SRS Slow Control Manual

https://espace.cern.ch/rd51-wg5/srs/Documentation/SRS_Slow_Control_Manual.pdf

Contents

List of tables

1 Overview

The slow-control of the SRS system is carried out using UDP over IP protocol on the available Gigabit Ethernet port of the FEC cards. When using a SRU unit to bundle many FEC cards together, the SRU will act as a packet switch, forwarding the slow-control frames to the FEC cards via the DTC links.

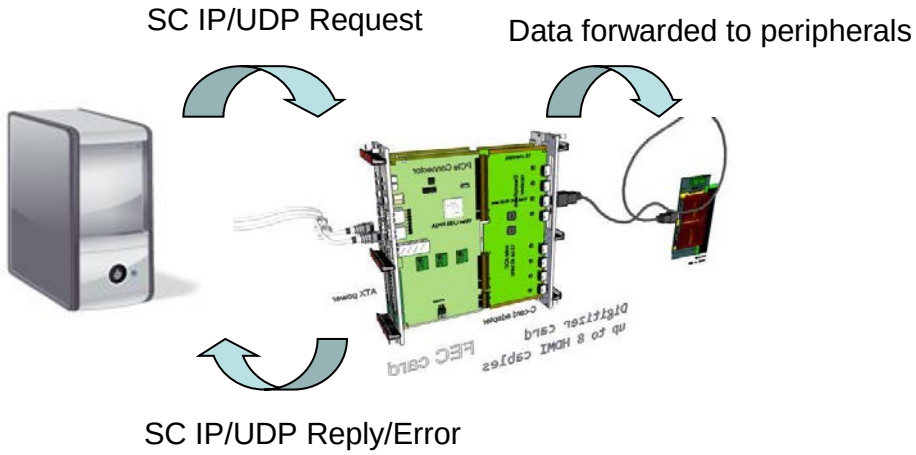

Figure 1. Overview of the SRS slow-control flow

The components of the slow-control system are: the slow-control PC (SC-PC), the network (point-to-point connection/network switch/SRU), the FEC card and the peripherals that need to be configured. Peripherals can be either virtual devices (usually residing in the FEC firmware) or real hardware objects which are connected to the FEC FPGA, located on the FEC card, the A/B/C-Module Card or on the front-end hybrids. Generally the real peripherals have a logic interface located in the FEC firmware, which translates the slow-control commands in the format that the external device understands. The slow control protocol assures that, from the user point of view, the real or virtual attribute of a peripheral is transparent.

The slow-control transactions use a request/reply protocol between the SC-PC and the peripherals. The network and the FEC card guarantee the communication between the two. In particular the FEC Card can filter out ill-formed requests and issue error response packets. The reply packets are generated by the peripheral logic and travel back to the IP address which generated the request.

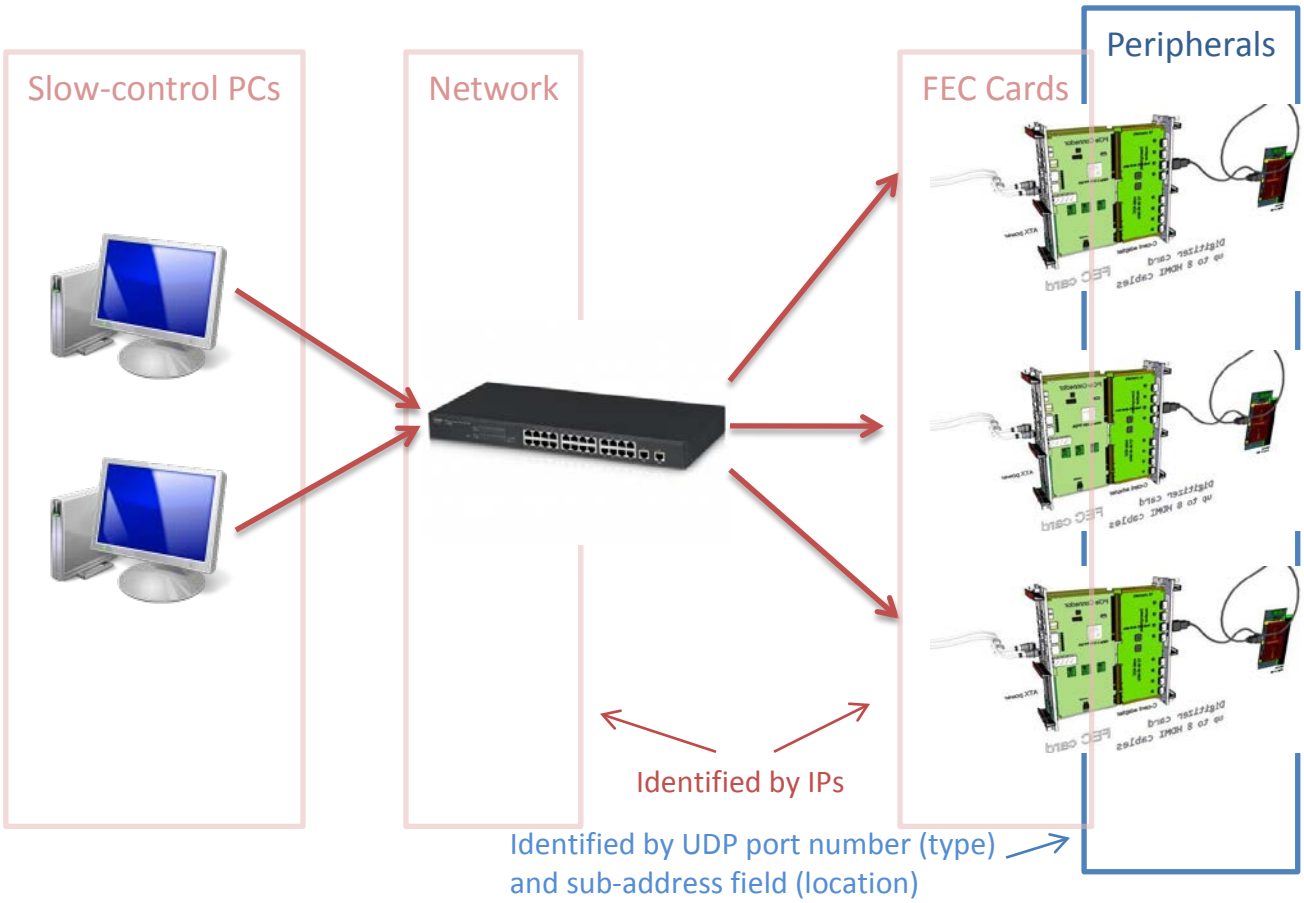

Figure 2. SRS slow-control components

All SC nodes on the network (FECs and SC clients) have an unique IP address and MAC address used to identify them using the Ethernet/IP infrastructure. SC peripherals behave like services residing on individual FEC cards. They are, therefore, identified by 3 numbers:

1. The UDP port number which identifies the *type of peripheral* (eg. System Registers, ADC Card registers, APV Hybrid, etc.)
2. The IP address of the FEC which hosts the individual peripheral;
3. The sub-address field (part of the SC frame format, see next chapter for details) which identifies the location of the peripheral within the host FEC card.

The SC client can address a specific peripheral by sending SC requests either directly to the IP address of the FEC card which hosts the peripheral, or using a broadcast address, always using the UDP port assigned to the peripheral type. If the FEC addressed either by an individual IP address or by a broadcast address does not host a peripheral identified by the specific UDP port address, the action will have no consequences on the register space hosted by the specific FEC card. By default, the FEC card will reply with an error reply. This behavior can be disabled, to avoid unnecessary traffic.

2 Slow-control Format

2.1 SC Request

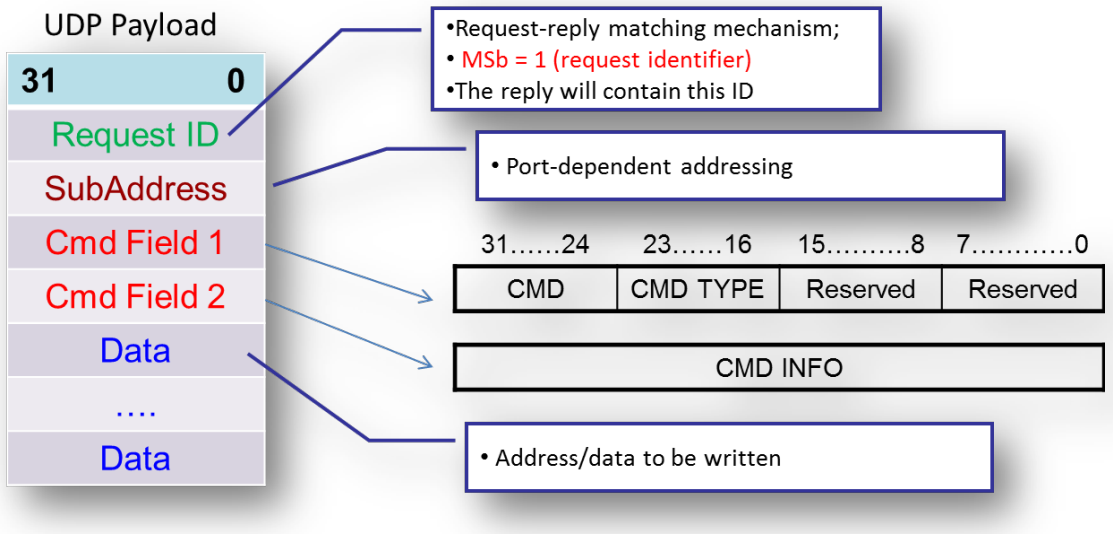

Figure 3. SC request format

SC Request fields:

- **Request ID**: The SC client has to set this field with a unique request identifier. MSb must be set to 1 to signal a request (valid values: 0x80000000 – 0xFFFFFFFF). The reply will contain the same Request ID with the MSb set to 0.
- **SubAddress**: This field is used to identify the location of the peripheral to be programmed. The syntax of this field is defined differently for each peripheral type (port). Peripherals without multiplicity will ignore this field.
- **CMD Field 1**: This field determines the type of request (read, write, etc) See table below for details.
- **CMD Field 2**: Additional data required by the CMD Field (see below).
- **DATA fields**: For write commands these can be either an array of valid data to be written in a write burst or a succession of address and data fields for a write pairs command. For read commands this fields contain either dummy data (read burst) or a list of addresses (read list).

	Byte 3	Byte 2	Byte 1	Byte 0
CMD FIELD 1:	CMD	CMD TYPE	CMD_LENGTH*	
CMD FIELD 2:	CMD INFO			

* reserved. Set to 0xFFFF

Table 1. SC commands

Command	CMD	CMD TYPE	CMD INFO	Description
Write Pairs	0xAA	0xAA	Don't care	Command is followed by address & data pairs
Write Burst	0x AA	0xBB	First address to write to	Followed by data fields to be written at consecutive addresses starting with the address in CMD INFO field
Read Burst	0xBB	0xBB	First address to be read	Followed by dummy data fields. The peripheral will read data for as many consecutive registers as dummy fields are in the request.
Read List	0xBB	0xAA	Don't care	Followed by a list of addresses to be read
Reset	-	-		Not yet implemented

2.2 SC Reply

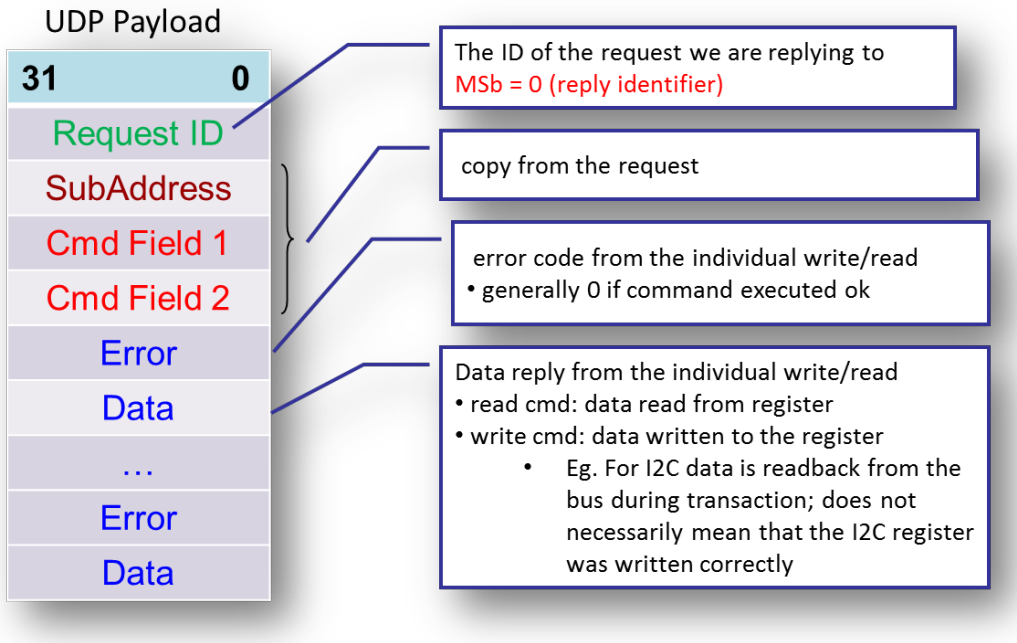

Figure 4. SC Reply Format

SC Reply fields:

- **Request ID:** copy of the corresponding request ID. MSb is set to 0.
- **SubAddress:** copy from the request

- **CMD Field 1&2:** copy from the request
- **Error:** error code generated for the read or write operation on the corresponding register. The syntax of this field depends on the peripheral. Generally a value of 0 indicates no error. For I2C peripherals, for instance, the acknowledge bits are shifted into this field.
- **Data:** data that has been written to the register (write) or data read from the register (read).

Error and Data fields are repeated for as many registers are read or written.

2.3 SC Error

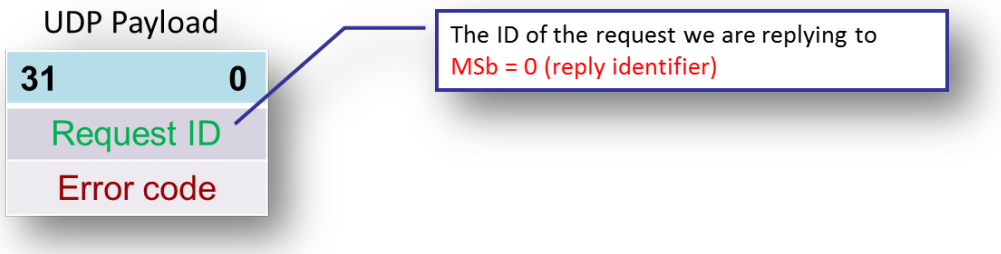

Figure 5. SC Error reply format

Frame receiver errors: fatal errors causing frame to be dropped

- [bit 31] – destination port unavailable
- [bit 30] – illegal source port (≠ 6007 – default SC source port; the value can be changed in the system port)
- [bit 29] – buffer full
- [bit 28] - illegal length (incomplete 32-bit word)
- [bit 27] – illegal length (< 4 words)
- [bit 26] – reply id error

Frame decoder errors: fatal errors causing frame to be dropped

- [bit 19] - command unrecognized
- [bit 18] - illformed command
- [bit 16] - checksum error

3 Peripherals

Table 2. List of available peripherals

Name	Port (hex)	Port (dec)	Use	I/F type	Description	User level	
SYS_PORT	1777	6007	runtime	reg	System registers. Dynamic control of IP address, MAC address, GbE parameters, ...	expert	SYSTEM
FEC_BI2C_PORT	1787	6023	debug setup	I2C	Access to the FEC I2C line B. Used to program the on-board EEPROM	expert	
FEC_AI2C_PORT	1788	6024	debug setup	I2C	Access to the FEC I2C line A. Used to read/program the A/C-card EEPROM and access the monitoring devices (voltage, current, temperature)	expert	
APVAPP_PORT	1797	6039	runtime	reg	APV Application registers. APV trigger sequencer and event builder	user	APV Application
APZRAM_PORT	1798	6040	runtime	RAM	Access to the pedestal and sigma calibration memory of the APZ firmware	user	
APV_PORT	1877	6263	runtime	I2C	Access to the hybrid I2C bus: APV and PLL registers	user	
ADCCARD_PORT	1977	6519	runtime	I2C	I2C registers of the ADC CCARD. (power, equalization, ch reset, clk enable)	user	ADC C-Card
ADCCTRL_PORT	1978	6520	runtime setup	SPI	Access to the control registers of the ADC on the ADC C-Card. Write-only	expert	

3.1 System Registers (SYS_PORT = 6007)

Subaddress : not used (anything)

Name	Address (hex)	Byte count	Description
VERSION	00	2	Firmware version identifier. Reserved
FPGAMAC_VENDORID	01	3	Local MAC address, vendor identifier part.
FPGAMAC_ID	02	3	Local MAC address, device identifier part.
FPGA_IP	03	4	Local (FEC) IP address
DAQPORT	04	2	UDP port for data transfer
SCPORT	05	2	UDP port for slow-control
FRAMEDLY	06	2	Delay between UDP frames.
TOTFRAMES	07	2	DATE flow-control parameter. *Experimental*
ETHMODE	08	2	Ethernet control register. Reserved
SCMODE	09	2	Slow-control control register. Reserved
DAQ_IP	0A	4	DAQ destination IP.
DTCC CTRL	0B	4	Control register for the DTCC link (only with DTC fw)
MCLK_SEL	0C	1	Main clock selection register
MCLK_STATUS	0D	4	Main clock status register (read/only)
RESERVED	0E		
VERSION_HW	0F	2	FW version register (hardwired – read-only)
SYS_RSTREG	FFFFFFFF	FFFF8000 FFFF0001	Reboot FEC Warm-init

Table 3. System registers

FPGAMAC_VENDORID, **FPGAMAC_ID**, **FPGA_IP**: local MAC and IP address of the FEC card. The values are read from the EEPROM at startup. Writing to this registers will have immediate effect, but the FEC will revert to the EEPROM values after reboot. Default IP address is 10.0.0.2
DAQPORT: destination UDP port used by the FEC card to transmit DAQ frames. Default is 6006.
SCPORT: UDP port used for slow-control transactions. If this value is changed, all port numbers assigned to all peripherals will scale accordingly. This port number is also used for the System Registers port. Default is 6007.
DAQ_IP: the IP address the FEC will send DAQ data to. Default is 10.0.0.3.
SYS_RSTREG: register used to send reset commands. Setting bit 15 will reboot the FPGA. Setting bit 0 will initiate a warm initialization of the card. Upon warm-init, all registers will revert to default values (i.e. System registers will be read from EEPROM)

Default practice with DATE is to define separate LAN network segments for each FEC card in the system, therefore the IP addresses are set as follows

FPGA_IP = *10.0.x.2*
DAQ_IP = *10.0.x.3*

where x is from 0 to the total number of FECs (minus 1).

MCLK_SEL: Selects the source of the main (40MHz) clock. There are 3 possible sources:

1) DTC clock. One of the FEC DTC ports can be connected to a CTF card, which distributes clock and trigger signals. The source of this clock is controlled by the DTC_SWAPLANES and DTC_SWAPPORTS registers.
2) Ethernet clock. The clock recovered from the RX path of the Gigabit Ethernet link can be used to synthesize a 40MHz clock for the application unit. In principle, if more FEC cards are connected to the same network switch, this provides a way to synchronize all cards, if the network switch uses the same reference clock for all ports. The synchronization status depends on the mode of operation of the switch itself.
3) Local clock oscillator. Each FEC has a local oscillator which is the default source for the application clock.

The default behavior is to automatically switch from the local oscillator to the DTC clock , when the latter is present. The clock present at the DTC has to be within 10000 ppm of the local oscillator clock and stable for at least 0.5 seconds. MCLK_SEL register can inhibit the DTC clock detection or force the use of the Ethernet recovered clock. Each time the clock source is changed, an internal state-machine triggers a reset of the application unit.

Bit	31 - 8	7	6	5	4	3	2	1	0
Name	RESERVED	ETHCLK_SEL	RESERVED	RESERVED	DTCTRG_INVERT	DTC_SWAPLANES	DTC_SWAPPORTS	DTCTRG_INH	DTCCLK_INH

Table 4. MCLK_SEL bit map

Bit	Name	Description
0	**DTCCLK_INH**	DTC clock inhibit. Set this bit to disable the use of the DTC clock
1	**DTCTRG_INH**	DTC trigger inhibit. Set this bit to disable the use of the DTC trigger
2	**DTC_SWAPPORTS**	Swap DTC ports. Set this bit to use the top DTC port (J1) for CTF connection. By default, the bottom DTC port (J2) is used.
3	**DTC_SWAPLANES**	Swap the clock and trigger inputs on the DTC connector. Used for debugging. Do not set for normal operation
4	**DTCTRG_INVERT**	Invert the polarity of the trigger
5 - 6	reserved	
7	**ETHCLK_SEL**	Force the use of the Ethernet clock as Main application clock.
31..8	RESERVED	

Table 5. MCLK_SEL description

Note. *It is always recommendable to disconnect the CTF cable from the FEC card when not used. A dangling cable may temporarily pick up a clock signal through EMI and trigger erratic reset of the application unit.*

Note. *The APV chips may lose sync when the application clock is switched from one source to another, therefore it is recommendable to issue a APV sync pulse, or even a full initialization of the APV hybrids.*

MCLK_STATUS: Reports the status of the main clock state machine.

Bit	31 - 16	15 -8	7 – 6	5 - 4	3 - 2	1	0
Name	**DTCCLK_MEASURE**			**MCLK_SELECTION**		**ETHCLK_LOCKED**	**DTC0CLK_LOCKED**
Descr.	An estimate of the DTC clock frequency wrt. the local oscillator normalized to 10000 (1 unit = 100 ppm).	reserved	reserved	Clock selection decision. 00 = local 01 = DTC 10 = EthRX 11 = invalid	reserved	Indicates a stable clock is recovered from the Ethernet port.	Indicates a stable clock is present at DTC.

Table 6. MCLK_STATUS bitmap

ETHCLK_SEL	DTC0CLK_INH	ETHCLK_LOCKED	DTC0CLK_LOCKED	Main Clock Source
0	0	x	0	LOCAL
0	0	x	1	DTC
0	1	x	x	LOCAL
1	x	0	x	LOCAL
1	x	1	x	ETH

Table 7. Clock selection truth table

DTCC_CTRL: Control register for the DTCC link. This register is available only with DTC firmware.

Bit	Name	Description
31 - 24	**DTC_TRAILERBYTE**	Byte repeated in the trailer on the DTC data channel
23 – 16	**DTC_PADDINGBYTE**	Padding byte added at the end of a data frame for aligning to a specific word boundary
15 - 12	**DTC_TRAILERCNT**	Number of trailer words (32 bit) at the end of a data event. If set to 0, no trailer is generated.
11	**DTC_TRGIDALL**	Send TRGID (64 bit) at the beginning of each data frame (if DTC_TRGIDENABLE is 1). By default TRGID is sent only with the first frame.
10	**DTC_TRGIDENABLE**	Send TRGID (64 bit) in the data channel
9 - 8	**DTC_PADDINGTYPE**	Padding boundary: 0 – none *1 – 16 bit* 2 – 32 bit *3 – 64 bit (reserved)* *Only 32 bit padding is implemented*
7..2	RESERVED	
1	**DTC_NOFLOWCTRL**	Disable the flow control between DTC and ETH for the SC channel
0	**DTC_DATAOVERETH**	Send data via the SC channel (Data over DTC over ETH). Data frames will pass through the Ethernet infrastructure in the DCS LAN, and not through the DTC data channel (and the Event Builder in the SRU, S-Link, etc.)

Table 8. DTCC_CTRL register

3.2 APV Application Registers (port 6039)

Subaddress : not used (anything)

Name	Address (hex)	Byte count	default	Access mode	Description	Fw. ver.
APV TRIGGER CONTROL REGISTERS:						
BCLK_MODE	00	1	0x04 (run mode)	RW	Controls the trigger sequencer for the APV. See table below for details	
BCLK_TRGBURST	01	1	4	RW	controls how many time slots the APV chip is reading from its memory for each trigger	
BCLK_FREQ	02	2	40000 (0x9C40)	RW	Period of the trigger sequencer.	
BCLK_TRGDELAY	03	2	256 (0x100)	RW	Delay between the external/internal trigger and the APV trigger	
BCLK_TPDELAY	04	2	128 (0x80)	RW	Delay between the external/internal trigger and the APV test-pulse	
BCLK_ROSYNC	05	2	300 (0x12C)	RW	Delay between the external/internal trigger and the start of data recording	
	06				Reserved	
ADC_STATUS	07	3	0x3FFFF	R	(*debug info)* Status of the ADC deserialization code. Read-only	2.06
EVENTBUILD REGISTERS:						
EVBLD_CHENABLE	08	2	0xFFFF	RW	Channel-enable mask for the data transmission. Even bits are masters and odd bits are slaves. If bit is set, corresponding channel is enabled	
EVBLD_DATALENGTH	09	2	2500	RW	Length of the data capture window	
EVBLD_MODE	0A	1	0	RW	Event Builder mode register. 0 = (default) use frame-of-event counter (8-bit) 1 = use frame-of-run counter (32-bit) 2 = use timestamp(24-bit) and frame-of-event ctr	2.05
EVBLD_EVENTINFOTYPE	0B	1	0	RW	Controls the data format.	
EVBLD_EVENTINFODATA	0C	4	-	RW	Data for the optional info-filed in the data format	
RUN CONTROL:						
RO_ENABLE	0F		0	RW	Readout Enable register (bit 0). Triggers are accepted for acquisition when this bit is 1	2.01
RST_REG	FFFFFFFF			W	Reset register. Bit 0 = APV sync reset	2.02

Table 9. APV application registers

Name	Address (hex)	Byte count	default	Access Mode	Description	Fw. ver.
APZ REGITERS[1]*:*						
APZ_SYNC_DET	10	2	0	R	Presence of the APV sync pulses on each channel. Read-only	APZ
APZ_STATUS	11	4	0x80	R	Status of the APZ processor. Read-only	APZ
APZ_APVSELECT	12	1	0	RW	Selects one APV channel for single channel commands	APZ
APZ_NSAMPLES	13	1	0	RW	Overrides the *number of samples* parameter. If set to 0 (default) the parameter is calculated internally from **BCLK_TRGBURST**.	APZ
APZ_ZEROSUPP_THR	14	2	0	RW	Zero-suppression threshold Byte 0 = fractional thr part (6 bits, msb) Byte 1 = integer thr part (6 bits, lsb)	APZ
APZ_ZEROSUPP_PRMS	15	2	0	RW	Zero-suppression parameters	APZ
	16 – 1C				Reserved	
APV_SYNC_LOWTHR	1D	2	0	RW	Low threshold for the APV sync-pulse detection. If set to 0 (default) the threshold is internally hard wired (*1100)*	APZ
APV_SYNC_HIGHTHR	1E	2	0	RW	High threshold for the APV sync-pulse detection. If set to 0 (default) the threshold is internally hard wired (*3000)*	APZ
APZ_CMD	1F	1	0	RW	Command register for the APZ processor.	APZ

Table 10. APZ (APV with zero suppression code) application registers

[1] These registers are only present in the Zero-suppression (APZ) firmware variant

3.2.1 APV Trigger control registers

BCLK_MODE (address 00) bit description:

Bit	7 - 4	3	2	1	0
Descr.	**reserved**	**TRGIN polarity**	**TRIGGER mode**	**APV Test Pulse**	**APV Reset**
V=0		NIM	Internally generated continuous loop	Test pulse disabled	Disabled. Default for run mode.
V=1		Inverse NIM	External. Controlled by TRGIN	Test pulse enabled	Enabled. Used with the test-pulse. Do not use in run mode

Table 11. BCLK_MODE bit map

Example:

3 (`b00000011`) => continuous loop with test pulse and reset (test mode)

4 (`b00000100`) => triggered externally, no test-pulse, no reset (running mode - acquisition controlled by external trigger)

BCLK_TRGBURST (address 0x01)**:** controls how many time slots the APV chip is reading from its memory for each trigger. The formula is **(n + 1) x 3**. Setting this to 4 means a number of 15 time slots. The maximum is 30 time slots (n = 9).

BCLK_FREQ (address 0x02)**:**

a) (**APVAPP_PORT.BCLK_MODE.TRIGGER_MODE** = 1) When trigger source is set to external this parameter controls the deadtime introduced by the FPGA. After accepting a trigger, the FPGA will ignore all triggers incoming for **Bclk_freq x 64 x 25ns** time. 40000 means 1ms. This time should not be lower than the total acquisition time of one event which is about 222 us (with default parameters).

b) (**APVAPP_PORT.BCLK_MODE.TRIGGER_MODE** = 0) When the trigger source is set to internal this parameter controls the repetition rate of the internal generated trigger.

BCLK_TRGDELAY (address 0x03)**:** The FPGA delays the trigger for this number of clock-cycles (25ns) until propagating it to the chip. When used with the APV25 front-end ASIC, the effective latency of the trigger is the difference between **APV_PORT.APV_LATENCY** and **APVAPP_PORT.BCLK_TRGDELAY.**

3.2.2 Event builder registers

EVBLD_CHENABLE: (address 0x08) – Channel-enable mask for the data transmission. Even bits are masters and odd bits are slaves. If bit is set, corresponding channel is enabled. Channel mapping:

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HDMI channel	7		6		5		4		3		2		1		0	
Master/Slave	S	M	S	M	S	M	S	M	S	M	S	M	S	M	S	M

Table 12. EVBLD_CHENABLE bit map

EVBLD_DATALENGTH (address 0x09) – Length of the data capture window in words (16-bit), in raw data (ADC) mode or bypass mode of the APZ code. Maximum allowed value without exceeding the UDP jumbo frame limit is 4000.

EVBLD_MODE (address 0x0A) – Event Builder mode register. Controls the format of the *FRAME COUNTER* field of the SRS data format.

EVBLD_MODE value	Mode	*FRAME COUNTER field*			
		Byte 3	**Byte 2**	**Byte 1**	**Byte 0**
0	Single-FEC mode (*default)*	0x00	0x00	0x00	**F#**
1	Test Mode	**GF#**			
2	Multiple-FEC mode	**TS**			**F#**

Table 13. Event Builder modes (EVBLD_MODE register)

F# = frame-of-event counter (1 byte): starts from 0 for a new event and increases for every frame of the same event.

GF# = frame-of-run counter: 4-byte global frame counter which starts from 0 at the beginning of the run and and counts continuously (does not reset for each event).

TS = 3-byte timestamp tag attributed to each event.

EVBLD_EVENTINFODATA (address 0x0C) – Controls the content of the HEADER INFO FIELD of the SRS Data Format.

Bit	31 - 16	15 -8	7 - 0
Name	HINFO_LABEL	HINFO_SEL	reserved
Descr.	Run label that can be copied in the most significant bytes of HEADER INFO FIELD of every event	Selects the content of HEADER INFO FIELD (see table)	

Table 14. EVBLD_EVENTINFODATA register

HINFO_SEL	HEADER INFO FIELD			
	31 - 24	24 - 16	15 -8	7 - 0
0x01	TRIGGER COUNTER (2 bytes)		EVBLD_DATALENGTH	
0x02	TRIGGER COUNTER (4 bytes)			
other	HINFO_LABEL		EVBLD_DATALENGTH	

Table 15. Content of HEADER INFO FIELD function of HINFO_SEL

3.2.3 APV Processor and zero-suppression (APZ)

(Register map for the APV zero-suppression FW variant – third revision – beta release)

APZ_CMD (address 0x1F): Command register for the APV processor. *(Note. This register was revised and moved from address 0x11 (for the alpha release) to 0x1F).*This register is used to trigger different calibration routines of the APZ processor. Calibration commands are can be run on single channels (identified by APZ_APVSELECT register) or on multiple channels (enabled channels in EVBLD_CHENABLE register).

***Warning**. When **phase calibration** is run on **multiple channels**, if both MASTER and SLAVE hybrids of the same HDMI slot are enabled in the EVBLD_CHENABLE register, the optimal phase for the MASTER hybrid will be overwritten by the one for the SLAVE (this is due to the fact that the MASTER-SLAVE pair shares a single PLL device). Generally this does not pose a problem, but it may lead to the situation when a faulty SLAVE can disturb the operation of both hybrids.*

CMD	Name	Description	Channel(s) defined by …
0x00	RUN/ABORT	Default run mode. Any other can be aborted by writing 0 to the APZ_CMD register	**EVBLD_CHENABLE**
0x01	CAL_PHASE_SINGLE	Calibrate phase, single channel	**APZ_APVSELECT**
0x02	CAL_PED_SINGLE	Calibrate pedestal (and sigma), single channel	**APZ_APVSELECT**
0x03	CAL_FULL_SINGLE	Calibrate both phase and pedestal (and sigma) values, single channel	**APZ_APVSELECT**
0x0F	BYPASS	Bypass APZ processor. Data is readout in raw format from a single channel	**APZ_APVSELECT**
0x10	CAL_FULL_ALL	Calibrate all channels enabled by EVBLD_CHENABLE (full calibration). Channels are treated sequentially; current channel being treated is displayed in CALIB_ALL_CRT field of APZ_STATUS register. **Return to run mode (APZ_CMD = 0) after this command is compulsory**	**EVBLD_CHENABLE**
0x11	CAL_PHASE_ALL	As above, phase only	**EVBLD_CHENABLE**
0x12	CAL_PED_ALL	As above, pedestal and sigma only	**EVBLD_CHENABLE**
0xFF	RESET	Reset APZ processor	-

Table 16. APZ_CMD table

APZ_SYNC_DET (address 0x10): Detects if an APV front-end ASIC is present at each ADC channel and is correctly configured for 40MHz acquisition[2]. Each of the 16 bits of the register corresponds to one channel, with the same mapping as for the EVBLD_CHENABLE register (see Table 10).

APZ_STATUS (address 0x11): status of the APV processor.

Bit	31 - 16						15 - 12	11 - 8	7	6	5	4	3	2	1	0
Name	CH15	CH14	CH13	.....	CH1	CH0	RESERVED	CALIB_ALL_CRT	APZ_ENABLED	APZ_BYPASS	CMD_DONE	CALIB_ALL_DONE	WATCHDOG_FLAG	PHASE_ALIGNED	PEDCAL_BUSY	PHASECAL_BUSY
	APZ_CHANNEL_STATUS															

Table 17. APZ_STATUS bitmap

Bit	Name	Description
0	PHASECAL_BUSY	Clock-phase calibration running
1	PEDCAL_BUSY	Pedestal calibration running
2	PHASE_ALIGNED	Clock-phase calibration routine completed successfully
3	WATCHDOG_FLAG	Last pedestal calibration terminated by a watchdog reset
4	CALIB_ALL_DONE	"Calibrate All" command finished
5	CMD_DONE	Command finished
6	APZ_BYPASS_N	Bypassing APZ code. Reading out raw data from channel **APZ_APVSELECT**
7	APZ_ENABLED	APZ code enabled. Reading out zero-suppressed data
11..8	CALIB_ALL_CRT	When "Calibrate All" command is active this field indicates the current channel being treated.
31..16	APZ_CHANNEL_STATUS	Indicates channels that were successfully calibrated by either "calibrate all" or "calibrate single" commands. When channels are calibrated one by one the corresponding bit is updated each time "calibrate single" command is executed. All bits are cleared by a "APZ reset" command.

Table 18. APZ_STATUS description

APZ_APVSELECT (address 0x12): Selects the APV for the pedestal calibration, clock-phase calibration ("calibrate single" command), and raw data monitoring (in APZ bypass mode). Valid range: 0 -15.

APZ_NSAMPLES (address 0x13): Tells the APV processor how many time-samples to process for each trigger. If set to 0 the register is set internally using the BCLK_TRGBURST value, using the **(n + 1) x 3** formula. The register is provided to allow the user to limit the acquisition to a specific number of time-samples from the APV output stream. **Note:** *Experimental use only, use with care. If the register is set to a higher number of samples than the APV, the data processor may hang.*

APZ_ZEROSUPP_THR (address 0x14): Optional threshold register for the zero-suppression operation. The value of the register is multiplied with the sigma value of each channel, and the result is used as

[2] See the APV User Manual for more information on the APV's readout modes and synchronisation sequence.

threshold for the zero-suppression. This register is only used when bit 4 (threshold mode) of register **APZ_ZEROSUPP_PRMS.**

Bit	15 - 0			
Name	Zero-suppression threshold (APZ_ZEROSUPP_THR)			
Description	Controls the threshold multiplier for signal detection. The value is given			
	15 - 14	13 - 8	7 - 2	1 - 0
	reserved	Integer part of the threshold	Fractional part of the threshold	reserved

Table 19 APZ_ZEROSUPP_THR bitmap

APZ_ZEROSUPP_PRMS (address 0x15): Configuration register for the zero-suppression unit.

Bit	..5	4	3	2	1	0
Name		Threshold mode	Force signal	Disable pedestal correction		Peak find mode
Descr.	reserved	0 = auto 1 = APZ_ZEROSUPP_THR	No channel is suppressed. All 128 APV channels are acquired in APZ format	Pedestal value is forced to 0 for all channels, therefore uncorrected data is acquired.	reserved	Data is further reduced by acquiring only the peak sample and its relative sample position.

Table 20. APZ_ZEROSUPP_PRMS bit map

3.3 Pedestal and pedestal sigma calibration memory (APZ fw – APZRAM_PORT = 6040)

Subaddress: LSB = APV selector (0-15)

This port allows read and write access to the calibration memory holding the values of the pedestal and pedestal variations (sigma) corresponding to each APV channel, present in the zero-suppression (APZ) firmware variant. The calibration routines store the result of this calibration in this memory. The values can be read or overwritten using this port.

Address. The most significant bit of the address field is used as a sigma(1)/pedestal(0) selector. The lest significant byte of the address field contains the APV channel position in transmission order

Bit	31	30 .. 8	7 .. 0
Descr	0 = pedestal 1 = sigma	reserved	APV channel number (0- 127)

Table 21. APZRAM address bit map

The correspondence between the physical APV channel number and the transmission order is given by the following formula (for more details see the APV User Manual):

$$Channel\ No. = 32 * (n\ MOD\ 4) + 8 * INT(n / 4) - 31 * INT(n / 16)$$

Data. Unsigned integer value for the pedestal or the sigma**.** The effective number of bits in the data field is 12.

	Address	Description
Pedestals	0x00000000	Pedestal channel ord. 0
	0x00000001	Pedestal channel ord. 1
	0x00000002	Pedestal channel ord. 2
	…	
	0x0000007F	Pedestal channel ord. 127
	…	reserved
Sigma	0x80000000	Pedestal variation (sigma) channel ord. 0
	0x80000001	Pedestal variation (sigma) channel ord. 1
	0x80000002	Pedestal variation (sigma) channel ord. 2
	…	
	0x8000007F	Pedestal variation (sigma) channel ord. 127

Table 22. APZRAM address mapping

3.4 APV Hybrid Registers (APV_PORT = 6263)

Subaddress:

31 - 24	23 - 16	15 - 8	7 - 0
XX	XX	Channel mask	Device

HDMI channel mapping:

4	5	6	7	0	1	2	3

R	R	R	R	R	R	D	D

R – Reserved
D – Device selector

PLL	0	0
Master APV	0	1
Slave APV	1	0
Both APVs	1	1

Example: 0xXXXXFF03 => all APVs
0xXXXXFF00 => all PLLs
(X = anything(hex))

Device	Register	r/w	internal address (I2C)[3]	address (hex)	Default value[4]	Default value (hex)[5]	Power-on value	Description
APV	ERROR	r	0000000x	00			0	SEU or Sync error
APV	MODE	r/w	0000001x	01	b00011001	19	4	See table below
APV	LATENCY	r/w	0000010x	02	128	80	132	Trigger latency
APV	MUXGAIN	r/w	0000011x	03	b00000100	04	4	Gain of the output buffer. One-hot value
APV	IPRE	r/w	0010000x	10	98	62	0	Preamplifier current
APV	IPCASC	r/w	0010001x	11	52	34	0	
APV	IPSF	r/w	0010010x	12	34	22	0	
APV	ISHA	r/w	0010011x	13	34	22	0	
APV	ISSF	r/w	0010100x	14	34	22	0	
APV	IPSP	r/w	0010101x	15	55	37	0	
APV	IMUXIN	r/w	0010110x	16	16	10	0	Output buffer pedestal control
APV	ICAL	r/w	0011000x	18	100	64	0	Calibration pulse strength
APV	VPSP	r/w	0011011x	19	40	28	0	
APV	VFS	r/w	0011010x	1A	60	3C	0	
APV	VFP	r/w	0011001x	1B	30	1E	0	
APV	CDRV	r/w	0011100x	1C	b11101111	EF	0	Calibration channel mask.
APV	CSEL	r/w	0011101x	1D	b11110111	F7	0	Calibration fine phase (3.125ns). One-hot value

Table 23. APV registers

[3] I2C binary address. This number is given only to facilitate the cross-reference with the APV User Manual
[4] Recommended values for normal operation. Users are encouraged to use the APV User Manual and other APV related literature for further reference
[5] See above

Device	Register	r/w	address (hex)	Default value	Default value (hex)	Power-on value	Description
PLL	CSR1_FINEDELAY	r/w	01	b00100000	20	0	Bit 3-0: CLK fine-phase adjustment (value ≤ 11) Bit 4: CLK phase flip Bit 5: enables access to TRG_DELAY register
PLL	TRG_DELAY	r/w	03	0	0	0	Trigger delay (clock cycles)

Table 24. Relevant PLL25 registers

APV_MODE register description (see APV User Manual for more information):

Bit number	Function	Value = 0	Value = 1
7	Not Used	-	-
6	Not Used	-	-
5	Preamp Polarity	Non-Inverting	Inverting
4	Read-out Frequency	20MHz	40MHz
3	Read-out Mode	Deconvolution	Peak
2	Calibration Inhibit	OFF	ON
1	Trigger Mode	3-sample	1-sample
0	Analogue Bias	OFF	ON

Table 25. APV_MODE bit map

Example:

`00011001 (hex: 19)` => 40 MHz, peak-mode, 3 samples-per-trigger with calibration pulse (test mode)

`00011101 (hex: 1D)` => same, without calibration pulse (running mode)

For more information on the APV registers please refer to the APV User Manual.

3.5 ADC C-Card: ADCCARD registers (ADCCARD_PORT = 6519)

Subaddress: not used (anything)

Name	Address (hex)	Byte count	Description	default	Power-on value
HYBRID_RST_N	00	1	Reset pin for each HDMI channel. Valid low for the APV hybrid.	xFF	xFF
PWRDOWN_CH0	01	1	Power-down control of the analog circuitry for the ***master*** path for each HDMI channel	x00	xFF
PWRDOWN_CH1	02	1	Power-down control of the analog circuitry for the ***slave*** path for each HDMI channel	x00	xFF
EQ_LEVEL_0	03	1	Equalization control (bit 0) for each HDMI channel	x00	xFF
EQ_LEVEL_1	04	1	Equalization control (bit 1) for each HDMI channel	x00	xFF
TRGOUT_ENABLE	05	1	Enables TRGOUT buffer for each HDMI channel	x00	xFF
BCLK_ENABLE	06	1	Enables BCLK buffer for each HDMI channel	xFF	xFF

Register Bit:	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Corresponding HDMI channel:	4	5	6	7	0	1	2	3

Table 26. ADCCARD registers

3.6 ADC C-Card: ADC SPI control

Subaddress:

31 - 24	23 - 16	15 - 8	7 - 0							
XX	XX	XX	Device							
Reserved			R	R	R	R	R	R	A	A
			R – Reserved A – ADC selector							
			None - invalid						0	0
			ADC 0 (ch 0 – 7)						0	1
			ADC 1 (ch 8 – 15)						1	0
			Both ADCs (ch 0 - 15)						1	1

The two ADCs on the ADC C-Card can be controlled using the registers listed in the ADC datasheet (http://www.ti.com/lit/ds/symlink/ads5281.pdf). The interface to the ADCs is write-only, so data cannot be read.

Address Field: the 8-bit register address corresponding to the ADC register (see ADS5281 datasheet)
Data Field: 16-bit data field

Some of the ADC registers control the ADC channels independently. The mapping between the ADC channel number (as in the ADCs datasheet) and the system channel (as recorded by the system, eg. APV channel, etc) is as follows:

	ADC0								ADC1							
ADC channel:	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
System channel:	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8
Corresponding HDMI plug:	3	3	2	2	1	1	0	0	7	7	6	6	5	5	4	4
HDMI sub-channel (M/S)	S	M	S	M	S	M	S	M	S	M	S	M	S	M	S	M

Table 27. Channel mapping on the 2 ADC on the ADC C-Card.

3.6.1 Extract of the TI - ADS5281 datasheet

SERIAL REGISTER MAP[6]

SUMMARY OF FUNCTIONS SUPPORTED BY SERIAL INTERFACE(1)(2)(3)(4)

ADDRESS IN HEX	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	NAME	DESCRIPTION	DEFAULT
00																X	RST	Self-clearing software RESET.	Inactive
0F									X	X	X	X	X	X	X	X	PDN_CH<8:1>	Channel-specific ADC power-down mode.	Inactive
								X									PDN_PARTIAL	Partial power-down mode (fast recovery from power-down).	Inactive
							X										PDN_COMPLETE	Register mode for complete power-down (slower recovery).	Inactive
						X											PDN_PIN_CFG	Configures the PD pin for partial power-down mode.	Complete power-down
11														X	X	X	ILVDS_LCLK<2:0>	LVDS current drive programmability for $LCLK_N$ and $LCLK_P$ pins.	3.5mA drive
										X	X	X					ILVDS_FRAME <2:0>	LVDS current drive programmability for $ADCLK_N$ and $ADCLK_P$ pins.	3.5mA drive
						X	X	X									ILVDS_DAT<2:0>	LVDS current drive programmability for OUT_N and OUT_P pins.	3.5mA drive
12		X															EN_LVDS_TERM	Enables internal termination for LVDS buffers.	Termination disabled
		1												X	X	X	TERM_LCLK<2:0>	Programmable termination for $LCLK_N$ and $LCLK_P$ buffers.	Termination disabled
		1								X	X	X					TERM_FRAME <2:0>	Programmable termination for $ADCLK_N$ and $ADCLK_P$ buffers.	Termination disabled
		1				X	X	X									TERM_DAT<2:0>	Programmable termination for OUT_N and OUT_P buffers.	Termination disabled
14									X	X	X	X	X	X	X	X	LFNS_CH<8:1>	Channel-specific, low-frequency noise suppression mode enable.	Inactive
24									X	X	X	X	X	X	X	X	INVERT_CH<8:1>	Swaps the polarity of the analog input pins electrically.	IN_P is positive input
25										X	0	0					EN_RAMP	Enables a repeating full-scale ramp pattern on the outputs.	Inactive
										0	X	0					DUALCUSTOM_ PAT	Enables the mode wherein the output toggles between two defined codes.	Inactive
										0	0	X					SINGLE_CUSTOM _PAT	Enables the mode wherein the output is a constant specified code.	Inactive
26															X	X	BITS_CUSTOM1 <11:10>	2MSBs for a single custom pattern (and for the first code of the dual custom pattern). <11> is the MSB.	Inactive
													X	X			BITS_CUSTOM2 <11:10>	2MSBs for the second code of the dual custom pattern.	Inactive
	X	X	X	X	X	X	X	X	X	X							BITS_CUSTOM1 <9:0>	10 lower bits for the single custom pattern (and for the first code of the dual custom pattern). <0> is the LSB.	Inactive
27	X	X	X	X	X	X	X	X	X	X							BITS_CUSTOM2 <9:0>	10 lower bits for the second code of the dual custom pattern.	Inactive
2A													X	X	X	X	GAIN_CH1<3:0>	Programmable gain channel 1.	0dB gain
									X	X	X	X					GAIN_CH2<3:0>	Programmable gain channel 2.	0dB gain
					X	X	X	X									GAIN_CH3<3:0>	Programmable gain channel 3.	0dB gain
	X	X	X	X													GAIN_CH4<3:0>	Programmable gain channel 4.	0dB gain
2B	X	X	X	X													GAIN_CH5<3:0>	Programmable gain channel 5.	0dB gain
					X	X	X	X									GAIN_CH6<3:0>	Programmable gain channel 6.	0dB gain
									X	X	X	X					GAIN_CH7<3:0>	Programmable gain channel 7.	0dB gain
													X	X	X	X	GAIN_CH8<3:0>	Programmable gain channel 8.	0dB gain

(1) The unused bits in each register (identified as blank table cells) must be programmed as '0'.
(2) X = Register bit referenced by the corresponding name and description (default is 0).

[6] Registers in the shaded sections of the table should not be used.

(3) Bits marked as '0' should be forced to 0, and bits marked as '1' should be forced to 1 when the particular register is programmed.
(4) Multiple functions in a register should be programmed in a single write operation.

POWER-DOWN MODES

ADDRESS IN HEX	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	NAME
0F									X	X	X	X	X	X	X	X	PDN_CH<8:1>
								X									PDN_PARTIAL
						0	X										PDN_COMPLETE
						X	0										PDN_PIN_CFG

Each of the eight channels can be individually powered down. PDN_CH<N> controls the power-down mode for the ADC channel <N>.

In addition to channel-specific power-down, the ADS528x also has two global power-down modes—partial power-down mode and complete power-down mode. Partial power-down mode partially powers down the chip; recovery from this mode is much quicker, provided that the clock has been running for at least 50 s before exiting this mode. Complete power-down mode, on the other hand, completely powers down the chip, and involves a much longer recovery time.

In addition to programming the device for either of these two power-down modes (through either the PDN_PARTIAL or PDN_COMPLETE bits, respectively), the PD pin itself can be configured as either a partial power-down pin or a complete power-down pin control. For example, if PDN_PIN_CFG = 0 (default), when the PD pin is high, the device enters complete power-down mode. However, if PDN_PIN_CFG = 1, when the PD pin is high, the device enters partial power-down mode.

LOW-FREQUENCY NOISE SUPPRESSION MODE

ADDRESS IN HEX	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	NAME
14									X	X	X	X	X	X	X	X	LFNS_CH<8:1>

The low-frequency noise suppression mode is specifically useful in applications where good noise performance is desired in the frequency band of 0MHz to 1MHz (around dc). Setting this mode shifts the low-frequency noise of the ADS528x to approximately $f_S/2$, thereby moving the noise floor around dc to a much lower value. LFNS_CH<8:1> enables this mode individually for each channel.

ANALOG INPUT INVERT

ADDRESS IN HEX	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	NAME
24									X	X	X	X	X	X	X	X	INVERT_CH<8:1>

Normally, the IN_P pin represents the positive analog input pin, and IN_N represents the complementary negative input. Setting the bits marked INVERT_CH<8:1> (individual control for each channel) causes the inputs to be swapped. IN_N now represents the positive input, and IN_P the negative input.

PROGRAMMABLE GAIN

ADDRESS IN HEX	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	NAME
2A													X	X	X	X	GAIN_CH1<3:0>
									X	X	X	X					GAIN_CH2<3:0>
					X	X	X	X									GAIN_CH3<3:0>
	X	X	X	X													GAIN_CH4<3:0>
2B	X	X	X	X													GAIN_CH5<3:0>
					X	X	X	X									GAIN_CH6<3:0>
									X	X	X	X					GAIN_CH7<3:0>
													X	X	X	X	GAIN_CH8<3:0>

In applications where the full-scale swing of the analog input signal is much less than the $2V_{PP}$ range supported by the ADS528x, a programmable gain can be set to achieve the full-scale output code even with a lower analog input swing. The programmable gain not only fills the output code range of the ADC, but also enhances the SNR of the device by utilizing quantization information from some extra

internal bits. The programmable gain for each channel can be individually set using a set of four bits, indicated as GAIN_CHN<3:0> for Channel N. The gain setting is coded in binary from 0dB to 12dB, as shown in Table 6.

Table 6. Gain Setting for Channel 1

GAIN_CH1<3>	GAIN_CH1<2>	GAIN_CH1<1>	GAIN_CH1<0>	CHANNEL 1 GAIN SETTING
0	0	0	0	0dB
0	0	0	1	1dB
0	0	1	0	2dB
0	0	1	1	3dB
0	1	0	0	4dB
0	1	0	1	5dB
0	1	1	0	6dB
0	1	1	1	7dB
1	0	0	0	8dB
1	0	0	1	9dB
1	0	1	0	10dB
1	0	1	1	11dB
1	1	0	0	12dB
1	1	0	1	Do not use
1	1	1	0	Do not use
1	1	1	1	Do not use

4 Appendix A. slow_control Linux program

Usage:

```
./slow_control file.txt
```

Contents of “file.txt”:

```
#Destination (FEC) IP address
10.0.0.2
#Peripheral port number
6039
#SC-Request Frame Data in 4-byte hex format
##Request ID (MSb = 1)
80000000
##Subaddress
00000000
##Command (Write pairs)
AAAAFFFF
##Command info (don’t care for Write pairs)
00000000
##Register address
00000000
##Data to be written
00000004
##Register address
00000001
##Data to be written
00000004
…
```

5 Appendix B. SDC (Scalable Detector Control) software.

Refer to https://twiki.cern.ch/twiki/bin/view/AtlasPublic/SDC