GANDALF Framework
User Guide

Version 1.1
December 2011

ALBERT-LUDWIGS-UNIVERSITÄT FREIBURG
PHYSIKALISCHES INSTITUT
ABTEILUNG PROF. DR. KAY KÖNIGSMANN
PHYSIK DER TEILCHEN UND KERNE

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
09/09/2011	1.0	Initial release CPLD design 2.1.3
12/06/2011	1.1	CPLD design 2.2.2 Fast Registers Configuration Memory Registers Contact

Table of Contents

1 GANDALF Firmware Registers

1.1 VME Interface Registers

Table 1.1: VME address for communication with the GANDALF CPLD. The [HEXID(8)] to address the GANDALF modules located in a VME crate can be selected by DIP switches (*SW1*, *SW2*) Please refer to http://hadron.physik.uni-freiburg.de/gandalf for updates.

VME address	Name	Description
0xE0[HEXID(8)]00FC	BOARDSTATUS	Returns 0x[CONF(4)][HEXID(8)]'00'[GeoAdd(6)]'00'[SN(10)] of module with [HEXID(8)].
0xE0[HEXID(8)]3000	R_SPY_FIFO	Reads the valid data word from the output of the *Spy FIFO*.
0xE0[HEXID(8)]0010	ARMBROADCAST	Resets module with [HEXID(8)] to accept broadcasting data.
0xE0[HEXID(8)]8000	BC_FPGA_CFG	VME address to write the broadcast configuration data for the DSP (*U9*) or the MEM-FPGA (*U25*) to the VME backplane (the DSP-FPGA has to be configured first). The rightmost GANDALF module located in the VME crate must be addressed.
0xE0[HEXID(8)]0014	BC_SWITCH	VME address to switch from DSP-FPGA (*U9*) configuration to MEM-FPGA (*U25*) configuration. The rightmost GANDALF module located in the VME crate must be addressed.
0xE0[HEXID(8)]0004	DISPLAY_W	VME address to write data to the front display. Use the DATAWORD(32) = DISP0(8)&DISP1(8)&DISP2(8)&DISP3(8) to define the values shown on the 4 segment display. The following values can be set with corresponding hex values 0x00 to 0x12: = *0,1,2,...,9,A,B,C,D,E,F,G,S,X*.
0xE0[HEXID(8)]2XXX	CFM_R_W	VME address to read or write data from/to the configuration memory. See section 1.3.
0xE0[HEXID(8)]7XXX	SET_FR	VME address to set a fast register command. See section 1.2.

1.2 Fast Registers

Table 1.2: Table of Fast Register Commands. In the VHDL firmware the Fast Registers are represented in a 256 bit bus. *VHDL_SIG* gives the corresponding number of the bus member. How to set a Fast Register is explained in 1.4. Please refer to http://hadron.physik.uni-freiburg.de/gandalf for updates.

FR Name	FR_ADDR	VHDL_SIG	PULSE_TYPE	Description
DAC_CALIB	0x00C	3	const (TYPE = 0x0/1)	Disables/Enables DAC calibration.
TRG_TEMP_RDOUT	0x010	4	pulse (TYPE = 0x2)	Triggers readout of all temperature values (AMC, FPGA temp sensors) and all voltage values (FPGA voltage sensors).
RD_EEPROM_UP	0x018	6	pulse (TYPE = 0x2)	Triggers readout of EEPROM data to configuration memory of mezzanine in card slot up (this takes 200ms).
WR_EEPROM_UP	0x01C	7	pulse (TYPE = 0x2)	Triggers storage of configuration memory data to EEPROM of mezzanine in card slot up (this takes 1.0s)
RD_EEPROM_DN	0x020	8	pulse (TYPE = 0x2)	Triggers readout of EEPROM data to configuration memory of mezzanine in card slot down (this takes 200ms).

Table 1.2: Table of Fast Register Commands (continued).

FR Name	FR_ADDR	VHDL_SIG	PULSE_TYPE	Description
WR_EEPROM_DN	0x024	9	pulse (TYPE = 0x2)	Triggers storage of configuration memory data to EEPROM of mezzanine in card slot down (this takes 1.0s).
$LOAD\_SI$	0x028	10	pulse (TYPE = 0x2)	Triggers configuration of the $SI5326$ clock synthesizer located on the GANDALF module and all SI5326 on the mounted AMCs. The configuration data is stored in the configuration memory registers SI_CONF_DATA0 - SI_CONF_DATA11
SET_DACs	0x02C	11	pulse (TYPE = 0x2)	Triggers configuration of the $AD5665R$ DACs located on the AMCs. The DAC values are stored in the configuration memory registers DAC_VAL0 - DAC_VAL3.

Table 1.2: Table of Fast Register Commands (continued).

FR Name	FR_ADDR	VHDL_SIG	PULSE_TYPE	Description
TRG_DAC_CALIB	0x030	12	pulse (TYPE = 0x2)	Triggers the automatic DAC calibration. New DAC values are written to the *DAC_VAL*0 - *DAC_VAL*3 registers.
LOAD_G_CONF_VAL	0x034	13	pulse (TYPE = 0x2)	Updates all configuration values written into the configuration memory to the active FPGA logic.
*VME_RESET*0	0x038	14	pulse (TYPE = 0x2)	Performs a reset on reset level 0.
*VME_RESET*1	0x03C	15	pulse (TYPE = 0x2)	Performs a reset on reset level 1.
*VME_RESET*2	0x040	16	pulse (TYPE = 0x2)	Performs a reset on reset level 2.
EXT_BOS	0x044	17	pulse (TYPE = 0x2)	Used to generate an artificial BOS[1] signal. Can be used if no TCS[2] is adapted.
EXT_EOS	0x048	18	pulse (TYPE = 0x2)	Used to generate an artificial EOS[3] signal. Can be used if no TCS is adapted.

[1]Begin Of Spill
[2]Trigger Control System
[3]End Of Spill

Table 1.2: Table of Fast Register Commands (continued).

FR Name	FR_ADDR	VHDL_SIG	PULSE_TYPE	Description
ART_TRG	0x04C	19	pulse (TYPE = 0x2)	Used to generate an artificial trigger signal. Can be used if no TCS is adapted.
SLINK_RESET	0x050	20	pulse (TYPE = 0x2)	Triggers the S-LINK reset procedure as explained in the S-LINK specification.
SMUX_RESET	0x054	21	pulse (TYPE = 0x2)	Triggers a reset signal on the *SRESET* pin C30 (P2,see Tab. 3.2) to perform a common reset on up to four SMUX transition cards.
WR_STATUS	0x058	22	pulse (TYPE = 0x2)	Writes the status flags of the GANDALF to cfmem adress STATUS1.
RES_CT_BOS	0x05C	23	pulse (TYPE = 0x2)	Enables the reset of the coarse time at the next BOS. This syncronizes differen G boards.

Table 1.2: Table of Fast Register Commands (continued).

FR Name	FR_ADDR	VHDL_SIG	PULSE_TYPE	Description
SELF_TRG	0x060 - 0x09C	24 - 39	const (TYPE = 0x1)	Specifies the analog channel which is allowed to generate a "self trigger", if the analog signal reaches a value above the constant defined in the configuration registers *THRES_VAL*0 - *THRES_VAL*3.
TOGGLE_TCS_RATE	0x0A0	40	const (TYPE = 0x0 or 0x1)	Used to manually toggle the RATE Pin. This FastRegister can be used if the CLC016 on the Fibre GIMLI can not lock to the clock signal.
FR_ReadoutTigerReady	0x0A4	41	const (TYPE = 0x1)	Must be set to 1, when ReadoutTiger is ready. Enables the VXS SLINK outputs.
$F\_FR_{T}riggerTigerReady$	0x0A8	42	const (TYPE = 0x1)	Must be set to 1, when TriggerTiger is ready. Enables the VXS Trigger outputs.
FR_StartVXSLinkCal	0x0A0	43	const (TYPE = 0x0 or 0x1)	Used to start the calibration of the VXS data transfer link.

Table 1.2: Table of Fast Register Commands (continued).

FR Name	FR_ADDR	VHDL_SIG	PULSE_TYPE	Description
PROGRAMM_UP_CARD	0x0C8	50	const (TYPE = 0x1)	Used to trigger the programming of the FPGA located on the OMC in Mezzanine Card Slot Up.
PROGRAMM_DN_CARD	0x0CC	51	const (TYPE = 0x1)	Used to trigger the programming of the FPGA located on the OMC in Mezzanine Card Slot Down.
READ_INIT_DONE	0x0D0	52	pulse (TYPE = 0x2)	Read INIT and DONE-Pin of the ARWEN FPGA.
DATA_VALID	0x0D4	53	pulse (TYPE = 0x2)	Used for programming.
SWEEP_SI	0x0F0	60	pulse (TYPE = 0x2)	Starts sweep process of the SI clock sythesizers for phase offset alignment.
UP_SWEEP_STAT	0x0F4	61	pulse (TYPE = 0x2)	Updates the counters for the statistics sweep algorithm.
RESET_SI	0x0F8	62	pulse (TYPE = 0x2)	Resets all mounted SI chips (GANDALF and Mezzanine Card slots) using the reset in pin of the chip.

1.3 Configuration Memory Registers

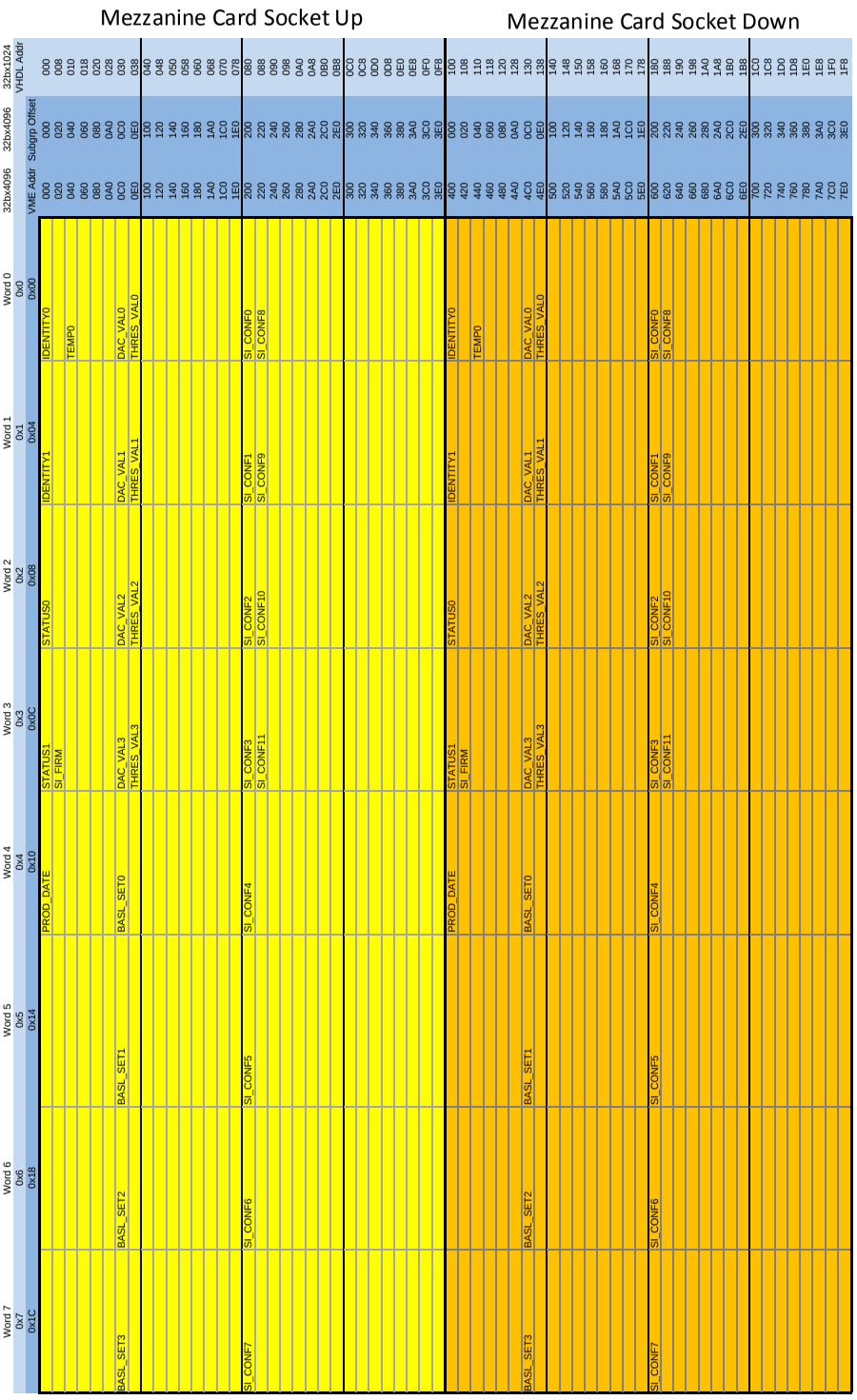

Figure 1.1: Overview of the Configuration Memory Registers for the mounted Mezzanine Cards on Mezzanine Card Socket Up and Down. A detailed description of the registers can be found in Tab. 1.3. Please refer to http://hadron.physik.uni-freiburg.de/gandalf for updates.

GANDALF module

Word 7	Word 6	Word 5	Word 4	Word 3	Word 2	Word 1	Word 0	32bx4096	32bx4096	32bx1024
0x7	0x6	0x5	0x4	0x3	0x2	0x1	0x0			VHDL Addr
0x1C	0x18	0x14	0x10	0x0C	0x08	0x04	0x00	VME Addr	Subgrp Offset	
			PROD_DATE	STATUS1	STATUS0	IDENTITY1	IDENTITY0	800	000	200
				SI_FIRM	MEM_FIRM	DSP_FIRM	CPLD_FIRM	820	020	208
						TEMP1	TEMP0	840	040	210
						VCCAUX1	VCCAUX0	860	060	218
						VCCINT1	VCCINT0	880	080	220
UCD_RL7	UCD_RL6	UCD_RL5	UCD_RL4	UCD_RL3	UCD_RL2	UCD_RL1	UCD_RL0	8A0	0A0	228
								8C0	0C0	230
								8E0	0E0	238
								900	100	240
								920	120	248
								940	140	250
								960	160	258
								980	180	260
								9A0	1A0	268
								9C0	1C0	270
								9E0	1E0	278
SI_CONF7	SI_CONF6	SI_CONF5	SI_CONF4	SI_CONF3	SI_CONF2	SI_CONF1	SI_CONF0	A00	200	280
				SI_CONF11	SI_CONF10	SI_CONF9	SI_CONF8	A20	220	288
								A40	240	290
								A60	260	298
						ARWEN_CF(1)	ARWEN_CF(0)	A80	280	2A0
								AA0	2A0	2A8
								AC0	2C0	2B0
								AE0	2E0	2B8
G_CONF7	G_CONF6	G_CONF5	G_CONF4	G_CONF3	G_CONF2	G_CONF1	G_CONF0	B00	300	2C0
FRA_LAT7	FRA_LAT6	FRA_LAT5	FRA_LAT4	FRA_LAT3	FRA_LAT2	FRA_LAT1	FRA_LAT0	B20	320	2C8
FRA_LAT15	FRA_LAT14	FRA_LAT13	FRA_LAT12	FRA_LAT11	FRA_LAT10	FRA_LAT9	FRA_LAT8	B40	340	2D0
							SCALER	B60	360	2D8
								B80	380	2E0
						TDC_CONFIG1	TDC_CONFIG0	BA0	3A0	2E8
								BC0	3C0	2F0
								BE0	3E0	2F8
				PAT_GEN_PAR3	PAT_GEN_PAR2	PAT_GEN_PAR1	PAT_GEN_PAR0	C00	400	300
								C20	420	308
								C40	440	310
								C60	460	318
								C80	480	320
								CA0	4A0	328
								CC0	4C0	330
								CE0	4E0	338
								D00	500	340
								D20	520	348
								D40	540	350
								D60	560	358
								D80	580	360
								DA0	5A0	368
								DC0	5C0	370
								DE0	5E0	378
								E00	600	380
								E20	620	388
								E40	640	390
								E60	660	398
								E80	680	3A0
								EA0	6A0	3A8
								EC0	6C0	3B0
								EE0	6E0	3B8
								F00	700	3C0
								F20	720	3C8
								F40	740	3D0
								F60	760	3D8
								F80	780	3E0
								FA0	7A0	3E8
								FC0	7C0	3F0
								FE0	7E0	3F8

Figure 1.2: Overview of the Configuration Memory Registers for the GANDALF module. A detailed description of the registers can be found in Tab. 1.3. Please refer to http://hadron.physik.uni-freiburg.de/gandalf for updates.

Table 1.3: Table of Configuration Memory Registers. *CM_REG* gives the name of the configuration memory register and *VME_CMR_ADDR* the corresponding VME address. The colours orange and yellow declare that the register is used for mezzanine cards installed in mezzanine card slot up and down. Green coloured VME addresses declare that the register is used for the GANDALF module. See also Figs. 1.1, 1.2. The corresponding VHDL address for the configuration memory can also be found in this Figures. *CMR_DETAIL* gives the names and in brackets the size of the detail of a register. How to set a Configuration Memory Register is explained in 1.4. Please refer to http://hadron.physik.uni-freiburg.de/gandalf for updates.

CM_REG	VME_CMR_ADDR	CMR_DETAIL	Description
*IDENTITY*0 (11 dnto 0)	0x000 0x400	*SERIAL_NO* (12)	Defines the Serial Number of the installed mezzanine card in mezzanine card slot up and down.
*IDENTITY*0 (15 dnto 12)	0x000 0x400	*CARD_TYPE* (4)	Defines the mezzanine card type: **0x0** AMC, **0x1** DMC, **0x2** OMC
*IDENTITY*0 (17 dnto 16)	0x000 0x400	*ADC_RES* (2)	Defines the resolution of the used ADCs, if AMC is installed: **0x0** 12bit, **0x1** 14 bit
*IDENTITY*0 (19 dnto 18)	0x000 0x400	*ADC_CONFIG* (2)	Defines the configuration of the input circuit of the used AMC: **0x0** normal, **0x1** interleaved
*IDENTITY*0 (21 dnto 20)	0x000 0x400	*DMC_CONFIG* (2)	Defines the differential buffer orientation used with the DMC: **0x0** input, **0x1** output
*IDENTITY*0 (6 dnto 0)	0x800	*STAT_MON* (10)	Defines the status of the System Monitoring (TBD).
*IDENTITY*0 (11 dnto 7)	0x800	*GEO_ADDR* (10)	Defines the geographic address (slot number in Crate) where the GANDALF module is installed to.
*IDENTITY*0 (21 dnto 12)	0x800	*GEO_ID* (10)	Defines the COMPASS experiment geographic ID. For GANDALF scaler this is the base geo ID of cable 0. Increment the ID for each cable.

Table 1.3: Table of Configuration Memory Registers (continued).

CM_REG	VME_CMR_ADDR	CMR_DETAIL	Description
$IDENTITY0$ (31 dnto 22)	0x800	$SERIAL\_NO$ (10)	Defines the Serial Number of the GANDALF module.
$IDENTITY1$ (9 dnto 0)	0x804	SRC_ID (10)	Defines the COMPASS experiment source ID for the GANDALF module.
$IDENTITY1$ (19 dnto 10)	0x804	$2ND\_SRC\_ID$ (10)	If second chip definitions like chipF1, chip Scaler etc are implemented this defines the COMPASS experiment source ID of the second implementation.
$IDENTITY1$ (29 dnto 20)	0x804	$SMUX\_ID$ (10)	If data in SMUX format is produced by the GANDALF module this defines its source ID.
$STATUS0$ (31 dnto 30)	0x008 0x408	$EEPROM\_CONF$ (2)	Defines if the EEPROM is configured: **0x0**,**0x3** not configured, **0x1** configured
$STATUS0$ (29 dnto 28)	0x008 0x408	$MEZZ\_INST$ (2)	Defines if a mezzanine card is installed (is defined physically when no EEPROM is attached to the I2C bus): **0x1** not installed, **0x0**,**0x3** installed

Table 1.3: Table of Configuration Memory Registers (continued).

CM_REG	VME_CMR_ADDR	CMR_DETAIL	Description
*STATUS*0 (27 dnto 20)	0x808	*DESIGN_TYPE* (8)	Defines which firmware is loaded into the GANDALF module: **0x00** GANDALF base design, **0x04** GANDALF transient analyzer, **0x08** GANDALF TDC, **0x09** GANDALF scaler, **0x10** GANDALF meantimer, **0x11** GANDALF TDC with Scaler, **0x14** GANDALF Arwen readout, **0xF0** GANDALF pattern generator, **0xF1** GANDALF module test
*STATUS*1 (2 dnto 0)	0x80C	SI_G_STATUS(3)	(**2** downto **0**) represents the status of the *out_of_phase*, *loss_of_signal* and *loss_of_lock* of the SI located on the GANDALF module.
*STATUS*1 (6 dnto 4)	0x80C	SI_B_STATUS(3)	(**2** downto **0**) represents the status of the *out_of_phase*, *loss_of_signal* and *loss_of_lock* of the SI located on the MCS down.
*STATUS*1 (10 dnto 8)	0x80C	SI_A_STATUS(3)	(**2** downto **0**) represents the status of the *out_of_phase*, *loss_of_signal* and *loss_of_lock* of the SI located on the MCS up.

Table 1.3: Table of Configuration Memory Registers (continued).

CM_REG	VME_CMR_ADDR	CMR_DETAIL	Description
*STATUS*1 (14 dnto 12)	0x80C	TCS_SL_STATUS(3)	(**2** downto **0**) represents the status of the *io_m_error* (representing the io manager status), *VLDOWN* (representing the SLINK status) and *TCS_LOL* (representing the TCS lock status of the CLC016).
*STATUS*1 (18 dnto 16)	0x80C	RESET_STATUS(3)	(**2** downto **0**) represents the status of the resets *RST_Global_Stup_**i**.*
*STATUS*1 (22 dnto 20)	0x80C	ALARM_STATUS(3)	(**2** downto **0**) represents the status of the alarm flags gernerated by the sysmon core inside the Virtex-5 FPGA in the following order (2) VCCAUX, (1)VCCINT, (0) TEMP. The default threshold ranges are: • Temperature: (upper limit 95C) • VCCINT: (0.8V - 1.1V) • VCCAUX: (2.38V - 2.63V)
PROD_DATE (31 dnto 0)	0x010 0x410 0x810	*PROD_DATE* (32)	Defines the production date of the mezzanine card or the GANDALF module. Date format is 0xDDMMYYYY.

Table 1.3: Table of Configuration Memory Registers (continued).

CM_REG	VME_CMR_ADDR	CMR_DETAIL	Description
CPLD_FIRM (31 dnto 0)	0x820	*CPLD_FIRM* (32)	Defines the firmware version of the CPLD (*U8*). Format is for firmware is 0xVVVVYYYY, where VVVV gives the firmware revision number and YYYY the year of release.
DSP_FIRM (31 dnto 0)	0x824	*DSP_FIRM* (32)	Defines the firmware version of the DSP-FPGA (*U9*).
MEM_FIRM (31 dnto 0)	0x828	*MEM_FIRM* (32)	Defines the firmware version of the MEM-FPGA (*U25*).
SI_FIRM (31 dnto 0)	0x02C 0x42C 0x82C	*SI_FIRM* (32)	Defines the firmware version of the SI (*U11*).
*TEMP*0 (9 dnto 0)	0x040 0x440	*AMC_TMP_TOP* (10)	Temperature measured by the *TMP*175 placed on the top layer of the AMC. All temperature and voltage values are updated after toggling Fast Register *TRG_TEMP_RDOUT* (Tab. 1.2).
*TEMP*0 (25 dnto 16)	0x040 0x440	*AMC_TMP_BOT* (10)	Temperature measured by the *TMP*175 placed on the bottom layer of the AMC.
*TEMP*0 (9 dnto 0)	0x840	*DSP_TMP* (10)	Temperature measured by the System-Monitor (inside the DSP-FPGA (*U9*)).
*TEMP*0 (19 dnto 10)	0x840	*DSP_TMP_MIN* (10)	Minimum temperature measured by the SystemMonitor since configuration.
*TEMP*0 (29 dnto 20)	0x840	*DSP_TMP_MAX* (10)	Maximum temperature measured by the SystemMonitor since configuration.

Table 1.3: Table of Configuration Memory Registers (continued).

CM_REG	VME_CMR_ADDR	CMR_DETAIL	Description
$TEMP1$ (9 dnto 0)	0x844	MEM_TMP (10)	Temperature measured by the System-Monitor (inside the MEM-FPGA (*U25*)).
$TEMP1$ (19 dnto 10)	0x844	MEM_TMP_MIN (10)	Minimum temperature measured by the SystemMonitor since configuration.
$TEMP1$ (29 dnto 20)	0x844	MEM_TMP_MAX (10)	Maximum temperature measured by the SystemMonitor since configuration.
$VCCAUX0$ (9 dnto 0)	0x860	DSP_AUX (10)	Auxiliary voltage ($U_{AUX} = 2.5\,\mathrm{V}$) measured by the SystemMonitor (inside the DSP-FPGA (*U9*)).
$VCCAUX0$ (19 dnto 10)	0x860	DSP_AUX_MIN (10)	Minimum auxiliary voltage measured by the SystemMonitor since configuration.
$VCCAUX0$ (29 dnto 20)	0x860	DSP_AUX_MAX (10)	Maximum auxiliary voltage measured by the SystemMonitor since configuration.
$VCCAUX1$ (9 dnto 0)	0x864	MEM_AUX (10)	Auxiliary voltage ($U_{AUX} = 2.5\,\mathrm{V}$) measured by the SystemMonitor (inside the MEM-FPGA (*U25*)).
$VCCAUX1$ (19 dnto 10)	0x864	MEM_AUX_MIN (10)	Minimum auxiliary voltage measured by the SystemMonitor since configuration.
$VCCAUX1$ (29 dnto 20)	0x864	MEM_AUX_MAX (10)	Maximum auxiliary voltage measured by the SystemMonitor since configuration.
$VCCINT0$ (9 dnto 0)	0x880	DSP_INT (10)	Internal voltage ($U_{INT} = 1.0\,\mathrm{V}$) measured by the SystemMonitor (inside the DSP-FPGA (*U9*)).
$VCCINT0$ (19 dnto 10)	0x880	DSP_INT_MIN (10)	Minimum internal voltage measured by the SystemMonitor since configuration.

Table 1.3: Table of Configuration Memory Registers (continued).

CM_REG	VME_CMR_ADDR	CMR_DETAIL	Description
*VCCINT*0 (29 dnto 20)	0x880	*DSP_INT_MAX* (10)	Maximum internal voltage measured by the SystemMonitor since configuration.
*VCCINT*1 (9 dnto 0)	0x884	*MEM_INT* (10)	Temperature measured by the System-Monitor (inside the MEM-FPGA (*U25*)).
*VCCINT*1 (19 dnto 10)	0x884	*MEM_INT_MIN* (10)	Minimum internal voltage measured by the SystemMonitor since configuration.
*VCCINT*1 (29 dnto 20)	0x884	*MEM_INT_MAX* (10)	Maximum internal voltage measured by the SystemMonitor since configuration.
UCD_RL0-7 (31 dnto 0)	0x8A0 - 0x81C	TBD	Voltage information measured from the UCD9081 can be stored here. TBD.
DAC_VAL0-3 (15 dnto 0)	0x0C0-0x0CC 0x4C0-0x4CC	*DAC_VAL0-3* (16)	DAC values for the first four ADCs (CH0 - CH3) located on a AMC.
DAC_VAL0-3 (31 dnto 16)	0x0C0-0x0CC 0x4C0-0x4CC	*DAC_VAL4-7* (16)	DAC values for the second four ADCs (CH4 - CH7) located on a AMC.
BASL_SET0-3 (15 dnto 0)	0x0D0-0x0DC 0x4D0-0x4DC	*BASL_SET0-3* (16)	DAC values for the first four ADCs (CH0 - CH3) located on a AMC.
BASL_SET0-3 (31 dnto 16)	0x0D0-0x0DC 0x4D0-0x4DC	*BASL_SET4-7* (16)	DAC values for the second four ADCs (CH4 - CH7) located on a AMC.
THR_VAL0-3 (15 dnto 0)	0x0E0-0x0EC 0x4E0-0x4EC	*THR_VAL0-3* (16)	Threshold values (in ADC LSB) to generate a self trigger for the first four ADCs (CH0 - CH3) located on a AMC.
THR_VAL0-3 (31 dnto 16)	0x0E0-0x0EC 0x4E0-0x4EC	*THR_VAL4-7* (16)	Threshold values (in ADC LSB) to generate a self trigger for the second four ADCs (CH4 - CH7) located on a AMC.

Table 1.3: Table of Configuration Memory Registers (continued).

CM_REG	VME_CMR_ADDR	CMR_DETAIL	Description
SI_CONF0-11 (31 dnto 0)	0x200-0x22C 0x600-0x62C 0xA00-0xA2C	*SI_CONF0-11* (32)	Configuration register for the programming of the SI5326. The registers are arranged as defined in the text format output of the software utility, *DSPLLsim*, provided by Silicon Labs.
ARWEN_CF0-1 (31 dnto 0)	0xA80-0xA84	*ARWEN_CF0-1* (32)	Address where the Configuration-File and the INIT and DONE data is written to.
G_CONF0 (15 dnto 0)	0xB00	*GEN_LAT* (16)	Generic latency value for the time window wherein occuring signals were processed. This register is used in transient analyzer (GTA) and TDC mode. In GTA TCS trigger arrives the internal logic with 25ns latency. The step size is 4ns in interleaved mode.
G_CONF0 (31 dnto 16)	0xB00	*GEN_FRA* (16)	Generic window size value for the time window wherein occuring signals were processed. This register is used in transient analyzer and TDC mode.
G_CONF1 (15 dnto 0)	0xB04	*GEN_THR* (16)	Generic threshold value to generate a self trigger.
G_CONF1 (31 dnto 16)	0xB04	*GEN_FT* (16)	Generic threshold value to generate a fast trigger send to the TIGER module.
G_CONF2 (13 dnto 0)	0xB08	*GEN_BASE* (14)	Baseline Level to where the input circuit are set to.

Table 1.3: Table of Configuration Memory Registers (continued).

CM_REG	VME_CMR_ADDR	CMR_DETAIL	Description
G_CONF2 (16)	0xB08	*GEN_CLK_SRC* (1)	Defines type of mounted GIMLI card. **0** fiber, **1** OCXO
G_CONF3 (7 dnto 0)	0xB0C	*DAC_SET_TRL* (8)	Defines number of failed DAC calibrations until error occurs.
G_CONF3 (31 dnto 16)	0xB0C	*DAC_INIT* (16)	Defines initial DAC value.
G_CONF4 (5 dnto 0)	0xB10	*CAL_TRG* (6)	Defines TCS calibration trigger type accepted by GANDALF.
G_CONF4 (8)	0xB10	*CTRG_EN* (1)	**1** Enables or **0** disables calibration trigger
G_CONF4 (13 dnto 10)	0xB10	*RDM* (4)	Defines readout mode parameter (see section 2.1).
G_CONF5 (5 dnto 0)	0xB14	*FRAC* (6)	Defines Fraction Value f for the cfd algorithm. $f = \frac{1}{2^{FRAC}}$
G_CONF5 (12 dnto 8)	0xB14	*DELAY* (5)	Defines Delay Value for the cfd algorithm.
G_CONF5 (23 dnto 16)	0xB14	*THRESHOLD* (8)	Defines Threshold value in ADC LSB. This value defines the trigger condition for the cf algorithm and is used as offset for the integral calculation.
G_CONF6 (31 dnto 0)	0xB18	TBD	TBD
G_CONF7 (31 dnto 0)	0xB1C	TBD	TBD
FRA_LAT0-15 (15 dnto 0)	0xB20-0xB4C	*LATENCY0-15* (16)	Latency value for the time window wherein occuring signals were processed (CH0-CH16). This register is used in transient analyzer and TDC mode.

Table 1.3: Table of Configuration Memory Registers (continued).

CM_REG	VME_CMR_ADDR	CMR_DETAIL	Description
FRA_LAT0-15 (31 dnto 16)	0xB20-0xB4C	*FRAMEW0-15* (16)	Window size/Framewidth value for the time window wherein occurring signals were processed (CH0-CH16). This register is used in transient analyzer and TDC mode.
SCALER (31 dnto 23)	0xB60	*SC_LATENCY* (9)	Latency for the GANDALF Scaler.
SCALER (22 dnto 15)	0xB60	*GATE_LATENCY* (8)	Gate Latency for the GANDALF Scaler.
TDC_CONFIG0-1	0xBA0-0xBA4	*TDC_CONFIG0-1* (32)	Configuration Register for the M1-TDC.
PAT_GEN_PAR0-3	0xC00-0xC0C	*PAT_GEN_PAR0-3* (32)	Configuration Register for the Pattern Generator.

1.4 The *vme_write* command

- **Fast Register:** To set a Fast Register use the command

  #>./vme_write E0[HEXID(8)][GADDRESS(16)] [PULSE_TYPE(4)],

  where

  $$[GADDRESS(16)] = [CMD\_SEL(4)][FR\_ADDR(12)],$$

  and $[HEXID(8)]$ is the selectable module address which can be changed using the DIP switches (see Fig.1.3). $[CMD\_SEL(4)]$ is 0x7 for Fast Registers. $[FR\_ADDR(12)]$ is the addressed Fast Register and $[PULSE\_TYPE(4)]$ defines the pulse type. There are 256 Fast Registers (see Tab.1.2) and 3 pulse types. With the $[PULSE\_TYPE(4)]$, the type of the Fast Register can be defined: 0x0 sets the Register to const '0', 0x1 to '1', 0x2 generates a pulse with a width of one clock cycle of the 40 MHZ clock provided by the CDCE clock synthesizer chip.

- **Configuration Memory Register:** To write data to a Configuration Memory Register use the command

  #>./vme_write E0[HEXID(8)][GADDRESS(16)] [DATAWORD(32)]

  where

  $$[GADDRESS(16)] = [CMD\_SEL(4)][CFMEM\_ADDR(12)]$$

  $[CMD\_SEL(4)]$ is 0x2 for the communication with the Configuration Memory. $[CFMEM\_ADDR(12)]$ is the "VME address" of the Configuration Memory Register and $[DATAWORD(32)]$ can obtain the data which has to be written to the addressed Configuration Memory Register.
  To address the registers of the memory from the FPGA logic, the mapping of the RAMB36 is used, which differs from the VME address range. To calculate the address offsets, the following rule is used:

  $$\begin{aligned}[RAMB36\_ADDR(12)] \times 4 &= [CFMEM\_ADDR(12)] \\ &= [\text{CFMEM\_SUBADDR(12)}] + \text{SUBGRP\_OFFSET},\end{aligned}$$

  where $[RAMB36\_ADDR(12)]$ is the address value for RAM access with a range of 0x3FF, $[CFMEM\_ADDR(12)]$ has a range of 0xFFF and $[CFMEM\_SUBADDR(12)]$ has a range of 0x3FF for the mezzanines subgroups and 0x7FF for the GANDALF subgroup (see Fig.1.1 & 1.2).
  To read data from CFM use the command

  #>./vme_write E0[HEXID(8)][GADDRESS(16)].

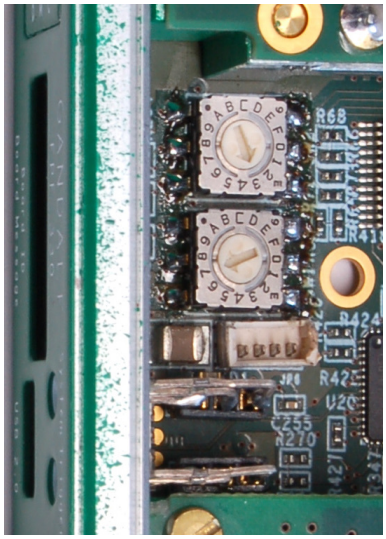

Figure 1.3: Picture of the DIP switches to select the [HEXID(8)] of the GANDALF modules. In this picture for example the [HEXID(8)] = 0x37 is set.

1.5 The *gandalf_status* command

2 GANDALF Data Format Definitions

2.1 GANDALF Data Format

The data format chosen for the GANDALF module is compatible with the S-LINK protocol as defined in the S-LINK specification The structure of the S-LINK header is given in the first three words in Tab. 2.4. The *format(8)* byte, written in the third word of the header, covers extended information about the configuration of the GANDALF modules (see Tab. 2.1). There is a separation between the first event in a run (FER) and a first event in a spill (FES). A detailed description of the COMPASS online data format can be found in

2.1.1 GANDALF Transient Analyzer Data

A GANDALF module equipped with two AMCs can operate as a *transient analyzer*. The AMC can be used in *normal* or *time-interleaved* digitization mode. Therefore, the number of effective digitization channels changes respectively. In *normal mode* 8 ADCs of one AMC build eight effective channels. In the *time-interleaved mode* the 8 ADCs build 4 effective channels. This enumeration of effective channels is used to declare the corresponding channels in the data format.
When the GANDALF module is used as transient analyzer, the following information is given in the FER and FES:

- The serial numbers and configuration status (e.g. *normal* or *time-interleaved mode*) of the used modules and mezzanine cards.
- The sourceID of the GANDALF module and its location in the VME crate (crate slot number).
- The version number of the firmware for the FPGAs.
- Temperatures and voltages measured by the monitoring system.
- DAC and threshold values for the different analog input channels.

The detailed data format structure of the FER and the FES is given in Tabs. 2.4 and 2.5 respectively. Some of the values mentioned above can be set by writing them into the *Configuration Memory* and are printed in bold letters (App. 1.3).
For events generated by a physics trigger, from two kinds of data modes can be selected:

- **Normal Data Mode**

  In this mode, the GANDALF module outputs the processed data in data blocks per hit, if the algorithm has detected one or more pulses inside the given window with the corresponding latency. In *Normal Data Mode* the dataset for one physical event consists of n data blocks and includes the following information:

  – Integral of all sampling values measured in the frame. A frame is a defined number of sampling words.

  – The maximum amplitude of the corresponding hit.

  – The calculated time information of the corresponding hit. It is separated in 28 bit coarse time information (number of sampling units) and 10 bit high resolution time information (subdividing the sampling unit by 1024).

  The detailed structure of an event in *Normal Data Mode* is given in Tab. 2.6.

- **Debug Data Mode**

  In this mode, the GANDALF module outputs both, the during the defined time window recorded frame and the processed data. The length of the frames depends on the selected window length.The structure of the GANDALF *Debug Data Mode* consists of an event header, data blocks and an event trailer, containing the following information:

  – System monitoring information: FPGA configured, Si locked, TCS status, voltages and temperatures.

  – Readout mode (RDM): Normal or Debug Data Mode.

  – The window size and the digitized frame data.

  – Processed data, as mentioned above, for each hit inside the frame.

  – A pulse detection flag which signalizes that the algorithm has detected hits inside the frame.

The detailed structure of an event in *Debug Data Mode* is given in Tab. 2.7. With the register *PRESCALE* (see Tab. 1.3), the occurrence of events in Debug Data format can be selected. By setting the integer value n in the *PRESCALE* register, every n-th event is transferred in Debug Data format and all remaining events are in *Normal Data Mode*. For $n = 0x0$, any event is processed in *Debug Data Mode*.

2.1.2 GANDALF TDC Data

A GANDALF module equipped with two DMCs can operate as a 128 channel TDC. In this mode, the GANDALF module generates the same data structure as the CATCH module with 16 installed F1-TDCs. A GANDALF module generates TDC data in the same format as a CATCH module to comply with existing offline analysis software The window and latency values for the TDC functionality can be set by choosing the corresponding registers in the *Configuration Memory* (see Tab. 1.3).

Table 2.1: Format definition

7	6	5	4	3	2	1	0
1 = first event of a run (else 0)	1 = HOTL. CMC 0 = TDC-CMC/ G-TDC	1 = l & t edge 0 = l or t edge	1 = high res 0 = norm res	1 = latch mode 0 = no latch mode		1 = sparsified mode 0 = debug mode	1=TDC readout
1 = first event of a run (else 0)	0 = GeSiCA (APV) 1 = GANDALF-ADC 2 = GANDALF-Scaler 3 = Scaler-CMC 4 = FI-ADC HOTLink 5 = RICH HOTFiber 6 = 7 = Scaler HOTLink			0 = nml mode, processed data 1 = nml mode, frame data 2 = nml mode, debug data 3 = 4 = ilm mode, processed data 5 = ilm mode, frame data 6 = ilm mode, debug data 7 =			0=ADC or scaler readout

Table 2.2: Format of TDC data in the debug mode

31 0

0-16× (Header, n× DATA, Trailer)

0	tbo	event no. (6)	trigger time (9)	xor	ch/ch. ID (6)	port (4)	PLL locked
1	0	ch/ch. ID (6)	DATA (16)			port (4)	PLL locked
...							
1	0	ch/ch. ID (6)	DATA (16)			port (4)	PLL locked
0	tbo	event no. (6)	trigger time (9)	xor	ch/ch. ID (6)	port (4)	PLL locked

error word preceding Header/Data/Trailer

FFF (12)	error coding (20)

Table 2.3: Format of TDC data in the sparsified mode

31 0

only data words will be written

geo ID (10)	ch/ch. ID (6)	DATA (16)
...		
geo ID (10)	ch/ch. ID (6)	DATA (16)

error word preceding data with error

FFF (12)		error coding (20)			
1	0	ch/ch. ID (6)	DATA (16)	port (4)	PLL locked

error word preceding faulty header/trailer

FFF (12)		error coding (20)					
0	tbo	event no. (6)	trigger time (9)	xor	ch/ch. ID (6)	port (4)	PLL locked

Table 2.4: First Event of Run Header from GANDALF.

31 0

S-Link Header

err	ev. type (5)	source ID (10)			event size excl. CTRL (16)		
stat	spill no. (11)		event no. (20)				
format (8)			#errorwords (8)		tcs error (8)		status (8)

First Event of Run Header

GANDALF S/N (10)			GANDALF srcID (10)		CrateSlot (5)		#AMC(2)		SysMon (5)
DSP-FPGA usrID (16)					MEM-FPGA usrID(16)				
CPLD Firmware Version (32)									
DSP Firmware Version (32)									
MEM Firmware Version(32)									
TBD (24)								SI Conf. Type(8)	
TBD (16)					GANDALF status (16)				
TBD (6)	DSP-FPGA TEMP (10)				TBD (6)	MEM-FPGA TEMP (10)			
TBD (6)	DSP-FPGA VCC1V0 (10)				TBD (6)	MEM-FPGA VCC1V0 (10)			
TBD (6)	DSP-FPGA VCCAUX (10)				TBD (6)	MEM-FPGA VCCAUX (10)			

TBD (16)	AMC1 S/N (10)	AMC1 cfg (6)
TBD (16)	AMC2 S/N (10)	AMC2 cfg (6)

Table 2.4: Format of the First Event of Spill Header from GANDALF (continued).

31 0

TBD (6)	LOLS (2)	AMC1 Ch. ON (8)	CALDONE (8)	CALERR (8)
TBD (6)	LOLS (2)	AMC2 Ch. ON (8)	CALDONE (8)	CALERR (8)
TBD (8)		AMC1 Temp. top&bot (24)		
TBD (8)		AMC2 Temp. top&bot (24)		

DAC 0 value (16)	DAC 8 value (16)
...	
DAC 7 value (16)	DAC 15 value (16)

Thrsld. 0 value (16)	Thrsld. 8 value (16)
...	
Thrsld. 7 value (16)	Thrsld. 15 value (16)

Data words

Table 2.5: First Event of Spill Header from GANDALF.

31 0

S-Link Header

err	ev. type (5)	source ID (10)	event size excl. CTRL (16)	
stat	spill no. (11)	event no. (20)		
format (8)		#errorwords (8)	tcs error (8)	status (8)

First Event of Spill Header

GANDALF S/N (10)	GANDALF srcID (10)	CrateSlot (5)	#AMC(2)	SysMon (5)
GANDALF status (16)		TBD (16)		
TBD (6)	DSP-FPGA TEMP (10)	TBD (6)	MEM-FPGA TEMP (10)	
TBD (6)	DSP-FPGA VCC1V0 (10)	TBD (6)	MEM-FPGA VCC1V0 (10)	
TBD (6)	DSP-FPGA VCCAUX (10)	TBD (6)	MEM-FPGA VCCAUX (10)	

TBD (16)			AMC1 S/N (10)	AMC1 cfg (6)
TBD (16)			AMC2 S/N (10)	AMC2 cfg (6)
TBD (6)	LOLS (2)	AMC1 Ch. ON (8)	CALDONE (8)	CALERR (8)
TBD (6)	LOLS (2)	AMC2 Ch. ON (8)	CALDONE (8)	CALERR (8)
TBD (8)		AMC1 Temp. top&bot (24)		
TBD (8)		AMC2 Temp. top&bot (24)		

Table 2.5: Format of the First Event of Spill Header from GANDALF (continued).

31 0

DAC 0 value (16)	DAC 8 value (16)
...	
DAC 7 value (16)	DAC 15 value (16)
Thrsld. 0 value (16)	Thrsld. 8 value (16)
...	
Thrsld. 7 value (16)	Thrsld. 15 value (16)
Data words	

Table 2.6: Data format of the GANDALF Normal Data Mode.

31 0

0-16× (n× DATA Blocks)

1	ch (4)	Baseline (11)	Integral (16)
1	Coarse Time DATA MSB (17)		max Amplitude (14)
1	Coarse Time DATA LSB (21)		High Res Time(10)

Table 2.7: Data format of the GANDALF Debug Data Mode.

31 0

0-16× (Header, n× DATA Blocks, Trailer)

0	0	event no. (6)	ch. ID (4)	SysMon (5)	Framesize (11)	RDM (4)

1	0	Data Word 0 (14)	0	0	Data Word 1 (14)
1	0	Data Word 2 (14)	0	0	Data Word 3 (14)

...

1	0	Data Word n-1 (14)	0	0	Data Word n (14)

1	ch (4)	Baseline (11)	Integral (16)	
1	Coarse Time DATA MSB (17)		max Amplitude (14)	
1	Coarse Time DATA LSB (21)		High Res Time(10)	
1	cfd threshold (8)	cfd frac (6)	cfd delay (5)	Frame Time(12)

0	1	event no. (6)	ch. ID (4)	SysMon (5)	Framesize (11)	RDM (4)

3 Connectivity Tables

3.1 GANDALF Backplane Connectors

Table 3.1: VME connector (*J1*) pinout.

	Z	A	B	C	D
1	-	D(0)	-	D(8)	+5V
2	GND	D(1)	-	D(9)	GND
3	-	D(2)	-	D(10)	+12V
4	GND	D(3)	-	D(11)	+12V
5	-	D(4)	-	D(12)	-
6	GND	D(5)	-	D(13)	+12V
7	-	D(6)	-	D(14)	+12V
8	GND	D(7)	-	D(15)	-
9	-	GND	-	GND	GAP
10	GND	-	BG3IN	-	GA0
11	-	GND	BG3OUT	BERR	GA1
12	GND	DS1	-	SYSRES	+3.3V
13	-	DS0	-	LWORD	GA2
14	GND	WRITE	-	AM5	+3.3V
15	-	GND	-	A(23)	GA3
16	GND	DTACK	AM0	A(22)	+3.3V
17	-	GND	AM1	A(21)	GA4
18	GND	AS	AM2	A(20)	+3.3V
19	-	GND	AM3	A(19)	-
20	GND	IACK	GND	A(18)	+3.3V
21	-	IACKIN	-	A(17)	-
22	GND	IACKOUT	-	A(16)	+3.3V
23	-	AM4	GND	A(15)	-
24	GND	A(7)	-	A(14)	+3.3V
25	-	A(6)	-	A(13)	-
26	GND	A(5)	-	A(12)	+3.3V
27	-	A(4)	-	A(11)	LI/I
28	GND	A(3)	-	A(10)	+3.3V
29	-	A(2)	-	A(9)	LI/O
30	GND	A(1)	-	A(8)	+3.3V
31	-	-12V	-	+12V	GND
32	GND	+5V	+5V	+5V	+5V

Table 3.2: VME connector (*J2*) pinout. The UD bus in row A and C are used for the S-LINK interface.

	Z	A	B	C	D
1	-	SINIT	+5V	SDONE	-
2	GND	SCLK	GND	SDIN	-
3	-	GND	-	SPROG	-
4	GND	UD(0)	A(24)	GND	-
5	-	UD(2)	A(25)	UD(1)	-
6	GND	UD(4)	A(26)	UD(3)	-
7	-	UD(6)	A(27)	UD(5)	-
8	GND	GND	A(28)	UD(7)	-
9	-	UD(8)	A(29)	GND	-
10	GND	UD(10)	A(30)	UD(9)	-
11	-	UD(12)	A(31)	UD(11)	-
12	GND	UD(14)	GND	UD(13)	-
13	-	GND	+5V	UD(15)	-
14	GND	UD(16)	D(16)	UD(17)	-
15	-	UD(18)	D(17)	GND	-
16	GND	UD(20)	D(18)	UD(19)	-
17	-	UD(22)	D(19)	UD(21)	-
18	GND	GND	D(20)	UD(23)	-
19	-	UD(24)	D(21)	UD(25)	-
20	GND	UD(26)	D(22)	GND	-
21	-	UD(28)	D(23)	UD(27)	-
22	GND	UD(30)	GND	UD(29)	-
23	-	GND	D(24)	UD(31)	-
24	GND	UCTRL	D(25)	UDW0	-
25	-	UDW1	D(26)	UDTEST	-
26	GND	UDRESET	D(27)	GND	-
27	-	GND	D(28)	UDWEN	-
28	GND	UDCLK	D(29)	GND	-
29	-	GND	D(30)	LFF	-
30	GND	LDOWN	D(31)	SRESET	-
31	-	-	GND	-	GND
32	GND	-	+5V	-	+5V

Table 3.3: VXS connector (*J9*) pinout.

	A	B	C	D	E	F	G
1	TRG_2P	TRG_2N	GND	TRG_0P	TRG_0N	GND	VXS_SCL
2	GND	TRG_6P	TRG_6N	GND	TRG_1P	TRG_1N	GND
3	TRG_3P	TRG_3N	GND	TRG_4P	TRG_4N	GND	VXS_SDA
4	GND	TRG_12P	TRG_12N	GND	TRG_5P	TRG_5N	GND
5	-	-	GND	-	-	GND	-
6	GND	-	-	GND	-	-	GND
7	-	-	GND	-	-	GND	-
8	GND	-	-	GND	-	-	GND
9	-	-	GND	-	-	GND	-
10	GND	-	-	GND	-	-	GND
11	-	-	GND	-	-	GND	-
12	GND	TRG_14P	TRG_14N	GND	TRG_10P	TRG_10N	GND
13	TRG_8P	TRG_8N	GND	TRG_7P	TRG_7N	GND	-
14	GND	TRG_13P	TRG_13N	GND	TRG_15P	TRG_15N	GND
15	TRG_11P	TRG_11N	GND	TRG_9P	TRG_9N	GND	-

3.2 Mezzanine Card Socket Up/Down Connectors

Table 3.4: Mezzanine Card Socket Up connector pinout (*J5*).

1	PORT1_0N	VCC-12V	2
3	PORT1_0P	VCC-12V	4
5	PORT1_1N	VCA3V3	6
7	PORT1_1P	VCA3V3	8
9	PORT1_2N	VCA3V3	10
11	PORT1_2P	VCA3V3	12
13	PORT1_3N	VCA3V3	14
15	PORT1_3P	VCA3V3	16
17	PORT1_4N	VCA3V3	18
19	PORT1_4P	VCA3V3	20
21	PORT1_5N	PORT3_0N	22
23	PORT1_5P	PORT3_0P	24
25	PORT1_6N	PORT3_1N	26
27	PORT1_6P	PORT3_1P	28
29	PORT1_7N	PORT3_2N	30
31	PORT1_7P	PORT3_2P	32
33	PORT1_8N	PORT3_3N	34
35	PORT1_8P	PORT3_3P	36
37	PORT1_9N	PORT3_4N	38
39	PORT1_9P	PORT3_4P	40
41	PORT1_10N	PORT3_5N	42
43	PORT1_10P	PORT3_5P	44
45	PORT1_11N	PORT3_6N	46
47	PORT1_11P	PORT3_6P	48
49	PORT1_12N	PORT3_7N	50
51	PORT1_12P	PORT3_7P	52
53	PORT1_13N	PORT3_8N	54
55	PORT1_13P	PORT3_8P	56
57	PORT1_DRYN	PORT3_9N	58
59	PORT1_DRYP	PORT3_9P	60
61	GND	PORT3_10N	62
63	ADC_NC	PORT3_10P	64
65	ADCOFF	PORT3_11N	66
67	GND	PORT3_11P	68
69	SI_RST#	PORT3_12N	70
71	SI_LOL	PORT3_12P	72

73	GND	PORT3_13N	74
75	GND	PORT3_13P	76
77	CLK_155MHZ_P	PORT3_DRYN	78
79	CLK_155MHZ_N	PORT3_DRYP	80
81	GND	PORT5_0N	82
83	GND	PORT5_0P	84
85	GND	PORT5_1N	86
87	GND	PORT5_1P	88
89	GND	PORT5_2N	90
91	GND	PORT5_2P	92
93	GND	PORT5_3N	94
95	GND	PORT5_3P	96
97	GND	PORT5_4N	98
99	GND	PORT5_4P	100
101	PORT7_0N	PORT5_5N	102
103	PORT7_0P	PORT5_5P	104
105	PORT7_1N	PORT5_6N	106
107	PORT7_1P	PORT5_6P	108
109	PORT7_2N	PORT5_7N	110
111	PORT7_2P	PORT5_7P	112
113	PORT7_3N	PORT5_8N	114
115	PORT7_3P	PORT5_8P	116
117	PORT7_4N	PORT5_9N	118
119	PORT7_4P	PORT5_9P	120
121	PORT7_5N	PORT5_10N	122
123	PORT7_5P	PORT5_10P	124
125	PORT7_6N	PORT5_11N	126
127	PORT7_6P	PORT5_11P	128
129	PORT7_7N	PORT5_12N	130
131	PORT7_7P	PORT5_12P	132
133	PORT7_8N	PORT5_13N	134
135	PORT7_8P	PORT5_13P	136
137	PORT7_9N	PORT5_DRYN	138
139	PORT7_9P	PORT5_DRYP	140
141	PORT7_10N	VCA5V0	142
143	PORT7_10P	VCA5V0	144
145	PORT7_11N	VCA5V0	146
147	PORT7_11P	VCA5V0	148
149	PORT7_12N	VCA5V0	150
151	PORT7_12P	VCA5V0	152
153	PORT7_13N	VCA5V0	154

155	PORT7_13P	VCA5V0	156
157	PORT7_DRYN	VCA5V0	158
159	PORT7_DRYP	VCA5V0	160

Table 3.5: Mezzanine Card Socket Up connector pinout (*J6*).

1	AMCTMS	PORT0_DRYP	2
3	AMCTCK	PORT0_DRYN	4
5	AMCTDI	PORT0_13P	6
7	AMCTDO	PORT0_13N	8
9	GND	PORT0_12P	10
11	GP_SDA	PORT0_12N	12
13	GP_SCL	PORT0_11P	14
15	GND	PORT0_11N	16
17	GND	PORT0_10P	18
19	GND	PORT0_10N	20
21	PORT2_DRYP	PORT0_9P	22
23	PORT2_DRYN	PORT0_9N	24
25	PORT2_13P	PORT0_8P	26
27	PORT2_13N	PORT0_8N	28
29	PORT2_12P	PORT0_7P	30
31	PORT2_12N	PORT0_7N	32
33	PORT2_11P	PORT0_6P	34
35	PORT2_11N	PORT0_6N	36
37	PORT2_10P	PORT0_5P	38
39	PORT2_10N	PORT0_5N	40
41	PORT2_9P	PORT0_4P	42
43	PORT2_9N	PORT0_4N	44
45	PORT2_8P	PORT0_3P	46
47	PORT2_8N	PORT0_3N	48
49	PORT2_7P	PORT0_2P	50
51	PORT2_7N	PORT0_2N	52
53	PORT2_6P	PORT0_1P	54
55	PORT2_6N	PORT0_1N	56
57	PORT2_5P	PORT0_0P	58
59	PORT2_5N	PORT0_0N	60
61	PORT2_4P	GND	62
63	PORT2_4N	GND	64
65	PORT2_3P	GND	66
67	PORT2_3N	SI_SDA	68
69	PORT2_2P	SI_SCL	70
71	PORT2_2N	GND	72
73	PORT2_1P	SI_LOS	74
75	PORT2_1N	MEZZ_ADDR	76
77	PORT2_0P	GND	78
79	PORT2_0N	CLK_38MHZ	80
81	PORT4_DRYP	GND	82

83	PORT4_DRYN	GND	84
85	PORT4_13P	GND	86
87	PORT4_13N	GND	88
89	PORT4_12P	GND	90
91	PORT4_12N	GND	92
93	PORT4_11P	GND	94
95	PORT4_11N	GND	96
97	PORT4_10P	GND	98
99	PORT4_10N	GND	100
101	PORT4_9P	PORT6_DRYP	102
103	PORT4_9N	PORT6_DRYN	104
105	PORT4_8P	PORT6_13P	106
107	PORT4_8N	PORT6_13N	108
109	PORT4_7P	PORT6_12P	110
111	PORT4_7N	PORT6_12N	112
113	PORT4_6P	PORT6_11P	114
115	PORT4_6N	PORT6_11N	116
117	PORT4_5P	PORT6_10P	118
119	PORT4_5N	PORT6_10N	120
121	PORT4_4P	PORT6_9P	122
123	PORT4_4N	PORT6_9N	124
125	PORT4_3P	PORT6_8P	126
127	PORT4_3N	PORT6_8N	128
129	PORT4_2P	PORT6_7P	130
131	PORT4_2N	PORT6_7N	132
133	PORT4_1P	PORT6_6P	134
135	PORT4_1N	PORT6_6N	136
137	PORT4_0P	PORT6_5P	138
139	PORT4_0N	PORT6_5N	140
141	VCC3V3	PORT6_4P	142
143	VCC3V3	PORT6_4N	144
145	VCC3V3	PORT6_3P	146
147	VCC3V3	PORT6_3N	148
149	VCC3V3	PORT6_2P	150
151	VCC3V3	PORT6_2N	152
153	VCC+12V	PORT6_1P	154
155	VCC+12V	PORT6_1N	156
157	VCC+12V	PORT6_0P	158
159	VCC+12V	PORT6_0N	160

Table 3.6: Mezzanine Card Socket Down connector pinout (*J7*).

1	PORT9_0N	VCC-12V	2
3	PORT9_0P	VCC-12V	4
5	PORT9_1N	VCA3V3	6
7	PORT9_1P	VCA3V3	8
9	PORT9_2N	VCA3V3	10
11	PORT9_2P	VCA3V3	12
13	PORT9_3N	VCA3V3	14
15	PORT9_3P	VCA3V3	16
17	PORT9_4N	VCA3V3	18
19	PORT9_4P	VCA3V3	20
21	PORT9_5N	PORT11_0N	22
23	PORT9_5P	PORT11_0P	24
25	PORT9_6N	PORT11_1N	26
27	PORT9_6P	PORT11_1P	28
29	PORT9_7N	PORT11_2N	30
31	PORT9_7P	PORT11_2P	32
33	PORT9_8N	PORT11_3N	34
35	PORT9_8P	PORT11_3P	36
37	PORT9_9N	PORT11_4N	38
39	PORT9_9P	PORT11_4P	40
41	PORT9_10N	PORT11_5N	42
43	PORT9_10P	PORT11_5P	44
45	PORT9_11N	PORT11_6N	46
47	PORT9_11P	PORT11_6P	48
49	PORT9_12N	PORT11_7N	50
51	PORT9_12P	PORT11_7P	52
53	PORT9_13N	PORT11_8N	54
55	PORT9_13P	PORT11_8P	56
57	PORT9_DRYN	PORT11_9N	58
59	PORT9_DRYP	PORT11_9P	60
61	GND	PORT11_10N	62
63	ADC_NC	PORT11_10P	64
65	ADCOFF	PORT11_11N	66
67	GND	PORT11_11P	68
69	SI_RST#	PORT11_12N	70
71	SI_LOL	PORT11_12P	72
73	GND	PORT11_13N	74
75	GND	PORT11_13P	76
77	CLK_155MHZ_P	PORT11_DRYN	78
79	CLK_155MHZ_N	PORT11_DRYP	80
81	GND	PORT13_0N	82

83	GND	PORT13_0P	84
85	GND	PORT13_1N	86
87	GND	PORT13_1P	88
89	GND	PORT13_2N	90
91	GND	PORT13_2P	92
93	GND	PORT13_3N	94
95	GND	PORT13_3P	96
97	GND	PORT13_4N	98
99	GND	PORT13_4P	100
101	PORT15_0N	PORT13_5N	102
103	PORT15_0P	PORT13_5P	104
105	PORT15_1N	PORT13_6N	106
107	PORT15_1P	PORT13_6P	108
109	PORT15_2N	PORT13_7N	110
111	PORT15_2P	PORT13_7P	112
113	PORT15_3N	PORT13_8N	114
115	PORT15_3P	PORT13_8P	116
117	PORT15_4N	PORT13_9N	118
119	PORT15_4P	PORT13_9P	120
121	PORT15_5N	PORT13_10N	122
123	PORT15_5P	PORT13_10P	124
125	PORT15_6N	PORT13_11N	126
127	PORT15_6P	PORT13_11P	128
129	PORT15_7N	PORT13_12N	130
131	PORT15_7P	PORT13_12P	132
133	PORT15_8N	PORT13_13N	134
135	PORT15_8P	PORT13_13P	136
137	PORT15_9N	PORT13_DRYN	138
139	PORT15_9P	PORT13_DRYP	140
141	PORT15_10N	VCA5V0	142
143	PORT15_10P	VCA5V0	144
145	PORT15_11N	VCA5V0	146
147	PORT15_11P	VCA5V0	148
149	PORT15_12N	VCA5V0	150
151	PORT15_12P	VCA5V0	152
153	PORT15_13N	VCA5V0	154
155	PORT15_13P	VCA5V0	156
157	PORT15_DRYN	VCA5V0	158
159	PORT15_DRYP	VCA5V0	160

Table 3.7: Mezzanine Card Socket Down connector pinout (*J8*).

1	AMCTMS	PORT8_DRYP	2
3	AMCTCK	PORT8_DRYN	4
5	AMCTDI	PORT8_13P	6
7	AMCTDO	PORT8_13N	8
9	GND	PORT8_12P	10
11	GP_SDA	PORT8_12N	12
13	GP_SCL	PORT8_11P	14
15	GND	PORT8_11N	16
17	GND	PORT8_10P	18
19	GND	PORT8_10N	20
21	PORT10_DRYP	PORT8_9P	22
23	PORT10_DRYN	PORT8_9N	24
25	PORT10_13P	PORT8_8P	26
27	PORT10_13N	PORT8_8N	28
29	PORT10_12P	PORT8_7P	30
31	PORT10_12N	PORT8_7N	32
33	PORT10_11P	PORT8_6P	34
35	PORT10_11N	PORT8_6N	36
37	PORT10_10P	PORT8_5P	38
39	PORT10_10N	PORT8_5N	40
41	PORT10_9P	PORT8_4P	42
43	PORT10_9N	PORT8_4N	44
45	PORT10_8P	PORT8_3P	46
47	PORT10_8N	PORT8_3N	48
49	PORT10_7P	PORT8_2P	50
51	PORT10_7N	PORT8_2N	52
53	PORT10_6P	PORT8_1P	54
55	PORT10_6N	PORT8_1N	56
57	PORT10_5P	PORT8_0P	58
59	PORT10_5N	PORT8_0N	60
61	PORT10_4P	GND	62
63	PORT10_4N	GND	64
65	PORT10_3P	GND	66
67	PORT10_3N	SI_SDA	68
69	PORT10_2P	SI_SCL	70
71	PORT10_2N	GND	72
73	PORT10_1P	SI_LOS	74
75	PORT10_1N	MEZZ_ADDR	76
77	PORT10_0P	GND	78
79	PORT10_0N	CLK_38MHZ	80
81	PORT12_DRYP	GND	82

83	PORT12_DRYN	GND	84
85	PORT12_13P	GND	86
87	PORT12_13N	GND	88
89	PORT12_12P	GND	90
91	PORT12_12N	GND	92
93	PORT12_11P	GND	94
95	PORT12_11N	GND	96
97	PORT12_10P	GND	98
99	PORT12_10N	GND	100
101	PORT12_9P	PORT14_DRYP	102
103	PORT12_9N	PORT14_DRYN	104
105	PORT12_8P	PORT14_13P	106
107	PORT12_8N	PORT14_13N	108
109	PORT12_7P	PORT14_12P	110
111	PORT12_7N	PORT14_12N	112
113	PORT12_6P	PORT14_11P	114
115	PORT12_6N	PORT14_11N	116
117	PORT12_5P	PORT14_10P	118
119	PORT12_5N	PORT14_10N	120
121	PORT12_4P	PORT14_9P	122
123	PORT12_4N	PORT14_9N	124
125	PORT12_3P	PORT14_8P	126
127	PORT12_3N	PORT14_8N	128
129	PORT12_2P	PORT14_7P	130
131	PORT12_2N	PORT14_7N	132
133	PORT12_1P	PORT14_6P	134
135	PORT12_1N	PORT14_6N	136
137	PORT12_0P	PORT14_5P	138
139	PORT12_0N	PORT14_5N	140
141	VCC3V3	PORT14_4P	142
143	VCC3V3	PORT14_4N	144
145	VCC3V3	PORT14_3P	146
147	VCC3V3	PORT14_3N	148
149	VCC3V3	PORT14_2P	150
151	VCC3V3	PORT14_2N	152
153	VCC+12V	PORT14_1P	154
155	VCC+12V	PORT14_1N	156
157	VCC+12V	PORT14_0P	158
159	VCC+12V	PORT14_0N	160

3.3 Mezzanine Card Socket Central Connector

Table 3.8: Mezzanine Card Socket Central connector pinout (*J10*).

1	+3.3V	+3.3V	2
3	GND	GND	4
5	CLK_N	GND	6
7	CLK_P	GND	8
9	GND	LOCK	10
11	GND	RATE	12
13	DATA_N	GND	14
15	DATA_P	GND	16
17	GND	GND	18
19	+5V	+5V	20

4 VXS Crate Power Supply Specification

Table 4.1: Selection of power supply modules for the VXS crate. The total maximum power of a VXS crate with this power supply configuration 2760W. The crate including the power supply was manufactured by Wiener Inc.

Voltage	# modules	Current
+12.0V	3	138A
+ 5.0V	1	115A
+ 3.3V	1	115A
-12.0V	1	46A

5 GANDALF Module Device Overview

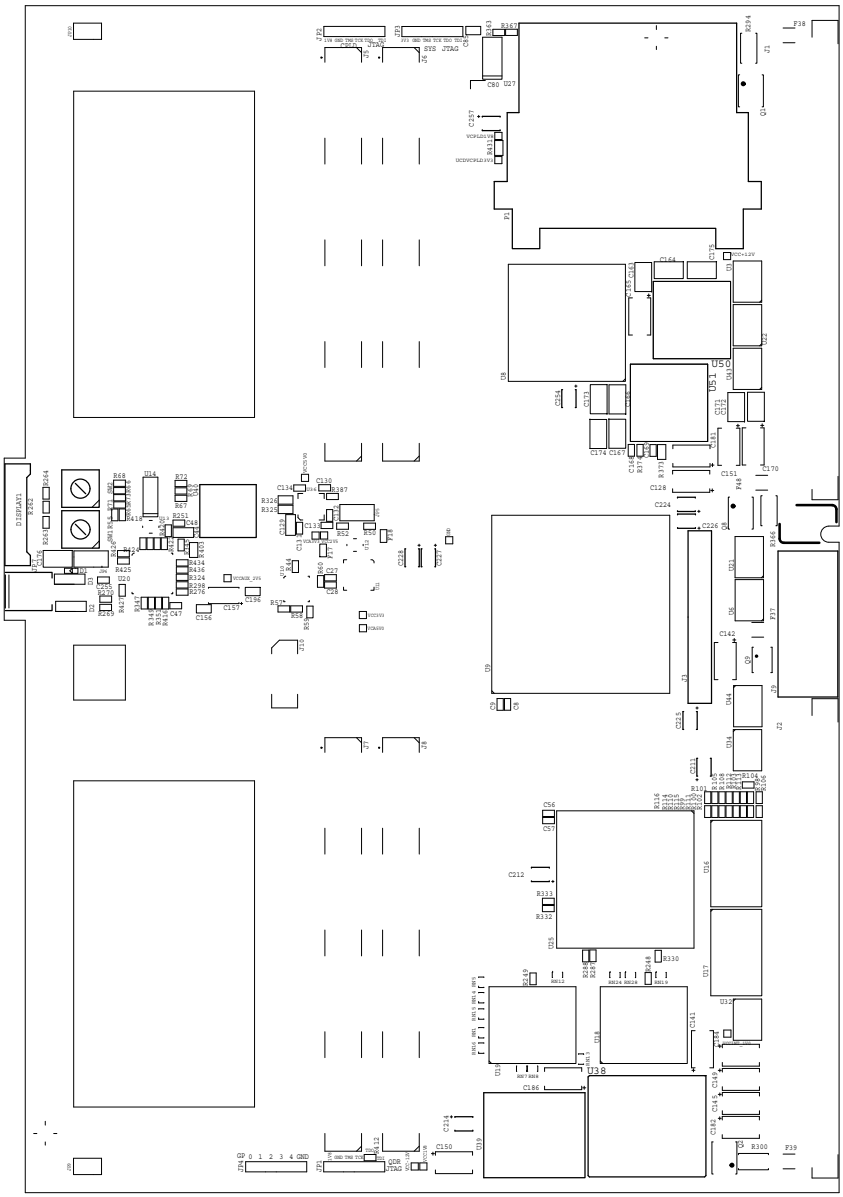

Figure 5.1: Overview of the placed devices including Reference Designators on the top side of the PCB.

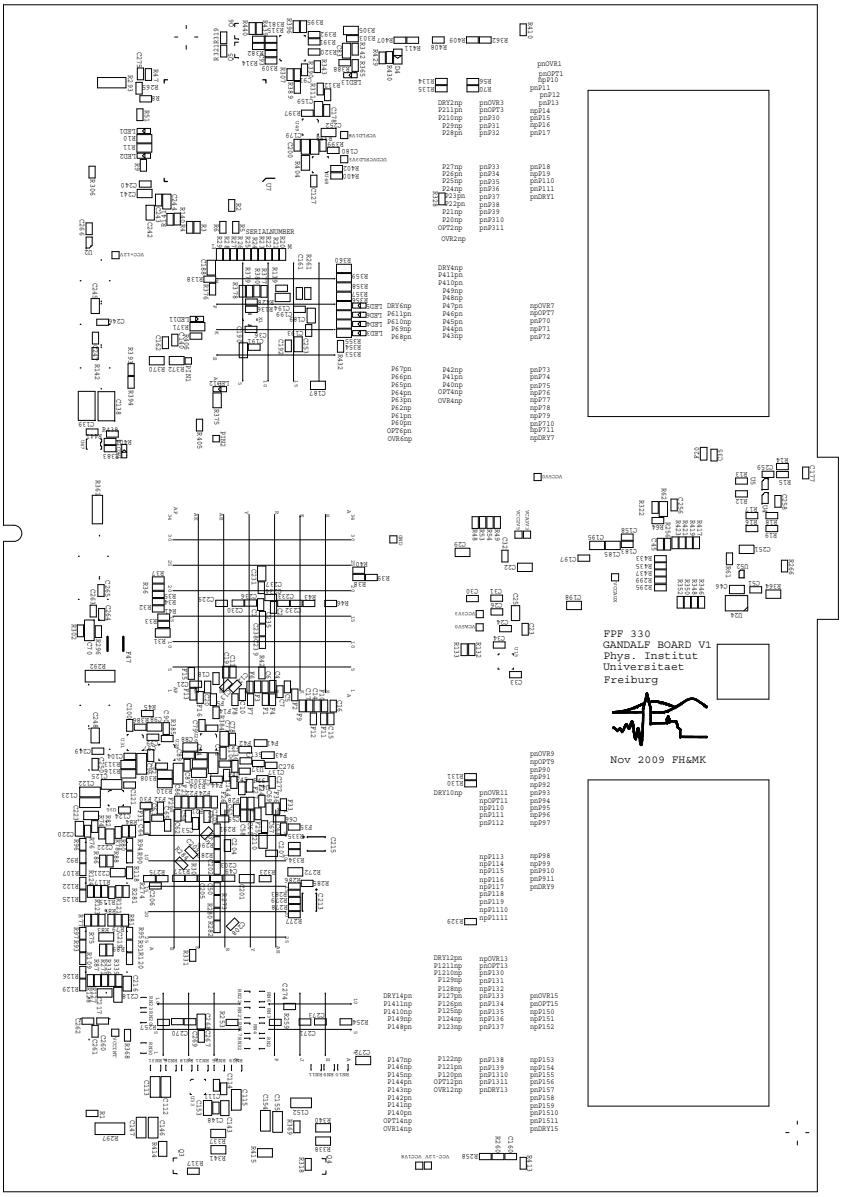

Figure 5.2: Overview of the placed devices including Reference Designators on the bottom site of the PCB.

6 Contact

You have more questions? Please contact:

Dr. Florian Herrmann		
Physikalisches Institut	Room:	706
Universitaet Freiburg	Phone:	(+49)761-203-5877
Hermann-Herder-Str. 3	Fax:	(+49)761-203-5705
79104 Freiburg	Email:	florian.herrmann@cern.ch