REC_CHIP_Addresses

To be done:

Change chipID to 1_704x,
check 2 status bits to VME chip: out_of_sync=1 after power-up, instead of XOR
use bc-error = bcr_tim */bcr_intern

Version 0000 1005 23.8.2006 : Orbit counter stored by L1A
Version 0000 1004 18.8.2006 : L1A stops spying, new status regs,
Version 0000 1003 24.3.2006 : L1A starts spying,

1 Connection of Calorimeter Channels and Muons

REC1: MU12, MU34, CA1, CA2
REC2: CA3, 4, 5, 6
REC3: CA7, 8, 9, 10
CHIP IDENTIFIERS: REC1 0001 7141 REC1 0001 7142 REC3 0001 7143
VERSION NUMBERS: x0000 00nn = Test versions
x0001 00nn = Run versions
Input data can be spied or simulated usign DP-memories:

MUON CONNECTIONS:
MUON12(29:0) simulated by 2 memories in REC1:
MU12_L for bits 15:0
MU12_H for bits 29:16 with b31, b30=0
MUON34(29:0) simulated by 2 memories in REC1:
MU34_L for bits 15:0
MU34_H for bits 29:16 with b31, b30=0

CONNECTION Table for Calorimeter trigger data							
PSB Slot nr	**PSB conn**	**Pair in cable**	**PSB chan**	**Backplane signal**	**SIM/SPY Memory (15:0)**	**Signal to COND (15:0)**	**Interlacing of objects**
13	IN6-7	1	7	CA1(31:16)	REC1: CA1_24	CA1_24	2 then 4
13	IN6-7	0	6	CA1(15:0)	REC1: CA1_13	CA1_13	1 then 3
13	IN4-5	1	5	CA2(31:16)	REC1: CA2_24	CA2_24	2 then 4
13	IN4-5	0	4	CA2(15:0)	REC1: CA2_13	CA2_13	1 then 3
13	IN2-3	1	3	CA3(31:16)	REC2: CA3_24	CA3_24	2 then 4
13	IN2-3	0	2	CA3(15:0)	REC2: CA3_13	CA3_13	1 then 3
13	IN0-1	1	1	CA4(31:16)	REC2: CA4_24	CA4_24	2 then 4
13	IN0-1	0	0	CA4(15:0)	REC2: CA4_13	CA4_13	1 then 3
14	IN6-7	1	7	CA5(31:16)	REC2: CA5_24	CA5_24	2 then 4
14	IN6-7	0	6	CA5(15:0)	REC2: CA5_13	CA5_13	1 then 3
14	IN4-5	1	5	CA6(31:16)	REC2: CA6_24	CA6_24	2 then 4
14	IN4-5	0	4	CA6(15:0)	REC2: CA6_13	CA6_13	1 then 3
14	IN2-3	1	3	CA7(31:16)	REC3: CA7_24	CA7_24	2 then 4
14	IN2-3	0	2	CA7(15:0)	REC3: CA7_13	CA7_13	1 then 3
14	IN0-1	1	1	CA8(31:16)	REC3: CA8_24	CA8_24	2 then 4
14	IN0-1	0	0	CA8(15:0)	REC3: CA8_13	CA8_13	1 then 3
15	IN2-3	1	3	CA9(31:16)	REC3: CA9_24	CA9_24	2 then 4
15	IN2-3	0	2	CA9(15:0)	REC3: CA9_13	CA9_13	1 then 3
15	IN0-1	1	1	CA10(31:16)	REC3:CA10_24	CA10_24	2 then 4
15	IN0-1	0	0	CA10(15:0)	REC3:CA10_13	CA10_13	1 then 3

PSB1 => slot 13, PSB2 => slot 14, PSB3 => slot 15

2 Address overview

A REC chip contains 8 memories, two for each Calorimeter group. The muons in the REC1 chip need 4 memories. Because of the limited space in the XC2V2000 chip the memory size has been reduced to 4k x 16 bit containing data of about 50% of a LHC orbit. But the memory base addresses are defined for 8k size.
4k x16 bits ➔ 4 RAM_blocks ➔ 4 x 8 memories ➔ 32 RAMB < 56 (XC2V2000)
[8k x16 bits ➔ 8 RAM_blocks ➔ 8 x 8 memories ➔ 64 RAMB > 56 (XC2V2000)]
Remark: The 30 bit muon word goes into 2 memories; bits 15-0 into MU12(34)_L and bits 29-16 into MU12(34)_H. *The bits MU12(34)_H(31:30) ="00" and are not used.*

ADDRESS OVERVIEW										
Chip	**24**	**23-20**	**19-16**	**15**	**14**	**13**	**12**	**11-8**	**7 – 1(0)**	**Source**
REC1	0	1000	0000	0	0	0	0	0000	Registers	
REC1	0	1000	0010	0	0	Sim_Spy_mem MU12_L (4kx16)				*GMT*
REC1	0	1000	0010	0	1	Sim_Spy_mem MU12_H (4kx16)				*GMT*
REC1	0	1000	0010	1	0	Sim_Spy_mem MU34_L (4kx16)				*GMT*
REC1	0	1000	0010	1	1	Sim_Spy_mem MU34_H (4kx16)				*GMT*
REC1	0	1000	0011	0	0	Sim_Spy_mem CA1_13 (4kx16)				*PSB1*
REC1	0	1000	0011	0	1	Sim_Spy_mem CA1_24 (4kx16)				*PSB1*
REC1	0	1000	0011	1	0	Sim_Spy_mem CA2_13 (4kx16)				*PSB1*
REC1	0	1000	0011	1	1	Sim_Spy_mem CA2_24 (4kx16)				*PSB1*
REC2	0	1100	0000	0	0	0	0	0000	Registers	
REC2	0	1100	0010	0	0	Sim_Spy_mem CA3_13 (4kx16)				*PSB1*
REC2	0	1100	0010	0	1	Sim_Spy_mem CA3_24 (4kx16)				*PSB1*
REC2	0	1100	0010	1	0	Sim_Spy_mem CA4_13 (4kx16)				*PSB1*
REC2	0	1100	0010	1	1	Sim_Spy_mem CA4_24 (4kx16)				*PSB1*
REC2	0	1100	0011	0	0	Sim_Spy_mem CA5_13 (4kx16)				*PSB2*
REC2	0	1100	0011	0	1	Sim_Spy_mem CA5_24 (4kx16)				*PSB2*
REC2	0	1100	0011	1	0	Sim_Spy_mem CA6_13 (4kx16)				*PSB2*
REC2	0	1100	0011	1	1	Sim_Spy_mem CA6_24 (4kx16)				*PSB2*
REC3	1	0000	0000	0	0	0	0	0000	Registers	
REC3	1	0000	0010	0	0	Sim_Spy_mem CA7_13 (4kx16)				*PSB2*
REC3	1	0000	0010	0	1	Sim_Spy_mem CA7_24 (4kx16)				*PSB2*
REC3	1	0000	0010	1	0	Sim_Spy_mem CA8_13 (4kx16)				*PSB2*
REC3	1	0000	0010	1	1	Sim_Spy_mem CA8_24 (4kx16)				*PSB2*
REC3	1	0000	0011	0	0	Sim_Spy_mem CA9_13 (4kx16)				*PSB3*
REC3	1	0000	0011	0	1	Sim_Spy_mem CA9_24 (4kx16)				*PSB3*
REC3	1	0000	0011	1	0	Sim_Spy_mem CA10_13 (4kx16)				*PSB3*
REC3	1	0000	0011	1	1	Sim_Spy_mem CA10_24 (4kx16)				*PSB3*
All REC	1	0100	0000	0000				0000	7 write only addresses	

3 Common Commands for REC1, 2, 3

RUN/STOP registers see also description of VME_CHIP_GTL.pdf
The commands go concurrently to all REC and COND chips to send and receive test data.

The test data shall be loaded first into the SIM/ SPY memories.
If no test data are sent the hardware forwards the GMT resp. PSB data from the backplane to the COND chips.

0140 00F0	run_v	w/	send SIM data continuously
0140 00F2	run_v_until_end	w/	send SIM data until end of memory
0140 00F4	run_sync	w/	send SIM data continuously and start with BCRES
0140 00F6	run_sync_until_end	w/	send SIM memory 1x and start with BCRES (1 orbit)
0140 00F8	stop_v	w/	stop transfer of SIM data immediately
0140 00FA	stop_sync	w/	stop transfer of SIM data at next BCRES
0140 00FC	stop_spying_with_next_L1A	w/	the next L1A will stop spying; before send 'run_sync' and set sim_modes=0

	4k(8k) > orbit	1…2k < orbit ALGO, REC mem's	Remark
run_v	Start immediately Run continuously		Test with Oscilloscope
run_v_until_end	Start immediately Stop at End of Mem	Start immediately Stop at End of Mem	Test without BCRes
run_sync	Start with BCRes Run continuously		Test with Oscilloscope
run_sync_until_end	**Start with BCRes Stop at End of Orbit**	**Start with BCRes Stop at End of Memory**	**Testprogram**
stop_v	Stop immediately		Test with Oscilloscope
stop_sync	Stop at End of Orbit	Stop at End of Memory	Test with Oscilloscope
stop_spying_with next_L1A	L1A stops immediatley		Check SPY STATUS word

4 REC1 chip

4.1 REC1 Sim Spy Memories

0082 0000	SIM_SPY_ MU12_L	w/r
0082 4000	SIM_SPY_ MU12_H	w/r
0082 8000	SIM_SPY_ MU34_L	w/r
0082 C000	SIM_SPY_ MU34_H	w/r
0083 0000	SIM_SPY_ CA1_13	w/r
0083 4000	SIM_SPY_ CA1_24	w/r
0083 8000	SIM_SPY_ CA2_13	w/r
0083 C000	SIM_SPY_ CA2_24	w/r

4.2 Register overview

0080 0000	VERSION_NR_L	-/r	// 0001 ..first version
0080 0002	VERSION_NR_H	-/r	// 0000=test versions, 0001= run versions
0080 0004	CHIP_IDL	-/r	// 7142

0080 0006	CHIP_IDH	-/r	// 0001
0080 0008	REC_STATUS	-/r	
0080 000A	BC_ERRORS	-/r	
0080 000C	--	-/r	
0080 000E	--	-/r	
0080 0010	BCRES_DELAY	w/r	
0080 0012	MAX_BC_NUMBER	w/r	
0080 0014	TESTMASK0	w/r	// tespoint 0 bits
0080 0016	TESTMASK1	w/r	// tespoint 1 bits
0080 0018	TESTMASK2	w/r	// tespoint 2 bits
0080 001A	TESTMASK3	w/r	// tespoint 3 bits
0080 001C	CMD_REG	w/r	
0080 001E	CMD_PULSE	w/-	
0080 0020	L1A_SPY_STATUS0	-/r	
0080 0022	L1A_SPY_STATUS1	-/r	
0080 0024	L1A_SPY_STATUS2	-/r	
0080 0026	L1A_SPY_STATUS3	-/r	
0080 0028	ORBIT_NR_L	-/r	
0080 002A	ORBIT_NR_H	-/r	
0080 00F0	run_v	w/	
0080 00F2	run_v_until_end	w/	
0080 00F4	run_sync	w/	
0080 00F6	run_sync_until_end	w/	
0080 00F8	stop_v	w/	
0080 00FA	stop_sync	w/	

5 REC2 chip

5.1 REC2 Sim Spy Memories

00C2 0000	SIM_SPY_ CA3_13	w/r	
00C2 4000	SIM_SPY_ CA3_24	w/r	
00C2 8000	SIM_SPY_ CA4_13	w/r	
00C2 C000	SIM_SPY_ CA4_24	w/r	
00C3 0000	SIM_SPY_ CA5_13	w/r	
00C3 4000	SIM_SPY_ CA5_24	w/r	
00C3 8000	SIM_SPY_ CA6_13	w/r	
00C3 C000	SIM_SPY_ CA6_24	w/r	

5.2 Register overview

00C0 0000	VERSION_NR_L	-/r	// 0001 ..first version
00C0 0002	VERSION_NR_H	-/r	// 0000=test versions, 0001= run versions
00C0 0004	CHIP_IDL	-/r	// 7142
00C0 0006	CHIP_IDH	-/r	// 0001
00C0 0008	REC_STATUS	-/r	
00C0 000A	BC_ERRORS	-/r	
00C0 000C	--	-/r	
00C0 000E	--	-/r	
00C0 0010	BCRES_DELAY	w/r	
00C0 0012	MAX_BC_NUMBER	w/r	
00C0 0014	TESTMASK0	w/r	// tespoint 0 bits

00C0 0016	TESTMASK1	w/r	// tespoint 1 bits
00C0 0018	TESTMASK2	w/r	// tespoint 2 bits
00C0 001A	TESTMASK3	w/r	// tespoint 3 bits
00C0 001C	CMD_REG	w/r	
00C0 001E	CMD_PULSE	w/-	
00C0 0020	L1A_SPY_STATUS0	-/r	
00C0 0022	L1A_SPY_STATUS1	-/r	
00C0 0024	L1A_SPY_STATUS2	-/r	
00C0 0026	L1A_SPY_STATUS3	-/r	
00C0 0028	ORBIT_NR_L	-/r	
00C0 002A	ORBIT_NR_H	-/r	

00C0 00F0	run_v	w/
00C0 00F2	run_v_until_end	w/
00C0 00F4	run_sync	w/
00C0 00F6	run_sync_until_end	w/
00C0 00F8	stop_v	w/
00C0 00FA	stop_sync	w/

6 REC3 chip

6.1 REC3 Sim Spy Memories

0102 0000	SIM_SPY_ CA7_13	w/r
0102 4000	SIM_SPY_ CA7_24	w/r
0102 8000	SIM_SPY_ CA8_13	w/r
0102 C000	SIM_SPY_ CA8_24	w/r
0103 0000	SIM_SPY_ CA9_13	w/r
0103 4000	SIM_SPY_ CA9_24	w/r
0103 8000	SIM_SPY_ CA10_13	w/r
0103 C000	SIM_SPY_ CA10_14	w/r

6.2 Register overview

0100 0000	VERSION_NR_L	-/r	// 0001 ..first version
0100 0002	VERSION_NR_H	-/r	// 0000=test versions, 0001= run versions
0100 0004	CHIP_IDL	-/r	// 7142
0100 0006	CHIP_IDH	-/r	// 0001
0100 0008	REC_STATUS	-/r	
0100 000A	BC_ERRORS	-/r	
0100 000C	--	-/r	
0100 000E	--	-/r	
0100 0010	BCRES_DELAY	w/r	
0100 0012	MAX_BC_NUMBER	w/r	
0100 0014	TESTMASK0	w/r	// tespoint 0 bits
0100 0016	TESTMASK1	w/r	// tespoint 1 bits
0100 0018	TESTMASK2	w/r	// tespoint 2 bits
0100 001A	TESTMASK3	w/r	// tespoint 3 bits
0100 001C	CMD_REG	w/r	
0100 001E	CMD_PULSE	w/-	
0100 0020	L1A_SPY_STATUS0	-/r	

0100 0022	L1A_SPY_STATUS1	-/r
0100 0024	L1A_SPY_STATUS2	-/r
0100 0026	L1A_SPY_STATUS3	-/r
0100 0028	ORBIT_NR_L	-/r
0100 002A	ORBIT_NR_H	-/r

0100 00F0	run_v	w/
0100 00F2	run_v_until_end	w/
0100 00F4	run_sync	w/
0100 00F6	run_sync_until_end	w/
0100 00F8	stop_v	w/
0100 00FA	stop_sync	w/

7 REC1, REC2, REC3 Register description

****** Addresses: Replace xxx by 008 for REC1, 00C for REC2 and 010 for REC3.***

7.1 VERSION_NR

xxx0 0000	VERSION_NR_L	-/r	// 0001 ..first version
xxx0 0002	VERSION_NR_H	-/r	// 0000=test versions, 0001= run versions

Defined by firmware in REC chip.

7.2 CHIP_ID

xxx0 0004	CHIP_IDL	-/r	// 7142
xxx0 0006	CHIP_IDH	-/r	// 0001

Defined by firmware in REC chip.

7.3 REC_STATUS register

xxx0 0008 REC_STATUS -/r

Read only register.

Bit 15 – 8 : 0000 0000

Bit 7: BC-error

Bit 6: 0

Bit 5 and bit 4: REC_STATUS(1:0)

~~-- 00 REC chip is DISCONNECTED … default status after power up~~

~~-- 01 REC chip is BUSY … during transfer of test data~~

~~-- 10 REC chip is READY … when CMD_REG(4) is set =1~~

~~-- 11 REC chip is BAD CODE … error status after enabled bc_error~~

7.Sept06: A.T. implemented in Version1005

Code	REC chip Status	Forward to TCS as:
00	REC_chip is DISCONNECTED */after power up*	
01	BUSY *// during transfer of test data*	
10	READY *// when CMD_REG(4)=1*	
11	ERROR *// enabled bc_error or FPGA is not configured (Pull-up resistors provide '11'.)*	

7.Sept06: A.T. to be implemented in next Version1006

Code	REC chip Status (To be done)	Forward to TCS as:
00	All ok, REC_chip is READY	ready
01	BUSY */ during transfer of test data*	busy

10	OUT_OF_SYNC: BC-cntr error, ..	out_of_sync
11	ERROR *// hardware error or FPGA is not configured (Pull-up resistors provide '11'.)*	error

bit 3: simulating(3) …channel 3 sending simulation data from REC memory
bit 2: simulating(2) …channel 2 sending simulation data from REC memory
bit 1: simulating(1) …channel 1 sending simulation data from REC memory
bit 0: simulating(0) …channel 0 sending simulation data from REC memory

7.4 BC_ERRORS

xxx0 000A BC_ERRORS -/r
BC_ERRORS shows the number of errors since the last readout access. The error counter is incremented by every disagreement between the internal BCRes and the BCRes signal sent by the TIM board.
A read access first moves the actual counter content into a readout register. Then it clears the counter and puts the saved counter content onto the VME data bus.

7.5 BCRES Delay

xxx0 0010 BCRES_DELAY **write & read**
--

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
max value = 48
Default value after power-up: =X"0000"

7.6 MAX_BC_NUMBER

xxx0 0012 MAX_BC_NUMBER w/r //=orbit length -1
Default value = 3563 dec = 0DEB hex has to be set by software before doing any tests, also after power up. Will be set by hardware after Power-up in next firmware release.
The number is used by a comparator to generate an internal bunch counter reset signal that is used when no common BCRES arrives from the back-plane.
If it does not agree with external BCRES signal from the TIM board then a BC_ERROR flag will be set. See also VHDL file: P:\GlobalTrigger\GTL\gtl9u\rec_chips\vhdl_files\V10001\time_run_control.behavioral.vhd

7.7 TESTMASK0,1,2,3

xxx0 0014	TESTMASK0	w/r	// tespoint 0 bits
xxx0 0016	TESTMASK1	w/r	// tespoint 1 bits
xxx0 0018	TESTMASK2	w/r	// tespoint 2 bits
xxx0 001A	TESTMASK3	w/r	// tespoint 3 bits

Mask bit =1 … switches internal signal to test point.
Mask bit =0 … inhibits internal signal
If more mask bits for a test point are set to '1' then the 'OR' of all signals is shown.
The internal signals arrive after 50 ns at the testpoints; delayed by 2 FFs.

Bit #	TESTMASK 0	TESTMASK 1	TESTMASK 2	TESTMASK 3
15	vme_en	Vme_wr	dtack	**Clk40 ***
14	wr_regsf0_fe(0)	wr_regsf0_fe(4)	wr_regsf0_fe(1)	wr_regsf0_fe(3)
13	out_ch1(31)	Rd_regs0_e(0)	out_ch3(31)	wr_regsf0_fe(2)
12	out_ch1(15)	Rd_regs10_1e(0)	out_ch3(15)	wr_regsf0_fe(5)
11	out_ch0(31)	wr_regs10_1e(0)	out_ch2(31)	Vme_en_spy(0)
10	out_ch0(15)	Vme_rd_spy(0)	out_ch2(15)	Vme_we_spy(0)
9	**BCRes_int ***	**L1Res_int ***	**Inc_bcerror ***	Bcres_dlyed
8	Simulating(0)	Simulating(1)	Simulating(2)	Simulating(3)
7	debug_ch0(7)	debug_ch1(7)	debug_ch2(7)	debug_ch3(7)
6	debug_ch0(6)	debug_ch1(6)	debug_ch2(6)	debug_ch3(6)
5	debug_ch0(5)	debug_ch1(5)	debug_ch2(5)	debug_ch3(5)
4	debug_ch0(4)	debug_ch1(4)	debug_ch2(4)	debug_ch3(4)
3	debug_ch0(3)	debug_ch1(3)	debug_ch2(3)	debug_ch3(3)
2	debug_ch0(2)	debug_ch1(2)	debug_ch2(2)	debug_ch3(2)
1	debug_ch0(1)	debug_ch1(1)	debug_ch2(1)	debug_ch3(1)
0	debug_ch0(0)	debug_ch1(0)	debug_ch2(0)	debug_ch3(0)

***) Default values after power-up**

Debug signals from sim_spy control:

```
debug(7) <= start_syy;
debug(6) <= stop_syy;
debug(5) <= start_syuend;
debug(4) <= stop_syuend;
debug(3) <=  set_ensimspy;
debug(2) <=  clr_ensimspy;
debug(1) <= stop_syncff;
debug(0) <= stop_syncff1;
```

Various delays between external/internal signals and arrival on test point:

Clk40 ➔	at TP3	<5 ns
EN_REC2 ➔ vme_enn➔ vme_en	at TP0	~100 ns
WR_REC2 ➔ vme_wr	at TP1	~75 ns
NDTACK_REC2 ➔ dtack	at TP2	~50 ns

```
run_v          <= wr_regsf0_fe(0);  -- start immediately and run without stopping
run_v_u_end    <= wr_regsf0_fe(1);  -- start immediately and run 1x until end of memory
run_sync       <= wr_regsf0_fe(2);  -- start at begin of orbit and run without stopping
run_sync_u_end <= wr_regsf0_fe(3);  -- start at begin of orbit and run 1x until end of orbit
stop_v         <= wr_regsf0_fe(4);  -- stop immediately
stop_sync      <= wr_regsf0_fe(5);  -- stop at end of orbit
```

7.7.1 Error

Error: SIM/SPY does not work properly in SYNC modes ➔ Check:

- MAX_BC_NUMBER = x"0deb"
- BCRES Signal from backplane
- BCRES_int on TESTPOINT0 (set TESTMASK0 =x0200)
- BCRES_dlyed on TESTPOINT3 (set TESTMASK3 =x0200)

Normal period of BCRES signals: 89 100 ns = 25ns*3564

7.8 COMMAND REGISTER

Bit 15 -8: not used
Bit7: RES_ORBIT_WITH_BCR
V1005: = 1 resets Orbit Counter with next BCRES signal when ResOrbit has arrived
= 0 resets Orbit Counter immediately as in TCS chip(Version V0007) :default
Bit 6: EN_L1A_FOR _SPY
V1004: bit is not used anymore
V1003: =1 enables the L1A Signal to work like a "run_sync_until_end" command to do crate-wide data spying of the same orbit.
Bit 5: EN_BC_ERROR = 1 enables BC_ERROR to set REC_STATUS accordingly
Bit 4: REC_IS_READY = 1 REC chip is READY ➔ goes to STATUS bits
Bit 3: SIMU_MODE CH3 = 1 …insert simulation data; =0 …forward trigger data
Bit 2: SIMU_MODE CH2
Bit 1: SIMU_MODE CH1
Bit 0: SIMU_MODE CH0
Default value after Power-up:
=X"0010" …READY=1, forwardTriggerdata, disabled BC-error

7.9 Command Pulse

Bit 15 -0: not used

7.10 Spy Status register

***** V1004: Spy Status when stopped by L1A ******

xxx0 0020	L1A_SPY_STATUS0	-/r	for Channel 0
xxx0 0022	L1A_SPY_STATUS1	-/r	for Channel 0
xxx0 0024	L1A_SPY_STATUS2	-/r	for Channel 0
xxx0 0026	L1A_SPY_STATUS3	-/r	for Channel 0

bit15: waiting flag,
bit12-0: last spy address = 2*bc_nr ….13 bit address: range = 0….7127

SPY_STATUS=X"0FFF" other spy/sim procedure without L1A is active or has been done
SPY_STATUS=X"8000" waiting_flag: Waiting for L1A to stop spying
SPY_STATUS=X"1aaa" L1A arrived at spy_memory address '1aaa' and waiting flag =0.
See Remark 3 below.

Remark1:
The command pulse 'stop_spying_with_next_L1A' sets the STATUS word: X"8000" (waiting_flag=1 and saved address =000).
The next L1A reset the waiting flag and loads the actual address into the status word.
Any run_xxx command clears the status word (X"0000").
Remark2:

2 addresses contain data of 1 BC (bunch crossing) resp. of 1 event

Remark3:

The REC chips contain 4k/80 MHz dual-port memories. During one orbit the memory is filled one time until address = 4095 and a second time until address = 3031 (=2*3564 -1 – 4096).

When reading the memory we have to consider:

- 4095 < Spy_status < 7128 ➔ read from address = spy_status – 4095
- 0 ≤ Spy_status < 4096 ➔ read from address = spy_status

When reading a history datum h (= h/2 bunch crossings before)

- If (Spy_status - h) <0 and (Spy_status < 4096) then
  ➔ read from address = 3031 + spy_status - h
- If (Spy_status - h) >0 and (Spy_status < 4096) then
  ➔ read from address = Spy_status –h
- ((Spy_status - h) < 4096) and (Spy_status > 4095)
  ➔ read from address = Spy_status - h
- ((Spy_status - h) > 4095) and (4095<Spy_status < 7128)
  ➔ read from address = Spy_status - h - 4095

7.11 ORBIT_NR

xxx0 0028	ORBIT_NR_L	-/r
xxx0 002A	ORBIT_NR_H	-/r

Contains 32 bit Orbitnumber corresponding to stored SPY_STATUS0 word.

If SPY_STATUS0 = X"0FFF" … ORBIT_NR_L/H = actual orbit number
// before sending a 'stop_spying_with_next_L1A';
// after power_up, after a run_xxx command pulse

If SPY_STATUS0 = X"0abc" … ORBIT_NR_L/H = 'frozen' orbit_nr when L1A arrived.
// Then the register is kept frozen until a following run_xx command.

If SPY_STATUS0 = X"8000" … ORBIT_NR_L/H = 'FFFF_FFFF'
// while waiting for L1A

7.12 RUN-STOP commands

The commands are valid for all channels in the REC chip.

xxx0 00F0	run_v	w/
xxx0 00F2	run_v_until_end	w/
xxx0 00F4	run_sync	w/
xxx0 00F6	run_sync_until_end	w/
xxx0 00F8	stop_v	w/
xxx0 00FA	stop_sync	w/
xxx0 00FC	stop_spying_with_next_L1A	w/

8 Description

8.1 SPYING of trigger data with L1A

- set channels to spy mode ➔ command_register (3:0) =0000
- run_sync (xxx0 00F4) ➔ spying starts with next bcres and runs continuously;
  ➔ spy_status=0000

- send 'stop_spying_with_next_L1A' ➔ spy_status=8000 …waiting_flag=1
  - Program reads spy_status registers until waiting_flag=0
- The next L1A stops spying. ➔ spy_status=0abc = spy-address when L1A arrived
- Read SPY_STATUS and then the sim_spy memories to find the trigger data.

As the L1A signal is sent to all boards in the GT crate, it is possible to stop spying at the same time in the PSB boards(with firmware:V10013) and later also in the GMT_IN chips (to be done). This option is useful to check latencies and to find synchronisation problems.

8.2 Common run_stop commands

Inside the GTL9U board the common RUN-STOP signals allow sending simulation data concurrently from the REC- to the COND chips.
They can also be used to stop spying concurrently to compare input data to each other.