1
X"4321", -- TRIG_TYPE_A // tech & rndm & cal & phys
X"0765", -- TRIG_TYPE_B // 0 & error & test & trace
X"0000", -- BGO_PERIOD_L // 0 & 0 & priv_orb & 0
X"0008", -- BGO_PERIOD_S // test & cal & trace & bcr //bit3=en,
X"0000" -- PTC_EMU_DELAY
);
CONSTANT init_values_ptc3 : vec16_array (0 to 7) :=
(X"4200", -- CMD_REG // dis_bad_code, use_valid_bc
X"0001", -- RNDM_PERIOD // should be ~ 10 Hz
X"0001", -- RNDM_PRESCALE_FACT // default =1
X"4321", -- TRIG_TYPE_A // tech & rndm & cal & phys
X"0765", -- TRIG_TYPE_B // 0 & error & test & trace
X"0000", -- BGO_PERIOD_L // 0 & 0 & priv_orb & 0
X"0008", -- BGO_PERIOD_S // test & cal & trace & bcr //bit3=en,
X"0000" -- PTC_EMU_DELAY
);

```

```

CONSTANT init_values_ptc4 : vec16_array (0 to 7) :=
(X"4200", -- CMD_REG // dis_bad_code, use_valid_bc
X"0001", -- RNDM_PERIOD // should be ~ 10 Hz
X"0001", -- RNDM_PRESCALE_FACT // default =1
X"4321", -- TRIG_TYPE_A // tech & rndm & cal & phys
X"0765", -- TRIG_TYPE_B // 0 & error & test & trace
X"0000", -- BGO_PERIOD_L // 0 & 0 & priv_orb & 0
X"0008", -- BGO_PERIOD_S // test & cal & trace & bcr //bit3=en,
X"0000" -- PTC_EMU_DELAY
);
CONSTANT init_values_ptc5 : vec16_array (0 to 7) :=
(X"4200", -- CMD_REG // dis_bad_code, use_valid_bc
X"0001", -- RNDM_PERIOD // should be ~ 10 Hz
X"0001", -- RNDM_PRESCALE_FACT // default =1
X"4321", -- TRIG_TYPE_A // tech & rndm & cal & phys
X"0765", -- TRIG_TYPE_B // 0 & error & test & trace
X"0000", -- BGO_PERIOD_L // 0 & 0 & priv_orb & 0
X"0008", -- BGO_PERIOD_S // test & cal & trace & bcr //bit3=en,
X"0000" -- PTC_EMU_DELAY
);
CONSTANT init_values_ptc6 : vec16_array (0 to 7) :=
(X"4200", -- CMD_REG // dis_bad_code, use_valid_bc
X"0001", -- RNDM_PERIOD // should be ~ 10 Hz
X"0001", -- RNDM_PRESCALE_FACT // default =1
X"4321", -- TRIG_TYPE_A // tech & rndm & cal & phys
X"0765", -- TRIG_TYPE_B // 0 & error & test & trace
X"0000", -- BGO_PERIOD_L // 0 & 0 & priv_orb & 0
X"0008", -- BGO_PERIOD_S // test & cal & trace & bcr //bit3=en,
X"0000" -- PTC_EMU_DELAY
);
CONSTANT init_values_ptc7 : vec16_array (0 to 7) :=
(X"4200", -- CMD_REG // dis_bad_code, use_valid_bc
X"0001", -- RNDM_PERIOD // should be ~ 10 Hz
X"0001", -- RNDM_PRESCALE_FACT // default =1
X"4321", -- TRIG_TYPE_A // tech & rndm & cal & phys
X"0765", -- TRIG_TYPE_B // 0 & error & test & trace
X"0000", -- BGO_PERIOD_L // 0 & 0 & priv_orb & 0
X"0008", -- BGO_PERIOD_S // test & cal & trace & bcr //bit3=en,
X"0000" -- PTC_EMU_DELAY
);

```

4.4 COMMON Registers

	A 31-24	A23 – 1(0)	
Common Registers	Base	+ 48 0000 until 48 003E	Write and Read-back access
	Base	+ 48 0040	Read access only
	Base	+ 48 0042	Write access only
	Base	+ 48 0044 – 5E	Read access only
Monitoring Memories <i>NOT implemented</i>	Base	+ 48 1000	Mon0 RAM: Status_PTC_7654
	Base	+ 48 1800	Mon1 RAM: Status_PTC_3210
	Base	+ 48 2000	Mon2 RAM: ORBITnr_31_16
	Base	+ 48 2800	Mon3 RAM: ORBITnr_15_0
	Base	+ 48 3000	Mon4 RAM: BCnr
	Base		

4.4.1 THROTTLE RULES

The registers define the allowed number of triggers within 4 time intervals.

DELAY = trigger accumulation time

LIMIT = number of triggers for normal rates

LOW_LIMIT = number of triggers for reduced rates

Hardware Signals: THR_RULE[127:0]

Address	Name	D15 - 0			
Base+ 48 0000	RULE4_ DLY	RULE4_DLY[15:0]			
Base+ 48 0002	RULE4_ LIM	15 - 9 0000 000	8 -0 RULE4 LIMIT		
Base+ 48 0004	RULE4_ LOW-LIM	15 - 9 0000 000	8 -0 RULE4 LOW_LIMIT		
Base+ 48 0006	RULE3_ DLYH	0000	dly3_(27:24)	dly3_(23:20)	dly3_(19:16)
Base+ 48 0008	RULE3_ DLYL	dly3_(15:12)	dly3_(11:8)	dly3_(7:4)	dly3_(3:0)
Base+ 48 000A	RULE3_ LIM	15 - 14 00	13 - 8 RULE3 LIMIT	7-6 00	5 -0 RULE3 LOW_LIMIT
Base+ 48 000C	RULE 2_ DLY	dly2_(15:12)	dly2_(11:8)	dly2_(7:4)	dly2_(3:0)
Base+ 48 000E	RULE1_2	15 - 12 RULE2_ LIMIT	11 - 8 RULE2_ LOW_LIMIT	7 - 3 0000 0	2 - 0 RULE1_ MIN_DT

The accumulation time in which a certain number of triggers are allowed, is implemented by serially connected SRL16 delay units. Each programmable SRL16 delays the signals for 1bx plus the programmed 4bit number. Therefore small delays have to be added to get the final accumulation time. See also the table with default values to see how to program the accumulation time. FF's before the delay line and the history counter add a delay of 2 ticks.

Set the LIMIT values in Rule 2, 3 and 4 one count less than the desired number of triggers because of the '>' condition in the VHDL code.

Rule1 Value

= **DEAD-TIME (unit=BC) after a L1A**

Example: value=2: L1A is i

Rule2 contains 4 programmable SRL16 delays connected serially.

Rule2 delay = 2 + 4 + dly2_(15:12) + dly2_(11:8) + dly2_(7:4) + dly2_(3:0)

Rule2 range: 6 ... 66 (max=6+4*15)

Rule3 contains 4x16bx delays + 7 programmable SRL16 delays connected serially.

Rule3 delay = 2 + 64 + 7 + dly3_(27:24) + dly3_(23:20) + dly3_(19:16) + dly3_(15:12) + dly3_(11:8) + dly3_(7:4) + dly3_(3:0)

Rule3 range: 73 ... 178 (max=73+7*15)

Rule4 is implemented with a 2k DPRAM.

The accumulation time = $2 + \text{RULE4_DLY}[15:0]$ with a maximum value of 2047 (2k-1). Bits 15-11 are ignored.

Rule4 range: 2 ... 2049

Remark:

RULE 4 is implemented using a RAMBLOCK that is not accessible by VME software.

-- STANDARD RULES of July 2007:

--No more than 1 Level 1 Accept per 75 nsec (minimum 2 bx between L1As),

--No more than 2 Level 1 Accepts per 625 nsec (25 bx),

--No more than 3 Level 1 Accepts per 2.5 microsec (100 bx),

--No more than 4 Level 1 Accepts per 6 microsec (240 bx).

X"00EE", --0 RULE4_DLY // X"00EE" = 240 -2 bx

X"0003", --1 RULE4_LIM // norm=4 -1

X"0001", --2 RULE4_LOW_LIM // low =2-1

X"0000", --3 RULE3_DLYH // 66(constant)+(1+0)+(1+0)+(1+0)+(1+0)

X"00FC", --4 RULE3_DLYL // +(1+0)+(1+0)+(1+F)+(1+C) = 100 bx

X"0200", --5 RULE3_LIM // norm=3-1, low=1-1

X"00ab", --6 RULE2_DLY // (1+0)+(1+0)+(1+a)+(1+b)=25bx

X"1002", --7 RULE1_2LIM // norm=2-1, low=1-1, min_dt=2

4.4.2 TIME_SLOTS

DAQ partitions do not run concurrently. Each DAQ-partition runs for a programmed time.

Then the Time_Slot Distributor switches to the next partition in 'round robin' mode

The registers define the active time intervals of the DAQ_PTC. If the register content =0 then the DAQ partition does not run.

PTCx_PERIOD(7:0): The time slot unit corresponds to 10 ORBITS.

Address	Name	D15 - 8	D7 - 0	Default for Physics Run
Bb48 0010	TIMESLOT0	Not used = 00 hex	PTC0_PERIOD(7:0)	FF
Bb48 0012	TIMESLOT1	Not used = 00 hex	PTC1_PERIOD(7:0)	0
Bb48 0014	TIMESLOT2	Not used = 00 hex	PTC2_PERIOD(7:0)	0
Bb48 0016	TIMESLOT3	Not used = 00 hex	PTC3_PERIOD(7:0)	0
Bb48 0018	TIMESLOT4	Not used = 00 hex	PTC4_PERIOD(7:0)	0

Bb48 001A	TIMESLOT5	Not used = 00 hex	PTC5_PERIOD(7:0)	0
Bb48 001C	TIMESLOT6	Not used = 00 hex	PTC6_PERIOD(7:0)	0
Bb48 001E	TIMESLOT7	Not used = 00 hex	PTC7_PERIOD(7:0)	0

4.4.3 ASSIGN 32 Detector Partitions to 8 DAQ partitions

The registers are used to assign 32 detector partitions and the Luminosity counters to one of the 8 DAQ-partitions. ENABLE bits are used to disconnect the partitions from any DAQ-partition.

If the S-Links for EVM and DAQ run into troubles than all DAQ partitions should decrease the L1A rate or stop the L1A signals. Therefore all PTCs have to accept a WARN and ERR status signal from the EVM interface on the GTFE board .

Address: Bb48 0000+ ADDR

ADDR	D15 - 12		D11 - 8		D7 - 4		D3 - 0	
	EN	DAQ_NR[2:0]	EN	DAQ_NR[2:0]	EN	DAQ_NR[2:0]	EN	DAQ_NR[2:0]
+20	ASSIGN_PART3		ASSIGN_PART2		ASSIGN_PART1		ASSIGN_PART0	
+22	ASSIGN_PART7		ASSIGN_PART6		ASSIGN_PART5		ASSIGN_PART4	
+24	ASSIGN_PART11		ASSIGN_PART10		ASSIGN_PART9		ASSIGN_PART8	
+26	ASSIGN_PART15		ASSIGN_PART14		ASSIGN_PART13		ASSIGN_PART12	
+28	ASSIGN_PART19		ASSIGN_PART18		ASSIGN_PART17		ASSIGN_PART16	
+2A	ASSIGN_PART23		ASSIGN_PART22		ASSIGN_PART21		ASSIGN_PART20	
+2C	ASSIGN_PART27		ASSIGN_PART26		ASSIGN_PART25		ASSIGN_PART24	
+2E	ASSIGN_PART31		ASSIGN_PART30		ASSIGN_PART29		ASSIGN_PART28	
+30	ENABLE_IO				ASSIGN_LUM		ASSIGN_GT	

Register Names:

Bb48 0000+20 ASSIGN_PART3_0
 Bb48 0000+22 ASSIGN_PART7_4
 Bb48 0000+24 ASSIGN_PART11_8
 Bb48 0000+26 ASSIGN_PART15_12
 Bb48 0000+28 ASSIGN_PART19_16
 Bb48 0000+2A ASSIGN_PART23_20
 Bb48 0000+2C ASSIGN_PART27_24
 Bb48 0000+2E ASSIGN_PART31_28

Assignment is defined by :

Bit3: EN bit 1= partition is connected =0 the partition is disconnected.

Bit2-0: PTC_NR 0...7 = number of PTC (=partition controller)

Remark to Hardware: A decoder converts the register bits to:

ASSIGN_PART[31:0]_TO_DAQ[7:0] // = decoded bits

ASSIGN_LUM_TO_DAQ[7:0] // = decoded bits

ASSIGN_GT_TO_DAQ[7:0] // = decoded bits

4.4.4 EN_IO & ASSIGN_LUM_GT

Bb48 0030 ASSIGN_LUM_GT_ENIO

Bits 15: PTC0_CLEARS_ORBITNR // V0015

1= PTC0 resets the orbit counter when starting a new run.

0= PTC0 starts run without resetting the orbit counter.

Bit 14: **IGNORE_GT_STATUS** 0= default

1= ALL PTCs ignore the status of the GT_boards.

Bit 13: **IGNORE_GTFE_STATUS** 0= default

1= ALL PTCs ignore the status of the GTFE_board (EVM and DAQ chip)

Bit 12: **IGNORE_TCS_ROP_STATUS** 0= default

1= ALL PTCs ignore the status of the readout processor (TCS_ROP) that composes the tcs_event_record.

Bit 11: **DIS_BUSSW** // Default = 0 → connects signals from/to TIM board

Bit 10: **DIS_CHLINK_GTFE** ...not used anymore

// Default = 0 → sends event records via Backplane-Channel Link to the GTFE

// ROP module switches Channel Link chip to GTFE automatically on/off depending

// if record are valid or not.

Bit 9: **DIS_CHLINKS23_0**

// Default = 0 → enables Channel Link chips to receive status bits
of Partitions 23-0

Bit 8: **DIS_OUT_DAQ**

// Default = 0 → switches on LVDS drivers to send TCS status bits
to the DAQ_aTTS system.

Bits 7-4: **ASSIGN_LUM** = as other assignment bits

// power-up value: X"8" ... connected to PTC0

// sends 'RUN' and 'deadtime' signals to 'lum????'

Bits 3-0: **ASSIGN_GT** default value = B"0000"

In 'sharing mode' (ASSIGN_GT="0000") the Global Trigger sends it's status to all PTCs and accepts L1A signals from all active PTCs, but accepts the Bgo-commands from PTC0 only. The event readout circuit is shared.

In 'test mode' the Global Trigger sends it's status to one PTC and accepts L1A and Bgo command also from this PTC. That means that other PTCs cannot send event records.

"0000" Physics mode with multiple DAQ_partitions: (power up value)

GT sends status to all PTCs.

GT receives **Bgo commands from PTC0.**

GT accepts **L1A trigger signals from all PTC (0...7).**

“100i” **Test mode** with single PTC(i). i= 0...7 :

GT sends status to a single PTC(i).

GT receives Bgo commands and L1A signals from single PTC(i).

GT disregards L1A trigger signals from PTC(j ≠ i).

“0001”...”0111” ... GT is disconnected from TTC system.

GT disregards Bgo commands as well as L1A trigger signals from all PTC.

4.4.5 ORBIT_LENGTH-1

The register defines the length of the LHC orbit. The default value is **3564-1=3563** but can be changed for test to any length >8.

If the internal BCRES signal disagrees with the BCRES from the TIM board then the error flag BAD_BCNR will be set and the BC_ERRORS counter increased.

- *Always send a RESET_BAD_BCNR command when a (new) orbit length has been loaded. It clears the BAD_BCNR error flag from the first BCRES.*

Address	Name	D15 - 0	
Bb48 0032	ORBIT_LENGTH-1	D15-12 not used =0000	D11 - 0 ORBIT LENGTH - 1
		Default hex-value: 0DEB	

Hardware Signals: ORBIT_LENGTH[15:0]

Remark to LHC bunch structure:

Timing Distribution at the LHC, B.G. Taylor, 8th Workshop on Electronics for LHC Experiments, Colmar, 9 - 13 September 2002

$$1: (3 \times 72 + 2 \times 8 + 38) + 232 + (4 \times 72 + 3 \times 8 + 39) = 270 + 270 + 351 = 891$$

$$4 \times 891 = \mathbf{3564 \dots 1 \text{ LHC orbit}}$$

Small gap values: 8, 38, 39,

Long Gap: **119** = 39 + 72 + 8 (=one PS train missing)

$$\text{Valid BC} = 72 \times [(3+3+4) \times 4 - 1] = 72 \times 39 = 2808$$

PS train = 72 BC will be injected into SPS

SPS batch = 3 + 3 + 4 PS trains

LHC orbit = 4 SPS batches

4.4.6 GAP_LIMITH and GAP_LIMITL

V0018: NOT USED, MAYBE LATER !!!!

The gap limits define the position and length of long gap at the end of the LHC orbit. The default values can be changed for tests. All 8 DAQ_PTCs use the same limits.

Default values: GAP_LIMITH = 3563, GAP_LIMITL = 3444 (=3563-119)

Address	Name	D15 - 0	
Bb48 0034	GAP_LIMITH	15 - 12 0000	D11 - 0 GAP_LIMITH (default: 3563=0DEB)
Bb48 0036	GAP_LIMITL	15 - 12 0000	D11 - 0 GAP_LIMITL (default: 3444=0D74)

Hardware Signals: BC_THR_H[11:0], BC_THR_L[11:0]

4.4.7 SETTLE_TIME and RECOVER_TIME

The SETTLE TIME and RECOVER TIME are used by the BGO-control logic to send BGO commands to the detector partitions. The SETTLE TIME allows the readout electronics to move pending events into the Readout units before TCS sends a RESYNC or HARD RESET command. After a RESYNC or HARD RESET command the readout electronics needs some time to get ready. The RECOVER TIME retards the DAQ_PTC (partition controller) to move into the next state.

The unit = 1 LHC ORBIT ~89.1 us.

HARD_RESET RECOVER TIME: *Unit = 4 orbits to extend the maximum time to 1024 orbits ~91 msec*

Address	Name	D15 - 0	
Bb48 0038	SETTLE_ TIME	15 - 8 RESYNC SETTLE TIME Default*:C (orbits)	7 - 0 HARD_RESET SETTLE TIME Default*: C(orbits)
Bb48 003A	RECOVER_ TIME	15 - 8 RESYNC RECOVER TIME Default*: 1(orbit)	7 - 0 HARD_RESET RECOVER TIME Default*: n(*8orbits)

*) The default values (X”C”=12) will be changed at start-up of CMS in respect to the slowest detector partition. Max. value =255x89 us=22.7 msec.

Max Recover time for HardRes: 255*8*89,1 us = 181 ms ...because CSC needs about 90 ms

Version V101F:

-- STATE MACHINE for >>> Hardres, L1Reset <<<

-- RESYNC orbits: = (settling_time+1) + 1(active_orbit) + (recover_time +2)

-- HARDRES orbits: = (settling_time+1) + 1(active_orbit) + (8*recover_time +2)

4.4.8 BCRES_DELAY

bb48 003C	BCRES_DELAY	w/r	15:0	
------------------	--------------------	-----	------	--

Delay length up to 1 orbit-1= 3563 dec.

Unit = 25 ns=1 bunch crossing

The TIM board sends via the backplane BCRES signals to all boards in the GT crate and also to the TCS board. In the TCS chip the BCRES signal is delayed and drives the address counter of the BC-table, so that the Bgo-commands and calibration and test triggers are sent at the correct time to the detector partitions. The adjustable range of one LHC orbit allows sending the signals well in advance.

An additional small 'ROP_BCR_DELAY' is applied after the 'BCRES_DELAY' to put the correct bc-number into the event record.

4.4.9 GT-STATUS

Status signals as received from the TCSM chip.

Address	Name	15 - 0
Bb48 003E	GT_STATUS	x 3 x x

Hardware Signals: *status2vme(10) :=*

TCS_ROP_STATUS(3:0), FAST8180_[3:0], EVM_SLINK_STAT_[3:0], FAST_GT_[3:0]

Bit 15-12: TCS_ROP_STATUS as 4 bit 'fast_signals':

0000=0 : ROP_DISCONNECTED

1000=8 : ROP_READY

0010=2 : ROP_OUT_OF_SYNC

Bit 11-8: FAST8180 = **0011** hardwired check bits;

If the value is not =0011 then the status signals from the TCSM chip are transferred incorrectly.

The 40→80→40 Mhz Mux sends bits 9,8 first, then bits 11,10

Bit 7-4: EVM_STATUS_[3:0] status of EVM chip on GTFE board

or simulated in TCSM chip

Bit 3-0: GT_STATUS [3:0] ...from FDL combined status of GT boards or

simulated in TCSM chip

4.4.10 COMMON_STATUS

The 'read-only' register reflects the status of signals from the FDL and TIM board to the TCS board. The signals are not used in the TCS chip and are free for future requirements.

Address	Name	15 - 12	11 - 8	7 - 0
Bb48 0040	COMMON STATUS	EVM_FIFO	FDL_TCS	TIM_TCS[7:0]

Hardware Signals: TIM_FDL_STATUS[11:0], EVM_STAT_[2:0]

Bit 15: '0' before: *FREE_FASTIN*

// One of the undefined FASTIN connectors on the Conversion boards has been connected. They should not be used for the current design.

Bit14: ROP_FIFO_FULL Event FIFOs are full
 Bit13: ROP_FIFO_WARN75 Event FIFOs are 75% full
 Bit12: ROP_FIFO_WARN50 Event FIFOs are 50% full

An EVM-record consists of different 32 bit words. Each word is stored in a dedicated 512x32 bit FIFO. All FIFOs are filled and emptied concurrently by common read/write signals.

Bits 11: BAD_BCNR == BC_ERROR

If the BCRES signal from the backplane does not arrive with the internal BCRES signal then the BAD_BCNR error flag will be set and the BC_ERRORS counters increased. *At software initialisation time the first BCRES from the backplane will always set the error flag but it will be cleared either by the RESET_ALL or the RESET_BAD_BCNR command signals.*

Bit10: ROP_FIFO_EMPTY Event FIFOs are empty. All event have been sent to the GTFE.

Bit 9: FDL_TCS1 // wired signal not used

Bit 8: FDL_TCS0 // wired signal not used

Bit 7 – 0: TIM_TCS[7:0] // wired signals not used

BAD BC Number

The Bunch Crossing Counter is reset only either by the extern BCRES signal or automatically but never by software commands like the RESET_ALL command. If the external BCRES signal does not arrive every orbit, then the internal as well as the external BCRES signals are used. The ORBIT_LENGTH-1 register defines the internal BCRES signal. If the internal BCRES signal does not appear at the same time as the external BC-reset signal then the status flag BAD_BCNR becomes =1. Such an error might occur if the external BCRES signal is disturbed. The error becomes also permanent if the content of the ORBIT_LENGTH-1 register does not agree with the external BCRES signal.

Remark: The first external BCRES will set BAD_BCNR =1, but the flag is removed when the PTC clears the ORBIT counter after the START_RUN command. Also RESET_PTC and TCS_HARD RESET clear the BAD_BCNR flag.

BAD_BCNR =1 does not force the PTC into an Error state.

EVM error and warning flags

The error and warning flags of the circuit that builds the records for the Event Manager are included into all DAQ-partitions like the status bits of connected detector partitions.

4.4.11 COMMON_CMD pulses

Write access only. The write instruction generates a spurious signal to start or stop a procedure. If the data bit =0 nothing happens.

Address	Name	D15 - 4	D7 - 0
Bb48 0042	COMMON_CMD	See below	

Bit 15- 6: unused

~~5: RESET_ORBIT_NR~~ ... since V1022 removed

~~4= resets the common orbit counter~~

4: RESET_BAD_BCNR

Clears the BAD_BCNR flag that is set by the 1st BCRES from the back-plane.

Send RESET_BAD_BCNR after having loaded the ORBIT_LENGTH-1 register.

3: RESET_ALL

- In the Throttle Unit RESET_ALL reloads the RULE4 delay-address counters.
- Resets the TIME SLOT GENERATOR that just changes to next PTC.
- Clears the BAD_BCNR flag that is set by the 1st BCRES from the backplane

2: RESET_THROTTLE

The command pulse clears and restarts the throttle logic that cuts peaks of the trigger rate.

1: RESET_TRIGNR...not used...

Reason: TRIG_NR of a partition is cleared automatically when starting a new run.

0: PANIC_BUTTONswitches the PTC_SM immediately into the IDLE status regardless of the actual ptc_status. This 'sEC0et' command pulse must not be used during data taking because the connected subdetectors will become unsynchronized.

4.4.12 IDENTIFIER registers

The 'read-only' registers contain the hardwired identifiers. Except the board number all values are defined by firmware.

The numbers are defined in the EXCEL file in /GlobalTrigger/Documentation/ GT_chip_id.xls.

Address	Name	15 - 12	11 - 8	7 - 4	3 - 0
Bb48 0044	CHIP_ID_H	0	0	0	1
Bb48 0046	CHIP_ID_L	5	1*	3	1
Bb48 0048	Version Nr H	0	0	0	0/1
Bb48 004A	Version Nr L	n	n	n	n

CHIP Identifier_H = **0001** ← identifier of the GT crate.

CHIP Identifier_L = **5131** ..for board TCS9U#1

Bits 15:12 CARDNAME = 5 ← TCS9U

*) Bits 11:8 CARDNR = 1 ← ID2, 1, 0 JUMPERS are set 'LLH'

Bits 7:4 CHIPNAME = 3 ← TCS_chip **
 Bits 3:0 CHIP-NR =1 ← There is only one TCS chip on the board.

**) 1= VME64x chip, 2= VME_TCS chip, 3= TCS chip, 4= TCSM chip

Version Numbers:

TEST Designs → Vers. Nr <1 0000 hex

Version Number_H: = 0000 // Version Number_H: =0...FFFF

Final Designs → Vers. Nr ≥1 0000 hex

Version Number_H: = 0001 // Version Number_H: =0...FFFF

4.4.13 STATUS of DAQ- & Subdetector-Partitions

Actual status of DAQ (aTTs) and detector partitions as received by the TCSM chip(part23...0) and by the TCS chip (part31...24).

Address	Name	15 - 12	11 - 8	7 - 4	3 - 0
Bb48 004C	STATUS_DAQ_0300	STAT_ DAQ3	STAT_ DAQ2	STAT_ DAQ1	STAT_ DAQ0
Bb48 004E	STATUS_DAQ_0704	STAT_ DAQ7	STAT_ DAQ6	STAT_ DAQ5	STAT_ DAQ4
Bb48 0050	STATUS_P0300	STAT_P3	STAT_P2	STAT_P1	STAT_P0
Bb48 0052	STATUS_P0704	STAT_P7	STAT_P6	STAT_P5	STAT_P4
Bb48 0054	STATUS_P1108	STAT_P11	STAT_P10	STAT_P9	STAT_P8
Bb48 0056	STATUS_P1512	STAT_P15	STAT_P14	STAT_P13	STAT_P12
Bb48 0058	STATUS_P1916	STAT_P19	STAT_P18	STAT_P17	STAT_P16
Bb48 005A	STATUS_P2320	STAT_P23	STAT_P22	STAT_P21	STAT_P20
Bb48 005C	STATUS_P2724	STAT_P27	STAT_P26	STAT_P25	STAT_P24
Bb48 005E	STATUS_P3128	STAT_P31	STAT_P30	STAT_P29	STAT_P28

4.4.14 Combined STATUS for PTC

See also DAQ0 P_STATUS register for these values.

Actual combined status going to the DAQ partition controllers (PTC). The values are decoded.

Address	Name	15 - 8	7 - 0
Bb48 0060	STAT_to_PTC1_0	Status to PTC1	Status to PTC0
Bb48 0062	STAT_to_PTC3_2	Status to PTC3	Status to PTC2
Bb48 0064	STAT_to_PTC5_4	Status to PTC5	Status to PTC4

Bb48 0066	STAT to PTC7_6	Status to PTC7	Status to PTC6
------------------	----------------	----------------	----------------

Bit 15/7: Bad Code = bad interconnection or bad electronics (is ignored)

Bit 14/6: Partition Trigger = not used

Bit 13/5: ERROR = hardware error

Bit 12/4: READY = the partition is ready to receive L1A

Bit 11/3: BUSY = the partition is not ready for a certain time

Bit 10/2: OUT_OF_SYNC = synchronization error

Bit 9/1: WARNING = Buffer overflow warning

Bit 8/0: DISCONNECTED = partition electronics is switched off or disconnected

4.4.15 ORBIT NUMBER

There is only one common orbit counter. PTC0 command pulse(2) = 'start_run' also clears the orbit counter. 32 bits → 106 h = 4.4 d

bb48 006A	ORBITNRHH	-/r	bits [47:32]	Reserved for Orbit number
bb48 006C	ORBITNRH	-/r	bits [31:16]	Orbit number
bb48 006E	ORBITNRL	-/r	bits [15:0]	(32 bits → 106 hours)

4.4.16 PARTITION RUN NUMBERS

bb48 0070	PART0 RUN NR H	31:16	r/w	inserted into EVENT RECORD
bb48 0072	PART0 RUN NR L	15:0	r/w	
bb48 0074	PART1 RUN NR H	31:16	r/w	
bb48 0076	PART1 RUN NR L	15:0	r/w	
bb48 0078	PART2 RUN NR H	31:16	r/w	
bb48 007A	PART2 RUN NR L	15:0	r/w	
bb48 007C	PART3 RUN NR H	31:16	r/w	
bb48 007E	PART3 RUN NR L	15:0	r/w	
bb48 0080	PART4 RUN NR H	31:16	r/w	
bb48 0082	PART4 RUN NR L	15:0	r/w	
bb48 0084	PART5 RUN NR H	31:16	r/w	
bb48 0086	PART5 RUN NR L	15:0	r/w	
bb48 0088	PART6 RUN NR H	31:16	r/w	
bb48 008A	PART6 RUN NR L	15:0	r/w	
bb48 008C	PART7 RUN NR H	31:16	r/w	
bb48 008E	PART7 RUN NR L	15:0	r/w	

4.4.17 LUMINOSITY_PERIOD

bb48 0090	LUMINOSITY_PERIOD_H	31-16	r/w	2**24 orbits → 24,9 min max
bb48 0092	LUMINOSITY_PERIOD_L	15-0	r/w	2**24 orbits → 24,9 min max

Unit = 1 orbit

4.4.18 BOARD_ID

Default value for TCS board: X"CC07" -- CC=control 07=VME-slot nr.

bb48 0094	BOARD_ID	15:0	r/w	inserted into EVENT_RECORD
------------------	----------	------	-----	----------------------------

4.4.19 SIM_EMU_STATUS3128, 2724

Simulate status bits of Emulator partitions 31 ...24

bb48 0096	SIM_EMU_STATUS3128	15:0	r/w	simulated status of emulators
bb48 0098	SIM_EMU_STATUS2724	15:0	r/w	simulated status of emulators

	15-12	11-8	7-4	3-0
SIM_EMU_STATUS3128	PART31	PART30	PART29	PART28
SIM_EMU_STATUS2724	PART27	PART26	PART25	PART24

The 4 bit values are used to simulate the status of partitions 31 ...24.

Default values = 8888 hex ... = 'READY' to allow triggers.

0000 = DISCONNECTED

0001 = WARNING

0010 = OUT_OF_SYNC

0100 = BUSY

1000 = READY

1100 = ERROR

1111 = DISCONNECTED

See also Table 1 Input signals to central TCS to simulate other states.

The emulators send their status bits directly to the TCS chip to minimize the number of triggers after a warning. Therefore the simulation registers are placed also in the TCS chip.

The emulator states are then forwarded to the TCSM chip for monitoring.

4.4.20 SIM_EMU_CTRL

bb48 009A	SIM_EMU_CTRL	15:0	r/w	switches either real or simulated EMU-status to PTC
------------------	--------------	------	-----	--

Bit 15-8: not used

Bit 7: SIM_PART31
 Bit 6: SIM_PART30
 Bit 5: SIM_PART29
 Bit 4: SIM_PART28
 Bit 3: SIM_PART27
 Bit 2: SIM_PART26
 Bit 1: SIM_PART25
 Bit 0: SIM_PART24

/// '0' applies real state of emulator partition to PTC

/// '1' applies simulated state of emulator partition to PTC. Value is taken from registers SIM_EMU_STATUS3128 and SIM_EMU_STATUS2724.

4.4.21 SWITCH_TP2PAN

Switches one or more virtual testpoints to the frontpanel.

bb48 009C	SWITCH_TP2PAN	15:0	r/w	switches testpoints to front panel
------------------	---------------	------	-----	------------------------------------

bits	Testpoint to front panel
15..9	<i>not used</i>
8	<i>enable Testmask8</i>
7	<i>enable Testmask7</i>
6	<i>enable Testmask6</i>
5	<i>enable Testmask5</i>
4	<i>enable Testmask4</i>
3	<i>enable Testmask3</i>
2	<i>enable Testmask2</i>
1	<i>enable Testmask1</i>
0	<i>enable Testmask0</i>

4.4.22 TESTMASK0 ... 8

The selection of signals works in 2 stages. First connect one or more signals by testmask bits =1 to the corresponding virtual testpoint as defined in the tables below. Then connect the virtual testpoint (=result of the testmask) to the frontpanel.

bb48 009E	TESTMASK0	15:0	r/w	PTC0 signals
bb48 00A0	TESTMASK1	15:0	r/w	PTC1 signals
bb48 00A2	TESTMASK2	15:0	r/w	PTC2 signals
bb48 00A4	TESTMASK3	15:0	r/w	PTC3 signals
bb48 00A6	TESTMASK4	15:0	r/w	PTC4 signals
bb48 00A8	TESTMASK5	15:0	r/w	PTC5 signals

bb48 00AA	TESTMASK6	15:0	r/w	PTC6 signals
bb48 00AC	TESTMASK7	15:0	r/w	PTC7 signals
bb48 00AE	TESTMASK8	15:0	r/w	Common signals

bits	TESTMASK0	TESTMASK1	TESTMASK2	TESTMASK3	TESTMASK4
	<i>PTC0 signals</i>	<i>PTC1 signals</i>	<i>PTC2 signals</i>	<i>PTC3 signals</i>	<i>PTC4 signals</i>
15	l1a2front(0)	l1a2front(1)	l1a2front(2)	l1a2front(3)	l1a2front(4)
14	en time slot(0)	en time slot(1)	en time slot(2)	en time slot(3)	en time slot(4)
13	BGo_strobe(0)	BGo_strobe(1)	BGo_strobe(2)	BGo_strobe(3)	BGo_strobe(4)
12	sig2emu(2)=bcre	sig2emu(2)=bcre	sig2emu(2)=bcre	sig2emu(2)=bcre	sig2emu(2)=bcre
11	sig2emu(1)=resyn	sig2emu(1)=resyn	sig2emu(1)=resyn	sig2emu(1)=resyn	sig2emu(1)=resyn
10	sig2emu(0)=l1a	sig2emu(0)=l1a	sig2emu(0)=l1a	sig2emu(0)=l1a	sig2emu(0)=l1a
9	throttle_inhibit	throttle_inhibit	throttle_inhibit	throttle_inhibit	throttle_inhibit
8	finor (0)	finor (1)	finor (2)	finor (3)	finor (4)
7	low rate	low rate	low rate	low rate	low rate
6	norm rate	norm rate	norm rate	norm rate	norm rate
5	random trig	random trig	random trig	random trig	random trig
4	test trig	test trig	test trig	test trig	test trig
3	cal trig	cal trig	cal trig	cal trig	cal trig
2	inh l1a test	inh l1a test	inh l1a test	inh l1a test	inh l1a test
1	inh l1a cal	inh l1a cal	inh l1a cal	inh l1a cal	inh l1a cal
0	inh l1a priv	inh l1a priv	inh l1a priv	inh l1a priv	inh l1a priv

bits	TESTMASK5	TESTMASK6	TESTMASK7	TESTMASK8
	<i>PTC5 signals</i>	<i>PTC6 signals</i>	<i>PTC7 signals</i>	<i>Common signals</i>
15	l1a2front(5)	l1a2front(6)	l1a2front(7)	bcre
14	en time slot(5)	en time slot(6)	en time slot(7)	tcs_rop_full
13	BGo_strobe(5)	BGo_strobe(6)	BGo_strobe(7)	tcs_rop_warn
12	sig2emu(2)=bcre	sig2emu(2)=bcre	sig2emu(2)=bcre	tcs_rop_empty
11	sig2emu(1)=resyn	sig2emu(1)=resyn	sig2emu(1)=resyn	new_lum_seg*
10	sig2emu(0)=l1a	sig2emu(0)=l1a	sig2emu(0)=l1a	inh_norm_rate
9	throttle_inhibit	throttle_inhibit	throttle_inhibit	inh_low_rate
8	finor (5)	finor (6)	finor (7)	mask8_8:=header(3)
7	low rate	low rate	low rate	mask8_7:=header(2)
6	norm rate	norm rate	norm rate	mask8_6:=header(1)
5	random trig	random trig	random trig	mask8_5:=header(0)
4	test trig	test trig	test trig	vme_en
3	cal trig	cal trig	cal trig	dtack

2	inh_11a_test	inh_11a_test	inh_11a_test	active_beam
1	inh_11a_cal	inh_11a_cal	inh_11a_cal	rop_sel_word0
0	inh_11a_priv	inh_11a_priv	inh_11a_priv	end_of_record

*) 'new_lum_seg' is extended to a length of 1 orbit(89.1 us) to see it on the oscilloscope. It starts with the original 25 ns signal that is used in the PTC logic.

4.4.23 LUMINOSITY NUMBER

Reset when PTC0 sends 'start of run'.

Incremented every nn-th orbit as defined in the LUMINOSITY_PERIOD register.

Every new segment the content of the trigger counters are saved and the counters cleared.

bb48 00B0	LUMINOSITY_NR_H	31:16	-/r	Reserved for Luminosity number 31:16
bb48 00B2	LUMINOSITY_NR_L	15:0	-/r	Luminosity number 15:0

4.4.24 BC_ERRORS

Number of bc errors since last read access or power-up

bb48 00B4	BC_ERRORS	-/r	15:0	# of bc errors since last read access or power-up
------------------	------------------	-----	------	---

4.4.25 ROP_BCR_DELAY

bb48 00B6	ROP_BCR_DELAY	15:0	w/r	BCRES delay for ROP
------------------	----------------------	------	-----	---------------------

--UNIT= 1 BC (25 ns)

BCRES delay for the ROP to insert the correct BC number into the TCS event record.

This delay is applied after the main BCRES_DELAY.

Warning: Inhibit L1A signal while changing the delay values!!

Delay calculation:

15 - 12	11 - 8	7 - 4	3 - 0
Delay C	Delay B	Delay A	Delay= 0...3

Total Delay = Delay C + Delay B + Delay A + (0...3)

- DELAY =0 ➔ 0000 0000 0000 0000
- DELAY =1 ➔ 0000 0000 0000 0001
- DELAY =2 ➔ 0000 0000 0000 0010
- DELAY =3 ➔ 0000 0000 0000 0011
- DELAY =C+B+A+3 ➔ CCCC BBBB AAAA 0011
- For DELAY <4 the bits 15-4 have to be =0 !!
- Bits 3,2 are always =0; are not decoded

4.5 PTC0... PTC7 addresses

		A 19-16	A 15 -12	A 11-8	A 7 -4	A 3-0
PTC0 ... 7	Registers and Monitoring-Counters	x =0 - 7	0000	0000	PTC0-registers 00 ... 5E	
PTC0 ... 7	BC-TABLE0	x = 0 - 7	010	4k address		
PTC0 ... 7	BC-TABLE1	x = 0 - 7	011	4 k address		

Base =base address of TCS board

4 = TCS chip number

x =PTC_NR (=0...7)

4.5.1 PTC0...7_BC_TABLE0 ...4kx16 DPRAM

The BC-tables are used to define the time within an LHC orbit at which BGO commands and calibration and test triggers are sent to the TTC system.

The BC-table address corresponds to the BC-number at which the Bgo commands and the Calibration and Test trigger signals are sent via the L1AOUT board and the TTC system to the subdetectors and to DAQ_aTTS.

Example: If BC0 should be sent at BCnr=3540 then set bit 1 in BCTABLE0 to =1 at address 3540.

Remark: The read address of the BC-tables is provided by an address counter which is reset by a delayed BCRES signal from the TIM board. The delay is defined by the common BCRES_DELAY register and allows to adjust the internal BCRES signal in the TCS chip over a full orbit, so that even a 'negative delay' can be defined.

Address	Name	D15 - 0
Base+ 4x 4000 until 4x 5FFE	PTC0_BC_TABLE0	See bit descriptions below

```

BC_TABLE0_[15:0]           //read/write access
    15...12: PRIV_BGO_P(3:0) // bits for private BGO code
    11: PRIV_BGO(4)         //strobe bit
    10: STOP                 // 1=sends BGO code to partitions
    9:  START                // .....”.....
    8:  OC0 (RESET_ORBIT)    // .....”.....

```

- 7: EC0 (RESET_EVNR) //”.....
- 6: HARD_RESET //”.....
- 5: RESYNC //”.....
- 4: PRIV_ORBIT //”.....
- 3: PRIV_GAP // **not implemented**
- 2: TEST_TRIG // send a test trigger signal;
- 1: BCRES // 1=sends BGO=0001 code to partitions
- 0: VALID_BC // defines all small and long gaps in the LHC orbit
// 1= allows L1A in this BC, when enabled.
// 0= inhibits L1A in this BC, when enabled.
// Enable bit : See PTC_CMD_REG(9).
// VALID_BC defines the ‘ideal’ LHC orbit structure.

4.5.2 PTC0...7_BC_TABLE1 ...4kx4 DPRAM

PTCx = PTC0...PTC7

Address	Name	D15 - 0
Base+ 4x 6000 until 4x 7FFE	PTCx_ BC_TABLE1	See bit descriptions below.

BC_TABLE1_[3:0] //read/write access

15....7: 0000_0000_00 // not used

6: END_OF_CALCYC // 1 ...removes suppression of FinOR trigger signals after calibration cycle

5: START_OF_GAP // 1=sends BGO code=1011=B to partitions

4: WTE (warn_test_en) // 1=sends BGO code=1101=D to partitions

3: TRACE_EN * // 1=sends BGO code to partitions

2: TRACE_ACTIVE * // as long as =1 the first FINOR will be traced

1: TE (test_en) // 1=sends BGO code=xxxx=X to partitions
starts calibration cycle

0: CALTRIG // sends a CALIBRATION trigger pulse

4.5.3 BGO default values

Bgo command	Code	Code	#BC+	comment
BC0	0001	1	3540	sent every orbit *
Start_of_Gap	1011	11	3446	sent every orbit *
OC0=Orbit counter reset	1000	8	2000	Start of Run
EC0=Event counter reset	0111	7	3450	Start of Run + after resync
Resync/L1Reset	0101	5	2000	when required (start, out_of_sync, ...)
Hard Reset	0110	6	2000	when required (error, ...)
Start	1001	9	2000	Start of Run

Stop	1010	10	2000	End of Run
WTE (warning test enable)	1101	13	2800	prescaled calibration cycle (periodic, vme)
TE (test enable)	0010	2	3320	prescaled calibration cycle (periodic, vme)
L1A_cal	---	---	3470	prescaled calibration cycle (periodic, vme)
Private Orbit	0100	4	2400?	prescaled (periodic, vme)
Private Gap	0011	3	---	not implemented

*) BC0 and Start_of_Gap will be sent as soon as the corresponding bits have been loaded into the BC-table address.

4.6 PTC0... PTC7 REGISTER

PTC1 to PTC7 addresses are a sub-sample of the PTC0 addresses.

All names containing 'PTCx' have to be replaced by 'PTC0'... 'PTC7'.

The address bits A19-A16: PTC number.

Monitoring counters with addresses ...00xxto ...00xx exist only for PTC0.

$$PTC_x = PTC0...PTC7; \quad x = 0...7$$

	A 31-24	A23 – 1(0)	
PTCi- Registers	Base	+ 4x 0000 until 4x 000E	Write and Read-back access
	Base	+ 4x 0010 until 4x 005E	Read access only
	Base	+ 4x 0060	Write access only

4.6.1 PTC0...7_CMD_REG register

Write and read access.

Address	Name	D15 - 0
4x 0000	PTCx_CMD_REG	See description below.

Default for Test= 0201..random trigger, 0200

3/2009: TS-Database: X'1600' → same value set in V101F

Hardware Signals: PTC[7:0]_CMD_REG[15:0]

15: DIS_DISCONNECTED_CODE

// Default = 0;

// 1= disables 'DISCONNECTED' code on the status signals of connected partitions.

14: DIS_BAD_CODE

// Default = 0;

// 1= disables bad codes on the status signals of connected partitions.

Bit 13: **IGNORE_GTFE_STATUS** // Default = 0;

1= This PTC disregards the status of the GTFE board (S_Links to Event-Manager and to DAQ)

12: DIS_FINOR

// Default = 0; 1= disables the FINOR signal from the FDL board during tests.

Bit 11: **IGNORE_GT_STATUS** // Default = 0;

1= This PTC disregards the status of the GT boards

10: **ALLOW_TEST_TRIGGER** (old sense: EN_PRIVTRIG)

// Default = 0;

// 1= allows to run with test triggers generated either by BC-Table or by VME

9: **USE_VALID_BC_OF_BCTABLE**

// Default = 0;

=0 uses the Beam-Pickup-Counter (BPTX) signals to define active bunch crossings.

=1 uses the pre-programmed bits of the BC-Table to define active bunch crossings.

V100B: This option is used to control the dead-time counters. The FinOR and Random Triggers are allowed as defined by the BC-table to trigger also on long lived particles.

8: **DO NOT SEND EVENT RECORDS** (old name: RUN_WITHOUT_DAO)

// Default = 0;

// =1 inhibits the L1A signal going to the Readout Processor('EVM_OUT').

The PTC runs normally, the trigger and dead-time counters are incremented and L1A are distributed but no event records are sent (via Channel Links) to the EVM chip on the GTFE board.

Bit 7: 0 not used

Bit 6: 0 not used

Bit 5 & 4:

= 00 ...Emulator Trigger Signal is ignored

= 01 ...Emulator Trigger Signal used as TRIGGER

= 10 ...Emulator Trigger Signal inhibits FINOR signal

= 11 ...Emulator Trigger Signal inhibits FINOR signal

Bit 3: 0 not used

Bit 2: EN_ERR_TRIG V0016

In case of an error or out_of_sync status the PTC sends an "error_trigger" and increments the 'Test_Trigger' counter.

Bit 1: **IGNORE_TCS_ROP_STATUS** // Default = 0;

1= This PTC disregards the status of the readout processor (TCS_ROP) that composes the tcs_event_record.

0: EN_RNDM_TRIG

```
// =1 allows random triggers
// First set ENRNDM_TRIG =1 and then with the next VME instruction
// send START_RNDM_TRIG=1.
```

4.6.2 PTC0...7_RANDOM_FREQ

The register RANDOM_FREQ defines the frequency for the Poisson random trigger generator to send triggers with event type 'RANDOM'.

Address	Name		D15 - 0
4x 0002	PTCx_RANDOM_FREQ	w/r	'NN'

UNIT ~ 300 Hz

4.6.3 PTC0...7_RANDOM_PRESCALE_FACTOR

New in V0016

Address	Name		D15 - 0
4x 0004	PTCx_RANDOM_PRESCALE_FACTOR	w/r	

value = number of suppressed random triggers after a valid random trigger

0 → no prescaling

1 → prescale by 2

n → prescale by n+1

Signal name in firmware: rndm_prescale_fact

Maximum value = $2^{16} - 2$

4.6.4 PTC0...7_TRIG_TYPE_A and - TRIG_TYPE_B

Write and read access.

The register contains the trigger type identifiers for different kinds of triggers.

Physics trigger type: generated by Algorithms or Technical triggers.

The Technical trigger type is not assigned because technical trigger bits and algorithm bits are merged into a common Final_OR signal on the FDL board.

Address	Name	D15 - 0			
4x 0006	PTCx_TRIG_ TYPE_A	15 - 12	11 - 8	7 - 4	3 - 0
		TECHNICAL Trigger type 0100	RANDOM Trigger type 0011	CALIBRATION Trigger type 0010	PHYSICS Trigger type 0001
4x 0008	PTCx_TRIG_ TYPE_B	15 - 12	11 - 8	7 - 4	3 - 0
		EMULATOR Trigger type 1000	ERROR Trigger type 0111	TEST Trigger type 0110	TRACED physics Trigger type 0101

If tracing is active then all events are marked as traced events.

In case of colliding event types the identifier of the highest priority event is applied according

to the priority chain:

**Error trigger > Calibration trigger > Emulator trigger >
> Final-OR(physics) > Random trigger > Test trigger (lowest priority)**

// Other trigger type codes can be programmed as needed (simulated events..)

Hardware Signals: TRIG_TYPE_A[15:0], TRIG_TYPE_B[15:0]

4.6.5 PTC0...7_BGO_PERIOD_L and -BGO_PERIOD_S

Write and read access.

The registers define the time intervals for periodically generated signals.

HARDRES and RESYNC signals are not generated periodically anymore, only by cmd_pulses, see PTC0...7_CMD signals below.

- V000A: PRIVATE_BGO_CMD_PERIOD sends a private code. The code and time within the LHC_orbit is defined in the BC-TABLE0.
- In case of colliding activities the following priority chain becomes valid:
Test_trigger > priv_bgo > priv_orbit > calibration cycle > traced_event
- Private GAP is not implemented
- **BCRES is sent every orbit.**

Address	Name	D15 - 0			
Base+ 4x 000A	PTCx_ BGO_PERIOD_L	15 – 12 not used	11 – 8 PRIVATE_ BGO_CMD_ PERIOD	7 – 4 PRIVATE_ ORBIT_ PERIOD	3 – 0 PRIVATE_ GAP_ PERIOD
Base+ 4x 000C	PTCx_ BGO_PERIOD_S	15 – 12 TEST_ TRIGGER_ PERIOD	11 – 8 CALIBR_ PERIOD	7 – 4 TRACE_ PERIOD	3 – 0 BCRES_ PERIOD

The periods are defined by bits of the orbit counter.

Format: EN, 2², 2¹, 2⁰, Example: hex A = 1010 → EN=1, period=2

8=1000 → EN=1, every orbit

Bits 15, 11, 7 and 3 are 'enable' bits to activate the periodic Signal Generator.

Programmable values for private orbits, private gaps, calibration cycles, traced events:

Version V001A (June 2008) values:

Programmed value	Time of period	Frequency	Orbit Counter Bit used	→ orbits
0 (every orbit)	~ 89.1 us	11 kHz	0	1
1	~ 178.2 us	5.6 kHz	1	2
2	~ 1.4 ms	700 Hz	4	16
3	~ 11.4 ms	100 Hz	100 Hz counter	112

			~ 9,9792 ms	
4	~ 91 ms	11 Hz	10	1024
5	~ 730 ms	1.37 Hz	13	8192
6	~ 11.7 s		17	131 072
7	~ 47 s		19	524 288

Version V0017 (2008) values:

Programmed value	Time of period	Frequency	Orbit Counter Bit used	→ orbits
0 (<i>every orbit</i>)	~ 89.1 us	11 kHz	0	1
1	~ 178.2 us	5.6 kHz	1	2
2	~ 1.4 ms	700 Hz	4	16
3	~ 11.4 ms	87 Hz	7	128
4	~ 91 ms	11 Hz	10	1024
5	~ 730 ms	1.37 Hz	13	8192
6	~ 11.7 s		17	131 072
7	~ 47 s		19	524 288

4.6.6 PTC0...7_EMU_DELAY

Delays for Emulator signals BCRES (Bgo) and RESYNC(Bgo).

address	name	15 - 8		7 - 0	
bb4x 000E	PTCx_ EMU_ DELAY	EMU_RES_DLY		EMU_BCR_DELAY	
		res dly74	res dly30	bcr dly74	bcr dly30

Default value: X'0000'

A delay consists of the sum of 2 delay units:

EMU_RES_DLY = res_dly74 + res_dly30

EMU_BCR_DLY = bcr_dly74 + bcr_dly30

// The delay modules add an inherent delay of 2 BC.

// max. value = 32 = 15+15+2 written in hex. notation: = F+F+2

4.6.7 PTC0...7_P_STATUS register

Read access only.

The register contains the combined STATUS of the connected detector partitions and the encoded status of the PT-Controller. Bit 15 shows if the status bits are frozen because of an error state.

Address	Name	D15 - 0
---------	------	---------

4x 0010	PTCx_P_STATUS	See description below.
----------------	---------------	------------------------

STATUS REGISTER

Bit 15: FROZEN_PTC not implemented

14: TCS_ROP_FULL // FIFOs for event records are full.

13: TCS_ROP_WARN // FIFOs for event records are 75% full
 // = WARN75 of COMMON_STATUS register.

12: PTC_ERROR // 1 = State machine is in error state

11 -8: PTC_STATUS3...0 // encoded status bits according to table below.

STATUS of Partition Controller

CODE	Status name	Remark	Remark
0000	DISCONNECTED	---	Should never be possible
0001	WARNING	PTC allows 'low rate' triggers	Depends from status of connected detector partitions.
0010	OUT_OF_SYNC	PTC is sending a 'RESYNC' or 'HARDRES' Bgo command.	When successful the PTC returns to WARNING or READY mode, otherwise it goes to ERROR state
0100	BUSY	PTC is starting a RUN or a connected detector partition is BUSY.	PTC inhibits triggers and waits until detector partitions become ready or warning again.
1000	READY	PTC allows normal trigger rate.	
1010	IDLE	PTC is IDLE status	Waiting for the VME command pulse 'START_RUN'. A 'STOP_RUN' returns the PTC back to IDLE state.
1100	ERROR	A permanent error forced the PTC into the error status.	The PTC waits for VME commands: STOP_RUN forces the PTC to send BGo = 'STOP_RUN' to the detector partitions and to returns to the IDLE state. HARDRES forces the PTC to send: BGo = 'HARD_RESET', then Bgo = 'RESYNC' and then Bgo = 'RESET_EVENTNR' and to go to BUSY state. Depending from partitions it enters then either READY, WARNING, OUT_OF_SYNC or again ERROR.

others	Bad code	Illegal codes	
--------	----------	---------------	--

DECODED DETECTOR STATUS → input to PTC

7: =0	// free
6: BAD CODE	// 7...0 decoded status bits of connected partitions
5: DISCONNECTED	// an assigned partition is disconnected
4: ERROR	// an assigned partition shows an error
3: OUT_OF_SYNC	// an assigned partition is out of sync
2: BUSY	// an assigned partition is busy
1: WARNING	// an assigned partition is in warning state
0: READY	// all assigned partitions are ready

4.6.8 PTC0...7_FSM_STATES

bb4x 0012	PTCx_FSM_STATES	D15 -0
------------------	-----------------	--------

Shows the state of the PTC- STATE MACHINE.

SM-CODE (hex)	SM-STATE (hex)	PTC-CODE (hex)	ACTION
0001	IDLE	A = idle	after Power up, waiting for 'start run'
0002	INI_RES	4 = busy	sends Bgo-command = RESYNC
0003	INIT	4 = busy	--
0004	RES_ORBIT	4 = busy	sends Bgo-command = RES_ORBIT_NR
0005	START	4 = busy	sends Bgo-command = START
0006	CLR_EVNR	4 = busy	sends Bgo-command = RES_EVENT_NR
0007	BUSY	4 = busy	inhibit L1A, waiting for 'ready' or 'warning'
0008	READY	8 = ready	sending L1A with normal rate rules
0009	WARN	1 = warning	inhibits L1A // L1A with low rate rules
0010	RESUME1	4 = busy	returning after error or out_of_sync states
0011	RESUME2	4 = busy	returning after error or out_of_sync states
0012	RESYNC2	4 = busy	sends Bgo-command = RESYNC when sends 'resync' by VME in IDLE mode
0013	CLR_EVNR2	4 = busy	sends Bgo-command = RES_EVENT_NR done automatically after RESYNC2
EE01	OUT_OF_SYNC	2 = out of sync	inhibits L1A, waits either for 'ready' or vme-cmds 'resync' or 'stop run'
EE02	ERROR	C = error	inhibits L1A, waits either for 'ready' or vme-cmds 'hard_res' or 'stop_run'
EE03	RESYNC1	4 = busy	inhibits L1A, vme-cmd 'resync' sending Bgo-command = RESYNC during run

EE04	HARDRES1	4 = busy	inhibits L1A, vme-cmd 'hardres' sending Bgo-command = HARD_RESET during run
F001	STOP1	4 = busy	stop_run return path
F002	STOP2	4 = busy	stop_run return path
F003	STOP3	4 = busy	sends Bgo-command = STOP
EEE0	ERROR0	C = error	Error after initialization: Waits for vme cmds stop or resync or hardres by vme cmd.
E002	RESYNC0	4 = busy	Error after initialization: vme-cmd 'resync' sending Bgo-command = RESYNC
E003	HARDRES0	4 = busy	Error after initialization: vme-cmd 'hardres' sending Bgo-command = HARD_RESET
D000	DISCON= NECTED	0=disconnect	disconnected input to PTC, waiting for reconnection or vme-cmd 'stop_run'
0FF1	PANIC1	4 = busy	PANIC_BUTTON forces PTC to IDLE not implemented in V0015
0FF2	PANIC2	4 = busy	PANIC_BUTTON forces PTC to IDLE not implemented in V0015
0FF3	PANIC3	4 = busy	PANIC_BUTTON forces PTC to IDLE not implemented in V0015

4.6.9 PTC0...7_EMU_TRIG_DELAY

bb4x 00C0	PTCx EMU TRIG DELAY	w/r	15 : 0	Synchronizing Emulator-trigger to Finor
------------------	----------------------------	-----	--------	--

The delay is used to make the Inhibit/Trigger signal from the Emulator concurrent with the FINOR signal.

- Programming the delay register
- DELAY =0 0000 0000 0000 0000
- DELAY =1 0000 0000 0000 0001
- DELAY =2 0000 0000 0000 0010
- DELAY =3 0000 0000 0000 0011
- DELAY =C+B+A+3 cccc bbbb aaaa 0011
- For DELAY <4 the bits 15-4 have to be =0 !!
- !! Bits 3,2 are always =0; ..must not be set by software
- !! But if they are set then case 'DELAY =C+B+A+3' is assumed by hardware.

4.6.10 PTC0...7 MONITORING COUNTERS

Read access only. PTCx = PTC0...PTC7 (x= 0 ... 7)

40 bit counters implemented with bit41-47 =X"00". Extension to 44 or 48 bits is possible if required.

Synchronised readout:

Wait until Luminosity Counter has changed then read all monitoring counters to get the contents from the same luminosity segment.

4.6.10.1 Overview

Address	Name	COUNTER bits		TIME until overflow
4x 0020	PTCx TRIGGER NR 31 16	bits 31 - 16	all PTC	11.9h
4x 0022	PTCx TRIGGER NR 15 0	bits 15 - 0	all PTC	at 100kHz
4x 0024	PTCx EVENT NR 31 16	bits 31 - 16	all PTC	
4x 0026	PTCx EVENT NR 15 0	bits 15 - 0	all PTC	
4x 0028	PTCx FINOR DISTRIBUTED 31 16	bits 31 - 16	all PTC	
4x 002A	PTCx FINOR DISTRIBUTED 15 0	bits 15 - 0	all PTC	
4x 002C	PTCx CAL TRIGGER 31 16	bits 31 - 16	all PTC	
4x 002E	PTCx CAL TRIGGER 15 0	bits 15 - 0	all PTC	
40 0030	PTCx RANDOM TRIGGER 31 16	bits 31 - 16	all PTC	
40 0032	PTCx RANDOM TRIGGER 15 0	bits 15 - 0	all PTC	
40 0034	PTC0 TEST TRIGGER 31 16	bits 31 - 16	PTC0 only	+error trigger
40 0036	PTC0 TEST TRIGGER 15 0	bits 15 - 0	PTC0 only	
40 0038	PTC0 FINOR GENERATED 31 16	bits 31 - 16	PTC0 only	
40 003A	PTC0 FINOR GENERATED 15 0	bits 15 - 0	PTC0 only	
40 003C	PTC0 FINOR IN INVALID BC 31 16	bits 31 - 16	PTC0 only	
40 003E	PTC0 FINOR IN INVALID BC 15 0	bits 15 - 0	PTC0 only	
4x 0040 ++	until 4x_005E free	-----	-----	
4x 0060 ++	..0060 =cmd pulse,			
40 0070	PTCx DEADTIME 47 32	bits 47 - 32	all PTC	48bit→81d
40 0072	PTCx DEADTIME 31 16	bits 31 - 16	all PTC	44b →121h
40 0074	PTCx DEADTIME 15 0	bits 15 - 0	all PTC	40bit→7.6h
40 0076	PTCx LOST FINOR 47 32	bits 47 - 32	all PTC	
40 0078	PTCx LOST FINOR 31 16	bits 31 - 16	all PTC	
40 007A	PTCx LOST FINOR 15 0	bits 15 - 0	all PTC	
40 007C	PTC0 DEADTIMEA 47 32	bits 47 - 32	PTC0 only	
40 007E	PTC0 DEADTIMEA 31 16	bits 31 - 16	PTC0 only	
40 0080	PTC0 DEADTIMEA 15 0	bits 15 - 0	PTC0 only	
40 0082	PTC0 LOST FINORA 47 32	bits 47 - 32	PTC0 only	
40 0084	PTC0 LOST FINORA 31 16	bits 31 - 16	PTC0 only	
40 0086	PTC0 LOST FINORA 15 0	bits 15 - 0	PTC0 only	
40 0088	PTC0 PRIV DEADTIMEA 47 32	bits 47 - 32	PTC0 only	
40 008A	PTC0 PRIV DEADTIMEA 31 16	bits 31 - 16	PTC0 only	

40 008C	PTC0_PRIV_DEADTIMEA_15_0	bits 15 - 0	PTC0 only	
40 008E	PTC0_PTCSTATUS_DEADTIMEA_47_32	bits 47 - 32	PTC0 only	
40 0090	PTC0_PTCSTATUS_DEADTIMEA_31_16	bits 31 - 16	PTC0 only	
40 0092	PTC0_PTCSTATUS_DEADTIMEA_15_0	bits 15 - 0	PTC0 only	
40 0094	PTC0_THROTTLE_DEADTIMEA_47_32	bits 47 - 32	PTC0 only	
40 0096	PTC0_THROTTLE_DEADTIMEA_31_16	bits 31 - 16	PTC0 only	
40 0098	PTC0_THROTTLE_DEADTIMEA_15_0	bits 15 - 0	PTC0 only	
40 009A	PTC0_CALIBRATION_DEADTIMEA_47_32	bits 47 - 32	PTC0 only	
40 009C	PTC0_CALIBRATION_DEADTIMEA_31_16	bits 31 - 16	PTC0 only	
40 009E	PTC0_CALIBRATION_DEADTIMEA_15_0	bits 15 - 0	PTC0 only	
40 00A0	PTC0_TIMESLOT_DEADTIMEA_47_32	bits 47 - 32	PTC0 only	
40 00A2	PTC0_TIMESLOT_DEADTIMEA_31_16	bits 31 - 16	PTC0 only	
40 00A4	PTC0_TIMESLOT_DEADTIMEA_15_0	bits 15 - 0	PTC0 only	
40 00A6	PTC0_OTHER_DEADTIMEA_47_32	bits 47 - 32	PTC0 only	V0017:
40 00A8	PTC0_OTHER_DEADTIMEA_31_16	bits 31 - 16	PTC0 only	
40 00AA	PTC0_OTHER_DEADTIMEA_15_0	bits 15 - 0	PTC0 only	
40 00AC	PTC0_NR_OF_RESETS_31_16	bits 31 - 16	PTC0 only	
40 00AE	PTC0_NR_OF_RESETS_15_0	bits 15 - 0	PTC0 only	
40 00B0	PTC0_EMU_TRIG_DEADTIME_47_32	bits 47 - 32	PTC0 only	V1023
40 00B2	PTC0_EMU_TRIG_DEADTIME_31_16	bits 31 - 16	PTC0 only	V1023
40 00B4	PTC0_EMU_TRIG_DEADTIME_15_0	bits 15 - 0	PTC0 only	V1023

4.6.10.2 Description

counter	bits	reset	incr
PTCx_TRIGGER_NR ^{ACC}	32	start_run	LIAs since start of Run
PTCx_EVENT_NR ^{ACC}	32	start_run + res evnr	LIAs since last RESYNC
PTCx_FINOR_DISTRIBUTED ^R	32	start_run+ new_lum_seg	FIN_ORs sent as L1A
PTCx_CAL_TRIGGER ^R	32	start_run+ new_lum_seg	CAL_TRIGs sent as L1A
PTCx_RANDOM_TRIGGER ^R	32	start_run+ new_lum_seg	RNDM_TRIGs sent as L1A
PTC0_TEST_TRIGGER ^R	32	start_run+ new_lum_seg	TEST or ERROR TRIGGERS sent as L1A
PTC0_FINOR_GENERATED ^R	32	start_run+ new_lum_seg	all FIN_ORs from FDL board
PTC0_FINOR_IN_INVALID_BC ^R	32	start_run+ new_lum_seg	FIN_OR when beam is inactive
PTCx_DEADTIME ^{ACC}	40	start_run	total dead-time
PTCx_LOST_FINOR ^{ACC}	40	start_run	FIN_ORs during dead_time
PTC0_DEADTIMEA ^{ACC}	40	start_run	total dead time +)

PTC0_LOST_FINORA ^{ACC}	40	start_run	lost FINORs +)
PTC0_PRIV_DEADTIMEA ^{ACC}	40	start_run	private orbit dead-time +)
PTC0_PTCSTATUS_DEADTIMEA ^{ACC}	40	start_run	ptc inhibits FINORs +)
PTC0_THROTTLE_DEADTIMEA ^{ACC}	40	start_run	Throttle rule inhibits L1A +)
PTC0_CALIBRATION_DEADTIMEA ^{ACC}	40	start_run	Calibration cycles inhibit L1A +)
PTC0_TIMESLOT_DEADTIMEA ^{ACC}	40	start_run	+))
PTC0_OTHER_DEADTIMEA ^{ACC}	40	start_run	V0017: not used
PTC0_NR_OF_RESETS_31_16 ^{ACC}	32	start_run	Synchronisation cycles

+) while beam is active

R) RATE COUNTER: # per lum_segment

After every luminosity-segment-period the contents of the trigger counters are stored in read only registers and the counters are then reset. The software reads therefore the **number of trigger per luminosity-segment**.

^{ACC}) ACCUMULATING COUNTER: since start of run, resp. since last Resync

The dead-time-counters are cleared only at begin of a run and their contents are saved in read-only registers at the end of every luminosity-segment-period. Therefore software reads the **accumulated dead-time** from start of the run until the end of the previous luminosity-segment-period.

The trigger number and event number show the actual value independently from luminosity-segment-periods since start of run respectively since the last ‘resynchronization’ procedure.

PTC0_TRIGGER_NR shows the number of all L1A signals from all PTC(i) since ‘PTC0-start_of_run’ when running in ‘physics’ mode, when all PTCs will send event records to the GTFE board, to DAQ and to EVM (Event Manager).

PTCx_TRIGGER_NR shows the number of all L1A signals from this PTC(i) since start of run.

PTCx_EVENT_NR shows the number of L1A signals from this PTC(i) since last resynchronisation procedure.

PTCx_FINOR_DISTRIBUTED shows the number of FIN_OR signals (=‘physics’+technical triggers) distributed as L1A signals.

PTCx_CAL_TRIGGER shows the number of calibration triggers distributed as L1A signals.

PTCx_RANDOM_TRIGGER shows the number of random triggers distributed as L1A signals.

PTC0_TEST_TRIGGER shows the number of test triggers plus error triggers distributed as L1A signals.

If enabled every time the PTC goes into the error status an ERROR TRIGGER will be sent.

PTC0_FINOR_GENERATED shows the number of FIN_OR signals generated on the FDL board. The FDL board contains also counters for 8 FINOR signals.

PTC0_FINOR_IN_INVALID_BC shows the number of FIN_OR signals generated during invalid bunch crossings when either there was no collision (beam counters) or the valid_bc was set to =0 (LHC gaps).

PTCx_DEADTIME shows number of beam crossings where L1A was potentially inhibited.

PTCx_LOST_FINOR shows number of beam crossings where L1A was potentially inhibited, and FIN_OR is true.

PTC0_DEADTIMEA shows number of **active** beam crossings where L1A was potentially inhibited.

PTC0_LOST_FINORA shows number of **active** beam crossings where L1A was potentially inhibited, and FIN_OR is true.

PTC0_PRIV_DEADTIMEA shows number of **active** beam crossings where L1A was potentially inhibited by 'private orbits'.

PTC0_PTCSTATUS_DEADTIMEA shows number of **active** beam crossings where L1A was potentially inhibited by PTC state machine (busy, resynchronisation time, error state..).

After 'start run' an inherent ptc-status deadtime of 2 orbits+1 bx is inserted.

PTC0_THROTTLE_DEADTIMEA shows number of **active** beam crossings where L1A was potentially inhibited by throttle rules.

PTC0_CALIBRATION_DEADTIMEA shows number of **active** beam crossings where L1A was potentially inhibited by calibration cycles.

PTC0_TIMESLOT_DEADTIMEA shows number of **active** beam crossings where L1A was potentially inhibited by inactive timeslots when running in multi-partition mode.

PTC0_OTHER_DEADTIMEA (V0017: removed) has shown +/- 2bx dead-time around Bgo-commands

PTC0_NR_OF_RESETS shows number of resynchronisation cycles since start of run.

4.6.11 PTC0...7_CMD pulse signals

Write access only. $PTCx = PTC0...7$

PTC=_CMD signals are spurious pulses to start a procedure in the TCS chip and cannot be read back.

Address	Name	D15 - 0
Base+ 4x 0060	PTCx_CMD	See description below.

Write only access.

V0015: COMMON_CMD(5=): RESET_ORBIT_NR resets the ORBIT Counter!!!

V0014: 'START_RUN for PTC0 resets also the ORBIT Counter!!!

Hardware Signals: PTC[7:0]_CMD[9:0]

Bits 15,14, 13 are unused

12: DO_PRIV_BGO // Send a BGO command with a 'private' code. See also the BC_TABLE0 and the BGO_PERIOD register.

11: DO_TEST_TRIGGER // Send one L1A as test trigger

10: DO_PRIVATE_ORBIT // insert one private orbit

9: UNFREEZE_PTC // removes the 'FROZEN_PTC' state,

	// not implemented in V000A
8: START_RNDM_TRIG	//start the Random Generator (→ async reset)
7: DO_A_TRACED EVENT	//Send one L1A and set trigger_type=traced in tcs-record
6: DO_A_CALIBR_CYCLE	//send BGO='test_enable' and then a L1A as programmed in the BC-table
5: HARD_RESET	// force the PTC to make a Hard_reset procedure
4: RESYNCHRONIZE	// force the PTC to make a 'resync' procedure
3: STOP_RUN	// Sends Bgo='stop_run' & stops PTC state machine
2: START_RUN	// Initialize the 'start run' procedure
1: START_PTC	// not used anymore
0: PANIC_BUTTON	// Sets the PTC_SM to IDLE mode when Bgo_control doesn't work because of bad BC table programming.

4.6.11.1 UNFREEZE_PTC (not implemented in V000A)

If the CMD register bits FREEZE_IF_BAD_BCNR and/or FREEZE_IF_ERR are set then the bits in the P_STATUS register are frozen in case of errors. The state of the connected partitions and of the PTC logic can be checked reading the P_STATUS register. The UNFREEZE_PTC returns the latches to transparent mode so that the actual values can be read.

4.6.11.2 Restart the RANDOM TRIGGER

Random Triggers are inserted between Physics triggers according to the contents of the RANDOM_FREQ register when the PTCx_CMD_REG(0) =1= 'EN_RNDM_TRIG' allowing the generator to send triggers. If the random generator does not work a command pulse [PTCx_CMD CMD(8) 'START_RNDM_TRIG' =1] restarts the circuit.

4.6.11.3 SOFTWARE induced BGO instructions

Normally CALIBR_CYCLE, HARD_RESET and RESYNCHRONIZE are done periodically during the run or are sent by the PTC according to the STATUS bits of the connected readout electronics. Instead of periodic signals the software also can do these functions.

The BGO commands START_RUN, STOP_RUN are sent exclusively by software.

4.6.11.4 Start a PTC_PTC (Partition Controller)

- If the TCS board is inactive or in an undefined state then send RESET_TCS to reset the TCS chip (See addresses in VME chip.)
- If other PTC partitions are already active then send
 - o RESET_PTC to reset the control logic of the PTC partition.
- Now load all registers and the BC-Tables of this PTC partition.
- Then send START_PTC to move the PTC into the IDLE state that can be checked by the Run Control program.

- The Run Control program sends now START_RUN to start the first synchronisation of the connected partitions. See also the State Diagram of the PTC.

4.6.11.5 RESET_PTC

The reset pulse

- Clears FROZEN_PTC to return PSTATUS bits to actual values
- Resets the PTC into a 'DISCONNECTED' state:
 - o The PTC is inactive and all states are =0.
- Resets the PTC- Orbit number and the BAD_BCNr error flag
- Resets the BGO controller immediately into IDLE state.
- Resets the Random Trigger Generator
- removes CALIBR_CYCLE and TRACE_CYCLE
- Resets the Event number

Clears the RUNNING Flip-Flop.

5 TCSMON_CHIP

The TCSMON chip has to run synchronously to the TCS chip. It accepts the same VME commands (RESET, HARDRES, FREEZE...), the same Partitioning Assignment, the same Orbit Length etc. The TCSMON also receives the same synchronization signals from the Backplane. Therefore some registers are duplicated in the TCSMON chip but are read back only from the TCS chip.

5.1 Firmware Versions

5.1.1 V0003

First FPGA Advantage Version 2008

TCSM_CMD_REG(7:6) := save_mode(1:0)

TCSM_CMD_PULSE(4): **RESET_ORBIT_CNTR**

5.1.2 V0002 ... from 2004

Innoveda Version from 2004.

5.2 Description (to be updated)

The TCSMON chip monitors the status bits of 32 Detector Partitions and 8 PTC-Partitions. Each Partition sends 4 encoded status bits. If the status of a Partition changes the new value together with the actual BC number will be written into a 16 bit x 1k RAM. If the Orbit number has been incremented since the last write access then the new orbit value will be written in advance.

Format of the monitoring data.

RAM address (hex)	RAM content (hex)	Comments
0	0000	New orbit value follows
1	0000	New ORBIT[31:16]
2	0000	New ORBIT[15:0]
3	SBBB	STATUS[3:0] & BCNR[11:0]
4	SBBB	STATUS[3:0] & BCNR[11:0]
5	SBBB	STATUS[3:0] & BCNR[11:0]
6	SBBB	STATUS[3:0] & BCNR[11:0]
7	0000	New orbit value follows
8	0000	New ORBIT[31:16]
9	0000	New ORBIT[15:0]
A	SBBB	STATUS[3:0] & BCNR[11:0]
B	SBBB	STATUS[3:0] & BCNR[11:0]
C	SBBB	STATUS[3:0] & BCNR[11:0]
....

S= STATUS[3:0], BBB= BCNR[11:0], OOOO =orbit bits

The TCSMON chip receives the status bits of the 8 PTC-Partitions and of 24 Detector Partitions from Channel Links. Each Link carries the status bits of 4 Partitions. The other 4 bits are reserved for future applications and bit 20 is ignored. The 8 Emulator Partitions and the Global Trigger status are received as parallel data bits.

The RAMs contain at least the last 256 states if they are stopped to be readout via the VME interface.

The TCSMON chip runs synchronously to the TCS chip. It accepts the same VME commands (RESET, HARDRES, FREEZE...), the same Partitioning Assignment, the same Orbit Length etc. The TCSMON also receives the same synchronization signals from the Backplane.

5.3 VME_addresses

VME bus specification 1987:

Tables 2 and 4 combined:

Access	DS1*	DS0*	A01	LWORD*	D31-24	D23-16	D15-8	D7-0
BYTE0-1	low	low	0	high			Byte0	Byte1
BYTE2-3	low	low	1	high			Byte2	Byte3
BYTE0-3	low	low	0	low	Byte0	Byte1	Byte2	Byte3
BYTE0	low	high	0	high			Byte0	
BYTE1	high	low	0	high				Byte1

BYTE2	low	high	1	high			Byte2	
BYTE3	high	low	1	high				Byte3

A31- A24	A23 – A20	A19 – A16	COMMENT
TCS-base-address	Chip Name		
Base	8 =TCSM	8	Registers & Counters

Address bits A19-1 are connected to the TCSM chip.

5.3.1 Register Overview

A31-24	A23 – 1(0)	Register Name	Access mode	Comment
Base	+ 88 0000	TCSM_CMD_Register	write / read	<i>Command register</i>
Base	+ 88 0002	TCSM_CMD_PULSE	write /----	<i>Command pulses</i>
Base	+ 88 0004	GT_STATUS	-----/read	<i>GT crate</i>
Base	+ 88 0006	SIM_GT_STATUS	write / read	<i>Sim GT and GTFE</i>
Base	+ 88 0008	SIM_PARTS_H	write / read	<i>Enable Simu values</i>
Base	+ 88 000A	SIM_PARTS_L	write / read	<i>Enable Simu values</i>
Base	+ 88 000C	STATUS3128	-----/read	<i>Emulators 7-4(31-28)</i>
Base	+ 88 000E	STATUS2724	-----/read	<i>Emulators 3-0(27-24)</i>
Base	+ 88 0010	STATUS2320	-----/read	<i>Partitions 23 –20</i>
Base	+ 88 0012	STATUS1916	-----/read	<i>Partitions 19 - 16</i>
Base	+ 88 0014	STATUS1512	-----/read	<i>Partitions 15 -12</i>
Base	+ 88 0016	STATUS1108	-----/read	<i>Partitions 11 -8</i>
Base	+ 88 0018	STATUS0704	-----/read	<i>Partitions 7 -4</i>
Base	+ 88 001A	STATUS0300	-----/read	<i>Partitions 3 -0</i>
Base	+ 88 001C	DAQ_STATUS0704	-----/read	<i>DAQ-Part 7 -4</i>
Base	+ 88 001E	DAQ_STATUS0300	-----/read	<i>DAQ-Part 3 -0</i>
Base	+ 88 0020	V_STATUS2320	write / read	<i>Sim_value Part 23-20</i>
Base	+ 88 0022	V_STATUS1916	write / read	<i>Sim_value Part 19-16</i>
Base	+ 88 0024	V_STATUS1512	write / read	<i>Sim_value Part 15-12</i>
Base	+ 88 0026	V_STATUS1108	write / read	<i>Sim_value Part 11-8</i>
Base	+ 88 0028	V_STATUS0704	write / read	<i>Sim_value Part 7-4</i>
Base	+ 88 002A	V_STATUS0300	write / read	<i>Sim_value Part 3-0</i>
Base	+ 88 002C	V_DAQ_STATUS0704	write / read	<i>Sim_value DAQ 7-4</i>
Base	+ 88 002E	V_DAQ_STATUS0300	write / read	<i>Sim_value DAQ 3-0</i>
Base	+ 88 0044	TCSM_CHIP_ID_H	-----/read	<i>Chip Identifier 31:16</i>

Base	+ 88 0046	TCSM_CHIP_ID_L	-----/read	<i>Chip Identifier 15:0</i>
Base	+ 88 0048	TCSM_VERSION_H	-----/read	<i>Version number 31:16</i>
Base	+ 88 004A	TCSM_VERSION_L	-----/read	<i>Version number 15:0</i>

STATUS Counters:

8 bits used; bits 15 – 0 = X"00"; 42+42 counters

Base+88 0070 ...88 00C2	RED_CNTR_P0...P31, DAQ0 ...7, GT, GTFE_EVM
Base+88 0100 ... 88 0152	YELLOW_CNTR_P0...P31, DAQ0 ...7, GT, GTFE_EVM

PARTITION NAMES

Name of partitions are stored in registers: 4 characters ASCII code.

Default values: status of 17 March 08

Address	Register name	15 - 8	7 - 0
Base+88 0200	NAME_PART0_H	1 st char	2 nd char
Base+88 0202	NAME_PART0_L	3 rd char	4 th char
Base+
Base+88 0200	NAME_PART31_H	1st char	2nd char
Base+88 0202	NAME_PART31_L	3rd char	4th char

5.3.2 COMMAND REGISTER

Address	Name	D15-0
Base+ 88 0000	TCSM_CMD_REG	See description below Default:= X"0100"

Bit 15-9: not used

Bit 8: **ACCUM_MODE**

= **1** While monitoring (monitoring_flag = 1) the Status Counters are accumulating and stop with X"FF" in case of an overflow.

= **0** While monitoring the "new_luminosity_segment" signals reset the status counters periodically and move their contents into registers.

//The monitoring_flag is stored as bit 15 in register STATUS_GT_CRATE.

Bit 7: **SAVE_MODE1**

Bit 6: **SAVE_MODE0**

= **00** Shows always actual input states regardless of monitoring flag.

= **01** Saves all input states when a warning or busy shows up and stop monitoring → mon_flag =0

= **10** Saves all input states when error/out_of_sync/disconnected shows up and stop monitoring → mon_flag =0

= **11** as option =00

Procedure: Read actual states and monitoring counters (per lum_segment)

- Set save_mode = 00
- Read status registers to know the actual values.
- Send ‘start_monitoring’ → monitoring_flag = 1
 - Every ‘non-ready’ status signal will increase a counter.
 - The counter are reset periodically
- Read all monitoring counters showing the number of ‘bad’ input states
- Send ‘stop_monitoring’ → mon_flag = 0 to end the monitoring task.

Procedure: Read states when an input becomes not ready to find the partition making problems.

- Set save_mode =01 or 10
 - Send ‘start_monitoring’ → monitoring_flag = 1
 - Wait until ‘mon_flag’ becomes =0 because of a ‘bad’ input status.
 - Read status registers and monitoring counters
- //‘stop_monitoring’ also will stop the procedure.

// not implemented:

Bit 5 : MAKE_ERR_PULSE

Bit 4 : EN_IRQ_DIS

Bit 3 : EN_IRQ_WARN

Bit 2 : EN_IRQ_ERR

Bit 1 : EN_ROBUS

Bit 0: EN_MONLINK

=1 Enables Channel Link chip to append TCSM data to the GT-record after a L1A

= 0 default

5.3.3 COMMAND PULSES

Address	Name	D15-0
Base+ 88 0002	TCSM_CMD_PULSE	See description below

Remark: *The software can start monitoring counters and waiting for status flag concurrently by one VME instruction.*

Bit 15 –6: not used

Bit 5: **RESET_TCSM_COUNTERS** → resets **all** red-, yellow status counters

Bit 4: **RESET_TCSM_ALL** → resets state machines ???

Bit 3: **STOP_WAITING** → stops waiting for busy/warning or error/out_of_sync_disconnected status.

Bit 2: **START_WAITING** → Start waiting for busy/warning or error/out_of_sync_disconnected status. When one of the flags arrives then the 'waiting_flag' will be cleared.

Bit 1: **STOP_MONITORING** → stops monitoring counters of status bits immediately

Bit 0: **START_MONITORING** → Start monitoring counters of input states or continue when monitoring has been stopped either by the TCS chip or by a VME command. This instruction is used after all monitoring counters have been read out after a stop signal.

5.3.4 STATUS_GT_CRATE

Address	Name	D15	D14	D13-8	D 7-4	D 3-0
Base+ 88 0004	STATUS_GT _CRATE	MONITORING FLAG	WAITING FLAG	free	STATUS_ EVM[3:0]	STATUS_ GT[3:0]

Bit 15: MONITORING_FLAG

1= monitoring of input states: red/yellow counters will be incremented

0= no monitoring

Bit 14: WAITING_FLAG

1= waiting either for busy/warning or for error/out_of_sync/disconnected status when 'start_monitoring' has been issued.

0= not waiting anymore because:

- save_mode= 01/10: status reg's show states at appearance of flag or
- save_mode=00 : status reg's show actual states

Bit 7-4: Status sent by the EVM chip of the GTFE board. The value includes the status of both S-links and of the DAQ chip as well as the EVM chip.

Bit 3-0: Combined status of all GT-boards; received from the FDL board.

5.3.5 SIMULATION Control Registers

To be done: bit 10: SIM_LUM_STATUS, bits15-12: V_STATUS_LUM(3:0)

Address	Name	D15 -10	D9	D8	D 7-4	D 3-0
Base+ 88 0006	SIM_GT_ STATUS	free	SIM_ GTFE status	SIM_ GT status	V_STATUS_ GTFE[3:0]	V_STATUS_ GT[3:0]

Default value: ,0088' hex

SIM_GTFE status: =1 sends simulation values to TCS chip instead of GTFE status bits.
=0 default value

SIM_GT status: =1 sends simulation values to TCS chip instead of GT status bits.
 ^ =0 default value

V_STATUS_GTFE[3:0] = simulated status of the GTFE board
 Default value: =8 = 'READY'

V_STATUS_GT[3:0] = simulated status of the GT boards
 Default value: =8 = 'READY'

Status bit encoding (4 bits)

0 = F=disconnected 1= warn
 2=out_of_sync 4=busy
 8=ready C=error

Address	Name	D15-0	
Base+ 88 0008	SIM_ PARTS_H	SIM_DAQ_PART7...0	SIM_PART23.....16
Base+ 88 000A	SIM_ PARTS_L	SIM_PART150	

Default values = 0000 hex

SIM_PART0 = 1 sends simulation value of PART0 to TCS chip instead of true status.

SIM_PART1 = 1 sends simulation value of PART1 to TCS chip instead of true status.

.....

SIM_PART23 = 1 sends simulation value of PART23 to TCS chip instead of true status.

SIM_DAQ_PART0 = 1 sends simulation value of DAQ_PART0 to TCS chip.

.....

SIM_DAQ_PART7 = 1 sends simulation value of DAQ_PART0 to TCS chip.

5.3.6 Partition Status Registers

The STATUSxxxx registers show the actual status of all sub-detector- and DAQ-partitions.

Partitions 31 – 24 are foreseen for Emulators (Tracker APVE, ..) with direct inputs to the TCS chips.

Address	Name	D15-12	D11-8	D7-4	D3-0
Base+ 88 000C	STATUS 3128	STATUS_ PART31	STATUS_ PART30	STATUS_ PART29	STATUS_ PART28
Base+ 88 000E	STATUS 2724	STATUS_ PART27	STATUS_ PART26	STATUS_ PART25	STATUS_ PART24

Address	Name	D15-12	D11-8	D7-4	D3-0
---------	------	--------	-------	------	------

Base+ 88 0010	STATUS 2320	STATUS_ PART23	STATUS_ PART22	STATUS_ PART21	STATUS_ PART20
Base+ 88 0012	STATUS 1916	STATUS_ PART19	STATUS_ PART18	STATUS_ PART17	STATUS_ PART16
Base+ 88 0014	STATUS 1512	STATUS_ PART15	STATUS_ PART14	STATUS_ PART13	STATUS_ PART12
Base+ 88 0016	STATUS 1108	STATUS_ PART11	STATUS_ PART10	STATUS_ PART9	STATUS_ PART8
Base+ 88 0018	STATUS 0704	STATUS_ PART7	STATUS_ PART6	STATUS_ PART5	STATUS_ PART4
Base+ 88 001A	STATUS 0300	STATUS_ PART3	STATUS_ PART2	STATUS_ PART1	STATUS_ PART0
Base+ 88 001C	STATUS DAQ_0704	STATUS_ DAQ_PART7	STATUS_ DAQ_PART6	STATUS_ DAQ_PART5	STATUS_ DAQ_PART4
Base+ 88 001E	STATUS DAQ_0300	STATUS_ DAQ_PART3	STATUS_ DAQ_PART2	STATUS_ DAQ_PART1	STATUS_ DAQ_PART0

The 4 bit values show the status of partitions.

Table 1 Input signals to central TCS explains the 4 bit status code.

5.3.7 SIMULATION Value Registers

The V_STATUSxxxx register are used to insert simulated states instead of the sub-detector and DAQ-partition input signals.

The Emulator Partitions 31 – 24 cannot be simulated because their inputs go directly into the TCS chip.

Address	Name	D15-12	D11-8	D7-4	D3-0
Base+ 88 0020	V_STATUS 2320	V_STATUS_ PART23	V_STATUS_ PART22	V_STATUS_ PART21	V_STATUS_ PART20
Base+ 88 0022	V_STATUS 1916	V_STATUS_ PART19	V_STATUS_ PART18	V_STATUS_ PART17	V_STATUS_ PART16
Base+ 88 0024	V_STATUS 1512	V_STATUS_ PART15	V_STATUS_ PART14	V_STATUS_ PART13	V_STATUS_ PART12
Base+ 88 0026	V_STATUS 1108	V_STATUS_ PART11	V_STATUS_ PART10	V_STATUS_ PART9	V_STATUS_ PART8
Base+ 88 0028	V_STATUS 0704	V_STATUS_ PART7	V_STATUS_ PART6	V_STATUS_ PART5	V_STATUS_ PART4
Base+ 88 002A	V_STATUS 0300	V_STATUS_ PART3	V_STATUS_ PART2	V_STATUS_ PART1	V_STATUS_ PART0

Base+ 88 002C	V_STATUS DAQ_0704	V_STATUS_ DAQ PART7	V_STATUS_ DAQ PART6	V_STATUS_ DAQ PART5	V_STATUS_ DAQ PART4
Base+ 88 002E	V_STATUS DAQ_0300	V_STATUS_ DAQ PART3	V_STATUS_ DAQ PART2	V_STATUS_ DAQ PART1	V_STATUS_ DAQ PART0

The 4 bit values are used to simulate the status of partitions.

Default values = 8888 hex ... = 'READY' to allow triggers.

See also Table 1 Input signals to central TCS to simulate other states.

Status bit encoding (4 bits)

0 = F=disconnected	1= warn
2=out_of_sync	4=busy
8=ready	C=error

5.3.8 Identifier and Version Registers

The Identifier and Version registers are used to identify the chip design.

Address	Name	D15-12	D11-8	D7-4	D3-0
Base + 88 0044	TCSM_CHIP_ID_H	0 0 0 1 ='GT'			
Base + 88 0046	TCSM_CHIP_ID_L	CARD NAME=5	CARD NR=1	CHIP NAME=4	CHIP NR =1
Base + 88 0048	TCSM_VERSION_H	VERSION NUMBER 31:16			
Base + 88 004A	TCSM_VERSION_L	VERSION NUMBER 15:0			

5.3.9 MONITORING COUNTERS

	name	15 - 8	7 - 0
Base+88 0070	RED_CNTR_P0	00	
Base+88 0072	RED_CNTR_P1	00	
Base+88 0074	RED_CNTR_P2	00	
Base+88 0076	RED_CNTR_P3	00	
Base+88 0078	RED_CNTR_P4	00	
Base+88 007A	RED_CNTR_P5	00	
Base+88 007C	RED_CNTR_P6	00	

Base+88 007E	RED_CNTR_P7	00	
Base+88 0080	RED_CNTR_P8	00	
Base+88 0082	RED_CNTR_P9	00	
Base+88 0084	RED_CNTR_P10	00	
Base+88 0086	RED_CNTR_P11	00	
Base+88 0088	RED_CNTR_P12	00	
Base+88 008A	RED_CNTR_P13	00	
Base+88 008C	RED_CNTR_P14	00	
Base+88 008E	RED_CNTR_P15	00	
Base+88 0090	RED_CNTR_P16	00	
Base+88 0092	RED_CNTR_P17	00	
Base+88 0094	RED_CNTR_P18	00	
Base+88 0096	RED_CNTR_P19	00	
Base+88 0098	RED_CNTR_P20	00	
Base+88 009A	RED_CNTR_P21	00	
Base+88 009C	RED_CNTR_P22	00	
Base+88 009E	RED_CNTR_P23	00	
Base+88 00A0	RED_CNTR_P24	00	
Base+88 00A2	RED_CNTR_P25	00	
Base+88 00A4	RED_CNTR_P26	00	
Base+88 00A6	RED_CNTR_P27	00	
Base+88 00A8	RED_CNTR_P28	00	
Base+88 00AA	RED_CNTR_P29	00	
Base+88 00AC	RED_CNTR_P30	00	
Base+88 00AE	RED_CNTR_P31	00	
Base+88 00B0	RED_CNTR_DAQ0	00	
Base+88 00B2	RED_CNTR_DAQ1	00	
Base+88 00B4	RED_CNTR_DAQ2	00	
Base+88 00B6	RED_CNTR_DAQ3	00	
Base+88 00B8	RED_CNTR_DAQ4	00	
Base+88 00BA	RED_CNTR_DAQ5	00	
Base+88 00BC	RED_CNTR_DAQ6	00	
Base+88 00BE	RED_CNTR_DAQ7	00	
Base+88 00C0	RED_CNTR_GT	00	
Base+88 00C2	RED_CNTR_GTFE_EVM	00	
Base+88 00C4	---	00	00
Base+88 00C6	---	00	00
Base+88 00C8	---	00	00

Base+88 00CA	---	00	00
Base+88 00CC	---	00	00
Base+88 00CE	---	00	00

	name	15 - 8	7 - 0
Base+88 0100	YELLOW_CNTR_P0	00	
Base+88 0102	YELLOW_CNTR_P1	00	
Base+88 0104	YELLOW_CNTR_P2	00	
Base+88 0106	YELLOW_CNTR_P3	00	
Base+88 0108	YELLOW_CNTR_P4	00	
Base+88 010A	YELLOW_CNTR_P5	00	
Base+88 010C	YELLOW_CNTR_P6	00	
Base+88 010E	YELLOW_CNTR_P7	00	
Base+88 0110	YELLOW_CNTR_P8	00	
Base+88 0112	YELLOW_CNTR_P9	00	
Base+88 0114	YELLOW_CNTR_P10	00	
Base+88 0116	YELLOW_CNTR_P11	00	
Base+88 0118	YELLOW_CNTR_P12	00	
Base+88 011A	YELLOW_CNTR_P13	00	
Base+88 011C	YELLOW_CNTR_P14	00	
Base+88 011E	YELLOW_CNTR_P15	00	
Base+88 0120	YELLOW_CNTR_P16	00	
Base+88 0122	YELLOW_CNTR_P17	00	
Base+88 0124	YELLOW_CNTR_P18	00	
Base+88 0126	YELLOW_CNTR_P19	00	
Base+88 0128	YELLOW_CNTR_P20	00	
Base+88 012A	YELLOW_CNTR_P21	00	
Base+88 012C	YELLOW_CNTR_P22	00	
Base+88 012E	YELLOW_CNTR_P23	00	
Base+88 0130	YELLOW_CNTR_P24	00	
Base+88 0132	YELLOW_CNTR_P25	00	
Base+88 0134	YELLOW_CNTR_P26	00	
Base+88 0136	YELLOW_CNTR_P27	00	
Base+88 0138	YELLOW_CNTR_P28	00	
Base+88 013A	YELLOW_CNTR_P29	00	
Base+88 013C	YELLOW_CNTR_P30	00	
Base+88 013E	YELLOW_CNTR_P31	00	
Base+88 0140	YELLOW_CNTR_DAQ0	00	

Base+88 0142	YELLOW_CNTR_DAQ1	00	
Base+88 0144	YELLOW_CNTR_DAQ2	00	
Base+88 0146	YELLOW_CNTR_DAQ3	00	
Base+88 0148	YELLOW_CNTR_DAQ4	00	
Base+88 014A	YELLOW_CNTR_DAQ5	00	
Base+88 014C	YELLOW_CNTR_DAQ6	00	
Base+88 014E	YELLOW_CNTR_DAQ7	00	
Base+88 0150	YELLOW_CNTR_GT	00	
Base+88 0152	YELLOW_CNTR_GTFE_EVM	00	
Base+88 0154	---	00	00
Base+88 0156	---	00	00
Base+88 0158	---	00	00
Base+88 015A	---	00	00
Base+88 015C	---	00	00
Base+88 015E	---	00	00

5.3.10 Name of Detector Partitions

The name registers are only used to define names of assigned detectors partitions and are used in the TCS_GUI. No other logic function.

The partition names are stored as 4 character ASCII code.

Default values: status of 17 March 08

PART 23 ... 31 = Emulator partitions with direct connection to TCS chip.

Address	Register name	15 - 8	7 - 0	Default Detector Name	Hex value
Base+88 0200	NAME_PART0_H	1 st char	2 nd char	EB	4542
Base+88 0202	NAME_PART0_L	3 rd char	4 th char	+	2B20
Base+88 0204	NAME_PART1_H	1 st char	2 nd char	EB	4542
Base+88 0206	NAME_PART1_L	3 rd char	4 th char	-	2D20
Base+88 0208	NAME_PART2_H	1 st char	2 nd char	EE	4545
Base+88 020A	NAME_PART2_L	3 rd char	4 th char	+	2B20
Base+88 020C	NAME_PART3_H	1 st char	2 nd char	EE	4545
Base+88 020E	NAME_PART3_L	3 rd char	4 th char	-	2D20
Base+88 0210	NAME_PART4_H	1 st char	2 nd char	GT	4754
Base+88 0212	NAME_PART4_L	3 rd char	4 th char	st	7374
Base+88 0214	NAME_PART5_H	1 st char	2 nd char	HB	4842
Base+88 0216	NAME_PART5_L	3 rd char	4 th char	Ea	4561
Base+88 0218	NAME_PART6_H	1 st char	2 nd char	HB	4842

Base+88 021A	<u>NAME PART6 L</u>	3 rd char	4 th char	Eb	4562
Base+88 021C	<u>NAME PART7 H</u>	1 st char	2 nd char	HB	4842
Base+88 021E	<u>NAME PART7 L</u>	3 rd char	4 th char	Ec	4563
Base+88 0220	<u>NAME PART8 H</u>	1 st char	2 nd char	HF	4846
Base+88 0222	<u>NAME PART8 L</u>	3 rd char	4 th char		2020
Base+88 0224	<u>NAME PART9 H</u>	1 st char	2 nd char	HO	484F
Base+88 0226	<u>NAME PART9 L</u>	3 rd char	4 th char		2020
Base+88 0228	<u>NAME PART10 H</u>	1 st char	2 nd char	RC	5243
Base+88 022A	<u>NAME PART10 L</u>	3 rd char	4 th char	T	5420
Base+88 022C	<u>NAME PART11 H</u>	1 st char	2 nd char	GC	4743
Base+88 022E	<u>NAME PART11 L</u>	3 rd char	4 th char	T	5420
Base+88 0230	<u>NAME PART12 H</u>	1 st char	2 nd char	RP	5250
Base+88 0232	<u>NAME PART12 L</u>	3 rd char	4 th char	C	4320
Base+88 0234	<u>NAME PART13 H</u>	1 st char	2 nd char	DT	4454
Base+88 0236	<u>NAME PART13 L</u>	3 rd char	4 th char	0	3020
Base+88 0238	<u>NAME PART14 H</u>	1 st char	2 nd char	DT	4454
Base+88 023A	<u>NAME PART14 L</u>	3 rd char	4 th char	+	2B20
Base+88 023C	<u>NAME PART15 H</u>	1 st char	2 nd char	DT	4454
Base+88 023E	<u>NAME PART15 L</u>	3 rd char	4 th char	-	2D20
Base+88 0240	<u>NAME PART16 H</u>	1 st char	2 nd char	CS	4353
Base+88 0242	<u>NAME PART16 L</u>	3 rd char	4 th char	C+	432B
Base+88 0244	<u>NAME PART17 H</u>	1 st char	2 nd char	CS	4353
Base+88 0246	<u>NAME PART17 L</u>	3 rd char	4 th char	C-	432D
Base+88 0248	<u>NAME PART18 H</u>	1 st char	2 nd char	DT	4454
Base+88 024A	<u>NAME PART18 L</u>	3 rd char	4 th char	TF	5446
Base+88 024C	<u>NAME PART19 H</u>	1 st char	2 nd char	CS	4353
Base+88 024E	<u>NAME PART19 L</u>	3 rd char	4 th char	CT	4354
Base+88 0250	<u>NAME PART20 H</u>	1 st char	2 nd char	CA	4341
Base+88 0252	<u>NAME PART20 L</u>	3 rd char	4 th char	ST	5354
Base+88 0254	<u>NAME PART21 H</u>	1 st char	2 nd char	TO	544F
Base+88 0256	<u>NAME PART21 L</u>	3 rd char	4 th char	TM	544D
Base+88 0258	<u>NAME PART22 H</u>	1 st char	2 nd char	ZD	5A44
Base+88 025A	<u>NAME PART22 L</u>	3 rd char	4 th char	C	4320
Base+88 025C	<u>NAME PART23 H</u>	1 st char	2 nd char	<i>no</i>	6E6F
Base+88 025E	<u>NAME PART23 L</u>	3 rd char	4 th char	<i>ne</i>	6E65
Base+88 0260	<u>NAME PART24 H</u>	1 st char	2 nd char	TI	5449
Base+88 0262	<u>NAME PART24 L</u>	3 rd char	4 th char	BD	4244
Base+88 0264	<u>NAME PART25 H</u>	1 st char	2 nd char	TO	544F

Base+88 0266	NAME_PART25_L	3 rd char	4 th char	B	4220
Base+88 0268	NAME_PART26_H	1 st char	2 nd char	TE	5445
Base+88 026A	NAME_PART26_L	3 rd char	4 th char	C+	432B
Base+88 026C	NAME_PART27_H	1 st char	2 nd char	TE	5445
Base+88 026E	NAME_PART27_L	3 rd char	4 th char	C-	432D
Base+88 0270	NAME_PART28_H	1 st char	2 nd char	BP	4250
Base+88 0272	NAME_PART28_L	3 rd char	4 th char	IX	4958
Base+88 0274	NAME_PART29_H	1 st char	2 nd char	FP	4650
Base+88 0276	NAME_PART29_L	3 rd char	4 th char	IX	4958
Base+88 0278	NAME_PART30_H	1 st char	2 nd char	SE	5345
Base+88 027A	NAME_PART30_L	3 rd char	4 th char	+	2B20
Base+88 027C	NAME_PART31_H	1st char	2nd char	SE	5345
Base+88 027E	NAME_PART31_L	3rd char	4th char	-	2D20

Abbreviations:

EB+, EB-	Electromagnetic Calorimeter, barrel, positive/negative side
EE+, EE-	Electromagnetic Calorimeter, endcap, positive/negative side
GTst	Global Trigger
HBEa,b,c	Hadron Calorimeter, barrel and endcap as 3 groups
HF	Hadron Calorimeter, forward
HO	Hadron Calorimeter, outer barrel
RCT	Regional Calorimeter Trigger
GCT	Global Calorimeter Trigger
RPC	Resistive Plate Chambers
DT0	Drift Tube 0
DT+, DT-	Drift Tube, positive/ negative side
CSC+, CSC-	Cathode Strip Chambers, positive/ negative side
DTTF	Drift Tube Track Finder
CSCT	Cathode Strip Chamber Track Finder
CAST	Castor
TOTM	Totem
ZDC	Zero Degree Detector
None	free channel
TIBD	Tracker, inner
TOB	Tracker, outer, barrel
TEC+, TEC-	Tracker Endcap, positive/ negative side
BPIX	Barrel Pixel detector
FPIX	Forward Pixel detector
SE+, SE-	Preshower Endcap

6 VME chip

7 Other hardware chips

Channel Link Receiver DS90CR218A

Parallel outputs:

VOL: 0.3V: +2mA

VOH: 2.7...3.3V: -0.4mA

LVDS Receivers SN75LVDT386

Serial Inputs with internal 110 Ohm termination

Parallel outputs:

VOL: 0.5V: +8mA

VOH: 2.5V: -8mA

8 Check of Schematics against Pin-Layout

1.) ViewDraw: TCS Schematics save & check to generate new .WIR files

2.) Start Perlscript p:\GlobalTrigger\TCS\TCS90\TCS90.PL

Datafiles TCSMON.LST and TCS.LST will be generated.

3.) EXCEL

open TCSMON.LST , use „;“ as additional delimiter

sort by column A, column B

copy cells A1:C774

in file P:\GlobalTrigger\Tcs\Tcs9u\Doc\TSC_MON_CHIP.XLS on sheet

tcs_mon_schematics replace cells A2:C775

Checks will be done automatic within EXCEL file.

9 TEST Procedures

First do all tests for DAQ_PTC0 (=DAQ Partition Controller)

9.1 TCS Status Test

9.1.1 Setup for EVM in simulation mode

On the TCS GUI side:

1) assign DP0 and GT to PTC0

2) set simulation mode for DP0, GT and EVM

- 3) set status of DP0, GT and EVM to ready
- 4) start time slot generator
- 5) reset throttle logic
- 6) click button enable All PTCs
- 7) click button start RUN

After clicking start RUN the status of all partitions should be ready. If this is the case continue, otherwise stop here and find error first.

To change status of the simulated EVM use the partition status widget.

<i>EVM Status</i>	<i>PTC-IN</i>	<i>PTC-OUT</i>
Ready	Ready	Ready
Warning Overflow	Warning Overflow	Warning Overflow
Busy	Busy	Busy
Disconnected	Disconnected	Busy
Ready	Ready	Ready
Out Of Sync	Out Of Sync	Error ^{*)}
Ready	Ready	Ready
Error	Error	Error ^{*)}
Bad Code	Bad Code	Ready

^{*)} now you have to set EVM status to ready and stop RUN and start RUN again

9.1.2 Setup for EVM in real mode

On the GTFE side:

- 1) start gtfe.exe
- 2) register and execute sequence named daq_evm_ro_records_slot17_rw_regs.dat (should be found under
~/GT_INTERCONNECTION_TESTS/cern_afs_copy/GT/GTFE/HAL_sequence)
- 3) set item DAQ_CMD_REG and EVM_CMD_REG to 0x1b02
- 4) execute option 49) and 55) and res_bc_error
- 5) execute option 51) to set EVM status to ready

On the TCS GUI side:

- 1) assign DP0 and GT to PTC0
- 2) set simulation mode for DP0 and GT
- 3) set real mode for EVM
- 4) set status of DP0 and GT to ready
- 5) start time slot generator
- 6) reset throttle logic
- 7) click button enable All PTCs
- 8) click button start RUN

After clicking start RUN the status of all partitions should be ready. If this is the case continue, otherwise stop here and find error first.

9.2 STATUS TEST in GT crate

Check if the STATUS bits of the inserted GT/GMT board are combined correctly in the FDL board.

9.2.1 EVM Status

9.2.2 DAQ Status

In FDL disable all missing boards.

Set all other GT/GMT boards to 'READY'

Connect only the GT to PTC0 and set SIM MODE=0

Check if the GT-STATUS input on the TCS board (tcs9u_gui) is also 'ready'

Set one of the enabled boards to BUSY' → GT_STATUS=busy.

And check if triggers are inhibited when PTC0 is not in READY status.

9.2.3 SET BOARDS to READY

- **FDL: FDL: General Register (rw) 0x24C04 (32 bits)**

Bit 27 – 16: Set the 'disconnected' bits =1 for all missing boards

Bit 7,6,5: Set FDL to 'ready'

Bit 0: sel_sim: 0: spy mode (for Algo and TTrig bits)
 1: simulation mode

Bit 1: spy_mode 0: 1 orbit start with next bces
 1: run synchronous with each bces

Bit 2: sel_long 0: ROP sends 3 BX 1: ROP sends 5 BX

Bit 3: sel_warn_50 0: warning is generated at 75%
 1: warning is generated at 50% fifo status

Bit 4: use_BCRes_dly 0: Bypass bces delay circuit
 1: Use bces delay

Bit 5: disconnected 1: set FDL status to disconnected

Bit 6: busy 1: set FDL status to busy

Bit 7: ready 1: set FDL status to ready

Bit 8: en_backrecv 1: enables receivers for backplane signals

Bit 9: en_chan_e 1: enable channel link chip for GTFE-DAQ

Bit 10: en_chan_d 1: enable channel link chip for GTFE-EVM

Bit 11: en_freeze 1: enables freeze_mon command from TCS

Bit 15-12: not used

Status disable Bits:

Bit 16: disable status signals of slot 9 (PSBT)

Bit 17: disable status signals of slot 11 (GTL1)

Bit 18: disable status signals of slot 12 (GTL2)

- Bit 19: disable status signals of slot 13 (PSB1)
- Bit 20: disable status signals of slot 14 (PSB2)
- Bit 21: disable status signals of slot 15 (PSB3)
- Bit 22: disable status signals of slot 16 (TIM)
- Bit 23: disable status signals of slot 17 (GTFE)
- Bit 24: disable status signals of slot 18 (GMT)
- Bit 25: disable status signals of slot 19 (PSB4)
- Bit 26: disable status signals of slot 20 (PSB5)
- Bit 27: disable status signals of slot 21 (PSB6)

- **Set the inserted boards to 'READY'**

- **PSB: XX10 0046 (hex) ROP_SETUP write & read**

Bit 15: not used

Bit 14: 1= send TP6 signal to Panel →Signal of TESTPOINT 6 goes to Frontpanel

Bit 13: 1= send TP5 signal to Panel

Bit 12: 1= send TP4 signal to Panel

Bit 11: 1= send TP3 signal to Panel

Bit 10: 1= send TP2 signal to Panel

Bit 9: 1= send TP1 signal to Panel

Bit 8: 1= send TP0 signal to Panel

Bit 7 – 4 are not used

Bit 3:**en_robust** =0 (default)

=1 enable the Bgo commands as the TIM board sends via the ROBUST
 =0 the PSB uses the encoded command (L1Res, Bcres, L1A) signals sent by the TIM board.

Bit 2: **five_bx_event** =0 (default)

=1: A readout record contains data from 5 bunch crossings around the triggering bx (-2, -1, 0, +1, +2).

=0: A readout record contains data from 3 bunch crossings around the triggering bx (-1, 0, +1)

Bit 1 and bit 0: PSB_MODE

= 0 0 PSB is **DISCONNECTED** from readout (=default)

= 0 1 PSB is **BUSY** with other tasks and cannot receive any L1A for the time being. But the ROP, all counters and registers are correct to continue the data taking run.

= 1 0 PSB is **READY** and waits for the Bgo command 'RUN' to receive L1A and to send events to the GTFE board. For tests the 'RUN' command can also be simulated by a vme cmd pulse.

= 1 1 PSB sends **BAD CODE**...should never be set except for a test

- Set GTL to 'READY' and disable sigs from REC and COND chips

- Trig_ctrl_reg0 = X"9F"

- Trig_ctrl_reg1 = X"1F"

The status signals of the REC and COND chips are combined in the VME chip. A merged 4 bit- STATUS is sent to the FDL board to be combined with other boards. (status code 00=ok, 01=hardw_error, 10=out_of_sync, 11=FPGA not configured)

Trigger-control register 0: 0x1C00040

Bit7: VME_STATE_1 00=disconnected, 01=busy, 10=ready, 11=bad code

Bit6: VME_STATE_0

Bit5: ----

Bit4: DISABLE_ERROR_COND2

Bit3: DISABLE_ERROR_COND1

Bit2: DISABLE_ERROR_REC3

Bit1: DISABLE_ERROR_REC2

Bit0: DISABLE_ERROR_REC1

Trigger-control register 1: 0x1C00042

Bit7-5: not used

Bit4: DISABLE_OUT_SYNC_COND2

Bit3: DISABLE_OUT_SYNC_COND1

Bit2: DISABLE_OUT_SYNC_REC3

Bit1: DISABLE_OUT_SYNC_REC2

Bit0: DISABLE_OUT_SYNC_REC1

- Set TIM to 'READY' : bit15 =1, other bits as defined by selected running mode.

TIM_COMMAND_REG 0x1 003A

Bit 15, 14: TIM STATUS for TCS

00 = TIM DISCONNECTED

10 = TIM_READY

01 = TIM_BUSY

11 = TIM DISCONNECTED

The programmed status is included into the TIM status code sent via the FDL to the TCS board. The TIM status can be overruled by error flags.

RO_CMD_REG 0x1 003C

Set bit8 and bit7 =0 to disable checks.

Bit 8: EN_BC_CHECK =0

Bit 7: EN_TTC_CHECK=0

- Set GMT to 'READY':

Method will be added later. In the meantime disable STATUS code from the

GMT board.

- Set GTFE to READY: ignore all errors and set status to ready

DAQ_IGNORE_ERRORS = X"00FF"

EVM_IGNORE_ERRORS = X"00FF"

DAQ_CMD_REG: bit1&0 = 10=ready, 01=busy, 00=11=disconnected

EVM_CMD_REG: bit1&0 = 10=ready, 01=busy, 00=11=disconnected

BB10 0042 DAQ_IGNORE_ERRORS w/r // bits 15_0

If one of the ignore bits is set then the respective flags of active Channel Links are not transmitted to the TCS Trigger Control board (via the FDL board).

If any of the not ignored errors is active then the ERR_LED is illuminated.

Bit15-6 not used

Bit7 := 1 ignore SLINK DOWN FLAG → error

Bit6 := 1 ignore BC_ERROR → error

Bit5 := 1 ignore FIFO WARNING FLAGS → warning

Bit4 := not used

Bit3 := 1 ignore DATA_LOST FLAGS → out_of_sync

Bit2 := 1 ignore BAD_HEADER FLAGS → error

Bit1 := 1 ignore CHANNEL_LINK_TOO_LATE FLAGS → error

Bit0 := 1 ignore FIFO FULL FLAGS → out_of_sync

9.3 TEST POINTS for internal TCS signals

TP-Name	MEZZ957	MEZZ896	FPGA-Pin	Net	Position
TEST1_TCS	AK15/ CON1-662	AE14/ CON4-60	io	---	TCS9U
TEST2_TCS	AK14/ CON1-660	AE13/ CON4-58	io	---	TCS9U
GCLK2P	AL15/ CON1-672	AE15/ CON4-70	GCLK2P	CLK_FBIN_TCS	MEZZ957
GCLK0P	AJ15/ CON1-669	AH15/ CON4-67	GCLK0P	CLK_TO_TCS	MEZZ957
GCLK2S	C15/ CON1-496	C14/ CON3-66	GCLK2S	VA_I_10	MEZZ957 MEZZ896
GCLK0S	H15/ CON1-499	F14/ CON3-69	GCLK0S	---	MEZZ957 MEZZ896
GCLK7P	E17/	H16/	GCLK7P	FASTIN_CYC	MEZZ896

	CON1-507	CON3-77			
GCLK5P	A18/ CON1-517	C17/ CON3-87	GCLK5P	TIMTCS10	MEZZ896
B30 (nord-east)	E31/ CON1-227	B30/ CON2-12	io	NEN_EVMLINK	MEZZ896
AK29 (south-east)	AK29/ CON1-749	AK29/ CON4-147	io	TTCVI_9_2 = 'BGO_2' bit	MEZZ896

CON3 =NORTH, CON4=SOUTH side

All test points without connected net can be used to monitor chip-internal signals.

(depending from design version)

TEST1_TCS: shows enabled signals (programmable)

TEST2_TCS: shows enabled signals (programmable)

GCLK0S: CLK

Remark: GCLK0S considered since file tcs_chip_16june04.ucf.

9.4 TEST POINTS for internal TCSM signals

All test points without connected net can be used to monitor chip-internal signals.

TP-Name	MEZZ896	FPGA-Pin	Net	Position
TEST1_TCSM	AC17/ CON4-79	io	---	TCS9U
TEST2_TCSM	AC16/ CON4-77	io	---	TCS9U
GCLK2S	C14/ CON3-66	GCLK2S	---	MEZZ896
GCLK0S	F14/ CON3-69	GCLK0S	---	MEZZ896
GCLK7P	H16/ CON3-77	GCLK7P	---	MEZZ896
GCLK5P	C17/ CON3-87	GCLK5P	---	MEZZ896
B30 (nord-east)	B30/ CON2-12	io	BCRES_FROM_TIM_M	MEZZ896
AK29 (south-east)	AK29/ CON4-147	io	---	MEZZ896

CON3 =NORTH, CON4=SOUTH side

Used as Test points for internal signals: (depending from design version)

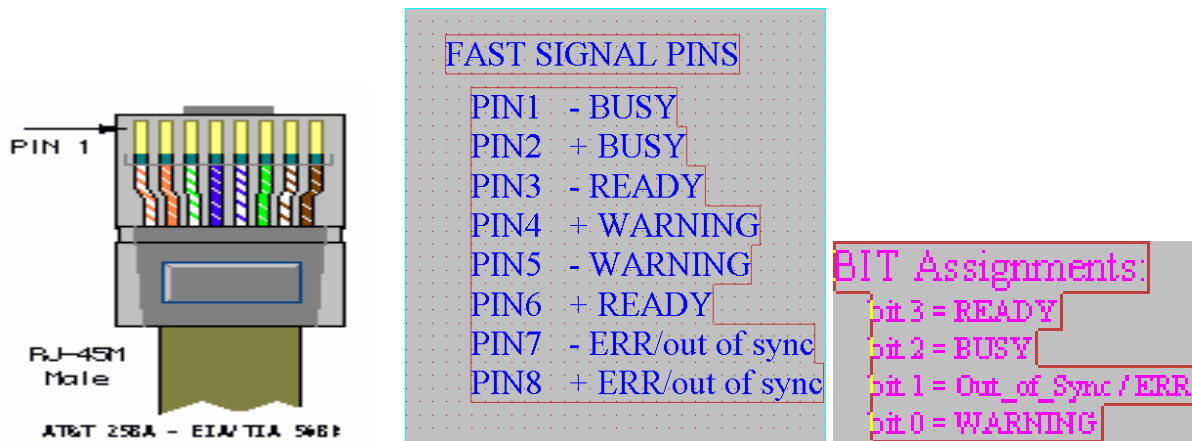
TEST1_TCSM: CLK

TEST2_TCSM: FAST81 = interlaced 1 and 0 looking like a 40 MHz clock

GCLK2S: XXXX
 GCLK0S: XXXX
 GCLK7P: XXXX
 GCLK5P: XXXX
 AK29: XXXX

9.5 JTAG TEST board for external STATUS bits

The JTAG TEST board contains 4 DS92LV090A transceivers for 32 LVDS signals, which are connected to a 68p SCSI connector. 8 Ethernet cables will be connected to the SCSI connector according to table below.



SIGNAL Name	STATUS bit	Bit#	SCSI pin	Ether pin	Ether cable
- READY	3	N31	64	3	P7
+READY	3	31	63	6	P7
- BUSY	2	N30	62	1	P7
+BUSY	2	30	61	2	P7
- Out of SYNC	1	N29	60	7	P7
+ Out of SYNC	1	29	59	8	P7
- WARNING	0	N28	58	5	P7
+ WARNING	0	28	57	4	P7
		N27	56	3	P6
		27	55	6	P6
		N26	54	1	P6
		26	53	2	P6
		N25	52	7	P6

		25	51	8	P6	
		N24	50	5	P6	
		24	49	4	P6	
		N23	48	3	P5	
		23	47	6	P5	
		N22	46	1	P5	
		22	45	2	P5	
		N21	44	7	P5	
		21	43	8	P5	
		N20	42	5	P5	
		20	41	4	P5	
		N19	40	3	P4	
		19	39	6	P4	
		N18	38	1	P4	
		18	37	2	P4	
		N17	36	7	P4	
		17	35	8	P4	
		N16	34	5	P4	
		16	33	4	P4	
		N15	32	3	P3	
		15	31	6	P3	
		N14	30	1	P3	
		14	29	2	P3	
		N13	28	7	P3	
		13	27	8	P3	
		N12	26	5	P3	
		12	25	4	P3	
		N11	24	3	P2	
		11	23	6	P2	
		N10	22	1	P2	
		10	21	2	P2	
		N9	20	7	P2	
		9	19	8	P2	
		N8	18	5	P2	
		8	17	4	P2	
		N7	16	3	P1	
		7	15	6	P1	
		N6	14	1	P1	

		6	13	2	P1	
		N5	12	7	P1	
		5	11	8	P1	
		N4	10	5	P1	
		4	9	4	P1	
		N3	8	3	P0	
		3	7	6	P0	
		N2	6	1	P0	
		2	5	2	P0	
		N1	4	7	P0	
		1	3	8	P0	
		N0	2	5	P0	
		0	1	4	P0	

Altera chip contains two 16 bit transmit registers and sends 32 bits to the 68p SCSI connector.

1	1	1	1	1	1	9	8	7	6	5	4	3	2	1	0	Register Name
5	4	3	2	1	0											
Status-part3				Status-part2				Status-part1				Status-part0				STATUS_PART3_0
3	3	2	2	2	2	2	2	2	2	2	2	1	1	1	1	Register Name
1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	
Status-part7				Status-part6				Status-part5				Status-part4				STATUS_PART7_4

Status codes: Disconnected=0, Warning=1, Out_of_Sync=2, Busy=4, Ready=8, Error=C

Example:

8	8	8	8	STATUS_PART7_4
---	---	---	---	----------------

Set the DS92LV090A transceiver into transmit mode:

Set in Altera chip DE_H =H =driver enable and /RE_H =H =Receiver disable.

9.6 Internal STATUS bits from TCSM chip

28 x 4 bits exist to insert STATUS bits for Tests into the System.

The inputs from Emulators cannot be simulated by internal Signals in TCSM

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Register Name
----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---	---------------

Status-part3	Status-part2	Status-part1	Status-part0	SIM STATUS P23 20
Status-part3	Status-part2	Status-part1	Status-part0	SIM STATUS P19 16
Status-part3	Status-part2	Status-part1	Status-part0	SIM STATUS P15 12
Status-part3	Status-part2	Status-part1	Status-part0	SIM STATUS P11 8
Status-part3	Status-part2	Status-part1	Status-part0	SIM STATUS P7 4
Status-part3	Status-part2	Status-part1	Status-part0	SIM STATUS P3 0

Register Name	Default for TESTs
SIM STATUS P23 20	8888 =READY
SIM STATUS P19 16	8888 =READY
SIM STATUS P15 12	8888 =READY
SIM STATUS P11 8	8888 =READY
SIM STATUS P7 4	8888 =READY
SIM STATUS P3 0	8888 =READY

9.7 Test Program with external status bits

- Configure Altera chip on JTAG board
- Set Altera and DS92LV090A into transmit mode
- load STATUS_PART3_0 and STATUS_PART7_4 with new state
- read status registers of PTC (Partition Controller) in TCS chip
- Check by Oscilloscope: L1A, BGO codes
- *Manuell: both functions will be activated by pushing a button/keyboard*
- **Automatic Procedure:**

Setup a list of states and apply the next pattern every xxx msec

9.8 Test Program with internal status bits

- Configure TCSM
- Set TCSM inputs to internal SIMULATION STATUS bits
- load new states into SIM_STATUS_PART....registers
- read status registers of PTC in TCS chip
- Check by Oscilloscope: L1A, BGO codes
- *both functions will be activated by pushing a button/keyboard*
- **Automatic Procedure:**

Setup a list of states and apply the next pattern every xxx msec

(If a memory in TCSM has been implemented)

9.9 Status Signal Decoder

Change input of a partition, check if DAQ-partition changes it's status accordingly by reading

status register of Partition Controller (PTC)

9.10 Periodic Generator

Set DAQ0_BGO_PERIOD_L

Set DAQ0_BGO_PERIOD_S

Bit3=EN bit2,1,0 =frequency

And monitor if BGO command for programmed BGO command appears periodically as programmed.

Osci

9.11 Random Trigger

Load default setup for registers.

Set values for Random Trigger.

Assign all Partitions to DAQ_PTC0

Set all partitions =READY=8...

Monitor

9.12 Partition Assignment

DEFAULT ASSIGNMENT for TEST: Connect all Part's to DAQ_PTC0

ADDR	D15 - 12		D11 - 8		D7 - 4		D3 - 0	
	EN	DAQ_NR[2:0]	EN	DAQ_NR[2:0]	EN	DAQ_NR[2:0]	EN	DAQ_NR[2:0]
+20		8*		8		8		8
+22		8		8		8		8
+24		8		8		8		8
+26		8		8		8		8
+28		8		8		8		8
+2A		8		8		8		8
+2C		8		8		8		8
+2E		8		8		8		8
+30		0+		0+		0*		8**

* for LUM, ** for GT + disables/enables IO chips on TCS board

Bit3=EN bit2,1,0 =DAQ_nr

Examples: 8=1000=select DAQ_PTC0, 9=1001=select DAQ_PTC1...

Set internal STATUS registers bits in TCSM and switch them to TCS

Load different assignments for DAQ-PTCs,

change state of newly connected partitions and

check behavior of DAQ_PTC with new partitions and read also DAQ0_P_STATUS register that shows the state of the DAQ_PTC because of the STATE inputs from TCSM

If STATUS → out_of SYNC → BGO cmd=L1RESET is sent to TTCci
 If STATUS → WARNING → TCS/PTC status register: WARNING bit is set
 etc

9.13 State Machine

Use default setup of PTC.

Run a list of input states and check behavior of PTC

9.14 Calibration Triggers and Throttle Logic

Use default setup register values for calibration triggers.

Load BC-Tables for high calibration rates.

Set input states to READY.

Program should allow changing the calibration registers only.

Set Periodic Generator values to run calibration every orbit.

Set CALIBR_PERIOD=8=1000 in register DAQ0_BGO_PERIOD_S =0800 hex

Run with different frequencies and check dead-time counters.

9.15 Default values for DAQ_PTC0 registers

- Orbit length → check period of BCRES
-

	test default	random trigger	calibration trigger	physics trigger	
00400000	0200	0201	0200		DAQ0_CMD_REG
00400002	0FFF	0FFF	0FFF		DAQ0_RANDOM_FREQ
00400004	A5B6	A5B6	A5B6		DAQ0_RANDOM_ START_VALUE
00400006	4321	4321	4321		DAQ0_TRIG_TYPE_A
00400008	0006	0006	0006		DAQ0_TRIG_TYPE_B
0040000A	0000	0000	0000		DAQ0_BGO_PERIOD_L
0040000C	0000	0000	0800		DAQ0_BGO_PERIOD_S
0040000E	0101	0101	0101		DAQ0_ACTIVE_TIME

9.16 Default values for Common Registers

// Write and read access

addr	Default (hex)	comments
48 0000	00F0=240dec	Rule4 time
48 0002	0004	#trigs norm

48 0004	0002	#trigs warn
48 0006	0000	Rule3 time h
48 0008	00FE	Rule3 time l : 100bx=71+15(F)+14(E)=
48 000A	0301	Rule3: trigs# norm // trigs# warn
48 000C	00F6	Rule2 time: 25bx= 4+15(F)+6
48 000E	2102	Rule2: trigs# norm //trigs# warn; // Rule1_min_dt
48 0010	00FF=255dec	Daq0 timeslot: only daq0 is running
48 0012	0	Daq1 timeslot
48 0014	0	Daq2 timeslot
48 0016	0	Daq3 timeslot
48 0018	0	Daq4 timeslot
48 001A	0	Daq5 timeslot
48 001C	0	Daq6 timeslot
48 001E	0	Daq7 timeslot
48 0020	8888	Assign part's 3,2,1,0 all to DAQ_PTC0
48 0022	8888	Assign 7 6 5 4
48 0024	8888	Assign 11 10 9 8
48 0026	8888	Assign 15 14 13 12
48 0028	8888	Assign 19 18 17 16
48 002A	8888	Assign 23 22 21 20
48 002C	8888	Assign 27 26 25 24
48 002E	8888	Assign 31 30 29 28
48 0030	0088	Disable bits //Assign LUM, GT
48 0032	8DEB	ORBIT_LENGTH-1: bit15=en internal BCRES
48 0034	0DEB	GAP_LIMITH: 3563
48 0036	0D74	GAP_LIMITL: 3444 gap begin
48 0038	0800	SETTLE TIME : max orbits // min orbits
48 003A	0800	RECOVER TIME: max orbits // min orbits
48 003C	1000	MON_CNTR_PERIOD= 4096 orbits =0.366 sec
48 003E	xxxx	Free address

9.17 BC-Table 0 and 1 with default content

Insert les then 5 calibration cycles per orbit. (4→<50 kHz)

BGO for BCRES at BC=0 or 3563..

Insert all other BGO commands at different BC positions in first half of table.

Bit positions of BC TABLE1.

Bit#	BC_TABLE_0	comment	Bit#	BC_TABLE_1	comment
15	PRIV_BGO_3		15	0	
14	PRIV_BGO_2		14	0	
13	PRIV_BGO_1		13	0	
12	PRIV_BGO_0		12	0	
11	PRIV_BGO_STROBE		11	0	
10	STOP		10	0	
9	START		9	0	
8	RESET_ORBIT		8	0	
7	RESET_EVNR		7	0	
6	HARD_RESET		6	0	
5	RESYNC		5	0	
4	PRIV_ORBIT		4	0	
3	PRIV_GAP		3	TRACE_EN	
2	PRIV_TRIG	Trigger pulse	2	TRACE_ACTIVE	
1	BCRES		1	TE (test en)	
0	VALID_BC		0	CALTRIG	Trigger pulse

9.18 BC-Table 0 and 1 for high calibration rate

Insert 9 or more calibration cycles per orbit. (9→100 kHz)