Timepix2 Manual (v2.1)

J. Alozy, R. Ballabriga, I. Kremastiotis, X. Llopart, T. Poikela, V. Sriskaran, W. Wong

CERN, Geneva

Last Modified: April 12, 2019

Document Versions

- Version 1.0: Original version, based on “Timepix2 Chip Description v6.0”
    - Note: This is a preliminary draft that is pre-released in order to provide information to readout system designers. Please refer to later versions for a complete manual.
- Version 2.0: This is the first official release of the Timepix2 Manual. Please note that at the time of writing this version, not all functions reported in this document had been fully tested yet.
- Version 2.1: Added warning to use spacing when sweeping the threshold through noise (e.g. during threshold equalisation) or when using test pulses. Also added Jerome to the author list.

Table of Contents

Project Description

Timepix2, developed by the Medipix2 Collaboration, is an upgraded version of Timepix (Llopart, Timepix Manual v1.0, 2006). The chip is designed for 130 nm CMOS technology of the TSMC foundry in Taiwan (whereas Timepix was in a 250 nm IBM process).

Features of Timepix2 include:

- 28 bits/pixel
- Simultaneous ToT and ToA
- Separate ToT and ToA clock frequencies
- Readout dead-time-free modes
- Fast clear of pixel counters
- Separate, programmable digital and analogue test pulses
- Digital and analogue pixel masking (turn off power consumption in unused pixels)
- Adaptive frontend gain
- Reduced threshold dispersion
- Digital diagnostics modes
- Test points in analogue frontend
- Matrix occupancy monitor
- Digital-only pixels with wirebond input
- Serial port (full frame)
- 32b Parallel port (full frame)
- Serial port (zero column suppression)
- Serial port daisy-chaining
- IO compatible with TSVs

Figure 1 shows the Timepix2 floorplan and Table 1 lists the main differences between the chip footprints of the Timepix and Timepix2 ASICs.

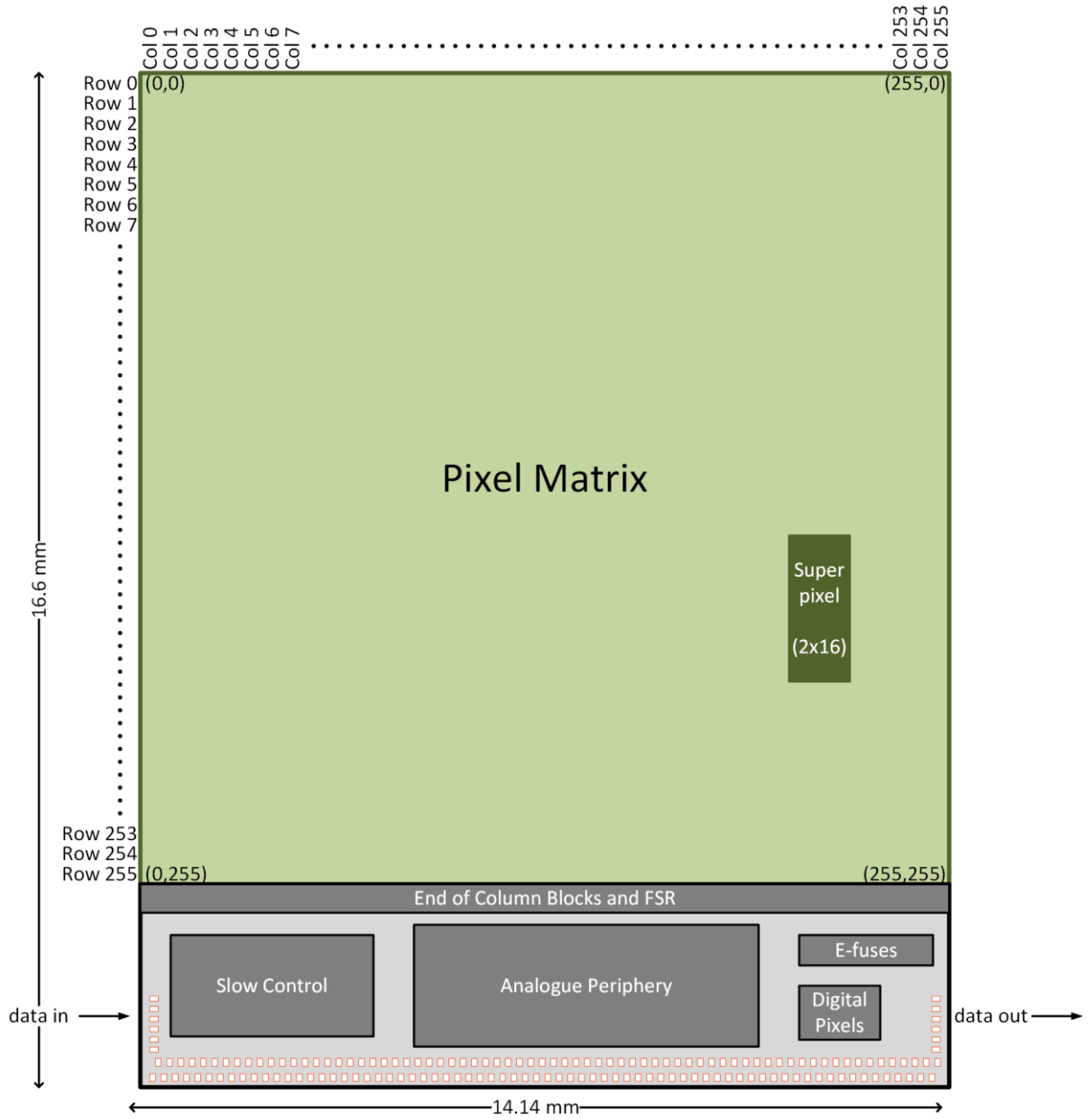

Figure 1. Timepix2 Floorplan

Table 1. Main differences between chip footprints of Timepix and Timepix2

	Timepix	Timepix2
Size [µm]	14111 x 16120	14140 x 16366
Pixel size	55 x 55 µm^2	55 x 55 µm^2
Pixel pad size	20 x 20 (octagon)	12 x 12 (octagon)
IO Pad pitch [µm]	120	200 (100 staggered)
Number IO Pads	127	152
TSV ready	NO	YES
Differential pads	LVDS	SLVS

The Timepix2 ASIC consists of a matrix of 256 x 256 square pixels of 55 µm side length, which are grouped in “super pixels” consisting of 2x16 pixels to consolidate resources and reduce power

consumption in the matrix. The chip periphery consists of the main chip programming slow control block, the analogue periphery, electronic fuses (for a unique chip serial number), and digital-only pixels that process inputs from external instruments rather than from the sensor.

Multiple Timepix2 chips can be tiled along the three sides that do not border the periphery. Table 2 lists the distances between the matrix and the edge of the chip.

Table 2. Distances from Pixel to Chip Edge

Side of chip	Centre of pixel to chip edge	Distance from pixel edge to chip edge
Left	57.35 µm	29.85 µm
Top	51.28 µm	23.78 µm
Right	57.65 µm	30.15 µm

Pixel Operations

The Timepix2 pixel matrix contains 256 columns of 256 rows of pixels, with 55 µm pitch. The pixels are organised in groups of superpixels of 2x16 pixels to consolidate common resources and reduce power consumption. Individual pixels or entire double columns can be powered down when not in use. Table 3 lists the different programmable features of the pixel matrix and indicates how to enable them. Each pixel contains memory to store unique configuration bits (CONF vector) and unique trim DAC codes (TRIM vector). Features that are enabled through the Operation Mode Register (OMR) are global (i.e. common to all pixels).

Table 3. Pixel feature programming

Pixel feature (analogue frontend)	How to program
Frontend charge collection polarity	OMR[10]
Enable frontend masking (turn off discriminator, significantly reduce power consumption of preamplifier)	OMR[9] + Set CONF command (enable MaskBit)
Enable frontend adaptive gain mode	OMR[8]
Frontend trim DAC code	Set TRIM command
Analogue test pulse charge injection (to test the frontend)	Set CONF command (enable TestBit) + Generate AnalogueTP command

Pixel feature (digital)	How to program
Digital operation mode	OMR[15:12]
Disable counting (readout still enabled)	Set CONF command (enable MaskBit)
Digital test pulse injection	Set CONF command (enable TestBit) + Generate DigitalTP command
Enable superpixel clock gating	OMR[11]
Enable double column clock gating	OMR[1]
Stagger the shutter from left to right of matrix	OMR[3]
Permit ToA counter to overflow	OMR[0]

Analog Front-end

The analog front-end consists of a Krummenacher-type (Krummenacher, 1991) charge sensitive preamplifier (CSA) is followed by a single threshold voltage discriminator. The CSA compensates for leakage current from the sensor and can be programmed to process either charge polarity. Each pixel also has a local threshold 5-bit trimming digital to analog converter (DAC) that tunes the threshold of the discriminator. Table 4 summarises the analog front-end parameters.

Table 4. Pixel analog front-end parameters

Parameter	Value
Technology	TSMC 130nm CMOS
Number of pixels	256 x 256 @ 55 um pitch
Analog front-end size	~55 x 14 um
Analog supply voltage	1.2 V
Detector capacitance	50 fF
Test capacitance	Nominal ~4fF (min: 3.5fF-max: 4.8fF), 100mV Δ_{TEST} ->(2.2ke$^-$ - 3ke$^-$)injected
Front-end gain	Linear, possibility to configure for "logarithmic" gain mode only in positive polarity
Detector polarity	Both Electron collection: • Leakage current compensation optimal (I_{DET}>I_{KRUM}) • Non-monotonicity of the ToT vs Qin Hole collection • Leakage current limited (I_{DET}<I_{KRUM}/2) • Logarithmic gain mode available
Leakage current	Electron collection: Up to 12nA/pixel (non-uniform) Hole collection: 2nA/pixel
Minimum threshold	~600e-
Peaking time	~100 ns (Adaptive gain=0) ~200ns (Adaptive gain=1)
Operating temperature	-20°C<T<70°C
Power consumption	5uA/pixel @1.2V i.e. ~165mW/cm^2 (low power mode)
Analog Power masking	Available per pixel

Adaptive Gain

Note: the adaptive gain mode is only available for positive charge polarities (i.e. hole collection).

The gain of the CSA is inversely proportional to the feedback capacitance. Figure 2 shows the two parallel paths for the feedback capacitance: one path consisting of a fixed metal-to-metal capacitance and a second path consisting of a metal-oxide-semiconductor (MOS) capacitance. When the OMR bit "En_AdaptiveGain" is disabled, the CSA feedback capacitance is only the fixed metal-to-metal capacitance and the CSA gain is constant for all input charge values. When the "En_AdaptiveGain" bit is enabled, the CSA gain depends on the input charge quantity, as the capacitance of the MOS capacitor depends on its gate voltage, which is connected to the preamplifier input. In this mode, the CSA capacitance is the combined capacitance of both the fixed metal-to-metal capacitor and the MOS capacitor, and the gain is high with low input charges and low with high input charges. When the adaptive gain mode is enabled, the time over threshold (ToT) of the discriminator output pulse increases monotonically with input charge up to 3.42 MeV (assuming a silicon sensor). The polarity of the signals at the preamplifier and discriminator output is shown in Figure 3 and Figure 4.

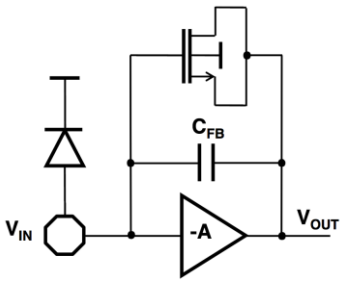

Figure 2. CSA with adaptive gain.

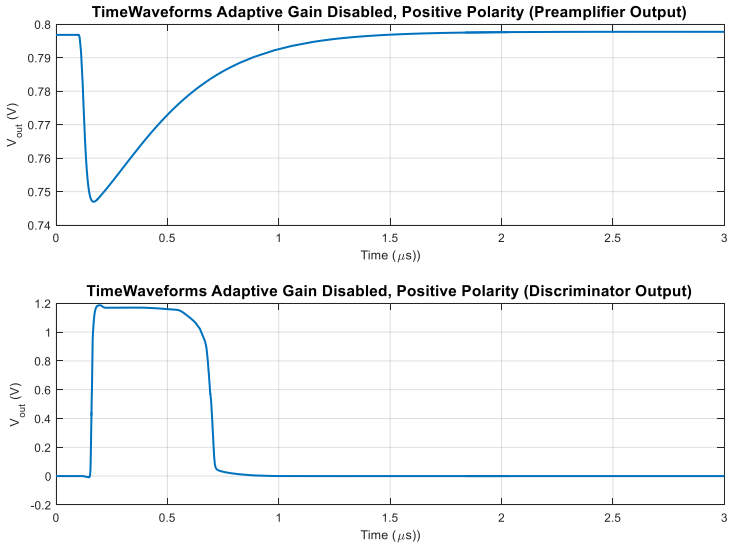

Figure 3. Pulses at the preamplifier and discriminator outputs for positive polarity.

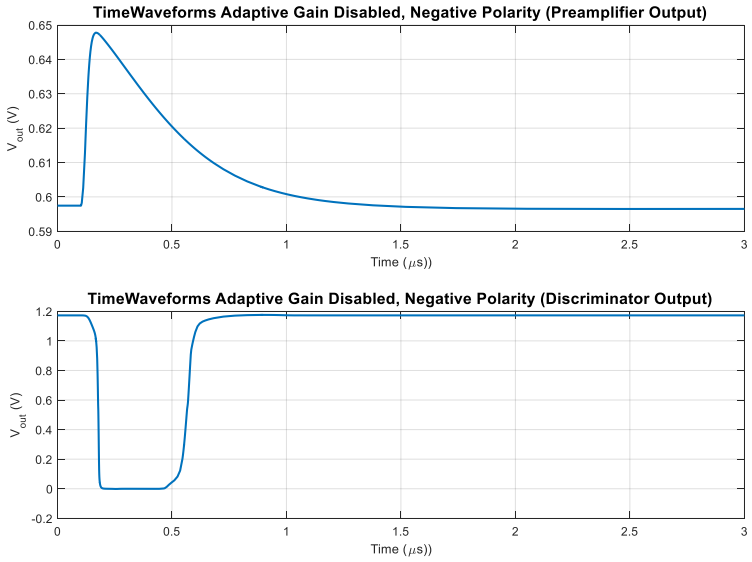

Figure 4. Pulses at the preamplifier and discriminator outputs for negative polarity.

Analog Front-end Masking

The CSA and discriminator are powered by current sources. If analog masking is globally enabled in the matrix through the OMR bit "En_AnalogMasking" *and* the MaskBit is enabled for a given pixel, the current source to the discriminator for that pixel is turned off and the current supplied to the CSA is reduced from several microamps to a few nanoamps. The front-end power consumption in that pixel is therefore just the few nanoamps consumed by the CSA. The CSA is kept slightly on to permit local leakage compensation of the sensor. The "on" and "off" currents are programmed by the global DACs in the periphery. When the Mask bit of a pixel is enabled, all the digital to analog converter control signals are programmed to logic 1 to disable the power consumption of the threshold adjustment DAC.

Table 5. Digital lines to control the analog double pixel.

Note: This table is included as a reference for chip designers; it is not intended for general users

Digital lines	
B_R<1:4>, Bn_R<0:4>	Threshold adjustment for the 5-bit DAC for the right pixel (All signals are set to 1 when the pixel is Masked (MaskBitAnalog_R=1)
B_L<1:4>, Bn_L<1:4>	Threshold adjustment for the 5-bit DAC for the right pixel (All signals are set to 1 when the pixel is Masked (MaskBitAnalog_L=1)
TPSwitch_R, TPSwitchBAR_R	Signals to control the test pulse circuitry for the right pixel
TPSwitch_L, TPSwitchBAR_L	Signals to control the test pulse circuitry for the left pixel
EnAdaptiveGain_dig	Adaptive Gain Enable (positive logic)
MaskBitAnalog_BAR_R	Bit to indicate if the pixel circuit is in a low power consumption mode (MaskBitAnalog_BAR_R=0) or in nominal operation (MaskBitAnalog_BAR_R=1)
MaskBitAnalog_BAR_L	Bit to indicate if the pixel circuit is in a low power consumption mode (MaskBitAnalog_BAR_R=0) or in nominal operation (MaskBitAnalog_BAR_R=1)
DiscOut_R	Discriminator Output Pulse (Pixel on the Right)
DiscOut_L	Discriminator Output Pulse (Pixel on the Left)

Table 6. Expected front-end parameters.

Adaptive gain	Gain (mV/ke$^-$)	Noise (mV r.m.s.)	Noise (e$^-$ r.m.s.)	Unequalized threshold (mV r.m.s.)	Unequalized threshold (e$^-$ r.m.s.)	Equalized threshold (mV r.m.s.)	Equalized threshold (e$^-$ r.m.s.)	Minimum Threshold (e$^-$)
0	24.7	1.51	61	11.54	460	0.7	28	400
1	19.2	0.98	51.2	11.54	607	0.7	37	380

Table 7. List of biasing lines for the pixel and their description.

Note: This table is included as a reference for chip designers; it is not intended for general users

	Analog biasing line	Description	Target transistor	Default value (MSB…LSB)	Protection diodes
1	Vanalog1 (V) (DAC_OUT_VTP_COARSE) (8bits)	Analog voltage 1 for the test pulse injection	**Voltage**		
2	Vanalog2 (V) (DAC_OUT_VTP_FINE) (10bits)	Analog voltage 2 for the test pulse injection	**Voltage**		
3	VBiasPreampPMOS (8bits)	Current DAC, preamplifier differential pair	**PMOS reg 3/2**	**1.5uA**/780mV	pdio
4	VBiasPreampPMOS_stb (4bits)	Current DAC, preamplifier differential pair (power off value)	**PMOS reg 1/2**	**100nA**	Pdio
5	VBiasPreampCasc (8bits)	Voltage DAC, cascode biasing line	**Voltage, PMOS reg 4/0.2**	**410mV**	Pdio
6	VGND (8bits)	Voltage of the input node through preamp virtual ground	**Voltage**, PMOS reg 14/0.15	**750mV**	Pdio
7	VBiasLevelShifter (8bits)	Current DAC, Level shifter current	**PMOS reg 4x 1/0.3**	**2uA**/650mV	Pdio
8	VBiasLevelShifter_stb (4 bits)	Current DAC, Level shifter current (power off value)	**PMOS reg 4x 0.5/0.3**	**50nA**	Pdio
9	VFBK (8bits)	Voltage	**Voltage**, NMOS hvt 1.16/0.2 ELT	**800mV / 550mV (750mV all corners)**	Pdio
10	Vbiaskrum (8bits)	Preamplifier return to zero current	**NMOS hvt ELT 3.44/0.8**	**2nA** / 200mV	Ndio
11	VTH (14bits)	Voltage DAC, Threshold voltage	**Voltage**, NMOS reg 4/0.18	**780mV / 570mV**	Ndio
12	VBiasDiscPMOS (8bits)	Discriminator bias current (PMOS)	**PMOS reg 4x 1.79/0.5**	**750nA**	Pdio
13	VBiasDiscNMOS (8bits)	Discriminator bias current (NMOS)	**NMOS HVT 1/5**	**710nA 1.5u**	Ndio
14	VBiasCascDisc (8bits)	Discriminator cascode (PMOS)	**Voltage**, PMOS 2/0.14	**600mV**	Pdio
15	VBiasDAC (8bits)	Threshold DAC ILSB current	**NMOS 6.64/1 ELT**	**70nA**/250mV	Ndio
16	VBiasDACCasc (8bits)	(DAC) (reference voltage generated on the previous DAC)	**NMOS hvt 1/0.2**	**70nA**/560mV	Ndio

Digital Pixel Operation Modes

The Timepix2 pixel can be programmed to operate in one of several modes. Each pixel contains 28 counter bits and the use of those counter bits depends on the mode. All pixels in the matrix operate in the mode[1] that is programmed into the Operation Mode Register (OMR) of the chip periphery (see Section The Operation Mode Register (OMR)).
Table 8 lists the pixel operation modes. The counter chain configurations are also listed in the final two columns, where the counter chains are listed in order of most significant bit chain followed by least significant bit chain.

Table 8. Pixel operation modes

Mode Name	Description	1st Counter	2nd Counter
ToT10/ToA18	Simultaneous ToT and 1st hit ToA (with sequential read/write) Mode options (programmable): 1) 1st hit or integral ToT 2) Overflow (wraparound) of ToA counter	10-bit ToT {CounterA}	18-bit ToA {CounterD, CounterC, CounterB}
ToT14/ToA14		14-bit ToT {CounterD, CounterA}	14-bit ToA {CounterC, CounterB}
ContToT10/ Event4	Continuous read/write ToT Mode options: 1) 1st hit or integral ToT (programmable) 2) Supplementary 4-bit eventing counting (readout optional)	10-bit ToT {CounterA} 4-bit #Events {CounterD}	10-bit ToT {CounterB} 4-bit #Events {CounterC}
ContToT14	Continuous read/write ToT Mode option: 1st hit or integral ToT (programmable)	14-bit ToT {CounterD, CounterA}	14-bit ToT {CounterC, CounterB}
ContToA10	Continuous read/write 1st hit ToA	10-bit ToA {CounterA}	10-bit ToA {CounterB}
ContToA14		14-bit ToA {CounterD, CounterA}	14-bit ToA {CounterC, CounterB}
ContEvent10	Continuous read/write event counting	10-bit #Events {CounterA}	10-bit #Events {CounterB}
ContEvent14		14-bit #Events {CounterD, CounterA}	14-bit #Events {CounterC, CounterB}

The Shutter is an active low signal: it is “open” when Shutter = 0 and “closed” when Shutter = 1. Any hits that arrive while the Shutter is closed are ignored. 1st hit time over threshold (ToT) is evaluated on the first discriminator output pulse that arrives after the Shutter opens. Integral ToT cumulatively counts the ToT of all hits that arrive while the Shutter is open. If a pulse is still high when the Shutter

[1] Recall that the original Timepix pixels contained three modes: Medipix mode, Timepix mode, and ToT mode. The mode programming was based on configuration bits, so different pixels could be programmed with different modes. In Timepix2, the modes are much more complex and all pixels in the matrix will operate under the same mode.

closes, the ToT counter continues to count until the end of the pulse or until the maximum counter value is reached (whichever happens sooner). The time of arrival (ToA) is time from the rising edge of the first discriminator output pulse to the closing of the Shutter. Figure 5 - Figure 7 explain counting in the various modes with respect to the Shutter state. In the examples of Figure 6 and Figure 7, the ToT counting is still active when the Shutter closes. An IO pad called READREADYn is an output from the chip that indicates whether any ToT counters are still active following a Shutter transition. This signal asserts when the ToT counters are ready for readout.

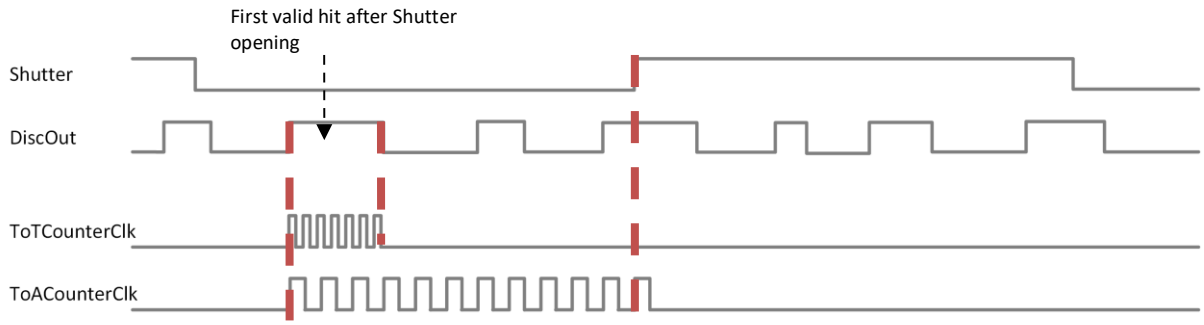

Figure 5. Evaluation of 1st hit ToT and ToA in the simultaneous ToT/ToA mode with sequential readout

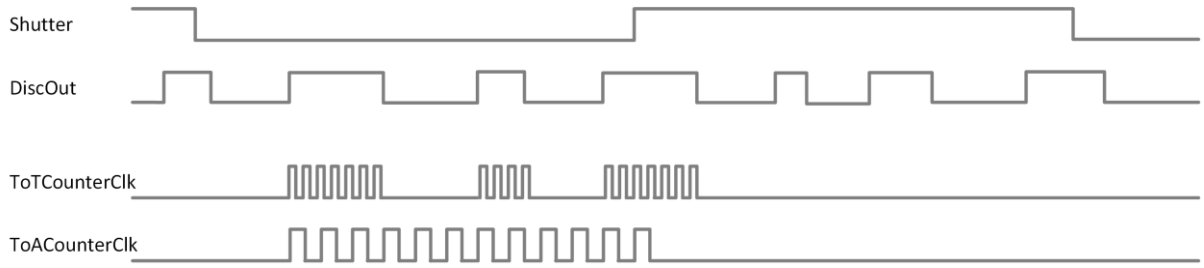

Figure 6. Evaluation of integral ToT in the simultaneous ToT/ToA mode with sequential readout

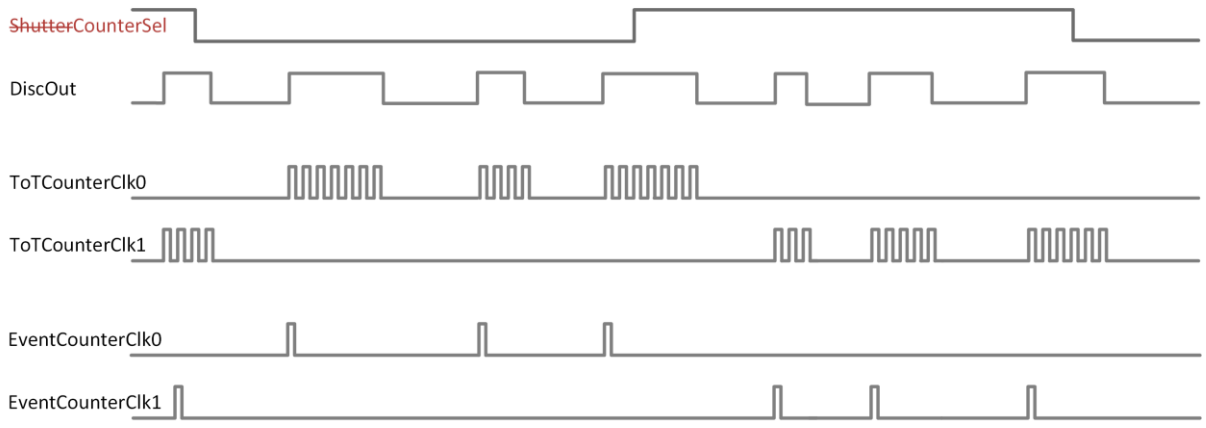

Figure 7. Evalulation of integral ToT and event counting in the continuous read/write ToT mode

Digital Counter Configurations

Each pixel has four digital counters configured as linear feedback shift registers (LFSR). Table 9 lists the four counter chains and the locations of the exclusive-OR taps to create the pseudo random code. The schematics in Figure 8 conceptually explain how the pseudo random counters are implemented and how they can be combined to create longer words. It should be noted that these pseudo random counters provide unique codes for the full counter depth. The use of the counter chains in the various pixel operation modes was previously listed in Table 8.

Table 9. Counter Word Sizes

Counter Name	Counter Depth	XOR tap positions
CounterA	10 bits	(6, 9)
CounterB	10 bits	(6, 9)
CounterC	4 bits	(2, 3)
CounterD	4 bits	(2, 3)

Each n-bit counter outputs 2^n unique pseudo-random codes. Multiple LFSR chains can be combined to create longer effective depths. For example, the combination of a 4-bit and 10-bit counter will provide 2^{14} unique codes[2]. All sets of pseudo-random codes regardless of total depth can be decoded offline using a 4-bit lookup table (LUT) and a 10-bit LUT.

A multiplexor at the data input selects between counting mode (i.e. LFSR configuration) or readout mode (i.e. serial shift register configuration).

Figure 8 shows the example schematic of an effective 14-bit counter created by joining the 10-bit CounterA with the 4-bit CounterD. The clock of CounterD toggles each time CounterD overflows. The clock of CounterA depends on the Operation Modes described on subsequent pages. The effective 14-bit word can be calculated offline:

$$14\text{-bit value} = (\text{DecodedValue}_{\text{CounterD}} \times 2^{10}) + \text{DecodedValue}_{\text{CounterA}}$$

Similarly, if the pixel operation mode includes 18-bit counting, the 18-bit word combination of CounterB, CounterC and CounterD can be calculated:

$$18\text{-bit value} = (\text{DecodedValue}_{\text{CounterD}} \times 2^{14}) + (\text{DecodedValue}_{\text{CounterC}} \times 2^{10}) + \text{DecodedValue}_{\text{CounterB}}$$

Table 10 lists the decoded values of the pseudo-random output of the 4-bit LFSRs (i.e. CounterC and CounterD). The 10-bit LUT can be found in Table 32 of the Appendix. The zero values of the counters are 10'b0000000000 and 4'b0000.

The counter configuration and depth is determined by pixel operation mode programming in the Operation Mode Register (OMR) of the chip periphery.

---

[2] Recall that in Timepix, the 14-bit counter could count to a maximum of 11810. In Timepix2, the 14-bit counter outputs 16484 unique codes.

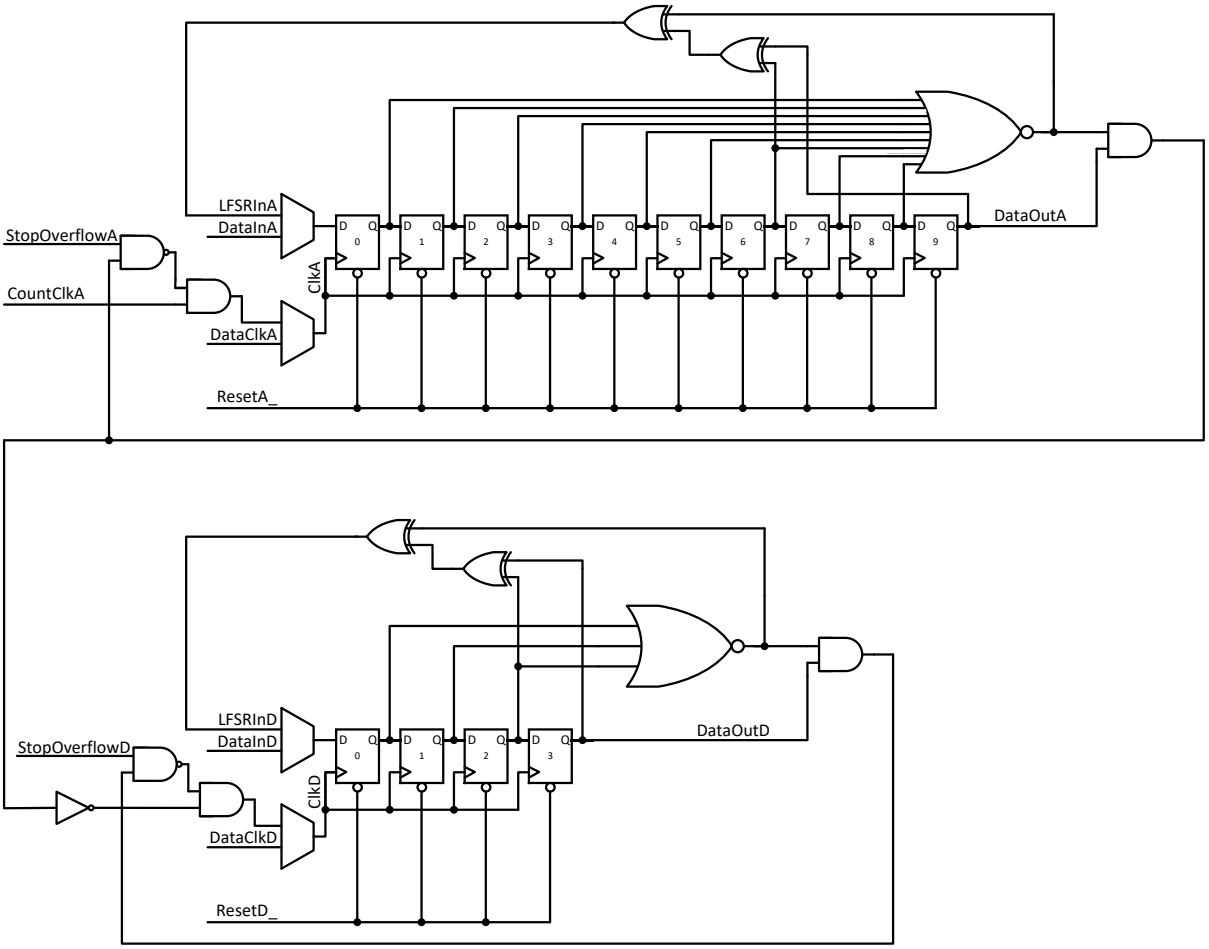

Figure 8. Combining CounterA and CounterD to create an effective 14-bit LFSR counter (note the diagram here is conceptual and does not accurately represent the final circuit)

Table 10. LUT for the 4-bit LFSRs (CounterC and CounterD)

LFSR Pseudo-Random Output	Decoded Decimal Value
4'b0000	0
4'b0001	1
4'b0010	2
4'b0100	3
4'b1001	4
4'b0011	5
4'b0110	6
4'b1101	7
4'b1010	8
4'b0101	9
4'b1011	10
4'b0111	11
4'b1111	12
4'b1110	13
4'b1100	14
4'b1000	15

Clocks

The pixel has three input clocks: time over threshold (ToT) clock, time of arrival (ToA) clock, and the data (writing and readout) clock. The ToTClk is directly from the MCLOCK_IN wirebond pad. The ToAClk is the output of a clock divider in the periphery (which uses MCLOCK_IN as the input clock). The DataClk is directly from the DCLOCK_IN wirebond pad. All three clock domains can run up to 100 MHz and expect a 50% duty cycle.

Counter Overflow Prevention

The ToT counter stops counting when the maximum value of the ToT counter (for the programmed counter depth) is reached. Similar overflow prevention logic exists for ToA counting, but this can be disabled by the OMR bit “En_ToAOverflow”. If this OMR bit is enabled, the ToA counter wraps around back to 0 after the maximum value and continues counting until the Shutter closes.

Reading Out Counter Values

The four counter chains are read out one counter at a time, regardless of Pixel Operation Mode of effective counter depth.

Programming Individual Pixel Settings

Configuration Bits

Global chip programming (i.e. programming that applies to all the pixels) is done through the OMR. Configuration bits, on the other hand, are stored in each pixel and provide individual pixel settings.

Whereas most programmable global control signals in Timepix2 are active low signals, the pixel configuration bits are exceptionally *active high* signals. Table 11 describes the CONF Vector.

Upon chip initialisation via a GlobalReset, the MaskBit and TestBit of all pixels in the matrix are initialised to their "off" value, which means no pixels are masked and no test pulses are enabled.

The CONF Vector is loaded into the pixel using the SETCONF programming command. Although there are only two configuration bits, the loading of these bits goes through the pixel register chain of the 4-bit COUNTERC. Therefore, four configuration bits are loaded into each pixel, where the two most significant bits are padded with 0's.

The CONF Vector can also be read out of the chip for digital diagnostics. When a GETCONF programming command is executed, a 4-bit word is shifted out of each pixel. Since there are five trim bits, the MSB of the TRIM Vector is concatenated onto the CONF Vector during a GETCONF command. Table 12 lists the bit assignments of the values shifted out from the pixel memory during a GETCONF command.

Table 11. CONF Vector loaded in each pixel (in SETCONF command)

CONF Index	Bit Assignment	Description
CONF[3:2]	*Unassigned*	Unused bits (pad with 0 for SET CONF Command)
CONF[1]	MaskBit	Selectively disables a pixel **0 : Normal operation in the digital circuits. In the analogue circuits, the "on" biasing currents are selected for the frontend**. This is the default value following a GlobalReset. 1 : Disables digital counting in the pixel (but the pixel will still shift digital data through for readout) and powers down the analogue frontend by selecting the "off" biasing currents (if the OMR bit "En_AnalogMasking is also asserted)
CONF[0]	TestBit	Selects testpulse input. To be used with "Enable Analogue Test" in the OMR. **0 : Input to the frontend is charge from the sensor, and input to the digital counters is the Discriminator output.** (default after GlobalReset) 1 : If "Enable Analogue Test" in the OMR is asserted, an Analogue TestPulse is injected into the frontend. If "Enable Analogue Test" is off, the frontend is bypassed and a Digital TestPulse is input to the digital circuits

Trim Bits

The trim bits provide the digital code for the in-pixel threshold adjustment DAC. There are five trim bits. Although the TRIM Vector is a digital value that is programmed and read out digitally, its use is in the analogue domain. The discriminator threshold tuning DAC is controlled by nine bits: the *four* most significant bits of the TRIM Vector, TRIM[4:1], and the five complements of the TRIM Vector, TRIM_BAR[4:0]. When the OMR bit "En_AnalogMasking" is disabled, TRIM_BAR is the direct complement of TRIM. In the exceptional case where the OMR bit "En_AnalogMasking" is enabled to power down the frontend of all pixels with the configuration bit MaskBit = 1, TRIM[4:1] = 4'b1111 and TRIM_BAR[4:0] = 5'b11111 for those pixels, irrespective of what was loaded into the pixel memory during a SETTRIM command. This code powers down the pixel DAC completely.

Upon chip initialisation via a GlobalReset, the TRIM Vector is reset to all 0's, and the OMR bit "En_AnalogMasking" is disabled. During a threshold equalisation, the midrange value, i.e. TRIM[4:0] = 5'b10000, should be used as the starting point, as the midrange value assigns equal current to each branch of the discriminator.

The TRIM Vector is loaded into the pixel using the SETTRIM programming command. Although there are five trim bits, the loading of these bits goes through the pixel register chain of the 10-bit COUNTERA. Therefore, five trim bits are loaded into each pixel, where the five most significant bits are padded with 0's.

The TRIM Vector can also be read out of the chip for digital diagnostics. When a GETTRIM programming command is executed, a *4-bit* word is shifted out of each pixel. Since there are five trim bits, the MSB of the TRIM Vector (i.e. TRIM[4]) is concatenated onto the CONF Vector during a GETCONF command, and the four remaining bits (i.e. TRIM[3:0]) are shifted out during a GETTRIM command. Table 12 lists the bit assignments of the values shifted out from the pixel memory during the GETCONF and GETTRIM commands.

Table 12. Values shifted out during GETCONF and GETTRIM commands

Command	4-bit word read out from each pixel
GETCONF	{TRIM[4], 0, CONF[1], CONF[0]}
GETTRIM	If En_AnalogMasking is enabled and MaskBit = 1: {111, TRIM[0]} Otherwise: {TRIM[3:0]}

Threshold Equalisation

The periphery DACs VTHCOARSE and VTHFINE[3] (Table 19) set the global threshold, which is common to all pixels in the matrix. Each pixel has a 5-bit programmable trim code that can be tuned to equalise the global threshold, and the LSB current for the threshold tuning DAC in the pixel is set by the periphery DAC VBIAS_THS. The trim DAC supplies or sinks current to the two differential branches of the frontend discriminator. The midrange code 0x10 provides equal current to both branches and should be the used as the starting code to determine the untuned threshold dispersion. According to Monte Carlo simulations of the frontend, the expected dispersion after tuning should be 40 e^- rms and 30 e^- rms with and without adaptive gain, respectively.

Note: During threshold equalisation, please disable the OMR bit "En_AnalogMasking", and please use a spacing of at least 4.

[3] VTHCOARSE and VTHFINE were formerly called the THL DAC in the Medipix2MXR and Timepix ASICs.

Chip IO Interface

Wirebond Pads

The Timepix2 chip contains 152 wirebond pads along the bottom of the chip periphery. The pads (shown in Figure 9) are 200.8 µm by 104 µm, staggered across two rows and spaced 200 µm apart.

Figure 9. Timepix2 wirebond pad diagram

Zoomed in versions of Figure 9 are in the Appendix; the image is also available for download.

Pin Naming Conventions

For differential SLVS pairs of pins, the lower case "p" and "n" prefix on pin names denote the complementary positive and negative signals, respectively. To simplify the text, the prefixes will be dropped in the rest of this document. The main SLVS clock and data inputs and outputs are repeated along the left and right edge of the chip to facilitate daisy chained connections of multiple chips.

Active low signals are denoted by the "n" suffix on pin names.

Table 14 lists the wirebond pads in the Timepix2 chip.

Input Data Timing

The recommended minimum setup and hold times (see Figure 10) for DATA_IN are described in Table 13.

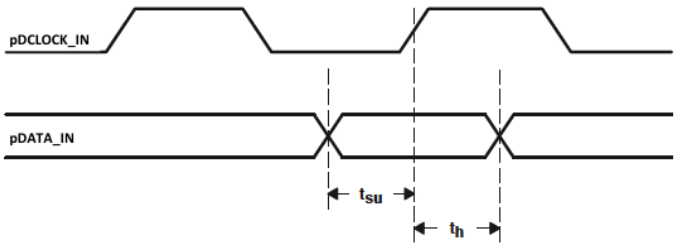

Figure 10. Input data and clock timing

Table 13. Setup and Hold Time of DATA_IN

	Value
Setup time, t_{su}	250 ps
Hold time, t_h	700 ps

Power Pins

VDDIO: 2.5V
VDD (Digital): 1.2V
VDDA (Analog): 1.2V
VDD33: 2.5V (needs to be applied during both eFuse burning and eFuse readout)

Through-Silicon-Vias (TSV)

The Timepix2 wirebond pads are compatible with through silicon via (TSV) technologies. TSVs can be drilled from the backside of the pads.

Table 14. Pin definitions

Note: Pads 1-70 are the top row of staggered pads.
Pads 71-140 are the bottom row of staggered pads.
141-146 are the pads located to the left side of the chip for daisy chaining
147-152 are the pads located to the right side of the chip for daisy chaining.
The origin of the pad layout was taken as point (0,0).
The coordinates were measured at the center of the pad/TSV.

Pin #	Pin Name	Input/ Output	Signal Type	Description	Wirebond Pad		TSV	
					X [µm]	Y [µm]	X [µm]	Y [µm]
1	nDATA_IN (bottom)	Input	SLVS pair	Serial input data. Expected value of positive terminal on global reset (if connected) = LOW. (1 pair on left edge, 1 pair on bottom-left edge)	220	456.86	220	499.755
2	pDATA_IN (bottom)				420	456.86	420	499.755
3	nDCLOCK_IN (bottom)	Input	SLVS pair	Data shift clock. Values on DATA_IN are sampled on the rising edge of DCLOCK_IN. Expected to toggle during global and data reset procedures. Maximum frequency: 100 MHz. (1 pair on left edge, 1 pair on bottom-left edge)	620	456.86	620	499.755
4	pDCLOCK_IN (bottom)				820	456.86	820	499.755
5	nENABLE_INn (bottom)	Input	SLVS pair	*Active low* chip enable. Expected value of positive terminal on global reset (if connected) = HIGH. (1 pair on left edge, 1 pair on bottom-left edge)	1020	456.86	1020	499.755
6	pENABLE_INn (bottom)				1220	456.86	1220	499.755
7	nMCLOCK	Input	SLVS pair	Measurement clock. This is the reference clock for ToT measurements and for the clock divider to generate the ToA clock. Maximum frequency: 100 MHz.	1420	456.86	1420	499.755
8	pMCLOCK	Input	SLVS pair		1620	456.86	1620	499.755
9	VDDA	-	Power		1820	456.86	1820	499.755
10	VDDA	-	Power		2020	456.86	2020	499.755
11	DIGPIXIN[7]	Input	Digital CMOS	Input stimuli (i.e. discriminator output) for eight digital-only pixels in the periphery. Expected value on global reset = LOW. Connection optional; if connected, the digital pixels need to be enabled in the OMR.	2220	456.86	2220	499.755
12	DIGPIXIN[6]	Input	Digital CMOS		2420	456.86	2420	499.755
13	DIGPIXIN[5]	Input	Digital CMOS		2620	456.86	2620	499.755
14	DIGPIXIN[4]	Input	Digital CMOS		2820	456.86	2820	499.755
15	DIGPIXIN[3]	Input	Digital CMOS		3020	456.86	3020	499.755
16	DIGPIXIN[2]	Input	Digital CMOS		3220	456.86	3220	499.755
17	DIGPIXIN[1]	Input	Digital CMOS		3420	456.86	3420	499.755
18	DIGPIXIN[0]	Input	Digital CMOS		3620	456.86	3620	499.755
19	GLOBALRESETn	Input	Digital CMOS	*Active low* chip reset. When asserted, it should be held low for a minimum of 824 DCLOCK_IN toggles.	3820	456.86	3820	499.755

20	SHUTTERn/ COUNTERSEL	Input	Digital CMOS	*Active low* electronic shutter (in simultaneous ToT/ToA modes) and counter select signal (in continuous read/write modes). In simultaneous ToT/ToA modes: SHUTTERn should be high during counter readout. In continuous read/write modes: counter mode (i.e. LFSR counter or serial shift register) depends on polarity of COUNTERSEL. Expected value on global reset = HIGH. Note: this signal is asynchronous from the wirebond pad, but is then internally synchronized to MCLOCK.	4020	456.86	4020	499.755
21	VDD	-	Power		4220	456.86	4220	499.755
22	READREADYn	Output	Digital CMOS	*Active low* boolean flag to indicate that the ToT counters have completed counting and can be changed to readout mode after a transition in the SHUTTERn/COUNTERSEL pin. Note: if a Get (ToT) COUNTER* command is executed before this flag is asserted, the ToT counters may output corrupted values. This signal can be ignored in operation modes that do not count ToT (i.e. continuous read/write ToA or event counting modes).	4420	456.86	4420	499.755
23	MATRIX_OCCn	Output	Digital CMOS/ Hi-Z	*Active low* boolean flag to indicate that the number of columns in the matrix that contain data has surpassed the Matrix Occupancy Threshold (which is a programmable value). Connection optional; if connected, please enable it in the OMR. Default value after global reset = Hi-Z.	4620	456.86	4620	499.755
24	PBUS_ACCESSn	Output	Digital CMOS/ Hi-Z	*Active low* boolean flag to indicate that the parallel port is actively driving data. This pin outputs Hi-Z when the chip is not executing a Get COUNTER*_PAR command. If no Get COUNTER*_PAR commands are foreseen to be used, this pin can be left unconnected. Default value after global reset = Hi-Z.	4820	456.86	4820	499.755
25	DIGITAL_OUT[0]	Output	Digital CMOS/ Hi-Z	Test point to probe digital internal chip signals. Outputs the signal selected by the command Set DIGPROBESEL_0. Connection optional; outputs Hi-Z unless a value is set to the slow control register DIGPROBESEL_0. Default value after global reset = Hi-Z.	5020	456.86	5020	499.755
26	DIGITAL_OUT[1]	Output	Digital CMOS/ Hi-Z	Test point to probe digital internal chip signals. Outputs the signal selected by the command Set DIGPROBESEL_1. Connection optional; outputs Hi-Z unless a value is set to the slow control register DIGPROBESEL_1. Default value after global reset = Hi-Z.	5220	456.86	5220	499.755
27	DOUT[0]	Output	Digital CMOS/ Hi-Z	Parallel output data. Connection optional; the parallel port will drive Hi-Z unless a Get COUNTER*_PAR is executed. Default value after global reset = Hi-Z.	5420	456.86	5420	499.755
28	DOUT[1]	Output	Digital CMOS/ Hi-Z		5620	456.86	5620	499.755
29	DOUT[2]	Output	Digital CMOS/ Hi-Z		5820	456.86	5820	499.755
30	DOUT[3]	Output	Digital CMOS/ Hi-Z		6020	456.86	6020	499.755
31	DOUT[4]	Output	Digital CMOS/ Hi-Z		6220	456.86	6220	499.755
32	DOUT[5]	Output	Digital CMOS/ Hi-Z		6420	456.86	6420	499.755
33	DOUT[6]	Output	Digital CMOS/ Hi-Z		6620	456.86	6620	499.755
34	DOUT[7]	Output	Digital CMOS/ Hi-Z		6820	456.86	6820	499.755
35	DOUT[8]	Output	Digital CMOS/ Hi-Z		7020	456.86	7020	499.755

36	DOUT[9]	Output	Digital CMOS/ Hi-Z		7220	456.86	7220	499.755
37	DOUT[10]	Output	Digital CMOS/ Hi-Z		7420	456.86	7420	499.755
38	DOUT[11]	Output	Digital CMOS/ Hi-Z		7620	456.86	7620	499.755
39	DOUT[12]	Output	Digital CMOS/ Hi-Z		7820	456.86	7820	499.755
40	DOUT[13]	Output	Digital CMOS/ Hi-Z		8020	456.86	8020	499.755
41	DOUT[14]	Output	Digital CMOS/ Hi-Z		8220	456.86	8220	499.755
42	DOUT[15]	Output	Digital CMOS/ Hi-Z		8420	456.86	8420	499.755
43	DOUT[16]	Output	Digital CMOS/ Hi-Z		8620	456.86	8620	499.755
44	DOUT[17]	Output	Digital CMOS/ Hi-Z		8820	456.86	8820	499.755
45	DOUT[18]	Output	Digital CMOS/ Hi-Z		9020	456.86	9020	499.755
46	DOUT[19]	Output	Digital CMOS/ Hi-Z		9220	456.86	9220	499.755
47	DOUT[20]	Output	Digital CMOS/ Hi-Z		9420	456.86	9420	499.755
48	DOUT[21]	Output	Digital CMOS/ Hi-Z		9620	456.86	9620	499.755
49	DOUT[22]	Output	Digital CMOS/ Hi-Z		9820	456.86	9820	499.755
50	DOUT[23]	Output	Digital CMOS/ Hi-Z		10020	456.86	10020	499.755
51	DOUT[24]	Output	Digital CMOS/ Hi-Z		10220	456.86	10220	499.755
52	DOUT[25]	Output	Digital CMOS/ Hi-Z		10420	456.86	10420	499.755
53	DOUT[26]	Output	Digital CMOS/ Hi-Z		10620	456.86	10620	499.755
54	DOUT[27]	Output	Digital CMOS/ Hi-Z		10820	456.86	10820	499.755
55	DOUT[28]	Output	Digital CMOS/ Hi-Z		11020	456.86	11020	499.755
56	DOUT[29]	Output	Digital CMOS/ Hi-Z		11220	456.86	11220	499.755
57	DOUT[30]	Output	Digital CMOS/ Hi-Z		11420	456.86	11420	499.755
58	DOUT[31]	Output	Digital CMOS/ Hi-Z		11620	456.86	11620	499.755

59	VDDA	-	Power		11820	456.86	11820	499.755
60	DAC_OUT	Output	Analog	This pin probes the output voltage of a DAC or the bandgap reference circuit, depending on the value stored in the slow control register DACOUTSEL.	12020	456.86	12020	499.755
61	PREAMP_OUT	Output	Analog	This pin probes the output voltage from the preamplifier of the pixel in the bottom row of the matrix, corresponding to the column selected in the slow control register PREAMP_OUT_SEL. Connection optional; if PREAMP_OUT_SEL is set to the default value of 0, no preamplifier output will be selected.	12220	456.86	12220	499.755
62	VDDA	-	Power		12420	456.86	12420	499.755
63	VDDA	-	Power		12620	456.86	12620	499.755
64	VDDA33	-	Power	2.5V needs to be applied to this pin during eFuse burning, and also during readout of the eFuse (Get ChipID).	12820	456.86	12820	499.755
65	pENABLE_OUTn (bottom)	Output	SLVS pair	*Active low* enable for the next chip in a daisy chain. Connection optional if no daisy chain. (1 pair on right edge, 1 pair on bottom-right edge)	13020	456.86	13020	499.755
66	nENABLE_OUTn (bottom)				13220	456.86	13220	499.755
67	pDCLOCK_OUT (bottom)	Output	SLVS pair	Data shift output clock. The data on the DATA_OUT pin pairs are synchronized to this clock. (1 pair on right edge, 1 pair on bottom-right edge)	13420	456.86	13420	499.755
68	nDCLOCK_OUT (bottom)				13620	456.86	13620	499.755
69	pDATA_OUT (bottom)	Output	SLVS pair	Serial output data. Data is shifted out on the rising edge of DCLOCK_OUT. (1 pair on right edge, 1 pair on bottom-right edge)	13820	456.86	13820	499.755
70	nDATA_OUT (bottom)				14020	456.86	14020	499.755
71	VDD	-	Power		120	117	120	74.105
72	GND	-	Power		320	117	320	74.105
73	VDD	-	Power		520	117	520	74.105
74	GND	-	Power		720	117	720	74.105
75	VDD	-	Power		920	117	920	74.105
76	GND	-	Power		1120	117	1120	74.105
77	VDD	-	Power		1320	117	1320	74.105
78	GND	-	Power		1520	117	1520	74.105
79	GNDA	-	Power		1720	117	1720	74.105
80	GNDA	-	Power		1920	117	1920	74.105
81	VDDA	-	Power		2120	117	2120	74.105
82	GNDA	-	Power		2320	117	2320	74.105
83	VDD	-	Power		2520	117	2520	74.105
84	GND	-	Power		2720	117	2720	74.105
85	VDDIO	-	Power		2920	117	2920	74.105
86	GNDA	-	Power		3120	117	3120	74.105
87	VDDA	-	Power		3320	117	3320	74.105
88	GNDA	-	Power		3520	117	3520	74.105
89	VDD	-	Power		3720	117	3720	74.105
90	GND	-	Power		3920	117	3920	74.105
91	VDDIO	-	Power		4120	117	4120	74.105
92	VDDA	-	Power		4320	117	4320	74.105
93	VDDA	-	Power		4520	117	4520	74.105

94	GNDA	-	Power		4720	117	4720	74.105
95	VDD	-	Power		4920	117	4920	74.105
96	GND	-	Power		5120	117	5120	74.105
97	VDDIO	-	Power		5320	117	5320	74.105
98	GNDA	-	Power		5520	117	5520	74.105
99	VDDA	-	Power		5720	117	5720	74.105
100	GNDA	-	Power		5920	117	5920	74.105
101	VDD	-	Power		6120	117	6120	74.105
102	GND	-	Power		6320	117	6320	74.105
103	VDDIO	-	Power		6520	117	6520	74.105
104	VDDA	-	Power		6720	117	6720	74.105
105	VDDA	-	Power		6920	117	6920	74.105
106	GNDA	-	Power		7120	117	7120	74.105
107	VDD	-	Power		7320	117	7320	74.105
108	GND	-	Power		7520	117	7520	74.105
109	VDDIO	-	Power		7720	117	7720	74.105
110	GNDA	-	Power		7920	117	7920	74.105
111	VDDA	-	Power		8120	117	8120	74.105
112	GNDA	-	Power		8320	117	8320	74.105
113	VDD	-	Power		8520	117	8520	74.105
114	GND	-	Power		8720	117	8720	74.105
115	VDDIO	-	Power		8920	117	8920	74.105
116	VDDA	-	Power		9120	117	9120	74.105
117	VDDA	-	Power		9320	117	9320	74.105
118	GNDA	-	Power		9520	117	9520	74.105
119	VDD	-	Power		9720	117	9720	74.105
120	GND	-	Power		9920	117	9920	74.105
121	VDDIO	-	Power		10120	117	10120	74.105
122	GNDA	-	Power		10320	117	10320	74.105
123	VDDA	-	Power		10520	117	10520	74.105
124	GNDA	-	Power		10720	117	10720	74.105
125	VDD	-	Power		10920	117	10920	74.105
126	GND	-	Power		11120	117	11120	74.105
127	VDDIO	-	Power		11320	117	11320	74.105
128	VDDA	-	Power		11520	117	11520	74.105
129	GND	-	Power		11720	117	11720	74.105
130	GNDA	-	Power		11920	117	11920	74.105
131	VDDA	-	Power		12120	117	12120	74.105
132	GNDA	-	Power		12320	117	12320	74.105
133	GNDA	-	Power		12520	117	12520	74.105
134	GNDA33	-	Power		12720	117	12720	74.105
135	GND	-	Power		12920	117	12920	74.105
136	GND	-	Power		13120	117	13120	74.105

137	VDD	-	Power		13320	117	13320	74.105
138	GND	-	Power		13520	117	13520	74.105
139	VDD	-	Power		13720	117	13720	74.105
140	GND	-	Power		13920	117	13920	74.105
141	pENABLE_INn (left)	Input	SLVS pair	Redundant pad, connection optional.	173.8	1710	216.695	1710
142	nENABLE_INn (left)				173.8	1510	216.695	1510
143	pDCLOCK_IN (left)	Input	SLVS pair	Redundant pad, connection optional.	173.8	1310	216.695	1310
144	nDCLOCK_IN (left)				173.8	1110	216.695	1110
145	pDATA_IN (left)	Input	SLVS pair	Redundant pad, connection optional.	173.8	910	216.695	910
146	nDATA_IN (left)				173.8	710	216.695	710
147	pENABLE_OUTn (right)	Output	SLVS pair	Redundant pad, connection optional.	13966.2	1710	13923.305	1710
148	nENABLE_OUTn (right)				13966.2	1510	13923.305	1510
149	pDCLOCK_OUT (right)	Output	SLVS pair	Redundant pad, connection optional.	13966.2	1310	13923.305	1310
150	nDCLOCK_OUT (right)				13966.2	1110	13923.305	1110
151	pDATA_OUT (right)	Output	SLVS pair	Redundant pad, connection optional.	13966.2	910	13923.305	910
152	nDATA_OUT (right)				13966.2	710	13923.305	710

Chip Programming

The Timepix pins M0 and M1 are removed from the Timepix2 interface. Instead, Timepix2 is controlled by a command-based protocol that permits writing to and reading back from registers in the chip periphery.

Resets

There are two types of resets on Timepix2. A global reset wirebond pad (GLOBALRESETn) is used to initialise registers on the chip, including memory and counters in the pixels. Four programming command sequences called DATARESETA, DATARESETB, DATARESETC, and DATARESETD are used to clear the counters in the pixels.

GlobalReset

A GlobalReset takes place when the GLOBALRESETn wirebond pad is asserted for at least 824 cycles of DCLOCK_IN, with DCLOCK_IN toggling the entire time that GLOBALRESETn is low. A GlobalReset initialises the chip, including resetting all programming registers in the chip periphery to their default values, initialising state machines, and resetting all memory and counters in the pixels. To avoid power spikes in the chip, not all blocks are initialised at once, and at least 824 clock cycles are required to completely initialise the chip.

DataReset

Whereas a GlobalReset is executed by the assertion of a wirebond pad, DataResets are executed through commands (see Section Programming Protocols). The DataResets clear counters in the pixels. There are four separate DataReset commands to clear each counter respectively: DATARESETA, DATARESETB, DATARESETC, and DATARESETD. In addition to counter clearing, the DataReset schemes are intended for use in the simultaneous ToT/ToA modes, or as potential workarounds for unforeseen chip behaviour. If a DataReset is performed in one of the continuous read/write modes, there are some caveats to be noted:

Resetting a continuous read/write counter on the fly should only be done with the correct Shutter polarity for a given mode (see

1) Table 15).
2) Combinations of continuous read/write counters should be reset in sequence without toggling the Shutter (e.g. in 14-bit continuous read/write modes, CounterA and CounterD should be reset in sequence, and CounterB and CounterC should be reset in sequence).

Table 15. Resetting the counters in specific modes

Mode	Value on SHUTTER Pin	Resettable Counters
ToT10/ToA18	Low	CounterA, CounterB, CounterC, CounterD
ToT14/ToA14	Low	CounterA, CounterB, CounterC, CounterD
ContiToT10	Low	CounterA, CounterD
	High	CounterB, CounterC
ContiToT14	Low	CounterA, CounterD
	High	CounterB, CounterC
ContToA10	Low	CounterA, CounterD
	High	CounterB, CounterC
ContToA14	Low	CounterA, CounterD
	High	CounterB, CounterC
ContCount10	Low	CounterA, CounterD
	High	CounterB, CounterC
ContCount14	Low	CounterA, CounterD
	High	CounterB, CounterC

Note: The counters are also cleared during a regular readout. Therefore, DataResets are not necessary unless the user would like to quickly clear the counters during a measurement.

The Enable Pins

The active low ENABLE_INn and ENABLE_OUTn pins are used as chip select signals for data communication. This permits multiple chips to be daisy-chained in a readout system, with only one chip active for communication at a time. Following the falling edge of ENABLE_INn, the chip samples the DATA_IN pin on the rising edge of DCLOCK_IN.

Registers in the Chip Periphery

Global programming (or status monitoring) on the chip is stored in the registers listed in Table 16. Details on each register are provided in the following subsections.

Table 16. Programming Registers in the Chip Periphery

Register Name	# Bits	Default value (after GlobalReset)	Programmable?
OMR	16	0x017C	Programmable
TOAFREQSEL	5	0x03	Programmable
DACOUTSEL	5	0x1F	Programmable
NTESTPULSES	16	0x8000	Programmable
TPLENGTH	16	0x2710	Programmable
COLHITTHRESHOLD	9	0x0080	Programmable
DIGPROBESEL_0	8	0x00	Programmable
DIGPROBESEL_1	8	0x00	Programmable
COLMASKREG	256	0x0	Read-only
CTPR	256	0x0	Read-only
COLHITREG_S0	256	0x0	Read-only
COLHITREG_S1	256	0x0	Read-only
COLHITCTR_S0	9	0x0000	Read-only
COLHITCTR_S1	9	0x0000	Read-only
SHUTTIMEREG_S0	24	0x000000	Read-only
SHUTTIMEREG_S1	24	0x000000	Read-only
VBIAS_PREAMP_ON	8	0x07	Programmable
VBIAS_PREAMP_OFF	4	0x08	Programmable
VBIAS_LS_ON	8	0x07	Programmable
VBIAS_LS_OFF	4	0x08	Programmable
VCASC_PREAMP	8	0x80	Programmable
VFBK	8	0x80	Programmable
VTHCOARSE	4	0x08	Programmable
VTHFINE	10	0x0200	Programmable
VBIAS_IKRUM	8	0x07	Programmable
VBIAS_DISCPMOS	8	0x07	Programmable
VBIAS_DISCNMOS	8	0x07	Programmable
VCASC_DISC	8	0x80	Programmable
VBIAS_THS	8	0x07	Programmable
VGND	8	0x80	Programmable
VTPCOARSE	8	0x80	Programmable
VTPFINE	10	0x0200	Programmable
VBIAS_SLVS	8	0x07	Programmable
VCM_SLVS	8	0x08	Programmable
VBIAS_RES	4	0x08	Programmable
CHIPID	32	0x00000000 (before burning)	Burnable
CHIPIDADDR	5	0x00	Programmable
MASKZCS	256	0x0	Programmable
PREAMP_OUT_SEL	256	0x0	Programmable

The Operation Mode Register (OMR)

The Operation Mode Register (OMR) stores the chip-level programming data in Table 17.

Note: Changing the polarity selection bit may trigger false counts as the Discriminator polarity output mux changes state. In the sequential read/write mode, it is best to set the IO pad SHUTTERn = 1 while changing the OMR.

Table 17. Operation Mode Register Bit Assignments

OMR Index	Signal Name	Description
15:12	OpMode[3:0]	**Pixel operation mode** 0x0: TOT10/TOA18 - 1st hit 10b ToT and 18b ToA (default OMR value after GlobalReset, also default if user tries to program any value not on this list) 0x1: TOT14/TOA14 - 1st hit 14b ToT and 14b ToA 0x2: CONTTOT10 – continuous read/write 10b 1st hit ToT (and 4b event counting) 0x3: CONTTOT14 – continuous read/write 14b 1st hit ToT 0x4: CONTTOA10 – continuous read/write 10b ToA 0x5: CONTTOA14 – continuous read/write 14b ToA 0x6: CONTCOUNT10 – continuous read/write 10b event counting 0x7: CONTCOUNT14 – continuous read/write 14b event counting 0x8: ITOT10/TOA18 - integral 10b ToT and 18b ToA 0x9: ITOT14/TOA14 – integral 14b ToT and 14b ToA 0xA: CONTITOT10 – continuous read/write integral 10b ToT (and 4b event counting) 0xB: CONTITOT14 – continuous read/write integral 14b ToT
11	En_SPixClkGate	**Enable superpixel clock gating** Active low, default is 0 (enabled)
10	PolaritySel	**Sensor polarity select** 0: holes, 1: electrons – default is 0 (holes)
9	En_AnalogMasking	**Enable analogue frontend masking/power down** Active low, default is 0 (enabled)
8	En_AdaptiveGain	**Enable adaptive gain in the analogue frontend preamplifier** Active low, default is 1 (disabled)
7	*unassigned*	Hardwired to 0
6	EnIO_PIXEL_PROBE	**Enable the wirebond pad PREAMP_OUT** Active low, default is 1 (disabled)
5	EnIO_MATRIX_OCCn	**Enable the wirebond pad MATRIX_OCCn** Active low, default is 1 (disabled)
4	EnIO_DIGPIXIN	**Enable the digital pixels in the periphery (and enable the wirebond pads DIGPIXIN[7:0]** Active low, default is 1 (disabled)
3	En_StaggeredShutter	**Enable staggering of Shutter signal across double columns in the pixel matrix** Active low, default is 1 (disabled) If enabled, the internal Shutter is staggered into 8 DCLOCK_IN cycles and distributed across the double columns in the pixel matrix. Note: Regardless of whether this signal is enabled or not, the internal Shutter is synchronized to MCLOCK within the chip.
2	En_ShutterTimer	**Enable Shutter Timer block** Active low, default is 1 (disabled)
1	En_ColPairClkGate	**Enable double column clock gating at the end of column** Active low, default is 0 (enabled) If enabled, the ToTClk is gated at the EoC if both columns in the double column are entirely masked. Similarly, the digital testpulse is gated if no pixel in the double column has the TestBit enabled.
0	En_ToAOverflow	Enable ToA counter overflow Active low, default is 0 (enabled)

ToA Clock Frequency Select (TOAFREQSEL)

The ToA counting clock (ToAClk) is the output of a clock frequency divider in the chip periphery that uses MCLOCK as the input reference clock. Table 18 lists the frequencies that can be selected for the ToAClk.

Table 18. ToAClk frequency selection (assuming f_{MCLOCK} = 100 MHz)

TOAFREQSEL value	ToAClk Frequency
0x00	50 Mhz
0x01	25 MHz
0x02	12.5 MHz
0x03	6.25 MHz
0x04	3.125 MHz
0x05	1.563 MHz
0x06	781.25 kHz
0x07	390.625 kHz
0x08	195.313 kHz
0x09	97.656 kHz
0x0A	48.828 kHz
0x0B	24.414 kHz
0x0C	12.207 kHz
0x0D	6.104 kHz
0x0E	3.052 kHz
0x0F	1.526 kHz
0x10	762.939 Hz
0x11	381.47 Hz
0x12	190.735 Hz
0x13	95.367 Hz
0x14	47.684 Hz
0x15	23.842 Hz
0x16	11.921 Hz
0x17	5.96 Hz
0x18	2.98 Hz
0x19	1.49 Hz
0x1A	0.745 Hz
0x1B	0.373 Hz
0x1C	0.186 Hz
0x1D	0.093 Hz
0x1E	100 MHz (bypass)
0x1F	0 Hz (disable)

DAC Output Select (DACOUTSEL)

The DACOUTSEL register value selects which periphery DAC output to multiplex to the DAC_OUT wirebond pad for monitoring. Table 19 lists the DACOUTSEL codes for the periphery DACs and Table 20 lists the external access codes for non-DAC signals that can also be monitored on DAC_OUT.

Global DACs

Table 19 lists the global biasing DACs in the Timepix2 chip periphery. The default values are the values to which the DAC registers are initialised on GlobalReset. The current DACs do not initialise to their midrange values in order to avoid large power consumption in the pixel matrix during chip initialisation. The recommended nominal values for regular chip operation are also provided[4]. Unless otherwise specified, the nominal value of a DAC to bias a pixel frontend in electron collection mode is the same as in hole collection mode. Commands for programming and reading back individual DAC codes are listed in Table 22 and Table 23, respectively. Please note that the reset (default) values of VBIAS_SLVS and VCM_SLVS are too low for proper operation of the SLVS drivers; please set these DACs to the nominal value before executing any readout.

Table 19. Periphery DAC parameters

DAC	DACOUTSEL code	# Bits	Default value	Nominal value (holes)	Nominal value (electrons)*	DAC Range	DAC Step/LSB
VBIAS_PREAMP_ON	0x00	8	0x07	0x68 (1.5µA)		0-3.7µA	14.45µA
VBIAS_PREAMP_OFF	0x15	4	0x08	0x2 (109.5nA)		0-876nA	54.75nA
VBIAS_LS_ON	0x17	8	0x07	0x5B (2µA)		0-5.6µA	21.88µA
VBIAS_LS_OFF	0x18	4	0x08	0x1 (54.8nA)		0-876nA	54.75nA
VCASC_PREAMP	0x06	8	0x80	0x5F (408.5mV)		0-1.1V	4.3mV
VFBK	0x07	8	0x80	0xBA (799.8mV)	0x80 (550.4mV)	0-1.1V	4.3mV
VTHCOARSE	0x08	4	0x08	Depends on VFBK (~780mV)	Depends on VFBK (~570mV)		64 mV
VTHFINE		10	0x0200				400 µV
VBIAS_IKRUM	0x01	8	0x07	0x05 (2.2nA)		0-112nA	0.44nA
VBIAS_DISCPMOS	0x16	8	0x07	0x45 (754.9nA)		0-2.8µA	10.94nA
VBIAS_DISCNMOS	0x02	8	0x07	0x31 (708.1µA)		0-3.7µA	14.45µA
VCASC_DISC	0x09	8	0x80	0x8C (602mV)		0-1.1V	4.3mV
VBIAS_THS	0x03	8	0x07	0x4C (69.9nA)		0-235nA	0.92nA
VGND	0x0A	8	0x80	0xAE (748.2mV)		0-1.1V	4.3mV
VTPCOARSE	0x04	8	0x80	Quantity of test charge, Q_{test}, depends on the difference in output voltage from these two DACs		0-1.1V	4.3mV
VTPFINE	0x05	10	0x0200				400 µV
VBIAS_SLVS	0x0C	8	0x07	0xC8			
VCM_SLVS	0x0B	8	0x08	0x80		0-1.1V	4.3mV
VBIAS_RES	N/A	4	0x08	0x08 (29.5nA)		24-40nA	1nA

*If not specified, then the nominal value is the same as for hole collection

[4] Note to users: The nominal values provided here are derived from simulation. If you find that different setting are more practical in real use, please let me know so I can update the manual (winnie.wong'at'cern.ch)

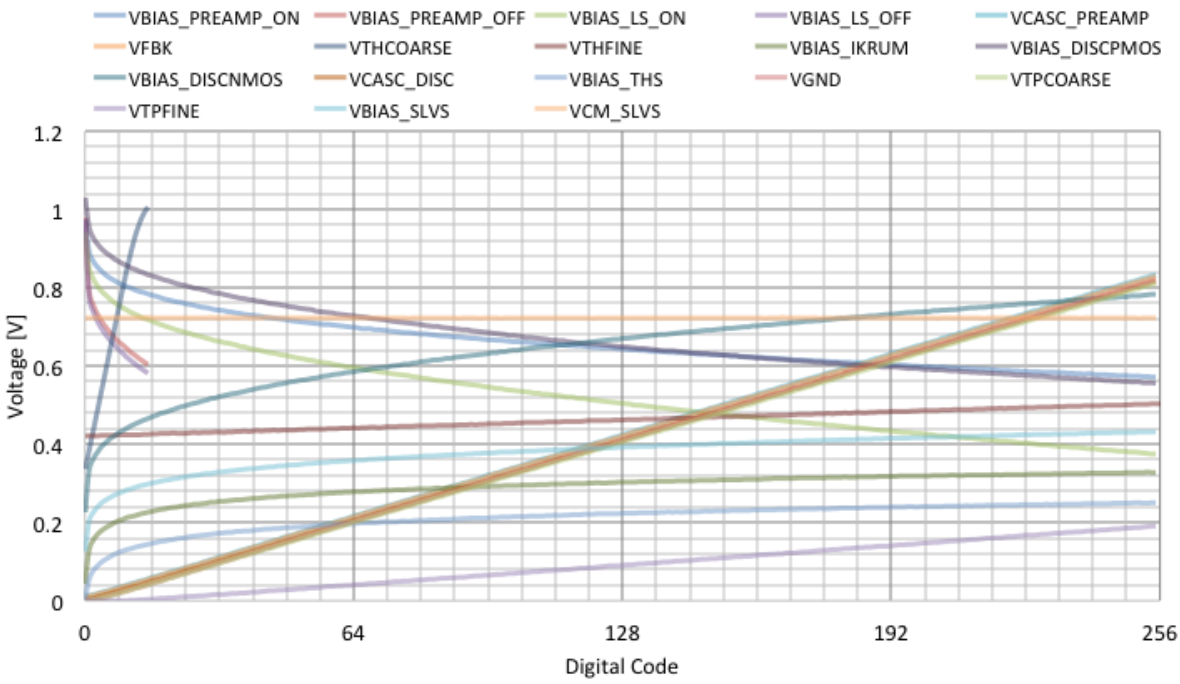

Figure 11. DAC scans (chip measurement)

Band Gap Voltage Reference and Temperature Sensor

The BandGap circuitry is used to generate a stable voltage reference for the DACs. A PTAT circuit is used to monitor the temperature onchip. A simulation of this circuit is shown in Figure 12 which shows a ~1.9 mV/ºC temperature dependent slope.

The onchip temperature is calculated by measuring the PTAT temperature (vbg_tmp) and the Band_Gap Output (vbg) internal monitoring voltages (see Table 20). From simulations the slope and offset has been extracted as shown in Figure 12.

Table 20. DACOUTSEL external access codes for DAC_OUT signals that are not periphery DACs

Signal	DACOUTSEL Code	Nominal Voltage	Description
VDD23	0x14	800 mV	Internal VDD monitoring
VDD13	0x13	400 mV	Internal VDD monitoring
VDDA23	0x12	800 mV	Internal VDDA monitoring
VDDA13	0x11	400 mV	Internal VDDA monitoring
bias_dac	0x10	850 mV	Biasing DAC voltage
bias_dac_cas	0x0F	575 mV	Biasing DAC cascode voltage
vbg	0x0E	300-350 mV	Band gap voltage output
vbg_temp	0x0D		Band gap temperature voltage output

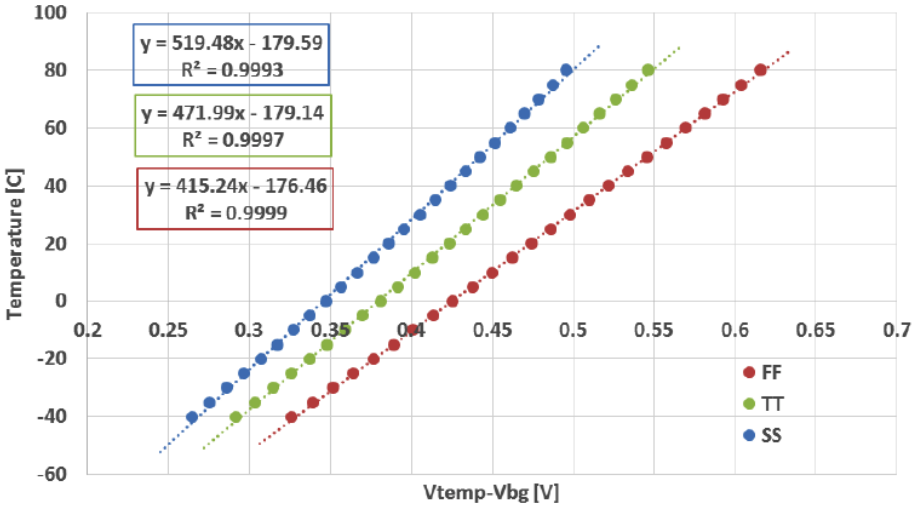

Figure 12. Temperature readout (simulation) [Velopix Manual]

On-chip Supply Voltage Monitoring

The analogue and digital on-chip voltage can be measured through two resistor bridges as shown in Figure 13.

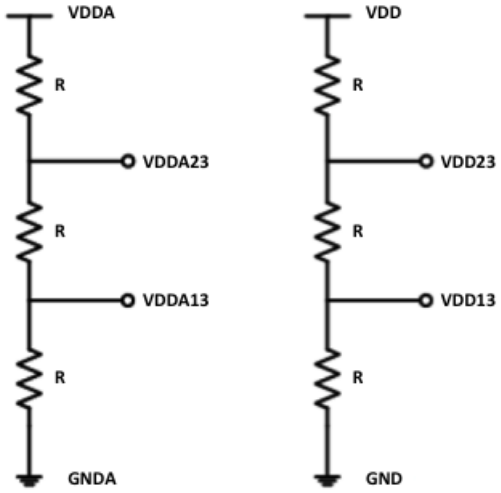

Figure 13. On-chip resistor bridges uses to monitor the core supplies [Velopix Manual]

DAC LSB adjustment

Due to the mismatch between the simulated and measured band gap output voltage, Timepix2 has a 4-bit dynamic range adjustment. This adjustment is configured through the VBIAS_RES register (see Table 19, Table 22 and Table 23). A simulation of an 8-bit voltage DAC (such as VFBK, VGND, VCASC_PREAMP and VCASC_DISC) is shown in Figure 14 for the different settings of VBIAS_RES.

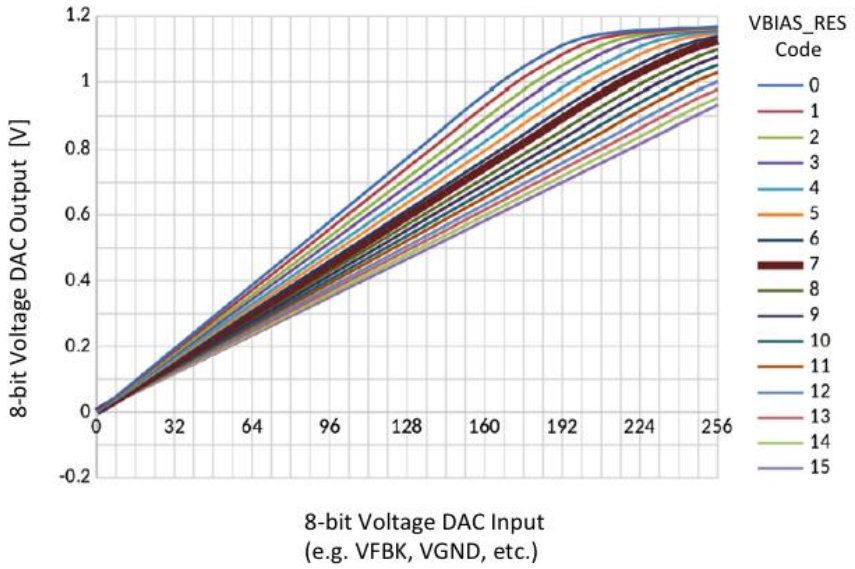

Figure 14. 8b Voltage DAC output for different settings of VBIAS_RES (nominal value in bold) [Velopix Manual].

The NTESTPULSES and TPLENGTH Registers

The NTESTPULSES and TPLENGTH registers must be set prior to executing the Generate AnalogTP and Generate DigitalTP commands. The Test Pulse (whether analogue or digital) has a 50% duty cycle. TPLENGTH is the width of half of the Test Pulse period. The value set in TPLENGTH is in units of number of periods of MCLOCK and must be > 1.

The COLHITTHRESHOLD Registers and the MATRIX_OCCn IO pin

When enabled by the OMR, the MATRIX_OCCn wirebond pad asserts when the number of columns that contain hits exceeds the value stored in the register COLHITTHRESHOLD.

$$0 > \text{COLHITTHRESHOLD} \leq 256$$

The DIGPROBESEL_0 and DIGPROBESEL_1 Registers

The DIGPROBESEL_0 and DIGPROBESEL_1 registers select which internal chip signals are multiplexed to the DIGITAL_OUT[0] and DIGITAL_OUT[1] wirebond pads, respectively.

Table 21. DIGPROBESEL mux selection

DIGPROBESEL_0/1 value	Signal on DIGITAL_OUT[0/1]
0x01	fsrSerialOut_FSR0
0x02	fsrSerialOut_FSR1
0x03	fsrStartClk0
0x04	fsrTieColInTo0
0x05	fsrSerialData
0x06	fsrSerialData_FSR2
0x07	fsrClear_
0x08	fsrParallelLoad
0x09	fsrSuppressZeroCols
0x0A	fsrLoadColEn
0x0B	fsrResetColEn_
0x0C	fsrLoadFSRtoReg
0x0D - 0x0F	fsrDataSel[0:2]
0x10 - 0x11	fsrPathSel[0:1]
0x12	fsrClk0En
0x13	fsrClk1En
0x14 - 0x1B	pixDataClkEn_[0:7]
0x1C - 0x23	pixDataReset_[0:7]
0x24 - 0x2B	pixDataReset_Delay_[0:7]
0x2C - 0x33	pixReset_[0:7]
0x34 - 0x3B	pixReadA[0:7]
0x3C - 0x43	pixReadB[0:7]
0x44 - 0x4B	pixReadC[0:7]
0x4C - 0x53	pixReadD[0:7]
0x54 - 0x57	pixMode[0:3]
0x58 - 0x5F	pixLoadConfLatches[0:7]
0x60 - 0x67	pixLoadTrimLatches[0:7]
0x68 - 0x6F	pixParallelLoadCD[0:7]
0x70	pixTPDigital
0x71	pixToAClk
0x72	pixAnalogTestEn
0x73	pixAnalogMaskEn
0x74	pixPolaritySel
0x75	pixDisToAOF
0x76	pixIsSimMode
0x77	pixIsMode0or8
0x78	pixIsMode2or10
0x79	pixIs14BitMode
0x7A	pixIsToTMode
0x7B	pixIsToAMode
0x7C	pixIsEventMode
0x7D	pixIntegrateToT
0x7E	pixIsLatchMode
0x7F	pixChangeMode
0x80	pixfreezeCOLHITREG
0x81	pixfsrLoadColEn
0x82	pixAdaptiveGainEn

0x83	pixColPairClkGateEn
0x84 - 0x8B	pixShutter[0:7]
0x8C - 0x93	pixShutter_PreEdge[0:7]
0x94	efuseLD
0x95	efusePROG
0x96	DIGPIXIN_ie
0x97	PBUS_ACCESSn_oen
0x98	DIGPROBE_oen
0x99	MATRIX_OCCn_oen
0x9A	PIXEL_PROBE_oen
0x9B	pixDisSPixClkGate
0x9C	InitPeriphery_
0x9D	InitEoC_
0x9E	InitCounterA_
0x9F	InitCounterB_
0xA0	InitCounterC_
0xA1	InitCounterD_
0xA2	InitPixLatches_
0xA3	InitMatrix_
0xA4	InitFSR_

The COLMASKREG Register

The COLMASKREG register is a 256-bit read-only register, where each bit corresponds to a column in the pixel matrix. A value of '1' in a register bit indicates that all pixels in that column have been masked. A value of '0' means that at least one pixel in that column is enabled. The MSB of COLMASKREG corresponds to the rightmost column, and the LSB corresponds to the leftmost column.

The CTPR Register

The Column Test Pulse Register (CTPR) register is a 256 bit read-only register, where each bit corresponds to a column in the pixel matrix. A value of '1' in a register bit indicates that at least one pixel in that column has an enabled TestBit. The MSB of CTPR corresponds to the rightmost column, and the LSB corresponds to the leftmost column. Note: In the original Timepix, the CTPR was a register that needed to be written to the chip. In Timepix2, this register is generated automatically and can be read out for digital diagnostics purposes.

The COLHITREG_S0 and COLHITREG_S1 Registers

The COLHITREG_S0 and COLHITREG_S1 registers are 256-bit read-only registers, where each bit corresponds to a column in the pixel matrix. A value of '1' in COLHITREG_S0 register bit indicates that there is at least one hit processed in that column while the SHUTTERn/COUNTERSEL wirebond pad was low. Similarly, a value of '1' in COLHITREG_S1 indicates a hit while the SHUTTERn/COUNTERSEL input was high (the COLHITREG_S1 register value is only meaningful in the continuous read/write modes).

The COLHITCTR_S0 and COLHITCTR_S1 Registers

The COLHITCTR_S0 and COLHITCTR_S1 registers are read-only registers that store the output of 9-bit adders that count the number of 1's in COLHITREG_S0 and COLHITREG_S1, respectively. The COLHITCTR_S1 register value is only meaningful in the continuous read/write modes.

The SHUTTIMEREG_S0 and SHUTTIMEREG_S1 Registers

The SHUTTIMEREG_S0 and SHUTTIMEREG_S1 registers are read-only registers that store the output of 24-bit counters that count the number of ToAClk cycles during which the wirebond pad SHUTTERn/COUNTERSEL is low and high, respectively. The SHUTTIMEREG_S1 register value is only meaningful in the continuous read/write modes. Note: To avoid unnecessary power consumption, the Shutter Timer blocks need to be enabled in the OMR.

The CHIPID and CHIPIDADDR Registers

The Timepix2 chip contains 32 electronic fuses (e-fuses) to be used as a unique chip identifier. The CHIPIDADDR register stores the address of the e-fuse bit to be blown by the command Burn CHIPID. The CHIPID register is read-only and outputs the values of the e-fuses; a '1' denotes blown e-fuse.

The MASKZCS Register

The MASKZCS register is a 256-bit register where each bit corresponds to a column in the pixel matrix. It is intended as a failsafe/workaround in case any columns display erroneous behaviour during zero column suppression (ZCS) readout. When a bit is '1', the associated column is masked for readout in ZCS mode. The MSB of MASKZCS corresponds to the rightmost column, and the LSB corresponds to the leftmost column.

The PREAMP_OUT_SEL Register

For monitoring purposes, the preamplifier output of one single pixel in the bottom row of the matrix can be selected for output to the wirebond pad PREAMP_OUT (which needs to be enabled in the OMR). The PREAMP_OUT_SEL register is a 256-bit register where each bit corresponds to a column in the pixel matrix. *Up to only one bit* can be high at a given time. The MSB of PREAMP_OUT_SEL corresponds to the rightmost column, and the LSB corresponds to the leftmost column.

Programming Protocols

The slow control is programmed through a command-based protocol. Inputs to the chip are expected MSB-first. Outputs from the chip are also shifted out MSB-first.

Possible codes for SET commands are shown in Table 22.

Table 22. Set programming register commands

Command	Payload size	Effective payload size	Default value	COM/FIN/ACK code
Set OMR	16	16	0x017C	0x80
Set TOAFREQSEL	5	8	0x03	0x81
Set DACOUTSEL	5	8	0x1F	0x82
Set NTESTPULSES	16	16	0x8000	0x83
Set TPLENGTH	16	16	0x2710	0x84
Set COLHITTHRESHOLD	9	16	0x0080	0x85
Set DIGPROBESEL_0	8	8	0x00	0x86
Set DIGPROBESEL_1	8	8	0x00	0x87
Set VBIAS_PREAMP_ON	8	8	0x07	0x90
Set VBIAS_PREAMP_OFF	4	8	0x08	0x91
Set VBIAS_LS_ON	8	8	0x07	0x92
Set VBIAS_LS_OFF	4	8	0x08	0x93
Set VCASC_PREAMP	8	8	0x80	0x94
Set VFBK	8	8	0x80	0x95
Set VTHCOARSE	4	8	0x08	0x96
Set VTHFINE	10	16	0x0200	0x97
Set VBIAS_IKRUM	8	8	0x07	0x98
Set VBIAS_DISCPMOS	8	8	0x07	0x99
Set VBIAS_DISCNMOS	8	8	0x07	0x9A
Set VCASC_DISC	8	8	0x80	0x9B
Set VBIAS_THS	8	8	0x07	0x9C
Set VGND	8	8	0x80	0x9D
Set VTPCOARSE	8	8	0x80	0x9E
Set VTPFINE	10	16	0x0200	0x9F
Set VBIAS_SLVS	8	8	0x07	0x88
Set VCM_SLVS	8	8	0x08	0x89
Set VBIAS_RES	4	8	0x08	0x8A
Set CHIPIDADDR	5	8	0x00	0xC0
Set MASKZCS	256	256	0x0	0xC1
Set PREAMP_OUT_SEL	256	256	0x0	0xC2

The generic protocol for SET commands is shown in Figure 15. When ENABLE_INn is set to 0, the chip starts to look for a valid COM pattern in the input data. The data is sampled on the rising edge of the input clock DCLOCK_IN. After two valid and identical COM headers and 8 clock cycles, the data payload should be sent in, and the chip should send back the COM twice, meaning the command was understood properly. Then the chip will send an ACK byte, which is the usually same COM, but for non-recognised commands there is a special ACK code 0xCF. After this, the chip will echo back the data payload and the command finishes with a FIN byte at the data output. If the payload size is

not a multiple of a byte, the payload should be padded with 0's at the MSB to reach the effective payload size.

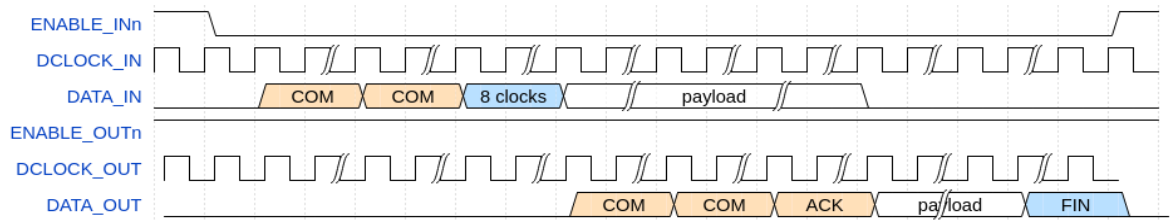

Figure 15: Slow control protocol for SET command.

Figure 16 shows a more detailed execution of SETOMR command. This command structure follows the generic SET command protocol.

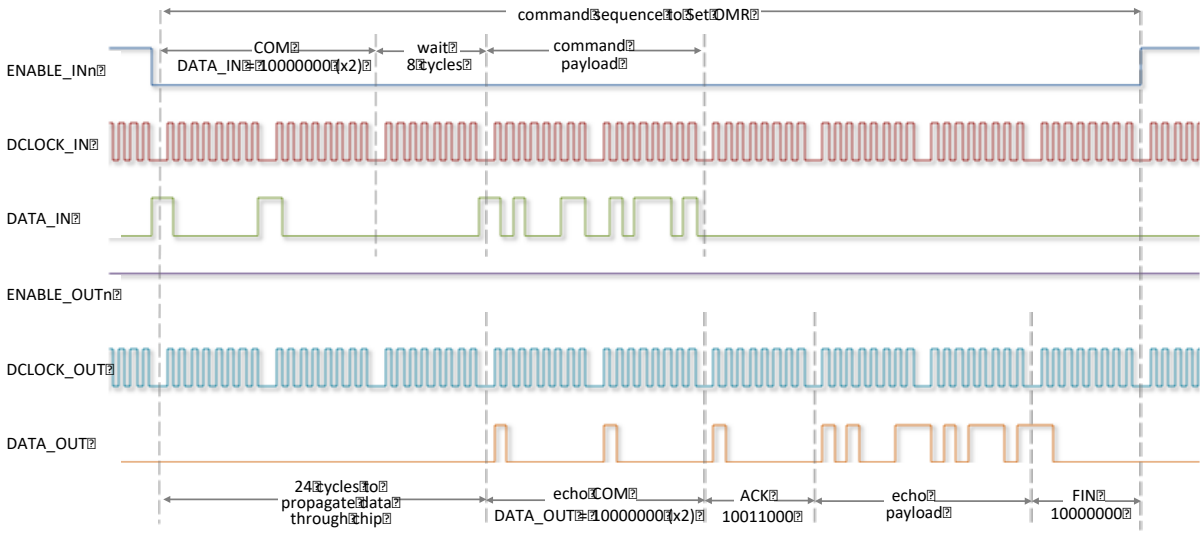

Figure 16. Command protocol for SETOMR

Table 23. Get programming register commands

Command	Payload size	Effective payload size	Default value	COM/FIN/ACK code
Get OMR	16	16	0x017C	0xA0
Get TOAFREQSEL	5	8	0x03	0xA1
Get DACOUTSEL	5	8	0x1F	0xA2
Get NTESTPULSES	16	16	0x8000	0xA3
Get TPLENGTH	16	16	0x2710	0xA4
Get COLHITTHRESHOLD	9	16	0x0080	0xA5
Get DIGPROBESEL_0	8	8	0x00	0xA6
Get DIGPROBESEL_1	8	8	0x00	0xA7
Get COLMASKREG	256	256	0x0	0xA8
Get CTPR	256	256	0x0	0xA9
Get COLHITREG_S0	256	256	0x0	0xAA
Get COLHITREG_S1	256	256	0x0	0xAB
Get COLHITCTR_S0	9	16	0x0000	0xAC

Get COLHITCTR_S1	9	16	0x0000	0xAD
Get SHUTTIMEREG_S0	24	24	0x000000	0xAE
Get SHUTTIMEREG_S1	24	24	0x000000	0xAF
Get VBIAS_PREAMP_ON	8	8	0x07	0xB0
Get VBIAS_PREAMP_OFF	4	8	0x08	0xB1
Get VBIAS_LS_ON	8	8	0x07	0xB2
Get VBIAS_LS_OFF	4	8	0x08	0xB3
Get VCASC_PREAMP	8	8	0x80	0xB4
Get VFBK	8	8	0x80	0xB5
Get VTHCOARSE	4	8	0x08	0xB6
Get VTHFINE	10	16	0x0200	0xB7
Get VBIAS_IKRUM	8	8	0x07	0xB8
Get VBIAS_DISCPMOS	8	8	0x07	0xB9
Get VBIAS_DISCNMOS	8	8	0x07	0xBA
Get VCASC_DISC	8	8	0x80	0xBB
Get VBIAS_THS	8	8	0x07	0xBC
Get VGND	8	8	0x80	0xBD
Get VTPCOARSE	8	8	0x80	0xBE
Get VTPFINE	10	16	0x0200	0xBF
Get VBIAS_SLVS	8	8	0x07	0xC8
Get VCM_SLVS	8	8	0x08	0xC9
Get VBIAS_RES	4	8	0x08	0xCA
Get CHIPID	32	32	0x00000000	0xC3
Get CHIPIDADDR	5	8	0x00	0xC4
Get MASKZCS	256	256	0x0	0xC5
Get PREAMP_OUT_SEL	256	256	0x0	0xC6

The generic protocol for GET command is shown in Figure 17. After the chip receives two valid COM bytes, it will respond by echoing the COM twice, followed by ACK. In the case of ACK = 0xCF, the command is not recognised. The payload follows immediately after the ACK, and a Get command finishes with a FIN byte (which should be identical to COM). Note that the receiver cannot determine the length of the payload from the data stream, as the payload may contain bytes identical to FIN.

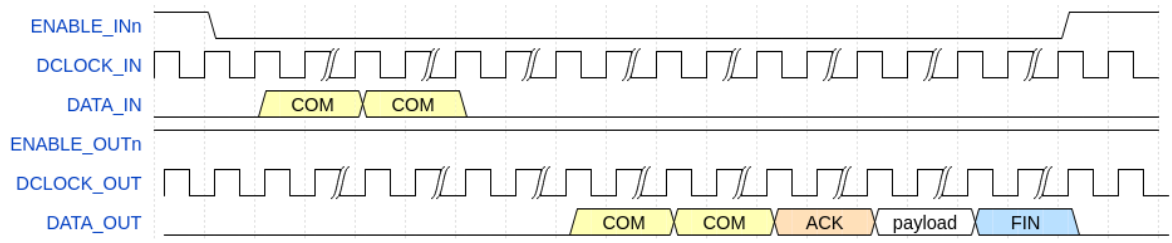

Figure 17: Slow control protocol for GET command.

Finally, Figure 18 shows two back-to-back commands for setting and getting the VBIAS_IKRUM. ENABLE_INn can stay at 0 as long as new commands are sent, but the FIN byte from previous command must be received before a new command can be started.

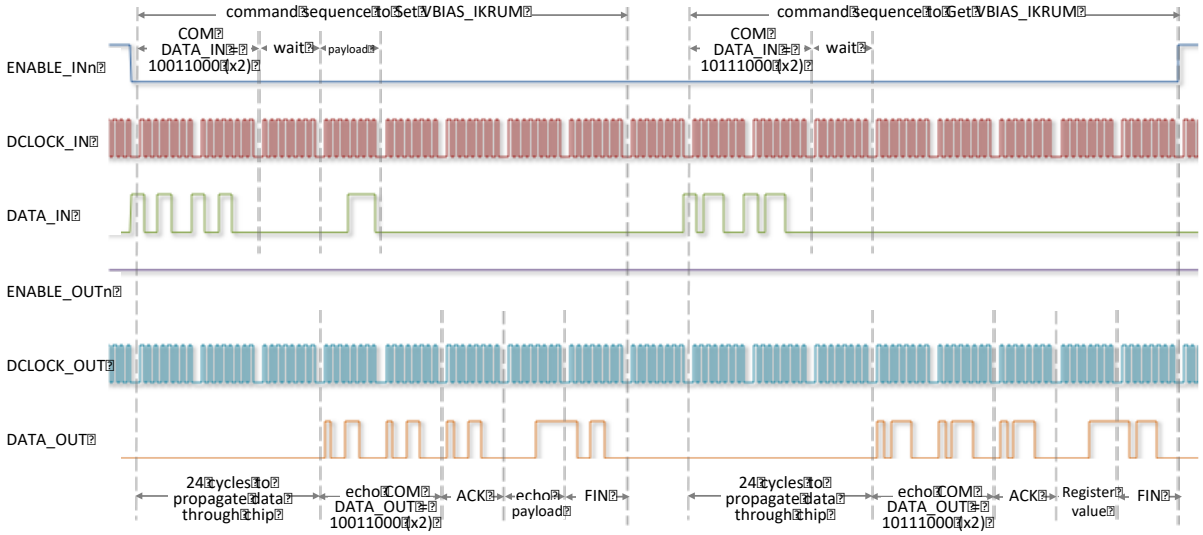

Figure 18. Command protocol for SETVBIAS_IKRUM followed by GETVIBAS_IKRUM

Burning the eFuses (ChipID)

The ChipID is hardcoded by 32 eFuses. During eFuse burning, the LSB of the 32 bits is CHIPIDADDR=0 and the MSB is CHIPIDADDR=31. Table 24 describes the command to burn an eFuse bit.

Table 24. Command to burn eFuse

Command	Payload size	COM/FIN/ACK code
Burn eFuse	0	0xCD

The following lists the steps to burn an eFuse for the ChipID:

1) Program the address of bit to burn through the command Set CHIPIDADDR (0xC0), see Table 22.
2) Apply 2.5V to the pin VDD33.
3) Execute command Burn eFuse.

The Burn eFuse command has no associated payload, but follows the generic protocol of the Get command very closely. To burn an eFuse bit, send the COM code twice. The chip will response by echoing the COM byte twice, followed by an ACK, followed by a long stream of 0s, and finally a FIN byte to mark the completion of the command.

Note: the frequency of DCLOCK should be 70-100MHz during eFuse burning.

Reading the 32-bit ChipID

After the eFuses have been burnt, the ChipID is treated as a read-only register. Note that when reading from the eFuses, 2.5V needs to be applied to pin VDD33.

Unrecognised command

Figure 19 shows how the chip handles an invalid command. The COM is echoed twice on the output, but instead of sending the ACK = COM response, an error ACK (0xCF) is sent out. This is followed by the registered erroneous COM, and finally followed by an error FIN (0xCF).

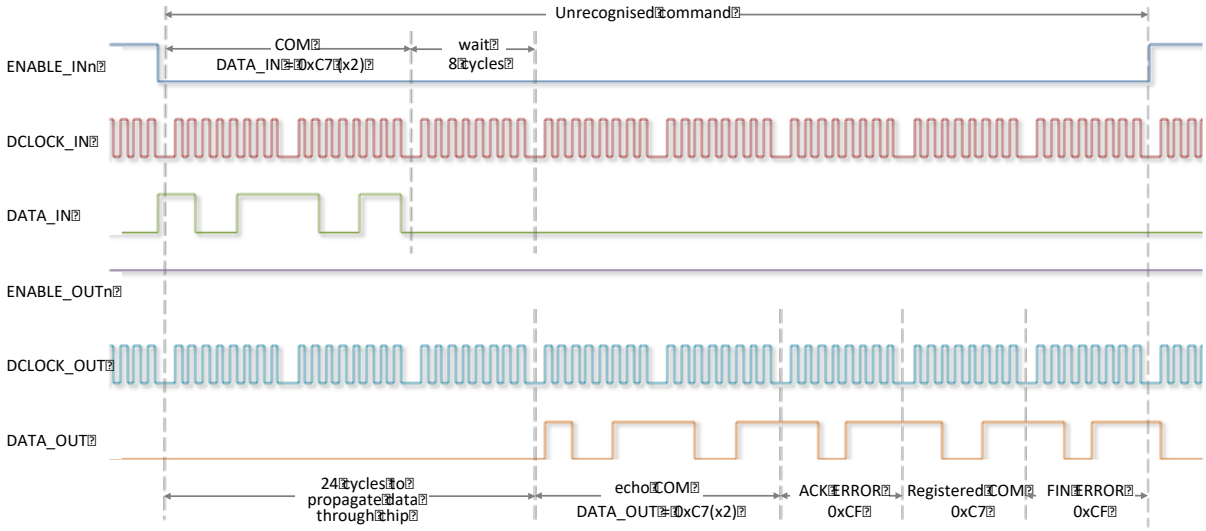

Figure 19. Command protocol in case of an unrecognised command

Writing (SET) and reading (GET) data in the pixels

Table 25 shows the different commands for writing into pixel counters or reading from them. Writing is mainly added for testing purposes, but can be used to set counters to desired initial values. Reading the counter values is done one counter at time, and there are 3 different possibilities: sequential readout via DATA_OUT, parallel readout via DOUT and zero-column suppressed (ZCS) readout via DATA_OUT.

For the ZCS Get commands, a minimum payload length is at least 16 columns, due to internal architectural limitations. Following the ACK, the ZCS Get commands output a 256 column hit map, followed by a 2nd ACK, the payload, and the FIN byte.

Table 26 lists the commands for writing individual pixel settings. Each pixel has two configuration (CONF) bits (see Table 11) and five threshold trim bits (see Table 12).

Counter data pipelining

The payload of the commands of Table 25 and Table 26 follow a data format similar to Medipix2 and Timepix. In Get Counter commands, data is shifted out MSB-first and pipelined through a fast shift register (FSR) at the end of column. Every sequence of 256 bits (or in the case of ZCS mode, every n bits) shifted out of the chip contains a single bit of data from 256 (n) columns. The output data starts from the bottom-right pixel. Figure 20 illustrates the example of data pipelined through the FSR in a Get CounterA command.

Similarly, during SET commands, data is expected to be scrambled in a way that permits it to be pipelined in through the FSR, starting with the MSB of the top-right pixel.

Table 25. Codes for SET and GET pixel counter commands (of pixels in the matrix).

Command	Payload size	Default value, comments	COM/FIN/ACK code
Set COUNTER A	655360	0 after reset	0xF6
Set COUNTER B	655360	0 after reset	0xF7
Set COUNTER C	262144	0 after reset	0xF8
Set COUNTER D	262144	0 after reset	0xF9
Get COUNTER A	655360	0	0xE0
Get COUNTER B	655360	0	0xE1
Get COUNTER C	262144	0	0xE2
Get COUNTER D	262144	0	0xE3
Get COUNTER A Parallel	655360	0	0xE4
Get COUNTER B Parallel	655360	0	0xE5
Get COUNTER C Parallel	262144	0	0xE6
Get COUNTER D Parallel	262144	0	0xE7
Get COUNTER A ZCS	256 + 8 + N x 2560	0, N >= 16	0xE8
Get COUNTER B ZCS	256 + 8 + N x 2560	0, N >= 16	0xE9
Get COUNTER C ZCS	256 + 8 + N x 1024	0, N >= 16	0xEA
Get COUNTER D ZCS	256 + 8 + N x 1024	0, N >= 16	0xEB

Table 26. Codes for SET and GET pixel CONF and TRIM commands.

Command	Payload size	Default value, comments	COM/FIN/ACK code
Set CONF	262144	0 after reset	0xD0
Set TRIM	262144	0 after reset	0xD1
Get CONF	262144	0	0xD8
Get TRIM	262144	0	0xD9

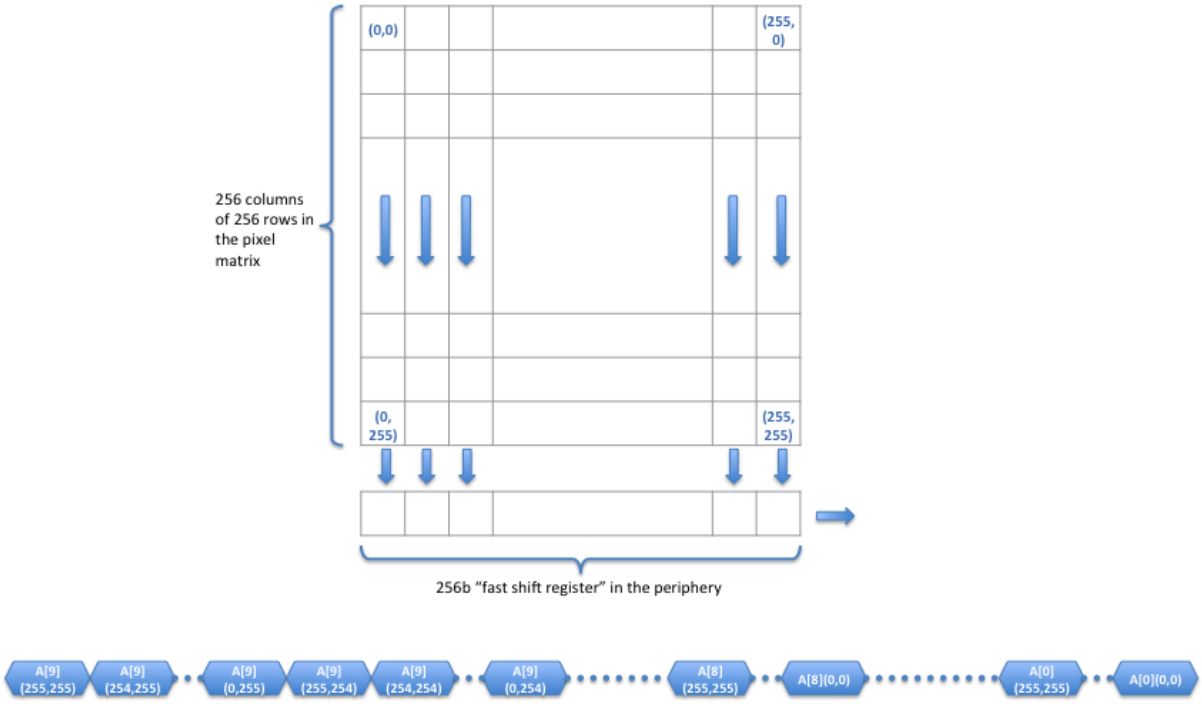

Figure 20. Example: Data pipelined through the fast shift register in a Get CounterA command

Reading data from the digital pixels

In addition to the readout commands of counters in the pixel matrix, a set of digital pixels can be read out from the periphery. These commands are shown in Table 27. The digital pixel functions follow the global programming of the OMR (note: the bit OMR[4] should be enabled to power on the digital pixels), but they do not have any configuration or trim bits associated with them. The counter data from the digital pixels are not pipelined through the FSR; the digital pixels are simply daisy chained and shifted out directly following the same protocol as read-only registers in the periphery.

Table 27. Codes for SET and GET pixel counter commands in the digital pixels of the periphery.

Command	Payload size	Default value, comments	COM/FIN/ACK code
Get COUNTER A DIGPIX	80 (8x 10b)	0	0xEC
Get COUNTER B DIGPIX	80 (8x 10b)	0	0xED
Get COUNTER C DIGPIX	32 (8x 4b)	0	0xEE
Get COUNTER D DIGPIX	32 (8x 4b)	0	0xEF

Pixel counter data reset

Codes for the pixel data reset commands are shown in Table 28. The resets have no associated payload, but they follow the generic protocol of Get command very closely. To perform any of the resets, send the corresponding COM twice. During the response payload, the chip is sending out 0 all the time, and this stream of 0s is finally followed by a corresponding FIN byte.

Table 28. Data reset commands.

Command	Payload size	COM/FIN/ACK code
Reset COUNTER A	0	0xFA
Reset COUNTER B	0	0xFB
Reset COUNTER C	0	0xFC
Reset COUNTER D	0	0xFD

Generating testpulses

Testpulses are generated by state machines on-chip via command programming. Codes for the generate test pulse commands are shown in Table 29. These commands have no associated payload, but they follow the generic protocol of Get command very closely. To perform a generate test pulse command, send the corresponding COM twice. During the response payload, the chip is sending out 0 all the time, and this stream of 0s is finally followed by a corresponding FIN byte.

Table 29. Generate test pulse commands.

Command	Payload size	COM/FIN/ACK code	Description
Generate AnalogTP	0	0xF0	Test charge injection into pixel frontend input via 5fF test capacitance
Generate DigitalTP	0	0xF1	Analogue frontend bypassed

The pin SHUTTERn should be set to 1 before sending the generate command. Following the ACK response from the chip, the test pulse injection can be started by setting SHUTTERn to 0. After the programmed number of test pulses has been generated, the chip will send a FIN byte.

The test pulse parameters are defined by the registers TPLENGTH and NTESTPULSES (see Table 22). The chip generates an internal Shutter signal that follows the IO pin SHUTTERn on the rising edge of MCLOCK_IN, with a latency of two MCLOCK periods. The testpulses are generated with respect to the internal Shutter signal and MCLOCK_IN.

Both digital and analogue test pulses are generated in the same way in the periphery; enabling of the test pulse within pixels is determined by the pixel configuration TestBit (see Table 11).

In the case of analogue test pulses, the quantity of test charge injected into the frontend is:

$$Q_{test} = \Delta V \times C_{test}$$

where ΔV is the difference in the voltages set by the DACs VTPCOARSE and VTPFINE (see Table 22. Set programming register commands), and C_{test} is 5 fF. Q_{test} is generated on each edge of the test pulse (the polarity of the charge depends on the direction of the pulse edge). For analogue test pulses, TPLENGTH should be > 200 µs to permit the frontend to settle between test pulses. The polarity of charge evaluated by the pixel depends on the global polarity setting of the OMR (see Table 17).

For digital test pulses, ToT counts in a pixel under test correspond to TPLENGTH (± 1 count), and ToA measurements are from the rising edge of the first Test Pulse to the rising edge of the internal Shutter. During the Generate DigitalTP command, the chip ignores the PolaritySel setting stored in the OMR, and sets the pixel circuit polarity to be compatible with the polarity of the Test Pulse.

Figure 21 and Figure 22 illustrate the test pulse generation protocol:

- The Test Pulse starts TPLENGTHxPeriod$_{MCLOCK}$ following the opening of the internal Shutter signal (Figure 21, zone 1)
- The Test Pulse stays high for TPLENGTHxPeriod$_{MCLOCK}$, from rising edge to rising edge of MCLOCK (Figure 21, zone 2a).
- The Test Pulse stays low for TPLENGTHxPeriod$_{MCLOCK}$, (50% duty cycle) from rising edge to rising edge of MCLOCK (Figure 21, zone 2b). The delay between the final Test Pulse and the output of the FIN byte provides the frontend enough time to complete processing of the charge injected by the final falling edge of Test Pulse.
- The Test Pulse has a 50% duty cycle. TPLENGTH is the width of half of the Test Pulse period.
- The value set in TPLENGTH is in units of number of periods of MCLOCK must be > 1.

The test pulse generator will not generate test pulses while SHUTTERn is high. In continuous read/write pixel modes, the SHUTTERn IO pin is treated as a shutter, rather than as a counter select. In the pixel matrix, only the counters that count while SHUTTERn is low can be used with test pulses.

It be noted that test pulses should not be applied to all pixels in the matrix simultaneously. Please use a spacing of at least 4.

Note: Due to a timing issue within the slow control state machine, the chip may sometimes (around 1% of the time) execute a single Generate *TP command twice. The recommended workaround (for chips that are not daisy chained) is to deassert the IO pin ENABLE_IN following the output of the FIN byte of a Generate *TP command. For daisy-chained chips, toggling ENABLE_IN would reset the daisy chain command sequence; therefore the recommendation for daisy-chained chips is to permit this error to occur and to include outlier rejection in data analysis routines.

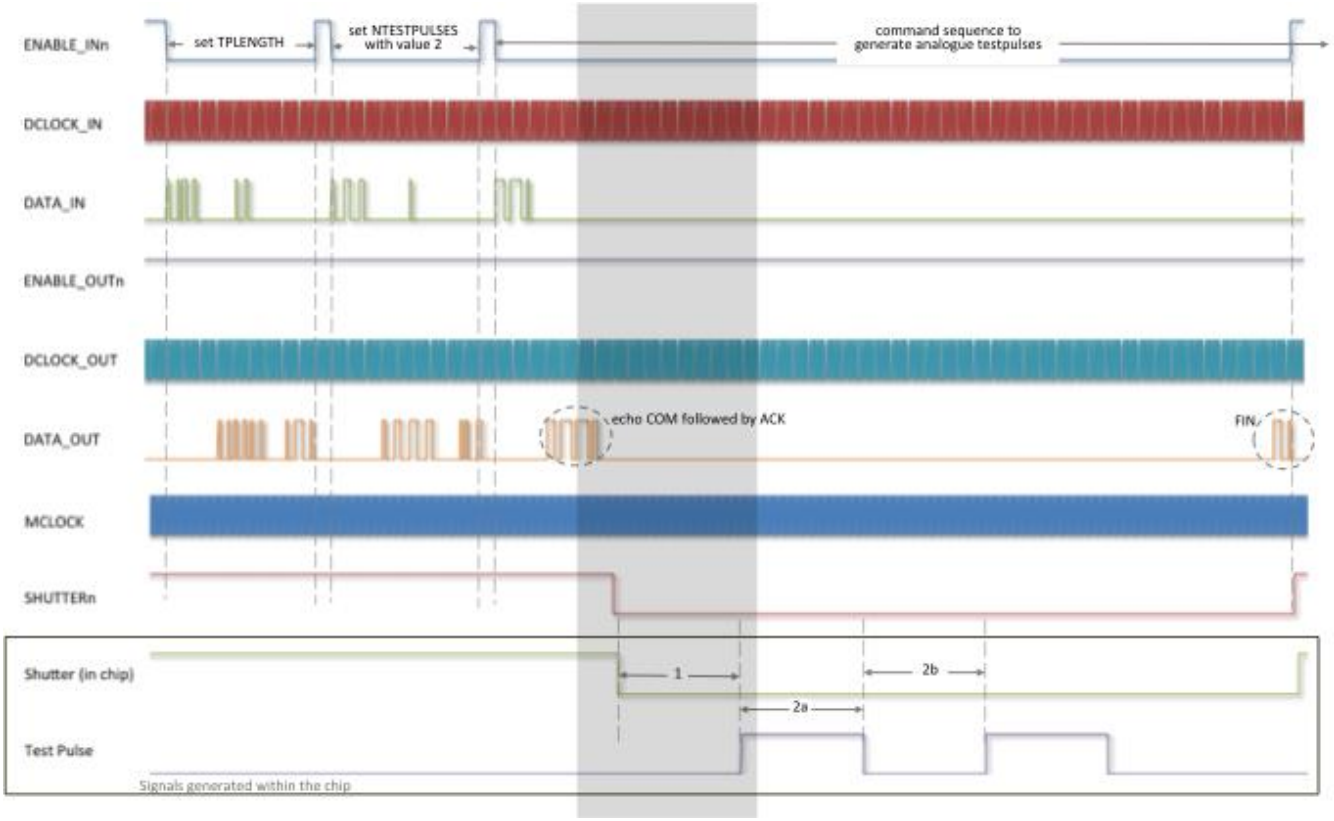

Figure 21. Command protocol and timing of internal signals for test pulse generation

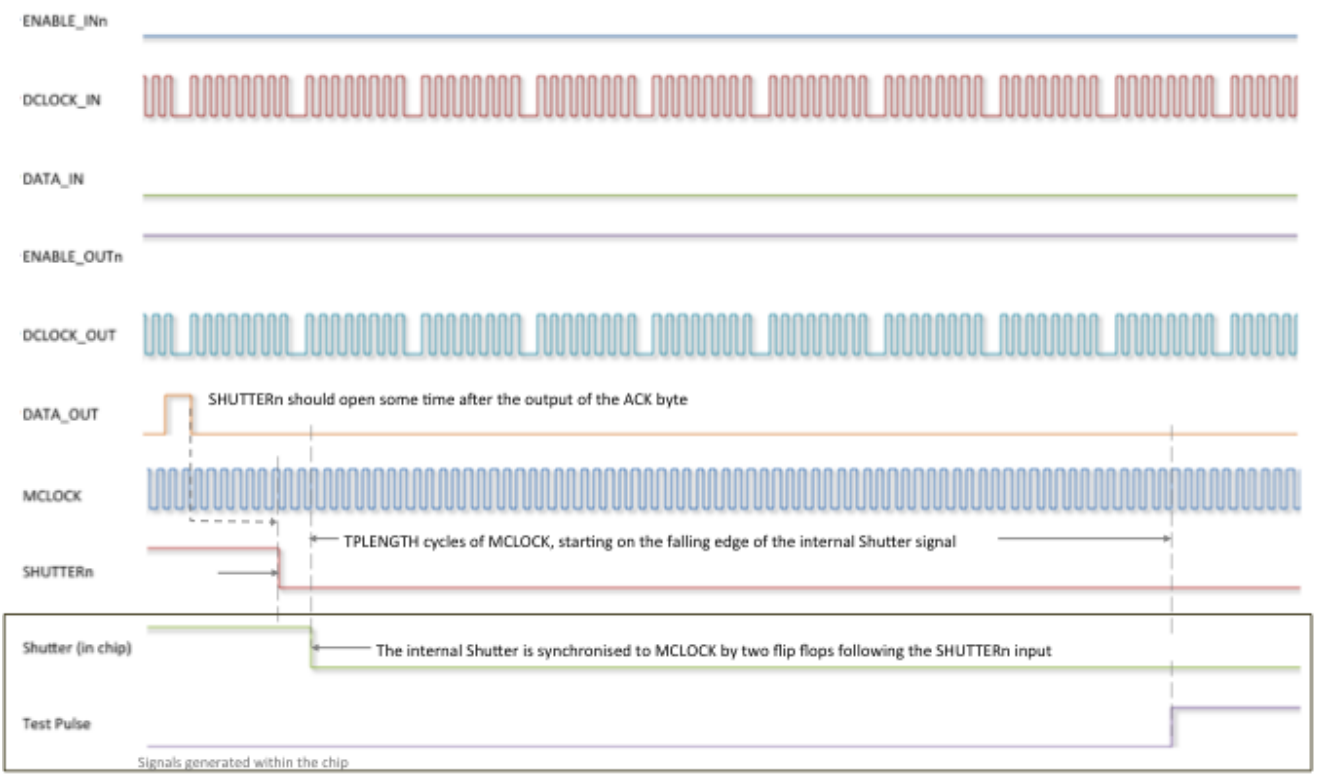

Figure 22. Test pulse generation (zoomed in to the shaded region of Figure 21

Logical Output Flags

The Timepix2 chip has three single-ended CMOS outputs that operate as boolean flags (see Table 14): READREADYn, MATRIX_OCCn, and PBUS_ACCESSn.

Read Ready Flag

In the ToT operation modes, ToT processing of large input charges can continue beyond the closing of the Shutter. The active low READREADYn signal resets to high following the transition of SHUTTERn (both rising and falling edge of SHUTTERn). When no ToT hits are being processed in the pixel matrix, READREADYn asserts to low. Readout from the ToT counters should not be executed until the assertion of READREADYn, or the ToT counters may be corrupted.

Note: At the time of writing this manual, the READREADYn signal had not been fully tested/debugged. Preliminary tests indicate that this signal may be glitchy. If the signal does not work, the recommended workaround is to force the DAQ system to wait a minimum duration of $2^{n-1} \times Period_{MCLOCK}$ before executing a “Get COUNTER*” command.

Matrix Occupancy Flag

When enabled by the OMR, the MATRIX_OCCn wirebond pad asserts when the number of columns that contain hits exceeds the value stored in the register COLHITTHRESHOLD.

$$0 > COLHITTHRESHOLD \leq 256$$

Note: Preliminary tests indicate that this signal may be glitchy – to be determined.

Parallel Bus Access Flag

The PBUS_ACCESSn output asserts when the parallel port is enabled (through a Get Counter * Parallel command), see Table 25.

Daisy Chaining Multiple Chips

Multiple Timepix2 chips can be daisy chained for programming and readout using the Bypass command described in Table 30. The Bypass command puts a chip in a mode where subsequent commands are ignored and propagated to the next chip through the serial DATA_OUT. A single Bypass command puts one chip in bypass, two Bypass commands puts two chips in bypass, etc.

To put a chip in Bypass mode, input the COM code twice. The chip will echo the COM code on DATA_OUT, followed by the FIN byte (note there exceptionally is no ACK in the bypass protocol). Subsequent commands pass through the chip without being executed. This continues until ENABLE_IN is de-asserted.

Table 30. Bypass Command

Command	Payload size	COM/FIN code
Bypass	0	0xF5

Figure 23 illustrates the IO connections between daisy-chained chips and Figure 24 provides an example on how to transmit commands to chips in the daisy chain. In this example, zone 1 shows the bypass command being input to Chip1. Zone 2 is the latency for the bypass command to be processed by Chip1. In Zone 3, Chip1 outputs the echo of the COM byte (twice), followed by FIN on $DATA\_OUT_{Chip1}$. In Zone 5a, the next command sequence (in this example, “set VFBK” with payload 0x83) is ignored by Chip1. In Zone 5b, Chip2 receives the “set VFBK” command and echoes the command in Zone 6a, sends out an ACK in Zone 6b, echoes the payload in Zone 6c, and finally, Chip2 outputs the FIN in Zone 6d. In Zone 7, de-asserting $ENABLE\_INn_{Chip1}$ takes Chip1 out of bypass mode. During the entire sequence in this example, Chip3 ignores input data while $ENABLE\_INn_{Chip3}$ is high, and simple passes $DATA\_IN_{Chip3}$ straight to $DATA\_OUT_{Chip3}$ (after some latency).

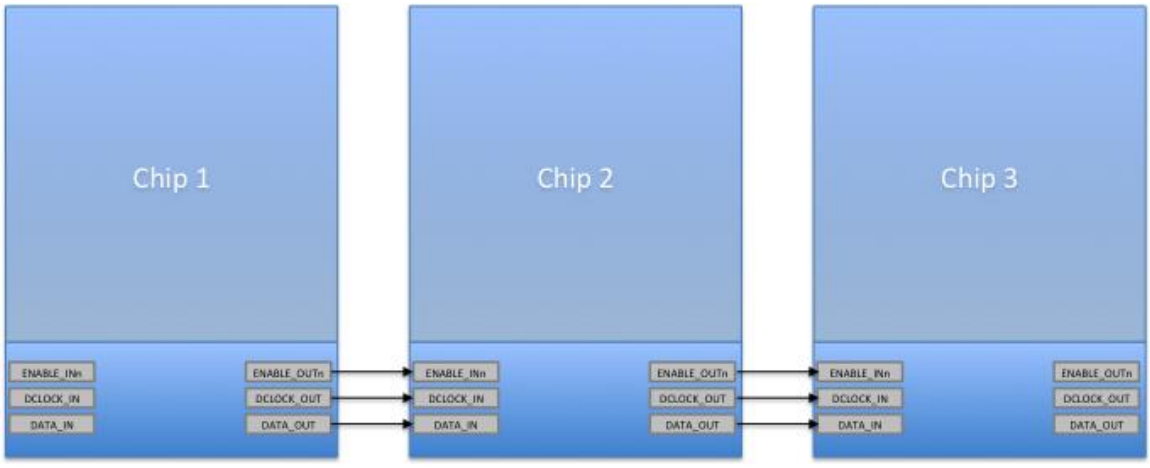

Figure 23. IO connections between daisy-chained chips

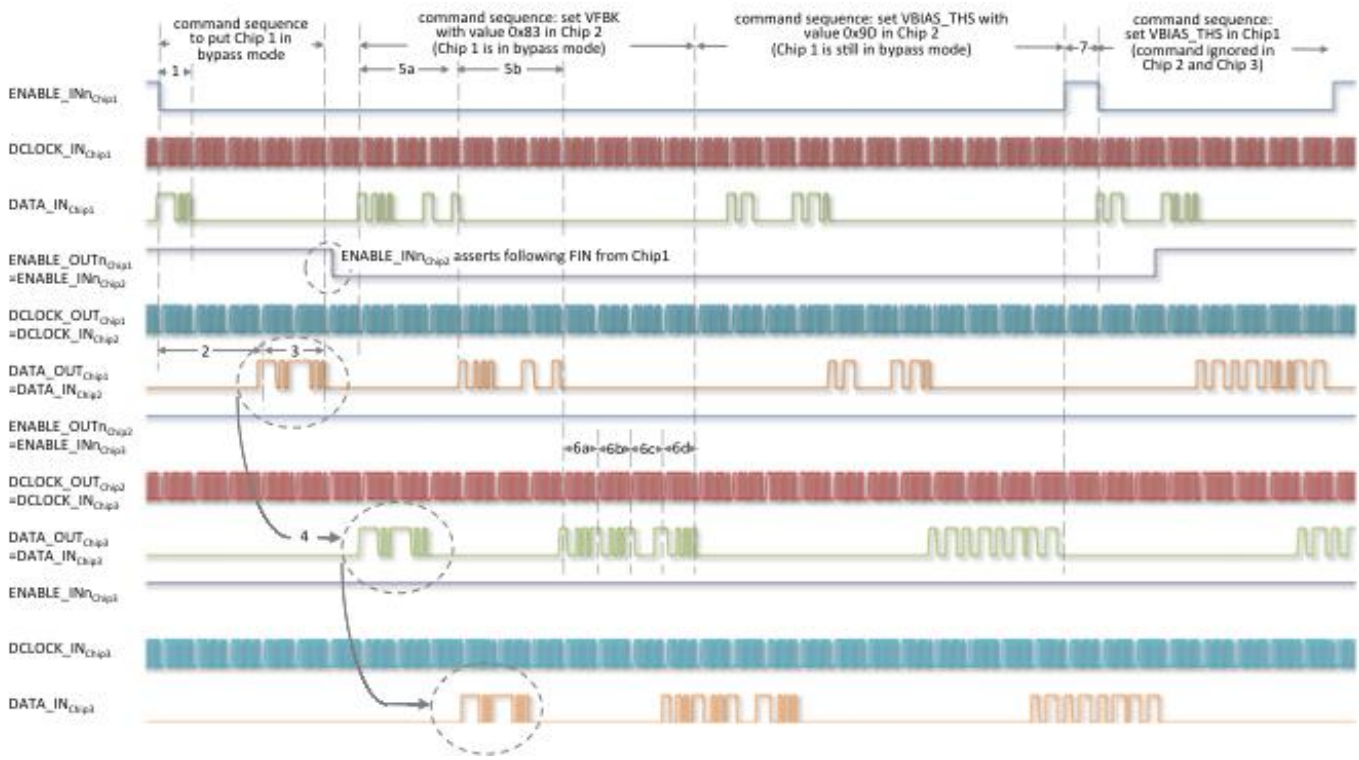

Figure 24. Example command sequence in three daisy-chained chips

Note: from verification, yield on chips that can be daisy-chained is expected to be lower than the yield on chips that as independent devices.

Framerates

Table 31 lists the time to read a frame from the Timepix2 chip, using the maximum data clock frequency of 100 MHz. In the simultaneous ToT and ToA operation mode, the minimum readout deadtime corresponds to t_{read} of 28 bits. In the continuous read/write modes, there is no readout deadtime, but the minimum counting period is defined by t_{read}. In ZCS mode, at least 16 columns must be read out, even if they are empty.

Table 31. Time to output frames in Timepix2, assuming 100 MHz clock

# bits/pixel	Full Frame, serial port		Full frame, parallel port		ZCS[5]	
	t_{read} [ms]	framerate [fps]	t_{read} [ms]	framerate [fps]	t_{read} [ms]	framerate [fps]
4	2.62	381	0.08	12207	0.16	6104
10	6.55	153	0.20	4883	0.41	2441
14	9.18	109	0.29	3488	0.57	1744
28	18.35	54	0.57	1744	1.15	872

[5] The readout time and framerate in ZCS mode depends on the number of columns that contain data. The values shown in Table 31 are based on 16 columns, which is the minimum number of columns that must be read out in ZCS mode.

Power on Sequence

There is no particular sequence that needs to be respected for VDD and VDDIO. However, for VDD33, VDD needs to be powered first, see Figure 25. (Powering on VDD and VDD33 simultaneously might result in burning random eFuses).

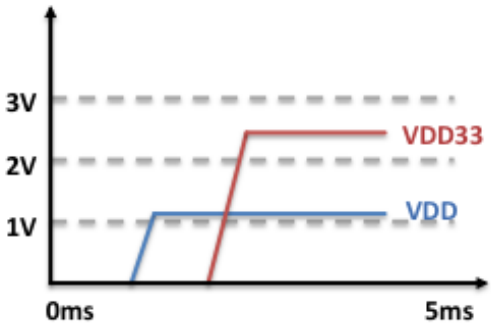

Figure 25. Power on sequence for VDD33

CERN Timepix2 Chipboard

Figure 26 shows the CERN chipboard, which can be cut to permit the tiling of Timepix2 detectors mounted on separate cards . It contains industry standard connectors, voltage regulators, a power measurement chip, multiple test points, and a backside pulse processing (BPP) circuit to correlate events in the pixel matrix with events detected through the backside pulse in the sensor. The BPP circuit comprises a charge sensitive preamplifier followed by a bandpass filter (see Figure 27). With the addition of a threshold discriminator to digitise the signal, the BPP output can be fed back to a digital pixel on the Timepix2 ASIC for ToT and ToA measurement. Since the BPP uses discrete components, the dynamic range can be customised for a particular application. For example, the choice of a CSA feedback capacitance of 1 pF would permit the charge integration of up to 100 MeV, while a 10 pF feedback capacitance would process up to 1 GeV in a silicon sensor.

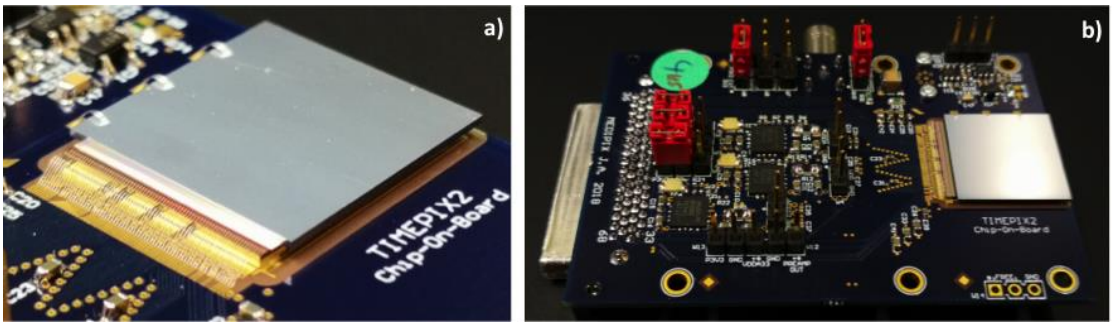

Figure 26. Timepix2 with silicon sensor mounted on the chipboard

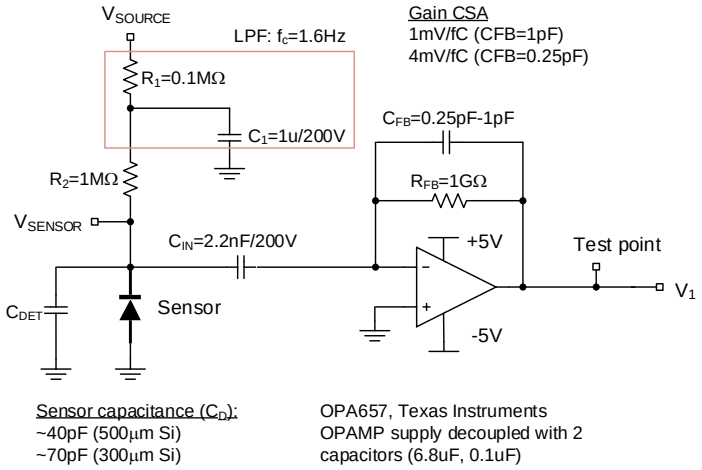

Figure 27. Schematic of the charge sensitive preamplifer for backside pulse processing on the Timepix2 chip-board

Works Cited

Gaspari, M. D. (2013). *Time-Over-Threshold monotonicity in TimePix3 and SmallPix + update on the TP3 analog design review.* Medipix3 Open Meetings.

Horswell, I. (2010). Medipix3 Counters.

Krummenacher, F. (1991). Pixel detectors with local intelligence: an IC designer point of view. *Nuclear Instruments and Methods in Physics Research A*, 527-532.

Llopart, X. (2006). *Timepix Manual v1.0.*

Llopart, X. (2007). *Design and Characterization of 64K Pixels Chips Working in Single Photon Processing Mode.* Sundsvall: Mid Sweden University Doctoral Thesis.

Wong, W. (2014). *Timepix Redesign Ideas.* Geneva: CERN.

Poikela, T and Llopart, X. (2016). *Velopix Operation Manual v0.12.*

Ballabriga, R. (2018). *Backside pulse processing circuitry*. Internal note.

Appendix 1: LUT for 10-bit LFSR

The 10-bit counter chains in the pixels are linear feedback shift registers with an XOR tap at the output of the 7th and 10th bits. Table 32 lists pseudo-random codes of the 10-bit counter. This LUT is also available for download.

Table 32. LUT for 10-bit LFSR

Decimal Value	LFSR Pseudo-Random Code	Decimal Value	LFSR Pseudo-Random Code	Decimal Value	LFSR Pseudo-Random Code	Decimal Value	LFSR Pseudo-Random Code
0	0000000000	256	0010000110	512	0100001000	768	1000100001
1	0000000001	257	0100001100	513	1000010000	769	0001000011
2	0000000010	258	1000011000	514	0000100001	770	0010000111
3	0000000100	259	0000110001	515	0001000010	771	0100001110
4	0000001000	260	0001100010	516	0010000101	772	1000011100
5	0000010000	261	0011000101	517	0100001010	773	0000111001
6	0000100000	262	0110001011	518	1000010100	774	0001110010
7	0001000000	263	1100010110	519	0000101001	775	0011100101
8	0010000001	264	1000101101	520	0001010010	776	0111001011
9	0100000010	265	0001011011	521	0010100101	777	1110010111
10	1000000100	266	0010110111	522	0101001010	778	1100101111
11	0000001001	267	0101101110	523	1010010101	779	1001011111
12	0000010010	268	1011011101	524	0100101011	780	0010111110
13	0000100100	269	0110111010	525	1001010110	781	0101111100
14	0001001000	270	1101110100	526	0010101100	782	1011111001
15	0010010001	271	1011101000	527	0101011000	783	0111110010
16	0100100010	272	0111010000	528	1010110001	784	1111100101
17	1001000100	273	1110100001	529	0101100011	785	1111001010
18	0010001000	274	1101000011	530	1011000111	786	1110010100
19	0100010000	275	1010000110	531	0110001110	787	1100101001
20	1000100000	276	0100001101	532	1100011100	788	1001010011
21	0001000001	277	1000011010	533	1000111001	789	0010100110
22	0010000011	278	0000110101	534	0001110011	790	0101001100
23	0100000110	279	0001101010	535	0011100111	791	1010011001
24	1000001100	280	0011010101	536	0111001111	792	0100110011
25	0000011001	281	0110101011	537	1110011111	793	1001100110
26	0000110010	282	1101010110	538	1100111111	794	0011001100
27	0001100100	283	1010101100	539	1001111111	795	0110011001
28	0011001001	284	0101011001	540	0011111110	796	1100110010
29	0110010011	285	1010110011	541	0111111101	797	1001100101
30	1100100110	286	0101100111	542	1111111011	798	0011001010
31	1001001101	287	1011001111	543	1111110110	799	0110010101
32	0010011010	288	0110011110	544	1111101100	800	1100101010
33	0100110100	289	1100111100	545	1111011000	801	1001010101
34	1001101000	290	1001111001	546	1110110000	802	0010101010
35	0011010000	291	0011110010	547	1101100001	803	0101010100
36	0110100001	292	0111100101	548	1011000010	804	1010101001
37	1101000010	293	1111001011	549	0110000100	805	0101010011
38	1010000100	294	1110010110	550	1100001000	806	1010100111
39	0100001001	295	1100101101	551	1000010001	807	0101001111
40	1000010010	296	1001011011	552	0000100011	808	1010011111
41	0000100101	297	0010110110	553	0001000110	809	0100111111
42	0001001010	298	0101101100	554	0010001101	810	1001111110
43	0010010101	299	1011011001	555	0100011010	811	0011111100
44	0100101010	300	0110110010	556	1000110100	812	0111111001
45	1001010100	301	1101100100	557	0001101001	813	1111110011

46	0010101000	302	1011001000	558	0011010011	814	1111100110
47	0101010000	303	0110010000	559	0110100111	815	1111001100
48	1010100001	304	1100100000	560	1101001110	816	1110011000
49	0101000011	305	1001000001	561	1010011100	817	1100110001
50	1010000111	306	0010000010	562	0100111001	818	1001100011
51	0100001111	307	0100000100	563	1001110010	819	0011000110
52	1000011110	308	1000001000	564	0011100100	820	0110001101
53	0000111101	309	0000010001	565	0111001001	821	1100011010
54	0001111010	310	0000100010	566	1110010011	822	1000110101
55	0011110101	311	0001000100	567	1100100111	823	0001101011
56	0111101011	312	0010001001	568	1001001111	824	0011010111
57	1111010111	313	0100010010	569	0010011110	825	0110101111
58	1110101110	314	1000100100	570	0100111100	826	1101011110
59	1101011101	315	0001001001	571	1001111000	827	1010111100
60	1010111010	316	0010010011	572	0011110000	828	0101111001
61	0101110101	317	0100100110	573	0111100001	829	1011110011
62	1011101011	318	1001001100	574	1111000011	830	0111100110
63	0111010110	319	0010011000	575	1110000110	831	1111001101
64	1110101101	320	0100110000	576	1100001101	832	1110011010
65	1101011011	321	1001100000	577	1000011011	833	1100110101
66	1010110110	322	0011000000	578	0000110111	834	1001101011
67	0101101101	323	0110000001	579	0001101110	835	0011010110
68	1011011011	324	1100000010	580	0011011101	836	0110101101
69	0110110110	325	1000000101	581	0110111011	837	1101011010
70	1101101100	326	0000001011	582	1101110110	838	1010110100
71	1011011000	327	0000010110	583	1011101100	839	0101101001
72	0110110000	328	0000101100	584	0111011000	840	1011010011
73	1101100000	329	0001011000	585	1110110001	841	0110100110
74	1011000000	330	0010110001	586	1101100011	842	1101001100
75	0110000000	331	0101100010	587	1011000110	843	1010011000
76	1100000000	332	1011000101	588	0110001100	844	0100110001
77	1000000001	333	0110001010	589	1100011000	845	1001100010
78	0000000011	334	1100010100	590	1000110001	846	0011000100
79	0000000110	335	1000101001	591	0001100011	847	0110001001
80	0000001100	336	0001010011	592	0011000111	848	1100010010
81	0000011000	337	0010100111	593	0110001111	849	1000100101
82	0000110000	338	0101001110	594	1100011110	850	0001001011
83	0001100000	339	1010011101	595	1000111101	851	0010010111
84	0011000001	340	0100111011	596	0001111011	852	0100101110
85	0110000011	341	1001110110	597	0011110111	853	1001011100
86	1100000110	342	0011101100	598	0111101111	854	0010111000
87	1000001101	343	0111011001	599	1111011111	855	0101110000
88	0000011011	344	1110110011	600	1110111110	856	1011100001
89	0000110110	345	1101100111	601	1101111101	857	0111000010
90	0001101100	346	1011001110	602	1011111010	858	1110000101
91	0011011001	347	0110011100	603	0111110100	859	1100001011
92	0110110011	348	1100111000	604	1111101001	860	1000010111
93	1101100110	349	1001110001	605	1111010010	861	0000101111
94	1011001100	350	0011100010	606	1110100100	862	0001011110
95	0110011000	351	0111000101	607	1101001001	863	0010111101
96	1100110000	352	1110001011	608	1010010010	864	0101111010
97	1001100001	353	1100010111	609	0100100101	865	1011110101
98	0011000010	354	1000101111	610	1001001010	866	0111101010
99	0110000101	355	0001011111	611	0010010100	867	1111010101
100	1100001010	356	0010111111	612	0100101000	868	1110101010
101	1000010101	357	0101111110	613	1001010000	869	1101010101
102	0000101011	358	1011111101	614	0010100000	870	1010101010
103	0001010110	359	0111111010	615	0101000000	871	0101010101

104	0010101101	360	1111110101	616	1010000001	872	1010101011
105	0101011010	361	1111101010	617	0100000011	873	0101010111
106	1010110101	362	1111010100	618	1000000110	874	1010101111
107	0101101011	363	1110101000	619	0000001101	875	0101011111
108	1011010111	364	1101010001	620	0000011010	876	1010111111
109	0110101110	365	1010100010	621	0000110100	877	0101111111
110	1101011100	366	0101000101	622	0001101000	878	1011111111
111	1010111000	367	1010001011	623	0011010001	879	0111111110
112	0101110001	368	0100010111	624	0110100011	880	1111111101
113	1011100011	369	1000101110	625	1101000110	881	1111111010
114	0111000110	370	0001011101	626	1010001100	882	1111110100
115	1110001101	371	0010111011	627	0100011001	883	1111101000
116	1100011011	372	0101110110	628	1000110010	884	1111010000
117	1000110111	373	1011101101	629	0001100101	885	1110100000
118	0001101111	374	0111011010	630	0011001011	886	1101000001
119	0011011111	375	1110110101	631	0110010111	887	1010000010
120	0110111111	376	1101101011	632	1100101110	888	0100000101
121	1101111110	377	1011010110	633	1001011101	889	1000001010
122	1011111100	378	0110101100	634	0010111010	890	0000010101
123	0111111000	379	1101011000	635	0101110100	891	0000101010
124	1111110001	380	1010110000	636	1011101001	892	0001010100
125	1111100010	381	0101100001	637	0111010010	893	0010101001
126	1111000100	382	1011000011	638	1110100101	894	0101010010
127	1110001000	383	0110000110	639	1101001011	895	1010100101
128	1100010001	384	1100001100	640	1010010110	896	0101001011
129	1000100011	385	1000011001	641	0100101101	897	1010010111
130	0001000111	386	0000110011	642	1001011010	898	0100101111
131	0010001111	387	0001100110	643	0010110100	899	1001011110
132	0100011110	388	0011001101	644	0101101000	900	0010111100
133	1000111100	389	0110011011	645	1011010001	901	0101111000
134	0001111001	390	1100110110	646	0110100010	902	1011110001
135	0011110011	391	1001101101	647	1101000100	903	0111100010
136	0111100111	392	0011011010	648	1010001000	904	1111000101
137	1111001111	393	0110110101	649	0100010001	905	1110001010
138	1110011110	394	1101101010	650	1000100010	906	1100010101
139	1100111101	395	1011010100	651	0001000101	907	1000101011
140	1001111011	396	0110101000	652	0010001011	908	0001010111
141	0011110110	397	1101010000	653	0100010110	909	0010101111
142	0111101101	398	1010100000	654	1000101100	910	0101011110
143	1111011011	399	0101000001	655	0001011001	911	1010111101
144	1110110110	400	1010000011	656	0010110011	912	0101111011
145	1101101101	401	0100000111	657	0101100110	913	1011110111
146	1011011010	402	1000001110	658	1011001101	914	0111101110
147	0110110100	403	0000011101	659	0110011010	915	1111011101
148	1101101000	404	0000111010	660	1100110100	916	1110111010
149	1011010000	405	0001110100	661	1001101001	917	1101110101
150	0110100000	406	0011101001	662	0011010010	918	1011101010
151	1101000000	407	0111010011	663	0110100101	919	0111010100
152	1010000000	408	1110100111	664	1101001010	920	1110101001
153	0100000001	409	1101001111	665	1010010100	921	1101010011
154	1000000010	410	1010011110	666	0100101001	922	1010100110
155	0000000101	411	0100111101	667	1001010010	923	0101001101
156	0000001010	412	1001111010	668	0010100100	924	1010011011
157	0000010100	413	0011110100	669	0101001000	925	0100110111
158	0000101000	414	0111101001	670	1010010001	926	1001101110
159	0001010000	415	1111010011	671	0100100011	927	0011011100
160	0010100001	416	1110100110	672	1001000110	928	0110111001
161	0101000010	417	1101001101	673	0010001100	929	1101110010

162	1010000101	418	1010011010	674	0100011000	930	1011100100
163	0100001011	419	0100110101	675	1000110000	931	0111001000
164	1000010110	420	1001101010	676	0001100001	932	1110010001
165	0000101101	421	0011010100	677	0011000011	933	1100100011
166	0001011010	422	0110101001	678	0110000111	934	1001000111
167	0010110101	423	1101010010	679	1100001110	935	0010001110
168	0101101010	424	1010100100	680	1000011101	936	0100011100
169	1011010101	425	0101001001	681	0000111011	937	1000111000
170	0110101010	426	1010010011	682	0001110110	938	0001110001
171	1101010100	427	0100100111	683	0011101101	939	0011100011
172	1010101000	428	1001001110	684	0111011011	940	0111000111
173	0101010001	429	0010011100	685	1110110111	941	1110001111
174	1010100011	430	0100111000	686	1101101111	942	1100011111
175	0101000111	431	1001110000	687	1011011110	943	1000111111
176	1010001111	432	0011100000	688	0110111100	944	0001111111
177	0100011111	433	0111000001	689	1101111000	945	0011111111
178	1000111110	434	1110000011	690	1011110000	946	0111111111
179	0001111101	435	1100000111	691	0111100000	947	1111111111
180	0011111011	436	1000001111	692	1111000001	948	1111111110
181	0111110111	437	0000011111	693	1110000010	949	1111111100
182	1111101111	438	0000111110	694	1100000101	950	1111111000
183	1111011110	439	0001111100	695	1000001011	951	1111110000
184	1110111100	440	0011111001	696	0000010111	952	1111100000
185	1101111001	441	0111110011	697	0000101110	953	1111000000
186	1011110010	442	1111100111	698	0001011100	954	1110000000
187	0111100100	443	1111001110	699	0010111001	955	1100000001
188	1111001001	444	1110011100	700	0101110010	956	1000000011
189	1110010010	445	1100111001	701	1011100101	957	0000000111
190	1100100101	446	1001110011	702	0111001010	958	0000001110
191	1001001011	447	0011100110	703	1110010101	959	0000011100
192	0010010110	448	0111001101	704	1100101011	960	0000111000
193	0100101100	449	1110011011	705	1001010111	961	0001110000
194	1001011000	450	1100110111	706	0010101110	962	0011100001
195	0010110000	451	1001101111	707	0101011100	963	0111000011
196	0101100000	452	0011011110	708	1010111001	964	1110000111
197	1011000001	453	0110111101	709	0101110011	965	1100001111
198	0110000010	454	1101111010	710	1011100111	966	1000011111
199	1100000100	455	1011110100	711	0111001110	967	0000111111
200	1000001001	456	0111101000	712	1110011101	968	0001111110
201	0000010011	457	1111010001	713	1100111011	969	0011111101
202	0000100110	458	1110100010	714	1001110111	970	0111111011
203	0001001100	459	1101000101	715	0011101110	971	1111110111
204	0010011001	460	1010001010	716	0111011101	972	1111101110
205	0100110010	461	0100010101	717	1110111011	973	1111011100
206	1001100100	462	1000101010	718	1101110111	974	1110111000
207	0011001000	463	0001010101	719	1011101110	975	1101110001
208	0110010001	464	0010101011	720	0111011100	976	1011100010
209	1100100010	465	0101010110	721	1110111001	977	0111000100
210	1001000101	466	1010101101	722	1101110011	978	1110001001
211	0010001010	467	0101011011	723	1011100110	979	1100010011
212	0100010100	468	1010110111	724	0111001100	980	1000100111
213	1000101000	469	0101101111	725	1110011001	981	0001001111
214	0001010001	470	1011011111	726	1100110011	982	0010011111
215	0010100011	471	0110111110	727	1001100111	983	0100111110
216	0101000110	472	1101111100	728	0011001110	984	1001111100
217	1010001101	473	1011111000	729	0110011101	985	0011111000
218	0100011011	474	0111110000	730	1100111010	986	0111110001
219	1000110110	475	1111100001	731	1001110101	987	1111100011

220	0001101101	476	1111000010	732	0011101010	988	1111000110
221	0011011011	477	1110000100	733	0111010101	989	1110001100
222	0110110111	478	1100001001	734	1110101011	990	1100011001
223	1101101110	479	1000010011	735	1101010111	991	1000110011
224	1011011100	480	0000100111	736	1010101110	992	0001100111
225	0110111000	481	0001001110	737	0101011101	993	0011001111
226	1101110000	482	0010011101	738	1010111011	994	0110011111
227	1011100000	483	0100111010	739	0101110111	995	1100111110
228	0111000000	484	1001110100	740	1011101111	996	1001111101
229	1110000001	485	0011101000	741	0111011110	997	0011111010
230	1100000011	486	0111010001	742	1110111101	998	0111110101
231	1000000111	487	1110100011	743	1101111011	999	1111101011
232	0000001111	488	1101000111	744	1011110110	1000	1111010110
233	0000011110	489	1010001110	745	0111101100	1001	1110101100
234	0000111100	490	0100011101	746	1111011001	1002	1101011001
235	0001111000	491	1000111010	747	1110110010	1003	1010110010
236	0011110001	492	0001110101	748	1101100101	1004	0101100101
237	0111100011	493	0011101011	749	1011001010	1005	1011001011
238	1111000111	494	0111010111	750	0110010100	1006	0110010110
239	1110001110	495	1110101111	751	1100101000	1007	1100101100
240	1100011101	496	1101011111	752	1001010001	1008	1001011001
241	1000111011	497	1010111110	753	0010100010	1009	0010110010
242	0001110111	498	0101111101	754	0101000100	1010	0101100100
243	0011101111	499	1011111011	755	1010001001	1011	1011001001
244	0111011111	500	0111110110	756	0100010011	1012	0110010010
245	1110111111	501	1111101101	757	1000100110	1013	1100100100
246	1101111111	502	1111011010	758	0001001101	1014	1001001001
247	1011111110	503	1110110100	759	0010011011	1015	0010010010
248	0111111100	504	1101101001	760	0100110110	1016	0100100100
249	1111111001	505	1011010010	761	1001101100	1017	1001001000
250	1111110010	506	0110100100	762	0011011000	1018	0010010000
251	1111100100	507	1101001000	763	0110110001	1019	0100100000
252	1111001000	508	1010010000	764	1101100010	1020	1001000000
253	1110010000	509	0100100001	765	1011000100	1021	0010000000
254	1100100001	510	1001000010	766	0110001000	1022	0100000000
255	1001000011	511	0010000100	767	1100010000	1023	1000000000

Appendix 2: IO Pad Figures

The following figures are zoomed in versions of the wirebond diagram in Figure 9, from left to right.

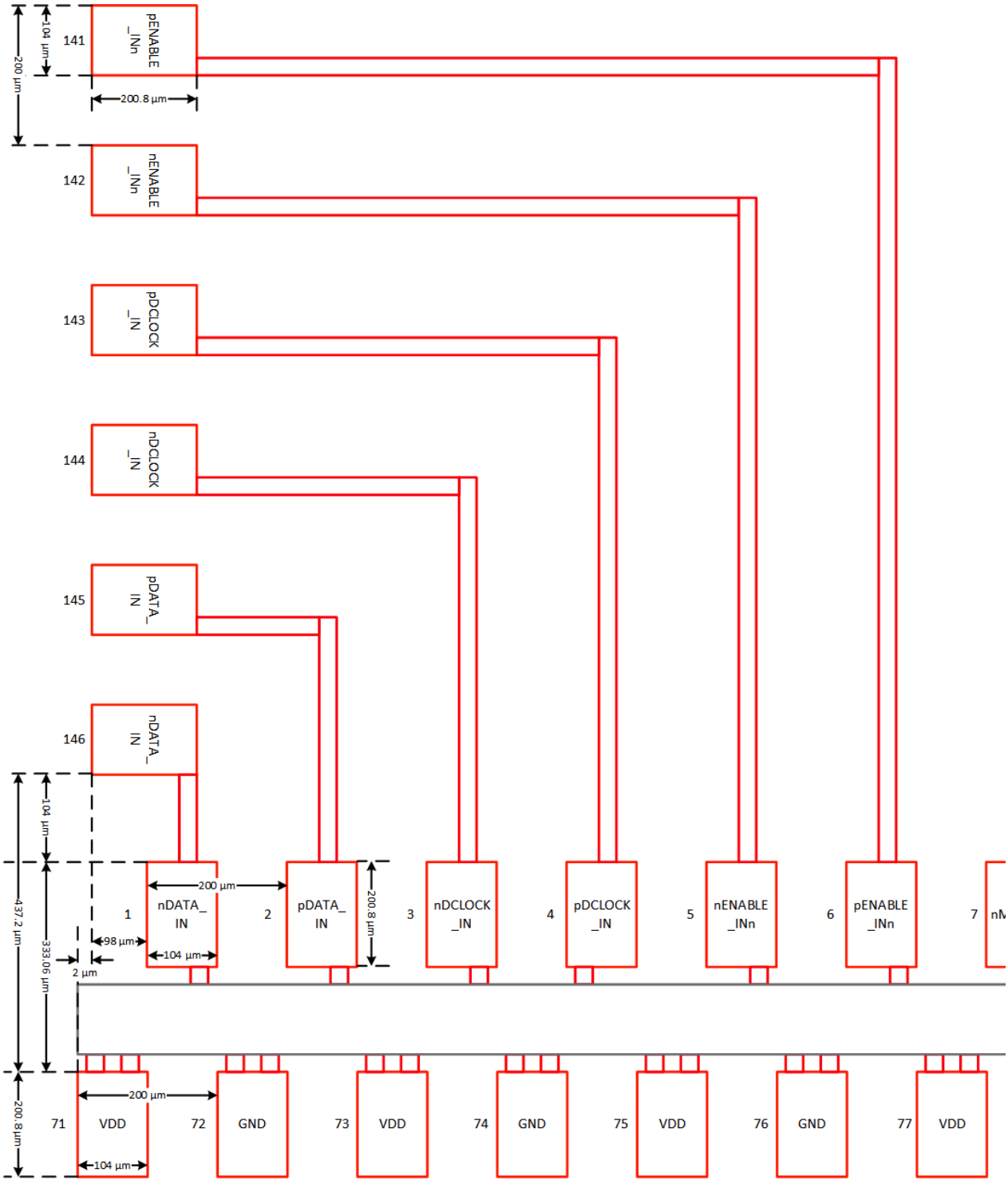

Figure 28. Timepix2 wirebond pad diagram (zoom 1)

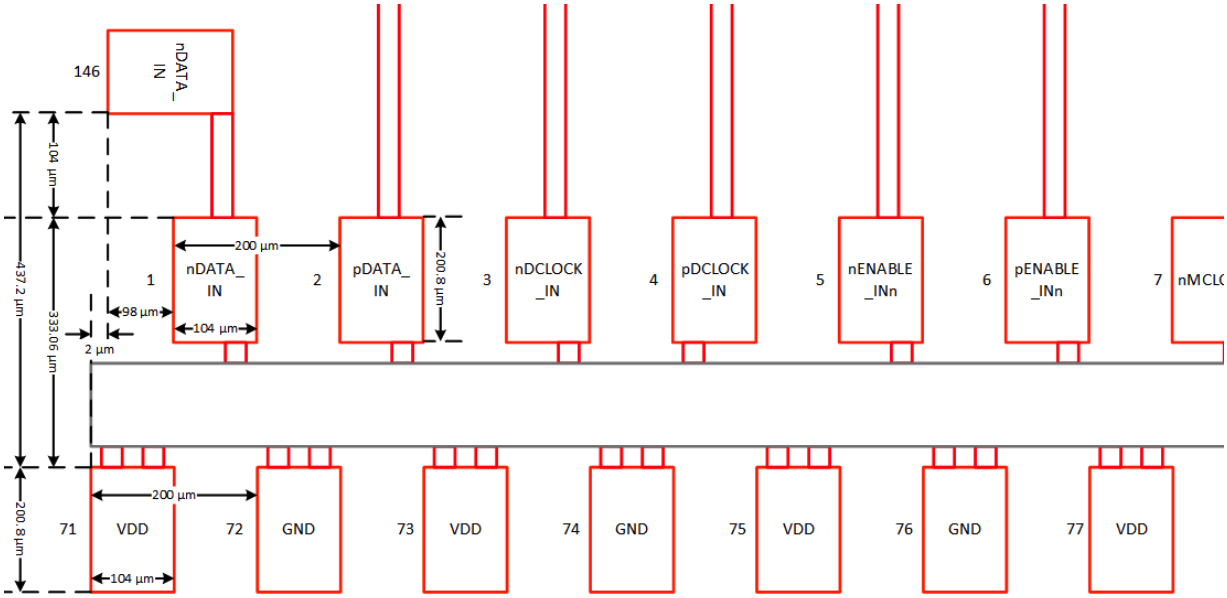

Figure 29. Timepix2 wirebond pad diagram (zoom 2)

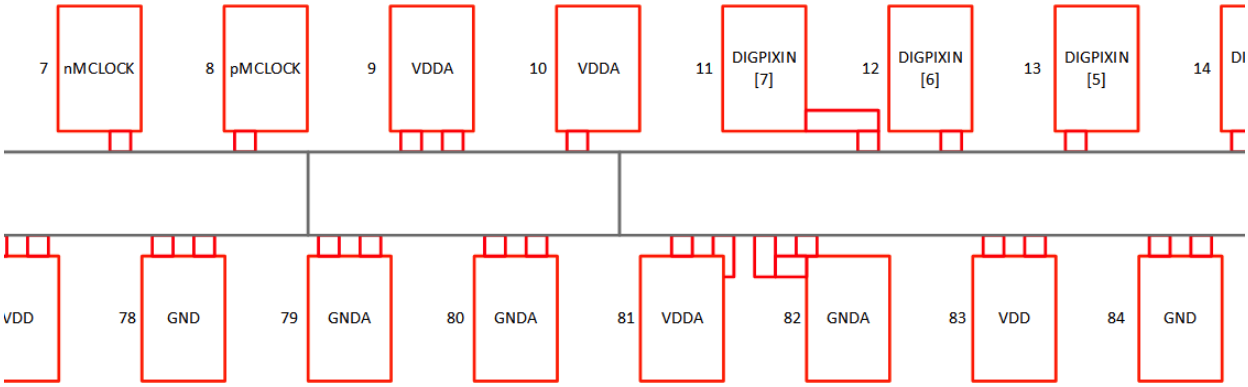

Figure 30. Timepix2 wirebond pad diagram (zoom 3)

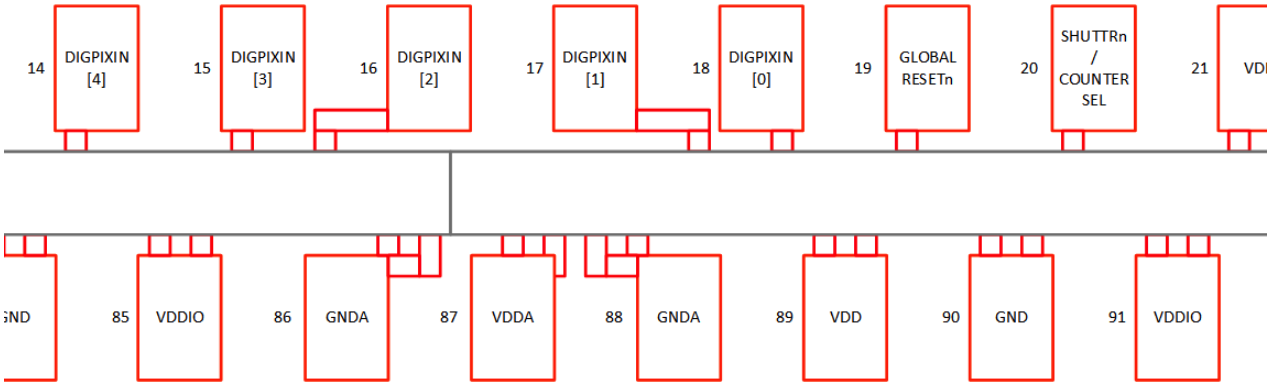

Figure 31. Timepix2 wirebond pad diagram (zoom 4)

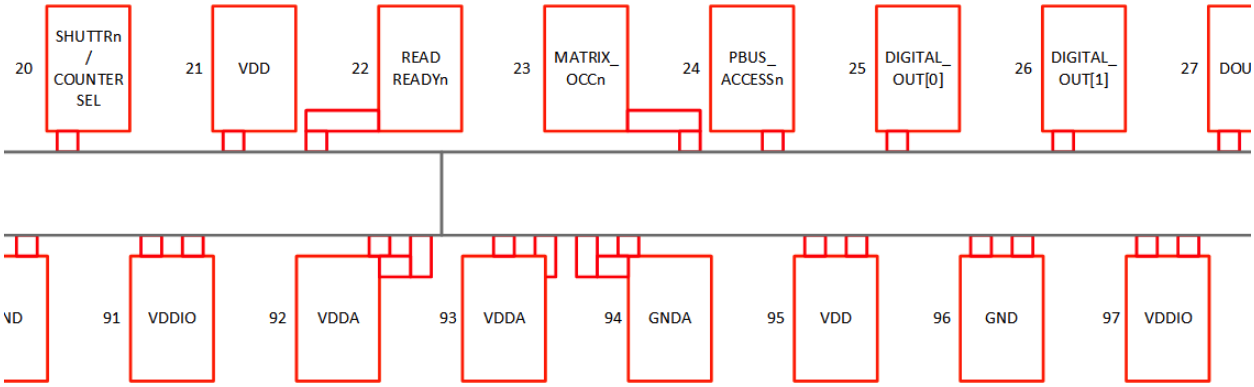

Figure 32. Timepix2 wirebond pad diagram (zoom 5)

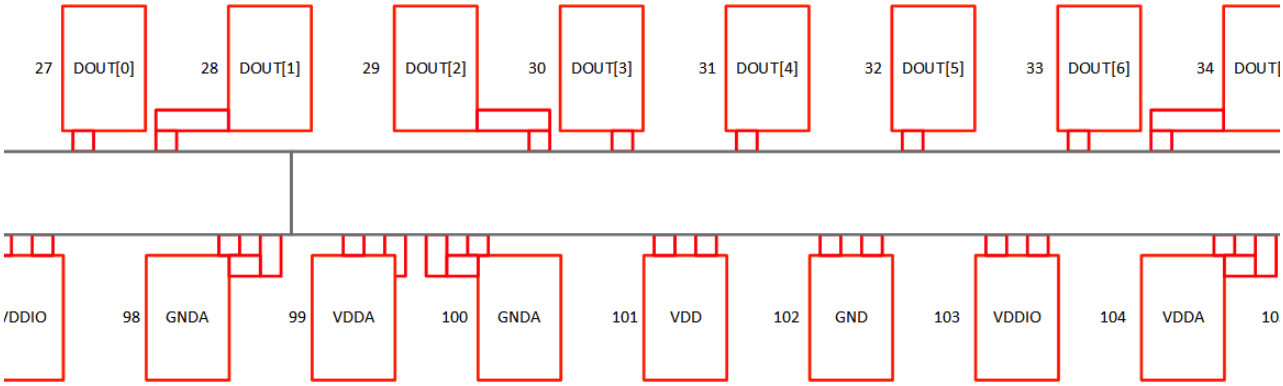

Figure 33. Timepix2 wirebond pad diagram (zoom 6)

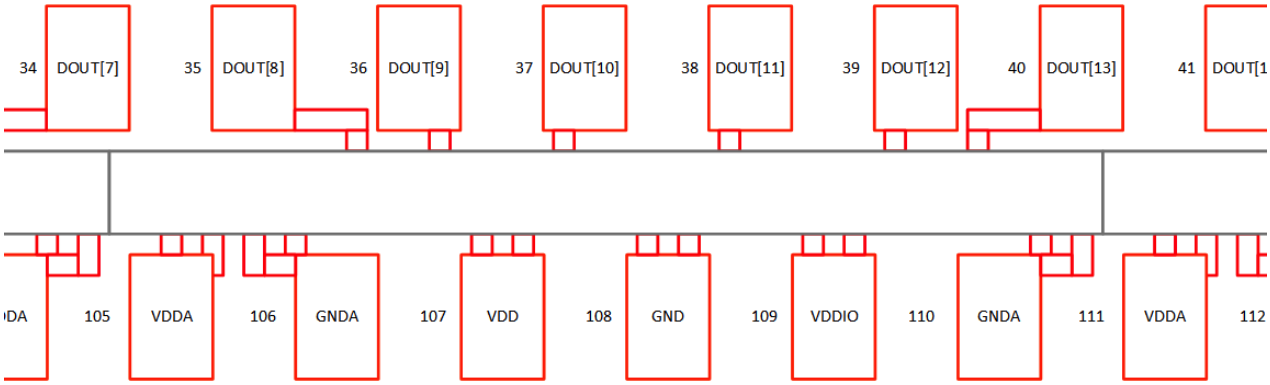

Figure 34. Timepix2 wirebond pad diagram (zoom 7)

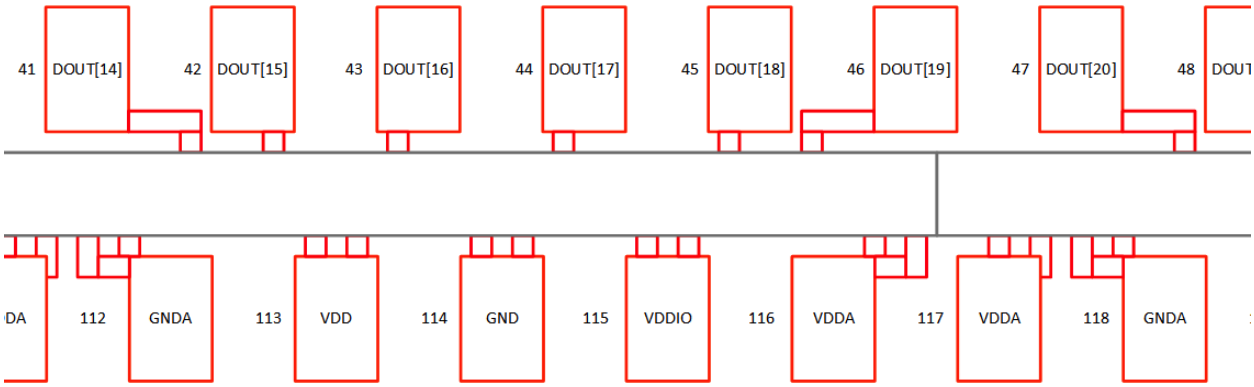

Figure 35. Timepix2 wirebond pad diagram (zoom 8)

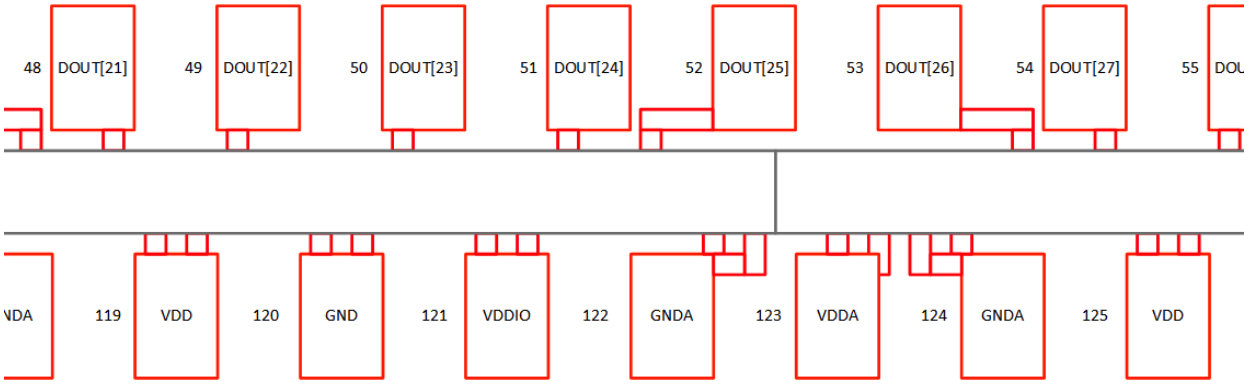

Figure 36. Timepix2 wirebond pad diagram (zoom 9)

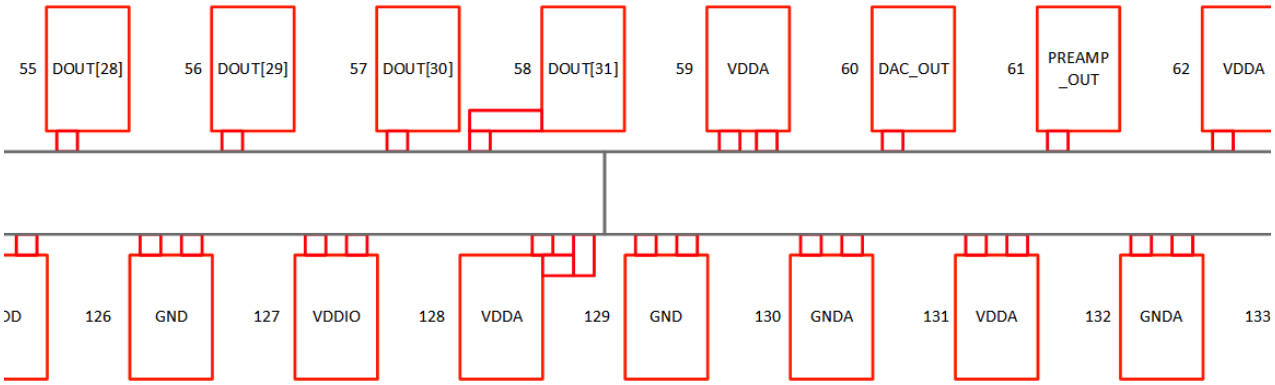

Figure 37. Timepix2 wirebond pad diagram (zoom 10)

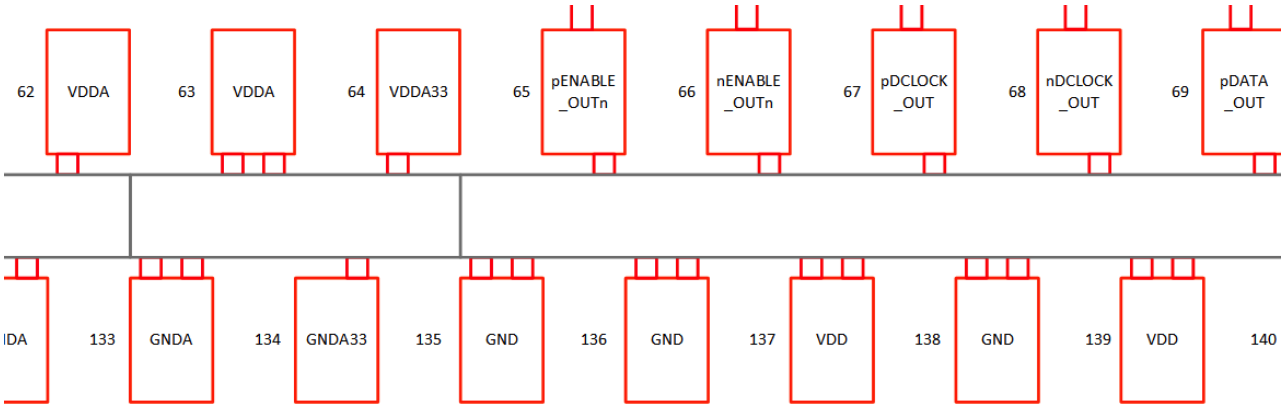

Figure 38. Timepix2 wirebond pad diagram (zoom 11)

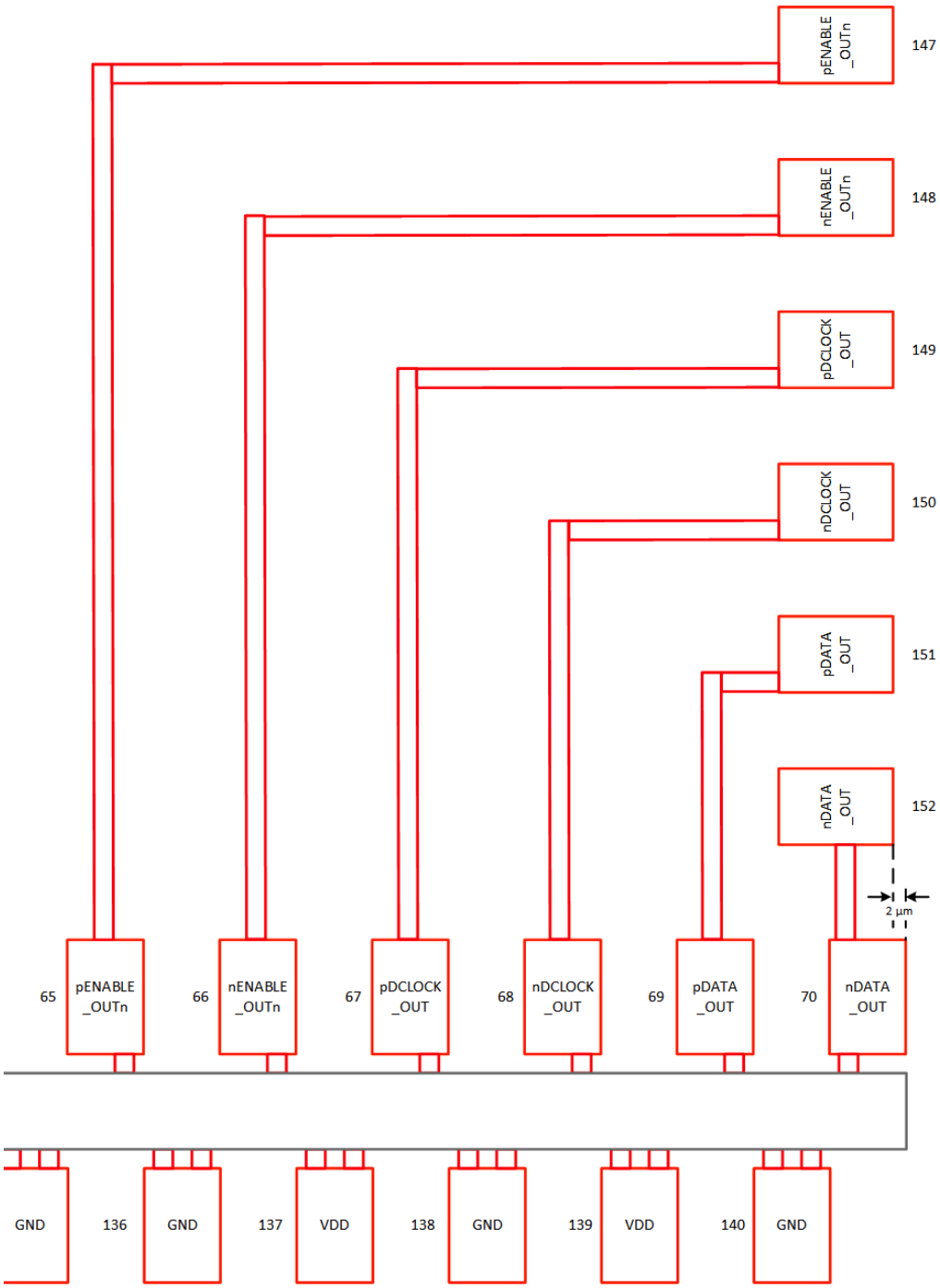

Figure 39. Timepix2 wirebond pad diagram (zoom 12)