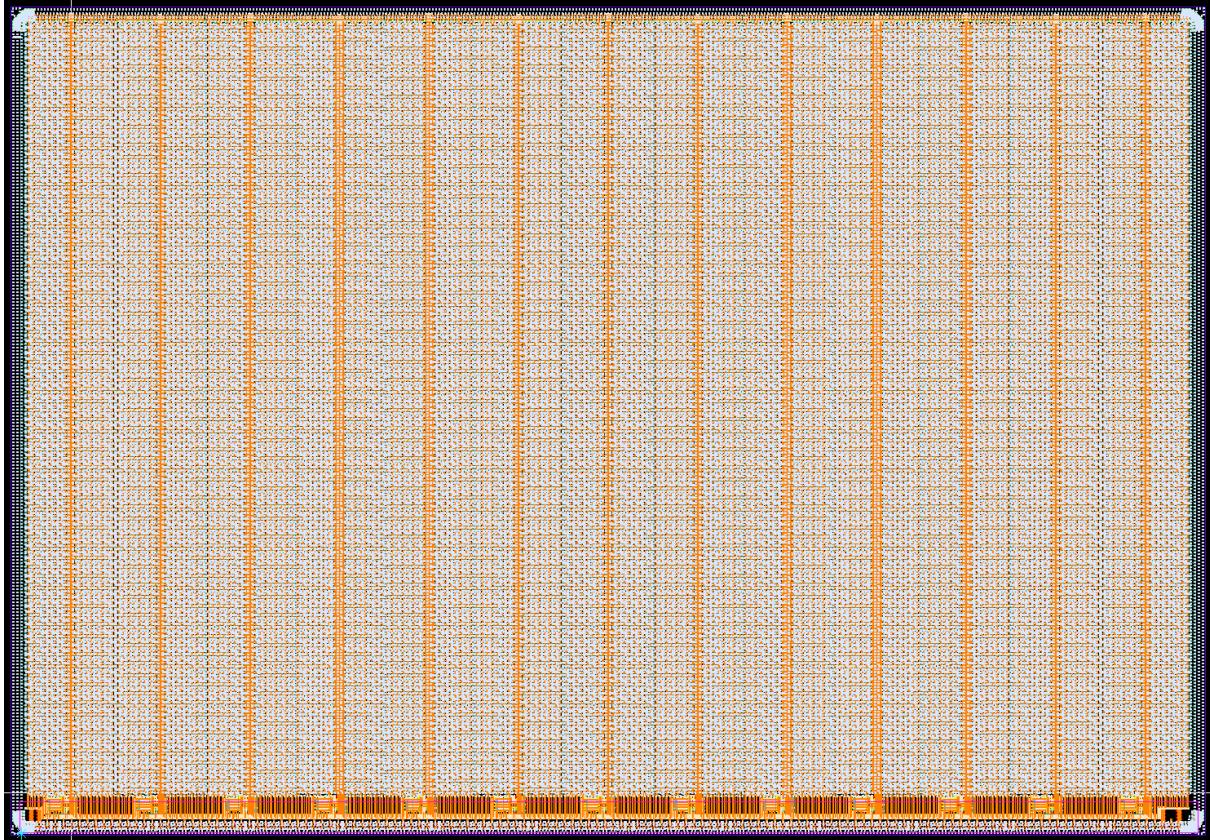


# FASER ASIC

## Technical document

contact: lorenzo.paolozzi@cern.ch, thanushan.kugathasan@cern.ch, roberto.cardella@unige.ch,  
carlo.fenoglio@unige.ch, yannick.favre@unige.ch

- This document describes the production ASIC for the FASER experiment covering specifications, architecture and features. The design features an in-pixel analog memory for the acquisition of the pixel charge and an in-pixel front-end electronics, comprising HBT-based discriminator.



- Preliminary document for internal review. Please, do not circulate.

- 9   **In RED, notes, potential additional features and items that will be revised or under decision are reported.**
- 10   **[YFAV: Yannick's notes are reported in this format]**
- 11   **[SGS: Sergio's comments]**
- 12   **[Brian: Brian's comments]**
- 13   **[Roberto: Roberto comments]**
- 14   **[ANS: Answers to comment when needed]**
- 15   **[TASK: TASK]**
- 16   **[Fulvio: In GREEN, I added some comment to highlight the modifications I did according to Yannick's suggestions. They can be deleted when the modifications are approved.]**
- 17

# 18 Contents

19	<b>1 Chip specifications</b>	4
20	<b>2 Chip architecture</b>	5
21	2.1 General architecture . . . . .	5
22	2.2 Pixel row . . . . .	8
23	2.3 Super-pixel . . . . .	10
24	2.4 Time to digital converter . . . . .	14
25	2.5 Typical readout operation . . . . .	15
26	2.6 Digital logic implementation . . . . .	18
27	2.6.1 Periphery . . . . .	18
28	2.6.2 TDC control logic . . . . .	19
29	<b>3 User and implementation guide</b>	22
30	3.1 SLOW CONTROL . . . . .	22
31	3.2 CONFIGURATION PROGRAMMING . . . . .	25
32	3.3 RESET . . . . .	26
33	3.4 PIXEL PROGRAMMING . . . . .	28
34	3.5 READOUT . . . . .	28
35	3.6 Testpulse . . . . .	30
36	3.6.1 Testpulse synchronization on multiple ASICs . . . . .	30
37	3.7 PINOUT . . . . .	31
38	3.8 LV/HV DESCRIPTION . . . . .	34
39	3.9 IMPLEMENTATION . . . . .	34
40	3.10 CALIBRATIONS . . . . .	35
41	<b>4 Simulated performance</b>	39
42	4.1 DAC characterization . . . . .	39
43	<b>5 Appendix</b>	42
44	5.1 Reset modes timing diagrams . . . . .	42
45	5.2 Reset FSM . . . . .	45

## 46 1 Chip specifications

- 47 The FASER chip is a monolithic silicon pixel sensor designed for the upgrade of the pre-shower module of the  
 48 FASER experiment at CERN. The chip will be able to discriminate the primary photons with energies ranging  
 49 from 100 GeV to 4 TeV, with a spatial separation of the primary photons above 200  $\mu\text{m}$ . The photons interact  
 50 synchronously (< 1 ps) on tungsten layers placed at few cm distance, generating particle showers.
- 51 The detector will be able to measure independently the charge in each pixel and time tag the events at super-pixel  
 52 level with a time resolution of 300 ps or better. The expected charge per pixel ranges from 0.5 fC to 64 fC. Charge  
 53 measurement is required at pixel level with a 3-bit resolution and logarithmic compression of the dynamic range. In  
 54 the ASIC, a 4-bit ADC per super-pixel is integrated. The additional bit is adopted as a safety margin to compensate  
 55 for potential mismatch and leakage of analog memories (see Section 2).
- 56 The detector will be produced using the SG13G2 SiGe BiCMOS process from IHP microelectronics, with the  
 57 standard processing on special wafers. The die size will be adapted to the extended reticle of the aforementioned  
 58 process.
- 59 The main chip specifications<sup>12</sup> are given in table 1. The ASIC implements a matrix of  $208 \times 128 = 26624$   
 60 hexagonal pixels with a side of approximately 65  $\mu\text{m}$ . The exact dimensions that can be used to implement the  
 61 pixel matrix in simulation and reconstruction tools are the pixel vertical pitch (Y coordinate) of 111.70  $\mu\text{m}$  and the  
 62 horizontal pitch (X coordinate) of 97.73  $\mu\text{m}$ . The X coordinate is parallel to the I/O pad line.

Table 1: Main chip parameters. Note the over-currents are peak currents (with a duration of the order of  $\sim$ 100 ns) generated only when particles are detected and they should not be added to the average currents reported in this table. The digital over-current accounts for the TDC contribution only when the TDC power saving mode circuit is enabled. **The maximum currents reported in this table are meant as maximum configurable in the ASIC, which is much larger than the maximum expected for operation, which is reported in table 8 in Section 3.**

	Nominal	Min	Max
Chip Area (XY) <sup>1</sup> [mm <sup>2</sup> ]	$22.154 \times 15.345$	-	-
Chip Thickness [ $\mu\text{m}$ ]	150	-	-
Number of pixels (XY)	$13 \times 8 \times 256 = 26624$	-	-
Hexagonal pixel side [ $\mu\text{m}$ ]	65.0	64.5	65.5
Pixel vertical pitch [ $\mu\text{m}$ ]	111.70	-	-
Pixel horizontal pitch [ $\mu\text{m}$ ]	97.73	-	-
Super pixel size (XY) [ $\mu\text{m}^2$ ]	$1661.41 \times 1787.2$	-	-
Digital column width [ $\mu\text{m}$ ]	60	-	-
Active Area (XY)[mm <sup>2</sup> ]	$21.61 \times 14.36$	-	-
Dead area periphery (I/O side) [ $\mu\text{m}$ ]	720	-	-
Dead area periphery (GR sides) [ $\mu\text{m}$ ]	260	-	-
Vcc_a [V]	1.2	1.2	1.8
Vdd_disc [V]	1.2	1.1	1.3
Vdd_dig [V]	1.2	1.1	1.3
Vdd_osc [V] (can be connected to Vdd_dig)	1.2	1.1	1.3
Analogue current [mA]	80	13	350
Discriminator current [mA]	105	26	500
Digital current (TDCs + drivers) [mA]	110	50	-
Global threshold current <sup>2</sup> [mA]	8	<1	Disc current
Analogue peak over-current / pixel [ $\mu\text{A}$ ]	0	0	0
Discriminator peak over-current / pixel [ $\mu\text{A}$ ]	0	0	3
Digital peak over-current [mA]	3	1	8

<sup>1</sup>The chip area does not account the extra silicon between the edge of the ASIC layout and the dicing line, which can vary between 30 - 45  $\mu\text{m}$  (per diced side).

<sup>2</sup>The global threshold can be generated internally or forced externally with the dedicated pad. The current specifications are reported for this second case. If generated internally, an extra maximum current of 25 mA should be added to the Discriminator line.

## 63 2 Chip architecture

### 64 2.1 General architecture

65 The active area of the chip is made by a matrix of hexagonal pixels, organised in 13 super-columns of  $16 \times 128$   
66 pixels each. Each super-column implements an active sensing region with pixels analog circuitry and a non-sensing  
67 region in the middle taken by the digital readout column. The super-columns are divided vertically in eight super-  
68 pixels of  $16 \times 16$  pixels, which constitute the basic block of the readout logic. Figure 1 shows a representation of  
69 the super-columns and super-pixels in the ASIC.

70 When a particle shower hits the active area, the pixel charge information is stored on an in-pixel analog memory  
71 followed by an in-pixel HBT-based comparator. Pixels in a super-pixel have grouped their fast-OR signals in 3  
72 groups to avoid a direct fast-OR with adjacent pixels. These 3 fast-OR signals per super-pixel are sent to the digital  
73 periphery to generate a trigger. The time of arrival of the fast-OR signals are digitized by a TDC located below  
74 the super-column. The digital periphery logic acts as arbiter, a programmable *readout delay* from 1 clock cycle up  
75 to  $255 * 4$  clock cycles is set between the trigger reception and the readout start to allow the charging of analog  
76 memories (see Figure 2). The typical *readout delay* should be set above 500 ns. This delay is needed to wait for the  
77 trailing edge of the discriminated hit. Different front-end configurations might require adjusting this delay value.  
78 As an example, shorter delays can be tolerated by increasing the pre-amplifier feedback current, thus reducing the  
79 time over the threshold, at the expense of charge resolution. In FASER, a long delay will be used to ensure the  
80 external trigger signal arrives to the readout electronics before the data from the pixels starts to arrive.<sup>3</sup>

81 During the super-column readout, the charge stored into the analog memories is digitized at super-pixel level.  
82 Corresponding pixels from different super-pixels are digitized at the same time using an analog mux 256:1 and a  
83 4-bit flash ADC. Those pixels are then transferred and the readout moves to the following pixels. This solution  
84 allows limiting the amount of area dedicated to the digital memories to 5 bits per super-pixel: 1 bit is used as a flag  
85 to indicate whether a hit occurred or not while the remaining 4 bits represent the output of the conversion from the  
86 ADCs. After changing word in the MUX, an *ADC delay* can be set before reading the ADC. The *ADC delay* is  
87 configurable in a range of 3 to 17 clock cycles with two clock cycles steps and it can be used to give time to the  
88 output of the mux to stabilize.

89 Figure 3 shows the simplified diagram of the FASER ASIC. As anticipated, each super-column is communicating  
90 with a super-column logic block. The latter is featuring an end-of-column logic (EOC in Figure 3) that aims to  
91 acquire the data from the correspondent TDC of the column and send them, together with the charge data from  
92 the pixels, to the periphery logic during readout. More in general, the EOC is acting as an interface of the super-  
93 column logic for the communication with the rest of the digital electronics. The fast-or generated from the analog  
94 columns are used to sample the TDC. Moreover, they are sent to the periphery logic to start the readout.

95 More details on the characteristics of the digital logic of the FASER ASIC are reported in Section 2.6.

96 **NOTE: by convention, all numbering for arrays and indexes start from 0. Nevertheless, odd (super)columns  
97 are counted starting from the first (super)column to the left. Similarly, even (super)column are counted  
98 from the second left-most (super)column.**

---

<sup>3</sup>To be noted that the *readout delay* sets the time window for collecting additional hit within a readout frame, before masking the pixel during readout. Delaying the start of the readout can be also used to delay the data output with respect to the arrival of the external trigger from the experiment.

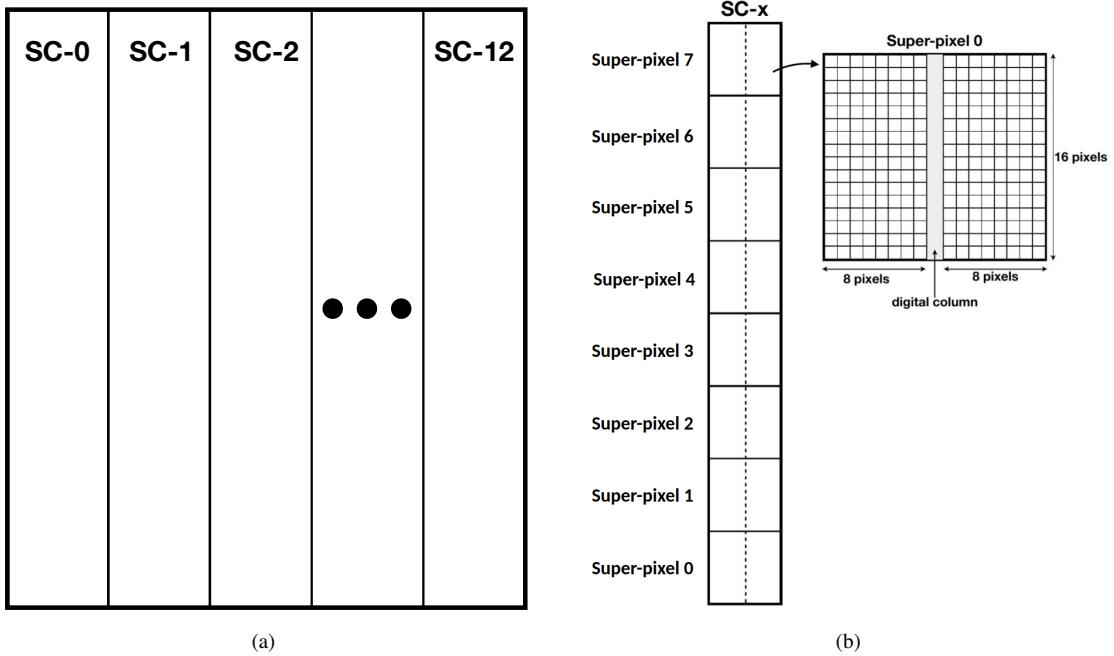


Figure 1: (a) Sketch of the FASER ASIC, composed of 13 super-columns. By convention, SC-0, SC-2, SC-4 and so on are defined as *odd columns*. The others are defined as *even columns*. (b) Sketch of the logical arrangement of pixels in eight super-pixels inside a super-column. Digital periphery is at the bottom of the super-column (below Super-pixel 0). The insert on the top-right shows a sketch of a super-pixel. For simplicity pixels have been drawn square instead of hexagonal.

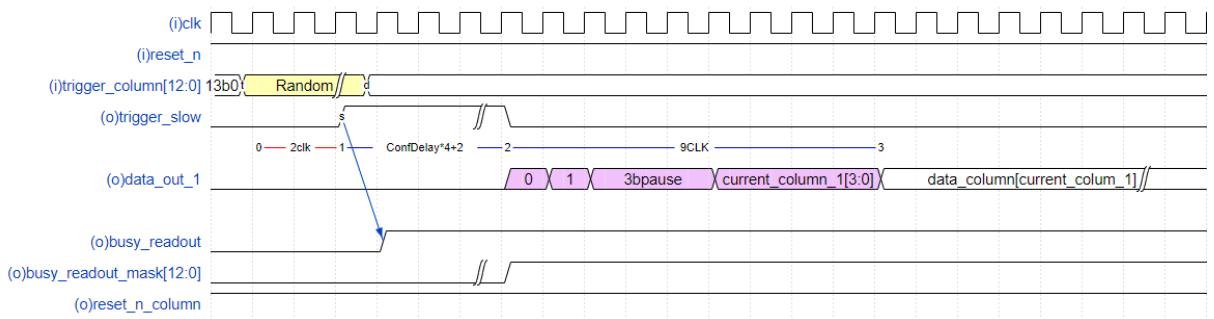


Figure 2: The OR of the pixels discriminator is sent to the TDCs and then to the periphery readout (*trigger\_column*). The *trigger\_slow* is generated by the periphery. It is synchronized version of the *trigger\_column* and its duration is set by  $\text{Conf\_delay} = \text{readout\_delay}$  (the value of *readout\_delay* is set during the programming phase and it will be explained in Chapter 3.2). The *readout delay* should be larger than the duration of the *trigger\_column*, which depends on front-end configuration and charge pulse, and could be as long as 500 ns. **NB:** the duration of *trigger\_column* in the picture is only an example. The readout is masked after the *readout\_delay*. The columns are reset for 100ns after the end of the readout.

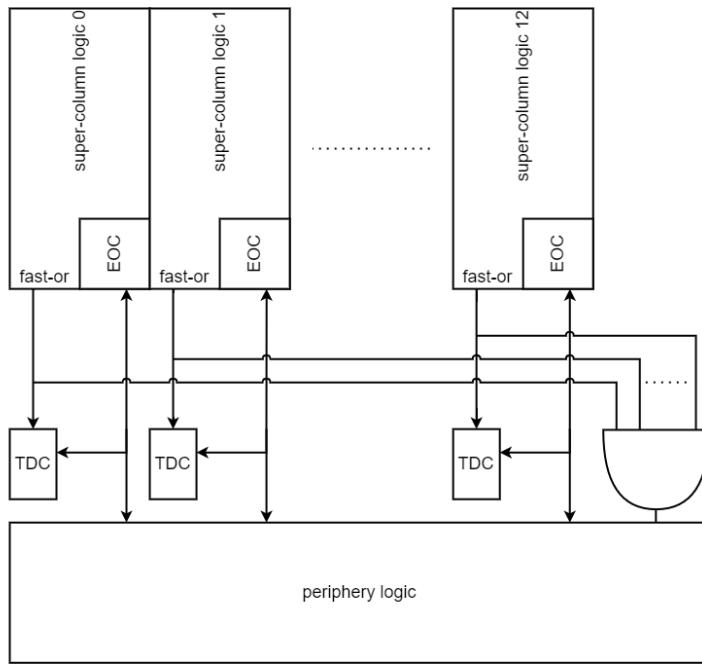


Figure 3: Conceptual diagram of the digital readout of the FASER ASIC. Each super-column logic aims to acquire the data from the corresponding analog super-column (not drawn in the figure for the sake of simplicity) and, more generally, to handle the communication with it (e.g., pixel masking configuration). In this diagram, the fast-OR lines, produced by the super-columns represent the logical OR of the discriminator outputs and they are used to start the readout. More details are given in Section 2.6.

## 99 2.2 Pixel row

100 The pixel row (Figure 4) is the basic building block of the analog matrix. It is made of 8 pixels per side and it  
 101 includes the pixel biases, the pre-amplifiers, the discriminator, calibration pulse generation and distribution and  
 102 local analog memory controls.

103 The pre-amplifier is designed to produce a signal time-over-threshold (TOT) proportional to the logarithm of the  
 104 input charge. While the signal is over threshold, the memory control connects the analog memory to the load  
 105 current, producing a linear charge. When the signal goes below threshold again, the memory is disconnected and  
 106 left floating until it is read by the flash ADC. The current leakage in the memory is low enough to guarantee an  
 107 error below 1 LSB for a readout time below 200  $\mu$ s (see Figure 5 for the analog memory control schematic).

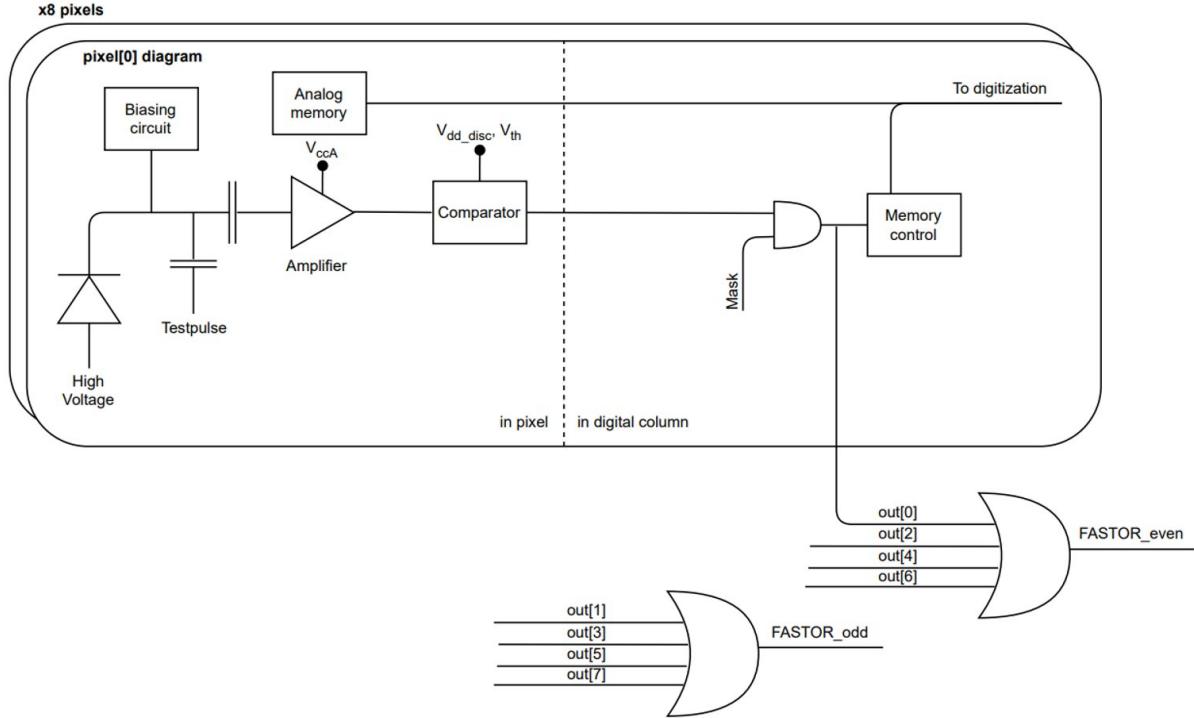


Figure 4: Pixel row conceptual diagram (one side). For simplicity, the testpulse circuits are not shown.

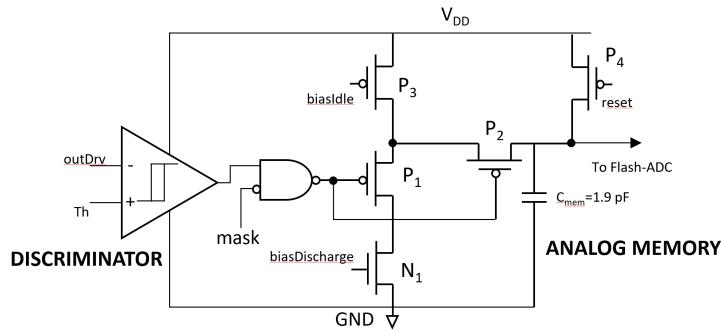


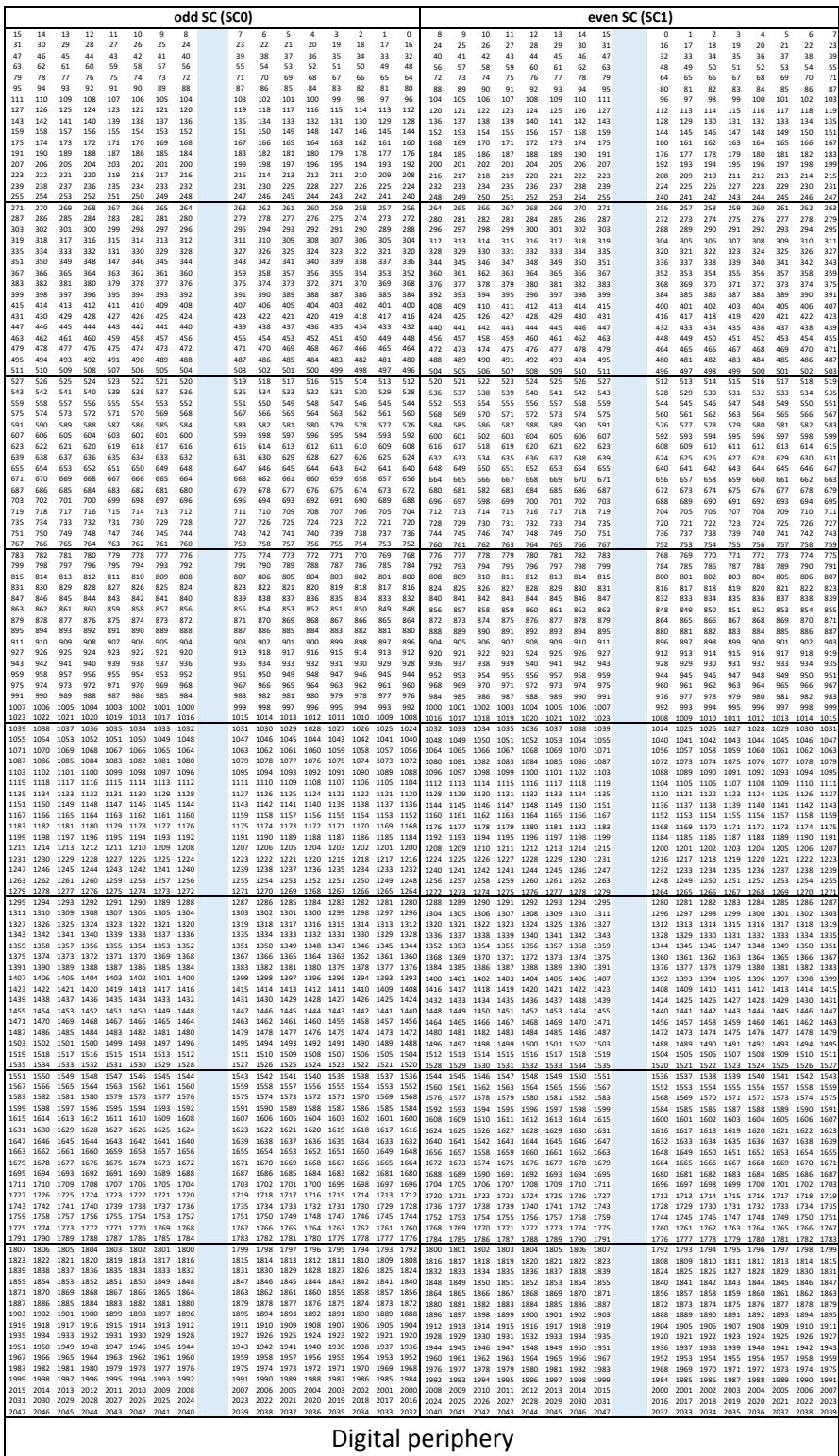
Figure 5: Analog memory control. The discriminator includes the masking block for the sake of simplicity.

108 Each pixel can be masked after the discrimination stage. The mask control bit is also used to enable the sending  
 109 of the test-pulse to the test-pulse capacitor (when mask is 0) and allows for the test-pulse distribution. The test  
 110 capacitors are connected to the pixels only if the *enable\_testpulse* switch is also active. More details on the testpulse  
 111 operation are provided in Sec. 3.6. Pixel masking for a super-column is done with a long shift register, mapped in  
 112 the digital periphery, as 128 arrays of 16 bit each: indeed, every super-column is composed by 128 rows and each  
 113 of them has 8+8 bit (left and right part of the super-column). Figure 6 shows the map of the mask (Masking bit

order is the index of the shift register as it comes out from the periphery: zero is the first bit to come out, 2047 is the last one). Figure 7 shows the map of the masking bit index used to activate the pulse injection in pixels.

Masking bit order - index of shift register as it comes out from the periphery: zero is the first bit to come out, 2047 is the last one

Map is geometrical



odd SC (SCO)																even SC (SC1)																
15	11	13	9	11	13	9	15	0	6	2	4	6	2	4	0	8	14	10	12	14	10	12	8	7	3	5	1	3	5	1	7	
31	27	29	25	27	29	25	31	16	22	20	22	18	20	24	16	24	30	26	28	24	30	26	28	24	23	19	21	17	19	21	17	23
47	43	45	41	43	45	41	47	32	38	34	36	38	34	36	32	40	46	42	44	46	42	44	40	39	35	37	33	35	37	33	39	
63	59	61	57	59	61	57	63	48	54	52	54	56	52	54	50	56	62	58	56	62	58	56	54	55	51	53	49	51	53	49	55	
79	75	77	73	75	77	73	79	44	49	46	48	49	46	48	44	47	52	49	46	52	49	46	48	72	67	70	65	69	67	65	71	
95	91	93	89	91	93	89	95	80	86	82	84	86	82	84	80	84	94	90	92	94	90	92	88	87	83	85	81	83	85	81	89	
111	107	109	105	107	109	105	111	96	102	98	102	100	98	100	96	104	110	106	108	108	110	106	108	104	103	99	101	97	99	101	97	103
127	123	125	121	123	125	123	127	112	118	114	116	118	114	116	112	120	120	122	124	120	122	124	120	119	115	117	113	115	117	113	119	
143	139	141	137	139	141	137	143	128	134	130	132	134	130	132	134	136	142	142	138	140	142	138	140	136	135	131	131	129	131	133	129	135
159	155	157	153	157	155	153	159	144	150	146	152	146	150	144	146	152	156	158	156	158	156	158	154	153	151	151	149	149	147	149	151	
175	171	173	169	171	173	169	175	160	166	164	166	164	166	160	162	172	170	174	170	172	170	174	167	167	163	163	161	163	165	163		
191	187	189	185	187	189	185	191	176	182	178	182	178	180	176	184	190	186	188	186	188	186	188	184	183	179	179	181	179	183	179	185	
207	203	205	201	203	205	201	207	192	198	196	198	194	196	192	190	200	200	202	200	204	200	202	200	199	195	197	193	195	197	193	199	
223	219	225	217	219	221	217	219	208	214	210	213	214	210	212	212	216	222	218	216	220	222	218	216	215	211	213	209	213	209	215		
239	235	237	233	235	237	233	239	224	230	226	232	230	226	228	224	232	238	234	236	238	234	236	232	231	227	229	225	227	229	225	231	
255	251	253	249	251	253	249	255	240	246	242	244	242	246	240	242	246	252	254	252	254	252	254	250	248	244	246	244	246	244	248		
271	267	269	265	267	269	265	271	256	252	258	260	252	258	256	252	256	260	266	260	266	260	266	264	263	259	261	259	261	257	263		
287	283	285	281	283	285	281	287	272	278	276	278	276	272	278	276	272	280	286	284	286	284	286	280	279	275	277	273	275	277	279		
303	299	305	297	301	299	307	303	288	294	292	294	292	288	296	302	298	304	302	298	304	302	298	304	295	291	293	289	295	291	293		
319	315	317	313	315	317	313	319	304	310	305	308	310	305	308	304	312	318	314	316	318	314	312	312	311	307	309	305	307	309	305	311	
335	331	333	329	331	333	329	335	320	325	322	324	320	322	320	324	332	338	336	334	332	338	336	332	337	323	325	323	325	323	325	337	
351	347	349	345	347	349	345	351	336	342	348	346	348	346	344	342	346	352	354	352	354	352	354	348	343	341	343	341	343	343			
367	363	365	361	363	365	361	367	352	358	354	356	354	356	352	354	356	360	366	362	364	362	364	360	359	355	357	353	355	357	353	359	
383	371	383	377	381	377	383	385	368	374	370	374	370	368	376	382	376	380	382	376	380	382	376	376	375	375	373	375	373	375	375		
399	395	397	393	395	397	393	399	384	390	386	388	390	386	388	384	392	394	394	396	394	396	392	391	387	389	385	387	389	385	391		
415	411	413	409	411	413	409	415	400	406	404	406	404	406	400	408	414	410	412	414	410	412	408	407	403	405	401	403	405	401	407		
431	427	429	425	427	429	425	431	413	420	418	420	418	420	416	414	420	416	420	416	420	416	420	413	429	427	429	427	429	427	429		
447	443	445	441	443	445	441	447	432	438	434	436	438	434	436	432	440	446	442	444	444	446	440	439	435	437	433	435	437	433	439		
463	459	461	457	459	461	457	463	448	454	450	452	454	452	450	452	462	458	460	462	458	460	454	455	451	453	449	451	453	449	455		
479	475	477	473	475	477	473	479	464	470	466	468	464	472	478	474	476	478	474	476	472	478	474	472	471	467	469	465	471	467	469		
495	491	493	489	491	493	489	495	480	482	484	486	484	488	480	482	494	492	494	492	494	492	494	487	487	483	481	483	485	481	487		
511	507	509	505	507	509	505	511	496	502	498	500	502	498	500	496	504	510	506	508	510	506	504	500	499	501	497	499	501	497	503		
527	523	525	521	523	525	521	527	513	520	518	520	518	520	516	514	520	516	520	516	520	516	520	513	519	517	519	517	519	515			
543	539	541	537	543	539	537	543	528	534	530	532	530	532	528	536	542	538	540	542	538	540	536	535	531	533	529	531	529	535			
559	555	557	553	555	557	553	559	544	554	540	556	544	558	544	552	558	554	556	558	554	556	552	551	547	549	544	547	549	545	551		
575	571	573	569	571	573	569	575	560	566	562	564	560	566	558	562	574	570	572	574	570	572	568	567	563	565	561	563	565	561	567		
591	587	585	583	587	585	583	591	576	582	580	582	580	582	576	584	590	588	590	588	590	588	586	585	589	587	585	589	587	585	593		
607	603	605	601	603	605	601	607	594	600	596	600	596	600	594	598	604	596	602	598	602	596	598	597	596	594	596	598	596	597			
623	619	621	617	621	623	617	625	610	616	612	616	612	616	608	612	616	612	616	612	616	612	616	615	617	615	617	615	617	615			
639	635	637	633	635	637	633	639	624	630	626	630	626	630	624	628	634	626	632	628	634	626	632	624	623	621	623	621	623	621	623		
655	651	653	649	651	653	649	655	641	647	643	645	641	647	639	643	649	645	647	645	647	645	643	641	643	641	643	641	643	647			
671	667	673	667	671	673	667	675	656	662	658	662	658	662	654	660	666	656	662	658	662	656	660	655	653	657	653	655	657	653			
687	683	685	681	683	685	681	687	668	674	670	674	670	674	668	672</td																	

119 distribution is detailed in Figure 9. Figure 10 shows the FAST-OR index of each pixel in the typical super-pixel of  
 120 an odd and even super-column.

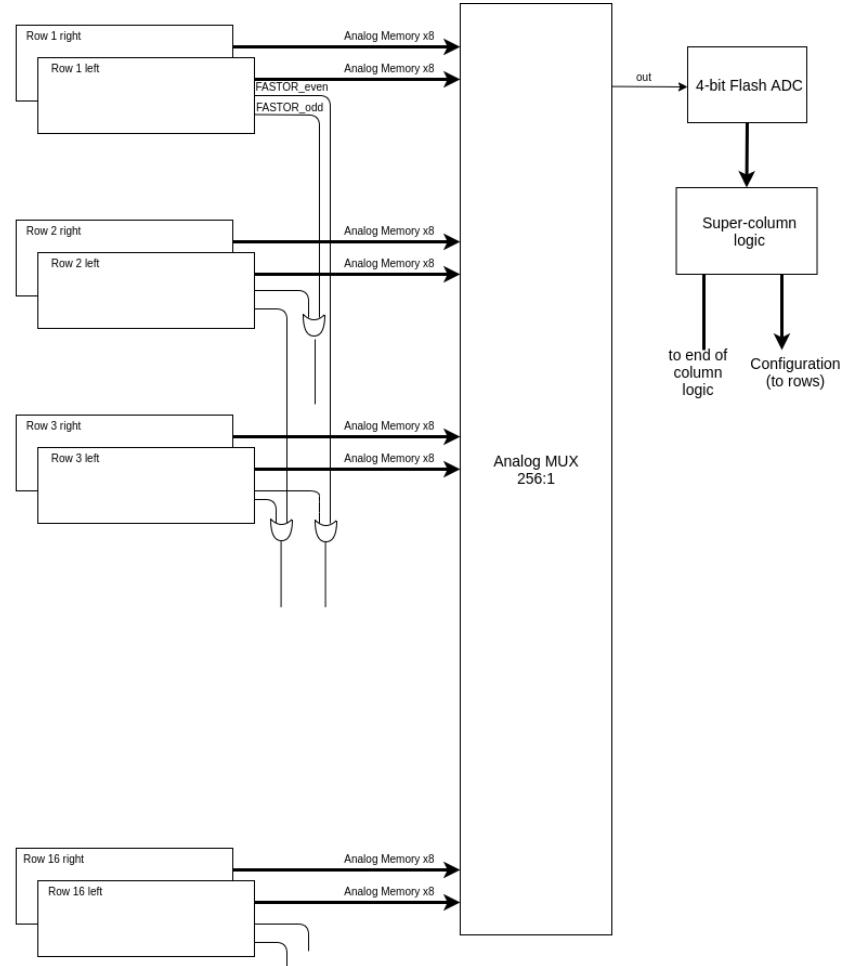


Figure 8: Conceptual diagram of the super-pixel. The connection of the fast-or signals is related to the one of an even super-column (the fast-or associated to the even pixels of the first row is logically connected to the one related to the odd pixels of the third row and so on).

- 121 The local biases are generated at super-pixel level to refer the current mirrors to the local power supplies, to  
 122 reduce the mismatch due to the voltage drop on the power supplies. The FAST-OR chains are buffered once per  
 123 super-pixel.  
 124 The 4-bits flash ADC is made using 15 comparators referred to increasing, equally spaced thresholds. The base  
 125 voltage is generated locally with the same circuit used for the memory control, to reproduce the saturation value  
 126 of the analog capacitors.

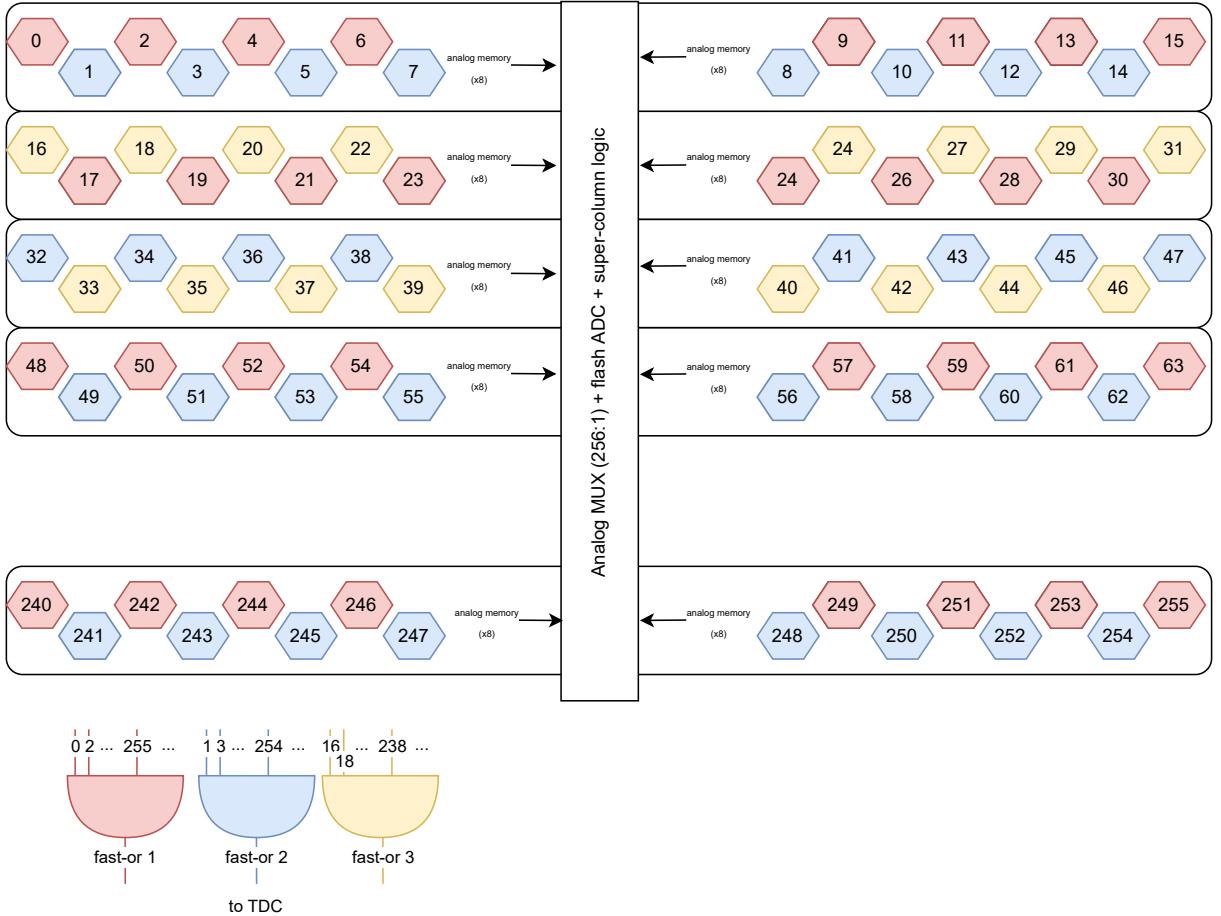


Figure 9: Row distribution of a super-pixel of an odd super-column with corresponding fast-or signals. The rectangles indicate the rows. The latter are not physically separated as in this diagram. The configuration for even super-pixel is similar and reported in Figure 10. The super-column is made of 8 super-pixels. The 24 resulting fast-or lines are sent to the TDC.

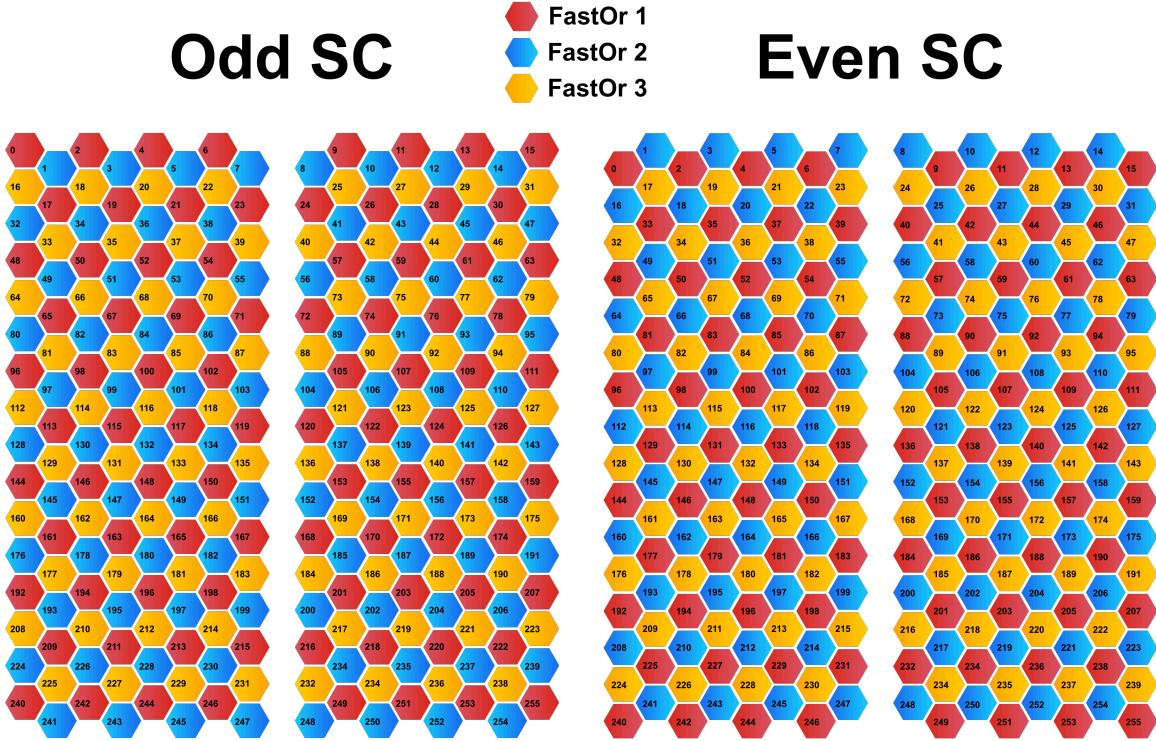


Figure 10: FAST-OR index for super-pixels of an odd/even column (for odd/even convention column counting starts from 1, from the left). Same color corresponds to the same FAST-OR line. It is important to notice that the difference between even and odd super-column is given by the necessity to stuck the super-column together avoiding dead detection area among them. [Lorenzo: Reminder for me: check with Theo if this is the map of the FAST-OR output of the chip, or if it is the FAST-OR as sent to the TDC. There is a difference between the two and I suspect that this is the output. There is no practical difference for ASIC functionality, but an operator who is using the FAST-OR for debugging may be misled by this figure.]

## 2.4 Time to digital converter

128 A time-to-digital (TDC) converter is located at the end of each super-column. It digitizes the time of arrival of  
 129 the 24 FAST-OR signals before super-pixel merging (3 FAST-OR  $\times$  8 super-pixels). The TDCs are composed  
 130 by a 7-block ring oscillator and 24 counters, all latched independently. These counters correspond to 24 readout  
 131 channels (1 per FAST-OR signal) with a 7-bit fractional data (status of the ring oscillator) and a 11-bit counter data.  
 132 The readout channels measure the time between the FAST-OR and the start of readout. The TDC also contains  
 133 1 reference channel with 7-bits fractional data and 6-bits counter data used to measure the period of two clock  
 134 cycles. Figure 11 shows the map of the interconnection between the FAST-OR lines and the TDC channels for odd  
 135 and even super-columns.

ODD COLUMNS			EVEN COLUMNS		
TDC Channel	FASTOR ID input	Super Pixel ID	TDC Channel	FASTOR ID input	Super Pixel ID
0	2	0	0	2	0
1	3	1	1	1	1
2	1	1	2	3	1
3	2	2	3	2	2
4	3	3	4	1	3
5	1	3	5	3	3
6	2	4	6	2	4
7	3	5	7	1	5
8	1	5	8	3	5
9	2	6	9	2	6
10	3	7	10	1	7
11	1	7	11	3	7
12	2	7	12	2	7
13	1	6	13	3	6
14	3	6	14	1	6
15	2	5	15	2	5
16	1	4	16	3	4
17	3	4	17	1	4
18	2	3	18	2	3
19	1	2	19	3	2
20	3	2	20	1	2
21	2	1	21	2	1
22	1	0	22	3	0
23	3	0	23	1	0

Figure 11: Map of the FAST-OR interconnection to TDC channels for even/odd super columns (for odd/even convention column counting starts from 1, from the left).

136 **2.5 Typical readout operation**

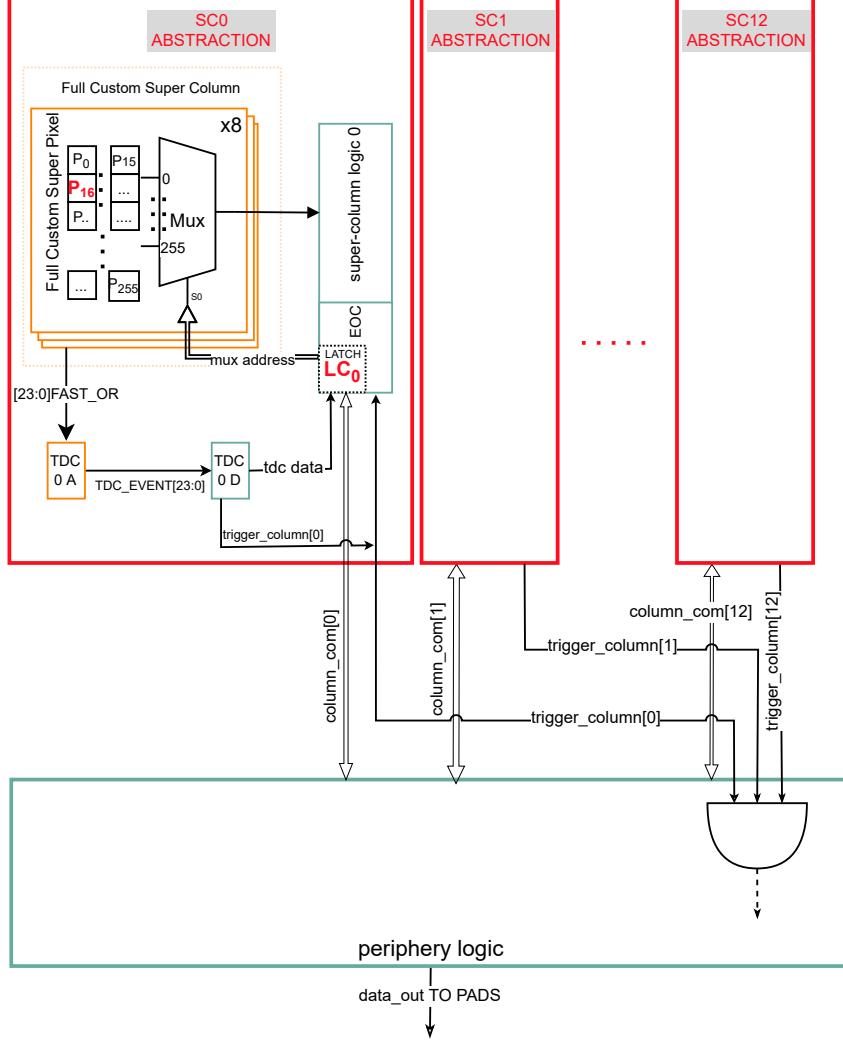


Figure 12: Simplified schematic of the FASER chip including components having a major role in the readout process. The blocks are separated depending on the ASIC implementation flow used. In green synthesized blocks, in yellow full-custom blocks. The red blocks are an abstraction level used in this picture to simplify the description of the readout. The squares in *Full Custom Super Pixel* represent the analog memories embedded in the pixels. *TDC 0 A* represents the full custom TDC, while *TDC 0 D* is the synthesized *TDC\_control*

- 137 An example of the readout process, from the pixel hit to the output data, is illustrated in this section. Figure 12  
 138 presents a simplified schematic of the blocks and interconnection that perform the readout. Let us suppose that  
 139 pixel 16 of *SC0* (in red in Figure 12) is hit by a particle. The discriminated signal will be linked asynchronously to  
 140 one of the *FAST\_OR[23:0]* that connect the *Full Custom Super Column* to the *TDC\_0\_A*. The latter will latch the  
 141 corresponding TDC channels allowing the measurement of the time of arrival. The same asynchronous signal is  
 142 forwarded to the *TDC\_0\_D* (aka *TDC\_control*) via the *TDC\_EVENT[23:0]* signal lines. The *TDC\_control* imple-  
 143 ments a second level of OR as illustrated in Figure 10 (Note: in Figure 10 *FAST-OR* corresponds to *TDC\_EVENT*).  
 144 The output of this OR is *trigger\_column[0]*, which is still asynchronous. *trigger\_column[0]* is both registered by  
 145 *super-column logic 0* (see *LC\_0* in Figure 12) and it is sent to the *periphery logic* to initiate the readout process.
- 146 The first phase of the readout process is to wait for all the analog memories to be fully charged. The length of the  
 147 waiting time corresponds to  $4 \times \text{readout\_config} + 2$  clock cycles, where *readout\_config* is a register described  
 148 in Sec. 3.2. During this time, a calibration signal and a reference signal are sent to the TDC (analog and digital).  
 149 These signals are omitted in the figure for simplicity.

150 At the end of the waiting period the analog memory  $P16$  holds a charge value that can be read by the flash-ADC  
151 in the *Full Custom Super Column* (omitted in the Figure 12).

152 After the waiting period, *periphery logic* will mask all the new possible assertions of  $trigger\_column[12 : 0]$ . As  
153 a consequence, all the hits occurring from this moment till the end of the readout will be lost.

154 The *periphery logic* will poll all the *Super Columns* (from left to right) using the communication busses indicated  
155 as  $column\_com[12 : 0]$  and send out the data to the external world. The *super-column logic X* blocks will send  
156 out ADC and TDC data only if  $LC_X$  is asserted. The readout phase is described below:

157 1. *periphery logic* sends to  $data\_out$  PAD an header containing as described in Sec.3.5.

158 2. *periphery logic* asks data from super column 0.

159 3. *Super column 0* has as latched a hit, therefore the *EOC* will send out ADC and TDC data.

160 (a) *Super column logic 0* sets the analog MUX address to 0, connecting  $P_0$  to the ADC in each super pixel.

161 (b) Pixel readout:

- 162 • Wait for *pause\_ADC* (see sec. 3.2).
- 163 • Read data from  $P_0$  in SuperPixel 0 (if there is) and transmit.
- 164 • Read data from  $P_0$  in SuperPixel 1 (if there is) and transmit.
- 165 • ... repeat until SuperPixel 7.

166 (c) *Super column logic 0* sets the analog MUX to 1, connecting  $P_1$  to the ADC in each super pixel and  
167 repeat pixel readout.

168 (d) ... continue until analog MUX is set to 255 and all pixels are read.

169 4. *Super column 0* gets the *TDC data* and sends it to the *periphery logic*, which forwards it to the output.

170 5. *periphery logic* moves to *Super column 1* and sends a header to the output.

171 6. Supposing that *Super column 1* didn't latch any pixel hit, the polling of its memories (both ADC and TDC)  
172 will be skipped.

173 7. *periphery logic* moves to next Super column, and repeat until *Super column 12* is read out.

174 **Note** The activation and duration of both  $[23:0]FAST\_OR$  and  $trigger\_column[0]$  depends on the analog front-end  
175 configuration and on the charge released by the particle.

176 100 clock cycles after the end of the readout of the last super column, the mask on  $trigger\_column[12:0]$  is released  
177 and a reset to the analog memories is sent for 20 clock cycles. The matrix is ready to detect new particle hits.

178 Figure 13 shows the order with which the ADC data are sent out from a super-column (determined by the inter-  
179 connection of the MUX with with pixel matrix).

		odd SC (SCO)												even SC (SC1)																																																																																																																																																
1023	1015	1007	999	991	983	975	967	959	951	943	935	927	919	911	903	1991	1993	1997	2007	2015	2023	2031	2039	2047	967	975	983	991	999	1007	1015	1023																																																																																																																														
959	951	943	935	927	919	911	903	895	887	879	871	863	855	847	839	1989	1991	1993	1997	1999	2007	2015	2023	2031	2047	2049	2051	2053	2055	2057	2059	2061																																																																																																																														
893	885	877	869	861	853	845	837	829	821	813	805	797	789	781	773	1799	1807	1815	1823	1831	1839	1847	1855	1863	1871	1879	1887	1895	1903	1911	1919	1927																																																																																																																														
831	823	815	807	799	791	783	775	767	759	751	743	735	727	721	713	1735	1743	1751	1759	1767	1775	1783	1791	1791	1793	1775	1767	1759	1751	1743	1735	1727																																																																																																																														
767	759	751	743	735	727	721	713	705	697	689	681	673	665	657	649	1671	1679	1687	1695	1703	1711	1719	1727	1647	655	663	671	679	687	695	703	1727																																																																																																																														
703	699	687	679	671	663	655	647	639	631	623	615	607	599	591	583	1607	1615	1623	1631	1639	1647	1655	1663	583	591	595	607	615	623	631	639	1663																																																																																																																														
639	631	623	615	607	599	591	583	575	567	559	551	543	535	527	519	1575	1587	1599	1609	1619	1629	1639	1649	519	527	535	543	551	559	567	575	1589																																																																																																																														
575	567	559	551	543	535	527	519	511	503	495	487	479	471	463	455	1479	1487	1495	1503	1511	1519	1527	1535	475	487	495	503	511	519	527	535	1535																																																																																																																														
513	505	497	489	481	473	465	457	449	441	433	425	417	409	401	393	1447	1453	1465	1463	1473	1481	1489	1497	405	413	421	429	437	445	453	461	1471																																																																																																																														
447	439	431	423	415	407	399	391	383	375	367	359	351	343	335	327	1415	1423	1431	1439	1447	1455	1463	1471	391	399	407	415	423	431	439	447	1471																																																																																																																														
383	375	367	359	351	343	335	327	319	311	303	295	287	279	271	263	1287	1295	1303	1311	1319	1327	1335	1343	263	271	279	287	295	303	311	319	1343																																																																																																																														
319	311	303	295	287	279	271	263	255	247	239	231	223	215	207	199	1287	1295	1303	1311	1319	1327	1335	1343	207	215	223	231	239	247	255	1343	1335																																																																																																																														
255	247	239	231	223	215	207	199	191	183	175	167	159	151	143	135	127	123	125	129	133	137	141	145	149	153	157	161	165	169	173	177																																																																																																																															
191	183	175	167	159	151	143	135	127	119	111	103	95	87	79	71	109	111	115	119	123	127	131	135	139	143	147	151	155	159	163	167	171	175																																																																																																																													
137	129	111	103	95	87	79	71	63	55	47	39	31	23	15	7	1031	1039	1047	1053	1063	1071	1079	1087	15	23	31	39	47	55	63	1087	1079	1071	1063	1055	1047	1039	1021																																																																																																																								
63	55	47	39	31	23	15	7	1	1031	1039	1047	1053	1063	1071	1079	1087	1087	1095	1095	1097	1097	1099	1099	1099	1099	1099	1099	1099	1099	1099	1099	1099	1099	1099	1099	1099	1099	1099	1099																																																																																																																							
1012	1014	1006	998	990	982	974	966	958	950	942	934	926	918	910	902	904	1998	2000	2014	2022	2030	2038	2046	966	972	984	990	998	1006	1014	1022	2046	2054	2062	2070	2078	2086	2094	1998																																																																																																																							
958	950	942	934	926	918	910	902	894	886	878	870	862	854	846	838	1926	1934	1942	1950	1958	1966	1974	1982	902	910	918	926	934	942	950	958	1982	1987	1994	1998	2002	2006	1998																																																																																																																								
894	886	878	870	862	854	846	838	830	822	814	806	798	790	782	774	1898	1900	1902	1905	1907	1910	1918	1926	838	846	854	862	870	878	886	894	1918	1924	1930	1936	1942	1948	1954																																																																																																																								
830	822	814	806	800	792	784	776	768	760	752	744	736	728	720	712	1880	1882	1884	1886	1888	1890	1898	1906	808	816	824	832	840	848	856	864	1884	1886	1892	1898	1904	1910	1916																																																																																																																								
776	768	760	752	744	736	728	720	712	704	696	688	680	672	664	656	1734	1742	1750	1758	1766	1774	1782	1790	710	718	726	734	742	750	758	766	1790	1798	1806	1814	1822	1830	1838																																																																																																																								
720	698	686	674	662	650	638	626	614	602	589	577	565	553	541	529	1670	1678	1686	1694	1702	1710	1718	1726	646	654	662	670	678	686	694	1726	1734	1742	1750	1758	1766	1774																																																																																																																									
674	666	658	650	642	634	626	618	610	602	594	586	578	570	562	554	1624	1632	1640	1648	1656	1664	1672	1680	573	581	589	597	605	613	621	629	1661	1669	1677	1685	1693	1698	1706																																																																																																																								
616	598	580	562	544	526	508	490	472	454	436	418	400	382	364	346	1598	1606	1614	1622	1630	1638	1646	1654	598	606	614	622	630	638	646	1654	1662	1670	1678	1686	1694	1702																																																																																																																									
564	546	528	510	492	474	456	438	420	402	384	366	348	330	312	294	1541	1549	1557	1565	1573	1581	1589	1597	541	553	561	571	581	591	601	1597	1605	1613	1621	1629	1637	1645																																																																																																																									
508	490	482	474	466	458	450	442	434	426	418	410	402	394	386	378	1540	1548	1556	1564	1572	1580	1588	1596	512	524	532	540	548	556	564	572	1596	1604	1612	1620	1628	1636	1644																																																																																																																								
448	430	422	414	406	398	390	382	374	366	358	350	342	334	326	318	1530	1538	1546	1554	1562	1570	1578	1586	324	336	348	360	372	384	396	1536	1544	1552	1560	1568	1576	1584																																																																																																																									
388	370	362	354	346	338	330	322	314	306	298	290	282	274	266	258	1532	1540	1548	1556	1564	1572	1580	1588	326	338	350	362	374	386	398	1532	1540	1548	1556	1564	1572	1580																																																																																																																									
330	309	293	285	277	269	261	253	245	237	229	221	213	205	197	189	181	173	175	177	179	181	183	185	187	189	191	193	195	197	199	201	203	199																																																																																																																													
275	233	225	217	209	201	193	185	177	169	161	153	145	137	129	121	123	125	127	129	131	133	135	137	139	141	143	145	147	149	151	153	155	157	159	161	163	165	167	169	171	173	175	177	179	181	183	185	187	189	191	193	195	197	199	201	203																																																																																																						
217	209	201	193	185	177	169	161	153	145	137	129	121	113	105	97	89	81	73	65	57	49	41	33	25	17	9	1	2019	2026	2033	2040	2047	2054	2061	2068	2075	2082	2089	2096	2103	2110	2117	2124	2131	2138	2145	2152	2159	2166	2173	2180	2187	2194	2198	2205	2212	2219	2226	2233	2240	2247	2254	2261	2268	2275	2282	2289	2296	2293	2300	2307	2314	2321	2328	2335	2342	2349	2356	2363	2370	2377	2384	2391	2398	2405	2412	2419	2426	2433	2440	2447	2454	2461	2468	2475	2482	2489	2496	2503	2510	2517	2524	2531	2538	2545	2552	2559	2566	2573	2580	2587	2594	2601	2608	2615	2622	2629	2636	2643	2650	2657	2664	2671	2678	2685	2692	2699	2706	2713	2720	2727	2734	2741	2748	2755	2762	2769	2776	2783	2790	2797	2804	2811	2818	2825	2832	2839	2846	2853	2860	2867	2874	2881	2888	2895	2902	2909	2916

180 **2.6 Digital logic implementation**

181 This section includes implementation details. The normal operator can skip this section.

182 In this section a description of the digital logic of the FASER final ASIC is provided. As anticipated, the digital  
183 electronics is divided in 13 super-column logic systems and the periphery. There are two different clock domains:

- 184 • readout clock (rclk). Nominal frequency of 200 MHz, used to read the value of TDC and analog memories  
185 during readout phase. The ouput data is sent out with this clock.
- 186 • programming clock (pclk). Nominal frequency of 20 MHz, used for the programming phase of the ASIC  
187 (biasing and masking). More details are given in Section 3.1.

188 The super-column logic is used to collect data from the matrix during readout and send them to the periphery  
189 logic. Moreover, it contains the registers for pixel masking (and consequently masking) and distributes the pulsing  
190 signals.

191 The super-column logic is composed of 8 super-pixel blocks and the end-of-column (EOC) system. Each super-  
192 pixel block communicates with the corresponding super-pixel to:

- 193 • send the masking data to the pixels during programming phase.
- 194 • perform the polling of the input of the ADC: during readout, the super-pixel changes periodically the address  
195 of the 256-to-1 analog mux used to connect the analog memory of each pixel to the input of the ADC.
- 196 • acquire the data provided by the ADC and perform data pruning (see Section 3.5).

197 The super-pixel blocks are also used to propagate both of the rclk and pclk signals through all over the super-  
198 column.

199 The EOC block acts as an interface between the periphery logic and the rest of the super-column. The system  
200 sends the column data to the periphery (including the data collected from the TDC which communicates directly  
201 with the EOC) during readout.

202 **2.6.1 Periphery**

203 In Figure 14 a schematic of the periphery logic is reported. This block represents the main digital system of the  
204 ASIC and handles the communication between the I/O pads of the chip and the rest of the matrix. As for the super-  
205 column logic, it works with two main clock domains: rclk (reported in red in Figure 14) and pclk (programming  
206 clock or SPI clock, reported in blue in Figure 14).

207 The periphery logic is organized in 4 main sub-blocks. Their description is reported below. This are implementa-  
208 tion details which the normal operator can skip.

- 209 • **readout control.** it receives the FAST-OR inputs forwarded by the TDC\_control, initialize the readout and  
210 collect the data produced by the columns. These data are then sent to I/O pads during readout together with  
211 the *trigger\_slow* signal that can be read during the testing phases. The *trigger\_slow* is a synchronized version  
212 of the FAST-OR, which stays high for a number of clock cycles corresponding to (*config\_delay* \* 4 + 2).  
213 This block is synchronized on the rclk (200 MHz).
- 214 • **periphery control.** This sub-system aims to organize the programming phase of the ASIC. For this rea-  
215 son it is synchronized with pclk (20 MHz). The latter can be indicated as SPI clk since the slow control  
216 communication is based on a SPI-like bus (more details on the slow control in Section 3.1).
- 217 • **reg control.** The configuration bits (6 bit ADDRESS and 8 bit DATA) received by the periphery control are  
218 sent to the end register, described by the ADDRESS by this block.
- 219 • **test-pulse generator.** This system generates a test-pulse to be sent to the matrix. The test-pulse signal is  
220 synchronized on a divided version of the rclk (the division factor is 4096). The test-pulse delay is set during  
221 the programming phase.

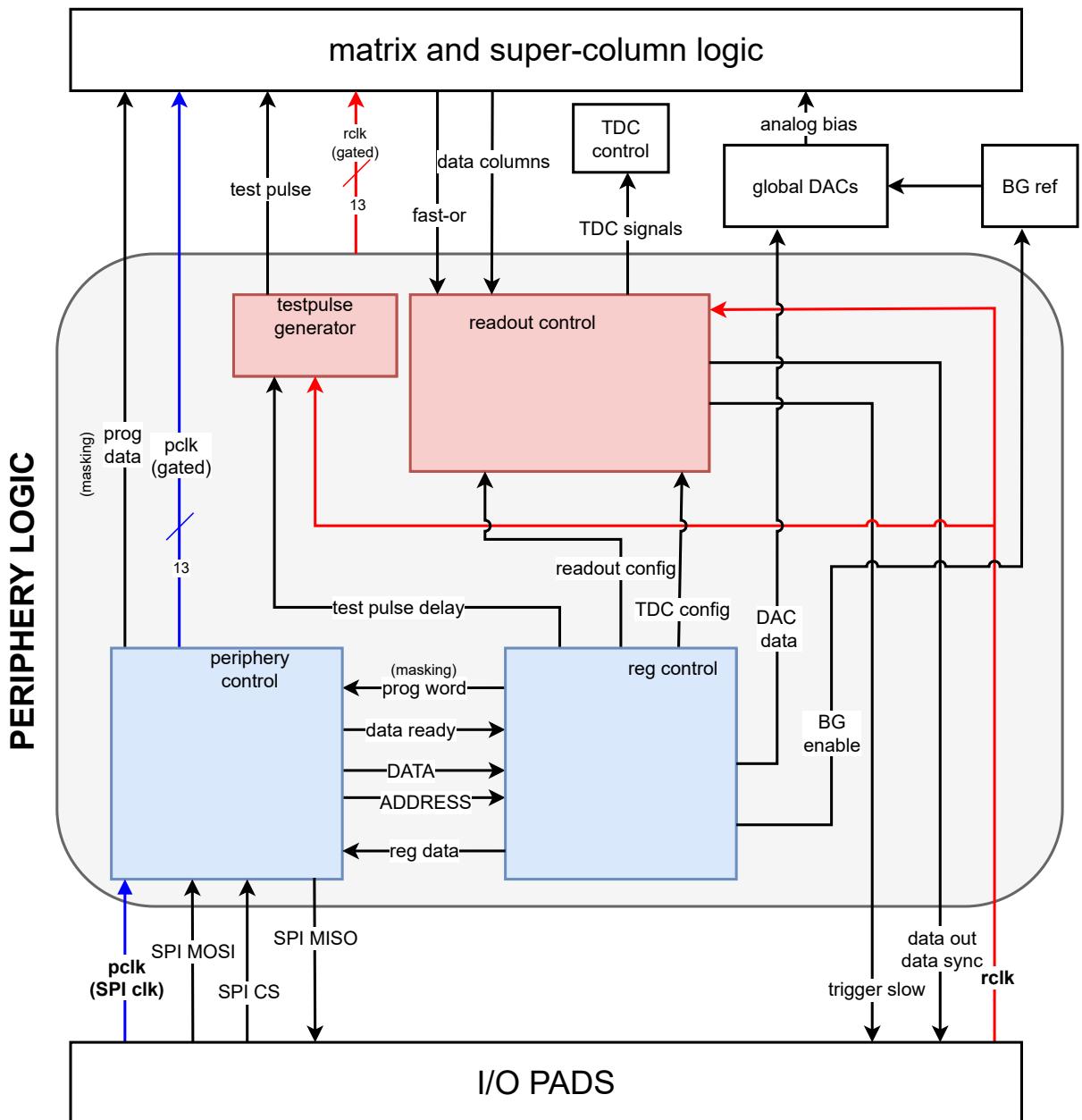


Figure 14: Scheme of the periphery logic. Red lines correspond to the fast readout clock (rclk); blue lines to the slow programming clock (pclk). For simplicity the reset logic is omitted.

222 • **reset\_control** Is the block that generate the resets.

223 **2.6.2 TDC control logic**

224 The TDC control logic is a digital system that generates the signals that the TDC needs in order to measure the  
225 time intervals for event reconstruction. The ASIC is featuring one TDC per super-column and each of them is  
226 driven by their own TDC control logic.

227 An event bus of 24 lines (one per TDC channel) is used by the this logic to produce the latching signals for the  
228 converter. The TDC is based on a free-running ring-oscillator connected to a set of D-latches (that are in transparent  
229 mode when their gating signal is set to '1'). Therefore, it needs to be calibrated every time an event occurs. The  
230 TDC control logic is able to generate the latching signals for the calibration of the ring-oscillator when the TDC

Signal	Hierarchy level	Description
SPI_MOSI	Periphery I/O	Programming signal from the FPGA to the chip
SPI_MISO	Periphery I/O	Programming signal from the chip to the FPGA
SPI_CS	Periphery I/O	Chip select signal
prog_data	Periphery I/O	Masking data to the masking shift register
pclk (gated)	Periphery I/O	Gated programming clock (20 MHz) sent to one column at a time to mask it during configuration
prog_word	reg_control to periphery	8-bit signal with masking data
reg_data	reg_control to periphery	8-bit signal with configuration register data (not masking)
data_ready	periphery reg_control to	Enable signal to latch the DATA in the register corresponding to ADDRESS
DATA	periphery reg_control to	8-bit signal with programming data to be stored in the corresponding register
ADDRESS	periphery reg_control to	Encoded address of the register in which DATA must be stored
DAC_data	Periphery I/O	8-bit signal with DAC configuration data
BG_enable	Periphery I/O	Enable signal for band-gap reference analog circuit. It corresponds to bandgap_config[1] register bit
testpulse_delay	reg_control to testpulse_generator	Configuration signal for testpulse duration
testpulse	Periphery I/O	1-bit testpulse signal
relk (gated)	Periphery I/O	Gated readout clock (200 MHz) sent to one column at a time during readout
fast-or	Periphery I/O	FAST-OR signals from the chip. These have already been compressed in the TDC_control logic, they are one per column here (act as a hit flag for the columns)
data_columns	Periphery I/O	Serial data out from columns to readout_control
TDC_signals	Periphery I/O	TDC reference and calibration signals to TDC_control component
readout_config	reg_control to readout_control	Delay value from active FAST-OR edge to readout beginning
TDC_config	reg_control to readout_control	Delay value from active FAST-OR edge to readout beginning
trigger_slow	Periphery I/O	Synchronized FAST-OR signal generated in readout_control
data_out	Periphery I/O	Serial data out coming from readout_control

Table 2: Short description of the main signals in the periphery logic, as they are shown in Figure 14

<sup>231</sup> reference and TDC calibration signals are asserted by the readout control logic in the periphery (in Figure 14 they  
<sup>232</sup> are reported as 'TDC signals'). Figure 15 shows the signals generated by the periphery logic as well as the ranges  
<sup>233</sup> that can be used to configure the delays.

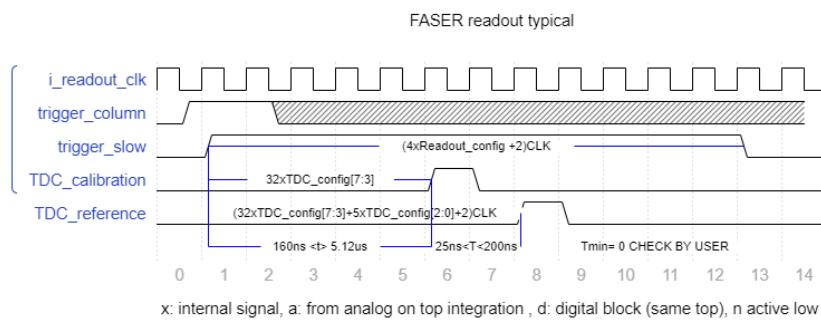
<sup>234</sup> [Brian: It is also not clear to me how this works and how the reference is set across the 13 SC wrt to the first hit?]

## SPECS

delta T on rising edge	min	max	step	bit
trigger_column ->TDC_reference	160 ns	5.12 us	160ns (32*clk)	5
TDC_calibration ->TDC_reference	25 ns	200 ns	25ns (5*clk)	3

Both *TDC\_reference* and *TDC\_calibration* need to arrive before or at the same time of falling edge of *trigger\_slow* (set by *readout\_config*)

## Timing Diagram



NB: Case with *TDC\_Calibration* = 0 : *TDC\_calibration* is generated at the clock after *trigger\_slow*. *TDC\_reference* will come one clock cycle after.

Figure 15: Timing diagram for calibration latching signals generation.

## 3 User and implementation guide

This section is meant for implementation of the ASIC in the new FASER preshower.

### 3.1 SLOW CONTROL

Slow control uses an SPI-like interface, with four single-ended lines, SPI clk (pclk), SPI\_CS (chip select, CS in Figure 14), SPI MOSI (data) and SPI MISO. Command words are always 20 bits.

- Chip select is active low, needs to be low for the duration of the command.
- Clock is only sent during the command.
- Clock polarity is 0 (0 when idle).
- Data changes during the falling edge of the clock (CPHA=0).

When the slow control bus is not used, the clock can be gated. Only 20 clock pulses needs to be sent (one per data bit) and the clock needs to be 0 when idle. During data transmission the Chip Select line needs to be low to indicate that the communication is targeted at that chip. This helps connecting multiple chips in parallel by sharing clock and data lines, although there is also a chip ID field that allows addressing specific chips.

Data are sampled by the clock on the rising edge. A more detailed timing diagram is shown in Figure 16 and 18:

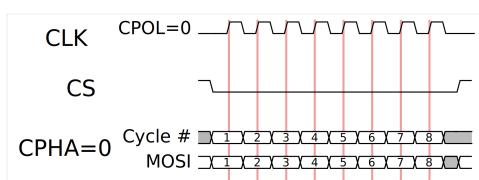


Figure 16: Timing diagram of SPI low control.

The SPI signals are 1.2 CMOS (full-swing). The system is designed and simulated to work at 20 MHz. The commands follow the structure reported in Figure 17:

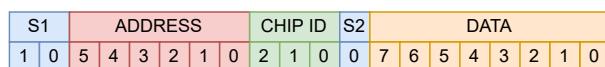


Figure 17: Command structure of SPI interface. Data are sent from left to right, MSB first, i.e. S1[1] is sent first, then S1[0], ADDRESS[5], ADDRESS[4] ... until DATA[0]

Spare bit S1 b1 b0	Command	Note
00	Writing DATA through SPI	input DATA is written in a register defined by ADDRESS
11	RESET	See Section 3.3
10	masking programming	push masking words in super-column, see Section 3.4
01	readback	SPI MISO readback the data from the register

Table 3: Spare bits S1 and associated functions. Refer to Figure 17 for details about SPI command.

The chip ID is a 3-bits value which is hardcoded via wirebonds. Every command checks the chip ID with the hardcoded ID of the chip. If it doesn't match, the command is ignored.

Table 3 shows a summary of the possible functions that can be selected with S1:

- 254     • If S1 is '11', a reset is asserted. The rest of the command is ignored (in fact, the reset is issued before the  
 255        data is even sent). This reset issues the spi\_reset described in Section 3.3. NB it does not reset the columns  
 256        neither the configuration bit.

- 257     • If S1 is '00', the 8-bits data word is written in an 8-bits periphery register, chosen according to the command  
 258        address. The list of valid addresses is reported in table 4 The MISO will send out the previously stored data.  
 259        All DACs are 8-bits, so they can be programmed with a single SPI command each.

- 260     • If S1 is '10', push masking words in super-column, see Section 3.4.

- 261     • If S1 is '01', SPI MISO readback the data from the register.

262     **N.B.:** the SPI-like interface is independent on the readout logic. This means, for example, that the chip can be  
 263        programmed and configured also without the readout clock.

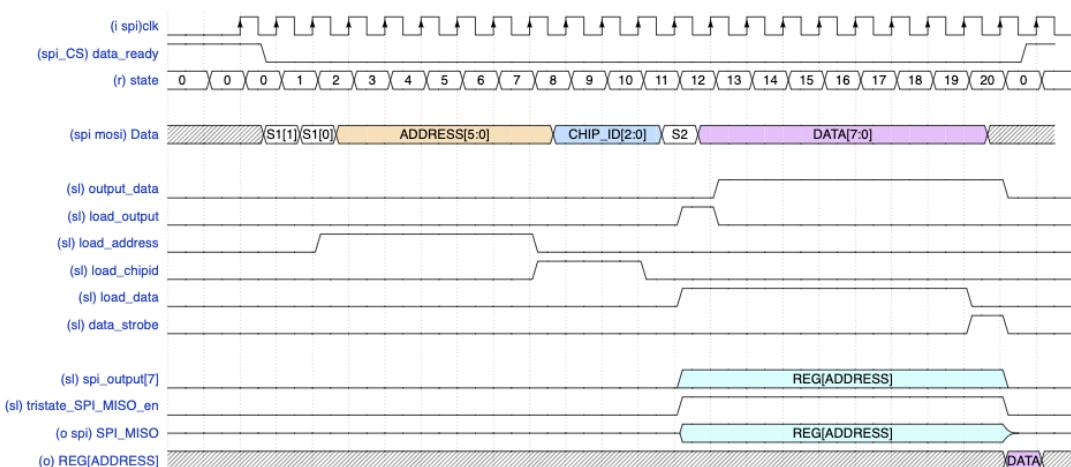


Figure 18: Detailed timing diagram of SPI interface. *sl* signals are internal signals (operator should ignore). Interface signals needed to operate are : (i\_spi) clk , (spi\_CS) dataready, (spi mosi) data, (o\_spi) SPI\_MISO

Table 4: List of slow control addresses. LSB value is reported for the linear region, more details in Section 4

<b>Address b5..b0</b>	<b>Register</b>	<b>Description</b>	<b>Range</b>	<b>LBS value</b>	<b>DAC setting at startup</b>
000101	bias_preamp	Control of the bias current of the BJT-based preamplifier.	0 $\div$ 13 $\mu\text{A}$	0.051 $\mu\text{A}$	2
000110	bias_feedback	Control of the impedance of the feedback of the preamplifier.	0 $\div$ 110 nA	0.43 nA	20
000111	bias_disc	Control of the bias current of the discriminator	0 $\div$ 18.5 $\mu\text{A}$	0.23 $\mu\text{A}$	2
001000	bias_idle	Control of the idle current of the analog memory circuit	0 $\div$ 1.4 nA	0.0075 nA	5
001001	bias_LVDS	Control of the voltage swing of the LVDS drivers	0 $\div$ 20.4 mA	1.6 mA	1
001010	bias_load	Control of the current that charges the analog memories	0 $\div$ 5.5 $\mu\text{A}$	0.03 $\mu\text{A}$	5
001011	bandgap_config	* see Section 3.2	-	-	64
001100	bias_testpulse	Amplitude of the pulse generated by the test pulse injection system.	0 V $\div$ 0.77 V	-	0
001101	threshold_set	8 LSBs for the threshold	1.2 V $\div$ 0.43 V	700 mV	0
011101	threshold_offset	9nth additional bit for the threshold	1.2 V $\div$ 0.43 V	700 mV	0
010011	bias_pixel	Controls the bias current of the pixel nwell.	0 $\div$ 40.3 pA	10 pA	100
110010	testpulse_delay	Low time of the testpulse period (controls duty cycle)			0
011110	config_global	* see Section 3.2	-	-	0
011111	readout_config	* see Section 3.2	-	-	0
000010	programming_word	* see Section 3.4	-	-	0
000011	TDC_config	* see Section 3.2	-	-	0

## 3.2 CONFIGURATION PROGRAMMING

Most of the configuration registers are directly connected to 8-bits DACs to set analog biases. Some of the registers are however used for the digital configuration of the chip. **All of these enable or disable signals are active high.** They are described in Table 5.

Table 5: Digital configuration registers description.

### Config\_global

Bit 7	6	5	4	3	2	1	0
Delay before ADC sampling ( <i>pause_adc</i> )	Enable override for super-column 9,10,11,12 TDC	Enable override for super-column 6,7,8 TDC	Enable override for super-column 3,4,5 TDC	Enable override for super-column 0,1,2 TDC	Enable second LVDS output		

The override of the TDC indicates the possibility to activate a power gating system for the ring-oscillator based TDC that was designed to reduced the steady state power consumption of the converter.

### Readout\_config

Bit 7	6	5	4	3	2	1	0
Delay between the trigger and the automatic start of readout in units of 2 clock cycles. See Section 3.5.							

### TDC\_config

Bit 7	6	5	4	3	2	1	0
Delay between the trigger and the TDC reference signal. NOTE: it should be smaller than readout_config.							

### Bandgap\_config

Bit 7	6	5	4	3	2	1	0
DON'T CARE	en_din_synch	en_dout_osc	DON'T CARE	config reset test-pulse	DON'T CARE	Bandagap enable_n	Enable testpulse

- The various DACs have different ranges so that the nominal biases correspond to a code somewhat in the middle of their dynamics. The reset value is reported.
- Bandgap\_config* bits are used to enable/disable the bandgap (effectively turning on/off all the biases of the chip), the testpulse system and to configure the various reset configurations of the ASIC. More details on the reset are reported in Section 3.3.
- N.B:** *bandgap\_enable\_n* is **active low** (when 0, the bandgap circuit is active).
- The registers *Bandgap\_config[6 : 5]* are respectively used to enable the synchronization of signals from the column (*en\_din\_synch*) and to send an oscillating signal in the DOUT pin (with frequency equal to 1/4 the clock frequency) to tune the phase between data and clock in the DAQ *en\_dout\_osc*. The block diagram related to the use of these two registers is n Fig. 19.

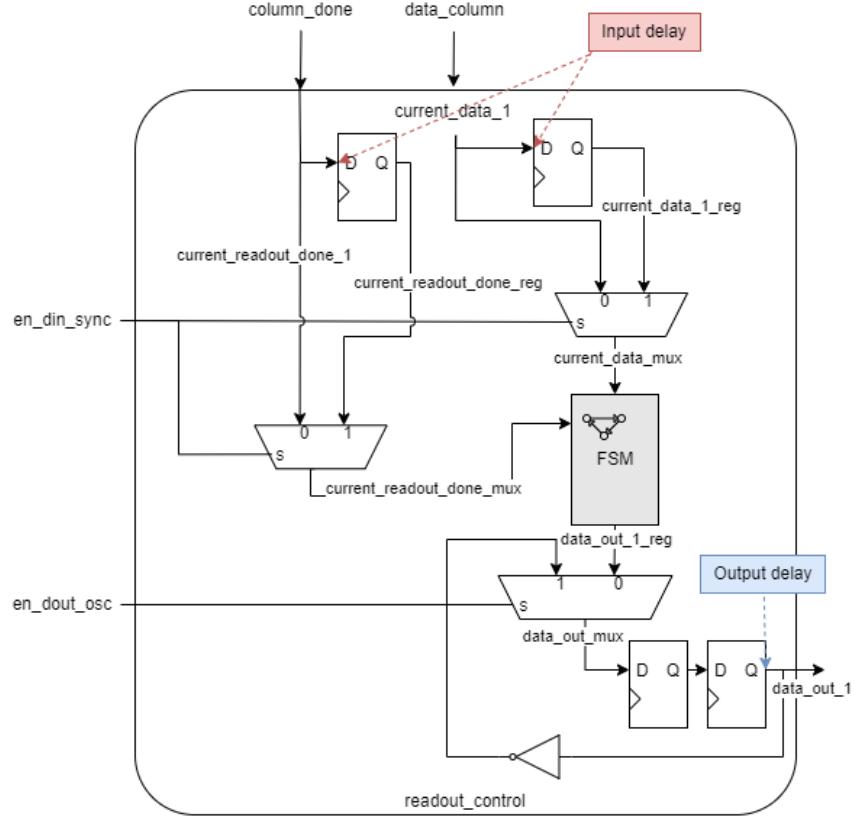


Figure 19: Block diagram describing the implementation of the synchronization of signals from the column, and the implementation of the oscillating signal of in dout pin.

### 278 3.3 RESET

279 The reset can be issued using 3 inputs: the reset\_n pad, the reset\_synth\_n pad and the SPI reset (see Section 3.2)  
 280 Several reset modalities have been implemented in the chip:

- 281 • **Global (power-on reset)**. It resets all the state machines and configuration registers to the default value.  
 282 Use the pad *rst\_n*. It needs the configuration register config\_rst\_testpulse (3rd bit of bandgap config) to be  
 283 low.
- 284 • **memory only reset** resets the matrix analog matrix. It does not happen if a readout is occurring ( in the  
 285 timing diagrams see signal *busy<sub>readout</sub>*). NB also the supercolumn state is reset. To issue this reset, the  
 286 reset\_synth\_n should be kept hold for at least 4 readout clock cycle and realeased before 8 clock cycles. It  
 287 needs the configuration register config\_rst\_testpulse (4th bit of bandgap config) to be low.
- 288 • **flush reset**. It resets the readout discarding data even if a readout is occurring. To issue this reset, the  
 289 reset\_synth\_n should be kept hold for at least 8 readout clock cycles.
- 290 • **testpulse reset** It stops sending testpulses. It is issued asserting reset\_synth\_n and it need the configuration  
 291 register config\_rst\_testpulse (4th bit of bandgap config) to be high.
- 292 • **reset via spi (reset\_soft)** A reset can be issued with spi spare bits S1=11 as stated in Sec. 3.1 This function  
 293 resets ONLY the readout in the periphery and does not delete data from the columns. It is equivalent to the  
 294 one present in the pre production prototype and should not be used under normal condition.

295 The timing diagrams of the reset modalities are reported in the appendix (Sec. 5.1). The FSM controlling the three  
 296 synchronous resets (flush, memory and testpulse) is reported in Sec. 5.2. A schematic showing the equivalent logic

297 generating these three reset is shown in Fig. 20. This circuit is not the one physically implemented on the ASIC,  
298 but shows the equivalent functionalities for an easier understanding.

299 The timing diagram of the power on is reported in Fig 21

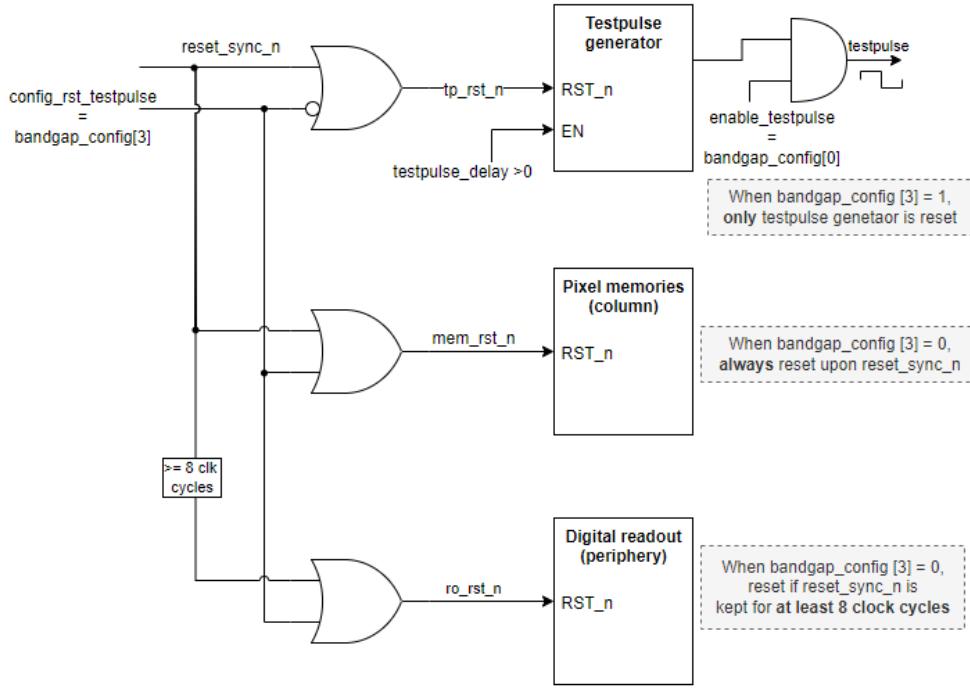


Figure 20: Sync reset logic abstraction. This circuit reproduces some of the functionalities of the FSM controlling the generation of the three synchronous resets (flush, memory and testpulse) and explains how these relate to each other. A complete diagram of the FSM is reported in Sec. 5.2

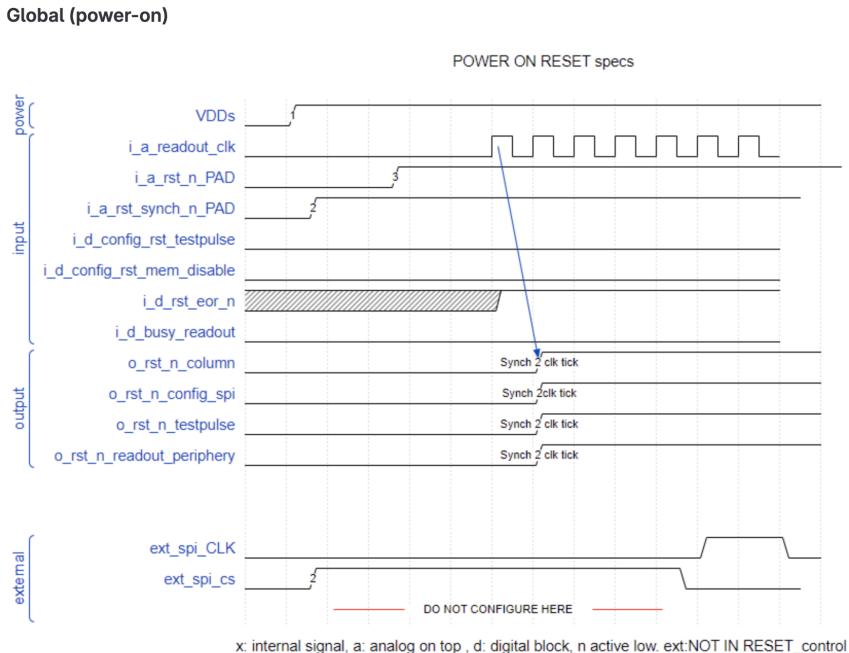


Figure 21: Power-ON timing diagram.

300 **N.B.:** Both the reset\_n\_sync and the reset\_n (global reset) need the readout clock signal to be sent **after** de-  
301 assertion, to effectively release the reset inside the chip. This means, for example, that after power-up, the chip is  
302 in a global reset state and can not even be configured, until the clock signal is sent. The reset de-assertion inside  
303 the chip, takes two readout clock cycles, so after this time the chip is not in reset state anymore, and the readout  
304 clock can be stopped if needed.

## 305 3.4 PIXEL PROGRAMMING

306 Programming of the pixel is done by sending a stream of data to the super-columns. Each super-column features a  
307 long shift register, one bit per pixel, storing the mask bit status (if 1, the pixel is masked). In order to program the  
308 entire super-column, a total of 2048-bits needs to be written (this is done with 512 SPI commands). The procedure  
309 to do this is to write 8-bits of data to a temporary register called programming\_word (code '00010'). This data is  
310 then shifted in a selected super-column by sending a special SPI command with S1 = '10'. The super-column to  
311 send the data to is chosen according to S2 bit and the 3 LSb of ADDRESS field. More in detail, the super-columns  
312 are coded on 4-bits addresses given by SC\_add={S2, ADDRESS[2:0]}.

313 Since the chip has 13 super-columns and each requires 512 SPI commands to be programmed, a full programming  
314 takes 6656 SPI commands. Note that the mask bits are not reset at power-on. It is thus necessary to program the chip after power-on to

315 For example, to push a sequence of 8-bits in the super-column 5 (SC\_add='0101') of a chip (in red) with ID '111'  
316 (in green), two consecutive SPI commands are needed (remember, programming\_word register is associated to the  
317 ADDRESS='000010':

318 00 000010 111 0 DDDDDDDD ← write 8-bits word

319 10 XXX101 111 0 XXXXXXXX ← push programmed word in super-column 5

320 This sequence will be repeated 256 times until the entire 2048-bits word is pushed in the super-column. The Xs in  
321 the commands don't matter for the purpose of the programming.

## 322 3.5 READOUT

323 After a configurable delay the chip reads out its content using one or both LVDS outputs. The digital logic was  
324 designed to run at 200 MHz, so each output provides a bandwidth of 200 Mbps in single data rate. The readout  
325 doesn't require any synchronization between input signals, as the chip will just send data at the same frequency as  
326 the input clock. The delay within the chip between the input clock and the output data lines is difficult to estimate  
327 accurately, the design targets a value below 1ns.

328 In Fig 22 is shown a schematic of the expected data stream for a readout. Bits in red are fixed and should be used  
329 as a reference for decoding data.

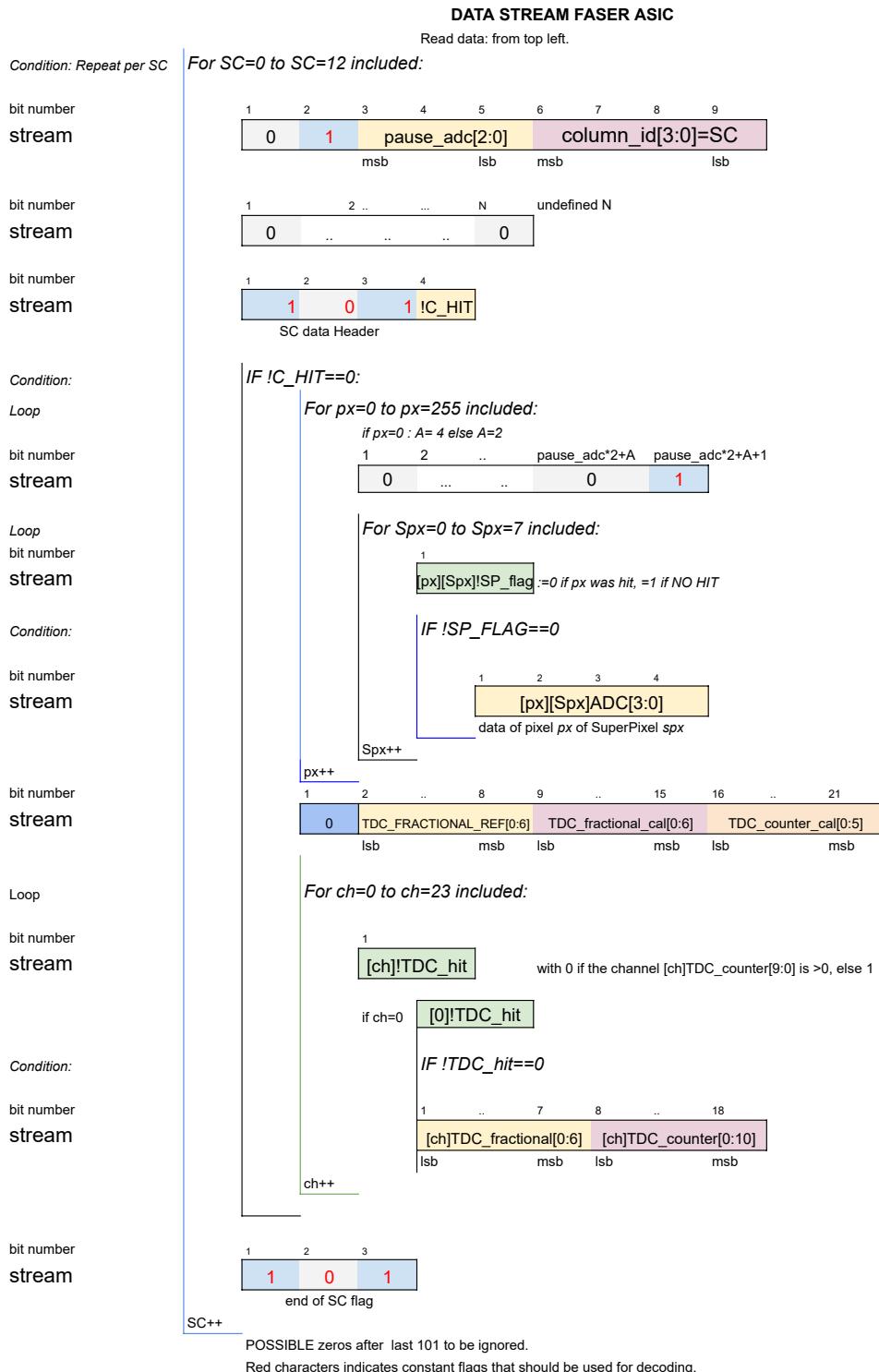


Figure 22: The expected data stream from a readout of the FASER ASIC. SC: super column. C\_Hit flags of column hit. px : pixel. Spx : super pixel. SP\_flag : flag of superpixel hit. ch: channel of TDC. **Known behaviour** TDC\_hit is sent twice in case of ch=0.

### 3.6 Testpulse

The test pulse is internally generated. To be enabled the register Enable testpulse must be active high. The expected operation of the testpulse is shown in Fig. 23. The testpulse generate a signal with a fixed time high ( $4096 \times 3 \times \text{readout\_clock\_period} = 61440\text{ns}$  with a readout\_clock of 200MHz). The duration of the time low is set with the config 8 bit register *TESTPULSE\_DELAY*. When this register is set to zero, no tespulse is generated.

A typical testpulse scan should hold active low the *reset\_n\_PAD\_synch* while configuring the chip (i.e. : DACs, pixel programming and tespulse duration). After config is complete, the release of *reset\_n\_PAD\_synch* will start the generation of the *tespulse\_in* signal. The injection of charge in the pixels is performed on the rising edge of the signal *tespulse\_in*.

**Known and waived bugs:** configuring *testpulse\_delay*=255 and 254 will reduce the time high of the pulse respectively to  $4096 \times 1 \times \text{readout\_clock\_period} = 40\mu\text{s}$  and  $4096 \times 2 \times \text{readout\_clock\_period} = 20\mu\text{s}$ , due to internal overflow.

**NB:** Enabling the testpulse prior to the *reset\_n\_PAD\_synch* active low assertion will lead to unkown behavior.

Table 6: Specification of the internal testpulse generator. For time value a readout\_clock of 200MHz has been considered

SPEC	RANGE
Time between rise and fall edge (Thigh)	$3 \times 4096 \times \text{CLK\_T} (61'440 \text{ ns}) *$
Time between fall and rise (Tlow)	(20 480 ns min / 5 222 400 ns max )
Time Between consecutive injections:	Tlow +Thigh
testpulse_delay step	$4096 \times \text{CLK\_T} (20 480\text{ns})$

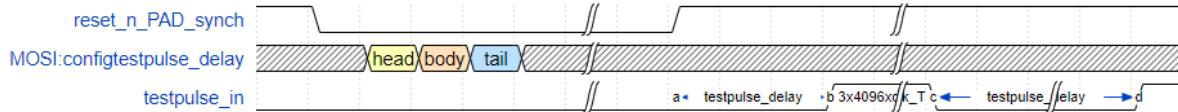


Figure 23: Typical operation of the tespulse. *rst\_n\_PAD\_synch* is asserted active low during configuration (represented with MOSI in the figure). *testpulse\_in* is generated and injects charge in the pixel on his rising edge.

#### 3.6.1 Testpulse synchronization on multiple ASICs

On a FASER module housing six ASICs, it could be necessary to synchronize the testpulse in all of them for debug purposes. To do this, the synchronous reset must be employed as shown in the waveform in Fig. 24 where the procedure is reproduced for four ASICs.

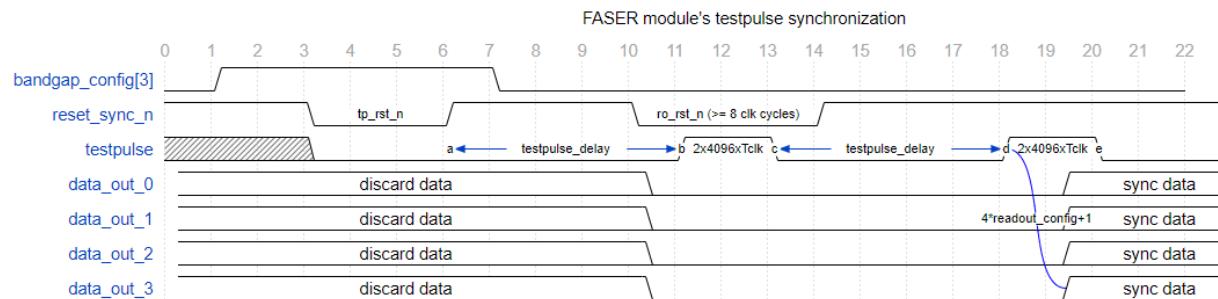


Figure 24: Required procedure to start the testpulse synchronously on every chip of a module. In this case only the data from 4 ASIC are shown.

The first step would be to enable the reset testpulse mode through SPI configuration on every chip and then assert the synchronous reset. With the common de-assertion of this reset (edge *a* in Fig 20) to all the chips, all the testpulse generators of every ASIC will start at the same time, ensuring the synchronization. At this point it is

350 possible to exit the testpulse reset mode de-asserting the bandgap\_config[3] bit.  
351 During all this time, the readout is not reset and might continuously send data to the output or get the readout  
352 machine in a continuous retriggering. This data should be discarded, and to start a new, clean readout with synch  
353 pulses, a **flush reset** is required (Sec. 3.3).  
  
354 When the *reset\_synth\_n* in the **flush reset mode** is de-asserted, a new synchronous readout on every chip starts  
355 after the next testpulse injection (edge *d*).  
  
356 **N.B.** While the testpulse reset mode is enabled (*bandgap\_config*[3]=1) the reset is not sent to the readout, there-  
357 fore the DAQ MUST NOT implement any automatic reset at the end of the readout, since this will be in conflict  
358 with the synchronization of the testpulse. Moreover, before sending the flush reset (*ro\_rst\_n* in the diagram), the  
359 *bandgap\_config*[3] must be de-asserted on **every chip**. This is mandatory, otherwise the *reset\_sync\_n* might reset  
360 the readout in some ASIC, and the testpulse in some others where the *bandgap\_config* still has not changed. This  
361 would cause the complete loss of synchronization of the pulses achieved so far.

## 362 3.7 PINOUT

363 The pads are on one side and going left to right (with the wirebonds at the bottom), with a 150 µm pitch. The pads  
364 are described in Figure 25.  
  
365 **NOTE:** RESET\_N PAD is a CMOS input used for the power-on reset, but on the flex it can be connected to  
366 VDD\_DIG via a RC circuit to create a slow rise time at the input with a RC value between 2 ms and 20 ms, with  
367 larger values preferable, where possible. The values for the resistance and capacitance can be selected according  
368 to the size of the component and space available on the flex.

N.	Name	Type	Flex net	Flex net	N.	Name	Type	Flex net	Flex net	N.	Name	Type	Flex net
1	GND_DISC	ground	GND	VDD_DIG	30	VDD_DIG	power	GND	ground	88	GND_A	ground	GND
2	VDD_DISC	power	VDD_DIG	VDD_DIG	31	VDD_OSC	power	VCC_A	power	89	VCC_A	power	VDD_DIG
3	VDD_OSC	power	VDD_DIG	VDD_DIG	32	VSS	ground	GND	ground	90	GND_DIG	ground	VDD_DIG
4	VSS	ground	GND	VDD_DIG	33	VDD	power	VDD_OSC	power	91	VDD_DIG	power	VDD_DIG
5	VDD	power	VDD_DIG	VDD_DIG	34	GND_A	ground	GND	ground	92	VDD_OSC	power	VDD_DIG
6	GND_A	ground	GND	VCC_A	35	VCC_A	power	VCC_A	power	93	VSS	ground	GND
7	VCC_A	power	VCC_A	GND	36	GND_DIG	ground	GND	power	94	RESET_N	CMOS_IN	RESET<1>
8	GND_DIG	ground	GND	VDD_DIG	37	VDD_DIG	power	GND	power	95	RESET_N_sync	CMOS_IN	VCC_A
9	VDD_DIG	power	VDD_DIG	VDD_DIG	38	VDD_DIG	power	VCC_A	power	96	VDD	power	VDD_DIG
10	VDD_OSC	power	VDD_DIG	VDD_DIG	39	VSS	ground	GND	ground	97	GND_A	ground	VDD_DIG
11	VSS	ground	GND	VDD_DIG	40	TRIG_F_3+	LVDS_OUT	N/A	power	98	VCC_A	power	VDD_DIG
12	VDD	power	VDD_DIG	VDD_DIG	41	TRIG_F_3	LVDS_OUT	N/A	power	99	GND_DIG	ground	VDD_DIG
13	GND_A	ground	GND	VDD_DIG	42	VDD	power	VDD_OSC	power	100	VDD_DIG	power	VDD_DIG
14	VCC_A	power	VCC_A	GND	43	GND_A	ground	GND	ground	101	VDD_OSC	power	VDD_DIG
15	GND_DIG	ground	GND	VCC_A	44	VCC_A	power	VCC_A	power	102	VSS	ground	GND
16	VDD_DIG	power	VDD_DIG	VDD_DIG	45	GND_DIG	ground	GND	LVDS_OUT	103	SPI_MOSI	CMOS_IN	THRESHOLD
17	VDD_OSC	power	VDD_DIG	VDD_DIG	46	VDD_DIG	power	VDD_DIG	LVDS_IN	104	SPI_MISO	CMOS_OUT	VCC_A
18	VSS	ground	GND	VDD_DIG	47	VDD	power	VDD_DIG	LVDS_IN	105	SPI_CLK	CMOS_IN	VDD_DIG
19	VDD	power	VDD_DIG	VDD_DIG	48	VSS	ground	GND	power	106	SPI_CS	CMOS_IN	VDD_DIG
20	GND_A	ground	GND	GND	49	TRIG_F_2+	LVDS_OUT	N/A	power	107	VDD	power	VDD_DIG
21	VCC_A	power	VCC_A	VCC_A	50	TRIG_F_2	LVDS_OUT	N/A	ground	108	GND_A	ground	VCC_A
22	GND_DIG	ground	GND	VDD_DIG	51	VDD	power	VDD_DIG	power	109	VCC_A	power	GND
23	VDD_DIG	power	VDD_DIG	VDD_DIG	52	GND_A	ground	GND	ground	110	GND_DIG	ground	VDD_DIG
24	VDD_OSC	power	VDD_DIG	VDD_DIG	53	VCC_A	power	VCC_A	ground	111	VDD_DIG	power	VDD_DIG
25	VSS	ground	GND	GND	54	GND_DIG	ground	GND	LVDS_OUT	112	VDD_OSC	power	GND
26	VDD	power	VDD_DIG	VDD_DIG	55	VDD_DIG	power	VDD_DIG	N/A	113	VSS	ground	VDD_DIG
27	GND_A	ground	GND	GND	56	CHIP_ID<0>	POWER/GND	GND/VDD_DIG	LVDS_OUT	114	VDD	power	GND
28	VCC_A	power	VCC_A	VCC_A	57	CHIP_ID<1>	POWER/GND	GND/VDD_DIG	LVDS_OUT	115	GND_A	ground	VCC_A
29	GND_DIG	ground	GND	GND	58	CHIP_ID<2>	POWER/GND	GND/VDD_DIG	power	116	VCC_A	power	HV_SUB

Figure 25: Pad list.

369 Figure 26 shows a layout view with pad number for left and right side of the ASIC.

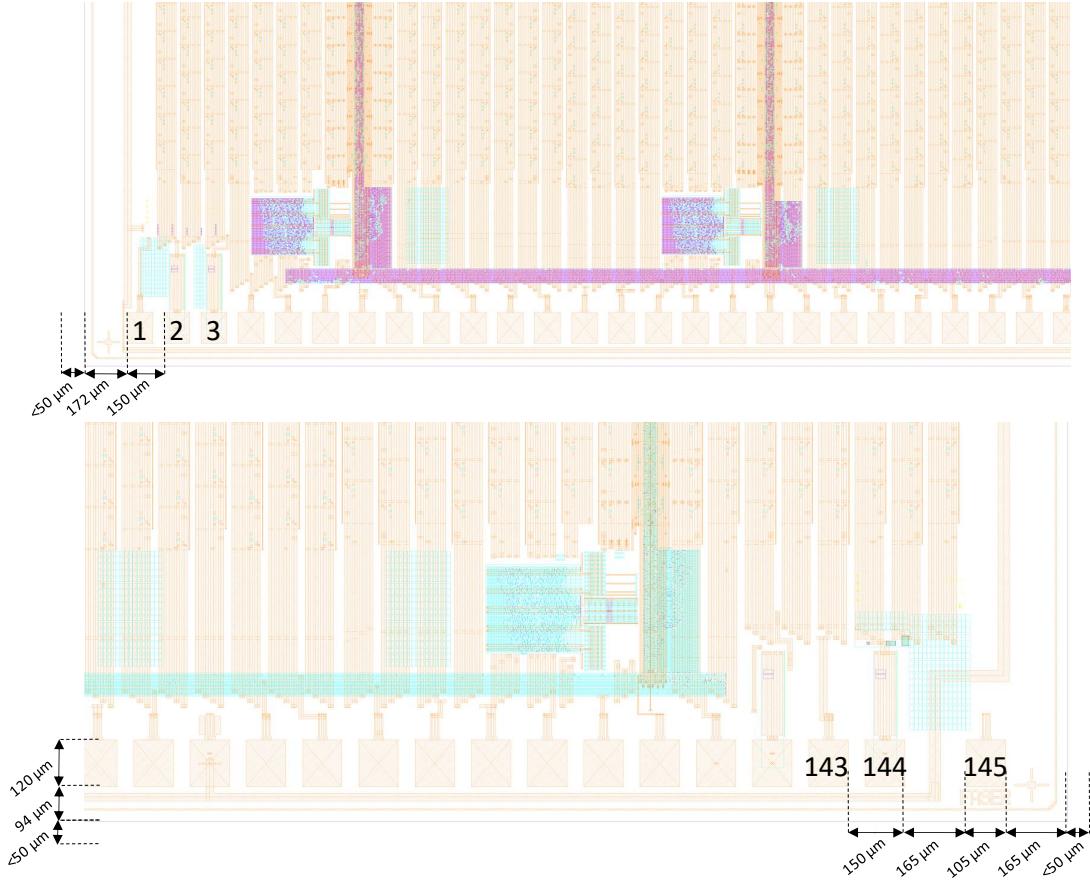


Figure 26: Top: left side of the chip. Bottom: right side of the chip. Maximum distance from left edge of the chip is estimated and depends on the width of the dicing line. The pads are all identical in size.

Table 7: I/O main characteristics.

I/O Type	Standard	Notes
Single-ended I/O	1.2 CMOS	full-swing
Differential Output	sub-LVDS	$\pm 250$ mV swing, 1.0 V common mode. Open drain, $100\ \Omega$ termination required (VDD_dig)
Differential Input	LVDS	Differential swing $> \pm 150$ mV, common mode $> 600$ mV required

370 Every single-ended pin is 1.2V CMOS (full-swing). The differential output signals are sub-LVDS, with a nominal  
 371 swing of  $\pm 250$  mV and common mode of 1.0 V. They are open drain, requiring 100 Ohm terminations towards  
 372 VDD\_dig externally. Differential inputs can read a larger range of signals, as long as the differential swing is  $> \pm$   
 373 150 mV and the common mode is  $> 600$  mV. They are high impedance internally, so lines need to be terminated  
 374 externally. In principle the input voltage should not go above Vdd\_dig, as it is the power supply connected to the  
 375 ESD protection of the padring. However, previous chips built in the same technology were successfully tested with  
 376 input above 2 V, at the cost of a marginally increased power consumption due to the ESD protection being partly  
 377 conductive. Table 7 reports a summary of the main characteristics of the I/O pads.

378 The pads of the I/O ring are characterized by a pitch of 150  $\mu$ m except for the rightmost one (HV\_SUB, 270  $\mu$ m  
 379 pitch from the neighboring pad, VCC\_A). Every pad has an opening of the passivation of 100  $\mu$ m  $\times$  120  $\mu$ m (XY).

## 380 3.8 LV/HV DESCRIPTION

Table 8: Supply and ground voltages characteristics. An internal generation of the threshold has been implemented, but not tested yet. If this internal threshold works, the external threshold supply will not be necessary anymore.  
**NOTE: The compliance of the power supplies should be set compatibly with the max current for operation reported in this table. The maximum current that can be configured in the chip is larger (see Table 1 in Section 1)** Please, read description below for implementation details on threshold and maximum currents.

Supply/Ground	Range [V]	Max. current (operation) [mA]	$\Delta I$ (operation) [mA]
Vdd_dig (Vdd_osc) / Vss_dig	1.1 - 1.3 (1.2)	110	<10
Vdd_disc / Gnd_disc	1.1 - 1.3 (1.2)	30 (+25 for thr)	5
Vcc_a / Vss_a	1.2 - 1.4 (1.2)	210	<1
Global threshold / Gnd_disc	0.5 - 0.9 (0.7)	60	<1

381 The ASIC required 3 independent low-voltage power supplies (see section 3.9 for more details and Table 8 for a  
382 summary):

- 383 • Vdd\_dig (Vdd\_osc) / Vss\_dig: Digital power supply for digital columns, periphery and oscillators. It should  
384 be programmable in a range from 1.1 V to 1.3 V (Nominal value 1.2 V). The maximum expected current  
385 absorption is 110 mA. **Extra power coming from super-column and periphery during readout is expected to  
386 be in the few mA range.** A safety factor of two (extra 50 mA) was applied to the TDC current absorption  
387 waiting for measurement on silicon.
- 388 • Vdd\_disc / Gnd\_disc: Power supply of the discriminators. It should be programmable in a range from 1.1 V  
389 to 1.3 V (Nominal value 1.2 V). In operation it is supplied on the same line as the digital power supply. The  
390 maximum expected current envisaged in operation is 130 mA<sup>4</sup>.
- 391 • Vcc\_a / Vss\_a: Analog power supply for the pre-amplifiers. It should be programmable in a range from  
392 1.2 V to 1.8 V (Nominal value 1.2 V). The maximum expected current absorption in operation is 210 mA.  
393 The maximum current that can be set is 1020 mA.
- 394 • Global threshold / Gnd\_disc: Global threshold for the discriminators. It can be generated internally in  
395 the chip (in which case the relative PAD can be used for monitoring of the threshold) or provided as an  
396 external power supply. **When generated internally**, a maximum current of 25 mA will be absorbed on the  
397 Vdd\_disc power supply. **If generated from outside**, it should be programmable in a range from 0.5 V to  
398 0.9 V (Nominal value 0.7 V). The maximum expected current absorption at 0.9V is 60 mA with maximum  
399 operation setting (30) on discriminator DAC. NOTE: if the global threshold is provided from outside the  
400 chip, the DACs dedicated to threshold set and threshold offset should be set to 0. Different values mean  
401 providing part of the threshold current from the Vdd\_disc power line.

402 The High Voltage can be applied both via a backside contact or using the top PAD HV\_sub. For FASER operation  
403 it will be provided by top PAD. It should be a negative HV programmable in a range from 0 V to  $-200$  V. The  
404 expected current absorption is below 1  $\mu$ A, but for debug purposes it is suggested to foresee a maximum current  
405 absorption as high as 100  $\mu$ A, if the system allows for it.

## 406 3.9 IMPLEMENTATION

407 All the power supply lines require a decoupling on board with 1 pF, 1 nF and 100 nF capacitors (one per power  
408 supply) locally near the chip, with the smaller capacitors as close as possible to the chips. An exception are  
409 the oscillator power supplies, which are referred to the digital power supply, but they requires local decoupling  
410 capacitors for each bonding pad. The board can have a common ground.

411 The differential lines (identified by the +/-) are SLVDS pairs and they should be terminated on the receiver.

<sup>4</sup>To be noted: this is the current set, which is pulled only when the pixel fires. The actual current in idle state is approximately a factor five smaller. Therefore this value largely overestimates the maximum current in operation for FASER

- 412 Readout\_clk lines are input lines and can operate at a maximum frequency of 200 MHz with a data rate of 200  
 413 Mbps in Single Data Rate (Section 3.5).
- 414 Trigger\_slow is a synchronous signal that anticipates the start of the readout. It is active **high**. TThe behaviour of  
 415 the signal is illustrated in Figure 2.
- 416 There are two data out lines (Data\_out[0] and Data\_out[1]). The chip can be configured to use either one  
 417 (Data\_out[0]) or both lines. The default implementation for FASER is using a single data out line.
- 418 Trigger\_fast lines (three LVDS pairs) are used only for the test board. They are asynchronous FAST-OR signals  
 419 and they can be sent on two  $50\ \Omega$  lines terminated on board next to SMA connectors, as shown in Figure 27.

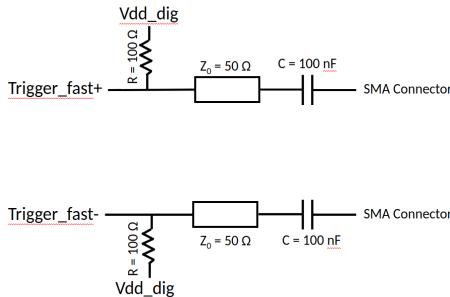


Figure 27: Implementation of trigger fast on the test board.

- 420 The chip\_id lines are used to identify the chip. They must be set either at Vss\_dig or Vdd\_dig. Each module  
 421 should have a unique identifier for different chips.
- 422 SPI lines (SPI\_CS, SPI\_CLK and SPI\_MOSI) are LVCMOS input signals (reference level is Vdd\_dig). See Section  
 423 3.1 for details on SPI implementation.
- 424 GLOB\_TH is used to provide the global threshold. Figure 28 shows an example of implementation for the test  
 425 board.

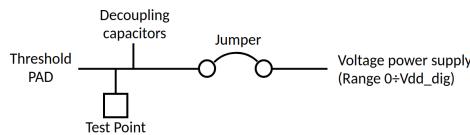


Figure 28: Implementation of the threshold sense/force circuit.

- 426 The high voltage PAD is connected to the HV power supply line. A dedicated filtering circuit is required on board  
 427 (Figure 29). The high voltage can be provided both by wire bonding to the High Voltage pad and by backside  
 428 biasing. The proposed solution for the FASER pre-shower is to provide high voltage only using the dedicated pad.

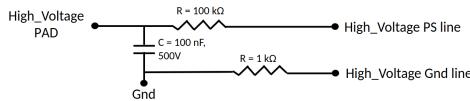


Figure 29: Implementation of the HV filtering circuit.

## 429 3.10 CALIBRATIONS

- 430 The ASIC has a single global threshold, which is controlled by two 8-bit DACs, one setting the offset, which will  
 431 be at a fixed value and it should not be changed during calibration, and one setting the fine threshold, which can be  
 432 used for calibration.

433 As there is no per-pixel threshold correction DAC, the threshold scan will only allow to check for the threshold  
434 dispersion within the matrix and define a global threshold that allow for quiet operation of the ASIC.

435 **We assume here that we inject the testpulse on 16 pixels per super-column, 1 super-column at the time**  
436 **(conservative). We scan over 32 threshold values (of the 256 possible).** More values would be beyond the  
437 interesting range of the threshold and with too small of a step.

438 The clock of the testpulse (*tp\_clock*) is  $4096 \cdot 5 \text{ ns} = 20.48 \mu\text{s}$ . The testpulse is high for 3 *tp\_clock* cycles and low  
439 for N *tp\_clock* cycles, where N can be set between 1 and 255. We assume N = 1, so the period of the testpulse is  
440 82  $\mu\text{s}$ .

441 The calibration procedure is made of three steps:

442 1. Threshold scan.

443 2. Noise occupancy.

444 3. Charge calibration.

445 **Threshold scan**

446 The first step of the calibration procedure is the threshold scan. It is used to set the thresholds at the average input  
447 equivalent value of 0.5, 1 or 2 fC (depending on the detector plane). Since there is a single threshold on the chip  
448 and no pixel-to-pixel calibration, a spread in threshold of 20% is accepted among different pixels.

449 The procedure below describes the calibration of a column (4096 pixels, 1 global threshold), it should be repeated  
450 13 times.

451

452 • Set the ASIC DAC "bias\_testpulse" at 1 fC (DAC input: 2). 1 SPI commands (20 bits @ 10 MHz, 2  $\mu\text{s}$ ).

453 • Mask all pixels. 512 SPI commands (10240 bits @ 10 MHz, 1024  $\mu\text{s}$ ).

454 • Set mask pattern to "rolling mask" configuration on the first pixels of the column (as in Figure 30). 16 SPI  
455 commands (320 bits @ 10 MHz, 32  $\mu\text{s}$ ), same command than 'Mask all pixels' but for the 16 first pixels.

456 • Threshold scan iterations (32 values):

457 – Set the global threshold. 2 SPI commands (40 bits @ 10 MHz, 4  $\mu\text{s}$ ).

458 – Rolling mask shift iterations (x128):

459 \* Test pulse iterations (100 values):

460 · Send 100 testpulse triggers. (bits for a chip with 13 super-columns: 13\*8 bits header + 4 bits  
461 column hit + 1\*20 bits TDC calib + 1\*24 TDC hit flag + 4\*16 Charge data + 2048 hit flags +  
462 3\*19 TDC data = 2321-bits @ 200 Mbps, 11.605  $\mu\text{s}$  + 5.120  $\mu\text{s}$  index switch = 16.725  $\mu\text{s}$ )

463 \* End 'Test pulse iterations'

464 \* Rolling mask shift: 4 SPI commands (80 bits @ 10 MHz, 8  $\mu\text{s}$  per shift, see remark below).

465 – End 'Rolling mask shift iterations'

466 • End 'Threshold scan iterations'

467 Remark on Rolling mask shift:

468 The masking bit is on a long shift register. After having set the pattern for the lowest pixels as in Figure 30, one  
469 can send 4 SPI commands (essentially the command that sets the mask to '1' in a row, it is all 1 for the mask to  
470 shift the pattern upwards by one row while keeping the lowest row masked. Once you reach the end of the column  
471 you need three special commands to produce the three patterns that were not tested in the first row, the two patterns  
472 that were not tested in the second row and the pattern that was not tested in the third row.

473

474 Numbers for one ASIC:

- 475 • Total number bits:  $13 \text{ super-columns} \cdot 32 \text{ thresholds} \cdot 128 \text{ patterns} \cdot 100 \text{ pulses} \cdot 2321\text{-bits} \approx 12 \text{ Gbits (1.5 GBytes)}$ .  
 476
- 477 • Total number of bits after compression:  $32 \text{ thresholds} \cdot 100 \text{ pulses} \cdot 26624 \text{ pixels} \cdot (15 \text{ bits address} + 4 \text{ bits charge info}) \approx 1.6 \text{ Gbits (200 MBytes)}$ .  
 478
- 479 • Calibration time:  $13 \text{ super-columns} \cdot (\text{mask settings } 1056 \mu\text{s} + (\text{thresholds } 32 \cdot (\text{Thd setting } 4 \mu\text{s} + 128 \text{ mask shifts} \cdot (\text{rolling mask } 8\mu\text{s} + 100 \text{ pulses} \cdot (82 \mu\text{s})))))) \approx 437\text{s (7min 17s)}$ .  
 480

481 **NOTE: The number of threshold values that must be used for calibration can be reduced once the stability  
 482 of the ASIC working point is defined.**

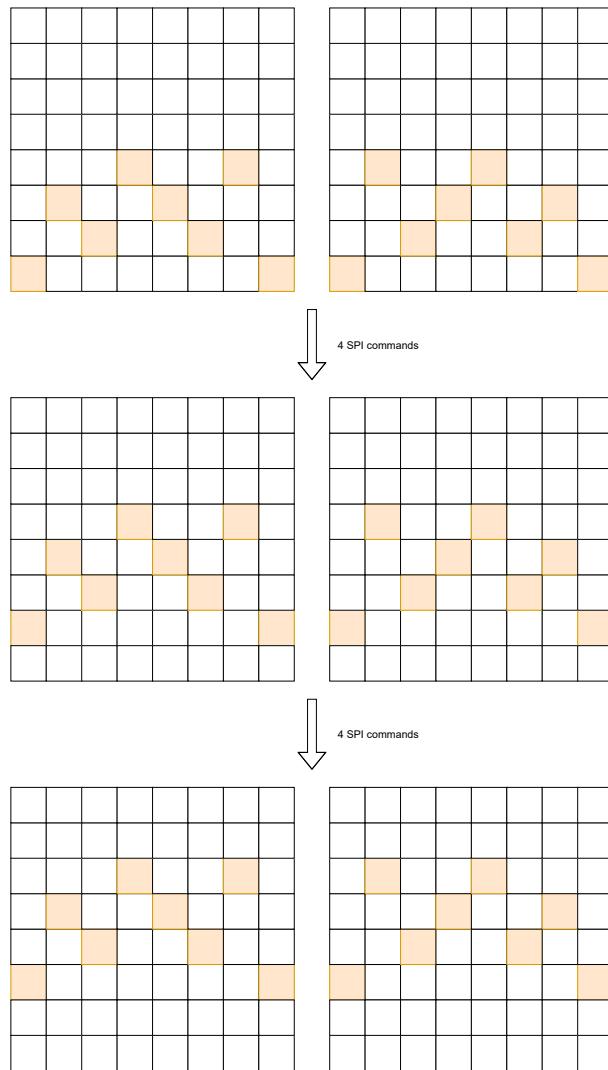


Figure 30: Unmasked pixel distribution for rolling mask and examples of rolling mask shift.

483 **Noise occupancy**

484 The noise occupancy measurement is used to verify that the detector is noise free at the selected threshold. It  
 485 requires setting the global thresholds identified with the previous calibration step and starting a readout for a  
 486 duration long enough to estimate an appropriate upper limit for the noise rate. Considering a background hit rate  
 487 per chip of  $\approx 1.73 \text{ Hz}$  (600 ms), to guarantee a negligible impact from noise, the chip should have a noise hit  
 488 rate of  $0.016 \text{ Hz}$  (60 s). A more conservative value may be appropriate to guarantee stable operation for long runs  
 489 between two calibrations.

490 **Charge calibration**

491 The charge calibration is used to generate the calibration data for all the pixels with different charge injections. The  
492 procedure described is used for a single super-column, it should be repeated 13 times for all the super columns.  
493 The charge calibration is required only after a power cycle of the ASIC.

494

495 The procedure for **one ASIC super-column** is the following:

- 496 • Mask all pixels. 512 SPI commands (10240 bits @ 10 MHz, 1024  $\mu$ s).
- 497 • Set mask pattern to "rolling mask" configuration on the first pixels of the column. 16 SPI commands (320  
498 bits @ 10 MHz, 32  $\mu$ s, same than for threshold scan).[YFAV: I think this step can be skipped since we can directly set the mak in the first 16 SPI commands of the previous item.]
- 499
- 500 • Charge pulses iterations (16 values):
- 501   – Set the ASIC DAC "bias\_testpulse". 1 SPI command (20 bits @ 10 MHz, 2  $\mu$ s).
- 502   – Rolling mask shift iterations (x128):
- 503     \* Test pulse iterations (100 values):
- 504       · Send 100 testpulse triggers. (bits for a chip with 13 super-column: 13\*8 bits header + 4 bits  
505        column hit + 1\*20 bits TDC calib + 1\*24 TDC hit flag + 4\*16 Charge data + 2048 hit flags +  
506        3\*19 TDC data = 2321-bits @ 200 Mbps, 11.605  $\mu$ s + 5.120  $\mu$ s index switch = 16.725  $\mu$ s)
- 507     \* End 'Test pulse iterations'
- 508     \* Rolling mask shift: 4 SPI commands (80 bits @ 10 MHz, 8  $\mu$ s per shift, same as for threshold  
509        scan.
- 510       – End 'Rolling mask shift iterations'
- 511 • End 'Charge pulses iterations'

512 Numbers for one ASIC:

- 513 • Total number of bits: 13 super-columns  $\cdot$  16 charge values  $\cdot$  128 patterns  $\cdot$  100 pulses  $\cdot$  2321-bits  $\approx$  6 Gbit  
514 (770 MBytes).
- 515
- 516 • Total number of bits after compression: 16 charge values  $\cdot$  100 pulses  $\cdot$  26624 pixels  $\cdot$  (15 bits address + 4  
517 bits charge info)  $\approx$  810 Mbits (100 MBytes).
- 518
- 519 • Calibration time: 13 super-columns  $\cdot$  (mask settings 1056  $\mu$ s + (Charges 16  $\cdot$  (bias setting 2  $\mu$ s + 128 mask  
520 shifts  $\cdot$  (rolling mask 8 $\mu$ s + 100 pulses  $\cdot$  (82  $\mu$ s))))  $\approx$  218s (3min 38s).

521 4 Simulated performance

522 4.1 DAC characterization

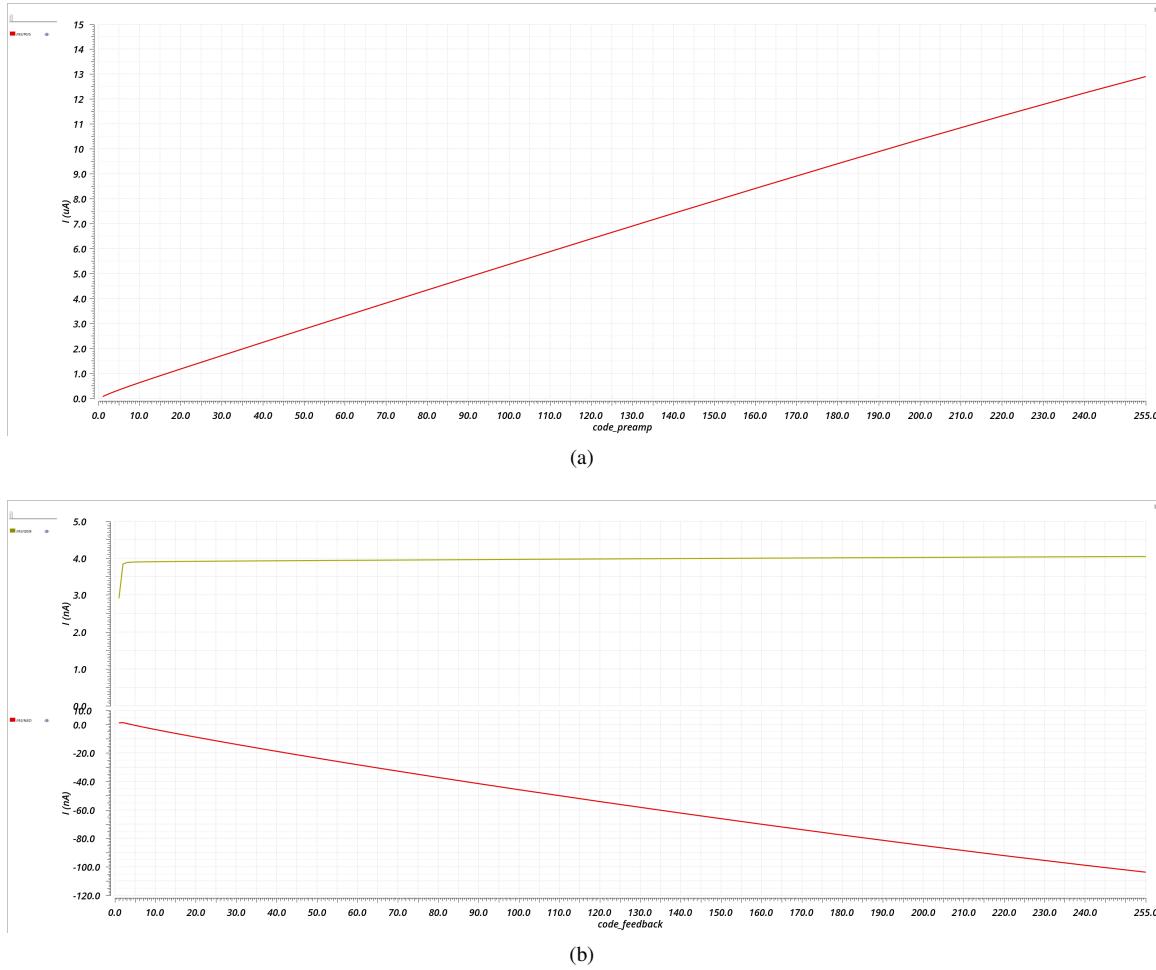
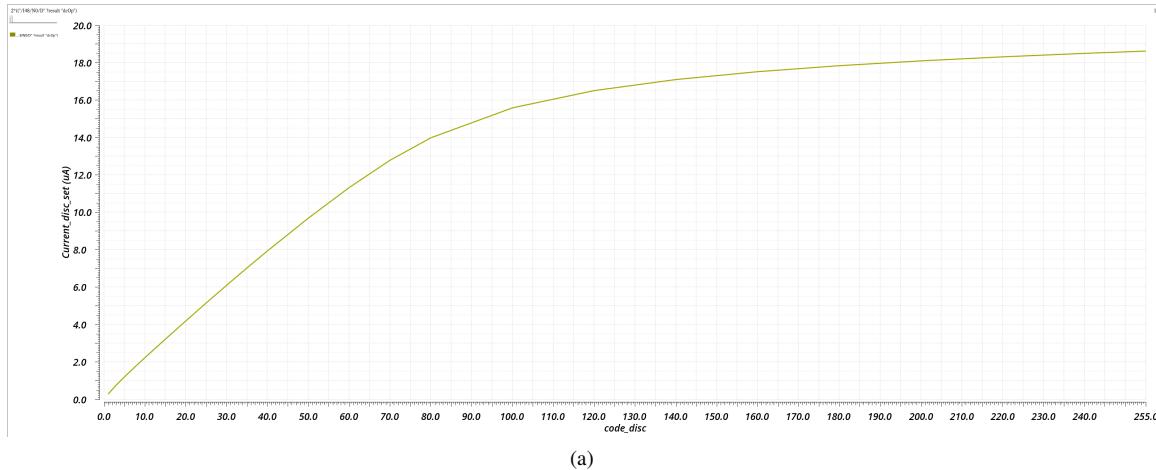
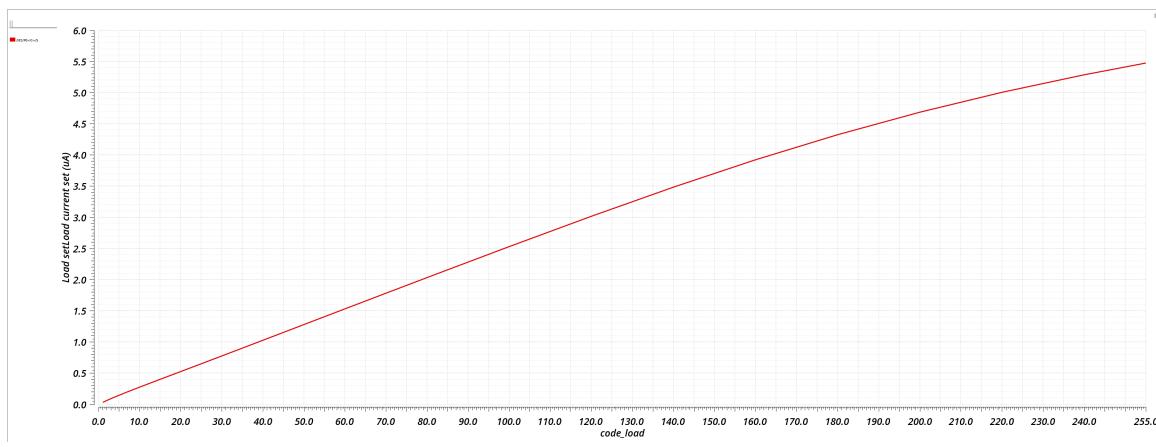


Figure 31: a) Preamp current vs preamp DAC setting, with high feedback current regime. b) Preamp base current and feedback current vs feedback DAC setting for preamp DAC setting = 50.

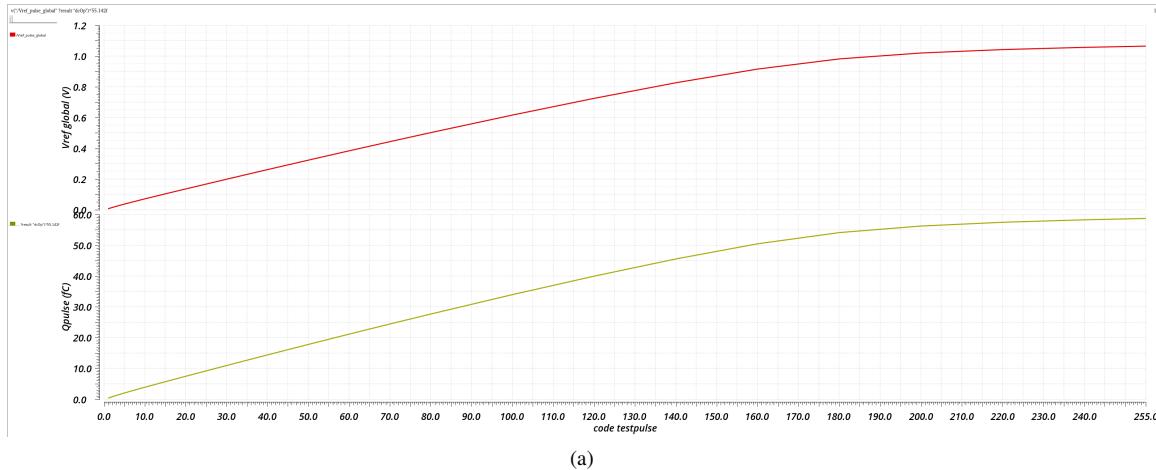


(a)

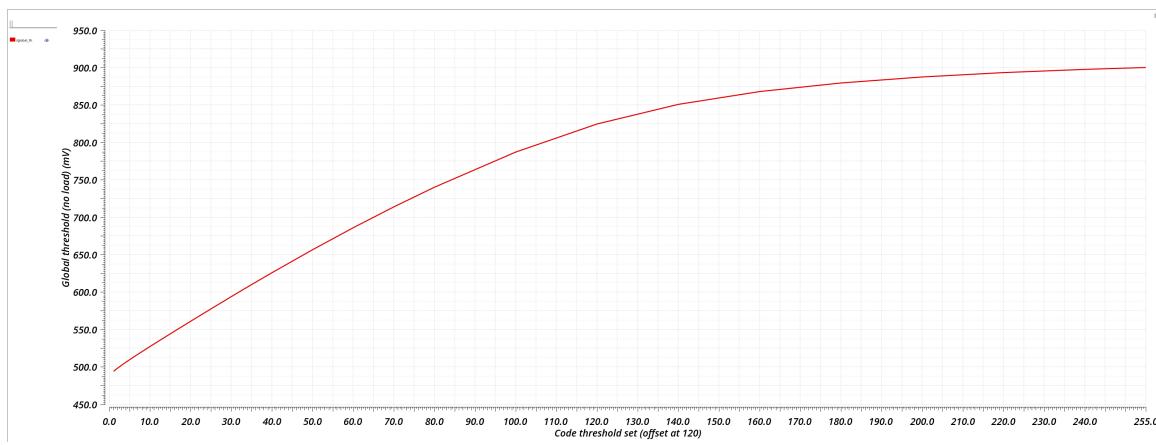


(b)

Figure 32: a) Discriminator maximum current vs discriminator DAC setting. b) Memoery load current vs load DAC setting.



(a)



(b)

Figure 33: a) Testpulse injected charge vs testpulse DAC setting. b) Global threshold vs threshold\_set DAC (at threshold\_offset DAC = 120).

## 5 Appendix

### 5.1 Reset modes timing diagrams

**Memory**

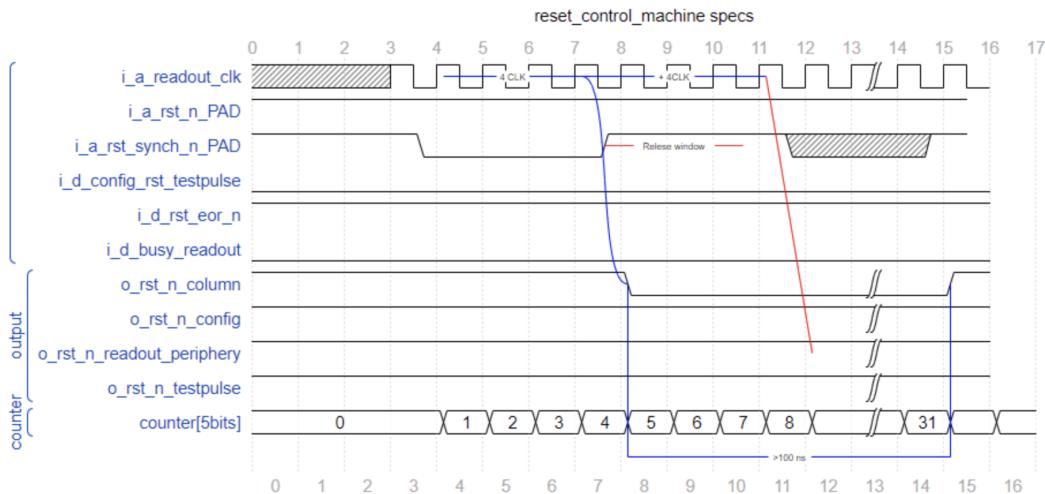


Figure 34: Timing diagram of the memory reset.

**Memory with busy\_readout active high**

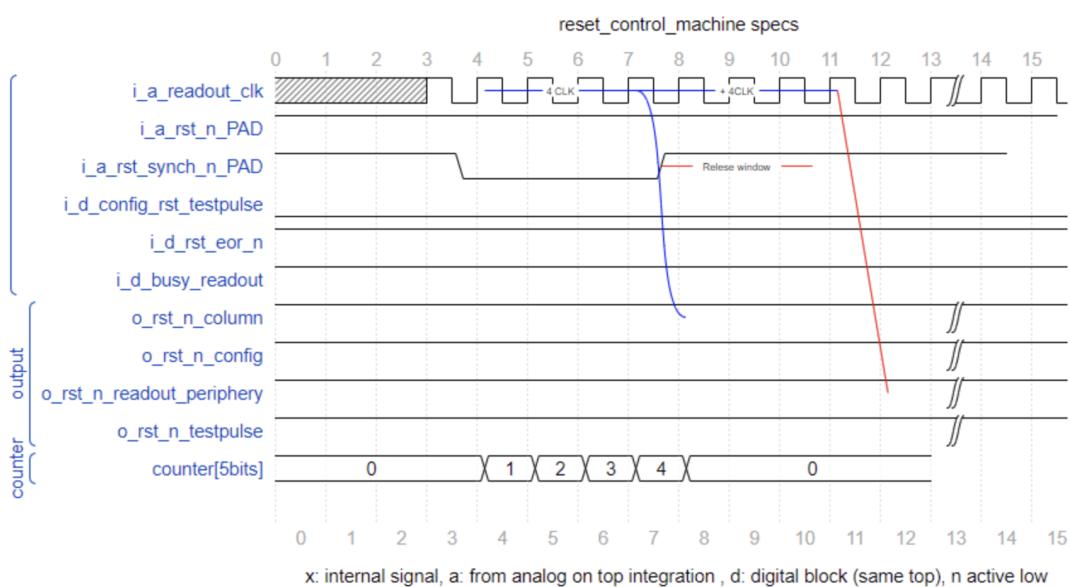
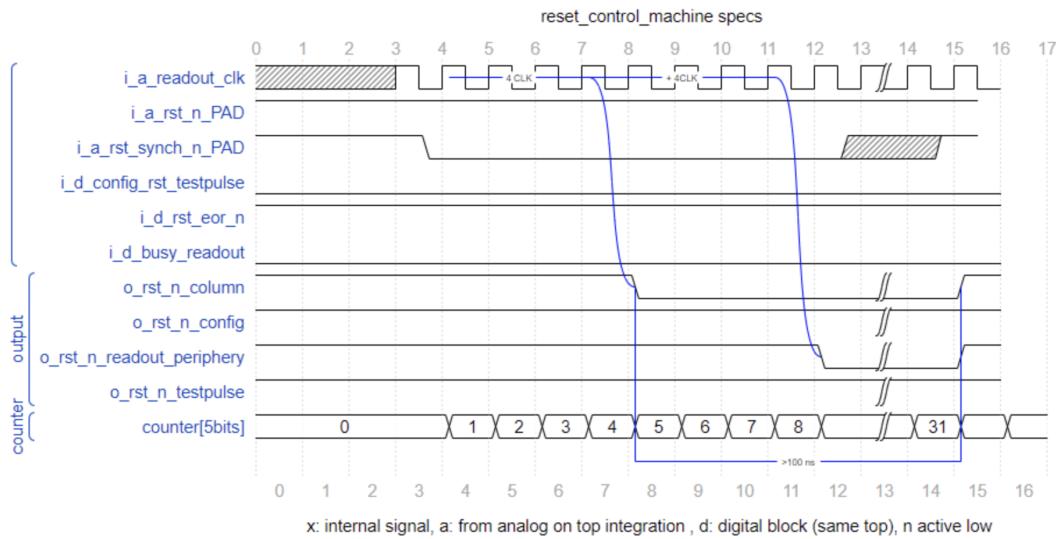


Figure 35: Timing diagram of the memory reset during a readout phase.

### Flush

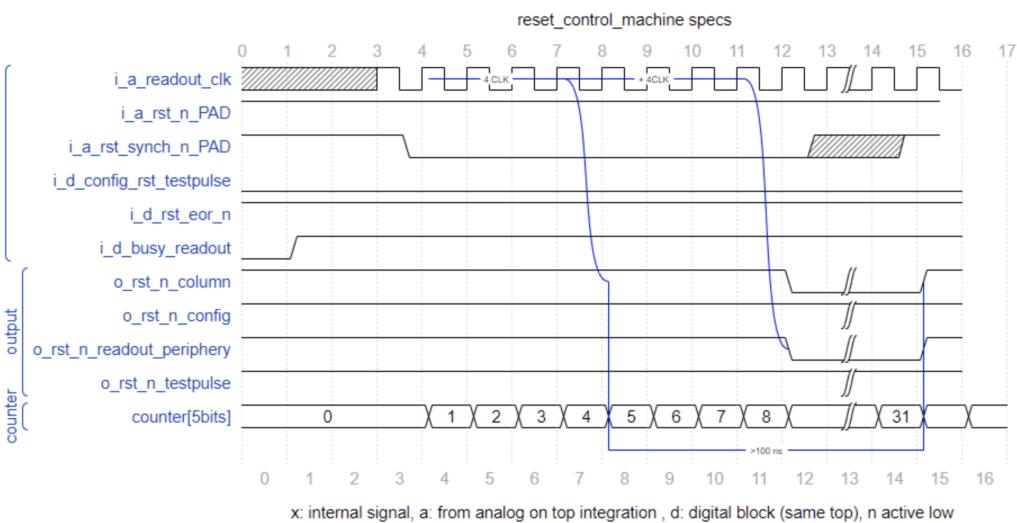


To: Readout periphery, Column, TDC

- busy readout has no effect
- Signals omitted in timing diagram are ignored

Figure 36: Timing diagram of the flush reset.

### Flush with busy readout



### Flush with arbitrary duration longer than 32

If **i\_rst\_synch\_n\_PAD** is not released before the 31 clock cycles, then the output **o\_rst\_n\_column** and **o\_rst\_n\_readout\_periphery** will be held active low until the clock cycle after the release of **i\_rst\_synch\_n\_PAD**.

Figure 37: Timing diagram of the flush reset during a readout phase.

## Testpulse

To testpulse

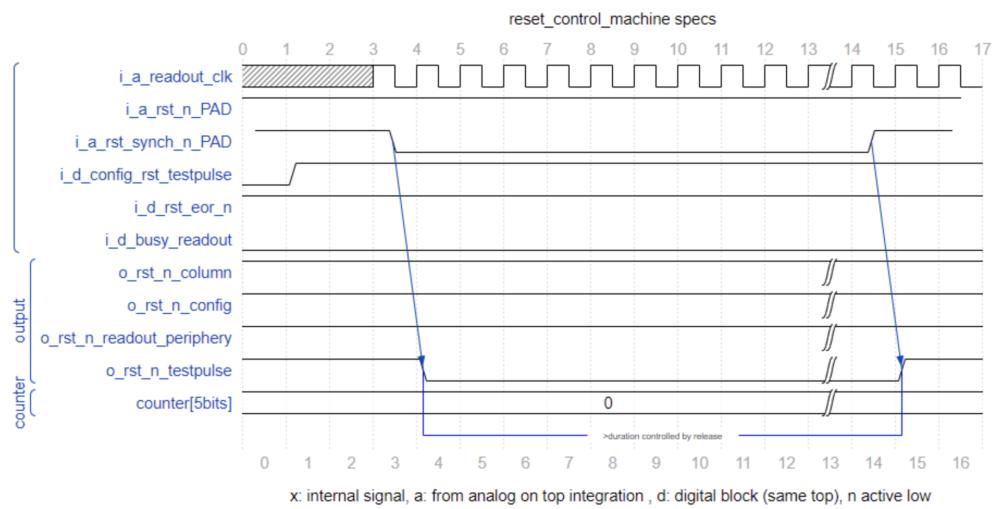


Figure 38: Timing diagram of the testpulse reset.

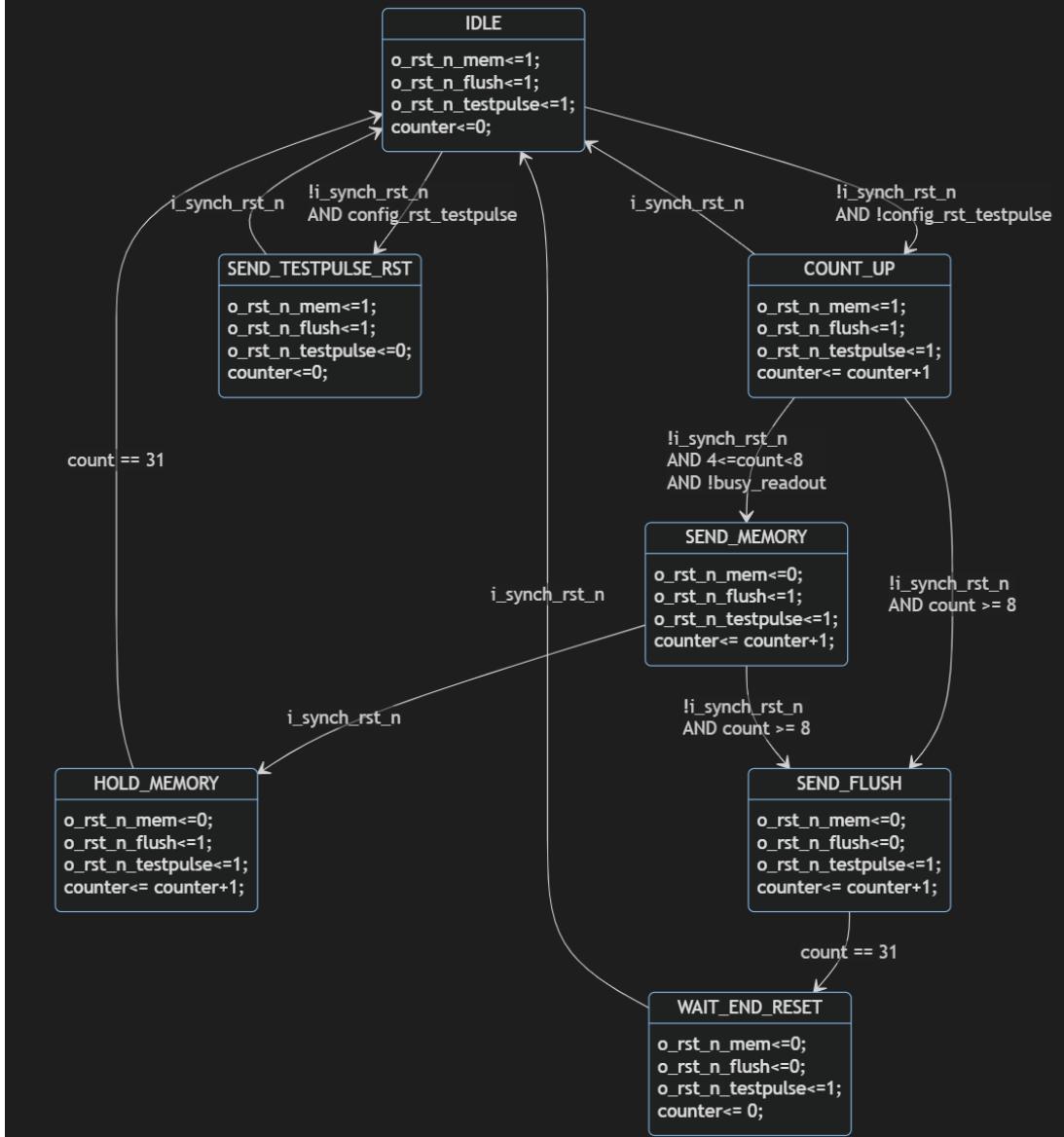


Figure 39: Reset fsm diagram. This block controls all the reset generated by the synchronous reset input signal(see Sec.3.3).