SALT ver 4 chip documentation

Miroslaw Firlej, Tomasz Fiutowski, Marek Idzik, Jakub Moron, Krzysztof Swientek
Szymon Bugiel, Roma Dasgupta, Marika Kuczynska

Created: November 2016
Last change: May 29, 2018
Document version 1.2

Contents

1. SALT overview 4

2. Analogue front-end 5
   2.1. Architecture .................................................. 5
   2.2. Test pulse generation .......................................... 7

3. Analog to Digital Converter 7

4. Common mode signals and biasing DACs 9
   4.1. Generation of common mode voltages ......................... 9
   4.2. Biasing DACs .................................................... 10

5. Clock generation 11
   5.1. Delay Locked-Loop (DLL) .................................... 11
       5.1.1. DLL configuration procedure - semi-automatic ............ 13
       5.1.2. DLL configuration/test remarks .......................... 13
   5.2. Phase Locked-Loop (PLL) .................................... 14
       5.2.1. PLL start-up ............................................... 15
       5.2.2. PLL configuration/test remarks .......................... 15

6. Digital Signal Processing 16
   6.1. Synchronisation between ADC and DSP ....................... 16
   6.2. Pedestal subtraction .......................................... 17
   6.3. Mean Common Mode Subtraction ............................... 18
6.4. Zero Suppression ............................................. 20
6.5. NZS data ....................................................... 21

7. Back-end data processing ..................................... 22
  7.1. Packet building and recording .............................. 22
  7.2. Idle creation and data stream splitting ................... 24

8. I/Os – pads and interfaces .................................... 26
  8.1. Reset circuitry ............................................... 26
  8.2. Communication interfaces ................................. 26
    8.2.1. Slow I²C interface ..................................... 27
    8.2.2. Fast data interface with DDR serializer ................ 30
    8.2.3. Fast TFC interface with DDR deserializer .......... 31
  8.3. I/O pads characterisation .................................. 32
    8.3.1. CMOS pads ............................................. 33
    8.3.2. SLVS transmitter ....................................... 34
    8.3.3. SLVS receiver .......................................... 35

9. Test and monitoring ............................................ 38
  9.1. Monitoring for analogue part ............................. 38
  9.2. Digital data processing monitoring ....................... 38
  9.3. Debugging circuitry ....................................... 39

10. Rad-hard protection ........................................ 40
    10.1. Total Ionizing Dose protection ......................... 40
    10.2. Single Event Upset protection and monitoring ...... 40

11. Padding ......................................................... 41

12. Optical Alignment Markers .................................. 53

A. ASIC registers ............................................... 55
    A.1. Global registers ......................................... 56
    A.2. Analogue and mixed-mode registers ..................... 57
    A.3. DSP registers ........................................... 65
    A.4. Memory registers and counters ......................... 79
    A.5. Serializer registers ..................................... 81
    A.6. Other configuration registers .......................... 83
    A.7. TFC counters ............................................ 89

B. Pseudo-random sequences .................................... 93

C. Changes in comparison to first 128-channel version .... 101

D. Changelog ..................................................... 102
Documentation Conventions

To aid the readers understanding, a consistent formatting style has been used throughout this manual.

- Internal signals are written using italic font.
- External connections names (pads) like supplies use CAPITAL LETTERS only.
- External signals names, however, are in capital letters but using ITALIC font also.
- Configuration elements like register names are written in sans serif font.
- Signals controlled by configuration bits use slanted sans serif font.

For numbers the Verilog prefix style is used:

- ’b for binary numbers e.g ’b1010,
- ’h for hexadecimal numbers e.g ’hA7,
- ’d for decimal numbers e.g ’d72,
- no prefix means that the number is in decimal notation.

There are also still some C style prefixes 0x for hexadecimal numbers, especially in I²C figures.
1. SALT overview

Silicon strip detectors in the upgraded Tracker of LHCb experiment requires a new readout Application Specific Circuit (ASIC). A project of a 128-channel ASIC called SALT (Silicon ASIC for LHCb Tracking), has been started. A block diagram of the first 128-channel SALT prototype (called just SALT) is shown in figure 1. SALT extracts and digitises analogue signals from the sensor, performs the digital processing and transmits the serial output data. It is designed in TSMC CMOS 130 nm technology, and uses a novel architecture comprising an analogue front-end and an ultra-low power (< 0.5 mW) fast (40 MSps) sampling 6-bit ADC in each channel.

The front-end comprises a charge preamplifier and a fast shaper ($T_{peak} = 25$ ns and fast recovery) required to distinguish between the LHC bunch crossings at 40 MHz. To achieve this a specific non-standard shaper is required. The front-end should work with sensor capacitances between 1.6 – 12 pF. In the last stage of analogue front-end a single-to-differential block converts a single-ended signal to a differential one. A fully differential 6-bit SAR ADC running at 40 MHz converts the analogue signal to the digital domain. In order to achieve highest speed and lowest power consumption the SAR logic is asynchronous and dynamic circuitry is used in the ADC logic and comparator. To synchronize ADC sampling instance with beam collisions a dedicated ultra-low power (< 1 mW) DLL is used to shift and align an external clock. The digital ADC output is processed by a Digital Signal Processing (DSP) block, which performs a pedestal subtraction, a mean common mode subtraction and a zero suppression. In following step the data packets are created and recorded in a local memory. After the DSP the data are serialized and the data rate is multiplied eight times. This is obtained by increasing the clock frequency four times (to 160 MHz) and by using a double data rate (DDR) transmission. An ultra-low power (< 1 mW) Phase Locked-Loop (PLL) is used to generate the 160 MHz clock from the 40 MHz system clock. The data is sent out by...
the SLVS interface. The ASIC is controlled via the LHCb common protocol consisting of two interfaces: the Timing and Fast Control (TFC) and the Experiment Control System (ECS) \[1, 2\]. The TFC interface delivers the 40 MHz clock and other crucial information and commands, synchronised with the experiment clock, while the ECS serves to configure and monitor the ASIC and it is realised through the Inter-Integrated Circuit (I²C) interface.

The specifications and the functionality of the full 128-channel SALT ASIC are almost the same as the previous 8-channel SALT ASICs. The differences are caused either by different floorplan of these chips, or by using not a final version of certain blocks in the 128-channel prototype of Silicon ASIC for LHCb Tracking (SALT) ASIC, or by the modifications requested after the SALT ASIC tests.

The main specifications of the SALT ASIC are shown in Table 1.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technology</td>
<td>TSMC CMOS 130 nm</td>
</tr>
<tr>
<td>Channels per ASIC</td>
<td>128</td>
</tr>
<tr>
<td>Input / Output pitch</td>
<td>80 $\mu$m / 140 $\mu$m</td>
</tr>
<tr>
<td>Total power dissipation</td>
<td>&lt; 768 mW</td>
</tr>
<tr>
<td>Radiation hardness</td>
<td>30 Mrad</td>
</tr>
<tr>
<td>Sensor input capacitance</td>
<td>1.6 – 12 pF</td>
</tr>
<tr>
<td>Noise</td>
<td>$\sim$1000 e$^{-}$ @10 pF + 50e$^{-}$ /pF</td>
</tr>
<tr>
<td>Maximum cross-talk</td>
<td>Less than 5% between channels</td>
</tr>
<tr>
<td>Signal polarity</td>
<td>Both electron and hole collection</td>
</tr>
<tr>
<td>Dynamic range</td>
<td>Input charge up to $\sim$30 000 e$^{-}$</td>
</tr>
<tr>
<td>Linearity</td>
<td>Within 5% over dynamic range</td>
</tr>
<tr>
<td>Pulse shape and tail</td>
<td>$T_{\text{peak}} \sim$25 ns, tail after 2$\times$$T_{\text{peak}} \sim$5% amplitude</td>
</tr>
<tr>
<td>Gain uniformity</td>
<td>Uniformity across channels within $\sim$5%</td>
</tr>
<tr>
<td>ADC bits</td>
<td>6 bits (5 bits for each polarity)</td>
</tr>
<tr>
<td>ADC sampling rate</td>
<td>40 MHz</td>
</tr>
<tr>
<td>DSP functions</td>
<td>Pedestal and common mode subtraction, zero suppression</td>
</tr>
<tr>
<td>Output formats</td>
<td>Non-zero suppressed, zero suppressed</td>
</tr>
<tr>
<td>Calibration modes</td>
<td>Analogue test pulses, digital data loading</td>
</tr>
<tr>
<td>Output serialiser</td>
<td>Three serial e-links, at 320 MBit/s</td>
</tr>
<tr>
<td>Slow controls interface</td>
<td>I²C</td>
</tr>
<tr>
<td>Digital signals, data, TFC, clock interface</td>
<td>Differential, SLVS</td>
</tr>
</tbody>
</table>

Table 1: Summary of the specifications of the SALT ASIC.

2. Analogue front-end

2.1. Architecture

The analogue front-end has to be very fast with a peaking time $\leq 25$ ns, should have a very short tail to minimise the pile-up and spill-over into the next bunch crossing, and
should also have very low power consumption (1–2 mW/channel). It should work with different strip sensors (capacitance range 5–20 pF), with input signal of both polarities and with good enough signal to noise ratio (S/N > 10), even in the worst operation conditions. One of the main challenges for the analogue block is to obtain a very short signal duration with a minimum possible power consumption. Preliminary studies showed that this is not possible with a standard semi-Gaussian shaping (with real poles in the transfer function), but that a more complex shaping (using complex poles and zeros in the transfer function) is required. A simplified block diagram of the preamplifier and shaper fulfilling the requested requirements is shown in figure 2.

![Simplified block diagram of preamplifier and shaper.](image)

Figure 2: Simplified block diagram of preamplifier and shaper.

Each channel contains an 8-bit trimming DAC for a precise baseline setting. The $v_{cm}$ reference voltage should be set to half power supply. At the input of each channel there is a small (100 fF) test capacitance. The inputs of every second capacitance are shorted together so the injection of test pulse can be done either to all odd or even channels through a two separate test pads. The single-ended output is sent to a single-to-differential converter translating a single-ended signal to differential. In the Krummenacher circuitry and baseline Digital to Analog Converter (DAC) the Enclosed Layout Transistors (ELT) NMOS transistors were implemented in current mirrors working with very small currents.

The analog front-end has four common parameters which are adjusted through internal DACs. These are: preamplifier bias current $I_{pre}$; preamplifier feedback bias current $I_{krum}$; shaper bias current $I_{sh}$; and single-to-differential converter bias current $I_{s2d}$. The configuration of these DACs, together with eight DACs for independent setting of each channel baseline, is described later in table 16.

The SALT analog front-end SALT contains:

- 128 channels with analog processing chain including preamplifier, shaper and single-to-differential converter
- Two additional test channels (one on each side) which are discussed later in Test and Monitoring section
2.2. Test pulse generation

A test pulse (called also calibration pulse) functionality allows to check the whole processing chain in the SALT chip without radiation or even sensor. A user is able to generate a pulse of known charge in well defined time instance at the front-end input to simulate the signal from a sensor. The current pulse from sensor in first approximation has a shape of Dirac’s delta. To create such current pulse a voltage step is differentiated on test capacitor at the preamplifier test input (see figure 2 in page 6). The input charge $Q$ can be calculated according to formula

$$Q_{in} = C_{test} V_{test},$$

where test capacitor $C_{test} = 100 \text{ fF}$, and voltage step height $V_{test}$ is controlled by calib_volt_cfg register (see table 20) via a 6-bit calib DAC. Voltage step height range is 0 – 170 mV which corresponds to $Q_{in}$ range 0 – 17 fC.

The test pulse is generated when SALT chip receives Calib TFC command. Since this pulse has to appear at the test input at the same time as the pulse from the sensor so it should be precisely delayed relative to the adc_clk clock. However, its delay is different from the delay of ADC sampling signal adc_clk because the latency from digital part to the front-end input is different from the latency to the ADC input. As a result the internal SALT DLL has separate clock output calib_clk with delay controlled by configuration register calib_clk_cfg (see table 20). Remark – global TFC delay of calib_tcf signal if different then of other TFC commands (see deserializer description, section 8.2.3).

Calibration pulse is only fired by Calib TFC command. Its width, polarity and height are controlled by configuration registers and are defined in circuit shown in figure 3. The first part of calibration circuit is synchroniser which moves calib_tfc from MAIN_CLK domain to calib_clk domain. In the next step pulse width is set in the stretcher (controlled by calib_pulse_len signal), then polarity is defined via calib_inv signal. The formed pulse is sent only to channels selected by calena signal. The test pulse generator located at the input of front-end preamplifier converts digital pulse to analogue pulse with the height defined by vcal voltage controlled via Calibration DAC (see calib_volt_cfg register in table 20).

In fact the described circuitry generates two charge pulses at the input of preamplifier, first positive and second negative (or vice versa), which is different from sensor behaviour producing only a single polarity pulse. The second pulse appears at the falling edge (or rising, depending on the pulse polarity set) of calibration pulse and should be distant enough to not overlap with the first one. It is worth to note that only the first pulse is well defined in time while the second has significant jitter.

3. Analog to Digital Converter

According to the SALT specifications a 6-bit resolution Analog to Digital Converter (ADC) with 40 MHz sampling rate is needed for the signal digitisation. One of the most
Figure 3: Test pulse generation; for configuration elements in ECS registers see table 20; "test pulse generator" is a part of analogue front-end channel.

important constraints of the ADC design is a very low power consumption, significantly below 1 mW. This requirement has naturally led to a Successive Approximation Register (SAR) ADC architecture. In order to further reduce the dissipated power, a Merged Capacitor Switching (MCS) scheme [3] was chosen. Since in the given technology the best capacitance matching is offered by a Metal-Insulator-Metal (MIM) capacitors, which are relatively large, a split capacitor DAC approach was used, as shown in Fig. 4. In order to increase the linearity of the ADC the input switches are bootstrapped, what reduces significantly their dynamic resistance. In addition to the DAC switching scheme two other features are implemented to reduce the power consumption: a dynamic comparator and an asynchronous control logic, as shown in Fig. 4. The dynamic comparator dissipates power only during the bit cycling process, while the main power saving in the asynchronous control logic comes from the fact that the fast clock signal for the conversion of subsequent bits is eliminated. Finally, some more power is saved by implementing part of the logic with a dynamic flip-flops. To improve the ADC robustness against Single Event Effect (SEE) events the ADC is reset by the sampling signal if the previous conversion is not yet completed. In practice it means that for SEE event two ADC conversions are lost (one caused by SEE and next when the reset is done), but after that the ADC is ready for the next conversion.

The ADC operation is controlled by 4-bit configuration value composed of:

- settling delay \textit{set\_del} (3-bit), which allows to modify the internal delay between the bit cycling and so to tune the ADC performance. The default value \texttt{b100} should ensure the proper effective resolution across all expected environment conditions, however due to the process variations, longer or shorter delay may be needed, depending on the change of propagation time in control logic. Faster logic will perform the bit cycling too fast, and so it will require longer settling delay. Slower logic, otherwise, will result in longer conversion, which may reduce the sampling
phase length below necessary minimum. This can be corrected by shorter settling delay. Since the settling delay determines the overall conversion length, changing it may affect the synchronisation between ADC and DSP (see section 6.1 for details);

• **rnd_counter** (1-bit), which enables the pseudo-random counter at the output of the ADC in order to test the synchronisation between ADC and DSP.

The same 4-bit configuration value, set by **ana.g.cfg** configuration register described later in table 16, is shared by all ADCs in the whole ASIC.

The SALT ASIC contains:

• 128 ADCs connected to the outputs of analog front-end (single-to-differential converters),
• two additional ADCs connected to two test channels (discussed later in section Test and Monitoring),
• four additional ADCs used for monitoring as discussed later in section Test and Monitoring.

### 4. Common mode signals and biasing DACs

#### 4.1. Generation of common mode voltages

Both the front-end and ADC use several common mode voltage signals. All these signals have the same voltage value of about 0.6V. Because the common mode signals have critical effect on the quality of signal processing four common voltages are generated locally in each channel: **vcm_a**, **vcm_b**, **vcm_c**, and **vcm_d** (copies of these signals taken from the test channel may be observed on pads **VCMA**, **VCMB**, **VCMC**). Voltages **vcm_a**, **vcm_b**, and **vcm_c** are generated by passing the reference current **I_{VCM}** through diode-connected NMOS transistor, while the **vcm_d** is generated from simple resistor divider. The reference current **I_{VCM}** is generated globally and defined by a dedicated DAC.
• Signal $v_{cm,a}$ — delivers common mode voltage to the first two stages of the shaper (see figure 2).
• Signal $v_{cm,b}$ — delivers common mode voltage to the third stage of the shaper (see figure 2).
• Signal $v_{cm,c}$ — delivers common mode voltage to the single-to-differential converter (see figure 2).
• Signal $v_{cm,d}$ — delivers common mode voltage to the ADC (see figure 4). Since this signal is very “dirty” and its precision is not important it is generated by resistive divider from VDDD power supply.

4.2. Biasing DACs

There are eight global DACs biasing various SALT blocks:

• 5-bit DAC setting the preamplifier bias current $I_{\text{pre}}$. Typical bias current is 25 $\mu$A for default DAC setting equal 15. Changing the DAC setting in the range 0–31 the current changes linearly in the range 0–50 $\mu$A.

• 5-bit DAC setting the Krummenacher circuit bias current $I_{\text{krum}}$. Typical bias current is 6 nA for default DAC setting equal 4. Changing the DAC setting in the range 0–31 the current changes linearly in the range 0–46 nA.

• 5-bit DAC setting the shaper bias current $I_{\text{sh}}$. Typical bias current is 3 $\mu$A for default DAC setting equal 12. Changing the DAC setting in the range 0–31 the current changes linearly in the range 0–8.2 $\mu$A.

• 5-bit DAC setting the single-to-differential converter bias current $I_{\text{s2d}}$. Typical bias current is 3 $\mu$A for default DAC setting equal 12. Changing the DAC setting in the range 0–31 the current changes linearly in the range 0–8.2 $\mu$A.

• 6-bit DAC setting the test pulse amplitude $V_{\text{cal}}$. Typical signal amplitude is 37 mV for default DAC setting equal 14. Changing the DAC setting in the range 0–63 the amplitude changes linearly in the range 0–170 mV.

• 6-bit DAC setting the reference current $I_{\text{VCM}}$ for generation of common voltages. Typical reference current is 65 $\mu$A for default DAC setting equal 32. Changing the DAC setting in the range 0–63 the current changes linearly in the range 8–121 $\mu$A.

• 5-bit DAC setting the SLVS bias current $I_{\text{slvs}}$. Typical bias current is 215 $\mu$A for default DAC setting equal 13. Changing the DAC setting in the range 0–31 the current changes linearly in the range 0–500 $\mu$A.
• 5-bit DAC setting the SLVS bias common mode voltage $V_{slvs}$. Typical DAC voltage is 400 $\mu$V for default DAC setting equal 10. Changing the DAC setting in the range 0–31 its voltage changes linearly in the range 0.1–1.1 V. This values do not correspond linearly to the SLVS common voltage, which changes in the range 0.1–0.4 V.

As already discussed, except these general purpose DACs, in each channel of the SALT there is a 8-bit current DAC which is used to set the front-end baseline voltage. Each channel can be set separately. Typical current is 0 $\mu$A for default DAC setting equal 128. Changing the DAC setting in the range 0–255 the current changes linearly from $-4.5 \mu$A to 4.5 $\mu$A.

All biasing DACs are summarized in table 2. The configuration of the above DACs is described in table 16.

<table>
<thead>
<tr>
<th>Name</th>
<th>Resolution</th>
<th>Range</th>
<th>LSB value</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I_{pre}$</td>
<td>5-bits</td>
<td>0–50 $\mu$A</td>
<td>1.59 $\mu$A</td>
<td>15 LSB</td>
</tr>
<tr>
<td>$I_{trim}$</td>
<td>5-bits</td>
<td>0–46 nA</td>
<td>1.47 nA</td>
<td>4 LSB</td>
</tr>
<tr>
<td>$I_{sh}$</td>
<td>5-bits</td>
<td>0–8.2 $\mu$A</td>
<td>267 nA</td>
<td>12 LSB</td>
</tr>
<tr>
<td>$I_{s2d}$</td>
<td>5-bits</td>
<td>0–8.2 $\mu$A</td>
<td>267 nA</td>
<td>12 LSB</td>
</tr>
<tr>
<td>$V_{cal}$</td>
<td>6-bits</td>
<td>0–170 mV</td>
<td>2.66 mV</td>
<td>14 LSB</td>
</tr>
<tr>
<td>$I_{VCM}$</td>
<td>6-bits</td>
<td>8–121 $\mu$A</td>
<td>1.8 $\mu$A</td>
<td>32 LSB</td>
</tr>
<tr>
<td>$I_{slvs}$</td>
<td>5-bits</td>
<td>0–500 $\mu$A</td>
<td>16.1 $\mu$A</td>
<td>13 LSB</td>
</tr>
<tr>
<td>$V_{slvs}$</td>
<td>5-bits</td>
<td>0.1–1.1 V</td>
<td>32.3 mV</td>
<td>10 LSB</td>
</tr>
<tr>
<td>$I_{trim}$</td>
<td>8-bits</td>
<td>$-4.5 \mu$A–4.5 $\mu$A</td>
<td>35 nA</td>
<td>128 LSB</td>
</tr>
</tbody>
</table>

Table 2: Biasing DACs

5. Clock generation

The SALT receives a 40 MHz external clock, which is used in digital part of the chip. As already mentioned the 40 MHz clock is precisely shifted/aligned, by a dedicated internal Delay Locked-Loop (DLL) block, to obtain the ADC sampling signal and to generate the test pulse. A fast 160 MHz clock for data serialisation and transmission is obtained by multiplication of the 40 MHz clock by a dedicated internal PLL block. Both DLL and PLL blocks are described in the following sections.

5.1. Delay Locked-Loop (DLL)

Figure 5 shows the block diagram of the DLL, which was designed to adjust clock phase. The VCDL contains 64 delay cells and the same number of independent clock phases is available at the DLL outputs. The circuit is equipped with two internal multiplexers, which provides the clock phase selection (2 from 64 phases). The VCDL delay can
be controlled by changing the bias current via an internal 7-bit DAC, which allows to compensate the bias current changes caused by technology corner, supply voltage, temperature, etc.. The CP current can be configured in similar way via another DAC. The DLL also contains the clock multiplexers, which provide clock phase or source selection. When the `dll_connect` is set to 0 (see figure 5), the input clock is connected directly to the outputs (bypass). The selection of the clock phase is possible only when `dll_connect` is set to 1. This functionality is created to avoid randomly changing clock phase on the DLL output during the synchronization process, which may be disadvantageous for the digital logic.

The DLL is always enabled, but the synchronisation requires special start-up procedure. When the `dll_start` is set to 0, the circuit is in standby mode. This mode should last at least 10µs and after that time a high state on `dll_start` starts the DLL synchronisation process. To better control DLL operation a voltage monitoring was added to VCDL control voltage $V_{\text{VCDL}}$. When $V_{\text{VCDL}}$ is near half of power supply voltage $V_{\text{vdd}/2}$ (600 ± 100 mV) the DLL operates in optimal conditions. This operation can by changed by changing `dll_vcdl_cfg` register, which sets the VCDL bias current. To simplify DLL configuration procedure the Current Test (CT) logic was added. This block is enabled only when `dll_start` is set to 0 and allows to verify whether the selected VCDL current is proper for the circuit. The difference between $V_{\text{DLL}}$ and $V_{\text{vdd}/2}$ voltage (field `dll_vcdl_voltage`) and CT logic output `dll_cur_ok` can be read from `dll_vcdl_mon` register (more details will be described later).

The DLL consists of Harmonic Lock Prevention block (inside PD) which prevents...
DLL from locking at total delays 2, 3, etc. higher than reference period (25 ns). This functionality is turned on by default, but it can be turned off by setting **dll_hlp** to 0.

### 5.1.1. DLL configuration procedure - semi-automatic

Semi-automatic DLL configuration procedure is based on CT logic and can be applied in the following steps.

1. Set the **dll_vcdl_cfg** registers to start value 96 (‘h60). For such setting of VCDL current the delay of all 64 DLL stages should be much smaller than 25 ns (for 40 MHz reference).

2. Wait at least 1 µs (probably slow control interface provides this delay).

3. Read the **dll_cur_ok** bit from **dll_vcdl_mon** register. It should be 0, otherwise increase the start value in register **dll_vcdl_cfg** and check again until **dll_cur_ok** will be 0.

4. Decrease **dll_vcdl_cfg** value by 1.

5. Wait at least 1 µs.

6. Read the **dll_cur_ok** bit from **dll_vcdl_mon** register. If it is 0 then return to step 4, otherwise the value in **dll_vcdl_cfg** is optimal for the configured ASIC. Note that different ASICs will have probably different optimal values in **dll_vcdl_cfg** register.

7. When the optimal value in **dll_vcdl_cfg** is set the **dll_start** has to be switched to 1 to perform the DLL synchronisation process.

### 5.1.2. DLL configuration/test remarks

1. The **dll_cp_cfg** register allows to set CP current. Its default value should be: ‘h1A, and should not be changed.

2. The VCDL current, sets by 7-bit DAC, is crucial for correct DLL operation. The default value ‘h39 in **dll_vcdl_cfg** register should be fine in the most of cases. For such setting the total delay of all 64 DLL stages is around 25 ns (for 40 MHz reference). The current can be adjusted if needed – higher values in **dll_vcdl_cur** field of the **dll_vcdl_cfg** register means shorter delay and vice versa.

3. To verify the proper DLL operation a difference between VCDL control voltage $V_{DLL}$ and half of power supply $V_{vdd/2}$ (optimal value) may be checked using **dll_vcdl_mon** register. When DLL operates in optimal condition the $V_{DLL}$ should be near $V_{vdd/2}$ voltage and **dll_vcdl_voltage** field in **dll_vcdl_mon** register should be small, around 0. The $V_{DLL}$ value near ground (large negative value in **dll_vcdl_voltage**) means that DLL total delay is too short and lock at 40 MHz reference (25 ns delay) is not possible. In this situation the VCDL bias current should be decreased until
In opposite case, when $V_{DLL}$ is near power supply voltage (large positive value in $dll_{vcdl\_voltage}$), the DLL total delay is too long so VCDL bias current should be increased.

### 5.2. Phase Locked-Loop (PLL)

Figure 6 shows the PLL, which is used for clock multiplication in the data serialization circuit. The architecture of the circuit is typical and commonly described in literature, but few improvements were implemented. The DAC circuits were added for the better control of bias currents. The Voltage Controlled Oscillator (VCO) provides the gain selection (1 of 4), which can be useful for the jitter optimization (if necessary). This feature was added to make the PLL a general purpose block, although in the SALT the maximum gain will be always used. The PLL can work in frequency range 80 MHz – 400 MHz and is characterized by very low power consumption (1.0 mW @ 320 MHz).

The VCO centre frequency is controlled directly by an internal 7-bit DAC, which allows to compensate the bias current changes caused by technology corner, supply voltage, temperature, etc.. The PLL works with constant division factor (by 4) and generates 16 clock phases at its outputs. Two internal multiplexers provide a clock phase selection (2 from 16 phases). To better control PLL operation a voltage monitoring was added to VCO control voltage $V_{PLL}$. When $V_{PLL}$ is around half of power supply voltage $V_{vdd}/2$ (600 ± 100 mV) the PLL operates in optimal condition. This condition can be changed by using the $pll_{vco\_cfg}$ register, which allows to change the VCO bias current. The difference between $V_{PLL}$ and $V_{vdd}/2$ voltage represented as a 6-bit signed number $pll_{vco\_voltage}$ may be read from $pll_{vco\_mon}$ register (see table 20 on page 83).
In fact, because of triplication of digital clock tree, there are three PLLs in the SALT. From the user point of view it should be transparent because all configuration registers are connected to all of these PLLs, so setting e.g. `pll.enable` enables all three PLLs. The only exception of this rule is the PLL monitoring function where the observed PLL has to be chosen (see register `mon_cfg` register in table 20). It is expected that there is no difference or the differences are small between PLLs when observing $V_{PLL}$ signal. (The last statement may need some update after measurements.)

5.2.1. PLL start-up

The PLL start-up procedure is as follows:

1. turn on PLL setting up `pll.enable` bit in `pll.main_cfg` register (see table 19),
2. check all three $V_{PLL}$ and correct them if needed,
3. connect PLL to the serializer by setting up `pll.connect` bit.

Step 2 may be omitted (not recommended) but steps 1 and 3 cannot be performed in single I2C transaction! The PLL needs some time after turn on to lock-up. It has to be locked before is connected. Next I2C transaction duration is enough.

5.2.2. PLL configuration/test remarks

1. Make sure that PLL is enabled and gain is configured by `pll.main_cfg` register to maximum value `pll.gain = 'b11`. It sets maximum VCO gain.

2. The `pll.cp_cfg` register allows to set CP current. Its default value should be: 'h1A, and should not be changed.

3. The `pll.vco_cfg` register allows to change the VCO bias current which is directly related to the output frequency. Default current value is: 'h1A (optimum from simulations for 160 MHz output). It should be changed when PLL has synchronization problems i.e. monitor register `pll.vco_mon` shows values far from zero (see next point).

4. The best method to verify the proper PLL operation is to check difference between VCO control voltage $V_{PLL}$ and $V_{vdd/2}$ (optimal value) using `pll.vco_mon` register. When PLL operates in optimal condition the $V_{PLL}$ should be near $V_{vdd/2}$ voltage and `pll.vco_voltage` field in `pll.vco_mon` register should be small, around 0. The $V_{PLL}$ value near ground (large negative value in `pll.vco_voltage`) means that PLL generate to high frequency and lock at 160 MHz output is not possible. In this situation the VCO bias current should be decreased (`pll.vco_cfg` register) until $V_{PLL}$ obtains proper value. In opposite case, when $V_{PLL}$ is near power supply voltage (large positive value in `pll.vco_voltage`) the frequency is to low so VCO bias current should be increased.
6. Digital Signal Processing

As it was already mentioned the DSP circuitry performs a pedestal subtraction, a mean common mode subtraction, and a zero suppression on the 6-bit data-stream coming from the ADC. The ADC samples are signed 6-bit numbers coded as 2’s complements. Before any processing these samples have to be transported to the DSP clock domain which is formally the first step of DSP. For a better testability the DSP allows also to transmit a raw ADC data or various combinations of partially processed data. A block diagram of the ASIC with the DSP components included is presented in figure 7.

![Block Diagram](image)

Figure 7: Data processing in the ASIC; vertical lines separate clock domains; FE – analogue front-end, ADC – Analog to Digital Converter, Ped&MCM – Pedestal and Mean Common Mode Subtractions, ZS – Zero Suppression, PCK – Packet Formation, RAM – Main data memory, IDL – Idle packets generator, Ser – Serialisers (just before e-link drivers), PLL – Phase Locked-Loop, DLL – Delay Locked-Loop.

In the DSP calculation in each place were subtraction is performed a saturation arithmetic is used. The subtraction with saturation operates here on numbers inside the range from $-32$ to $31$. If the result of the subtraction should be greater than the maximum, it is set to the maximum; if it should be below the minimum, it is the minimum value.

In the following sub-sections the detailed information about the DSP processing is given i.e. synchronisation, pedestal subtraction, mean common mode calculation and subtraction and zero suppression. These three operation (and synchronisation) form the core DSP processing. Other operations shown in figure 7 performed after core DSP i.e.: packet formation, buffering and data splitting for e-links are called back-end data processing and are described in the next section.

6.1. Synchronisation between ADC and DSP

The DLL creates a phase shift between $adc_{clk}$ and $MAIN_{CLK}$ clocks. As a result data from the ADC have to synchronized to DSP clock domain which is done by a circuit shown in figure 8. The $adc_{sync_{sel}}$ signal, located in $pack_{adc_{sync_{cfg}}}$ register (see table 17), select if the raw data $adc_{raw}$ from ADC are sampled on rising or falling clock edge of the main clock.
Correct setting of $\text{adc-sync\_sel}$ bit for different DLL delays (controlled by $\text{adc-clk\_sel}$ field in $\text{adc-clk\_cfg}$ register) is shown in table 3. The same table contains also formulas for calculation of time shift for different DLL delay and proper synchronisation.

<table>
<thead>
<tr>
<th>$\text{adc-clk_sel}$</th>
<th>$\text{adc-sync_sel}$</th>
<th>time shift [ns]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 – 7</td>
<td>1 (falling edge)</td>
<td>$\text{adc-clk_sel} \cdot \frac{25}{64} + 25$</td>
</tr>
<tr>
<td>8 – 39</td>
<td>0 (rising edge)</td>
<td>$\text{adc-clk_sel} \cdot \frac{25}{64}$</td>
</tr>
<tr>
<td>40 – 63</td>
<td>1 (falling edge)</td>
<td>$\text{adc-clk_sel} \cdot \frac{25}{64}$</td>
</tr>
</tbody>
</table>

Table 3: Correct $\text{adc-sync\_sel}$ bit setting for different DLL delays

The synchronisation can be check using pseudo-random counter located at the ADC output, which can be turn on by setting $\text{rnd\_counter}$ bit in $\text{ana\_cfg}$ register (see table 16). The sequence of this pseudo random counter starts from zero and is calculated according the formula (Verilog HDL code)

$$\text{cnt} <= \{\text{cnt[4:0]}, \text{cnt[5]} \wedge \text{cnt[4]} \wedge (\text{cnt[4]} \mid \text{cnt[3]} \mid \text{cnt[2]} \mid \text{cnt[1]} \mid \text{cnt[0]})\};$$

assuming that $\text{cnt}$ is an unsigned 6-bit variable declared as $\text{reg [5:0] cnt}$.

### 6.2. Pedestal subtraction

The DSP processing starts from the pedestal subtraction, which block diagram is shown in figure 9. The global signals (same for all channels) $\text{const\_value}$ and $\text{is\_const}$ are for testing purposes and in normal operation $\text{is\_const}$ should be zero. The signal $\text{invert}$, also global, turns on the arithmetic inversion of input data. It is necessary if ASIC receives the signal of negative polarity (for $n^+ - n$ silicon sensor), when the ADC sample value is expected to be negative. When the ASIC is connected to $p^+ - n$ sensor the signal $\text{invert}$ is zero. This operation is shown in the formula below:

$$\text{adc\_inv[i]} = \begin{cases} 
\text{adc\_sync[i]} & \text{invert} = 0 \\
\text{adc\_sync[i]} & \text{invert} = 1
\end{cases}$$

(2)
where \( \text{adc-sync}[i] \) is input sample from synchronizing circuitry and \( i \) is the channel number from 0 to 7.

After the inversion a pedestal subtraction with saturation is performed. The data after pedestal subtraction is sent to the next block only if the signal \( \text{mask} \), for a particular channel, is zero, as expressed in the formula below:

\[
\text{adc_ped}[i] = \begin{cases} 
0 & \text{mask}[i] = 1 \\
-32 & \text{mask}[i] = 0 \text{ and } \text{adc_inv}[i] - \text{ped}[i] < -32 \\
\text{adc_inv}[i] - \text{ped}[i] & \text{mask}[i] = 0 \text{ and } -32 \leq \text{adc_inv}[i] - \text{ped}[i] \leq 31 \\
31 & \text{mask}[i] = 0 \text{ and } \text{adc_inv}[i] - \text{ped}[i] > 31
\end{cases}
\]

(3)

The \( \text{mask} \) signal is intended to mask noisy or dead channels. In fact, there is one more multiplexer (not shown in the figure 9 but see figure 12), controlled by \( \text{mask} \), at the input of the pedestal block, which helps to reduce the power consumption by avoiding not needed switching activity.

All the configuration signals used in this block are controlled by the DSP configuration registers described in table 17 (page 65).

### 6.3. Mean Common Mode Subtraction

The idea of Mean Common Mode Subtraction (MCMS) block operation is to calculate an average value of all channels without a hit and subtract this value from all channels. In ideal case, when there are no disturbances and the pedestals from previous step are correct, the calculated average should be zero. We assume here that a possible disturbance is identical in each channel what, in general case, is an idealisation. A block diagram of the MCMS algorithm in SALT ASIC is shown in figure 10.

The channels with and without hit are distinguished by a comparison to a threshold. There are two thresholds in the MCMS block: \( \text{mcm_th} \) and \( \text{mcm_th2} \), with different functions. The main threshold \( \text{mcms_th} \) selects the channels which should be used for the MCMS algorithm. The second threshold is to reject channels which values are extremely low – this idea comes from experience with the Beetle ASIC working presently in the LHCb experiment.
Figure 10: Mean Common Mode Subtraction block; elements in frame are repeated in each channel; elements outside the frame are single in the whole ASIC.

Before the subtraction can be applied the number of non active channels \( mcm\_channels \) (channels without a hit) and the sum over non active channels have to be calculated. Both calculations are implemented as hierarchical pipelined structures and in both cases the operation takes two clock cycles. The next step is to calculate the average value. To perform this a division sub-block was designed. It is also implemented as a pipeline and the whole operation takes five clock cycles. As a result of the division the \( mcm\_value \) signal is produced.

Mathematically the whole process can be described as:

\[
mcm\_channels = \sum_{i=0}^{7} mcm\_ch[i], \quad (4)
\]

where

\[
mcm\_ch[i] = \begin{cases} 0 & \text{mask}[i] = 1 \text{ or } mcm\_th2 > \text{adc\_ped}[i] \text{ or } \text{adc\_ped}[i] \geq mcm\_th \\ 1 & \text{mask}[i] = 0 \text{ and } mcm\_th2 \leq \text{adc\_ped}[i] < mcm\_th \end{cases} \quad (5)
\]

The values \( \text{adc\_ped}[i] \) are the results of pedestal subtraction (formula 3), while \( mcm\_th \) and \( mcm\_th2 \) are the thresholds kept in configuration registers described in table 17.

A mean common mode value \( mcm\_value \) (6-bit signed value) is evaluated as:

\[
mcm\_value = \left[ \frac{1}{mcm\_channels} \sum_{i=0}^{7} mcm\_ch[i] \cdot \text{adc\_ped}[i] \right] \text{round}. \quad (6)
\]

To keep the precision the result of division is rounded to the nearest integer instead of being truncated.

To have the ability to deactivate both thresholds the inequalities in formula 5 (and figure 10) are slightly different for \( mcm\_th \) and \( mcm\_th2 \). Deactivation of \( mcm\_th2 \) is
possible by setting it to $-2^5$ (the smallest possible number), in such case non channel will be rejected by this threshold. The \texttt{mcm\_th} also needs to be set to $-2^5$ for deactivation, and in such case there is not channel smaller than this value, and so the MCMS module is inactive since both \texttt{mcm\_channels} and \texttt{mcm\_value} are zero.

The very last step of MCMS is the subtraction with saturation and channels masking. To keep the data consistency there is a FIFO of depth 7 located just before the subtraction block, see figure 10. The depth of this FIFO is calculated as a sum of latencies of sum over channels and division operations. The masking have the same role as in pedestal subtraction block. As the result at the output of MCMS block the samples have following values:

$$
adc\_mcm[i] = \begin{cases} 
0 & \text{mask}[i] = 1 \\
-32 & \text{mask}[i] = 0 \text{ and } \text{adc\_ped}[i] - \text{mcm\_value} < -32 \\
\text{adc\_ped}[i] - \text{mcm\_value} & \text{mask}[i] = 0 \text{ and } -32 <= \text{adc\_ped}[i] - \text{mcm\_value} <= 31 \\
31 & \text{mask}[i] = 0 \text{ and } \text{adc\_ped}[i] - \text{mcm\_value} > 31
\end{cases}
$$

(7)

where $i$ is the channel number, and \texttt{mcm\_value} is calculated with equation 6.

It is worth to mention that both \texttt{mcm\_channels} and \texttt{mcm\_value} values are sent to the packet building block to be used as a part of the Non Zero Suppression (NZS) packet.

### 6.4. Zero Suppression

The purpose of Zero Suppression (ZS) is to compress the output data by omitting the channels without hits and to send out only the channels with useful information (active channels). Active channels are selected by threshold \texttt{zs\_th} which is set by one of the configuration registers (see table 17). Internally the ZS block, which simplified diagram is shown in figure 11 is built as a large set of multiplexers forming a multibit output signal \texttt{adc\_zs}. The internal structure is pipelined so the latency is constant and equal two.
The `adc_zs` signal is a 12-bit vector containing 7 bits for the channel number and 5 bits for the sample value. The structure is identical to the data field of normal packet shown later in figure 14. The `zs_channels` signal contains the number of valid elements in `adc_zs` vector. The `zs_channels` is calculated according to formula:

\[
zs\_channels = \sum_{i=0}^{7} zs\_ch[i],
\]

where

\[
zs\_ch[i] = \begin{cases} 
0 & \text{adc\_mcm}[i] < zs\_th \\
1 & \text{adc\_mcm}[i] \geq zs\_th
\end{cases}
\]

It is clearly seen from formula 9 that if `zs\_th` is zero the algorithm is inactive i.e. all sampled data go to output vector `adc_zs`, regardless of the value, and `zs_channels` is equal eight.

The `mcm_channels` and `mcm_value` signals are delayed by the ZS algorithm latency only to keep the data consistency.

### 6.5. NZS data

The NZS data purpose is to monitor/verify the DSP operation outside or in between the normal data taking. In such situations the NZS data packet replaces a normal data packet. The NZS packet is created on demand if ASIC receives the TFC NZS command. The data can be taken from four different places in the DSP chain as shown in figure 12. The default data source is the data after the first masking multiplexer.

![Figure 12: NZS data sources and DSP configuration registers](image)

Because it is only a test/monitoring function there is no FIFO in parallel to the DSP chain but only one 48-bit register. This saves the power consumption but there is also a drawback, the TFC NZS command can’t be sent more frequently than every 11 clock cycles. These 11 cycles correspond to the DSP pipeline depth: 1 for pedestals,
7 for MCMS, 2 for ZS plus 1 additional clock cycle. In other case the data would be overwritten before a data packet is built.

7. Back-end data processing

The back-end data processing (i.e. data processing after the core DSP chain) is directly related to the TFC commands. The effect of TFC commands on back-end data processing is shown in figure 13. To keep synchronisation of TFC commands with data processing a number of FIFOs is necessary with the depths matching the number of pipeline stages in particular blocks. The first FIFO in figure 13 is required by LHCb DAQ and is implemented in the deserializer block (compare figure 33 in page 33).

![Figure 13: TFC commands control of back-end data processing](image)

The SALT ASIC should recognise eight TFC commands: Calib, Synch, Snapshot, BxVeto, NZS, Header, FEReset, and BXReset which interpretation is consistent with LHCb DAQ description.

7.1. Packet building and recording

The Packet Building Block (PCK) shown in figure 13 receives two data streams the NZS and the ZS. The TFC commands control which of these streams, if any, is used to form the output data packet. The PCK block can create all data packets shown in table 4 except Idle. The Idle packet is created in Idle Buffer (IDL) (see figure 13) when there is no data packet in memory. The data packet type created in PCK block directly depends of the current TFC command and memory status.

If PCK receives Synch TFC command the Sync data packet is generated regardless of the data input. The same is when PCK receives Header or BxVeto TFC but then HeaderOnly or BxVeto packets are created respectively. If there is neither Synch, Header nor BxVeto the Normal or the NZS data packet are expected depending on whether the NZS is set or not in the current TFC command. However if there is not enough space in the memory the BufferFull (instead of Normal) or BufferFullN (instead of NZS) packet is sent to the memory. Other exception of this behaviour is the situation when the Normal
Table 4: Output data packet format. Both parts Header and Data are 12-bit word aligned. The Parity is calculated only for Header part as even parity. The Flag distinguish between special and Normal packet type. Length field contains number of hits for Normal packet and type for special packet. Data part contains ADC samples or 12-bit sync pattern from configuration registers. ADC samples are in two forms: Hits or Values (see figures 14 and 15). Number of transmitted bits vary only for Normal packet and it constant for the others.

<table>
<thead>
<tr>
<th>Packet name</th>
<th>BXID</th>
<th>Parity Flag</th>
<th>Length</th>
<th>Data</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Idle</td>
<td>0000</td>
<td>1</td>
<td>1</td>
<td>'b11_0000</td>
<td>no enough data</td>
</tr>
<tr>
<td>BxVeto</td>
<td>bxid_cnt[3:0]</td>
<td>*</td>
<td>1</td>
<td>'b01_0001</td>
<td>BxVeto in TFCcmd</td>
</tr>
<tr>
<td>HeaderOnly</td>
<td>bxid_cnt[3:0]</td>
<td>*</td>
<td>1</td>
<td>'b01_0010</td>
<td>HeaderOnly in TFCcmd</td>
</tr>
<tr>
<td>BusyEvent</td>
<td>bxid_cnt[3:0]</td>
<td>*</td>
<td>1</td>
<td>'b01_0011</td>
<td>nHits &gt; 63</td>
</tr>
<tr>
<td>BufferFull</td>
<td>bxid_cnt[3:0]</td>
<td>*</td>
<td>1</td>
<td>'b01_0100</td>
<td>no space in memory</td>
</tr>
<tr>
<td>BufferFullN</td>
<td>bxid_cnt[3:0]</td>
<td>*</td>
<td>1</td>
<td>'b01_0101</td>
<td>no space in memory</td>
</tr>
<tr>
<td>NZS</td>
<td>bxid_cnt[3:0]</td>
<td>*</td>
<td>1</td>
<td>'b00_0110</td>
<td>Values NZS in TFCcmd</td>
</tr>
<tr>
<td>Normal</td>
<td>bxid_cnt[3:0]</td>
<td>*</td>
<td>0</td>
<td>nHits</td>
<td>Hits Normal event</td>
</tr>
</tbody>
</table>

Sync | bxid_cnt[11:0] | sync_pattern | Synch in TFCcmd |

Normal Packet: 4 bits 6 bits 12 bits 12 bits

<table>
<thead>
<tr>
<th>BXID</th>
<th>nHits</th>
<th>Hit 0</th>
<th>Hit 1</th>
<th>...</th>
<th>Hit nHits-1</th>
</tr>
</thead>
</table>

Hit: 7 bits 5 bits

channel number | channel value after DSP |

Figure 14: Normal packet structure. Hit part is repeated nHits times. Channels are ordered starting from the highest one.

A busy packet is expected but number of hits is greater than 63 or zs_channels (set via ECS interface) in such a case the BusyEvent packet is put into memory. All cases of data packet creation, with priorities, are shown in table 5. The Idle packet is created in the IDL block when there is no any packet in memory.

The PCK block contains the Bunch Crossing Identification number (BXID), a 12-bit counter bxid_cnt[11:0] which counts clock cycles and can be read via ECS protocol (see bxid_cnt registers in table A, appendix A). The last four bits of the counter are the part of header field in almost each data packet (see table 4). If the PCK block receive the BXReset TFC command the BXID counter is reset.

All data packets created in PCK block are sent to RAM memory (see figure 13). The size of memory is 5kb which is 648 bytes. If the memory if almost full it sends a feedback signal to PCK block and the BufferFull or BufferFullN packets are generated. The memory reacts only for FEReset TFC command, which clears it.
Figure 15: NZS packet structure: parameters for DSP monitoring and 128 channel values; for calculation of the control parameters see equation 6 and 4; for NZS data type see figure 12 and nzs_cfg configuration bits. Last bit in DSP Monitoring field is an even parity calculated for all bits in this field.

<table>
<thead>
<tr>
<th>Priority</th>
<th>Condition</th>
<th>Packet</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 (high)</td>
<td>mem_space = memSize (empty memory)</td>
<td>Idle (group of)</td>
</tr>
<tr>
<td>1</td>
<td>Synch in TFCcmd</td>
<td>Sync</td>
</tr>
<tr>
<td>2</td>
<td>Header in TFCcmd</td>
<td>HeaderOnly</td>
</tr>
<tr>
<td>3</td>
<td>BxVeto in TFCcmd</td>
<td>BxVeto</td>
</tr>
<tr>
<td>4</td>
<td>(NZS in TFCcmd) and (mem_space &lt; 67)</td>
<td>BufferFullN</td>
</tr>
<tr>
<td>5</td>
<td>NZS in TFCcmd</td>
<td>NZS</td>
</tr>
<tr>
<td>6</td>
<td>(nHits&gt;63) or (nHits&gt;=zs_channels)</td>
<td>BusyEvent</td>
</tr>
<tr>
<td>7</td>
<td>mem_space &lt; nHits</td>
<td>BufferFull</td>
</tr>
<tr>
<td>8 (low)</td>
<td>—</td>
<td>Normal</td>
</tr>
</tbody>
</table>

Table 5: Data packet selection conditions in falling priority order. The condition is written in pseudo-code. Variable meaning are: TFCcmd – current set of TFC commands, mem_space – space left in memory in 12-bit words, memSize – size of the memory, and nHits – number of hits.

### 7.2. Idle creation and data stream splitting

The last data processing block, just before the serialisation, is called the IDL in figure 13. It receives the data from RAM, adds Idles if necessary, and divides the data stream to bytes (8-bit elements) which are sent directly to the e-link serializers (data_out[5], data_out[4], . . . , data_out[0]). The Idle packets are created if there is no data to send, to keep the e-links active all the time.

The Idle packets are created in groups of size controlled by idle_cfg register (table 19 in appendix A). Because they are added to data stream only if the memory is empty the risk of memory overflow is low. The memory overflow may be avoided if idle group size is smaller then double the number of active e-links:

\[
\text{idle\_group\_size} \leq 2 \times \text{active\_elinks},
\]

where active_elinks = add_elinks + 3, and add_elinks is set in register elinks_cfg. The second restriction relates to minimum value of idle group size parameter. It should be
grater or equal then the number of active e-links:

\[ \text{active}_\text{elinks} \leq \text{idle}_\text{group}_\text{size}. \quad (11) \]

Only then the proper Idle grouping is guaranteed. The only exception is situation with 3 active e-links when the grouping works fine also for values 1 and 2.

The operation of IDL depends on a number of active e-links which can vary between 3 – 6 and is controlled by \text{add}_\text{elinks} signal from register \text{elinks}_\text{cfg} (table 19). All data packets ready to send (including Idles) have to be distributed over the number of active e-links. Before this operation data packets are virtually glued together forming continuous data stream which is then divided to 8-bit pieces suitable for the e-link serializers. The distribution process is shown in figure 16. The picture shows which part of data packet stream will be sent to the particular e-link (a digits are e-link numbers) depending on the number of active e-links. Each grey frame is effectively sent out in one clock cycle of \text{MAIN_CLK}. E-link 5 corresponds to signal \text{data}_{out}[5] from figures 12 and 13, e-link 4 to \text{data}_{out}[4], etc.. E-links which are not in use are completely off, consuming almost no any power.

![Data packet stream distribution over e-links – example](image)

Figure 16: Data packet stream distribution over e-links – example. The detailed distribution depends on the current number of active e-links. The picture shows to which e-link serializer sends particular part of data. The numbers indicate number of e-links to which a byte was sent. The e-links 5, 4 and 3 are always in use. Each active e-link corresponds to one signal \text{data}_{out} from figures 12 and 13.

Each e-link is completed with a differential Scalable Low-Voltage Signalling (SLVS) transmitter, therefore each output bit \text{data}_{out} is converted to a pair of differential signals \text{DDR}_{OUT,P} and \text{DDR}_{OUT,N} at the output of SLVS – see table 12 with pad list (page 46) and section 8.3.2 for SLVS transmitter description.
8. I/Os – pads and interfaces

8.1. Reset circuitry

All flip-flops in the whole ASIC, except some specific ones in serializer (which are part of data flow, not configuration), are reset asynchronously. No synchronous reset is used in the chip. Flip-flops are reset in general to zero value except configuration registers which have default values shown in tables in appendix A. The reset signal is also delivered to ADCs in all channels.

The reset signal is active low, so during normal chip operation it should be kept at high state. Although the reset can be applied asynchronously it needs a clock signal to be removed – the internal structure of reset is shown in figure 17.

For the reset signal \( RST_N \) asynchronous in relation to clock \( MAIN_CLK \) the race condition may occur during the reset removal. It happens when the gap between the reset rising edge (reset removal) and next clock rising edge is too small. In such case the resulting values of some of the flip-flops may be different than expected. To keep the reset asynchronous and to avoid the race condition a synchronizer is used (figure 17), which output \( \text{rst}_n \) is the source of the reset signal to the rest of the ASIC.

8.2. Communication interfaces

The SALT ASIC communicates with external world with the aid of three different interfaces/protocols:

- a slow (100 kHz) I\(^2\)C read-write interface is for ECS which controls digital logic;
- a fast (320 Mb/s) write interface with DDR serializer and SLVS transmitter is used to send the data;
- a fast (320 Mb/s) read interface with DDR deserializer and SLVS receiver is used to read the TFC commands.

These interfaces are described in the following subsections.
8.2.1. Slow \( \text{I}^2\text{C} \) interface

The SALT slow control Inter-Integrated Circuit (\( \text{I}^2\text{C} \)) interface needs a master 40 MHz clock to operate properly (see figure [1] on page [4]). Both \( \text{I}^2\text{C} \) lines the \( \text{I}^2\text{C}_{\text{SCL}} \) and \( \text{I}^2\text{C}_{\text{SDA}} \) are treated in ASIC as ordinary digital signals, so the only real clock input in the ASIC is the master clock \( \text{MAIN}_{\text{CLK}} \). The \( \text{I}^2\text{C} \) protocol was modified to achieve a maximum efficiency. The following examples show how to use the SALT slow control to obtain the desired functionality.

The \( \text{I}^2\text{C} \) master is designed to work with \( \text{I}^2\text{C}_{\text{SCL}} \) clock frequency of 100 kHz. Although it can work also with higher frequencies (up to several MHz) the lower \( \text{I}^2\text{C} \) clock frequencies are forbidden because of a watchdog circuitry preventing the module against hang up. The watchdog starts counting at every falling edge of \( \text{I}^2\text{C}_{\text{SCL}} \) signal and waits up to 12.8 \( \mu \text{s} \) till it reset the \( \text{I}^2\text{C} \) master FSM to Idle state. The watchdog operation is started at the beginning of \( \text{I}^2\text{C} \) data frame when the first falling edge on \( \text{I}^2\text{C}_{\text{SCL}} \) signal occurs (this is the part of START symbol). It is deactivated when STOP symbol is recognised.

Since the internal registers address space is 12-bit long, therefore the 4 MSB address bits are transferred during the \( \text{I}^2\text{C} \) addressing phase while the following 8 bits are sent as the first byte of data (only during write operation). The 4 MSB address bits are sent between the 3-bit chip id and the last R/W bit (see figure [18]). These 4 bits are sent only during write operation i.e. when R/W is set to 0, otherwise they are ignored (see figure [19]). In the design an auto-incrementation feature of the address counter has been implemented – the address value is increased by one after sending/receiving each acknowledge bit.

The \( \text{I}^2\text{C} \) address is defined via inputs \( \text{ID}[2:0] \) bonded to ground or supply voltage. However, the ASIC recognise also a broadcast address ‘b111. The broadcast is recognised simultaneously by all ASICs connected to common \( \text{I}^2\text{C} \) bus, so it should be used carefully e.g. reading is impossible.

A single register operations are shown in figures [18] – [20]. In all these figures blue colour means sending by master, while red one means reading by master. As it is shown, it is possible to change the data transmission direction during one transfer (between start and stop conditions).

![Figure 18: Single register write operation](image)

A multiple register writing and reading is also possible, thanks to the auto-incrementation feature. This however, requires caution as the 12-bit address is only set by the master during the write operation, using part of the first byte and the second byte. This means, that the data reading is started from the byte after the latest read/written. For exam-
Figure 19: Single register read operation

Figure 20: Single register mixed operation

Example 1 — writing  Let’s assume that the desired destination register address is ‘h002 and the chip id is ‘h3. Figure 24 is the way to save ‘hFA value into the mentioned register. Figure 25 demonstrates how to write values ‘hFA, ‘h0B, ‘hAF, ‘hBC to registers ‘h002–’h005.

Example 2 — reading  Lets assume that the internal registers were updated like in example 1 and the internal address counter points to ‘h002 again. If such situation occurs, a single read operation will end with the result as on figure 26 and the multiple read operation (with the same assumption) will result in the waveform like on figure 27.
Figure 22: Multiple register read operation

Figure 23: Multiple register mixed operation

Figure 24: Saving `hFA to `h002 address on `h3 chip

Figure 25: Saving multiple values to `h002–`h005 addresses on `h3 chip

Figure 26: Reading `h002 address on `h3 chip
Example 3 — reading with address setting  The presented earlier reading example must be prefaced by a write operation. To avoid using two frames, the transmission direction can be switched within one packet. Figure 28 presents reading a value from ’h002 address without ending the transmission. Figure 29 shows reading multiple values from a consecutive registers starting address ’h002.

8.2.2. Fast data interface with DDR serializer

The main function of serializer circuit, shown in figure 30, is to serialise the data generated in DSP (input data_out[i]) and to send it out via SLVS interface. The serializer is also called the e-port. Although the data rate is 320 Mb/s, the internal clock frequency is 160 MHz to save power. The Data Double Rate (DDR) signaling scheme is used for this aim. Because the ratio between the main clock MAIN_CLK and the output data rate is 8 the serializer sends a byte every main clock cycle. The bits are sent starting from MSb (bit 7) as shown in figure 31.

There is not phase control circuitry in the serializer block, so the receiver (e.g. FPGA) is responsible for phase adjustment to receive the correct data. From the other hand, a bit shift in a byte (or the first bit to send) is controlled by the ser_byte_start in the ser_g_cfg register (table 19). To facilitate the process of finding phase shift in FPGA and bit shift in the SALT there are two, alternative to data_out[i], data sources with known data value or sequence: pattern register (constant value) or internal counters (described below).
The last three data sources are related to TFC commands (\texttt{TFCcmd\_dsp}, \texttt{TFCcmd}, \texttt{TFCcmd\_fifo}) and can be used for the TFC interface configuration or debugging (see figures 13 and 32 for these signal sources).

The data source selection and other settings of this block are controlled by the configuration registers described in table 19 (page 81). These settings are common to all six serializer blocks. If the e-ports are not in the standard data mode they send out exactly the same data stream.

The internal counters can generate four predefined sequences: counting up, counting down, and two pseudo-random sequences called sequence 0 and sequence 1. Both pseudo-random sequences start from zero and are calculated according the formulas (Verilog HDL code):

\begin{verbatim}
seq0 <= {seq0[6:0],seq0[5]^seq0[4]^seq0[3]^~(|seq0[6:0])};
\end{verbatim}

assuming that \texttt{seq0} and \texttt{seq0} are unsigned 8-bit variables declared as \texttt{reg [7:0] seq0}, \texttt{seq1} and corresponds to sequence 0 and sequence 1 respectively. Both are shown in table 22 in appendix B (page 93).

### 8.2.3. Fast TFC Interface with DDR deserializer

In contrast to six serializers there is only one deserializer circuit in the SALT. The deserializer, shown in figure 32, receives the TFC commands synchronous to the main clock. As a result in each main clock cycle a different command is received and applied to

---

**Figure 30:** Serializer circuit with all possible data sources

**Figure 31:** E-link byte transmission; MSb is first
the DSP. The bit interpretation in deserializer, working also with DDR signaling scheme, is similar to serializer, so the first received bit is MSb (bit 7), as it was shown in figure 31 for the serializer. However, this interpretation is true only when the configuration of deserializer is correct.

It is required that the deserializer can adjust the phase of input DDR bit stream to receive the correct TFC commands. To cope with this, two phase synchronisation circuits, shown in figure 33, were added at the data input – one for each data_clk clock edge. The configuration procedure of deserializer requires an earlier configuration of correct serializer transmission. After that, the output of deserializer has to be send to the serializer and only then the correctness of input data transmission is verified.

For the input flip-flop of the synchronisation circuit (figure 33) one of clock phases, generated by the PLL circuitry, can be selected by setting the configuration register described in table 19. Although the pll_clk signals are in constant phase shift to data_clk clock the final relation between the selected pll_clk signal and the data clock is not known. Because of that some hold or setup time violation is possible at the second flip-flop, working with data_clk clock. To avoid this there are two flip-flops, one working on rising and one on falling clock edge. The clk_sel chooses which one will be used.

A correctly synchronized bits don’t end the configuration process of deserializer. The next step is to define the first bit of a byte. This parameter is controlled by the desercfg configuration register (see table 19 in page 81).

When the deserializer is properly configured each received byte contains a set of TFC commands which will control the DSP and readout operation. The meaning of the TFC command bits is presented in table 6.

To synchronize properly the TFC command stream with DAQ system a variable delay is needed in the ASIC. It was implemented as a FIFO, shown in figure 32 (TFC FIFO), and can be changed in range from 0 to 255 (see register tfc fifo cfg in table 19). The second FIFO (Calib FIFO) is intended for Calib TFC command only. Calib command calib tfc signal is connected directly to the test pulse generation circuit described in section 2.2. It is worth noting that the Calib TFC command is delayed independently in both TFC FIFO and Calib FIFO. From the first one it is sent to TFC counters block (see section 9.2) and DSP, while the second one sends it to the test pulse generation

Figure 32: Deserializer circuit – TFC command interface
Figure 33:Deserializer – phase synchronizer circuit; each phase synchronizer (from figure 32) has its own external multiplexer for \( \text{pll_clk} \) phase clock selection block.

<table>
<thead>
<tr>
<th>bit</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Calib</td>
<td>Run single calibration pulse</td>
</tr>
<tr>
<td>6</td>
<td>Synch</td>
<td>Send Sync data packet</td>
</tr>
<tr>
<td>5</td>
<td>Snapshot</td>
<td>Rewrite selected counters to Snap registers</td>
</tr>
<tr>
<td>4</td>
<td>BxVeto</td>
<td>Send BxVeto data packet</td>
</tr>
<tr>
<td>3</td>
<td>NZS</td>
<td>Omit zero suppression procedure</td>
</tr>
<tr>
<td>2</td>
<td>Header</td>
<td>Send HeaderOnly data packet</td>
</tr>
<tr>
<td>1</td>
<td>FEReset</td>
<td>Reset TFC registers and empty data buffers</td>
</tr>
<tr>
<td>0</td>
<td>BXReset</td>
<td>Reset BXID counter</td>
</tr>
</tbody>
</table>

Table 6: Location of bits in TFC command word

### 8.3. I/O pads characterisation

SALT chip uses two kinds of pads for digital communication: CMOS and SLVS. CMOS pad is in general bidirectional while for SLVS there are separate differential transmitter and receiver pads, and so SLVS transmitter/receiver is always connected to two openings.

#### 8.3.1. CMOS pads

All CMOS pads are identical and use library cell PDDW0204SCDG. Almost all of them are configured as input pads, only one is bidirectional.

If the pad works as input pad it has hysteresis of 270 mV which is shown in figure 34. The hysteresis stays constant with technology variation, but switching points may move which is shown in left plot in figure 34. As one can see the switching point of rising edge may vary between 685 – 730 mV with typical value 705 mV and switching point of falling edge varies between 415 – 460 mV with typical value 440 mV.

Similar situation can be observed with supply voltage variation in typical range ±10% (see right plot in in figure 34). Here the switching point variation is little larger: 640 –
Figure 34: CMOS pad input behaviour for different technology corners (left) and different ±10% supply voltage (right). In both plots blue curve is typical corner and nominal supply 1.2 V

760 mV (rising edge) and 390–480 mV (falling edge). Hysteresis also varies between 250–780 mV. The CMOS pad configured as output can easily drive 100 pF with rise time 65 ns. The rise time is defined as a time between 10% and 90% value of supply voltage. Output waveforms for different load are shown in figure 35.

Figure 35: CMOS pad output behaviour for different load capacitance (typical technology corner and nominal supply). Load capacitances (pF): 10, 20, 50, 100 (blue thick curve), 200 and 300. Pulse width is 500 ns.

8.3.2. SLVS transmitter

The data are sent out through a fast differential SLVS transmitter. The amplitude of output signal swing can be changed through the transmitter bias current, controlled by an internal current DAC. The output signal common voltage can be set in the range
100 – 450 mV through an internal voltage DAC.

Both control registers slvs_cur_cfg and slvs_vcm_cfg of transmitter biasing DACs are described in table 20.

In figures 36, 37, 38 the simulated transient responses of the transmitter are shown versus load capacitance, SLVS reference voltage, and SLVS bias current, respectively. In each figure both SLVS output phases are shown together with their difference.

Figure 36: SLVS transmitter transient response to 160 MHz clock input for different load capacitance 2 – 10 pF (at each output). Standard 100 Ω termination, default 215 µA bias current, 400 mV SLVS reference voltage, typical technology corner, and nominal supply.

8.3.3. SLVS receiver

The data is delivered to the deserialized through a fast differential SLVS receiver. The receiver is designed as a fast differential amplifier. It works in self-biasing configuration and so it has not controlled parameters.

All receivers have internal termination resistor of value 100 Ω which is off by default but can be turned on using others_g_cfg register (see table 20). There is only one control slvs_termination bit in others_g_cfg register so in all receivers of the whole chip the internal termination is on or off.
Figure 37: SLVS transmitter transient response to 160 MHz clock input for different SLVS reference voltage 100–600 mV. Standard 100 Ω termination, default 215 μA bias current, 2 pF load capacitance (at each output), typical technology corner, and nominal supply.
Figure 38: SLVS transmitter transient response to 160 MHz clock input for different SLVS bias current 100 – 400 µA. Standard 100 Ω termination, 400 mV SLVS reference voltage, 2 pF load capacitance (at each output), typical technology corner, and nominal supply.
9. Test and monitoring

A number of additional functionalities has been added to the SALT chip for monitoring, debugging and testing purposes. There are additional ADCs to monitor biasing voltage supplies, digital counters controlling number of received fast commands and sent data packets, etc. Single Event Up-set (SEU) are also continuously monitored as described in section 10.2.

9.1. Monitoring for analogue part

Four 6-bit ADCs (same as in all SALT channels) have been added for monitoring of following signals:

- DLL control voltage $V_{DLL}$;
- selected PLL control voltage signal $V_{PLL}$ from three PLL copies (triplicated for better radiation hardness). Selected signal depends on multiplexer setting;
- selected signal from: preamplifier bias current, Krummenacher bias current, shaper bias current, reference current for common mode voltages. Selected signal depends on multiplexer setting;
- selected signal from: single-to-differential converter bias current, calibration circuit bias current, SLVS transmitter bias current, SLVS transmitter common mode voltage. Selected signal depends on multiplexer setting.

All monitors are available via ECS registers described in table 20. By default the monitors are on but they can be turned off via bit \texttt{adc_mon_off} from register \texttt{others.gCfg} (see table 20 in appendix A).

9.2. Digital data processing monitoring

One of the important debug functionality required by LHCb DAQ system \cite{2} is TFC debug registers: counters and their snapshots. The list of implemented registers, their width and reaction for Snapshot and FeReset TFC commands is shown in table 7. Plus sign in FeReset column means that register is zeroed when FeReset TFC command is received.

All TFC counters have overflow bits grouped together in \texttt{tfcOverflowReg} register. This is the only read-write register related directly to TFC counters. Writing zero to this register clears all overflow bits.

The Snapshot command samples current contents of a counters to special registers individual for each counter. After that it can be read via \texttt{I2C} interface (ECS ); to get names and addresses of ECS registers see table 21. TFC counters overflow register is also sampled when Snapshot is received. Table 7 shows which registers are snapshot registers (plus sign in Snapshot column). It is also seen that snapshot registers don’t change when FeReset command is received.
<table>
<thead>
<tr>
<th>Counter (register)</th>
<th>Bits</th>
<th>Snapshot</th>
<th>FeReset</th>
<th>BXReset</th>
</tr>
</thead>
<tbody>
<tr>
<td>calib_cnt</td>
<td>32</td>
<td>-</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>calib_cnt_snap</td>
<td>32</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>synch_cnt</td>
<td>32</td>
<td>-</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>synch_cnt_snap</td>
<td>32</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>snapshot_cnt</td>
<td>32</td>
<td>-</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>snapshot_cnt_snap</td>
<td>32</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>bxveto_cnt</td>
<td>48</td>
<td>-</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>bxveto_cnt_snap</td>
<td>48</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>nzs_cnt</td>
<td>32</td>
<td>-</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>nzs_cnt_snap</td>
<td>32</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>header_cnt</td>
<td>48</td>
<td>-</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>header_cnt_snap</td>
<td>48</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>fereset_cnt</td>
<td>32</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>fereset_cnt_snap</td>
<td>32</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>bxreset_cnt</td>
<td>32</td>
<td>-</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>bxreset_cnt_snap</td>
<td>32</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 7: TFC counters; see table 21 for ECS registers

Except TFC counters there is a number of other counters and running registers useful for debug purposes listed in table 8 with them own snapshot registers. Both ordinary registers and their snapshots are accessible via I^2C. None of the registers from table 8 is sensitive for FeReset command.

<table>
<thead>
<tr>
<th>Counter (register)</th>
<th>Bits</th>
<th>Snapshot</th>
<th>FeReset</th>
<th>BXReset</th>
</tr>
</thead>
<tbody>
<tr>
<td>bxid_cnt</td>
<td>12</td>
<td>-</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>bxid_cnt_snap</td>
<td>12</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>idle_cnt</td>
<td>48</td>
<td>-</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>idle_cnt_snap</td>
<td>48</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>mem_space</td>
<td>16</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>mem_space_snap</td>
<td>16</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 8: Other debug counters; see tables 17 and 18 for ECS registers

The BXID counter (bxid_cnt) is the only counter in the whole ASIC which is reset on BXReset TFC command. The idle_cnt counts number of Idle data packets sent out. The mem_space shows the current space in the main RAM memory.

9.3. Debugging circuitry

Several functions, pads and circuits were added to the SALT for debugging purposes.
**Test channels**  Two complete front-end plus ADC channels are added in the analogue part, one at the top and one at the bottom of the chip, so in total there are 130 channels in SALT. These additional channels are not connected to input pads and may be pulsed only through the calibration circuitry.

These channels are off by default which means that ADC is off and calibration impulse is masked out. Other blocks are biased as any other channel. The ADC and calibration pulse may be on via bit `adc_test_off` from register `others_g_cfg` (see table 20). If the ADC is active their actual output value is available via ECS registers `adc_test0_reg` and `adc_test1_reg` (described in the table 20).

**Data clock output**  There is a number of output pads to observe a `data_clk` clock generated by PLLs. This functionality is inactive by default and may be activated by assigning high state to ENA_CLK_DEBUG pad.

Because of three PLLs inside the ASIC (effect of triplication, see section 10.2) there are three outputs with clocks `data_clkA`, `data_clkB`, and `data_clkC` generated by PLL A, PLL B and PLL C respectively. The just listed clock signal are connected to differential output pads DEBUG_DCLK_N[2:0] and DEBUG_DCLK_P[2:0]. Additionally an effective clock `data_clk` (after voter) also may be observed on differential output pads DATA_CLK_OUT_N and DATA_CLK_OUT_P.

10. Rad-hard protection

10.1. Total Ionizing Dose protection

The TSMC 130nm technology is already considered as radiation resistant. For this reason the typical protection by using ELT is not applied by default. Only in few specific blocks working with very small currents ( \(< 1 \mu A\) like biasing circuitry for Krummenacher feedback or 7-bit baseline DACs, the NMOS ELTs are used.

10.2. Single Event Upset protection and monitoring

In the SALT the SEE protection is generally limited to SEU because of used technology, main clock frequency and expected proton flux. The technique used here for SEU reduction is a Triple Modular Redundancy (TMR) applied to almost every flip-flop in the whole digital part. Additionally a clock signal MAIN_CLK and reset signals are also triplicated. As a result, to triplicate the complete clock and reset trees, there are three PLLs (called A, B, and C) and three reset synchronizers from figure 17. The described triplication should be transparent for the user except the PLLs monitoring where the user has to choose which PLL she/he wants to observe.

There are two exceptions from the above rules: DLL with circuitry connected to it and I²C master block. The DLL and both clock generated by it `adc_clk` and `calib_clk` are not triplicated. Because of that also test pulse generation block (refer to section 2.2) is not triplicated. The second exception is I²C master block because the whole block (not
flip-flops) with its watchdog is triplicated (see section 8.2.1). This TMR scheme with a watchdog implemented inside prevents the I\textsuperscript{2}C master block against SEU, SETs and FSM hang up.

Configuration registers have additional functions related to SEU and triplication. First, the values in all these registers are refreshed automatically when error is detected\textsuperscript{1}. Second, each ECS register block (refer to appendix A) has its own SEU counter shown in table 9. If the SEU is detected the counter is incremented by 1. These counters implements also saturation arithmetic so if the register value is 255 and the next error is detected the value will stay unchanged. The counter behaves as ordinary ECS register, so it is also triplicated. Moreover, it is read-write and may be reset by writing zero.

It is worth noting that the SEU counter can catch only one error in every clock cycle, so multiple corrupted registers in single clock cycle in a single ECS block will be counted as one SEU error. Second important remark is that errors detected in I\textsuperscript{2}C master block are also counted in the glb\_seu\_cnt\_reg counter in the smallest register block.

<table>
<thead>
<tr>
<th>SEU counter name</th>
<th>Number of connected bits</th>
<th>Number of connected bits</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>config. regs.</td>
<td>other</td>
</tr>
<tr>
<td>glb_seu_cnt_reg</td>
<td>5*8=40</td>
<td>1+I\textsuperscript{2}C</td>
</tr>
<tr>
<td>ana_seu_cnt_reg</td>
<td>135*8=1080</td>
<td>1</td>
</tr>
<tr>
<td>dsp_seu_cnt_reg</td>
<td>153*8=1224</td>
<td>1</td>
</tr>
<tr>
<td>mem_seu_cnt_reg</td>
<td>9*8=72</td>
<td>1</td>
</tr>
<tr>
<td>ser_seu_cnt_reg</td>
<td>11*8=88</td>
<td>1</td>
</tr>
<tr>
<td>other_seu_cnt_reg</td>
<td>33*8=264</td>
<td>1</td>
</tr>
<tr>
<td>tfc_seu_cnt_reg</td>
<td>1*8=8</td>
<td>593</td>
</tr>
</tbody>
</table>

\textbf{Table 9: Number of SEU protected bits connected to SEU counters (for ECS register names and addresses see appendix A)}

11. Padrning

The pad ring of the ASIC is shown in figure 39. All details concerning opening positions and dimensions are shown in following the figures: 40 (input), 41 (top), 42 (bottom) and 43 (back). Input and back pads are longer then top and bottom pads. The former will be first used for wafer screening and thereafter bonded to hybrid, the latter are intended only for wafer screening.

A reference number has been assigned to each pad. The numbering starts in the upper left corner of the die and runs counter-clockwise. The tables 10, 11, 12 and 13 summarise the signals and explain them. The pad coordinates refer to the centre of the pad opening.

\textsuperscript{1}This behaviour is different in comparison to RAM memory where data are also kept for longer time but no automatic correction will ever happen.
The origin of the coordinate system is defined by the lower left chip corner (0,0). The die size without sealring is 4515 µm × 10900 µm.

Table 10: **Front Pads**:

<table>
<thead>
<tr>
<th>No</th>
<th>Pin name</th>
<th>Coordinates</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>x [µm]</td>
<td>y [µm]</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>VSSA</td>
<td>103.8</td>
<td>10755</td>
<td>power</td>
</tr>
<tr>
<td>2</td>
<td>VSSA</td>
<td>103.8</td>
<td>10635</td>
<td>power</td>
</tr>
<tr>
<td>3</td>
<td>IN[0]</td>
<td>103.8</td>
<td>10530</td>
<td>input</td>
</tr>
<tr>
<td>4</td>
<td>IN[1]</td>
<td>103.8</td>
<td>10450</td>
<td>input</td>
</tr>
<tr>
<td>5</td>
<td>IN[2]</td>
<td>103.8</td>
<td>10370</td>
<td>input</td>
</tr>
<tr>
<td>6</td>
<td>IN[3]</td>
<td>103.8</td>
<td>10290</td>
<td>input</td>
</tr>
<tr>
<td>7</td>
<td>IN[4]</td>
<td>103.8</td>
<td>10210</td>
<td>input</td>
</tr>
<tr>
<td>8</td>
<td>IN[5]</td>
<td>103.8</td>
<td>10130</td>
<td>input</td>
</tr>
<tr>
<td>9</td>
<td>IN[6]</td>
<td>103.8</td>
<td>10050</td>
<td>input</td>
</tr>
<tr>
<td>10</td>
<td>IN[7]</td>
<td>103.8</td>
<td>9970</td>
<td>input</td>
</tr>
<tr>
<td>11</td>
<td>IN[8]</td>
<td>103.8</td>
<td>9890</td>
<td>input</td>
</tr>
<tr>
<td>12</td>
<td>IN[9]</td>
<td>103.8</td>
<td>9810</td>
<td>input</td>
</tr>
<tr>
<td>13</td>
<td>IN[10]</td>
<td>103.8</td>
<td>9730</td>
<td>input</td>
</tr>
<tr>
<td>14</td>
<td>IN[11]</td>
<td>103.8</td>
<td>9650</td>
<td>input</td>
</tr>
<tr>
<td>15</td>
<td>IN[12]</td>
<td>103.8</td>
<td>9570</td>
<td>input</td>
</tr>
<tr>
<td>16</td>
<td>IN[13]</td>
<td>103.8</td>
<td>9490</td>
<td>input</td>
</tr>
<tr>
<td>17</td>
<td>IN[14]</td>
<td>103.8</td>
<td>9410</td>
<td>input</td>
</tr>
<tr>
<td>18</td>
<td>IN[15]</td>
<td>103.8</td>
<td>9330</td>
<td>input</td>
</tr>
<tr>
<td>19</td>
<td>IN[16]</td>
<td>103.8</td>
<td>9250</td>
<td>input</td>
</tr>
<tr>
<td>20</td>
<td>IN[17]</td>
<td>103.8</td>
<td>9170</td>
<td>input</td>
</tr>
<tr>
<td>21</td>
<td>IN[18]</td>
<td>103.8</td>
<td>9090</td>
<td>input</td>
</tr>
<tr>
<td>22</td>
<td>IN[19]</td>
<td>103.8</td>
<td>9010</td>
<td>input</td>
</tr>
<tr>
<td>23</td>
<td>IN[20]</td>
<td>103.8</td>
<td>8930</td>
<td>input</td>
</tr>
<tr>
<td>24</td>
<td>IN[21]</td>
<td>103.8</td>
<td>8850</td>
<td>input</td>
</tr>
<tr>
<td>25</td>
<td>IN[22]</td>
<td>103.8</td>
<td>8770</td>
<td>input</td>
</tr>
<tr>
<td>26</td>
<td>IN[23]</td>
<td>103.8</td>
<td>8690</td>
<td>input</td>
</tr>
<tr>
<td>27</td>
<td>IN[24]</td>
<td>103.8</td>
<td>8610</td>
<td>input</td>
</tr>
<tr>
<td>28</td>
<td>IN[25]</td>
<td>103.8</td>
<td>8530</td>
<td>input</td>
</tr>
<tr>
<td>29</td>
<td>IN[26]</td>
<td>103.8</td>
<td>8450</td>
<td>input</td>
</tr>
<tr>
<td>30</td>
<td>IN[27]</td>
<td>103.8</td>
<td>8370</td>
<td>input</td>
</tr>
<tr>
<td>31</td>
<td>IN[28]</td>
<td>103.8</td>
<td>8290</td>
<td>input</td>
</tr>
<tr>
<td>32</td>
<td>IN[29]</td>
<td>103.8</td>
<td>8210</td>
<td>input</td>
</tr>
<tr>
<td>33</td>
<td>IN[30]</td>
<td>103.8</td>
<td>8130</td>
<td>input</td>
</tr>
<tr>
<td>34</td>
<td>IN[31]</td>
<td>103.8</td>
<td>8050</td>
<td>input</td>
</tr>
</tbody>
</table>

Continued on next page
<table>
<thead>
<tr>
<th>No</th>
<th>Pin name</th>
<th>Coordinates</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>x [µm]</td>
<td>y [µm]</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>IN[32]</td>
<td>103.8</td>
<td>7970</td>
<td>input</td>
</tr>
<tr>
<td>36</td>
<td>IN[33]</td>
<td>103.8</td>
<td>7890</td>
<td>input</td>
</tr>
<tr>
<td>37</td>
<td>IN[34]</td>
<td>103.8</td>
<td>7810</td>
<td>input</td>
</tr>
<tr>
<td>38</td>
<td>IN[35]</td>
<td>103.8</td>
<td>7730</td>
<td>input</td>
</tr>
<tr>
<td>39</td>
<td>IN[36]</td>
<td>103.8</td>
<td>7650</td>
<td>input</td>
</tr>
<tr>
<td>40</td>
<td>IN[37]</td>
<td>103.8</td>
<td>7570</td>
<td>input</td>
</tr>
<tr>
<td>41</td>
<td>IN[38]</td>
<td>103.8</td>
<td>7490</td>
<td>input</td>
</tr>
<tr>
<td>42</td>
<td>IN[39]</td>
<td>103.8</td>
<td>7410</td>
<td>input</td>
</tr>
<tr>
<td>43</td>
<td>IN[40]</td>
<td>103.8</td>
<td>7330</td>
<td>input</td>
</tr>
<tr>
<td>44</td>
<td>IN[41]</td>
<td>103.8</td>
<td>7250</td>
<td>input</td>
</tr>
<tr>
<td>45</td>
<td>IN[42]</td>
<td>103.8</td>
<td>7170</td>
<td>input</td>
</tr>
<tr>
<td>46</td>
<td>IN[43]</td>
<td>103.8</td>
<td>7090</td>
<td>input</td>
</tr>
<tr>
<td>47</td>
<td>IN[44]</td>
<td>103.8</td>
<td>7010</td>
<td>input</td>
</tr>
<tr>
<td>48</td>
<td>IN[45]</td>
<td>103.8</td>
<td>6930</td>
<td>input</td>
</tr>
<tr>
<td>49</td>
<td>IN[46]</td>
<td>103.8</td>
<td>6850</td>
<td>input</td>
</tr>
<tr>
<td>50</td>
<td>IN[47]</td>
<td>103.8</td>
<td>6770</td>
<td>input</td>
</tr>
<tr>
<td>51</td>
<td>IN[48]</td>
<td>103.8</td>
<td>6690</td>
<td>input</td>
</tr>
<tr>
<td>52</td>
<td>IN[49]</td>
<td>103.8</td>
<td>6610</td>
<td>input</td>
</tr>
<tr>
<td>53</td>
<td>IN[50]</td>
<td>103.8</td>
<td>6530</td>
<td>input</td>
</tr>
<tr>
<td>54</td>
<td>IN[51]</td>
<td>103.8</td>
<td>6450</td>
<td>input</td>
</tr>
<tr>
<td>55</td>
<td>IN[52]</td>
<td>103.8</td>
<td>6370</td>
<td>input</td>
</tr>
<tr>
<td>56</td>
<td>IN[53]</td>
<td>103.8</td>
<td>6290</td>
<td>input</td>
</tr>
<tr>
<td>57</td>
<td>IN[54]</td>
<td>103.8</td>
<td>6210</td>
<td>input</td>
</tr>
<tr>
<td>58</td>
<td>IN[55]</td>
<td>103.8</td>
<td>6130</td>
<td>input</td>
</tr>
<tr>
<td>59</td>
<td>IN[56]</td>
<td>103.8</td>
<td>6050</td>
<td>input</td>
</tr>
<tr>
<td>60</td>
<td>IN[57]</td>
<td>103.8</td>
<td>5970</td>
<td>input</td>
</tr>
<tr>
<td>61</td>
<td>IN[58]</td>
<td>103.8</td>
<td>5890</td>
<td>input</td>
</tr>
<tr>
<td>62</td>
<td>IN[59]</td>
<td>103.8</td>
<td>5810</td>
<td>input</td>
</tr>
<tr>
<td>63</td>
<td>IN[60]</td>
<td>103.8</td>
<td>5730</td>
<td>input</td>
</tr>
<tr>
<td>64</td>
<td>IN[61]</td>
<td>103.8</td>
<td>5650</td>
<td>input</td>
</tr>
<tr>
<td>65</td>
<td>IN[62]</td>
<td>103.8</td>
<td>5570</td>
<td>input</td>
</tr>
<tr>
<td>66</td>
<td>IN[63]</td>
<td>103.8</td>
<td>5490</td>
<td>input</td>
</tr>
<tr>
<td>67</td>
<td>IN[64]</td>
<td>103.8</td>
<td>5410</td>
<td>input</td>
</tr>
<tr>
<td>68</td>
<td>IN[65]</td>
<td>103.8</td>
<td>5330</td>
<td>input</td>
</tr>
<tr>
<td>69</td>
<td>IN[66]</td>
<td>103.8</td>
<td>5250</td>
<td>input</td>
</tr>
<tr>
<td>70</td>
<td>IN[67]</td>
<td>103.8</td>
<td>5170</td>
<td>input</td>
</tr>
<tr>
<td>71</td>
<td>IN[68]</td>
<td>103.8</td>
<td>5090</td>
<td>input</td>
</tr>
<tr>
<td>72</td>
<td>IN[69]</td>
<td>103.8</td>
<td>5010</td>
<td>input</td>
</tr>
<tr>
<td>73</td>
<td>IN[70]</td>
<td>103.8</td>
<td>4930</td>
<td>input</td>
</tr>
<tr>
<td>74</td>
<td>IN[71]</td>
<td>103.8</td>
<td>4850</td>
<td>input</td>
</tr>
</tbody>
</table>

Continued on next page
<table>
<thead>
<tr>
<th>No</th>
<th>Pin name</th>
<th>Coordinates</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>75</td>
<td>IN[72]</td>
<td>103.8</td>
<td>4770</td>
<td>input Analogue; input of channel 72</td>
</tr>
<tr>
<td>76</td>
<td>IN[73]</td>
<td>103.8</td>
<td>4690</td>
<td>input Analogue; input of channel 73</td>
</tr>
<tr>
<td>77</td>
<td>IN[74]</td>
<td>103.8</td>
<td>4610</td>
<td>input Analogue; input of channel 74</td>
</tr>
<tr>
<td>78</td>
<td>IN[75]</td>
<td>103.8</td>
<td>4530</td>
<td>input Analogue; input of channel 75</td>
</tr>
<tr>
<td>79</td>
<td>IN[76]</td>
<td>103.8</td>
<td>4450</td>
<td>input Analogue; input of channel 76</td>
</tr>
<tr>
<td>80</td>
<td>IN[77]</td>
<td>103.8</td>
<td>4370</td>
<td>input Analogue; input of channel 77</td>
</tr>
<tr>
<td>81</td>
<td>IN[78]</td>
<td>103.8</td>
<td>4290</td>
<td>input Analogue; input of channel 78</td>
</tr>
<tr>
<td>82</td>
<td>IN[79]</td>
<td>103.8</td>
<td>4210</td>
<td>input Analogue; input of channel 79</td>
</tr>
<tr>
<td>83</td>
<td>IN[80]</td>
<td>103.8</td>
<td>4130</td>
<td>input Analogue; input of channel 80</td>
</tr>
<tr>
<td>84</td>
<td>IN[81]</td>
<td>103.8</td>
<td>4050</td>
<td>input Analogue; input of channel 81</td>
</tr>
<tr>
<td>85</td>
<td>IN[82]</td>
<td>103.8</td>
<td>3970</td>
<td>input Analogue; input of channel 82</td>
</tr>
<tr>
<td>86</td>
<td>IN[83]</td>
<td>103.8</td>
<td>3890</td>
<td>input Analogue; input of channel 83</td>
</tr>
<tr>
<td>87</td>
<td>IN[84]</td>
<td>103.8</td>
<td>3810</td>
<td>input Analogue; input of channel 84</td>
</tr>
<tr>
<td>88</td>
<td>IN[85]</td>
<td>103.8</td>
<td>3730</td>
<td>input Analogue; input of channel 85</td>
</tr>
<tr>
<td>89</td>
<td>IN[86]</td>
<td>103.8</td>
<td>3650</td>
<td>input Analogue; input of channel 86</td>
</tr>
<tr>
<td>90</td>
<td>IN[87]</td>
<td>103.8</td>
<td>3570</td>
<td>input Analogue; input of channel 87</td>
</tr>
<tr>
<td>91</td>
<td>IN[88]</td>
<td>103.8</td>
<td>3490</td>
<td>input Analogue; input of channel 88</td>
</tr>
<tr>
<td>92</td>
<td>IN[89]</td>
<td>103.8</td>
<td>3410</td>
<td>input Analogue; input of channel 89</td>
</tr>
<tr>
<td>93</td>
<td>IN[90]</td>
<td>103.8</td>
<td>3330</td>
<td>input Analogue; input of channel 90</td>
</tr>
<tr>
<td>94</td>
<td>IN[91]</td>
<td>103.8</td>
<td>3250</td>
<td>input Analogue; input of channel 91</td>
</tr>
<tr>
<td>95</td>
<td>IN[92]</td>
<td>103.8</td>
<td>3170</td>
<td>input Analogue; input of channel 92</td>
</tr>
<tr>
<td>96</td>
<td>IN[93]</td>
<td>103.8</td>
<td>3090</td>
<td>input Analogue; input of channel 93</td>
</tr>
<tr>
<td>97</td>
<td>IN[94]</td>
<td>103.8</td>
<td>3010</td>
<td>input Analogue; input of channel 94</td>
</tr>
<tr>
<td>98</td>
<td>IN[95]</td>
<td>103.8</td>
<td>2930</td>
<td>input Analogue; input of channel 95</td>
</tr>
<tr>
<td>99</td>
<td>IN[96]</td>
<td>103.8</td>
<td>2850</td>
<td>input Analogue; input of channel 96</td>
</tr>
<tr>
<td>100</td>
<td>IN[97]</td>
<td>103.8</td>
<td>2770</td>
<td>input Analogue; input of channel 97</td>
</tr>
<tr>
<td>101</td>
<td>IN[98]</td>
<td>103.8</td>
<td>2690</td>
<td>input Analogue; input of channel 98</td>
</tr>
<tr>
<td>102</td>
<td>IN[99]</td>
<td>103.8</td>
<td>2610</td>
<td>input Analogue; input of channel 99</td>
</tr>
<tr>
<td>103</td>
<td>IN[100]</td>
<td>103.8</td>
<td>2530</td>
<td>input Analogue; input of channel 100</td>
</tr>
<tr>
<td>104</td>
<td>IN[101]</td>
<td>103.8</td>
<td>2450</td>
<td>input Analogue; input of channel 101</td>
</tr>
<tr>
<td>105</td>
<td>IN[102]</td>
<td>103.8</td>
<td>2370</td>
<td>input Analogue; input of channel 102</td>
</tr>
<tr>
<td>106</td>
<td>IN[103]</td>
<td>103.8</td>
<td>2290</td>
<td>input Analogue; input of channel 103</td>
</tr>
<tr>
<td>107</td>
<td>IN[104]</td>
<td>103.8</td>
<td>2210</td>
<td>input Analogue; input of channel 104</td>
</tr>
<tr>
<td>108</td>
<td>IN[105]</td>
<td>103.8</td>
<td>2130</td>
<td>input Analogue; input of channel 105</td>
</tr>
<tr>
<td>109</td>
<td>IN[106]</td>
<td>103.8</td>
<td>2050</td>
<td>input Analogue; input of channel 106</td>
</tr>
<tr>
<td>110</td>
<td>IN[107]</td>
<td>103.8</td>
<td>1970</td>
<td>input Analogue; input of channel 107</td>
</tr>
<tr>
<td>111</td>
<td>IN[108]</td>
<td>103.8</td>
<td>1890</td>
<td>input Analogue; input of channel 108</td>
</tr>
<tr>
<td>112</td>
<td>IN[109]</td>
<td>103.8</td>
<td>1810</td>
<td>input Analogue; input of channel 109</td>
</tr>
<tr>
<td>113</td>
<td>IN[110]</td>
<td>103.8</td>
<td>1730</td>
<td>input Analogue; input of channel 110</td>
</tr>
<tr>
<td>114</td>
<td>IN[111]</td>
<td>103.8</td>
<td>1650</td>
<td>input Analogue; input of channel 111</td>
</tr>
</tbody>
</table>

Continued on next page
Table 10 – continued from previous page

<table>
<thead>
<tr>
<th>No</th>
<th>Pin name</th>
<th>Coordinates</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>x [µm]</td>
<td>y [µm]</td>
<td></td>
</tr>
<tr>
<td>115</td>
<td>IN[112]</td>
<td>103.8</td>
<td>1570</td>
<td>input</td>
</tr>
<tr>
<td>116</td>
<td>IN[113]</td>
<td>103.8</td>
<td>1490</td>
<td>input</td>
</tr>
<tr>
<td>117</td>
<td>IN[114]</td>
<td>103.8</td>
<td>1410</td>
<td>input</td>
</tr>
<tr>
<td>118</td>
<td>IN[115]</td>
<td>103.8</td>
<td>1330</td>
<td>input</td>
</tr>
<tr>
<td>119</td>
<td>IN[116]</td>
<td>103.8</td>
<td>1250</td>
<td>input</td>
</tr>
<tr>
<td>120</td>
<td>IN[117]</td>
<td>103.8</td>
<td>1170</td>
<td>input</td>
</tr>
<tr>
<td>121</td>
<td>IN[118]</td>
<td>103.8</td>
<td>1090</td>
<td>input</td>
</tr>
<tr>
<td>122</td>
<td>IN[119]</td>
<td>103.8</td>
<td>1010</td>
<td>input</td>
</tr>
<tr>
<td>123</td>
<td>IN[120]</td>
<td>103.8</td>
<td>930</td>
<td>input</td>
</tr>
<tr>
<td>124</td>
<td>IN[121]</td>
<td>103.8</td>
<td>850</td>
<td>input</td>
</tr>
<tr>
<td>125</td>
<td>IN[122]</td>
<td>103.8</td>
<td>770</td>
<td>input</td>
</tr>
<tr>
<td>126</td>
<td>IN[123]</td>
<td>103.8</td>
<td>690</td>
<td>input</td>
</tr>
<tr>
<td>127</td>
<td>IN[124]</td>
<td>103.8</td>
<td>610</td>
<td>input</td>
</tr>
<tr>
<td>128</td>
<td>IN[125]</td>
<td>103.8</td>
<td>530</td>
<td>input</td>
</tr>
<tr>
<td>129</td>
<td>IN[126]</td>
<td>103.8</td>
<td>450</td>
<td>input</td>
</tr>
<tr>
<td>130</td>
<td>IN[127]</td>
<td>103.8</td>
<td>370</td>
<td>input</td>
</tr>
<tr>
<td>131</td>
<td>VSSA</td>
<td>103.8</td>
<td>265</td>
<td>power</td>
</tr>
<tr>
<td>132</td>
<td>VSSA</td>
<td>103.8</td>
<td>145</td>
<td>power</td>
</tr>
</tbody>
</table>

Table 11: **Bottom Pads**: all pads here are only for tests purpose; pads marked with (W) are intended for wafer screening (there is no power supply provided for the test pads buffers inside the ASIC – during tests all power supply pads have to be connected)

<table>
<thead>
<tr>
<th>No</th>
<th>Pin name</th>
<th>Coordinates</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>x [µm]</td>
<td>y [µm]</td>
<td></td>
</tr>
<tr>
<td>From LEFT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>133</td>
<td>PRE_BUF[1]</td>
<td>870</td>
<td>73.4</td>
<td>voltage out</td>
</tr>
<tr>
<td>134</td>
<td>SH1_BUF[1]</td>
<td>995</td>
<td>73.4</td>
<td>voltage out</td>
</tr>
<tr>
<td>135</td>
<td>SH2_BUF[1]</td>
<td>1120</td>
<td>73.4</td>
<td>voltage out</td>
</tr>
<tr>
<td>136</td>
<td>SH_BUF[1]</td>
<td>1245</td>
<td>73.4</td>
<td>voltage out</td>
</tr>
<tr>
<td>137</td>
<td>AP_BUF[1]</td>
<td>1370</td>
<td>73.4</td>
<td>voltage out</td>
</tr>
</tbody>
</table>

Continued on next page
Table 11 – continued from previous page

<table>
<thead>
<tr>
<th>No</th>
<th>Pin name</th>
<th>Coordinates</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>138</td>
<td>AN_BUF[1]</td>
<td>1495 73.4</td>
<td>voltage out</td>
<td>Test channel[1]: buffered s2d positive output</td>
</tr>
<tr>
<td>139</td>
<td>L_BUF[1]</td>
<td>1620 73.4</td>
<td>current in</td>
<td>Test channel[1]: buffers biasing current (default 25 µA from VDDA_BUF[1])</td>
</tr>
<tr>
<td>140</td>
<td>VSSPST_BUF[1]</td>
<td>1745 73.4</td>
<td>ESD (W)</td>
<td>Test pads ESD ground</td>
</tr>
<tr>
<td>141</td>
<td>VSSA_BUF[1]</td>
<td>1870 73.4</td>
<td>power</td>
<td>Test pads buffers ground</td>
</tr>
<tr>
<td>142</td>
<td>VDDA_BUF[1]</td>
<td>1995 73.4</td>
<td>power+ESD (W)</td>
<td>Test pads buffers and ESD power</td>
</tr>
<tr>
<td>143</td>
<td>ADC_INP</td>
<td>2120 73.4</td>
<td>voltage in</td>
<td>Test channel[1]: ADC positive input</td>
</tr>
<tr>
<td>144</td>
<td>ADC_INN</td>
<td>2245 73.4</td>
<td>voltage in</td>
<td>Test channel[1]: ADC negative input</td>
</tr>
<tr>
<td>145</td>
<td>V,S2D_DAC</td>
<td>2495 73.4</td>
<td>voltage out (W)</td>
<td>S2D biasing DAC output</td>
</tr>
<tr>
<td>146</td>
<td>V,PULSE_DAC</td>
<td>2620 73.4</td>
<td>voltage out (W)</td>
<td>Calibration test pulse amplitude DAC output</td>
</tr>
<tr>
<td>147</td>
<td>V,SLVS_BIAS_DAC</td>
<td>2745 73.4</td>
<td>voltage out (W)</td>
<td>SLVS biasing DAC output</td>
</tr>
<tr>
<td>148</td>
<td>V,SLVS_REF_DAC</td>
<td>2870 73.4</td>
<td>voltage out (W)</td>
<td>SLVS biasing DAC output</td>
</tr>
<tr>
<td>149</td>
<td>ENA_CLK_DEBUG</td>
<td>3070 73.4</td>
<td>CMOS in</td>
<td>Turns on clock debugging pads (pulled down)</td>
</tr>
</tbody>
</table>

Table 12: Backside Pads; pads marked with (W) are intended for wafer screening (there is no default value for any input signal provided inside the ASIC, all signal pads have to be connected)

<table>
<thead>
<tr>
<th>No</th>
<th>Pin name</th>
<th>Coordinates</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>150</td>
<td>VDD</td>
<td>4411.2 210</td>
<td>power</td>
<td>Digital supply</td>
</tr>
<tr>
<td>151</td>
<td>VSS</td>
<td>4411.2 350</td>
<td>power</td>
<td>Digital ground</td>
</tr>
<tr>
<td>152</td>
<td>VDD</td>
<td>4411.2 490</td>
<td>power</td>
<td>Digital supply</td>
</tr>
<tr>
<td>153</td>
<td>VSS</td>
<td>4411.2 630</td>
<td>power</td>
<td>Digital ground</td>
</tr>
<tr>
<td>154</td>
<td>VDDPST</td>
<td>4411.2 770</td>
<td>ESD</td>
<td>Digital ESD supply</td>
</tr>
<tr>
<td>155</td>
<td>ID[0]</td>
<td>4411.2 910</td>
<td>CMOS in</td>
<td>I²C address</td>
</tr>
<tr>
<td>156</td>
<td>ID[1]</td>
<td>4411.2 1050</td>
<td>CMOS in</td>
<td>I²C address</td>
</tr>
<tr>
<td>157</td>
<td>ID[2]</td>
<td>4411.2 1190</td>
<td>CMOS in</td>
<td>I²C address</td>
</tr>
</tbody>
</table>

Continued on next page
<table>
<thead>
<tr>
<th>No</th>
<th>Pin name</th>
<th>Coordinates x [µm]</th>
<th>Coordinates y [µm]</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>158</td>
<td>(I^2C) SCL</td>
<td>4411.2</td>
<td>1330</td>
<td>CMOS in</td>
<td>(I^2C) clock line</td>
</tr>
<tr>
<td>159</td>
<td>(I^2C) SDA</td>
<td>4411.2</td>
<td>1470</td>
<td>CMOS inout</td>
<td>(I^2C) data line</td>
</tr>
<tr>
<td>160</td>
<td>RST(_N)</td>
<td>4411.2</td>
<td>1610</td>
<td>CMOS in</td>
<td>Asynchronous reset</td>
</tr>
<tr>
<td>161</td>
<td>MAIN(_{CLK})(_N)</td>
<td>4411.2</td>
<td>1750</td>
<td>SLVS in</td>
<td>Main clock input (40 MHz)</td>
</tr>
<tr>
<td>162</td>
<td>MAIN(_{CLK})(_P)</td>
<td>4411.2</td>
<td>1890</td>
<td>SLVS in</td>
<td>Main clock input (40 MHz)</td>
</tr>
<tr>
<td>163</td>
<td>VSS(_{PST})</td>
<td>4411.2</td>
<td>2030</td>
<td>ESD</td>
<td>Digital ESD ground</td>
</tr>
<tr>
<td>164</td>
<td>VDD</td>
<td>4411.2</td>
<td>2170</td>
<td>power</td>
<td>Digital supply</td>
</tr>
<tr>
<td>165</td>
<td>VSS</td>
<td>4411.2</td>
<td>2310</td>
<td>power</td>
<td>Digital ground</td>
</tr>
<tr>
<td>166</td>
<td>VDD</td>
<td>4411.2</td>
<td>2450</td>
<td>power</td>
<td>Digital supply</td>
</tr>
<tr>
<td>167</td>
<td>VSS</td>
<td>4411.2</td>
<td>2590</td>
<td>power</td>
<td>Digital ground</td>
</tr>
</tbody>
</table>

**RING POWER CUT**

<table>
<thead>
<tr>
<th>No</th>
<th>Pin name</th>
<th>Coordinates x [µm]</th>
<th>Coordinates y [µm]</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>168</td>
<td>VREF(_D)</td>
<td>4411.2</td>
<td>2790</td>
<td>power</td>
<td>Analogue ADC supply</td>
</tr>
<tr>
<td>169</td>
<td>VSS(_{ADC})</td>
<td>4411.2</td>
<td>2930</td>
<td>power</td>
<td>Analogue ADC ground</td>
</tr>
<tr>
<td>170</td>
<td>VREF(_D)</td>
<td>4411.2</td>
<td>3070</td>
<td>power</td>
<td>Analogue ADC supply</td>
</tr>
<tr>
<td>171</td>
<td>VSS(_{ADC})</td>
<td>4411.2</td>
<td>3210</td>
<td>power</td>
<td>Analogue ADC ground</td>
</tr>
<tr>
<td>172</td>
<td>VDD(_{ADC})</td>
<td>4411.2</td>
<td>3350</td>
<td>power</td>
<td>Analogue ADC supply</td>
</tr>
<tr>
<td>173</td>
<td>VSS(_{ADC})</td>
<td>4411.2</td>
<td>3490</td>
<td>power</td>
<td>Analogue ADC ground</td>
</tr>
<tr>
<td>174</td>
<td>VDD(_{ADC})</td>
<td>4411.2</td>
<td>3630</td>
<td>power</td>
<td>Analogue ADC supply</td>
</tr>
<tr>
<td>175</td>
<td>VSS(_{ADC})</td>
<td>4411.2</td>
<td>3770</td>
<td>power</td>
<td>Analogue ADC ground</td>
</tr>
<tr>
<td>176</td>
<td>VDD(_{ADC})</td>
<td>4411.2</td>
<td>3910</td>
<td>power</td>
<td>Analogue ADC supply</td>
</tr>
<tr>
<td>177</td>
<td>VSS(_{ADC})</td>
<td>4411.2</td>
<td>4050</td>
<td>power</td>
<td>Analogue ADC ground</td>
</tr>
<tr>
<td>178</td>
<td>VDD(_{ADC})</td>
<td>4411.2</td>
<td>4190</td>
<td>power</td>
<td>Analogue ADC supply</td>
</tr>
<tr>
<td>179</td>
<td>VSS(_{ADC})</td>
<td>4411.2</td>
<td>4330</td>
<td>power</td>
<td>Analogue ADC ground</td>
</tr>
<tr>
<td>180</td>
<td>VDD(_{ADC})</td>
<td>4411.2</td>
<td>4470</td>
<td>power</td>
<td>Analogue ADC supply</td>
</tr>
<tr>
<td>181</td>
<td>VSS(_{PST})</td>
<td>4411.2</td>
<td>4610</td>
<td>ESD</td>
<td>Analogue ESD ground</td>
</tr>
<tr>
<td>182</td>
<td>VDD(_{PST})</td>
<td>4411.2</td>
<td>4750</td>
<td>ESD</td>
<td>Analogue ESD supply</td>
</tr>
<tr>
<td>183</td>
<td>VSS(_{PST})</td>
<td>4411.2</td>
<td>4890</td>
<td>ESD</td>
<td>Analogue ESD ground</td>
</tr>
<tr>
<td>184</td>
<td>VDD(_{PST})</td>
<td>4411.2</td>
<td>5030</td>
<td>ESD</td>
<td>Analogue ESD supply</td>
</tr>
<tr>
<td>185</td>
<td>VSSA</td>
<td>4411.2</td>
<td>5170</td>
<td>power</td>
<td>Analogue front-end ground</td>
</tr>
<tr>
<td>186</td>
<td>VDDA</td>
<td>4411.2</td>
<td>5310</td>
<td>power</td>
<td>Analogue front-end supply</td>
</tr>
<tr>
<td>187</td>
<td>VSSA</td>
<td>4411.2</td>
<td>5450</td>
<td>power</td>
<td>Analogue front-end ground</td>
</tr>
<tr>
<td>188</td>
<td>VDDA</td>
<td>4411.2</td>
<td>5590</td>
<td>power</td>
<td>Analogue front-end supply</td>
</tr>
<tr>
<td>189</td>
<td>VSSA</td>
<td>4411.2</td>
<td>5730</td>
<td>power</td>
<td>Analogue front-end ground</td>
</tr>
<tr>
<td>190</td>
<td>VDDA</td>
<td>4411.2</td>
<td>5870</td>
<td>power</td>
<td>Analogue front-end supply</td>
</tr>
<tr>
<td>191</td>
<td>VSSA</td>
<td>4411.2</td>
<td>6010</td>
<td>power</td>
<td>Analogue front-end ground</td>
</tr>
<tr>
<td>192</td>
<td>VDDA</td>
<td>4411.2</td>
<td>6150</td>
<td>power</td>
<td>Analogue front-end supply</td>
</tr>
<tr>
<td>193</td>
<td>VSSA</td>
<td>4411.2</td>
<td>6290</td>
<td>power</td>
<td>Analogue front-end ground</td>
</tr>
<tr>
<td>194</td>
<td>VDDA</td>
<td>4411.2</td>
<td>6430</td>
<td>power</td>
<td>Analogue front-end supply</td>
</tr>
<tr>
<td>195</td>
<td>VSSA</td>
<td>4411.2</td>
<td>6570</td>
<td>power</td>
<td>Analogue front-end ground</td>
</tr>
</tbody>
</table>

Continued on next page
<table>
<thead>
<tr>
<th>No</th>
<th>Pin name</th>
<th>Coordinates</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>x [µm]</td>
<td>y [µm]</td>
<td></td>
</tr>
<tr>
<td>196</td>
<td>VDDA</td>
<td>4411.2</td>
<td>6710</td>
<td>power</td>
</tr>
<tr>
<td>197</td>
<td>VSSA</td>
<td>4411.2</td>
<td>6850</td>
<td>power</td>
</tr>
<tr>
<td>198</td>
<td>VDDA</td>
<td>4411.2</td>
<td>6990</td>
<td>power</td>
</tr>
<tr>
<td>199</td>
<td>VSSA</td>
<td>4411.2</td>
<td>7130</td>
<td>power</td>
</tr>
<tr>
<td>200</td>
<td>VDDA</td>
<td>4411.2</td>
<td>7270</td>
<td>power</td>
</tr>
</tbody>
</table>

**RING POWER CUT**

<table>
<thead>
<tr>
<th>No</th>
<th>Pin name</th>
<th>Coordinates</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>x [µm]</td>
<td>y [µm]</td>
<td></td>
</tr>
<tr>
<td>201</td>
<td>VSS</td>
<td>4411.2</td>
<td>7470</td>
<td>power</td>
</tr>
<tr>
<td>202</td>
<td>VDD</td>
<td>4411.2</td>
<td>7610</td>
<td>power</td>
</tr>
<tr>
<td>203</td>
<td>VSS</td>
<td>4411.2</td>
<td>7750</td>
<td>power</td>
</tr>
<tr>
<td>204</td>
<td>VDD</td>
<td>4411.2</td>
<td>7890</td>
<td>power</td>
</tr>
<tr>
<td>205</td>
<td>VSSPST</td>
<td>4411.2</td>
<td>8030</td>
<td>ESD</td>
</tr>
<tr>
<td>206</td>
<td>DDR_TFC_N</td>
<td>4411.2</td>
<td>8170</td>
<td>SLVS in</td>
</tr>
<tr>
<td>207</td>
<td>DDR_TFC_P</td>
<td>4411.2</td>
<td>8310</td>
<td>SLVS in</td>
</tr>
<tr>
<td>217</td>
<td>DDR_OUT_P[1]</td>
<td>4411.2</td>
<td>9710</td>
<td>SLVS out</td>
</tr>
<tr>
<td>218</td>
<td>DDR_OUT_N[0]</td>
<td>4411.2</td>
<td>9850</td>
<td>SLVS out</td>
</tr>
<tr>
<td>219</td>
<td>DDR_OUT_P[0]</td>
<td>4411.2</td>
<td>9990</td>
<td>SLVS out</td>
</tr>
<tr>
<td>220</td>
<td>VDDPST</td>
<td>4411.2</td>
<td>10130</td>
<td>ESD</td>
</tr>
<tr>
<td>221</td>
<td>VSS</td>
<td>4411.2</td>
<td>10270</td>
<td>power</td>
</tr>
<tr>
<td>222</td>
<td>VDD</td>
<td>4411.2</td>
<td>10410</td>
<td>power</td>
</tr>
<tr>
<td>223</td>
<td>VSS</td>
<td>4411.2</td>
<td>10550</td>
<td>power</td>
</tr>
<tr>
<td>224</td>
<td>VDD</td>
<td>4411.2</td>
<td>10690</td>
<td>power</td>
</tr>
</tbody>
</table>

**TOP of the ASIC**
Figure 39: Pad layout of SALT ASIC. The die size is $4515 \, \mu m \times 10900 \, \mu m$. 

49
Figure 40: Front pads size and position

Figure 41: Top pads size and position

Figure 42: Bottom pads size and position
Figure 43: Back side pads size and position
Table 13: **Top Pads**: all pads here are only for tests purpose; pads marked with (W) are intended for wafer screening (there is no power supply provided for the test pads buffers inside the ASIC – during tests all power supply pads have to be connected)

<table>
<thead>
<tr>
<th>No</th>
<th>Pin name</th>
<th>Coordinates x [µm]</th>
<th>Coordinates y [µm]</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>225</td>
<td>DEBUG_DCLK_N[2]</td>
<td>4290</td>
<td>10826.6</td>
<td>SLVS out</td>
<td>Data clock from PLL C</td>
</tr>
<tr>
<td>226</td>
<td>DEBUG_DCLK_P[2]</td>
<td>4150</td>
<td>10826.6</td>
<td>SLVS out</td>
<td>Data clock from PLL C</td>
</tr>
<tr>
<td>227</td>
<td>DEBUG_DCLK_N[1]</td>
<td>4010</td>
<td>10826.6</td>
<td>SLVS out</td>
<td>Data clock from PLL B</td>
</tr>
<tr>
<td>228</td>
<td>DEBUG_DCLK_P[1]</td>
<td>3870</td>
<td>10826.6</td>
<td>SLVS out</td>
<td>Data clock from PLL B</td>
</tr>
<tr>
<td>229</td>
<td>DATA_CLK_OUT_N</td>
<td>3670</td>
<td>10826.6</td>
<td>SLVS out</td>
<td>Data clock output (voted)</td>
</tr>
<tr>
<td>230</td>
<td>DATA_CLK_OUT_P</td>
<td>3470</td>
<td>10826.6</td>
<td>SLVS out</td>
<td>Data clock output (voted)</td>
</tr>
<tr>
<td>231</td>
<td>DEBUG_DCLK_N[0]</td>
<td>3270</td>
<td>10826.6</td>
<td>SLVS out</td>
<td>Data clock from PLL A</td>
</tr>
<tr>
<td>232</td>
<td>DEBUG_DCLK_P[0]</td>
<td>3130</td>
<td>10826.6</td>
<td>SLVS out</td>
<td>Data clock from PLL A</td>
</tr>
<tr>
<td>233</td>
<td>V_SH_DAC</td>
<td>2870</td>
<td>10826.6</td>
<td>voltage out (W)</td>
<td>Shaper biasing DAC output</td>
</tr>
<tr>
<td>234</td>
<td>V_KRUM_DAC</td>
<td>2745</td>
<td>10826.6</td>
<td>voltage out (W)</td>
<td>Krum biasing DAC output</td>
</tr>
<tr>
<td>235</td>
<td>V_PRE_DAC</td>
<td>2620</td>
<td>10826.6</td>
<td>voltage out (W)</td>
<td>Pre-amp biasing DAC output</td>
</tr>
<tr>
<td>236</td>
<td>V_VCM_DAC</td>
<td>2495</td>
<td>10826.6</td>
<td>voltage out (W)</td>
<td>VCM generation reference current DAC output</td>
</tr>
<tr>
<td>237</td>
<td>VCMC</td>
<td>2370</td>
<td>10826.6</td>
<td>voltage out (W)</td>
<td>Common mode voltage C test pad</td>
</tr>
<tr>
<td>238</td>
<td>VCMB</td>
<td>2245</td>
<td>10826.6</td>
<td>voltage out (W)</td>
<td>Common mode voltage B test pad</td>
</tr>
<tr>
<td>239</td>
<td>VCMA</td>
<td>2120</td>
<td>10826.6</td>
<td>voltage out (W)</td>
<td>Common mode voltage A test pad</td>
</tr>
<tr>
<td>240</td>
<td>VDDA_BUF[0]</td>
<td>1995</td>
<td>10826.6</td>
<td>power+ESD (W)</td>
<td>Test pads buffers and ESD power</td>
</tr>
<tr>
<td>241</td>
<td>VSSA_BUF[0]</td>
<td>1870</td>
<td>10826.6</td>
<td>power</td>
<td>Test pads buffers ground</td>
</tr>
<tr>
<td>242</td>
<td>VSSPST_BUF[0]</td>
<td>1745</td>
<td>10826.6</td>
<td>ESD (W)</td>
<td>Test pads ESD ground</td>
</tr>
<tr>
<td>243</td>
<td>I_BUF[0]</td>
<td>1620</td>
<td>10826.6</td>
<td>current in</td>
<td>Test channel[0]: buffers biasing current (default 25 µA from VDDA_BUF[0])</td>
</tr>
<tr>
<td>244</td>
<td>AN_BUF[0]</td>
<td>1495</td>
<td>10826.6</td>
<td>voltage out</td>
<td>Test channel[0]: buffered s2d positive output</td>
</tr>
<tr>
<td>245</td>
<td>AP_BUF[0]</td>
<td>1370</td>
<td>10826.6</td>
<td>voltage out</td>
<td>Test channel[0]: buffered s2d positive output</td>
</tr>
</tbody>
</table>

Continued on next page
Table 13 – continued from previous page

<table>
<thead>
<tr>
<th>No</th>
<th>Pin name</th>
<th>Coordinates</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>x [µm]</td>
<td>y [µm]</td>
<td></td>
</tr>
<tr>
<td>246</td>
<td>SH_BUF[0]</td>
<td>1245</td>
<td>10826.6</td>
<td>voltage out</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Test channel[0]: buffered shaper 3rd stage output</td>
</tr>
<tr>
<td>247</td>
<td>SH2_BUF[0]</td>
<td>1120</td>
<td>10826.6</td>
<td>voltage out</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Test channel[0]: buffered shaper 2nd stage output</td>
</tr>
<tr>
<td>248</td>
<td>SH1_BUF[0]</td>
<td>995</td>
<td>10826.6</td>
<td>voltage out</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Test channel[0]: buffered shaper 1st stage output</td>
</tr>
<tr>
<td>249</td>
<td>PRE_BUF[0]</td>
<td>870</td>
<td>10826.6</td>
<td>voltage out</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Test channel[0]: buffered pre-amp output</td>
</tr>
</tbody>
</table>

12. Optical Alignment Markers

For an easier chip alignment on SALT ASIC four optical alignment markers were implemented on the top metal layer. Figure 44 shows the layout and the sizes of the alignment structure.

Figure 44: Optical alignment marker.
References


http://cds.cern.ch/record/1491666LHCb-PUB-2012-017.

A. ASIC registers

In this section the registers which are set internally and which control various SALT parameters are described. The 8-bit registers in a 12-bit address space are used. Registers are divided into six groups, shown in table 14, related to: serializer parameters, DSP parameters, analogue and mixed-mode parameters, memory settings, TFC debug counters and other parameters. The first 4 bits of the address are used to mark the group while the next 8 bits (set to 00 in address field of table 14) distinguish various registers within a group. Presently in SALT only a small fraction of 12-bit address space is used. The parameters settings of all groups are described in the following sub-sections.

<table>
<thead>
<tr>
<th>Base address</th>
<th>Register group</th>
<th>Register number</th>
</tr>
</thead>
<tbody>
<tr>
<td>'h000</td>
<td>Serializer</td>
<td>10</td>
</tr>
<tr>
<td>'h100</td>
<td>DSP</td>
<td>153</td>
</tr>
<tr>
<td>'h200</td>
<td>Analogue and mixed-mode</td>
<td>135</td>
</tr>
<tr>
<td>'h300</td>
<td>Others</td>
<td>33</td>
</tr>
<tr>
<td>'h400</td>
<td>TFC counters</td>
<td>2</td>
</tr>
<tr>
<td>'h500</td>
<td>Memory</td>
<td>8</td>
</tr>
<tr>
<td>'h600</td>
<td>Globals</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 14: Base addresses and number of registers in different configuration blocks; total number of registers are: 725 read-write and 162 read-only

55
### A.1. Global registers

Table 15: Global registers

<table>
<thead>
<tr>
<th>Addr</th>
<th>Name</th>
<th>Bit</th>
<th>Bit Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>'h600</td>
<td>reset0_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>soft reset</td>
<td>'h00 first soft reset register; the soft reset fires only if reset0_reg is different from 0 and equal to reset1_reg</td>
</tr>
<tr>
<td>'h601</td>
<td>reset1_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>soft reset</td>
<td>'h00 second soft reset register</td>
</tr>
<tr>
<td>'h602</td>
<td>reset_counter_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>'h00 reset_counter_reg +1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>soft reset</td>
<td>a counter which counts number of soft resets</td>
</tr>
<tr>
<td>'h603</td>
<td>glb_seu_cnt_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>soft reset</td>
<td>'h00 SEU counter</td>
</tr>
<tr>
<td>'h604</td>
<td>last_reset_mon</td>
<td>7:0</td>
<td>reset value</td>
<td>'h00 (read only) reset0_reg</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>soft reset</td>
<td>last value of reset0_reg which fired soft reset</td>
</tr>
</tbody>
</table>
**A.2. Analogue and mixed-mode registers**

Table 16: Analogue and mixed-mode configuration registers

<table>
<thead>
<tr>
<th>Addr</th>
<th>Name</th>
<th>Bit</th>
<th>Bit Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>'h200</td>
<td>ana_g_cfg</td>
<td>7</td>
<td>glob_trim</td>
<td>reset value Use global base line register for all trim DACs</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6:4</td>
<td>—</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>rnd_counter</td>
<td>reset value Use global base line register for all trim DACs</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2:0</td>
<td>set_del</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h201</td>
<td>preamp_cfg</td>
<td>7</td>
<td>—</td>
<td>reset value Preamplifier biasing DAC value</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4:0</td>
<td>preamp_dac</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h202</td>
<td>s2d_krum_cfg</td>
<td>7</td>
<td>s2d_low_gain</td>
<td>reset value Single-ended to differential converter gain setting; if s2d_low_gain is one gain is twice smaller then if it’s zero</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6:5</td>
<td>—</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4:0</td>
<td>krum_dac</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h203</td>
<td>shaper_cfg</td>
<td>7</td>
<td>—</td>
<td>reset value Shaper biasing DAC value</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4:0</td>
<td>shaper_dac</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h204</td>
<td>s2d_cfg</td>
<td>7</td>
<td>—</td>
<td>reset value Single-ended to differential converter biasing DAC value</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4:0</td>
<td>s2d_dac</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h205</td>
<td>baseline_g_cfg</td>
<td>7</td>
<td>all_trim[7:0]</td>
<td>reset value Value for all trim DACs if glob_trim=1</td>
</tr>
<tr>
<td>'h206</td>
<td>baseline0_cfg</td>
<td>7</td>
<td>trim_dac[7:0]</td>
<td>reset value Trim DAC for channel 0 and test channel 0</td>
</tr>
<tr>
<td>'h207</td>
<td>baseline1_cfg</td>
<td>7</td>
<td>trim_dac[15:8]</td>
<td>reset value Trim DAC for channel 1</td>
</tr>
<tr>
<td>'h208</td>
<td>baseline2_cfg</td>
<td>7</td>
<td>trim_dac[23:16]</td>
<td>reset value Trim DAC for channel 2</td>
</tr>
<tr>
<td>'h209</td>
<td>baseline3_cfg</td>
<td>7</td>
<td>trim_dac[31:24]</td>
<td>reset value Trim DAC for channel 3</td>
</tr>
<tr>
<td>'h20A</td>
<td>baseline4_cfg</td>
<td>7</td>
<td>—</td>
<td>reset value Tram DAC for channel 0</td>
</tr>
</tbody>
</table>

Continued on next page
<table>
<thead>
<tr>
<th>Addr</th>
<th>Name</th>
<th>Bit</th>
<th>Bit Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>'h20B</td>
<td>baseline5_cfg</td>
<td>7:0</td>
<td>trim_dac[39:32]</td>
<td>Trim DAC for channel 4</td>
</tr>
<tr>
<td>'h20C</td>
<td>baseline6_cfg</td>
<td>7:0</td>
<td>trim_dac[47:40]</td>
<td>Trim DAC for channel 5</td>
</tr>
<tr>
<td>'h20D</td>
<td>baseline7_cfg</td>
<td>7:0</td>
<td>trim_dac[55:48]</td>
<td>Trim DAC for channel 6</td>
</tr>
<tr>
<td>'h20E</td>
<td>baseline8_cfg</td>
<td>7:0</td>
<td>trim_dac[63:56]</td>
<td>Trim DAC for channel 7</td>
</tr>
<tr>
<td>'h20F</td>
<td>baseline9_cfg</td>
<td>7:0</td>
<td>trim_dac[71:64]</td>
<td>Trim DAC for channel 8</td>
</tr>
<tr>
<td>'h210</td>
<td>baseline10_cfg</td>
<td>7:0</td>
<td>trim_dac[79:72]</td>
<td>Trim DAC for channel 9</td>
</tr>
<tr>
<td>'h211</td>
<td>baseline11_cfg</td>
<td>7:0</td>
<td>trim_dac[87:80]</td>
<td>Trim DAC for channel 10</td>
</tr>
<tr>
<td>'h212</td>
<td>baseline12_cfg</td>
<td>7:0</td>
<td>trim_dac[95:88]</td>
<td>Trim DAC for channel 11</td>
</tr>
<tr>
<td>'h213</td>
<td>baseline13_cfg</td>
<td>7:0</td>
<td>trim_dac[103:96]</td>
<td>Trim DAC for channel 12</td>
</tr>
<tr>
<td>'h214</td>
<td>baseline14_cfg</td>
<td>7:0</td>
<td>trim_dac[111:104]</td>
<td>Trim DAC for channel 13</td>
</tr>
<tr>
<td>'h215</td>
<td>baseline15_cfg</td>
<td>7:0</td>
<td>trim_dac[119:112]</td>
<td>Trim DAC for channel 14</td>
</tr>
<tr>
<td>'h216</td>
<td>baseline16_cfg</td>
<td>7:0</td>
<td>trim_dac[127:120]</td>
<td>Trim DAC for channel 15</td>
</tr>
<tr>
<td>'h217</td>
<td>baseline17_cfg</td>
<td>7:0</td>
<td>trim_dac[135:128]</td>
<td>Trim DAC for channel 16</td>
</tr>
<tr>
<td>'h218</td>
<td>baseline18_cfg</td>
<td>7:0</td>
<td>trim_dac[143:136]</td>
<td>Trim DAC for channel 17</td>
</tr>
<tr>
<td>'h219</td>
<td>baseline19_cfg</td>
<td>7:0</td>
<td>trim_dac[151:144]</td>
<td>Trim DAC for channel 18</td>
</tr>
<tr>
<td>'h21A</td>
<td>baseline20_cfg</td>
<td>7:0</td>
<td>trim_dac[159:152]</td>
<td>Trim DAC for channel 19</td>
</tr>
<tr>
<td>'h21B</td>
<td>baseline21_cfg</td>
<td>7:0</td>
<td>trim_dac[167:160]</td>
<td>Trim DAC for channel 20</td>
</tr>
<tr>
<td>'h21C</td>
<td>baseline22_cfg</td>
<td>7:0</td>
<td>trim_dac[175:168]</td>
<td>Trim DAC for channel 21</td>
</tr>
<tr>
<td>'h21D</td>
<td>baseline23_cfg</td>
<td>7:0</td>
<td>trim_dac[183:176]</td>
<td>Trim DAC for channel 22</td>
</tr>
<tr>
<td>'h21E</td>
<td>baseline24_cfg</td>
<td>7:0</td>
<td>trim_dac[191:184]</td>
<td>Trim DAC for channel 23</td>
</tr>
<tr>
<td>Addr</td>
<td>Name</td>
<td>Bit</td>
<td>Bit Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>--------------</td>
<td>---------</td>
<td>--------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>'h21F</td>
<td>baseline25</td>
<td>7:0</td>
<td>trim_dac[199:192]</td>
<td>Trim DAC for channel 24</td>
</tr>
<tr>
<td></td>
<td>_cfg</td>
<td></td>
<td>'h21F</td>
<td>reset value</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>'h80</td>
</tr>
<tr>
<td>'h220</td>
<td>baseline26</td>
<td>7:0</td>
<td>trim_dac[207:200]</td>
<td>Trim DAC for channel 25</td>
</tr>
<tr>
<td></td>
<td>_cfg</td>
<td></td>
<td></td>
<td>'h80</td>
</tr>
<tr>
<td>'h221</td>
<td>baseline27</td>
<td>7:0</td>
<td>trim_dac[215:208]</td>
<td>Trim DAC for channel 26</td>
</tr>
<tr>
<td></td>
<td>_cfg</td>
<td></td>
<td></td>
<td>'h80</td>
</tr>
<tr>
<td>'h222</td>
<td>baseline28</td>
<td>7:0</td>
<td>trim_dac[223:216]</td>
<td>Trim DAC for channel 27</td>
</tr>
<tr>
<td></td>
<td>_cfg</td>
<td></td>
<td></td>
<td>'h80</td>
</tr>
<tr>
<td>'h223</td>
<td>baseline29</td>
<td>7:0</td>
<td>trim_dac[231:224]</td>
<td>Trim DAC for channel 28</td>
</tr>
<tr>
<td></td>
<td>_cfg</td>
<td></td>
<td></td>
<td>'h80</td>
</tr>
<tr>
<td>'h224</td>
<td>baseline30</td>
<td>7:0</td>
<td>trim_dac[239:232]</td>
<td>Trim DAC for channel 29</td>
</tr>
<tr>
<td></td>
<td>_cfg</td>
<td></td>
<td></td>
<td>'h80</td>
</tr>
<tr>
<td>'h225</td>
<td>baseline31</td>
<td>7:0</td>
<td>trim_dac[247:240]</td>
<td>Trim DAC for channel 30</td>
</tr>
<tr>
<td></td>
<td>_cfg</td>
<td></td>
<td></td>
<td>'h80</td>
</tr>
<tr>
<td>'h226</td>
<td>baseline32</td>
<td>7:0</td>
<td>trim_dac[255:248]</td>
<td>Trim DAC for channel 31</td>
</tr>
<tr>
<td></td>
<td>_cfg</td>
<td></td>
<td></td>
<td>'h80</td>
</tr>
<tr>
<td>'h227</td>
<td>baseline33</td>
<td>7:0</td>
<td>trim_dac[263:256]</td>
<td>Trim DAC for channel 32</td>
</tr>
<tr>
<td></td>
<td>_cfg</td>
<td></td>
<td></td>
<td>'h80</td>
</tr>
<tr>
<td>'h228</td>
<td>baseline34</td>
<td>7:0</td>
<td>trim_dac[271:264]</td>
<td>Trim DAC for channel 33</td>
</tr>
<tr>
<td></td>
<td>_cfg</td>
<td></td>
<td></td>
<td>'h80</td>
</tr>
<tr>
<td>'h229</td>
<td>baseline35</td>
<td>7:0</td>
<td>trim_dac[279:272]</td>
<td>Trim DAC for channel 34</td>
</tr>
<tr>
<td></td>
<td>_cfg</td>
<td></td>
<td></td>
<td>'h80</td>
</tr>
<tr>
<td>'h22A</td>
<td>baseline36</td>
<td>7:0</td>
<td>trim_dac[287:280]</td>
<td>Trim DAC for channel 35</td>
</tr>
<tr>
<td></td>
<td>_cfg</td>
<td></td>
<td></td>
<td>'h80</td>
</tr>
<tr>
<td>'h22B</td>
<td>baseline37</td>
<td>7:0</td>
<td>trim_dac[295:288]</td>
<td>Trim DAC for channel 36</td>
</tr>
<tr>
<td></td>
<td>_cfg</td>
<td></td>
<td></td>
<td>'h80</td>
</tr>
<tr>
<td>'h22C</td>
<td>baseline38</td>
<td>7:0</td>
<td>trim_dac[303:296]</td>
<td>Trim DAC for channel 37</td>
</tr>
<tr>
<td></td>
<td>_cfg</td>
<td></td>
<td></td>
<td>'h80</td>
</tr>
<tr>
<td>'h22D</td>
<td>baseline39</td>
<td>7:0</td>
<td>trim_dac[311:304]</td>
<td>Trim DAC for channel 38</td>
</tr>
<tr>
<td></td>
<td>_cfg</td>
<td></td>
<td></td>
<td>'h80</td>
</tr>
<tr>
<td>'h22E</td>
<td>baseline40</td>
<td>7:0</td>
<td>trim_dac[319:312]</td>
<td>Trim DAC for channel 39</td>
</tr>
<tr>
<td></td>
<td>_cfg</td>
<td></td>
<td></td>
<td>'h80</td>
</tr>
<tr>
<td>'h22F</td>
<td>baseline41</td>
<td>7:0</td>
<td>trim_dac[327:320]</td>
<td>Trim DAC for channel 40</td>
</tr>
<tr>
<td></td>
<td>_cfg</td>
<td></td>
<td></td>
<td>'h80</td>
</tr>
<tr>
<td>'h230</td>
<td>baseline42</td>
<td>7:0</td>
<td>trim_dac[335:328]</td>
<td>Trim DAC for channel 41</td>
</tr>
<tr>
<td></td>
<td>_cfg</td>
<td></td>
<td></td>
<td>'h80</td>
</tr>
<tr>
<td>'h231</td>
<td>baseline43</td>
<td>7:0</td>
<td>trim_dac[343:336]</td>
<td>Trim DAC for channel 42</td>
</tr>
<tr>
<td></td>
<td>_cfg</td>
<td></td>
<td></td>
<td>'h80</td>
</tr>
<tr>
<td>'h232</td>
<td>baseline44</td>
<td>7:0</td>
<td>trim_dac[351:344]</td>
<td>Trim DAC for channel 43</td>
</tr>
<tr>
<td></td>
<td>_cfg</td>
<td></td>
<td></td>
<td>'h80</td>
</tr>
</tbody>
</table>

Continued on next page
<table>
<thead>
<tr>
<th>Addr</th>
<th>Name</th>
<th>Bit</th>
<th>Bit Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>'h233</td>
<td>baseline45.cfg</td>
<td>7:0</td>
<td>trim_dac[359:352]</td>
<td>Trim DAC for channel 44</td>
</tr>
<tr>
<td>'h234</td>
<td>baseline46.cfg</td>
<td>7:0</td>
<td>trim_dac[367:360]</td>
<td>Trim DAC for channel 45</td>
</tr>
<tr>
<td>'h235</td>
<td>baseline47.cfg</td>
<td>7:0</td>
<td>trim_dac[375:368]</td>
<td>Trim DAC for channel 46</td>
</tr>
<tr>
<td>'h236</td>
<td>baseline48.cfg</td>
<td>7:0</td>
<td>trim_dac[383:376]</td>
<td>Trim DAC for channel 47</td>
</tr>
<tr>
<td>'h237</td>
<td>baseline49.cfg</td>
<td>7:0</td>
<td>trim_dac[391:384]</td>
<td>Trim DAC for channel 48</td>
</tr>
<tr>
<td>'h238</td>
<td>baseline50.cfg</td>
<td>7:0</td>
<td>trim_dac[399:392]</td>
<td>Trim DAC for channel 49</td>
</tr>
<tr>
<td>'h239</td>
<td>baseline51.cfg</td>
<td>7:0</td>
<td>trim_dac[407:400]</td>
<td>Trim DAC for channel 50</td>
</tr>
<tr>
<td>'h23A</td>
<td>baseline52.cfg</td>
<td>7:0</td>
<td>trim_dac[415:408]</td>
<td>Trim DAC for channel 51</td>
</tr>
<tr>
<td>'h23B</td>
<td>baseline53.cfg</td>
<td>7:0</td>
<td>trim_dac[423:416]</td>
<td>Trim DAC for channel 52</td>
</tr>
<tr>
<td>'h23C</td>
<td>baseline54.cfg</td>
<td>7:0</td>
<td>trim_dac[431:424]</td>
<td>Trim DAC for channel 53</td>
</tr>
<tr>
<td>'h23D</td>
<td>baseline55.cfg</td>
<td>7:0</td>
<td>trim_dac[439:432]</td>
<td>Trim DAC for channel 54</td>
</tr>
<tr>
<td>'h23E</td>
<td>baseline56.cfg</td>
<td>7:0</td>
<td>trim_dac[447:440]</td>
<td>Trim DAC for channel 55</td>
</tr>
<tr>
<td>'h23F</td>
<td>baseline57.cfg</td>
<td>7:0</td>
<td>trim_dac[455:448]</td>
<td>Trim DAC for channel 56</td>
</tr>
<tr>
<td>'h240</td>
<td>baseline58.cfg</td>
<td>7:0</td>
<td>trim_dac[463:456]</td>
<td>Trim DAC for channel 57</td>
</tr>
<tr>
<td>'h241</td>
<td>baseline59.cfg</td>
<td>7:0</td>
<td>trim_dac[471:464]</td>
<td>Trim DAC for channel 58</td>
</tr>
<tr>
<td>'h242</td>
<td>baseline60.cfg</td>
<td>7:0</td>
<td>trim_dac[479:472]</td>
<td>Trim DAC for channel 59</td>
</tr>
<tr>
<td>'h243</td>
<td>baseline61.cfg</td>
<td>7:0</td>
<td>trim_dac[487:480]</td>
<td>Trim DAC for channel 60</td>
</tr>
<tr>
<td>'h244</td>
<td>baseline62.cfg</td>
<td>7:0</td>
<td>trim_dac[495:488]</td>
<td>Trim DAC for channel 61</td>
</tr>
<tr>
<td>'h245</td>
<td>baseline63.cfg</td>
<td>7:0</td>
<td>trim_dac[503:496]</td>
<td>Trim DAC for channel 62</td>
</tr>
<tr>
<td>'h246</td>
<td>baseline64.cfg</td>
<td>7:0</td>
<td>trim_dac[511:504]</td>
<td>Trim DAC for channel 63</td>
</tr>
</tbody>
</table>

Continued on next page
Table 16 – continued from previous page

<table>
<thead>
<tr>
<th>Addr</th>
<th>Name</th>
<th>Bit</th>
<th>Bit Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>h247</td>
<td>baseline65</td>
<td>7:0</td>
<td>trim_dac</td>
<td>7:0 h80 Trim DAC for channel 64</td>
</tr>
<tr>
<td></td>
<td>_cfg</td>
<td></td>
<td>[519:512]</td>
<td></td>
</tr>
<tr>
<td>h248</td>
<td>baseline66</td>
<td>7:0</td>
<td>trim_dac</td>
<td>7:0 h80 Trim DAC for channel 65</td>
</tr>
<tr>
<td></td>
<td>_cfg</td>
<td></td>
<td>[527:520]</td>
<td></td>
</tr>
<tr>
<td>h249</td>
<td>baseline67</td>
<td>7:0</td>
<td>trim_dac</td>
<td>7:0 h80 Trim DAC for channel 66</td>
</tr>
<tr>
<td></td>
<td>_cfg</td>
<td></td>
<td>[535:528]</td>
<td></td>
</tr>
<tr>
<td>h24A</td>
<td>baseline68</td>
<td>7:0</td>
<td>trim_dac</td>
<td>7:0 h80 Trim DAC for channel 68</td>
</tr>
<tr>
<td></td>
<td>_cfg</td>
<td></td>
<td>[551:544]</td>
<td></td>
</tr>
<tr>
<td>h24B</td>
<td>baseline69</td>
<td>7:0</td>
<td>trim_dac</td>
<td>7:0 h80 Trim DAC for channel 69</td>
</tr>
<tr>
<td></td>
<td>_cfg</td>
<td></td>
<td>[559:552]</td>
<td></td>
</tr>
<tr>
<td>h24C</td>
<td>baseline70</td>
<td>7:0</td>
<td>trim_dac</td>
<td>7:0 h80 Trim DAC for channel 70</td>
</tr>
<tr>
<td></td>
<td>_cfg</td>
<td></td>
<td>[567:560]</td>
<td></td>
</tr>
<tr>
<td>h24D</td>
<td>baseline71</td>
<td>7:0</td>
<td>trim_dac</td>
<td>7:0 h80 Trim DAC for channel 71</td>
</tr>
<tr>
<td></td>
<td>_cfg</td>
<td></td>
<td>[575:568]</td>
<td></td>
</tr>
<tr>
<td>h24E</td>
<td>baseline72</td>
<td>7:0</td>
<td>trim_dac</td>
<td>7:0 h80 Trim DAC for channel 72</td>
</tr>
<tr>
<td></td>
<td>_cfg</td>
<td></td>
<td>[583:576]</td>
<td></td>
</tr>
<tr>
<td>h24F</td>
<td>baseline73</td>
<td>7:0</td>
<td>trim_dac</td>
<td>7:0 h80 Trim DAC for channel 73</td>
</tr>
<tr>
<td></td>
<td>_cfg</td>
<td></td>
<td>[591:584]</td>
<td></td>
</tr>
<tr>
<td>h250</td>
<td>baseline74</td>
<td>7:0</td>
<td>trim_dac</td>
<td>7:0 h80 Trim DAC for channel 74</td>
</tr>
<tr>
<td></td>
<td>_cfg</td>
<td></td>
<td>[599:592]</td>
<td></td>
</tr>
<tr>
<td>h251</td>
<td>baseline75</td>
<td>7:0</td>
<td>trim_dac</td>
<td>7:0 h80 Trim DAC for channel 75</td>
</tr>
<tr>
<td></td>
<td>_cfg</td>
<td></td>
<td>[607:600]</td>
<td></td>
</tr>
<tr>
<td>h252</td>
<td>baseline76</td>
<td>7:0</td>
<td>trim_dac</td>
<td>7:0 h80 Trim DAC for channel 76</td>
</tr>
<tr>
<td></td>
<td>_cfg</td>
<td></td>
<td>[615:608]</td>
<td></td>
</tr>
<tr>
<td>h253</td>
<td>baseline77</td>
<td>7:0</td>
<td>trim_dac</td>
<td>7:0 h80 Trim DAC for channel 77</td>
</tr>
<tr>
<td></td>
<td>_cfg</td>
<td></td>
<td>[623:616]</td>
<td></td>
</tr>
<tr>
<td>h254</td>
<td>baseline78</td>
<td>7:0</td>
<td>trim_dac</td>
<td>7:0 h80 Trim DAC for channel 78</td>
</tr>
<tr>
<td></td>
<td>_cfg</td>
<td></td>
<td>[631:624]</td>
<td></td>
</tr>
<tr>
<td>h255</td>
<td>baseline79</td>
<td>7:0</td>
<td>trim_dac</td>
<td>7:0 h80 Trim DAC for channel 79</td>
</tr>
<tr>
<td></td>
<td>_cfg</td>
<td></td>
<td>[639:632]</td>
<td></td>
</tr>
<tr>
<td>h256</td>
<td>baseline80</td>
<td>7:0</td>
<td>trim_dac</td>
<td>7:0 h80 Trim DAC for channel 80</td>
</tr>
<tr>
<td></td>
<td>_cfg</td>
<td></td>
<td>[647:640]</td>
<td></td>
</tr>
<tr>
<td>h257</td>
<td>baseline81</td>
<td>7:0</td>
<td>trim_dac</td>
<td>7:0 h80 Trim DAC for channel 81</td>
</tr>
<tr>
<td></td>
<td>_cfg</td>
<td></td>
<td>[655:648]</td>
<td></td>
</tr>
<tr>
<td>h258</td>
<td>baseline82</td>
<td>7:0</td>
<td>trim_dac</td>
<td>7:0 h80 Trim DAC for channel 82</td>
</tr>
<tr>
<td></td>
<td>_cfg</td>
<td></td>
<td>[663:656]</td>
<td></td>
</tr>
<tr>
<td>h259</td>
<td>baseline83</td>
<td>7:0</td>
<td>trim_dac</td>
<td>7:0 h80 Trim DAC for channel 83</td>
</tr>
<tr>
<td></td>
<td>_cfg</td>
<td></td>
<td>[671:664]</td>
<td></td>
</tr>
<tr>
<td>h25A</td>
<td>baseline84</td>
<td>7:0</td>
<td>reset value</td>
<td>7:0 h80 Trim DAC for channel 84</td>
</tr>
<tr>
<td></td>
<td>_cfg</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Continued on next page
<table>
<thead>
<tr>
<th>Addr</th>
<th>Name</th>
<th>Bit</th>
<th>Bit Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>'h25B</td>
<td>baseline85_cfg</td>
<td>7:0</td>
<td>reset value</td>
<td>h80</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>trim_dac[679:672]</td>
<td>Trim DAC for channel 84</td>
</tr>
<tr>
<td>'h25C</td>
<td>baseline86_cfg</td>
<td>7:0</td>
<td>reset value</td>
<td>h80</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>trim_dac[687:680]</td>
<td>Trim DAC for channel 85</td>
</tr>
<tr>
<td>'h25D</td>
<td>baseline87_cfg</td>
<td>7:0</td>
<td>reset value</td>
<td>h80</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>trim_dac[695:688]</td>
<td>Trim DAC for channel 86</td>
</tr>
<tr>
<td>'h25E</td>
<td>baseline88_cfg</td>
<td>7:0</td>
<td>reset value</td>
<td>h80</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>trim_dac[703:696]</td>
<td>Trim DAC for channel 87</td>
</tr>
<tr>
<td>'h25F</td>
<td>baseline89_cfg</td>
<td>7:0</td>
<td>reset value</td>
<td>h80</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>trim_dac[711:704]</td>
<td>Trim DAC for channel 88</td>
</tr>
<tr>
<td>'h260</td>
<td>baseline90_cfg</td>
<td>7:0</td>
<td>reset value</td>
<td>h80</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>trim_dac[719:712]</td>
<td>Trim DAC for channel 89</td>
</tr>
<tr>
<td>'h261</td>
<td>baseline91_cfg</td>
<td>7:0</td>
<td>reset value</td>
<td>h80</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>trim_dac[727:720]</td>
<td>Trim DAC for channel 90</td>
</tr>
<tr>
<td>'h262</td>
<td>baseline92_cfg</td>
<td>7:0</td>
<td>reset value</td>
<td>h80</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>trim_dac[735:728]</td>
<td>Trim DAC for channel 91</td>
</tr>
<tr>
<td>'h263</td>
<td>baseline93_cfg</td>
<td>7:0</td>
<td>reset value</td>
<td>h80</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>trim_dac[743:736]</td>
<td>Trim DAC for channel 92</td>
</tr>
<tr>
<td>'h264</td>
<td>baseline94_cfg</td>
<td>7:0</td>
<td>reset value</td>
<td>h80</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>trim_dac[751:744]</td>
<td>Trim DAC for channel 93</td>
</tr>
<tr>
<td>'h265</td>
<td>baseline95_cfg</td>
<td>7:0</td>
<td>reset value</td>
<td>h80</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>trim_dac[759:752]</td>
<td>Trim DAC for channel 94</td>
</tr>
<tr>
<td>'h266</td>
<td>baseline96_cfg</td>
<td>7:0</td>
<td>reset value</td>
<td>h80</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>trim_dac[767:760]</td>
<td>Trim DAC for channel 95</td>
</tr>
<tr>
<td>'h267</td>
<td>baseline97_cfg</td>
<td>7:0</td>
<td>reset value</td>
<td>h80</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>trim_dac[775:768]</td>
<td>Trim DAC for channel 96</td>
</tr>
<tr>
<td>'h268</td>
<td>baseline98_cfg</td>
<td>7:0</td>
<td>reset value</td>
<td>h80</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>trim_dac[783:776]</td>
<td>Trim DAC for channel 97</td>
</tr>
<tr>
<td>'h269</td>
<td>baseline99_cfg</td>
<td>7:0</td>
<td>reset value</td>
<td>h80</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>trim_dac[791:784]</td>
<td>Trim DAC for channel 98</td>
</tr>
<tr>
<td>'h26A</td>
<td>baseline100_cfg</td>
<td>7:0</td>
<td>reset value</td>
<td>h80</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>trim_dac[799:792]</td>
<td>Trim DAC for channel 99</td>
</tr>
<tr>
<td>'h26B</td>
<td>baseline101_cfg</td>
<td>7:0</td>
<td>reset value</td>
<td>h80</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>trim_dac[807:800]</td>
<td>Trim DAC for channel 100</td>
</tr>
<tr>
<td>'h26C</td>
<td>baseline102_cfg</td>
<td>7:0</td>
<td>reset value</td>
<td>h80</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>trim_dac[815:808]</td>
<td>Trim DAC for channel 101</td>
</tr>
<tr>
<td>'h26D</td>
<td>baseline103_cfg</td>
<td>7:0</td>
<td>reset value</td>
<td>h80</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>trim_dac[823:816]</td>
<td>Trim DAC for channel 102</td>
</tr>
<tr>
<td>'h26E</td>
<td>baseline104_cfg</td>
<td>7:0</td>
<td>reset value</td>
<td>h80</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>trim_dac[831:824]</td>
<td>Trim DAC for channel 103</td>
</tr>
</tbody>
</table>

Continued on next page
<table>
<thead>
<tr>
<th>Addr</th>
<th>Name</th>
<th>Bit</th>
<th>Bit Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>'h26F</td>
<td>baseline105_cfg</td>
<td>7:0</td>
<td>reset value trim_dac[839:832]</td>
<td>Trim DAC for channel 104</td>
</tr>
<tr>
<td>'h270</td>
<td>baseline106_cfg</td>
<td>7:0</td>
<td>reset value trim_dac[847:840]</td>
<td>Trim DAC for channel 105</td>
</tr>
<tr>
<td>'h271</td>
<td>baseline107_cfg</td>
<td>7:0</td>
<td>reset value trim_dac[855:848]</td>
<td>Trim DAC for channel 106</td>
</tr>
<tr>
<td>'h272</td>
<td>baseline108_cfg</td>
<td>7:0</td>
<td>reset value trim_dac[863:856]</td>
<td>Trim DAC for channel 107</td>
</tr>
<tr>
<td>'h273</td>
<td>baseline109_cfg</td>
<td>7:0</td>
<td>reset value trim_dac[871:864]</td>
<td>Trim DAC for channel 108</td>
</tr>
<tr>
<td>'h274</td>
<td>baseline110_cfg</td>
<td>7:0</td>
<td>reset value trim_dac[879:872]</td>
<td>Trim DAC for channel 109</td>
</tr>
<tr>
<td>'h275</td>
<td>baseline111_cfg</td>
<td>7:0</td>
<td>reset value trim_dac[887:880]</td>
<td>Trim DAC for channel 110</td>
</tr>
<tr>
<td>'h276</td>
<td>baseline112_cfg</td>
<td>7:0</td>
<td>reset value trim_dac[895:888]</td>
<td>Trim DAC for channel 111</td>
</tr>
<tr>
<td>'h277</td>
<td>baseline113_cfg</td>
<td>7:0</td>
<td>reset value trim_dac[903:896]</td>
<td>Trim DAC for channel 112</td>
</tr>
<tr>
<td>'h278</td>
<td>baseline114_cfg</td>
<td>7:0</td>
<td>reset value trim_dac[911:904]</td>
<td>Trim DAC for channel 113</td>
</tr>
<tr>
<td>'h279</td>
<td>baseline115_cfg</td>
<td>7:0</td>
<td>reset value trim_dac[919:912]</td>
<td>Trim DAC for channel 114</td>
</tr>
<tr>
<td>'h27A</td>
<td>baseline116_cfg</td>
<td>7:0</td>
<td>reset value trim_dac[927:920]</td>
<td>Trim DAC for channel 115</td>
</tr>
<tr>
<td>'h27B</td>
<td>baseline117_cfg</td>
<td>7:0</td>
<td>reset value trim_dac[935:928]</td>
<td>Trim DAC for channel 116</td>
</tr>
<tr>
<td>'h27C</td>
<td>baseline118_cfg</td>
<td>7:0</td>
<td>reset value trim_dac[943:936]</td>
<td>Trim DAC for channel 117</td>
</tr>
<tr>
<td>'h27D</td>
<td>baseline119_cfg</td>
<td>7:0</td>
<td>reset value trim_dac[951:944]</td>
<td>Trim DAC for channel 118</td>
</tr>
<tr>
<td>'h27E</td>
<td>baseline120_cfg</td>
<td>7:0</td>
<td>reset value trim_dac[959:952]</td>
<td>Trim DAC for channel 119</td>
</tr>
<tr>
<td>'h27F</td>
<td>baseline121_cfg</td>
<td>7:0</td>
<td>reset value trim_dac[967:960]</td>
<td>Trim DAC for channel 120</td>
</tr>
<tr>
<td>'h280</td>
<td>baseline122_cfg</td>
<td>7:0</td>
<td>reset value trim_dac[975:968]</td>
<td>Trim DAC for channel 121</td>
</tr>
<tr>
<td>'h281</td>
<td>baseline123_cfg</td>
<td>7:0</td>
<td>reset value trim_dac[983:976]</td>
<td>Trim DAC for channel 122</td>
</tr>
<tr>
<td>'h282</td>
<td>baseline124_cfg</td>
<td>7:0</td>
<td>reset value trim_dac[991:984]</td>
<td>Trim DAC for channel 123</td>
</tr>
</tbody>
</table>

Continued on next page
<table>
<thead>
<tr>
<th>Addr</th>
<th>Name</th>
<th>Bit</th>
<th>Bit Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>'h283</td>
<td>baseline125_cfg</td>
<td>7:0</td>
<td>trim_dac[999:992]</td>
<td>Trim DAC for channel 124</td>
</tr>
<tr>
<td>'h284</td>
<td>baseline126_cfg</td>
<td>7:0</td>
<td>trim_dac[1007:1000]</td>
<td>Trim DAC for channel 125</td>
</tr>
<tr>
<td>'h285</td>
<td>baseline127_cfg</td>
<td>7:0</td>
<td>trim_dac[1015:1008]</td>
<td>Trim DAC for channel 126 and test channel 1</td>
</tr>
<tr>
<td>'h286</td>
<td>ana_seu_cnt_reg</td>
<td>7:0</td>
<td>ana_seu_counter</td>
<td>SEU counter for analogue register block</td>
</tr>
</tbody>
</table>
A.3. DSP registers

Table 17: DSP configuration registers

<table>
<thead>
<tr>
<th>Addr</th>
<th>Name</th>
<th>Bit</th>
<th>Bit Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>'h100</td>
<td>ped_g_cfg</td>
<td>7</td>
<td></td>
<td>reset value</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>invert</td>
<td>Arithmetic negation of all input ADC samples</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>is_const</td>
<td>Use \textit{const_value} instead of ADC sample value at the input of all pedestal blocks</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>const_value</td>
<td>Value applied at all \textit{adc[i]} inputs if \textit{const_value} = 1</td>
</tr>
<tr>
<td>'h101</td>
<td>mcm_th.cfg</td>
<td>7:6</td>
<td></td>
<td>reset value</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>'h20 Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>mcm_th</td>
<td>Threshold value for MCMS block; signed number</td>
</tr>
<tr>
<td>'h102</td>
<td>mcm_th2.cfg</td>
<td>7:6</td>
<td></td>
<td>reset value</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>'h20 Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>mcm_th2</td>
<td>Second threshold value for MCMS block; signed number</td>
</tr>
<tr>
<td>'h103</td>
<td>n_zs_cfg</td>
<td>7</td>
<td></td>
<td>reset value</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>'h00 Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6:5</td>
<td>nzs_sel</td>
<td>Non-zero suppression data source:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• 00 – masked ADC data</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• 01 – synced ADC data</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• 10 – data after pedestals</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• 11 – data after MCMS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4:0</td>
<td>zs_th</td>
<td>Zero suppression threshold value; unsigned number</td>
</tr>
<tr>
<td>'h104</td>
<td>pack_adc_sync.cfg</td>
<td>7</td>
<td></td>
<td>reset value</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>adc_sync_sel</td>
<td>ADC data synchronized on one of \textit{main_clk} edges</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• 'b0 – rising</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• 'b1 – falling</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6:3</td>
<td></td>
<td>Reserved</td>
</tr>
</tbody>
</table>

Continued on next page
<table>
<thead>
<tr>
<th>Addr</th>
<th>Name</th>
<th>Bit</th>
<th>Bit Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>2:0</td>
<td>packet_types</td>
<td>Available packet types:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- 'b000 – all (normal operation)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- 'b001 – Synch &amp; Idle only</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- 'b010 – default HeaderOnly &amp; Idle; Sync if TFC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- 'b011 – default BxVeto &amp; Idle; Sync or HeaderOnly if TFC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- 'b100 – default BufferFullN &amp; Idle; Synch, HeaderOnly or BxVeto if appropriate TFC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- 'b100 – default BusyEvent &amp; Idle; Synch, HeaderOnly, BxVeto, NZS or BufferFullN if appropriate TFC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- 'b110 – default BufferFull &amp; Idle; only Normal packet is prohibited</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Addr</th>
<th>Name</th>
<th>Bit</th>
<th>Bit Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>h105</td>
<td>mask0_cfg</td>
<td>7</td>
<td>mask[7]</td>
<td>Reset value 'h00 Mask bit for channel 7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>mask[6]</td>
<td>Mask bit for channel 6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>mask[0]</td>
<td>Mask bit for channel 0</td>
</tr>
<tr>
<td>h106</td>
<td>mask1_cfg</td>
<td>7</td>
<td>mask[15]</td>
<td>Reset value 'h00 Mask bit for channel 15</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>mask[14]</td>
<td>Mask bit for channel 14</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>mask[8]</td>
<td>Mask bit for channel 8</td>
</tr>
<tr>
<td>h107</td>
<td>mask2_cfg</td>
<td>7</td>
<td>mask[23]</td>
<td>Reset value 'h00 Mask bit for channel 23</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>mask[22]</td>
<td>Mask bit for channel 22</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>mask[16]</td>
<td>Mask bit for channel 16</td>
</tr>
<tr>
<td>h108</td>
<td>mask3_cfg</td>
<td>7</td>
<td>mask[29]</td>
<td>Reset value 'h00</td>
</tr>
</tbody>
</table>

Continued on next page
<table>
<thead>
<tr>
<th>Addr</th>
<th>Name</th>
<th>Bit</th>
<th>Bit Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>'h109</td>
<td>mask4.cfg</td>
<td>7</td>
<td>mask[31]</td>
<td>Mask bit for channel 31</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>mask[30]</td>
<td>Mask bit for channel 30</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>...</td>
<td>Mask bit for channel 24</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>mask[24]</td>
<td>Mask bit for channel 24</td>
</tr>
<tr>
<td>'h10A</td>
<td>mask5.cfg</td>
<td>7</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>mask[39]</td>
<td>Mask bit for channel 39</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>...</td>
<td>Mask bit for channel 38</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>mask[32]</td>
<td>Mask bit for channel 32</td>
</tr>
<tr>
<td>'h10B</td>
<td>mask6.cfg</td>
<td>7</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>mask[47]</td>
<td>Mask bit for channel 47</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>...</td>
<td>Mask bit for channel 46</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>mask[40]</td>
<td>Mask bit for channel 40</td>
</tr>
<tr>
<td>'h10C</td>
<td>mask7.cfg</td>
<td>7</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>mask[55]</td>
<td>Mask bit for channel 55</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>...</td>
<td>Mask bit for channel 54</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>mask[48]</td>
<td>Mask bit for channel 48</td>
</tr>
<tr>
<td>'h10D</td>
<td>mask8.cfg</td>
<td>7</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>mask[63]</td>
<td>Mask bit for channel 63</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>...</td>
<td>Mask bit for channel 62</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>mask[56]</td>
<td>Mask bit for channel 56</td>
</tr>
<tr>
<td>'h10E</td>
<td>mask9.cfg</td>
<td>7</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>mask[71]</td>
<td>Mask bit for channel 71</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>...</td>
<td>Mask bit for channel 70</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>mask[64]</td>
<td>Mask bit for channel 64</td>
</tr>
<tr>
<td>'h10F</td>
<td>mask10.cfg</td>
<td>7</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>mask[79]</td>
<td>Mask bit for channel 79</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>...</td>
<td>Mask bit for channel 78</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>mask[72]</td>
<td>Mask bit for channel 72</td>
</tr>
<tr>
<td>'h110</td>
<td>mask11.cfg</td>
<td>7</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>mask[87]</td>
<td>Mask bit for channel 87</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>...</td>
<td>Mask bit for channel 86</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>mask[80]</td>
<td>Mask bit for channel 80</td>
</tr>
</tbody>
</table>

Continued on next page
Table 17 – continued from previous page

<table>
<thead>
<tr>
<th>Addr</th>
<th>Name</th>
<th>Bit</th>
<th>Bit Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>7</td>
<td>mask[95]</td>
<td>Mask bit for channel 95</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>mask[94]</td>
<td>Mask bit for channel 94</td>
</tr>
<tr>
<td></td>
<td></td>
<td>. . .</td>
<td>. . .</td>
<td>Mask bit for channel 88</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>mask[88]</td>
<td>Mask bit for channel 88</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7:6</td>
<td>—</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[0]</td>
<td>Pedestal bits for channel 0</td>
</tr>
<tr>
<td>'h111</td>
<td>mask12_cfg</td>
<td>7</td>
<td>mask[103]</td>
<td>Mask bit for channel 103</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>mask[102]</td>
<td>Mask bit for channel 102</td>
</tr>
<tr>
<td></td>
<td></td>
<td>. . .</td>
<td>. . .</td>
<td>Mask bit for channel 102</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>mask[96]</td>
<td>Mask bit for channel 96</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7:6</td>
<td>—</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[1]</td>
<td>Pedestal value for channel 1</td>
</tr>
<tr>
<td>'h112</td>
<td>mask13_cfg</td>
<td>7</td>
<td>mask[111]</td>
<td>Mask bit for channel 111</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>mask[110]</td>
<td>Mask bit for channel 110</td>
</tr>
<tr>
<td></td>
<td></td>
<td>. . .</td>
<td>. . .</td>
<td>Mask bit for channel 110</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>mask[104]</td>
<td>Mask bit for channel 104</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7:6</td>
<td>—</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[2]</td>
<td>Pedestal value for channel 2</td>
</tr>
<tr>
<td>'h113</td>
<td>mask14_cfg</td>
<td>7</td>
<td>mask[119]</td>
<td>Mask bit for channel 119</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>mask[118]</td>
<td>Mask bit for channel 118</td>
</tr>
<tr>
<td></td>
<td></td>
<td>. . .</td>
<td>. . .</td>
<td>Mask bit for channel 118</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>mask[112]</td>
<td>Mask bit for channel 112</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7:6</td>
<td>—</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[3]</td>
<td>Pedestal value for channel 3</td>
</tr>
<tr>
<td>'h114</td>
<td>mask15_cfg</td>
<td>7</td>
<td>mask[127]</td>
<td>Mask bit for channel 127</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>mask[126]</td>
<td>Mask bit for channel 126</td>
</tr>
<tr>
<td></td>
<td></td>
<td>. . .</td>
<td>. . .</td>
<td>Mask bit for channel 126</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>mask[120]</td>
<td>Mask bit for channel 120</td>
</tr>
<tr>
<td>'h115</td>
<td>ped0_cfg</td>
<td>7:6</td>
<td>—</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[0]</td>
<td>Pedestal bits for channel 0</td>
</tr>
<tr>
<td>'h116</td>
<td>ped1_cfg</td>
<td>7:6</td>
<td>—</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[1]</td>
<td>Pedestal value for channel 1</td>
</tr>
<tr>
<td>'h117</td>
<td>ped2_cfg</td>
<td>7:6</td>
<td>—</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[2]</td>
<td>Pedestal value for channel 2</td>
</tr>
<tr>
<td>'h118</td>
<td>ped3_cfg</td>
<td>7:6</td>
<td>—</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[3]</td>
<td>Pedestal value for channel 3</td>
</tr>
<tr>
<td>'h119</td>
<td>ped4_cfg</td>
<td>7:6</td>
<td>—</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[4]</td>
<td>Pedestal value for channel 4</td>
</tr>
<tr>
<td>'h11A</td>
<td>ped5_cfg</td>
<td>7:6</td>
<td>—</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[5]</td>
<td>Pedestal value for channel 4</td>
</tr>
<tr>
<td>Addr</td>
<td>Name</td>
<td>Bit</td>
<td>Bit Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>----------</td>
<td>-------</td>
<td>----------</td>
<td>--------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7:6</td>
<td>—</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>—</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h11B</td>
<td>ped6_cfg</td>
<td>7:6</td>
<td>—</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[6]</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7:6</td>
<td>reset value</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[7]</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h11C</td>
<td>ped7_cfg</td>
<td>7:6</td>
<td>—</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[8]</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7:6</td>
<td>reset value</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h11D</td>
<td>ped8_cfg</td>
<td>7:6</td>
<td>—</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[9]</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7:6</td>
<td>reset value</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h11E</td>
<td>ped9_cfg</td>
<td>7:6</td>
<td>—</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[10]</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7:6</td>
<td>reset value</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h11F</td>
<td>ped10_cfg</td>
<td>7:6</td>
<td>—</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7:6</td>
<td>reset value</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h120</td>
<td>ped11_cfg</td>
<td>7:6</td>
<td>—</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[12]</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7:6</td>
<td>reset value</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h121</td>
<td>ped12_cfg</td>
<td>7:6</td>
<td>—</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[13]</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7:6</td>
<td>reset value</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h122</td>
<td>ped13_cfg</td>
<td>7:6</td>
<td>—</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[14]</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7:6</td>
<td>reset value</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h123</td>
<td>ped14_cfg</td>
<td>7:6</td>
<td>—</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[15]</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7:6</td>
<td>reset value</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h124</td>
<td>ped15_cfg</td>
<td>7:6</td>
<td>—</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[16]</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7:6</td>
<td>reset value</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h125</td>
<td>ped16_cfg</td>
<td>7:6</td>
<td>—</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[17]</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7:6</td>
<td>reset value</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h126</td>
<td>ped17_cfg</td>
<td>7:6</td>
<td>—</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[18]</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7:6</td>
<td>reset value</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h127</td>
<td>ped18_cfg</td>
<td>7:6</td>
<td>—</td>
<td>'h00</td>
</tr>
</tbody>
</table>

Continued on next page
Table 17 – continued from previous page

<table>
<thead>
<tr>
<th>Addr</th>
<th>Name</th>
<th>Bit</th>
<th>Bit Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[18]</td>
<td>Pedestal value for channel 18</td>
</tr>
<tr>
<td>h128</td>
<td>ped19_cfg</td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[19]</td>
<td>Reserved</td>
</tr>
<tr>
<td>h129</td>
<td>ped20_cfg</td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[20]</td>
<td>Reserved</td>
</tr>
<tr>
<td>h12A</td>
<td>ped21_cfg</td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[21]</td>
<td>Reserved</td>
</tr>
<tr>
<td>h12B</td>
<td>ped22_cfg</td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[22]</td>
<td>Reserved</td>
</tr>
<tr>
<td>h12C</td>
<td>ped23_cfg</td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[23]</td>
<td>Reserved</td>
</tr>
<tr>
<td>h12D</td>
<td>ped24_cfg</td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[24]</td>
<td>Reserved</td>
</tr>
<tr>
<td>h12E</td>
<td>ped25_cfg</td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[25]</td>
<td>Reserved</td>
</tr>
<tr>
<td>h12F</td>
<td>ped26_cfg</td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[26]</td>
<td>Reserved</td>
</tr>
<tr>
<td>h130</td>
<td>ped27_cfg</td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[27]</td>
<td>Reserved</td>
</tr>
<tr>
<td>h131</td>
<td>ped28_cfg</td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[28]</td>
<td>Reserved</td>
</tr>
<tr>
<td>h132</td>
<td>ped29_cfg</td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[29]</td>
<td>Reserved</td>
</tr>
<tr>
<td>h133</td>
<td>ped30_cfg</td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[30]</td>
<td>Reserved</td>
</tr>
<tr>
<td>h134</td>
<td>ped31_cfg</td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[31]</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

Continued on next page
<table>
<thead>
<tr>
<th>Addr</th>
<th>Name</th>
<th>Bit</th>
<th>Bit Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>'h135</td>
<td>ped32_cfg</td>
<td>7:6</td>
<td>—</td>
<td>'h00 Reserved Pedestal value for channel 32</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[32]</td>
<td></td>
</tr>
<tr>
<td>'h136</td>
<td>ped33_cfg</td>
<td>7:6</td>
<td>—</td>
<td>'h00 Reserved Pedestal value for channel 33</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[33]</td>
<td></td>
</tr>
<tr>
<td>'h137</td>
<td>ped34_cfg</td>
<td>7:6</td>
<td>—</td>
<td>'h00 Reserved Pedestal value for channel 34</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[34]</td>
<td></td>
</tr>
<tr>
<td>'h138</td>
<td>ped35_cfg</td>
<td>7:6</td>
<td>—</td>
<td>'h00 Reserved Pedestal value for channel 35</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[35]</td>
<td></td>
</tr>
<tr>
<td>'h139</td>
<td>ped36_cfg</td>
<td>7:6</td>
<td>—</td>
<td>'h00 Reserved Pedestal value for channel 36</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[36]</td>
<td></td>
</tr>
<tr>
<td>'h13A</td>
<td>ped37_cfg</td>
<td>7:6</td>
<td>—</td>
<td>'h00 Reserved Pedestal value for channel 37</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[37]</td>
<td></td>
</tr>
<tr>
<td>'h13B</td>
<td>ped38_cfg</td>
<td>7:6</td>
<td>—</td>
<td>'h00 Reserved Pedestal value for channel 38</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[38]</td>
<td></td>
</tr>
<tr>
<td>'h13C</td>
<td>ped39_cfg</td>
<td>7:6</td>
<td>—</td>
<td>'h00 Reserved Pedestal value for channel 39</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[39]</td>
<td></td>
</tr>
<tr>
<td>'h13D</td>
<td>ped40_cfg</td>
<td>7:6</td>
<td>—</td>
<td>'h00 Reserved Pedestal value for channel 40</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[40]</td>
<td></td>
</tr>
<tr>
<td>'h13E</td>
<td>ped41_cfg</td>
<td>7:6</td>
<td>—</td>
<td>'h00 Reserved Pedestal value for channel 41</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[41]</td>
<td></td>
</tr>
<tr>
<td>'h13F</td>
<td>ped42_cfg</td>
<td>7:6</td>
<td>—</td>
<td>'h00 Reserved Pedestal value for channel 42</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[42]</td>
<td></td>
</tr>
<tr>
<td>'h140</td>
<td>ped43_cfg</td>
<td>7:6</td>
<td>—</td>
<td>'h00 Reserved Pedestal value for channel 43</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[43]</td>
<td></td>
</tr>
<tr>
<td>'h141</td>
<td>ped44_cfg</td>
<td>7:6</td>
<td>—</td>
<td>'h00 Reserved Pedestal value for channel 44</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[44]</td>
<td></td>
</tr>
<tr>
<td>'h142</td>
<td>ped45_cfg</td>
<td>7:6</td>
<td>—</td>
<td>'h00 Reserved Pedestal value for channel 45</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[45]</td>
<td></td>
</tr>
<tr>
<td>Addr</td>
<td>Name</td>
<td>Bit</td>
<td>Bit Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>-----</td>
<td>----------</td>
<td>------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7:6</td>
<td>—</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>—</td>
<td>Pedestal value for channel 45</td>
</tr>
<tr>
<td>'h143</td>
<td>ped46_cfg</td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[45]</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h144</td>
<td>ped47_cfg</td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[46]</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h145</td>
<td>ped48_cfg</td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[47]</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h146</td>
<td>ped49_cfg</td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[48]</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h147</td>
<td>ped50_cfg</td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[49]</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h148</td>
<td>ped51_cfg</td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[50]</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h149</td>
<td>ped52_cfg</td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[51]</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h14A</td>
<td>ped53_cfg</td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[52]</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h14B</td>
<td>ped54_cfg</td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[53]</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h14C</td>
<td>ped55_cfg</td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[54]</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h14D</td>
<td>ped56_cfg</td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[55]</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h14E</td>
<td>ped57_cfg</td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[56]</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h14F</td>
<td>ped58_cfg</td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[57]</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

Continued on next page
<table>
<thead>
<tr>
<th>Addr</th>
<th>Name</th>
<th>Bit</th>
<th>Bit Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>'h150</td>
<td>ped59_cfg</td>
<td>5:0</td>
<td>ped[58]</td>
<td>Pedestal value for channel 58</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[59]</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h151</td>
<td>ped60_cfg</td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[60]</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h152</td>
<td>ped61_cfg</td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[61]</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h153</td>
<td>ped62_cfg</td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[62]</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h154</td>
<td>ped63_cfg</td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[63]</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h155</td>
<td>ped64_cfg</td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[64]</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h156</td>
<td>ped65_cfg</td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[65]</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h157</td>
<td>ped66_cfg</td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[66]</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h158</td>
<td>ped67_cfg</td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[67]</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h159</td>
<td>ped68_cfg</td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[68]</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h15A</td>
<td>ped69_cfg</td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[69]</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h15B</td>
<td>ped70_cfg</td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[70]</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h15C</td>
<td>ped71_cfg</td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[71]</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

Continued on next page
<table>
<thead>
<tr>
<th>Addr</th>
<th>Name</th>
<th>Bit</th>
<th>Bit Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>'h15D</td>
<td>ped72_cfg</td>
<td>7:6</td>
<td>—</td>
<td>reset value 'h00 Reserved Pedestal value for channel 72</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[72]</td>
<td></td>
</tr>
<tr>
<td>'h15E</td>
<td>ped73_cfg</td>
<td>7:6</td>
<td>—</td>
<td>reset value 'h00 Reserved Pedestal value for channel 73</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[73]</td>
<td></td>
</tr>
<tr>
<td>'h15F</td>
<td>ped74_cfg</td>
<td>7:6</td>
<td>—</td>
<td>reset value 'h00 Reserved Pedestal value for channel 74</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[74]</td>
<td></td>
</tr>
<tr>
<td>'h160</td>
<td>ped75_cfg</td>
<td>7:6</td>
<td>—</td>
<td>'h00 Reserved Pedestal value for channel 75</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[75]</td>
<td></td>
</tr>
<tr>
<td>'h161</td>
<td>ped76_cfg</td>
<td>7:6</td>
<td>—</td>
<td>'h00 Reserved Pedestal value for channel 76</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[76]</td>
<td></td>
</tr>
<tr>
<td>'h162</td>
<td>ped77_cfg</td>
<td>7:6</td>
<td>—</td>
<td>'h00 Reserved Pedestal value for channel 77</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[77]</td>
<td></td>
</tr>
<tr>
<td>'h163</td>
<td>ped78_cfg</td>
<td>7:6</td>
<td>—</td>
<td>'h00 Reserved Pedestal value for channel 78</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[78]</td>
<td></td>
</tr>
<tr>
<td>'h164</td>
<td>ped79_cfg</td>
<td>7:6</td>
<td>—</td>
<td>'h00 Reserved Pedestal value for channel 79</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[79]</td>
<td></td>
</tr>
<tr>
<td>'h165</td>
<td>ped80_cfg</td>
<td>7:6</td>
<td>—</td>
<td>'h00 Reserved Pedestal value for channel 80</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[80]</td>
<td></td>
</tr>
<tr>
<td>'h166</td>
<td>ped81_cfg</td>
<td>7:6</td>
<td>—</td>
<td>'h00 Reserved Pedestal value for channel 81</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[81]</td>
<td></td>
</tr>
<tr>
<td>'h167</td>
<td>ped82_cfg</td>
<td>7:6</td>
<td>—</td>
<td>'h00 Reserved Pedestal value for channel 82</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[82]</td>
<td></td>
</tr>
<tr>
<td>'h168</td>
<td>ped83_cfg</td>
<td>7:6</td>
<td>—</td>
<td>'h00 Reserved Pedestal value for channel 83</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[83]</td>
<td></td>
</tr>
<tr>
<td>'h169</td>
<td>ped84_cfg</td>
<td>7:6</td>
<td>—</td>
<td>'h00 Reserved Pedestal value for channel 84</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[84]</td>
<td></td>
</tr>
<tr>
<td>'h16A</td>
<td>ped85_cfg</td>
<td>7:6</td>
<td>—</td>
<td>'h00 Reserved Pedestal value for channel 85</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[85]</td>
<td></td>
</tr>
</tbody>
</table>

Continued on next page
<table>
<thead>
<tr>
<th>Addr</th>
<th>Name</th>
<th>Bit</th>
<th>Bit Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>7:6</td>
<td>—</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[85]</td>
<td>Pedestal value for channel 85</td>
</tr>
<tr>
<td>h16B</td>
<td>ped86_cfg</td>
<td>7:6</td>
<td>—</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[86]</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Pedestal value for channel 86</td>
</tr>
<tr>
<td>h16C</td>
<td>ped87_cfg</td>
<td>7:6</td>
<td>—</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[87]</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Pedestal value for channel 87</td>
</tr>
<tr>
<td>h16D</td>
<td>ped88_cfg</td>
<td>7:6</td>
<td>—</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[88]</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Pedestal value for channel 88</td>
</tr>
<tr>
<td>h16E</td>
<td>ped89_cfg</td>
<td>7:6</td>
<td>—</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[89]</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Pedestal value for channel 89</td>
</tr>
<tr>
<td>h16F</td>
<td>ped90_cfg</td>
<td>7:6</td>
<td>—</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[90]</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Pedestal value for channel 90</td>
</tr>
<tr>
<td>h170</td>
<td>ped91_cfg</td>
<td>7:6</td>
<td>—</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[91]</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Pedestal value for channel 91</td>
</tr>
<tr>
<td>h171</td>
<td>ped92_cfg</td>
<td>7:6</td>
<td>—</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[92]</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Pedestal value for channel 92</td>
</tr>
<tr>
<td>h172</td>
<td>ped93_cfg</td>
<td>7:6</td>
<td>—</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[93]</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Pedestal value for channel 93</td>
</tr>
<tr>
<td>h173</td>
<td>ped94_cfg</td>
<td>7:6</td>
<td>—</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[94]</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Pedestal value for channel 94</td>
</tr>
<tr>
<td>h174</td>
<td>ped95_cfg</td>
<td>7:6</td>
<td>—</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[95]</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Pedestal value for channel 95</td>
</tr>
<tr>
<td>h175</td>
<td>ped96_cfg</td>
<td>7:6</td>
<td>—</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[96]</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Pedestal value for channel 96</td>
</tr>
<tr>
<td>h176</td>
<td>ped97_cfg</td>
<td>7:6</td>
<td>—</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[97]</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Pedestal value for channel 97</td>
</tr>
<tr>
<td>h177</td>
<td>ped98_cfg</td>
<td>7:6</td>
<td>—</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[98]</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Continued on next page</td>
</tr>
<tr>
<td>Addr</td>
<td>Name</td>
<td>Bit</td>
<td>Bit Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
<td>-----</td>
<td>----------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>'h178</td>
<td>ped99_cfg</td>
<td>5:0</td>
<td>ped[98]</td>
<td>Pedestal value for channel 98</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[99]</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h179</td>
<td>ped100_cfg</td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[100]</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h17A</td>
<td>ped101_cfg</td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[101]</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h17B</td>
<td>ped102_cfg</td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[102]</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h17C</td>
<td>ped103_cfg</td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[103]</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h17D</td>
<td>ped104_cfg</td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[104]</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h17E</td>
<td>ped105_cfg</td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[105]</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h17F</td>
<td>ped106_cfg</td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[106]</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h180</td>
<td>ped107_cfg</td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[107]</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h181</td>
<td>ped108_cfg</td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[108]</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h182</td>
<td>ped109_cfg</td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[109]</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h183</td>
<td>ped110_cfg</td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[110]</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h184</td>
<td>ped111_cfg</td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>ped[111]</td>
<td>Pedestal value for channel 111</td>
</tr>
<tr>
<td>Addr</td>
<td>Name</td>
<td>Bit</td>
<td>Bit Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-----------</td>
<td>-----</td>
<td>----------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>'h185</td>
<td>ped112_cfg</td>
<td>7:6</td>
<td>ped[112]</td>
<td>reset value</td>
</tr>
<tr>
<td>'h186</td>
<td>ped113_cfg</td>
<td>7:6</td>
<td>ped[113]</td>
<td>reset value</td>
</tr>
<tr>
<td>'h187</td>
<td>ped114_cfg</td>
<td>7:6</td>
<td>ped[114]</td>
<td>reset value</td>
</tr>
<tr>
<td>'h188</td>
<td>ped115_cfg</td>
<td>7:6</td>
<td>ped[115]</td>
<td>reset value</td>
</tr>
<tr>
<td>'h189</td>
<td>ped116_cfg</td>
<td>7:6</td>
<td>ped[116]</td>
<td>reset value</td>
</tr>
<tr>
<td>'h18A</td>
<td>ped117_cfg</td>
<td>7:6</td>
<td>ped[117]</td>
<td>reset value</td>
</tr>
<tr>
<td>'h18B</td>
<td>ped118_cfg</td>
<td>7:6</td>
<td>ped[118]</td>
<td>reset value</td>
</tr>
<tr>
<td>'h18C</td>
<td>ped119_cfg</td>
<td>7:6</td>
<td>ped[119]</td>
<td>reset value</td>
</tr>
<tr>
<td>'h18D</td>
<td>ped120_cfg</td>
<td>7:6</td>
<td>ped[120]</td>
<td>reset value</td>
</tr>
<tr>
<td>'h18E</td>
<td>ped121_cfg</td>
<td>7:6</td>
<td>ped[121]</td>
<td>reset value</td>
</tr>
<tr>
<td>'h18F</td>
<td>ped122_cfg</td>
<td>7:6</td>
<td>ped[122]</td>
<td>reset value</td>
</tr>
<tr>
<td>'h190</td>
<td>ped123_cfg</td>
<td>7:6</td>
<td>ped[123]</td>
<td>reset value</td>
</tr>
<tr>
<td>'h191</td>
<td>ped124_cfg</td>
<td>7:6</td>
<td>ped[124]</td>
<td>reset value</td>
</tr>
<tr>
<td>'h192</td>
<td>ped125_cfg</td>
<td>7:6</td>
<td>ped[125]</td>
<td>reset value</td>
</tr>
</tbody>
</table>

Continued on next page
<table>
<thead>
<tr>
<th>Addr</th>
<th>Name</th>
<th>Bit</th>
<th>Bit Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>7:6</td>
<td>—</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td></td>
<td>Pedestal value for channel 125</td>
</tr>
<tr>
<td>'h193</td>
<td>ped126_cfg</td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>—</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7:6</td>
<td>ped[126]</td>
<td>Pedestal value for channel 126</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>'h194</td>
<td>ped127_cfg</td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>—</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7:6</td>
<td>ped[127]</td>
<td>Pedestal value for channel 127</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>'h195</td>
<td>zs_channels_cfg</td>
<td>7:6</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>—</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7:6</td>
<td>zs_channels</td>
<td>Maximum number of channels in ZS package</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>'h196</td>
<td>sync0_cfg</td>
<td>7:0</td>
<td>reset value</td>
<td>'hAA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>sync_pattern[7:0]</td>
<td>lower part of constant value in</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Sync data packet</td>
</tr>
<tr>
<td>'h197</td>
<td>sync1_cfg</td>
<td>7:0</td>
<td>reset value</td>
<td>'h8C</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>sync_reversed</td>
<td>Exchange BXID and</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>sync_pattern in Sync data packet</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6:4</td>
<td>—</td>
<td>upper for bits of constant value in</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3:0</td>
<td>sync_pattern[11:8]</td>
<td>Sync data packet</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>'h198</td>
<td>dsp_seu_cnt_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>dsp_seu_counter</td>
<td>SEU counter for DSP register block</td>
</tr>
<tr>
<td>'h199</td>
<td>dsp_overflow_reg</td>
<td>7:2</td>
<td>reset value</td>
<td>'h00 (read-only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1:0</td>
<td>bxid_cnt_overflow_snap</td>
<td>bxid counter overflow snapshot</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>bxid_cnt_overflow</td>
<td>bxid counter overflow bit</td>
</tr>
<tr>
<td>'h19A</td>
<td>bxid_cnt0_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>'h00 (read-only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>bxid_cnt[7:0]</td>
<td>BXID counter</td>
</tr>
<tr>
<td>'h19B</td>
<td>bxid_cnt1_reg</td>
<td>7:4</td>
<td>reset value</td>
<td>'h00 (read-only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3:0</td>
<td>bxid_cnt[11:8]</td>
<td>BXID counter</td>
</tr>
<tr>
<td>'h19C</td>
<td>bxid_cnt0_snap_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>'h00 (read-only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>bxid_cnt_snap[7:0]</td>
<td>BXID counter snapshot</td>
</tr>
<tr>
<td>'h19D</td>
<td>bxid_cnt1_snap_reg</td>
<td>7:4</td>
<td>reset value</td>
<td>'h00 (read-only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3:0</td>
<td>bxid_cnt_snap[11:8]</td>
<td>BXID counter snapshot</td>
</tr>
</tbody>
</table>
Table 18: Memory subsystem configuration registers and counters

<table>
<thead>
<tr>
<th>Addr</th>
<th>Name</th>
<th>Bit</th>
<th>Bit Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>'h500</td>
<td>mem_pack_cfg</td>
<td>7:2</td>
<td>reset value</td>
<td>'h00 Reserved&lt;br&gt;Available packet types:&lt;br&gt;• 'b00 – all (normal memory operation)&lt;br&gt;• 'b01 – Idle only</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1:0</td>
<td>mem_packet_types</td>
<td></td>
</tr>
<tr>
<td>'h501</td>
<td></td>
<td>7:0</td>
<td>reset value</td>
<td>'h0F Reserved</td>
</tr>
<tr>
<td>'h502</td>
<td></td>
<td>7:0</td>
<td>reset value</td>
<td>'h99 Reserved</td>
</tr>
<tr>
<td>'h503</td>
<td></td>
<td>7:0</td>
<td>reset value</td>
<td>'h55 Reserved</td>
</tr>
<tr>
<td>'h504</td>
<td></td>
<td>7:0</td>
<td>reset value</td>
<td>'hAA Reserved</td>
</tr>
<tr>
<td>'h505</td>
<td></td>
<td>7:0</td>
<td>reset value</td>
<td>'hC Reserved</td>
</tr>
<tr>
<td>'h506</td>
<td>elinks_cfg</td>
<td>7:2</td>
<td>reset value</td>
<td>'h00 Reserved&lt;br&gt;Additional active e-links (over 3)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1:0</td>
<td>add_elinks</td>
<td></td>
</tr>
<tr>
<td>'h507</td>
<td>idle_cfg</td>
<td>7:5</td>
<td>reset value</td>
<td>'h06 Reserved&lt;br&gt;Number of Idle packets in group</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4:0</td>
<td>idle_group_size</td>
<td></td>
</tr>
<tr>
<td>'h508</td>
<td>mem_seu_cnt_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>'h00 SEU counter for memory register block</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>mem_seu_counter</td>
<td></td>
</tr>
<tr>
<td>'h509</td>
<td>mem_space0_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>'h00 (read-only)&lt;br&gt;Memory space value</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>mem_space[7:0]</td>
<td></td>
</tr>
<tr>
<td>'h50A</td>
<td>mem_space1_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>'h00 (read-only)&lt;br&gt;Memory space value</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>mem_space[15:8]</td>
<td></td>
</tr>
<tr>
<td>'h50B</td>
<td>mem_space0_snap_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>'h00 (read-only)&lt;br&gt;Memory space snapshot</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>mem_space_snap[7:0]</td>
<td></td>
</tr>
<tr>
<td>'h50C</td>
<td>mem_space1_snap_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>'h00 (read-only)&lt;br&gt;Memory space snapshot</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>mem_space_snap[15:8]</td>
<td></td>
</tr>
<tr>
<td>'h50D</td>
<td>idle_cnt0_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>'h00 (read-only)</td>
</tr>
</tbody>
</table>

Continued on next page
<table>
<thead>
<tr>
<th>Addr</th>
<th>Name</th>
<th>Bit</th>
<th>Bit Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>'h50E</td>
<td>idle_cnt1_reg</td>
<td>7:0</td>
<td>idle_cnt[7:0]</td>
<td>Idle packet counter</td>
</tr>
<tr>
<td></td>
<td>'h00 (read-only)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>'h50F</td>
<td>idle_cnt2_reg</td>
<td>7:0</td>
<td>idle_cnt[15:8]</td>
<td>Idle packet counter</td>
</tr>
<tr>
<td>'h510</td>
<td>idle_cnt3_reg</td>
<td>7:0</td>
<td>idle_cnt[23:16]</td>
<td>Idle packet counter</td>
</tr>
<tr>
<td>'h510</td>
<td>idle_cnt3_reg</td>
<td>7:0</td>
<td>idle_cnt[31:24]</td>
<td>Idle packet counter</td>
</tr>
<tr>
<td>'h511</td>
<td>idle_cnt4_reg</td>
<td>7:0</td>
<td>idle_cnt[39:32]</td>
<td>Idle packet counter</td>
</tr>
<tr>
<td>'h512</td>
<td>idle_cnt4_reg</td>
<td>7:0</td>
<td>idle_cnt[47:40]</td>
<td>Idle packet counter</td>
</tr>
<tr>
<td>'h513</td>
<td>idle_cnt0_snap_reg</td>
<td>7:0</td>
<td>idle_cnt_snap[7:0]</td>
<td>Idle counter snapshot</td>
</tr>
<tr>
<td>'h514</td>
<td>idle_cnt1_snap_reg</td>
<td>7:0</td>
<td>idle_cnt_snap[15:8]</td>
<td>Idle counter snapshot</td>
</tr>
<tr>
<td>'h515</td>
<td>idle_cnt2_snap_reg</td>
<td>7:0</td>
<td>idle_cnt_snap[23:16]</td>
<td>Idle counter snapshot</td>
</tr>
<tr>
<td>'h516</td>
<td>idle_cnt3_snap_reg</td>
<td>7:0</td>
<td>idle_cnt_snap[31:24]</td>
<td>Idle counter snapshot</td>
</tr>
<tr>
<td>'h517</td>
<td>idle_cnt4_snap_reg</td>
<td>7:0</td>
<td>idle_cnt_snap[39:32]</td>
<td>Idle counter snapshot</td>
</tr>
<tr>
<td>'h518</td>
<td>idle_cnt5_snap_reg</td>
<td>7:0</td>
<td>idle_cnt_snap[47:40]</td>
<td>Idle counter snapshot</td>
</tr>
</tbody>
</table>
### A.5. Serializer registers

Table 19: Serializer configuration registers

<table>
<thead>
<tr>
<th>Addr</th>
<th>Name</th>
<th>Bit</th>
<th>Bit Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>'h000</td>
<td>ser_source_cfg</td>
<td>reset value</td>
<td>'h22 = 'b0010_0_010</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>7:6</td>
<td>—</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:4</td>
<td>counter_mode</td>
<td>Internal counter mode:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• 'b00 – pseudorandom sequence 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• 'b01 – pseudorandom sequence 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• 'b10 – count up (+1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• 'b11 – count down (−1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>short_loop</td>
<td>TFC deserializer output connected directly to serializers in <em>data_clock</em> domain. Only if this bit is zero the field <em>serial_input_sel</em> is taken into account.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2:0</td>
<td>serial_input_sel</td>
<td>Selection of serializer input:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• 'b000 – DSP data</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• 'b001 – TFC deserializer output sent through <em>main_clock</em> domain</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• 'b010 – pattern register</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• 'b011 – internal counters</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• 'b100 – TFC deserializer output after FIFO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• 'b101 – TFC command after whole distributed FIFO in DSP block</td>
</tr>
<tr>
<td>'h001</td>
<td>pattern_cfg</td>
<td>reset value</td>
<td>'hAA = 'b1010_1010</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>7:0</td>
<td>—</td>
<td>Pattern value</td>
</tr>
<tr>
<td>'h002</td>
<td>deser_cfg</td>
<td>reset value</td>
<td>'h02 = 'b0000_000_10</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>7:5</td>
<td>—</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4:2</td>
<td>deser_byte_start</td>
<td>Start point of a byte (in deserializer) in relation to rising edge of main clock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1:0</td>
<td><em>data_clk_sel</em>[1:0]</td>
<td>First synchronizer clock edge selection; 0 is rising edge</td>
</tr>
<tr>
<td>'h003</td>
<td>pll_clk_cfg</td>
<td>reset value</td>
<td>'h08</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>7:4</td>
<td><em>pll_clk_sel</em>[1]</td>
<td>Pll clock phase selection for input synchronizer 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3:0</td>
<td><em>pll_clk_sel</em>[0]</td>
<td>Pll clock phase selection for input synchronizer 0</td>
</tr>
<tr>
<td>'h004</td>
<td>pll_main_cfg</td>
<td>reset value</td>
<td>'h0D = 'b0000_1101</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>7</td>
<td><em>pll_enable</em></td>
<td>Turn on PLL when set to 1</td>
</tr>
</tbody>
</table>

Continued on next page
Table 19 – continued from previous page

<table>
<thead>
<tr>
<th>Addr</th>
<th>Name</th>
<th>Bit</th>
<th>Bit Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>pll_connect</td>
<td>Connect data_clk after PLL start-up</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:4</td>
<td>—</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3:2</td>
<td>pll_gain</td>
<td>PLL gain</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1:0</td>
<td>—</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h005</td>
<td>pll_cp_cfg</td>
<td>6</td>
<td>—</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7</td>
<td>—</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6:0</td>
<td>pll_cp_cur</td>
<td>Charge pump current selection 0–40(\mu)A</td>
</tr>
<tr>
<td>'h006</td>
<td>pll_vco_cfg</td>
<td>6</td>
<td>—</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7</td>
<td>—</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6:0</td>
<td>pll_vco_cur</td>
<td>VCO current selection</td>
</tr>
<tr>
<td>'h007</td>
<td>tfc_fifo_cfg</td>
<td>7:0</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7:0</td>
<td>tfc_fifo_len</td>
<td>Current TFC FIFO delay</td>
</tr>
<tr>
<td>'h008</td>
<td>ser_g_cfg</td>
<td>7:3</td>
<td>—</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7:3</td>
<td>ser_byte_start</td>
<td>Start point of a byte in relation to rise edge of main clock</td>
</tr>
<tr>
<td>'h009</td>
<td>calib_fifo_cfg</td>
<td>7:0</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7:0</td>
<td>calib_fifo_len</td>
<td>Current calibration FIFO delay</td>
</tr>
<tr>
<td>'h00A</td>
<td>ser_seu_cnt_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>'h00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7:0</td>
<td>ser_seu_counter</td>
<td>SEU counter for serialiser register block</td>
</tr>
</tbody>
</table>
## A.6. Other configuration registers

Table 20: Other configuration registers

<table>
<thead>
<tr>
<th>Addr</th>
<th>Name</th>
<th>Bit</th>
<th>Bit Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>'h300</td>
<td>others_g_cfg</td>
<td>7</td>
<td>dll_hlp</td>
<td>reset value (0\text{–}\text{ MAIN_CLK} connected, \text{ 1– multiplexer outputs connected})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>dll_start</td>
<td>Start DLL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>dll_connect</td>
<td>source selection for (adc_clk &amp; calib_clk):</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(\bullet 0\text{–}\text{ MAIN_CLK} connected)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(\bullet 1\text{–}\text{ multiplexer outputs connected})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>slvs_termination</td>
<td>connect termination (100 Ω) to SLVS receivers</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>adc_mon_off</td>
<td>turn off monitoring ADCs: Band-gap, DLL, PLL, Band-gap, and Temperature</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>adc_test_off</td>
<td>mask out (turn off) test ADCs and the calibration impulse</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1:0</td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>'h301</td>
<td>dll_vcdl_cfg</td>
<td>7</td>
<td></td>
<td>reset value (h39)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6:0</td>
<td>dll_vcdl_cur</td>
<td>Delay line biasing current 0–40µA</td>
</tr>
<tr>
<td>'h302</td>
<td>dll_cp_cfg</td>
<td>7</td>
<td></td>
<td>reset value (h1A)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6:0</td>
<td>dll_cp_cur</td>
<td>Charge pump current 0–40µA</td>
</tr>
<tr>
<td>'h303</td>
<td>adc_clk_cfg</td>
<td>7:6</td>
<td></td>
<td>reset value (h00)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>adc_clk_sel</td>
<td>Reserved DLL clock phase selection for (adc_clk)</td>
</tr>
<tr>
<td>'h304</td>
<td>calib_clk_cfg</td>
<td>7:6</td>
<td></td>
<td>reset value (h00)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>calib_clk_sel</td>
<td>Reserved DLL clock phase selection for (calib_clk)</td>
</tr>
<tr>
<td>'h305</td>
<td>calib_main_cfg</td>
<td>7</td>
<td>calib_inv</td>
<td>invert calibration impulse</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>calib_neg_sync</td>
<td>synchronisation between (main_clk) and (calib_clk) is performed on falling edge</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>Addr</td>
<td>Name</td>
<td>Bit</td>
<td>Bit Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>--------------</td>
<td>-----</td>
<td>-------------------</td>
<td>-----------------------------------------------------</td>
</tr>
<tr>
<td>'h306</td>
<td>calib_volt_cfg</td>
<td>7:6</td>
<td>calib_enable[7]</td>
<td>Enable bit for channel 7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6:5</td>
<td>calib_enable[6]</td>
<td>Enable bit for channel 6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>calib_enable[0]</td>
<td>Enable bit for channel 0</td>
</tr>
<tr>
<td>'h307</td>
<td>calib_enable0_cfg</td>
<td>7</td>
<td>calib_enable[7]</td>
<td>Enable bit for channel 7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>calib_enable[6]</td>
<td>Enable bit for channel 6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>calib_enable[0]</td>
<td>Enable bit for channel 0</td>
</tr>
<tr>
<td>'h308</td>
<td>calib_enable1_cfg</td>
<td>7</td>
<td>calib_enable[15]</td>
<td>Enable bit for channel 15</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>calib_enable[14]</td>
<td>Enable bit for channel 14</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>calib_enable[8]</td>
<td>Enable bit for channel 8</td>
</tr>
<tr>
<td>'h309</td>
<td>calib_enable2_cfg</td>
<td>7</td>
<td>calib_enable[23]</td>
<td>Enable bit for channel 23</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>calib_enable[22]</td>
<td>Enable bit for channel 22</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>calib_enable[16]</td>
<td>Enable bit for channel 16</td>
</tr>
<tr>
<td>'h30A</td>
<td>calib_enable3_cfg</td>
<td>7</td>
<td>calib_enable[31]</td>
<td>Enable bit for channel 31</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>calib_enable[30]</td>
<td>Enable bit for channel 30</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>calib_enable[24]</td>
<td>Enable bit for channel 24</td>
</tr>
<tr>
<td>'h30B</td>
<td>calib_enable4_cfg</td>
<td>7</td>
<td>calib_enable[39]</td>
<td>Enable bit for channel 39</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>calib_enable[38]</td>
<td>Enable bit for channel 38</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>calib_enable[32]</td>
<td>Enable bit for channel 32</td>
</tr>
<tr>
<td>'h30C</td>
<td>calib_enable5_cfg</td>
<td>7</td>
<td>calib_enable[47]</td>
<td>Enable bit for channel 47</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>calib_enable[46]</td>
<td>Enable bit for channel 46</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>calib_enable[40]</td>
<td>Enable bit for channel 40</td>
</tr>
<tr>
<td>'h30D</td>
<td>calib_enable6_cfg</td>
<td>7</td>
<td>calib_enable[55]</td>
<td>Enable bit for channel 55</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>calib_enable[54]</td>
<td>Enable bit for channel 54</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>calib_enable[48]</td>
<td>Enable bit for channel 48</td>
</tr>
</tbody>
</table>

Continued on next page
<table>
<thead>
<tr>
<th>Addr</th>
<th>Name</th>
<th>Bit</th>
<th>Bit Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>'h30E</td>
<td>calib_enable7Cfg</td>
<td>7</td>
<td>calib_enable[63]</td>
<td>Enable bit for channel 63</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>calib_enable[62]</td>
<td>Enable bit for channel 62</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>...</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>calib_enable[56]</td>
<td>Enable bit for channel 56</td>
</tr>
<tr>
<td>'h30F</td>
<td>calib_enable8Cfg</td>
<td>7</td>
<td>calib_enable[71]</td>
<td>Enable bit for channel 71</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>calib_enable[70]</td>
<td>Enable bit for channel 70</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>...</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>calib_enable[64]</td>
<td>Enable bit for channel 64</td>
</tr>
<tr>
<td>'h310</td>
<td>calib_enable9Cfg</td>
<td>7</td>
<td>calib_enable[79]</td>
<td>Enable bit for channel 79</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>calib_enable[78]</td>
<td>Enable bit for channel 78</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>...</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>calib_enable[72]</td>
<td>Enable bit for channel 72</td>
</tr>
<tr>
<td>'h311</td>
<td>calib_enable10Cfg</td>
<td>7</td>
<td>calib_enable[87]</td>
<td>Enable bit for channel 87</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>calib_enable[86]</td>
<td>Enable bit for channel 86</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>...</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>calib_enable[80]</td>
<td>Enable bit for channel 80</td>
</tr>
<tr>
<td>'h312</td>
<td>calib_enable11Cfg</td>
<td>7</td>
<td>calib_enable[95]</td>
<td>Enable bit for channel 95</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>calib_enable[94]</td>
<td>Enable bit for channel 94</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>...</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>calib_enable[88]</td>
<td>Enable bit for channel 88</td>
</tr>
<tr>
<td>'h313</td>
<td>calib_enable12Cfg</td>
<td>7</td>
<td>calib_enable[103]</td>
<td>Enable bit for channel 103</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>calib_enable[102]</td>
<td>Enable bit for channel 102</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>...</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>calib_enable[96]</td>
<td>Enable bit for channel 96</td>
</tr>
<tr>
<td>'h314</td>
<td>calib_enable13Cfg</td>
<td>7</td>
<td>calib_enable[111]</td>
<td>Enable bit for channel 111</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>calib_enable[110]</td>
<td>Enable bit for channel 110</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>...</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>calib_enable[104]</td>
<td>Enable bit for channel 104</td>
</tr>
<tr>
<td>'h315</td>
<td>calib_enable14Cfg</td>
<td>7</td>
<td>calib_enable[119]</td>
<td>Enable bit for channel 119</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>calib_enable[118]</td>
<td>Enable bit for channel 118</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>...</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>calib_enable[112]</td>
<td>Enable bit for channel 112</td>
</tr>
</tbody>
</table>

Continued on next page
<table>
<thead>
<tr>
<th>Addr</th>
<th>Name</th>
<th>Bit</th>
<th>Bit Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>'h316</td>
<td>calib_enable15_cfg</td>
<td>7</td>
<td>calib_enable[127]</td>
<td>Enable bit for channel 127</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>calib_enable[126]</td>
<td>Enable bit for channel 126</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>calib_enable[120]</td>
<td>Enable bit for channel 120</td>
</tr>
<tr>
<td>'h317</td>
<td>slvs_cur_cfg</td>
<td>7:5</td>
<td>slvs_cur_dac</td>
<td>SLVS biasing current</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4:0</td>
<td>—</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h318</td>
<td>slvs_vcm_cfg</td>
<td>7:5</td>
<td>slvs_vcm_dac</td>
<td>SLVS common voltage</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4:0</td>
<td>—</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h319</td>
<td></td>
<td>7:0</td>
<td>—</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h31A</td>
<td></td>
<td>7:0</td>
<td>—</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h31B</td>
<td>vcm_cur_cfg</td>
<td>7:6</td>
<td>vcm_cur</td>
<td>Biasing DAC for vcm signal</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>—</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h31C</td>
<td></td>
<td>7:0</td>
<td>—</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h31D</td>
<td></td>
<td>7:0</td>
<td>—</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h31E</td>
<td></td>
<td>7:0</td>
<td>—</td>
<td>Reserved</td>
</tr>
<tr>
<td>'h31F</td>
<td>mon_cfg</td>
<td>7:6</td>
<td>pll_mon_sel</td>
<td>Select monitored PLL:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:4</td>
<td>—</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3:2</td>
<td>dac1_mon_sel</td>
<td>Select DAC connected to dac1_voltage in dac1_mon register:</td>
</tr>
</tbody>
</table>

Continued on next page
<table>
<thead>
<tr>
<th>Addr</th>
<th>Name</th>
<th>Bit</th>
<th>Bit Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>'h320</td>
<td>other_seu_cnt_reg</td>
<td>7:0</td>
<td>other_seu_counter</td>
<td>reset value 'h00 SEU counter for other register block</td>
</tr>
<tr>
<td>'h321</td>
<td>dll_vcdl_mon</td>
<td>7</td>
<td>dll_cur_ok</td>
<td>reset value ‘h00 (read-only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>—</td>
<td>1 if VCDL current OK when dll_start=0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>dll_vcdl_voltage</td>
<td>reset value Delay line control voltage $V_{DLL}$ (2’s complement, where zero is half of supply voltage)</td>
</tr>
<tr>
<td>'h322</td>
<td>pll_vco_mon</td>
<td>7:6</td>
<td>—</td>
<td>reset value 'h00 (read-only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>pll_vco_voltage</td>
<td>reset value PLL VCO control voltage $V_{PLL}$ (2’s complement, where zero is half of supply voltage)</td>
</tr>
<tr>
<td>'h323</td>
<td>adc_test0_reg</td>
<td>7:6</td>
<td>—</td>
<td>reset value 'h00 (read-only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>adc_test0_value</td>
<td>reset value Test ADC laying before channel 0 in the layout (2’s complement, where zero is half of supply voltage)</td>
</tr>
<tr>
<td>'h324</td>
<td>adc_test1_reg</td>
<td>7:6</td>
<td>—</td>
<td>reset value 'h00 (read-only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>adc_test1_value</td>
<td>reset value Test ADC laying after channel 127 in the layout (2’s complement, where zero is half of supply voltage)</td>
</tr>
<tr>
<td>'h325</td>
<td>dac0_mon</td>
<td>7:6</td>
<td>—</td>
<td>reset value 'h00 (read-only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>dac0_voltage</td>
<td>reset value Value of first monitoring ADC connected to biasing DAC (2’s complement, where zero is half of supply voltage) – see register mon_cfg</td>
</tr>
<tr>
<td>'h326</td>
<td>dac1_mon</td>
<td>reset value</td>
<td>'h00 (read-only)</td>
<td></td>
</tr>
</tbody>
</table>

Continued on next page
<table>
<thead>
<tr>
<th>Addr</th>
<th>Name</th>
<th>Bit</th>
<th>Bit Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>7:6</td>
<td>—</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5:0</td>
<td>dac1_voltage</td>
<td>Value of second monitoring ADC connected to biasing DAC (2’s complement, where zero is half of supply voltage) – see register mon_cfg</td>
</tr>
</tbody>
</table>
### A.7. TFC counters

<table>
<thead>
<tr>
<th>Addr</th>
<th>Name</th>
<th>Bit</th>
<th>Bit Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>'h400</td>
<td>tfc_seu_cnt_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>'h00 SEU counter for TFC counters block</td>
</tr>
<tr>
<td></td>
<td>tfc_seu_cnt</td>
<td></td>
<td>tfc_seu_counter</td>
<td>SEU counter for TFC counters block</td>
</tr>
<tr>
<td>'h401</td>
<td>tfc_overflow_reg</td>
<td></td>
<td>reset value</td>
<td>'h00 Calib counter overflow bit Synch counter overflow bit Snapshot counter overflow bit BxVeto counter overflow bit NZS counter overflow bit Header counter overflow bit FEReset counter overflow bit BXReset counter overflow bit</td>
</tr>
<tr>
<td></td>
<td>tfc_overflow</td>
<td></td>
<td>calib_overflow</td>
<td>Calib counter overflow bit Synch counter overflow bit Synch counter overflow bit Snapshot counter overflow bit BxVeto counter overflow bit NZS counter overflow bit Header counter overflow bit FEReset counter overflow bit BXReset counter overflow bit</td>
</tr>
<tr>
<td></td>
<td>synch_overflow</td>
<td></td>
<td>synch_overflow</td>
<td>Calib counter overflow bit Synch counter overflow bit Synch counter overflow bit Snapshot counter overflow bit BxVeto counter overflow bit NZS counter overflow bit Header counter overflow bit FEReset counter overflow bit BXReset counter overflow bit</td>
</tr>
<tr>
<td></td>
<td>snapshot</td>
<td></td>
<td>snapshot</td>
<td>Calib counter overflow bit Synch counter overflow bit Synch counter overflow bit Snapshot counter overflow bit BxVeto counter overflow bit NZS counter overflow bit Header counter overflow bit FEReset counter overflow bit BXReset counter overflow bit</td>
</tr>
<tr>
<td></td>
<td>bxveto</td>
<td></td>
<td>bxveto</td>
<td>Calib counter overflow bit Synch counter overflow bit Synch counter overflow bit Snapshot counter overflow bit BxVeto counter overflow bit NZS counter overflow bit Header counter overflow bit FEReset counter overflow bit BXReset counter overflow bit</td>
</tr>
<tr>
<td></td>
<td>nzs</td>
<td></td>
<td>nzs</td>
<td>Calib counter overflow bit Synch counter overflow bit Synch counter overflow bit Snapshot counter overflow bit BxVeto counter overflow bit NZS counter overflow bit Header counter overflow bit FEReset counter overflow bit BXReset counter overflow bit</td>
</tr>
<tr>
<td></td>
<td>header</td>
<td></td>
<td>header</td>
<td>Calib counter overflow bit Synch counter overflow bit Synch counter overflow bit Snapshot counter overflow bit BxVeto counter overflow bit NZS counter overflow bit Header counter overflow bit FEReset counter overflow bit BXReset counter overflow bit</td>
</tr>
<tr>
<td></td>
<td>fereset</td>
<td></td>
<td>fereset</td>
<td>Calib counter overflow bit Synch counter overflow bit Synch counter overflow bit Snapshot counter overflow bit BxVeto counter overflow bit NZS counter overflow bit Header counter overflow bit FEReset counter overflow bit BXReset counter overflow bit</td>
</tr>
<tr>
<td></td>
<td>bxreset</td>
<td></td>
<td>bxreset</td>
<td>Calib counter overflow bit Synch counter overflow bit Synch counter overflow bit Snapshot counter overflow bit BxVeto counter overflow bit NZS counter overflow bit Header counter overflow bit FEReset counter overflow bit BXReset counter overflow bit</td>
</tr>
<tr>
<td>'h402</td>
<td>tfc_overflow_snap_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>'h00 (read-only) Snapshot of tfc_overflow_reg register</td>
</tr>
<tr>
<td></td>
<td>tfc_overflow_snap</td>
<td></td>
<td>tfc_overflow_snap</td>
<td>Calib counter overflow bit Synch counter overflow bit Synch counter overflow bit Snapshot counter overflow bit BxVeto counter overflow bit NZS counter overflow bit Header counter overflow bit FEReset counter overflow bit BXReset counter overflow bit</td>
</tr>
<tr>
<td>'h403</td>
<td>calib_c0_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>'h00 (read-only) Calib TFC command counter</td>
</tr>
<tr>
<td></td>
<td>calib_c0</td>
<td></td>
<td>calib_c0[7:0]</td>
<td>Calib TFC command counter</td>
</tr>
<tr>
<td>'h404</td>
<td>calib_c1_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>'h00 (read-only) Calib TFC command counter</td>
</tr>
<tr>
<td></td>
<td>calib_c1</td>
<td></td>
<td>calib_c1[15:8]</td>
<td>Calib TFC command counter</td>
</tr>
<tr>
<td>'h405</td>
<td>calib_c2_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>'h00 (read-only) Calib TFC command counter</td>
</tr>
<tr>
<td></td>
<td>calib_c2</td>
<td></td>
<td>calib_c2[23:16]</td>
<td>Calib TFC command counter</td>
</tr>
<tr>
<td>'h406</td>
<td>calib_c3_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>'h00 (read-only) Calib TFC command counter</td>
</tr>
<tr>
<td></td>
<td>calib_c3</td>
<td></td>
<td>calib_c3[31:24]</td>
<td>Calib TFC command counter</td>
</tr>
<tr>
<td>'h407</td>
<td>calib_c0_snap_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>'h00 (read-only) Calib TFC counter snapshot</td>
</tr>
<tr>
<td></td>
<td>calib_c0_snap</td>
<td></td>
<td>calib_c0_snap[7:0]</td>
<td>Calib TFC counter snapshot</td>
</tr>
<tr>
<td>'h408</td>
<td>calib_c1_snap_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>'h00 (read-only) Calib TFC counter snapshot</td>
</tr>
<tr>
<td></td>
<td>calib_c1_snap</td>
<td></td>
<td>calib_c1_snap[15:8]</td>
<td>Calib TFC counter snapshot</td>
</tr>
<tr>
<td>'h409</td>
<td>calib_c2_snap_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>'h00 (read-only) Calib TFC counter snapshot</td>
</tr>
<tr>
<td></td>
<td>calib_c2_snap</td>
<td></td>
<td>calib_c2_snap[23:16]</td>
<td>Calib TFC counter snapshot</td>
</tr>
<tr>
<td>'h40A</td>
<td>calib_c3_snap_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>'h00 (read-only) Calib TFC counter snapshot</td>
</tr>
<tr>
<td></td>
<td>calib_c3_snap</td>
<td></td>
<td>calib_c3_snap[31:24]</td>
<td>Calib TFC counter snapshot</td>
</tr>
<tr>
<td>'h40B</td>
<td>synch_c0_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>'h00 (read-only) Synch TFC command counter</td>
</tr>
<tr>
<td></td>
<td>synch_c0</td>
<td></td>
<td>synch_c0[7:0]</td>
<td>Synch TFC command counter</td>
</tr>
<tr>
<td>'h40C</td>
<td>synch_c1_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>'h00 (read-only) Synch TFC command counter</td>
</tr>
<tr>
<td></td>
<td>synch_c1</td>
<td></td>
<td>synch_c1[15:8]</td>
<td>Synch TFC command counter</td>
</tr>
<tr>
<td>'h40D</td>
<td>synch_c2_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>'h00 (read-only) Synch TFC command counter</td>
</tr>
<tr>
<td></td>
<td>synch_c2</td>
<td></td>
<td>synch_c2[23:16]</td>
<td>Synch TFC command counter</td>
</tr>
</tbody>
</table>

Continued on next page
### Table 21 – continued from previous page

<table>
<thead>
<tr>
<th>Addr</th>
<th>Name</th>
<th>Bit</th>
<th>Bit Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>'h40E</td>
<td>synch_cnt3_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>synch_cnt[31:24]</td>
</tr>
<tr>
<td>'h40F</td>
<td>synch_cnt0_snap_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>synch_cnt_snap[7:0]</td>
</tr>
<tr>
<td>'h410</td>
<td>synch_cnt1_snap_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>synch_cnt_snap[15:8]</td>
</tr>
<tr>
<td>'h411</td>
<td>synch_cnt2_snap_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>synch_cnt_snap[23:16]</td>
</tr>
<tr>
<td>'h412</td>
<td>synch_cnt3_snap_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>synch_cnt_snap[31:24]</td>
</tr>
<tr>
<td>'h413</td>
<td>snapshot_cnt0_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>snapshot_cnt[7:0]</td>
</tr>
<tr>
<td>'h414</td>
<td>snapshot_cnt1_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>snapshot_cnt[15:8]</td>
</tr>
<tr>
<td>'h415</td>
<td>snapshot_cnt2_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>snapshot_cnt[23:16]</td>
</tr>
<tr>
<td>'h416</td>
<td>snapshot_cnt3_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>snapshot_cnt[31:24]</td>
</tr>
<tr>
<td>'h417</td>
<td>snapshot_cnt0_snap_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>snapshot_cnt_snap[7:0]</td>
</tr>
<tr>
<td>'h418</td>
<td>snapshot_cnt1_snap_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>snapshot_cnt_snap[15:8]</td>
</tr>
<tr>
<td>'h419</td>
<td>snapshot_cnt2_snap_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>snapshot_cnt_snap[23:16]</td>
</tr>
<tr>
<td>'h41A</td>
<td>snapshot_cnt3_snap_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>snapshot_cnt_snap[31:24]</td>
</tr>
<tr>
<td>'h41B</td>
<td>bxveto_cnt0_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>bxveto_cnt[7:0]</td>
</tr>
<tr>
<td>'h41C</td>
<td>bxveto_cnt1_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>bxveto_cnt[15:8]</td>
</tr>
</tbody>
</table>

Continued on next page
<table>
<thead>
<tr>
<th>Addr</th>
<th>Name</th>
<th>Bit</th>
<th>Bit Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>h41D</td>
<td>bxveto_cnt2_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>h00 (read-only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>bxveto_cnt[23:16]</td>
<td>BxVeto TFC command counter</td>
</tr>
<tr>
<td>h41E</td>
<td>bxveto_cnt3_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>h00 (read-only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>bxveto_cnt[31:24]</td>
<td>BxVeto TFC command counter</td>
</tr>
<tr>
<td>h41F</td>
<td>bxveto_cnt4_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>h00 (read-only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>bxveto_cnt[39:32]</td>
<td>BxVeto TFC command counter</td>
</tr>
<tr>
<td>h420</td>
<td>bxveto_cnt4_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>h00 (read-only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>bxveto_cnt[47:40]</td>
<td>BxVeto TFC command counter</td>
</tr>
<tr>
<td>h421</td>
<td>bxveto_cnt0_snap_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>h00 (read-only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>bxveto_cnt_snap[7:0]</td>
<td>BxVeto TFC counter snapshot</td>
</tr>
<tr>
<td>h422</td>
<td>bxveto_cnt1_snap_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>h00 (read-only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>bxveto_cnt_snap[15:8]</td>
<td>BxVeto TFC counter snapshot</td>
</tr>
<tr>
<td>h423</td>
<td>bxveto_cnt2_snap_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>h00 (read-only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>bxveto_cnt_snap[23:16]</td>
<td>BxVeto TFC counter snapshot</td>
</tr>
<tr>
<td>h424</td>
<td>bxveto_cnt3_snap_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>h00 (read-only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>bxveto_cnt_snap[31:24]</td>
<td>BxVeto TFC counter snapshot</td>
</tr>
<tr>
<td>h425</td>
<td>bxveto_cnt4_snap_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>h00 (read-only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>bxveto_cnt_snap[39:32]</td>
<td>BxVeto TFC counter snapshot</td>
</tr>
<tr>
<td>h426</td>
<td>bxveto_cnt5_snap_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>h00 (read-only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>bxveto_cnt_snap[47:40]</td>
<td>BxVeto TFC counter snapshot</td>
</tr>
<tr>
<td>h427</td>
<td>nzs_cnt0_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>h00 (read-only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>nzs_cnt[7:0]</td>
<td>NZS TFC command counter</td>
</tr>
<tr>
<td>h428</td>
<td>nzs_cnt1_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>h00 (read-only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>nzs_cnt[15:8]</td>
<td>NZS TFC command counter</td>
</tr>
<tr>
<td>h429</td>
<td>nzs_cnt2_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>h00 (read-only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>nzs_cnt[23:16]</td>
<td>NZS TFC command counter</td>
</tr>
<tr>
<td>h42A</td>
<td>nzs_cnt3_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>h00 (read-only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>nzs_cnt[31:24]</td>
<td>NZS TFC command counter</td>
</tr>
<tr>
<td>h42B</td>
<td>nzs_cnt0_snap_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>h00 (read-only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>nzs_cnt_snap[7:0]</td>
<td>NZS TFC counter snapshot</td>
</tr>
<tr>
<td>h42C</td>
<td>nzs_cnt1_snap_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>h00 (read-only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>nzs_cnt_snap[15:8]</td>
<td>NZS TFC counter snapshot</td>
</tr>
<tr>
<td>h42D</td>
<td>nzs_cnt2_snap_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>h00 (read-only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>nzs_cnt_snap[23:16]</td>
<td>NZS TFC counter snapshot</td>
</tr>
<tr>
<td>h42E</td>
<td>nzs_cnt3_snap_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>h00 (read-only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>nzs_cnt_snap[31:24]</td>
<td>NZS TFC counter snapshot</td>
</tr>
</tbody>
</table>

Continued on next page
<table>
<thead>
<tr>
<th>Addr</th>
<th>Name</th>
<th>Bit</th>
<th>Bit Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>'h42F</td>
<td>header_cnt0_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>'h00 (read-only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>header_cnt[7:0]</td>
<td>Header TFC command counter</td>
</tr>
<tr>
<td>'h430</td>
<td>header_cnt1_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>'h00 (read-only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>header_cnt[15:8]</td>
<td>Header TFC command counter</td>
</tr>
<tr>
<td>'h431</td>
<td>header_cnt2_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>'h00 (read-only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>header_cnt[23:16]</td>
<td>Header TFC command counter</td>
</tr>
<tr>
<td>'h432</td>
<td>header_cnt3_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>'h00 (read-only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>header_cnt[31:24]</td>
<td>Header TFC command counter</td>
</tr>
<tr>
<td>'h433</td>
<td>header_cnt4_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>'h00 (read-only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>header_cnt[39:32]</td>
<td>Header TFC command counter</td>
</tr>
<tr>
<td>'h434</td>
<td>header_cnt4_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>'h00 (read-only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>header_cnt[47:40]</td>
<td>Header TFC command counter</td>
</tr>
<tr>
<td>'h435</td>
<td>header_cnt0_snap_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>'h00 (read-only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>header_cnt_snap[7:0]</td>
<td>Header TFC counter snapshot</td>
</tr>
<tr>
<td>'h436</td>
<td>header_cnt1Snap_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>'h00 (read-only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>header_cnt_snap[15:8]</td>
<td>Header TFC counter snapshot</td>
</tr>
<tr>
<td>'h437</td>
<td>header_cnt2_snap_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>'h00 (read-only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>header_cnt_snap[23:16]</td>
<td>Header TFC counter snapshot</td>
</tr>
<tr>
<td>'h438</td>
<td>header_cnt3_snap_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>'h00 (read-only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>header_cnt_snap[31:24]</td>
<td>Header TFC counter snapshot</td>
</tr>
<tr>
<td>'h439</td>
<td>header_cnt4_snap_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>'h00 (read-only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>header_cnt_snap[39:32]</td>
<td>Header TFC counter snapshot</td>
</tr>
<tr>
<td>'h43A</td>
<td>header_cnt5_snap_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>'h00 (read-only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>header_cnt_snap[47:40]</td>
<td>Header TFC counter snapshot</td>
</tr>
<tr>
<td>'h43B</td>
<td>fereset_cnt0_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>'h00 (read-only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>fereset_cnt[7:0]</td>
<td>FEReset TFC command counter</td>
</tr>
<tr>
<td>'h43C</td>
<td>fereset_cnt1_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>'h00 (read-only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>fereset_cnt[15:8]</td>
<td>FEReset TFC command counter</td>
</tr>
<tr>
<td>'h43D</td>
<td>fereset_cnt2_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>'h00 (read-only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>fereset_cnt[23:16]</td>
<td>FEReset TFC command counter</td>
</tr>
<tr>
<td>'h43E</td>
<td>fereset_cnt3_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>'h00 (read-only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>fereset_cnt[31:24]</td>
<td>FEReset TFC command counter</td>
</tr>
<tr>
<td>'h43F</td>
<td>fereset_cnt0_snap_reg</td>
<td>7:0</td>
<td>reset value</td>
<td>'h00 (read-only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>fereset_cnt_snap[7:0]</td>
<td>FEReset TFC counter snapshot</td>
</tr>
<tr>
<td>'h440</td>
<td>fereset_cnt1_snap_reg</td>
<td></td>
<td>reset value</td>
<td>'h00 (read-only)</td>
</tr>
</tbody>
</table>

Continued on next page
<table>
<thead>
<tr>
<th>Addr</th>
<th>Name</th>
<th>Bit</th>
<th>Bit Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>'h441</td>
<td>fereset_cnt2_snap_reg</td>
<td>7:0</td>
<td>fereset_cnt_snap[23:16]</td>
<td>'h00 (read-only) FEReset TFC counter snapshot</td>
</tr>
<tr>
<td>'h442</td>
<td>fereset_cnt3_snap_reg</td>
<td>7:0</td>
<td>fereset_cnt_snap[31:24]</td>
<td>'h00 (read-only) FEReset TFC counter snapshot</td>
</tr>
<tr>
<td>'h443</td>
<td>bxreset_cnt0_reg</td>
<td>7:0</td>
<td>bxreset_cnt[7:0]</td>
<td>'h00 (read-only) BXReset TFC command counter</td>
</tr>
<tr>
<td>'h444</td>
<td>bxreset_cnt1_reg</td>
<td>7:0</td>
<td>bxreset_cnt[15:8]</td>
<td>'h00 (read-only) BXReset TFC command counter</td>
</tr>
<tr>
<td>'h445</td>
<td>bxreset_cnt2_reg</td>
<td>7:0</td>
<td>bxreset_cnt[23:16]</td>
<td>'h00 (read-only) BXReset TFC command counter</td>
</tr>
<tr>
<td>'h446</td>
<td>bxreset_cnt3_reg</td>
<td>7:0</td>
<td>bxreset_cnt[31:24]</td>
<td>'h00 (read-only) BXReset TFC command counter</td>
</tr>
<tr>
<td>'h447</td>
<td>bxreset_cnt0_snap_reg</td>
<td>7:0</td>
<td>bxreset_cnt_snap[7:0]</td>
<td>'h00 (read-only) BXReset TFC counter snapshot</td>
</tr>
<tr>
<td>'h448</td>
<td>bxreset_cnt1_snap_reg</td>
<td>7:0</td>
<td>bxreset_cnt_snap[15:8]</td>
<td>'h00 (read-only) BXReset TFC counter snapshot</td>
</tr>
<tr>
<td>'h449</td>
<td>bxreset_cnt2_snap_reg</td>
<td>7:0</td>
<td>bxreset_cnt_snap[23:16]</td>
<td>'h00 (read-only) BXReset TFC counter snapshot</td>
</tr>
<tr>
<td>'h44A</td>
<td>bxreset_cnt3_snap_reg</td>
<td>7:0</td>
<td>bxreset_cnt_snap[31:24]</td>
<td>'h00 (read-only) BXReset TFC counter snapshot</td>
</tr>
</tbody>
</table>

B. Pseudo-random sequences
Table 22: Pseudo-random sequences; sequence 0 of length 256, sequence 1 of length 18

<table>
<thead>
<tr>
<th>Clock cycle</th>
<th>Sequence 0</th>
<th>Sequence 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>5</td>
<td>17</td>
<td>19</td>
</tr>
<tr>
<td>6</td>
<td>35</td>
<td>39</td>
</tr>
<tr>
<td>7</td>
<td>71</td>
<td>79</td>
</tr>
<tr>
<td>8</td>
<td>142</td>
<td>158</td>
</tr>
<tr>
<td>9</td>
<td>28</td>
<td>60</td>
</tr>
<tr>
<td>10</td>
<td>56</td>
<td>121</td>
</tr>
<tr>
<td>11</td>
<td>113</td>
<td>242</td>
</tr>
<tr>
<td>12</td>
<td>226</td>
<td>228</td>
</tr>
<tr>
<td>13</td>
<td>196</td>
<td>200</td>
</tr>
<tr>
<td>14</td>
<td>137</td>
<td>144</td>
</tr>
<tr>
<td>15</td>
<td>18</td>
<td>32</td>
</tr>
<tr>
<td>16</td>
<td>37</td>
<td>64</td>
</tr>
<tr>
<td>17</td>
<td>75</td>
<td>128</td>
</tr>
<tr>
<td>18</td>
<td>151</td>
<td>0</td>
</tr>
<tr>
<td>19</td>
<td>46</td>
<td>1</td>
</tr>
<tr>
<td>20</td>
<td>92</td>
<td>2</td>
</tr>
<tr>
<td>21</td>
<td>184</td>
<td>4</td>
</tr>
<tr>
<td>22</td>
<td>112</td>
<td>9</td>
</tr>
<tr>
<td>23</td>
<td>224</td>
<td>19</td>
</tr>
<tr>
<td>24</td>
<td>192</td>
<td>39</td>
</tr>
<tr>
<td>25</td>
<td>129</td>
<td>79</td>
</tr>
<tr>
<td>26</td>
<td>3</td>
<td>158</td>
</tr>
<tr>
<td>27</td>
<td>6</td>
<td>60</td>
</tr>
<tr>
<td>28</td>
<td>12</td>
<td>121</td>
</tr>
<tr>
<td>29</td>
<td>25</td>
<td>242</td>
</tr>
<tr>
<td>30</td>
<td>50</td>
<td>228</td>
</tr>
<tr>
<td>31</td>
<td>100</td>
<td>200</td>
</tr>
<tr>
<td>32</td>
<td>201</td>
<td>144</td>
</tr>
<tr>
<td>33</td>
<td>146</td>
<td>32</td>
</tr>
<tr>
<td>34</td>
<td>36</td>
<td>64</td>
</tr>
<tr>
<td>35</td>
<td>73</td>
<td>128</td>
</tr>
<tr>
<td>36</td>
<td>147</td>
<td>0</td>
</tr>
<tr>
<td>37</td>
<td>38</td>
<td>1</td>
</tr>
<tr>
<td>38</td>
<td>77</td>
<td>2</td>
</tr>
<tr>
<td>39</td>
<td>155</td>
<td>4</td>
</tr>
</tbody>
</table>

Continued on next page
<table>
<thead>
<tr>
<th>Clock cycle</th>
<th>Sequence 0</th>
<th>Sequence 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>55</td>
<td>9</td>
</tr>
<tr>
<td>41</td>
<td>110</td>
<td>19</td>
</tr>
<tr>
<td>42</td>
<td>220</td>
<td>30</td>
</tr>
<tr>
<td>43</td>
<td>185</td>
<td>79</td>
</tr>
<tr>
<td>44</td>
<td>114</td>
<td>158</td>
</tr>
<tr>
<td>45</td>
<td>228</td>
<td>60</td>
</tr>
<tr>
<td>46</td>
<td>200</td>
<td>121</td>
</tr>
<tr>
<td>47</td>
<td>144</td>
<td>242</td>
</tr>
<tr>
<td>48</td>
<td>32</td>
<td>228</td>
</tr>
<tr>
<td>49</td>
<td>65</td>
<td>200</td>
</tr>
<tr>
<td>50</td>
<td>130</td>
<td>144</td>
</tr>
<tr>
<td>51</td>
<td>5</td>
<td>32</td>
</tr>
<tr>
<td>52</td>
<td>10</td>
<td>64</td>
</tr>
<tr>
<td>53</td>
<td>21</td>
<td>128</td>
</tr>
<tr>
<td>54</td>
<td>43</td>
<td>0</td>
</tr>
<tr>
<td>55</td>
<td>86</td>
<td>1</td>
</tr>
<tr>
<td>56</td>
<td>173</td>
<td>2</td>
</tr>
<tr>
<td>57</td>
<td>91</td>
<td>4</td>
</tr>
<tr>
<td>58</td>
<td>182</td>
<td>9</td>
</tr>
<tr>
<td>59</td>
<td>109</td>
<td>19</td>
</tr>
<tr>
<td>60</td>
<td>218</td>
<td>39</td>
</tr>
<tr>
<td>61</td>
<td>181</td>
<td>79</td>
</tr>
<tr>
<td>62</td>
<td>107</td>
<td>158</td>
</tr>
<tr>
<td>63</td>
<td>214</td>
<td>60</td>
</tr>
<tr>
<td>64</td>
<td>172</td>
<td>121</td>
</tr>
<tr>
<td>65</td>
<td>89</td>
<td>242</td>
</tr>
<tr>
<td>66</td>
<td>178</td>
<td>228</td>
</tr>
<tr>
<td>67</td>
<td>101</td>
<td>200</td>
</tr>
<tr>
<td>68</td>
<td>203</td>
<td>144</td>
</tr>
<tr>
<td>69</td>
<td>150</td>
<td>32</td>
</tr>
<tr>
<td>70</td>
<td>44</td>
<td>64</td>
</tr>
<tr>
<td>71</td>
<td>88</td>
<td>128</td>
</tr>
<tr>
<td>72</td>
<td>176</td>
<td>0</td>
</tr>
<tr>
<td>73</td>
<td>97</td>
<td>1</td>
</tr>
<tr>
<td>74</td>
<td>195</td>
<td>2</td>
</tr>
<tr>
<td>75</td>
<td>135</td>
<td>4</td>
</tr>
<tr>
<td>76</td>
<td>15</td>
<td>9</td>
</tr>
<tr>
<td>77</td>
<td>31</td>
<td>19</td>
</tr>
<tr>
<td>78</td>
<td>62</td>
<td>39</td>
</tr>
<tr>
<td>79</td>
<td>125</td>
<td>79</td>
</tr>
<tr>
<td>80</td>
<td>251</td>
<td>158</td>
</tr>
</tbody>
</table>

Continued on next page
<table>
<thead>
<tr>
<th>Clock cycle</th>
<th>Sequence 0</th>
<th>Sequence 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>81</td>
<td>246</td>
<td>60</td>
</tr>
<tr>
<td>82</td>
<td>237</td>
<td>121</td>
</tr>
<tr>
<td>83</td>
<td>219</td>
<td>242</td>
</tr>
<tr>
<td>84</td>
<td>183</td>
<td>228</td>
</tr>
<tr>
<td>85</td>
<td>111</td>
<td>200</td>
</tr>
<tr>
<td>86</td>
<td>222</td>
<td>144</td>
</tr>
<tr>
<td>87</td>
<td>189</td>
<td>32</td>
</tr>
<tr>
<td>88</td>
<td>122</td>
<td>64</td>
</tr>
<tr>
<td>89</td>
<td>245</td>
<td>128</td>
</tr>
<tr>
<td>90</td>
<td>235</td>
<td>0</td>
</tr>
<tr>
<td>91</td>
<td>215</td>
<td>1</td>
</tr>
<tr>
<td>92</td>
<td>174</td>
<td>2</td>
</tr>
<tr>
<td>93</td>
<td>93</td>
<td>4</td>
</tr>
<tr>
<td>94</td>
<td>186</td>
<td>9</td>
</tr>
<tr>
<td>95</td>
<td>116</td>
<td>19</td>
</tr>
<tr>
<td>96</td>
<td>232</td>
<td>39</td>
</tr>
<tr>
<td>97</td>
<td>209</td>
<td>79</td>
</tr>
<tr>
<td>98</td>
<td>162</td>
<td>158</td>
</tr>
<tr>
<td>99</td>
<td>68</td>
<td>60</td>
</tr>
<tr>
<td>100</td>
<td>136</td>
<td>121</td>
</tr>
<tr>
<td>101</td>
<td>16</td>
<td>242</td>
</tr>
<tr>
<td>102</td>
<td>33</td>
<td>228</td>
</tr>
<tr>
<td>103</td>
<td>67</td>
<td>200</td>
</tr>
<tr>
<td>104</td>
<td>134</td>
<td>144</td>
</tr>
<tr>
<td>105</td>
<td>13</td>
<td>32</td>
</tr>
<tr>
<td>106</td>
<td>27</td>
<td>64</td>
</tr>
<tr>
<td>107</td>
<td>54</td>
<td>128</td>
</tr>
<tr>
<td>108</td>
<td>108</td>
<td>0</td>
</tr>
<tr>
<td>109</td>
<td>216</td>
<td>1</td>
</tr>
<tr>
<td>110</td>
<td>177</td>
<td>2</td>
</tr>
<tr>
<td>111</td>
<td>99</td>
<td>4</td>
</tr>
<tr>
<td>112</td>
<td>199</td>
<td>9</td>
</tr>
<tr>
<td>113</td>
<td>143</td>
<td>19</td>
</tr>
<tr>
<td>114</td>
<td>30</td>
<td>39</td>
</tr>
<tr>
<td>115</td>
<td>60</td>
<td>79</td>
</tr>
<tr>
<td>116</td>
<td>121</td>
<td>158</td>
</tr>
<tr>
<td>117</td>
<td>243</td>
<td>60</td>
</tr>
<tr>
<td>118</td>
<td>231</td>
<td>121</td>
</tr>
<tr>
<td>119</td>
<td>206</td>
<td>242</td>
</tr>
<tr>
<td>120</td>
<td>156</td>
<td>228</td>
</tr>
<tr>
<td>121</td>
<td>57</td>
<td>200</td>
</tr>
</tbody>
</table>

Continued on next page
<table>
<thead>
<tr>
<th>Clock cycle</th>
<th>Sequence 0</th>
<th>Sequence 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>122</td>
<td>115</td>
<td>144</td>
</tr>
<tr>
<td>123</td>
<td>230</td>
<td>32</td>
</tr>
<tr>
<td>124</td>
<td>204</td>
<td>64</td>
</tr>
<tr>
<td>125</td>
<td>152</td>
<td>128</td>
</tr>
<tr>
<td>126</td>
<td>49</td>
<td>0</td>
</tr>
<tr>
<td>127</td>
<td>98</td>
<td>1</td>
</tr>
<tr>
<td>128</td>
<td>197</td>
<td>2</td>
</tr>
<tr>
<td>129</td>
<td>139</td>
<td>4</td>
</tr>
<tr>
<td>130</td>
<td>22</td>
<td>9</td>
</tr>
<tr>
<td>131</td>
<td>45</td>
<td>19</td>
</tr>
<tr>
<td>132</td>
<td>90</td>
<td>39</td>
</tr>
<tr>
<td>133</td>
<td>180</td>
<td>79</td>
</tr>
<tr>
<td>134</td>
<td>105</td>
<td>158</td>
</tr>
<tr>
<td>135</td>
<td>210</td>
<td>60</td>
</tr>
<tr>
<td>136</td>
<td>164</td>
<td>121</td>
</tr>
<tr>
<td>137</td>
<td>72</td>
<td>242</td>
</tr>
<tr>
<td>138</td>
<td>145</td>
<td>228</td>
</tr>
<tr>
<td>139</td>
<td>34</td>
<td>200</td>
</tr>
<tr>
<td>140</td>
<td>69</td>
<td>144</td>
</tr>
<tr>
<td>141</td>
<td>138</td>
<td>32</td>
</tr>
<tr>
<td>142</td>
<td>20</td>
<td>64</td>
</tr>
<tr>
<td>143</td>
<td>41</td>
<td>128</td>
</tr>
<tr>
<td>144</td>
<td>82</td>
<td>0</td>
</tr>
<tr>
<td>145</td>
<td>165</td>
<td>1</td>
</tr>
<tr>
<td>146</td>
<td>74</td>
<td>2</td>
</tr>
<tr>
<td>147</td>
<td>149</td>
<td>4</td>
</tr>
<tr>
<td>148</td>
<td>42</td>
<td>9</td>
</tr>
<tr>
<td>149</td>
<td>84</td>
<td>19</td>
</tr>
<tr>
<td>150</td>
<td>169</td>
<td>39</td>
</tr>
<tr>
<td>151</td>
<td>83</td>
<td>79</td>
</tr>
<tr>
<td>152</td>
<td>167</td>
<td>158</td>
</tr>
<tr>
<td>153</td>
<td>78</td>
<td>60</td>
</tr>
<tr>
<td>154</td>
<td>157</td>
<td>121</td>
</tr>
<tr>
<td>155</td>
<td>59</td>
<td>242</td>
</tr>
<tr>
<td>156</td>
<td>119</td>
<td>228</td>
</tr>
<tr>
<td>157</td>
<td>238</td>
<td>200</td>
</tr>
<tr>
<td>158</td>
<td>221</td>
<td>144</td>
</tr>
<tr>
<td>159</td>
<td>187</td>
<td>32</td>
</tr>
<tr>
<td>160</td>
<td>118</td>
<td>64</td>
</tr>
<tr>
<td>161</td>
<td>236</td>
<td>128</td>
</tr>
<tr>
<td>162</td>
<td>217</td>
<td>0</td>
</tr>
</tbody>
</table>

Continued on next page
<table>
<thead>
<tr>
<th>Clock cycle</th>
<th>Sequence 0</th>
<th>Sequence 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>163</td>
<td>179</td>
<td>1</td>
</tr>
<tr>
<td>164</td>
<td>103</td>
<td>2</td>
</tr>
<tr>
<td>165</td>
<td>207</td>
<td>4</td>
</tr>
<tr>
<td>166</td>
<td>158</td>
<td>9</td>
</tr>
<tr>
<td>167</td>
<td>61</td>
<td>19</td>
</tr>
<tr>
<td>168</td>
<td>123</td>
<td>30</td>
</tr>
<tr>
<td>169</td>
<td>247</td>
<td>79</td>
</tr>
<tr>
<td>170</td>
<td>239</td>
<td>158</td>
</tr>
<tr>
<td>171</td>
<td>223</td>
<td>60</td>
</tr>
<tr>
<td>172</td>
<td>191</td>
<td>121</td>
</tr>
<tr>
<td>173</td>
<td>126</td>
<td>242</td>
</tr>
<tr>
<td>174</td>
<td>253</td>
<td>228</td>
</tr>
<tr>
<td>175</td>
<td>250</td>
<td>200</td>
</tr>
<tr>
<td>176</td>
<td>244</td>
<td>144</td>
</tr>
<tr>
<td>177</td>
<td>233</td>
<td>32</td>
</tr>
<tr>
<td>178</td>
<td>211</td>
<td>64</td>
</tr>
<tr>
<td>179</td>
<td>166</td>
<td>128</td>
</tr>
<tr>
<td>180</td>
<td>76</td>
<td>0</td>
</tr>
<tr>
<td>181</td>
<td>153</td>
<td>1</td>
</tr>
<tr>
<td>182</td>
<td>51</td>
<td>2</td>
</tr>
<tr>
<td>183</td>
<td>102</td>
<td>4</td>
</tr>
<tr>
<td>184</td>
<td>205</td>
<td>9</td>
</tr>
<tr>
<td>185</td>
<td>154</td>
<td>19</td>
</tr>
<tr>
<td>186</td>
<td>53</td>
<td>39</td>
</tr>
<tr>
<td>187</td>
<td>106</td>
<td>79</td>
</tr>
<tr>
<td>188</td>
<td>212</td>
<td>158</td>
</tr>
<tr>
<td>189</td>
<td>168</td>
<td>60</td>
</tr>
<tr>
<td>190</td>
<td>81</td>
<td>121</td>
</tr>
<tr>
<td>191</td>
<td>163</td>
<td>242</td>
</tr>
<tr>
<td>192</td>
<td>70</td>
<td>228</td>
</tr>
<tr>
<td>193</td>
<td>140</td>
<td>200</td>
</tr>
<tr>
<td>194</td>
<td>24</td>
<td>144</td>
</tr>
<tr>
<td>195</td>
<td>48</td>
<td>32</td>
</tr>
<tr>
<td>196</td>
<td>96</td>
<td>64</td>
</tr>
<tr>
<td>197</td>
<td>193</td>
<td>128</td>
</tr>
<tr>
<td>198</td>
<td>131</td>
<td>0</td>
</tr>
<tr>
<td>199</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>200</td>
<td>14</td>
<td>2</td>
</tr>
<tr>
<td>201</td>
<td>29</td>
<td>4</td>
</tr>
<tr>
<td>202</td>
<td>58</td>
<td>9</td>
</tr>
<tr>
<td>203</td>
<td>117</td>
<td>19</td>
</tr>
</tbody>
</table>

Continued on next page
<table>
<thead>
<tr>
<th>Clock cycle</th>
<th>Sequence 0</th>
<th>Sequence 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>204</td>
<td>234</td>
<td>39</td>
</tr>
<tr>
<td>205</td>
<td>213</td>
<td>79</td>
</tr>
<tr>
<td>206</td>
<td>170</td>
<td>158</td>
</tr>
<tr>
<td>207</td>
<td>85</td>
<td>60</td>
</tr>
<tr>
<td>208</td>
<td>171</td>
<td>121</td>
</tr>
<tr>
<td>209</td>
<td>87</td>
<td>242</td>
</tr>
<tr>
<td>210</td>
<td>175</td>
<td>228</td>
</tr>
<tr>
<td>211</td>
<td>95</td>
<td>200</td>
</tr>
<tr>
<td>212</td>
<td>190</td>
<td>144</td>
</tr>
<tr>
<td>213</td>
<td>124</td>
<td>32</td>
</tr>
<tr>
<td>214</td>
<td>249</td>
<td>64</td>
</tr>
<tr>
<td>215</td>
<td>242</td>
<td>128</td>
</tr>
<tr>
<td>216</td>
<td>229</td>
<td>0</td>
</tr>
<tr>
<td>217</td>
<td>202</td>
<td>1</td>
</tr>
<tr>
<td>218</td>
<td>148</td>
<td>2</td>
</tr>
<tr>
<td>219</td>
<td>40</td>
<td>4</td>
</tr>
<tr>
<td>220</td>
<td>80</td>
<td>9</td>
</tr>
<tr>
<td>221</td>
<td>161</td>
<td>19</td>
</tr>
<tr>
<td>222</td>
<td>66</td>
<td>39</td>
</tr>
<tr>
<td>223</td>
<td>132</td>
<td>79</td>
</tr>
<tr>
<td>224</td>
<td>9</td>
<td>158</td>
</tr>
<tr>
<td>225</td>
<td>19</td>
<td>60</td>
</tr>
<tr>
<td>226</td>
<td>39</td>
<td>121</td>
</tr>
<tr>
<td>227</td>
<td>79</td>
<td>242</td>
</tr>
<tr>
<td>228</td>
<td>159</td>
<td>228</td>
</tr>
<tr>
<td>229</td>
<td>63</td>
<td>200</td>
</tr>
<tr>
<td>230</td>
<td>127</td>
<td>144</td>
</tr>
<tr>
<td>231</td>
<td>255</td>
<td>32</td>
</tr>
<tr>
<td>232</td>
<td>254</td>
<td>64</td>
</tr>
<tr>
<td>233</td>
<td>252</td>
<td>128</td>
</tr>
<tr>
<td>234</td>
<td>248</td>
<td>0</td>
</tr>
<tr>
<td>235</td>
<td>240</td>
<td>1</td>
</tr>
<tr>
<td>236</td>
<td>225</td>
<td>2</td>
</tr>
<tr>
<td>237</td>
<td>194</td>
<td>4</td>
</tr>
<tr>
<td>238</td>
<td>133</td>
<td>9</td>
</tr>
<tr>
<td>239</td>
<td>11</td>
<td>19</td>
</tr>
<tr>
<td>240</td>
<td>23</td>
<td>39</td>
</tr>
<tr>
<td>241</td>
<td>47</td>
<td>79</td>
</tr>
<tr>
<td>242</td>
<td>94</td>
<td>158</td>
</tr>
<tr>
<td>243</td>
<td>188</td>
<td>60</td>
</tr>
<tr>
<td>244</td>
<td>120</td>
<td>121</td>
</tr>
</tbody>
</table>

Continued on next page
<table>
<thead>
<tr>
<th>Clock cycle</th>
<th>Sequence 0</th>
<th>Sequence 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>245</td>
<td>241</td>
<td>242</td>
</tr>
<tr>
<td>246</td>
<td>227</td>
<td>228</td>
</tr>
<tr>
<td>247</td>
<td>198</td>
<td>200</td>
</tr>
<tr>
<td>248</td>
<td>141</td>
<td>144</td>
</tr>
<tr>
<td>249</td>
<td>26</td>
<td>32</td>
</tr>
<tr>
<td>250</td>
<td>52</td>
<td>64</td>
</tr>
<tr>
<td>251</td>
<td>104</td>
<td>128</td>
</tr>
<tr>
<td>252</td>
<td>208</td>
<td>0</td>
</tr>
<tr>
<td>253</td>
<td>160</td>
<td>1</td>
</tr>
<tr>
<td>254</td>
<td>64</td>
<td>2</td>
</tr>
<tr>
<td>255</td>
<td>128</td>
<td>4</td>
</tr>
<tr>
<td>256</td>
<td>0</td>
<td>9</td>
</tr>
</tbody>
</table>
C. Changes in comparison to first 128-channel version

- Analogue front-end:
  - modified for new input capacitance range \(1.6 – 12\) pF;
  - new class-A amplifiers in the shaper and S2Diff designed, instead of previous class-AB;
  - almost whole analogue part put in deep nwell.

- Common mode signals and biasing DACs:
  - bandgap reference voltage block and temperature sensor removed;
  - global common mode generators removed;
  - new common mode generators \((v_{cm\_a}, v_{cm\_b}, v_{cm\_c}, v_{cm\_d})\) added. All common mode voltages are generated locally in each channel;
  - new DAC giving reference current for biasing of common mode generators \((v_{cm\_a}, v_{cm\_b}, v_{cm\_c})\) added.

- Digital
  - Data format changed to 12-bit base one;
  - Synch data packet is 24-bit long and is generated as an ordinary packet;
  - Idle packets generated in groups (number of packet in a group is configurable);
  - NZS additional information changed: one parity bit for all fields, NonActive is now 8-bit field, memory space field added;
  - Saturation arithmetic in pedestal and MCM subtractions added;
  - Memory overflow issue removed;
  - Changed synchronisation between ADC and DSP (no async FIFO anymore);
  - Correct DAC default values in configuration registers for ALL DACs;
  - In digital part all D flip-flops, clock and reset are triplicated;
  - Synchronous connect of all tree PLLs (also triplicated!);
  - Addresses of registers dll\_vcd\_cfg and dll\_cp\_cfg exchanged.

- Digital - NOT DONE
  - Reset via I\(^2\)C , besides of hardware reset line; several types of the reset (optional);
  - Start bit selection in serializer have to be corrected.

- Power distribution network:
  - new power distribution network was done in order to reduce: number of supplies, serial resistance, and parasitic coupling between analogue and digital supply lines;
– number of power domains reduced to analogue, mixed-mode and digital;
– in analogue part there are analogue and mixed-mode domains plus additional vref voltage for ADC reference.

D. Changelog

- Version 1.2
  - Appendix “Changes in comparison to second 8-channel version” deleted
  - Register names calib_maskxx_crg exchanged to calib_enablexx_crg
  - Test channel turn off bit description improved
  - Reset synchronizer schematic updated
  - Idle grouping restrictions updated
  - Full sequences of pseudo random counters in serializer added
  - Table with adc_sync_sel bit setting for different DLL delays updated