TSI engineering run HVMAPS

LHCB/CLIC/TELEPIX Chips Description

Version 2 (Added figures 8 and 9 showing monitor- and digital pads)

Ivan Peric

Eight test chips have been designed and submitted on the TSI engineering run in May 2020.

All these chips have the same size, same bottom of column (hit buffers*, pixel registers, end of column buffers), same global periphery, pads. Just the pixels in the matrix are different.

There is space for 29 x 124 pixels (more precise pixel places or “slots”) of 25µm x 165µm size in the pixel matrix.

These pixel “slots” are sometimes occupied by single pixels (size: 25µm x 165µm) and sometimes large pixels (50µm x 165µm) or (100µm x 165µm) which then occupy 2 or 4 places.

There are 29 x 124 hit buffers below the matrix connected to the pixels. In the case of matrices with large pixels, some buffers are unused. *As mentioned, hit buffer structure is common to all chips, except for one detail the type of receiver, as can be seen in Figure 5.

The size of chip (x, y) is 5000µm x 4870µm. Block scheme of the chip-top is shown in Fig 1. Chip layout is in Fig 2.

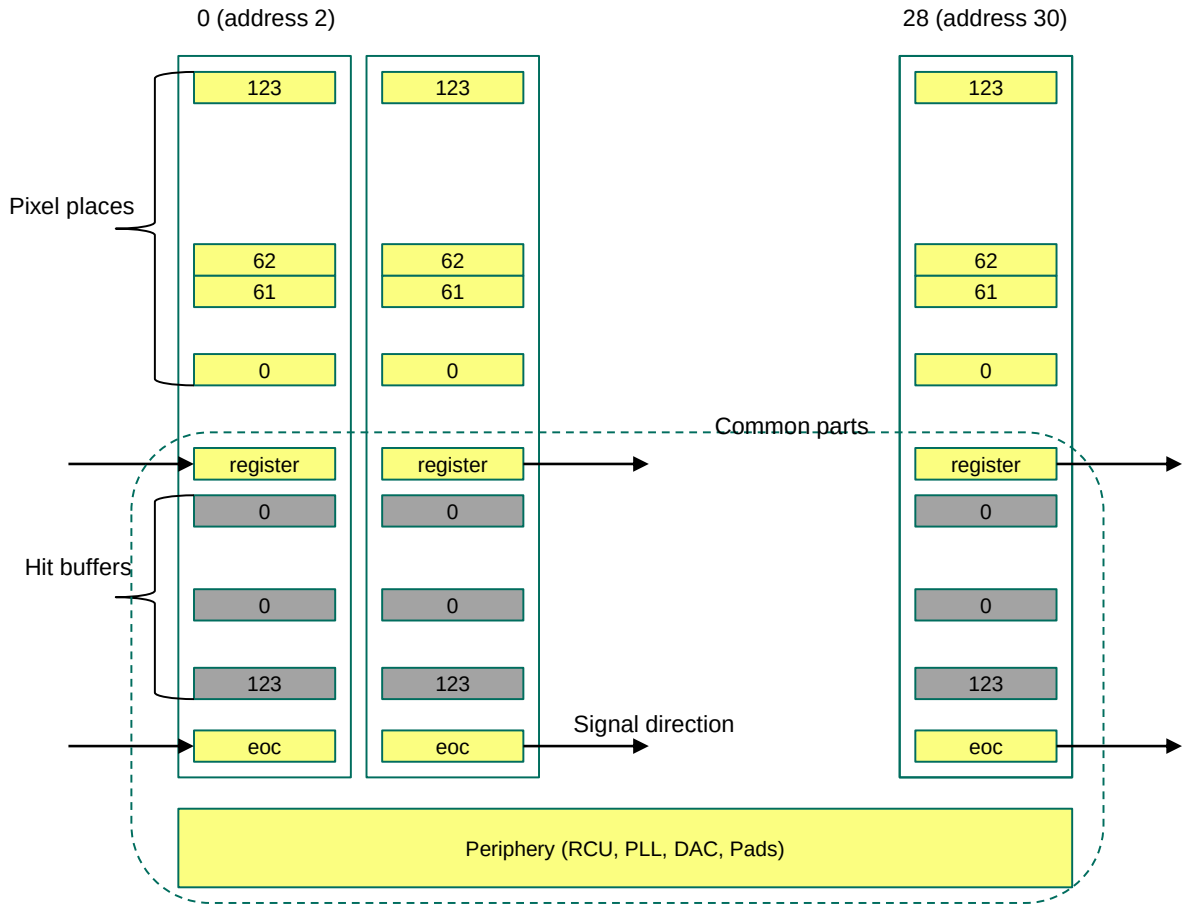

Figure 1: Block scheme of the chip-top

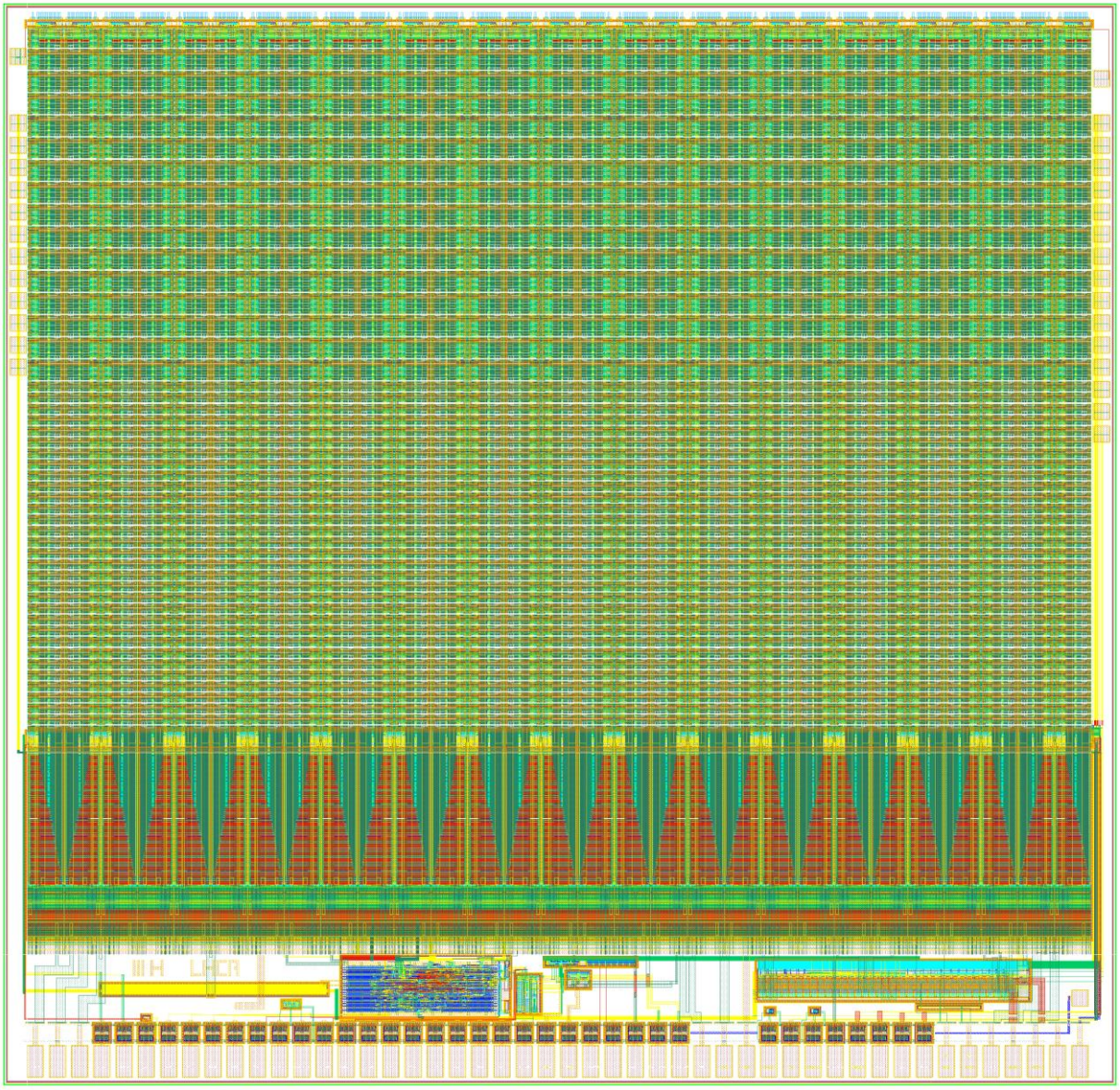

Figure 2: Chip layout

The chip has a pixel matrix organized in 29 columns, the column width is 165µm.

The chip has also a digital periphery block (RCU and PLL) and a bias DAC block.

One column contains 124 pixel places (Fig 1), occupied with pixels whose number varies depending on pixel size. It contains a pixel control register with 25 bits, 124 hit buffers and one End of Column (EoC) buffer.

The exact description of the matrix types is given at the end of the document.

We will just briefly mention the types.

1) Size types

“Standard column” or “variable column”

A standard column has 124 small pixels (pixel size 25µm x 165µm).

A variable column has 31 pixels of 50µm x 165µm size and 16 pixels of 100µm x 165µm size. 50µm pixels occupy slots 0, 2, 4 … 60 and the 100µm pixels occupy slots 64, 68, … 122.

2) Pixel comparator types

Pixel comparator can be of “version 1” or NMOS type, of “version 2” or CMOS type and of “version 3” or distributed type.

3) Amplifier types

Amplifier can be of CMOS, NMOS or PMOS type according to input transistor.

Amplifier can be also the “standard” one (has a standard set of circuit parameters), the low power type (parameters are optimized for low power) and a special one (or variable) with one circuit parameter different from the standard, for instance a circular cascode transistor.

Pixel structure is common for all types and is shown in Fig 3.

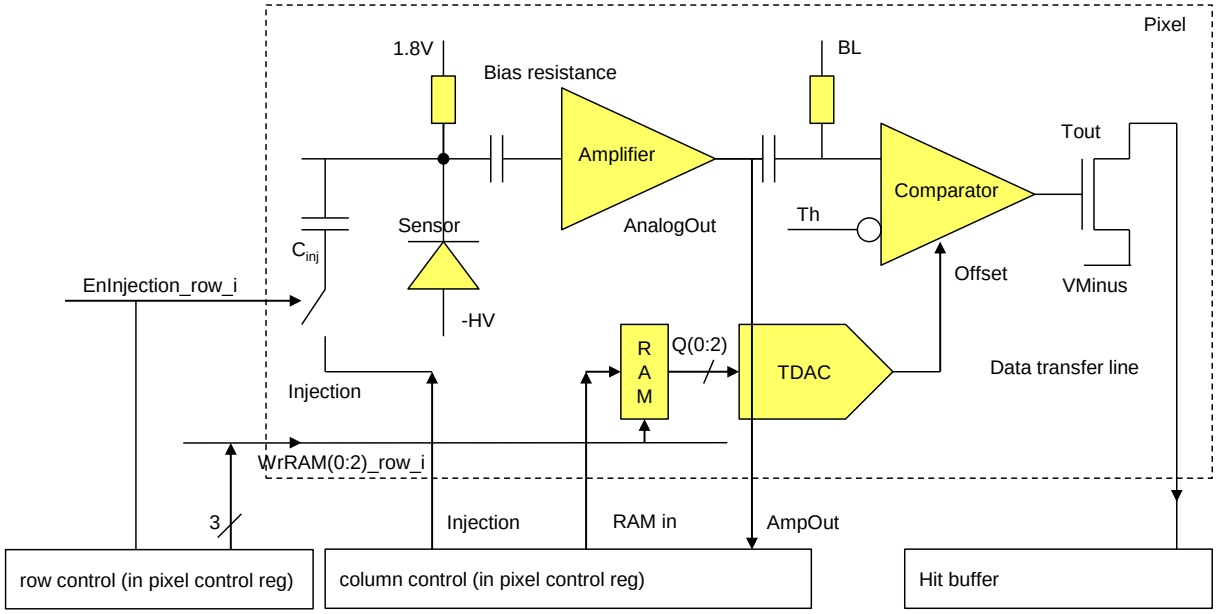

Figure 3: Pixel structure

Pixel contains a pixel n-well (sensor electrode) with following circuits inside it: charge sensitive amplifier, comparator, output transistor Tout, tune DAC (TDAC) (not in every pixel type), injection switch and capacitor C_{inj}. Writing into TDAC RAM and enabling of injection is controlled by setting bits in the pixel control register. This register is placed between the pixel column and the hit buffer column.

Figure 4 shows transistor level schematic of the PMOS amplifier, with sensor bias, feedback and CR filter.

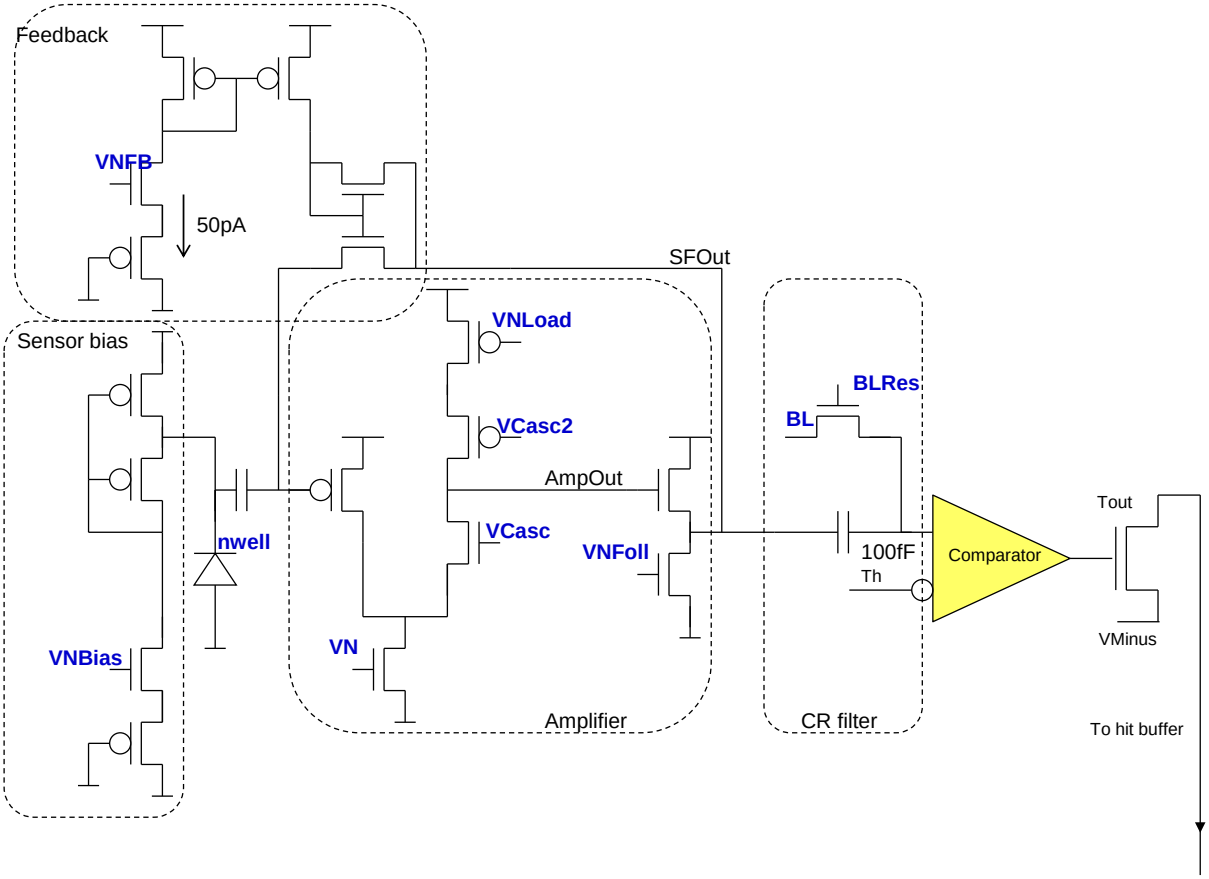

Figure 4: Transistor level schematic of the amplifier

Figure 5 shows the simplified block scheme of one column and the readout control unit (RCU) that is placed below the pixel matrix.

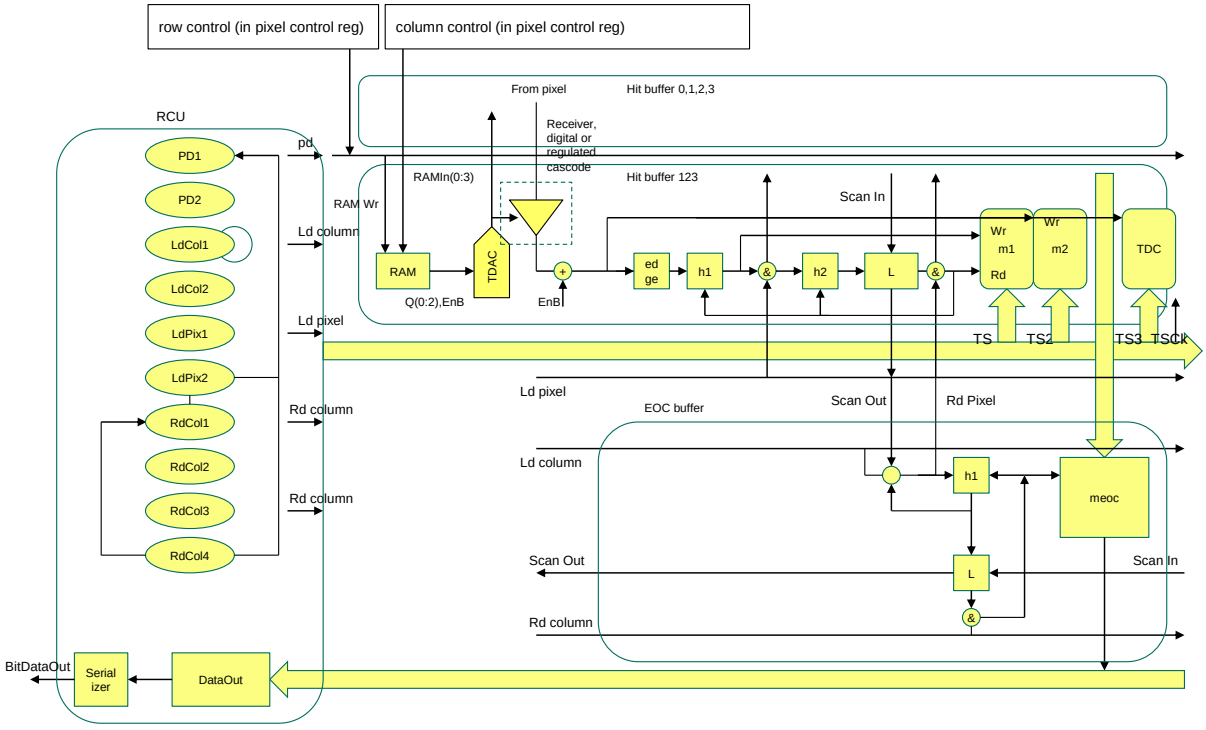

Figure 5: Simplified block scheme of one column, hit buffer, EoC-buffer and the readout control unit (RCU)

Block scheme of the hit buffer and end of column buffer (EoC) as well as the simplified state diagram of the state machine in the RCU can be seen.

The state machine receives scan out signal (called also priority out) from EoC buffers. It generates pull down (pd), Ld column and Rd column signals.

The readout works as follows:

The particle signal causes a pulse at the output of the pixel amplifier. The feedback capacitor of the amplifier (p+ diffusion to n-well capacitance of the node AmpOut in Fig 4) is continuously discharged by the feedback circuit (Fig 4). The amplifier output SFOut is fed to the comparator in the pixel. The output of the comparator is a pulse signal whose rising edge is coincident with the particle hit and whose width is proportional to the collected charge.

The hit buffer contains: RAM to store three tune bits and one enable B (active low) bit, tune DAC, digital receiver (for version 1 and 2 comparators) or regulated cascode receiver (for version 3 comparator), edge detector, two hit flags (h1 and h2), priority logic (L), memory m1 for leading edge time stamp TS (20 bits), memory m2 for falling edge time stamp TS2 (10 bits), memory m3 as part of TDC for 7 bit TS3. Address memory (ROM) which is not shown in the figure. The time stamps are Grey encoded. Time stamp TS has two 10bit blocks that count with the same codes. The 10 lower bits change on rising TSCk edge, the 10 higher bits change on falling TSCk edge.

The comparator output pulse sets the first hit flag. The second hit flag is set synchronously with the signal "Ld pixel" that is generated periodically by the state machine. The priority logic will select the first cell that contains a hit in the column. The state machine initiates the readout by issuing a signal "Ld column". Only the selected cell will process the Ld column signal issued by the state machine and transmit the stored hit data to the memory meoc of the End of Column buffer. The falling edge of Ld column deletes the hit flag which was previously loaded. The hit words are transferred from the EoC buffers to the RCU by the signal "Rd column" in the similar way.

TDC work in the following principle: after leading hit edge, analogue ramp signal starts from 0V. After rising TCk edge, the ramp slope is reduced. The time stamp TS3 is stored when the ramp signal exceeds a threshold of 1.4V.

Data word for one hit has 64-bits, which include 37 bits of time stamps (TS, TS2 and TS3), 10 bits of hit buffer row address (3 MS bits are unused) and 5 bits of column address. The readout control unit reads the hits selected for readout, formats the hit information and sends it out of the chip using a 1.6Gbps serial protocol. The RCU is also responsible for generating of time stamps and TCCk.

The following text block describes RCU in a simplified way, for more detailed description see the section Digital Periphery.

The main functional blocks of the RCU are the readout block, data formatting block and the time stamp counters. The readout block comprises a state machine that generates the signals for the readout of hit information from the readout cells. The data formatting block is a complex logic that receives the hit data from the state machine, performs 8-bit to 10-bit conversion and transmits the data serially. The serializer block contains a custom 8b/10b encoder and a serializer structure based on a binary tree. The chip contains a phase locked loop (PLL) with an input clock of maximally 160MHz (clk4n) and an output clock - called clk_800p - of maximally 800MHz. The serializer transmits a bit on the rising and the falling output clock edge, which leads to a maximal data rate of 1.6Gbps. Assuming this rate, the transmission of one hit takes 50ns. The time stamp counter clock frequency is programmable, it can be clk_4n or clk_800p.

More details about RCU are in the section Digital Periphery.

Digital periphery

The digital periphery has two blocks, one is the synthesized digital part - the readout control unit RCU, and the other is a full custom part with the PLL and an analogue 2->1 serializer.

RCU is generated from code. The full custom part uses differential current mode logic (DCL) instead of CMOS logic.

Readout control unit

As mentioned above, RCU contains following blocks (some explanations are redundant with the explanations above)

1) Configuration register
2) Time stamp counter
3) Readout block
4) One serializer-top block
5) SPI with “FIFO”

Inputs and outputs

RCU receives the signal PrioFromDet (called also scan out) and the data from the EoC

TSFromDet(19:0) is the time stamp of the rising signals edge. (TS in Fig 5)

TS2FromDet(9:0) is the time stamp of the falling signal edge. (TS2 in Fig 5)

TS3FromDet(6:0) is the “fine” time stamp coming from TDC. (TS3 in Fig 5)

ColAddFromDet(4:0) and RowAddFromDet(9:0) are the column- and row addresses.

RCU generates the signals PullDN, LdCol, LdPix, RdCol. (pd, Ld column, Rd column, Ld pixel in Fig 5)

RCU generates the following Grey coded time stamps:

TsToDet(9:0) that changes with rising edge of TSCk. (TS in Fig 5)

TsToDet(19:10) that changes with falling edge of TSCk. (TS in Fig 5)

TSToDet2(9:0) clock for the falling edge measurement. (TS2 in Fig 5)

TSToDet3(6:0) clock for TDC. (TS3 in Fig 5)

RCU also generates two bit data output BitDataOut(1:0).

The main clock input is clk_800p – BitDataOut changes at rising edge of this clock.

Clock output is clk_4n, it is 5x slower than clk_800p and used for PLL.

There is also separated clock for the time stamp counter TSCk.

Res_n is the asynchronous reset for serializer and the readout block.

SyncRes resets the time stamp counters.

Configuration register signals are:

Ck1, Ck2, SIn, SOut, Ld and Rb

SPI signals will be explained in next version of the document.

The clock names: clk_800p, clk_4n should indicate the period of the clocks. However, the indicated periods are by factor 0.64 smaller than can be achieved in reality. For example the real clock period of clk_800p cannot be smaller than 1.25ns (frequency = 800MHz).

Notice that there is one SyncRes signal. This signal should be generated externally. It should be generated in correct moment and synchronized with TSCk. SyncRes resets the time stamp counters.

State machine

Main part of RCU is the state machine, shown in Fig 6 with more details as in Fig 5. It receives the signal *ScanOut* and the data from the EoC buffers and it generates the signals PullDN, LdCol, LdPix, RdCol.

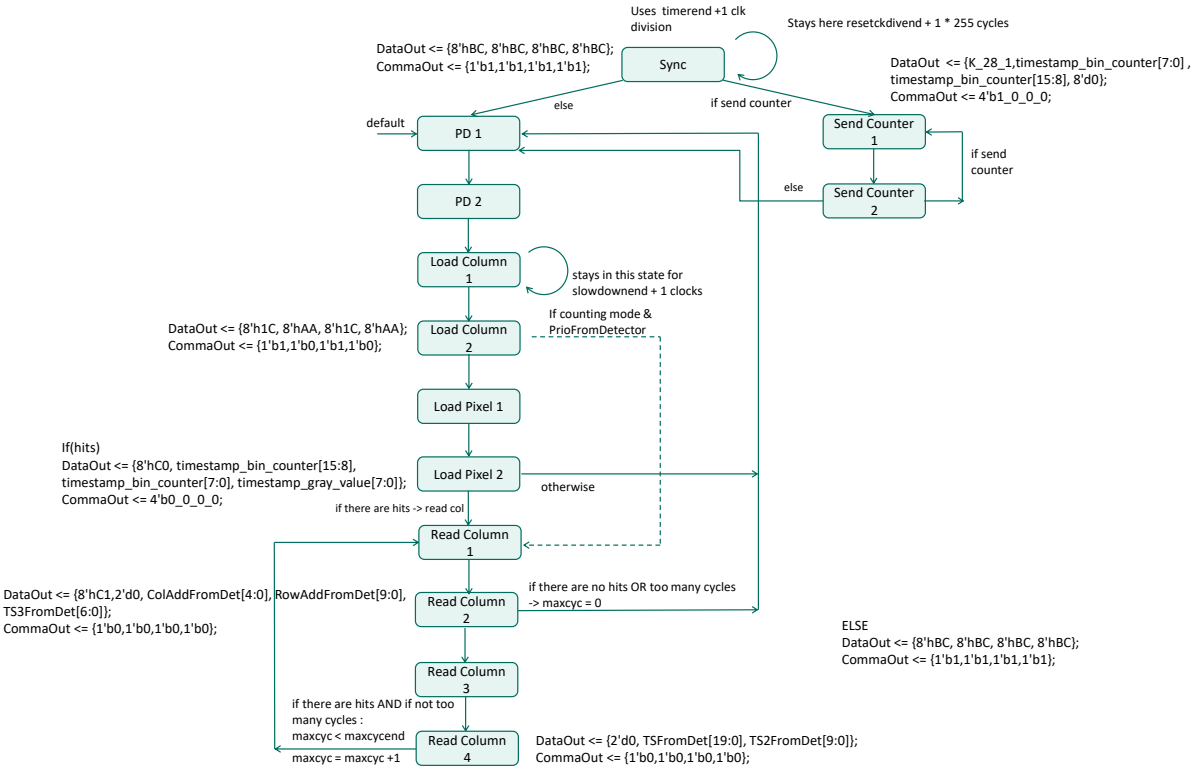

Figure 6: State machine

The state machine controls the readout from the pixel matrix, reads out the data and passes (transfers) the data to the serializer.

This works in the following way:

After reset, the state machine enters Sync state, it stays there some programmable clock periods and then enters PD1 state. In this state the output “pull down” is active. This signal pulls down (discharges) the column data busses. (See also Fig 5) In state PD2, the signal “pull down” gets inactive. Generally, states with index 2 are breaks between the control signals.

In state load column 1, the corresponding signal LdCol is generated. This signal is transmitted to the hit buffers and the buffer that has “active” data (the second hit butter set to 1) and that has the highest position puts its data (hit address, timestamps) at the column data bus. The data are copied into the end of column buffers (EoC-buffers). Duration of load column can be programmed by setting “slowdownend” bits. Value slowdownend = 7 is recommended. Load column 2 is the next state. There

is one control bit “counting mode” that influences the state machine. Standard setting is 0, in this case after Load column 2, Load Pixel1 follows.

For counting mode = 1 the state machine works in the way that new hits are only accepted after all EoC cells with the old hits are readout. In this case, when signal PrioFromDetector signalizes that there are still hits in the buffers, the state machine goes to Read column 1.

We will for the moment assume counting mode = 0. The next state is then Load Pixel 1 and it generates signal LdPix. The signal LdPix activates the hit buffers with stored hits (Fig 5). After the state Load Pixel 2, there are two possibilities. If there are hits in the EoC buffers, the next state is Read Column 1. If there are no hits in EoCs, the next state is PD1. In Read Column1, the corresponding signal RdCol is generated. This signal activates the first EoC-cell with data, it sends the hit data via the bus. The data are passed to the serializer in the states Read column 2 and 4.

The content of the register DataOut (size: four 8-bit words) will be converted to four 10-bit words (8bit/10bit conversion) and serialized starting from MSB. Four bits CommaOut(3:0) decide whether normal 8bit/10bit conversion is done (CommaOut bit 0) or a comma word is generated (CommaOut bit 1).

State machine works with “clk_8ns” clock that is 10x faster than the bit clock “clk_800p”. Serializer sends one bit at high and one bit at low level of clk_800p. This means, the serializer can transmit 20 bits for one clk_8n period. State machine transitions occur on rising clk_8ns clock when the timer counter equals “timerend” value. This timerend value is programmable, it is stored in configuration register. (Time counter starts from 0, counts to timerend and back to 0 at every clk_8ns edge.) This means, the speed of state machine transitions can be reduced by setting timerend > 0. For the fastest speed (timerend = 0), the state machine period is equal to the period of clk_8ns. Since state machine transfers the data to the serializer in every second state, the serializer have always time to transmit all 40 bits. There is no FIFO between the state machine and the serializer.

Now we will write which data are sent to the serialiezer in each state. In the case that we have timerend > 0, the data are transferred at the final clock period of the mentioned state.

Sync

```
DataOut <= {8'hBC, 8'hBC, 8'hBC, 8'hBC};

CommaOut <= {1'b1,1'b1,1'b1,1'b1};
```

Send counter 1

```
DataOut  <= {K_28_1,timestamp_bin_counter[7:0] , timestamp_bin_counter[15:8], 8'd0};

CommaOut <= 4'b1_0_0_0;
```

Load column 2

```
DataOut <= {8'h1C, 8'hAA, 8'h1C, 8'hAA};

CommaOut <= {1'b1,1'b0,1'b1,1'b0};
```

Load pixel 2 and there are hits

```
DataOut <= {8'hC0, timestamp_bin_counter[15:8], timestamp_bin_counter[7:0], timestamp_gray_value[7:0]};

CommaOut <= 4'b0_0_0_0;
```

Read column 2

DataOut <= {8'hC1,2'd0, ColAddFromDet[4:0], RowAddFromDet[9:0], TS3FromDet[6:0]};

CommaOut <= {1'b0,1'b0,1'b0,1'b0};

Read column 4

DataOut <= {2'd0, TSFromDet[19:0], TS2FromDet[9:0]};

CommaOut <= {1'b0,1'b0,1'b0,1'b0};

In all other cases, the data out register is filled with following pattern

DataOut <= {8'hBC, 8'hBC, 8'hBC, 8'hBC};

CommaOut <= {1'b1,1'b1,1'b1,1'b1};

TSFromDet is the time stamp of the rising hit edge received from the matrix. TS2FromDet is the time stamp of the falling signal edge. TS3FromDet is the time stamp from TDC. ColAddFromDet and RowAddFromDet are the column and row addresses.

Configuration registers

The configuration of the chip is stored in a single long shift register. It has three parts 1) the configuration register in RCU, 2) bias DAC register and 3) pixel configuration register.

Figure 7 shows the structure of one register bit. The bit uses Ck1, Ck2, Sin, Sout, Ld and Rb signals.

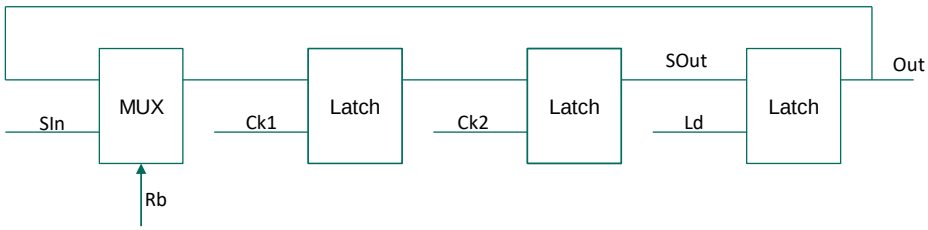

Figure 7: Structure of one bit

Configuration register in RCU has the following structure SIN_pad - > QConfig[0:56] -> SOUT_config.

SIN_pad is connected to chip pad, SOUT_config is internal chip line between two register parts.

Explanation of the bits:

ckdivend = QConfig [5:0] clock divider for the time stamp counter

ckdivend2 = QConfig [11:6] clock divider for the falling edge time stamp counter

timerend = QConfig [15:12] clock divider for the state machine

slowdownend = QConfig [19:16] length of the Load Column signal

maxcycend = QConfig [27:20] number of hits read from EoC buffers before new are loaded

resetckdivend = QConfig [31:28] clock divider for Sync state

sendcounter = QConfig [32] if 1 state machine goes to the debug- send counter state

ckdivend3 = QConfig [38:33]; //clock divider for the TDC time stamp counter

ts_counter_ts1_6bit = QConfig [39] if 1 the falling edge counter works as 6 bits (only used in HVMAPS25 chip)

countsheeps = QConfig [40] count mode enable

SRExtraBits[15:0] = QConfig[56:41];

- > SOUT_config

Connections of extra bits

SRExtraBits[0] inverting of clock – used in the PLL

SRExtraBits[1] external clock enable – used in the PLL

SRExtraBits[2] slow clock enable – used when ref clock is slower than about 40MHz

SRExtraBits[3] enable PLL

SRExtraBits[4] 1: clk_800p as TS clock 0: clk_4n as TS clock

SRExtraBits[5] only for MPRO chip: SelPkB

SRExtraBits[6] only for MPRO chip: SelPk

SRExtraBits[7] only for MPRO chip: Cluster Enable

SRExtraBits[8] only for MPRO chip: Cluster Enable B

SRExtraBits[9] only for MPRO chip: Always Enable

DAC register

SOUT_config - >

q00 (=0)

q01 (=0)

qon0 0: resistive reference; 1 band-gap reference (default 0, band gap not implemented)

qon1 1 on code (default 1)

qon2 0 on code (default 0)

qon3 1 on code (default 1)

Comment: every DAC has 6 bits sin -> DAC(5:0) -> sout, DAC5 is MSB of the DAC code.

anadac0 IBLRes (default 5) 1/resistance of the CR filter before comparator in pixel

anadac1 VNCompFine (default 5) bias current of the comparator in TDC

anadac2 VN (default 20) bias current of the input transistor in the pixel amplifier

anadac3 INFB (default 10) 1/resistance of the feedback of the pixel amplifier

anadac4 VNFoll (default 5) bias current for the source follower attached to pixel amplifier

anadac5 VNRegC (default 5) pull down current for the regulated cascode receiver

anadac6 VNDel (default 10) bias of the delay element of the edge detector in hit buffer

anadac7 IPBigFine (default 5) fast ramp current of the TDC

anadac8 IPDAC (default 0) bias current for the TDAC placed in hit buffer

anadac9 VN (default 0) bias current of the input transistor in the pixel amplifier, adds to anadac2

anadac10 INFoll2 (default 30?) bias current for the slow source follower in hidr pixels

anadac11 INBias (default 0) 1/bias resistance of the pixel n-well

anadac12 IPLoad (default 10) bias for the load current source of the pixel amplifier

anadac13 VNComp2 (default 5) leakage current for the distributed comparator

digdac0 vpvco (default 16)

digdac1 vnvco (default 16)

digdac2 vpdclmux (default 32)

digdac3 vndclmux (default 32)

digdac4 vpdeldcl (default 32)

digdac5 vndeldcl (default 32)

digdac6 vpdelpreemp (default 32)

digdac7 vndelpreemp (default 32)

digdac8 vpdcl (default 32)

digdac9 vndcl (default 32)

digdac10 vnlvds (default 16)

digdac11 vnlvdsdel (default 0)

digdac12 vppump (default 16)

anadac14 IPRegCasc (default 5) regulated cascode amplifier bias

anadac15 IPSmallFine (default 30?) slow ramp current of the TDC

anadac16 VNComp (default 10) bias current of the pixel amplifier

anadac17 IPFoll (default 10) ampout source follower current and hit bus pull up current

anadac18 VNDAC (default 0) bias current for the TDAC placed in pixel

anadac19 IPBiasRec (default 5) pull up current for the receiver of the hit buffer

anadac20 VNBiasRec (default 5) threshold of the receiver in the hit buffer

-> SOUT_DAC

Pixel configuration register

SOUT_DAC ->

Column 0

Bit 0: hit buffer RAM In for cell 0 (see Fig 5)

Bit 1: hit buffer RAM In for cell 1

Bit 2: hit buffer RAM In for cell 2

Bit 3: hit buffer RAM In for cell 3

Bit 4: pixel RAM In for all 3 cells (see Fig 1)

Bit (5:16): hit buffer RAM Write for rows (0:4), pixel Wr RAM(0:6)

Comment:

Notice there are 3 RAM write signals/pixel row. There can be up to 124 pixels in one column, however only 62 with in pixel RAM. Since there are only 62 rows with pixel RAM, there are 62x3=186 RAM write signals in total. The lowest 62 pixel never have RAM, the first row that can have RAM is row62.

Example: RAM(0:6) means RAM(0:2)_row62, RAM(0:2)_row63, RAM(0)_row64

Bit (17:21): enable injection pixel row (0:4)

Bit 22: enableB (active 0) hit bus for the column – hit bus is the OR function of receiver outputs (after local enable bit) in all hit buffers. Hit bus is designed to be fast – delay 10 ns with respect to hit should be possible.

Bit 23: amp out enable for the column – amp out is the multiplexer that allows measuring of SFOut signals of all pixels in the bottom row.

Bit 24: capacitive charge injection enable for the column

Column 1

Bit 0: hit buffer Ram In for cell 0

Bit 1: hit buffer Ram In for cell 1

Bit 2: hit buffer Ram In for cell 2

Bit 3: hit buffer Ram In for cell 3

Bit 4: pixel Ram In for all 3 cells

Bit (5:16): hit buffer write RAM for rows (9:5), pixel write RAM(13:7)

Comment:

Notice that the indices in RAM(i:j) are now descending, this is so in every odd column

Bit (17:21): enable injection pixel row (9:5)

Comment:

Notice that the indices in enable(i:j) are now descending, this is so in every odd column

Bit 22: enableB (active 0) hit bus for the column

Bit 23: amp out enable for the column

Bit 24: capacitive charge injection enable for the column

Column 2

Bit 0: hit buffer Ram In for cell 0

Bit 1: hit buffer Ram In for cell 1

Bit 2: hit buffer Ram In for cell 2

Bit 3: hit buffer Ram In for cell 3

Bit 4: pixel Ram In for all 3 cells

Bit (5:16): hit buffer write RAM for rows (10:14), pixel write RAM(14:20)

Bit (17:21): enable injection pixel row (10:14)

Bit 22: enableB (active 0) hit bus for the column

Bit 23: amp out enable for the column

Bit 24: capacitive charge injection enable for the column

…

Column 28

…

-> SOUT_Pad

Pads

The chips have IO pads placed at the bottom edge and debug (monitor) pads placed at the left and right edges. One chip Matrix_v3 has additional digital pads (placed at left and right edges) that allow daisy chain readout. **There pads are shown in figures 8 and 9.**

The list of the IO pads that are placed at the bottom of the chip

Pad pitch is 100µm

Number	Name	Protection	Description
1	Vssa!	n	Analog power for input PMOS of pixel amplifier 1.2V
2	Gnda!	n	Analog ground
3	Vdda!	n	Analog power 1.8V
4	Vddram	y	Vdd for the RAM cells (1.5V – 1.8V)
	Thfine (thtdac)	y	Threshold for TDC 1.4V
	Gate	y	Gate voltage for NMOS comparator 2.1V
	SCK	y	Digital in, CMOS 1.8V, SPI
	CSB	y	Digital in, CMOS 1.8V, SPI
	MOSI	y	Digital in, CMOS 1.8V, SPI
	MISO	x	Digital out, CMOS 1.8V, SPI
	Res_n	y	Digital in, CMOS 1.8V, async reset active 0

	SyncResP	y	Digital in, LVDS, sync reset (on chip termination)
	SyncResN	y	Digital in, LVDS, sync reset (on chip termination)
	CkRefN	y	Digital in, LVDS, reference ck for PLL (no term.)
	CkRefP	y	Digital in, LVDS, reference ck for PLL (no term.)
	CkExtP	y	Digital in, LVDS, external ck (no term.)
	CkExtN	y	Digital in, LVDS, external ck (no term.)
	CCk1	y	Digital in, CMOS 1.8V, shift register interface
	CCk2	y	Digital in, CMOS 1.8V, shift register interface
	CLd	y	Digital in, CMOS 1.8V, shift register interface
	CSIn	y	Digital in, CMOS 1.8V, shift register interface
	CRb	y	Digital in, CMOS 1.8V, shift register interface
	CSOut	y	Digital out, CMOS 1.8V, shift register interface
	VLow	y	Ground for PLL 0V
	VHigh	y	Power for PLL 1.8V
	NoToVCO	y	VCO bias, connect 1k and 1n in series
	ToVCO	y	VCO bias, connect 1k and 1n in series
	VminusPD	y	Ground for pull down transistor 0v
	DataOutN	y	Digital out, LVDS, data out (connect 100Ω pullup)
	DataOutP	y	Digital out, LVDS, data out (connect 100Ω pullup)
	Vss (TDAC)	n	Current drain for regulated cascode receiver – 0.3V
	Gnd!	n	Digital ground
	Vdd!	n	Digital power – 1.8V
	Hbout	y	Source follower output – hit bus, connect 1kΩ to ground
	Inj	y	Analog input injection signal
	Ampout	y	Source follower output – amplifier output, connect 1kΩ to ground
	Vminuspix	y	Source voltage for the pixel line driver 0.5V – 1.5V
	Vcasc2	y	PMOS cascode voltage in the pixel amplifier 0.8V
	BLPix	y	Base line voltage of the pixel CR filter 1V in pixel
	Thpix2	y	Threshold for the slow comparator (BLPix + 50mV)
	Thpix	y	Threshold for the main comparator (BLPix + 50mV)
	Pwell	n	Pixel pwell bias -1V to 0V
	Vnwell	n	Pixel nwell bias 1.8V to 2.8V
	Vssa!	n	Analog power for the input PMOS of pixel amplifier 1.2V
	Gnda!	n	Analog ground
	Vdda!	n	Analog power 1.8V
	Sub! (chip ring)	n	Substrate -60V
	Sub! (pixels)	n	Substrate -60V

Following tables explain the matrix options:

Matrix 1 (Logo on chip I.C)

MatrixV1 cmos:

Column (0:13) Column CMOS v1 standard

Column (14:28) Column NMOS v1 standard

Column CMOS v1 standard:

Pixel place (0:61) Standard Pixel CMOS v1 without TDAC (0:61)

Pixel place (62:123) Standard Pixel CMOS v1 with TDAC (0:61)

Column NMOS v1 standard:

Pixel place (0:61) Standard Pixel NMOS v1 without TDAC (0:61)

Pixel place (62:123) Standard Pixel NMOS v1 with TDAC (0:61)

Standard CMOS pixel v1:

NMOS comparator (v1)

CMOS amplifier type

double PLoad source

Circular cascode

Standard NMOS pixel v1:

NMOS comparator (v1)

NMOS amplifier type

double PLoad source

Circular cascode

Matrix 2 (Logo on chip I.V)

MatrixV1 VarSze variable:

Column (0:7) Column v1 VarSize – standard type (single source, linear cascode, 4u input transistor)

Column (8:15) Column v1 VarSize – 8u input transistor

Column (16:23) Column v1 VarSize – double source

Column (24:28) Column v1 VarSize – circular cascode

Column v1 VarSize variable:

Pixel place (0:61) Variable 50µm Pixel PMOS v1 without TDAC (0:30) -> connected to hit buffer 0:61:2 (0,2,4,…)

Pixel place (62:123) Variable 100µm Pixel PMOS v1 without TDAC (0:15) -> connected to hit buffer 62:123:4 (62, 66…)

Variable Pixel PMOS v1:

NMOS comparator (v1)

PMOS amplifier type

PLoad source – double sided or single

Cascode – linear or circular

4µm or 8µm input transistor width

Matrix 3 (Logo on chip II.N)

MatrixV2 nmos:

Column (0:13) Column v2 standard

Column (14:28) Column NMOS v2 standard

Column v2:

Pixel place (0:61) Standard Pixel PMOS v2 without TDAC (0:61)

Pixel place (62:123) Standard Pixel PMOS v2 with TDAC (0:61)

Column NMOS v2:

Pixel place (0:61) Standard Pixel NMOS v2 without TDAC (0:61)

Pixel place (62:123) Standard Pixel NMOS v2 with TDAC (0:61)

Standard NMOS pixel v2:

CMOS comparator (v2)

NMOS amplifier type

double PLoad source

Circular cascode

Standard PMOS pixel v2:

CMOS comparator (v2)

PMOS amplifier type (4µm)

double PLoad source

Circular cascode

Matrix 4 (Logo on chip II.V)

MatrixV2 VarSze variable:

Column (0:7) Column v2 VarSize – standard type (single source, linear cascode, 4u input transistor)

Column (8:15) Column v2 VarSize – 8u input transistor

Column (16:23) Column v2 VarSize – double source

Column (24:28) Column v2 VarSize – circular cascode

Column v2 VarSize variable:

Pixel place (0:61) Variable 50µm Pixel PMOS v2 without TDAC (0:30) -> connected to hit buffer 0:61:2 (0,2,4,…)

Pixel place (62:123) Variable 100µm Pixel PMOS v2 without TDAC (0:15) -> connected to hit buffer 62:123:4 (62, 66…)

Variable Pixel PMOS v2:

CMOS comparator (v2)

PMOS amplifier type

PLoad source – double sided or single

Cascode – linear or circular

4µm or 8µm input transistor width

Matrix 5 (Logo on chip III.NS CEPC)

MatrixV3 nmos low power (daisy chain readout)

Column (0:13) Column v3 low power

Column (14:28) Column NMOS v3 low power

Column v3 low power:

Pixel place (0:61) Low power Pixel PMOS v3 without TDAC (0:61)

Pixel place (62:123) Low power Pixel PMOS v3 with TDAC (0:61)

Column NMOS v3 low power:

Pixel place (0:61) Low power Pixel NMOS v3 without TDAC (0:61)

Pixel place (62:123) Low power Pixel NMOS v3 with TDAC (0:61)

Low power NMOS pixel v3:

Distributed comparator (v3)

NMOS amplifier type with long current source

Long PLoad source

Linear cascode

Low power PMOS pixel v3:

Distributed comparator (v3)

PMOS amplifier type (4µm with long current source)

Long PLoad source

Linear cascode

Matrix 6 (Logo on chip III.V)

MatrixV3 VarSize variable:

Column (0:7) Column v3 VarSize – standard type (single source, linear cascode, 4u input transistor)

Column (8:15) Column v3 VarSize – 8u input transistor

Column (16:23) Column v3 VarSize – double source

Column (24:28) Column v3 VarSize – circular cascode

Column v3 VarSize variable:

Pixel place (0:61) Variable 50µm Pixel PMOS v3 without TDAC (0:30) -> connected to hit buffer 0:61:2 (0,2,4,…)

Pixel place (62:123) Variable 100µm Pixel PMOS v3 without TDAC (0:15) -> connected to hit buffer 62:123:4 (62, 66…)

Variable Pixel PMOS v3:

Distributed comparator (v3)

PMOS amplifier type

PLoad source – double sided or single

Cascode – linear or circular

4µm or 8µm input transistor width

Matrix 7 (Logo on chip II.H)

MatrixV2 VarSize variable hidr:

Column (0:7) Column v2 VarSize hidr – standard type (single source, linear cascode, 4u input transistor)

Column (8:15) Column v2 VarSize hidr – 8u input transistor

Column (16:23) Column v2 VarSize hidr – double source

Column (24:28) Column v2 VarSize hidr – circular cascode

Column v2 VarSize hidr variable:

Pixel place (0:61) Variable 50µm hidr Pixel PMOS v2 without TDAC (0:30) -> connected to hit buffer 0:61 (even index fast comparator, odd index slow comparator)

Pixel place (62:123) Variable 100µm hidr Pixel PMOS v2 without TDAC (0:15) -> fast comparator connected to hit buffer 62:123:4 (62, 66…), slow comparator connected to 63:123:4

Variable hidr Pixel PMOS v2:

Two CMOS comparators (v2)

PMOS amplifier type

PLoad source – double sided or single

Cascode – linear or circular

4µm or 8µm input transistor width

Matrix 8 (Logo on chip III.H)

MatrixV3 VarSize variable hidr:

Column (0:7) Column v3 VarSize hidr – standard type (single source, linear cascode, 4u input transistor)

Column (8:15) Column v3 VarSize hidr – 8u input transistor

Column (16:23) Column v3 VarSize hidr – double source

Column (24:28) Column v3 VarSize hidr – circular cascode

Column v3 VarSize hidr variable:

Pixel place (0:61) Variable 50µm hidr Pixel PMOS v3 without TDAC (0:30) -> connected to hit buffer 0:61 (even index fast comparator, odd index slow comparator)

Pixel place (62:123) Variable 100µm hidr Pixel PMOS v3 without TDAC (0:15) -> fast comparator connected to hit buffer 62:123:4 (62, 66…), slow comparator connected to 63:123:4

Variable hidr Pixel PMOS v3:

Two distributed comparators (v3)

PMOS amplifier type

PLoad source – double sided or single

Cascode – linear or circular

4µm or 8µm input transistor width

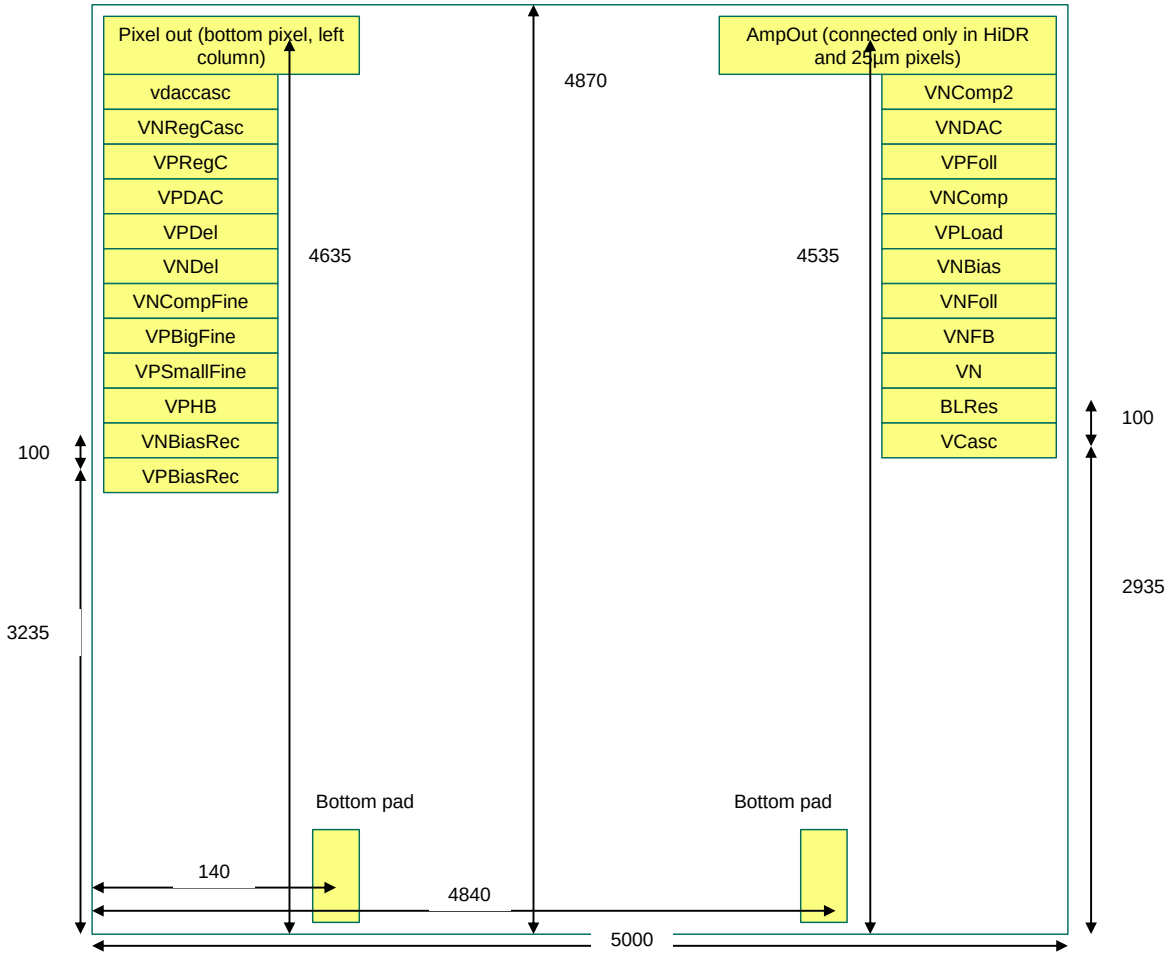

Figure 8: Debug (monitor) pads for all chips except Matrix 5 (Logo on chip III.NS CEPC). Dimensions are in µm.

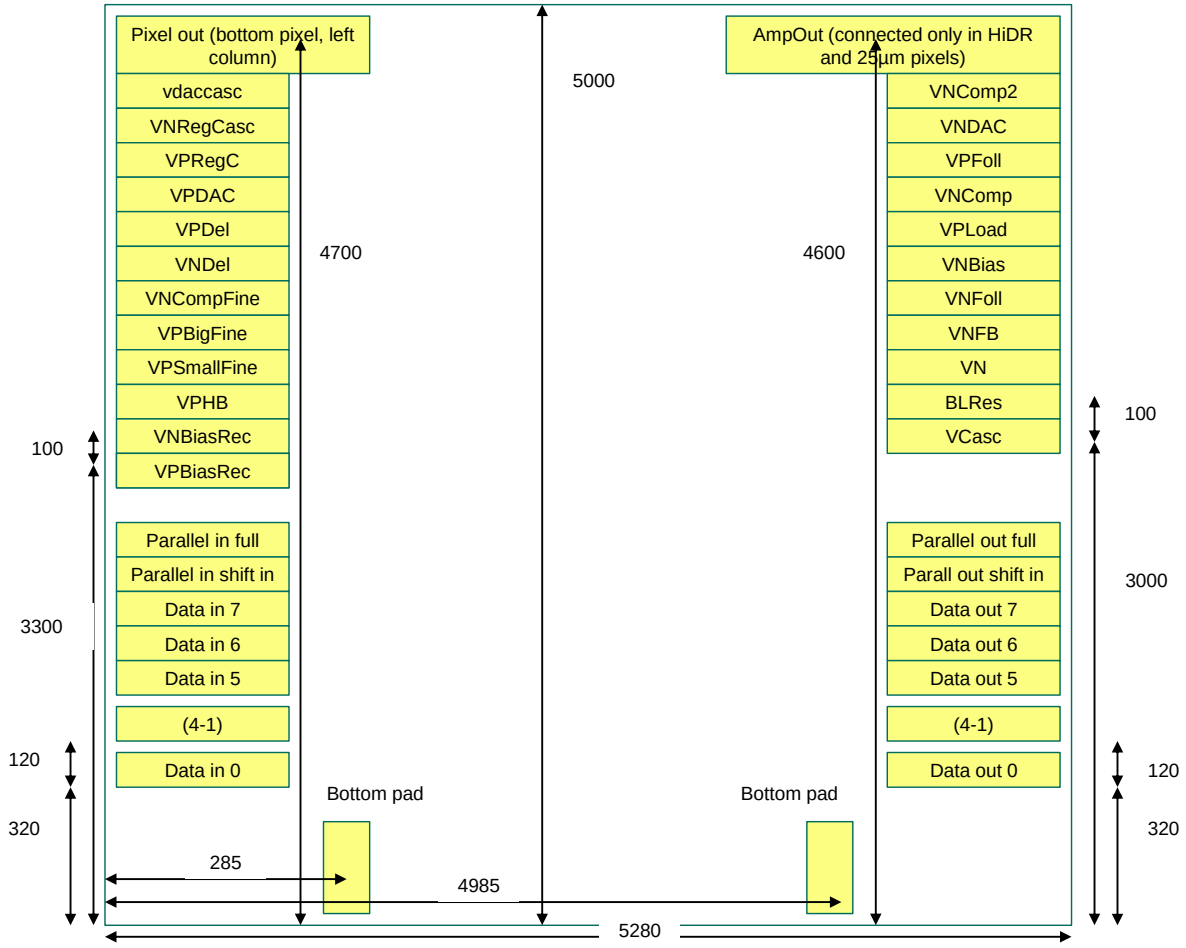

Figure 9: Debug (monitor) pads and digital pads for Matrix 5 (Logo on chip III.NS CEPC). Dimensions are in µm.