

# Apsel4D

## User Guide

---

*A digital sparsification and  
readout circuit for a  $128 \times 32$   
matrix of MAPS*

---

**Alessandro Gabrielli**

---

**INFN & Physics Department  
University of Bologna – Italy**

**on behalf of the SLIM5 collaboration**

## 1. Notes for the readers

---

- All the I/O ports and keywords are shown in **bold blue type**,
- In the figures of the circuit, the left ports are to be considered as inputs while the right ones are outputs and this rule applies to all the circuits described in this document,
- MPs = Macro Pixels, 256 altogether,
- MCs = Macro Columns, 32 altogether,
- MRs = Macro Rows, 8 altogether,
- This document is a regular extension of the previous user's manual of the **Apsel3D** chip that implements a similar readout circuit for 256 pixels instead of 4096. For **Apsel3D** version please see the following link: <http://www.bo.infn.it/slim5/Apsel3D/Readout32x8-Users-Guide.pdf>

### 1.2 Version

- Datasheet version 1.0 - 18 December 2007
- Datasheet version 1.1 - 03 April 2008
- Datasheet version 1.2 - 10 April 2008
- Datasheet version 1.3 - 15 April 2008
- Datasheet version 1.4 - 17 April 2008

For any question, please email to: [alessandro.gabrielli@bo.infn.it](mailto:alessandro.gabrielli@bo.infn.it)

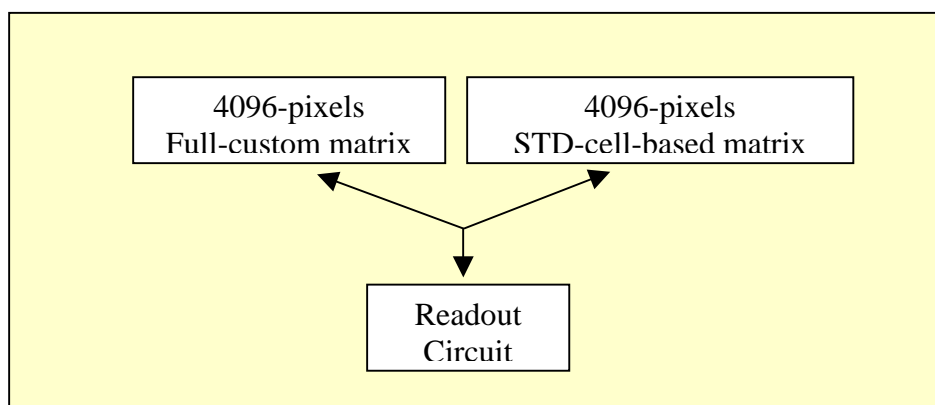
The updated document can be found in the following link:

<http://www.bo.infn.it/slim5/Apsel4D/Readout128x32-Users-Guide.pdf>

## 2. Introduction

The circuit is a digital architecture for a sparsified readout that interfaces with a matrix of 4096 Monolithic Active Pixel Sensor (MAPS). It is the base for a prototype of a mixed-mode ASIC, namely Apsel4D. It reads out and sparsifies the hits of a matrix of 4096 pixels. Once read, the hits are switched off. The matrix is divided into regions of 4 x 4 single pixels thus, 4096 pixels are clustered into 256 groups of 16 pixels each, here-in-after named macro-pixels (MPs). In addition, the matrix is arranged in 128 columns by 32 rows of single pixels or, from a different viewpoint, in 32 columns of MPs, called MCs, by 8 rows of MPs, called MRs. Basically, let us say that when the matrix has some hits (pixels that detect an over-threshold charge), it is swept from left to right and, at each clock period, all the hits present in a column of pixels, from 1 to 32, can be read out in parallel. This operation starts as long as a hardwired readout queue has free locations to temporarily store the information of the hits. In fact, at the hits' coordinates is associated a time mark (time-stamp) and the overall formatted data are either sent to the output port, or temporarily stored in a FIFO-like memory in case the output port is busy. Thus, in principle, the architecture can read out the matrix up to 32 hits at a time in case they belong to the same column and can send the formatted data to the output but, at the same time, the output port can only accept one-hit information at a time, and this is why a queuing system is necessary.

Moreover, the global architecture might be considered as a circuit that can run in two different operating modes, called **custom-mode** and **digital-mode**. In fact, it can be connected to an actual full-custom matrix of MAPS or to a digital matrix emulator composed of standard cells. In the first case the pixels may only be switched on via striking particles while in the second case the digital matrix must be loaded during an initial slow-control phase. The two different implementations share the same matrix's I/O pins but can be selected and activated only one at a time. For both modes, before running, a slow-control phase is required to load an internal configuration. In particular, 256 mask signals should be provided to select the MPs, which are to be read and which are not, for example in case they are too noisy or broken (**MP-mask**). Other 32 masks can deactivate any row of pixels (**ROW-mask**). Default mask, after a reset phase, is all-at-1, meaning no-mask both for **MP-masks** and for **ROW-masks**. Any **ROW-mask** is to avoid that a single burned pixel invalidates the entire MR it belongs to. Moreover, it must be selected which of the two operating modes is wanted and, consequently, which matrix is to be enabled. The default mode, after a reset phase, is the **digital-mode**. In addition, only for the digitally emulated matrix, 256 registers should be loaded to simulate a given charge injection over the silicon area. Default registers, after a reset phase, are all-at-0, meaning no hits. The readout circuit operates in the same manner for the two modes. Fig. 1 shows a sketch of the two operating modes of Apsel4D.



**Figure 1: Apsel4D operating modes: custom-mode and digital-mode**

## 2.1 The matrix organization

This is valid for both **custom-mode** and **digital-mode**. The entire matrix of figure 2, composed of 4096 pixels, is to be interpreted as follows:

- 32 MCs, addressed from left to right, range from 31 to 0,
- 32 rows of pixels, addressed from top to bottom, range from 31 to 0,
- 128 columns of pixels, addressed from left to right, range from 127 to 0,
- 4 columns of pixels inside each MP, from left to right, range from 3 to 0.

In this view each pixel is identified with a given MC, a given column inside the MC and a pixel row. By converting these coordinated in digital logic it turns out 5+2+5 bits, i.e. 12-bits altogether which address exactly 4096 pixels. This is the way the addresses are sent to the readout output port.

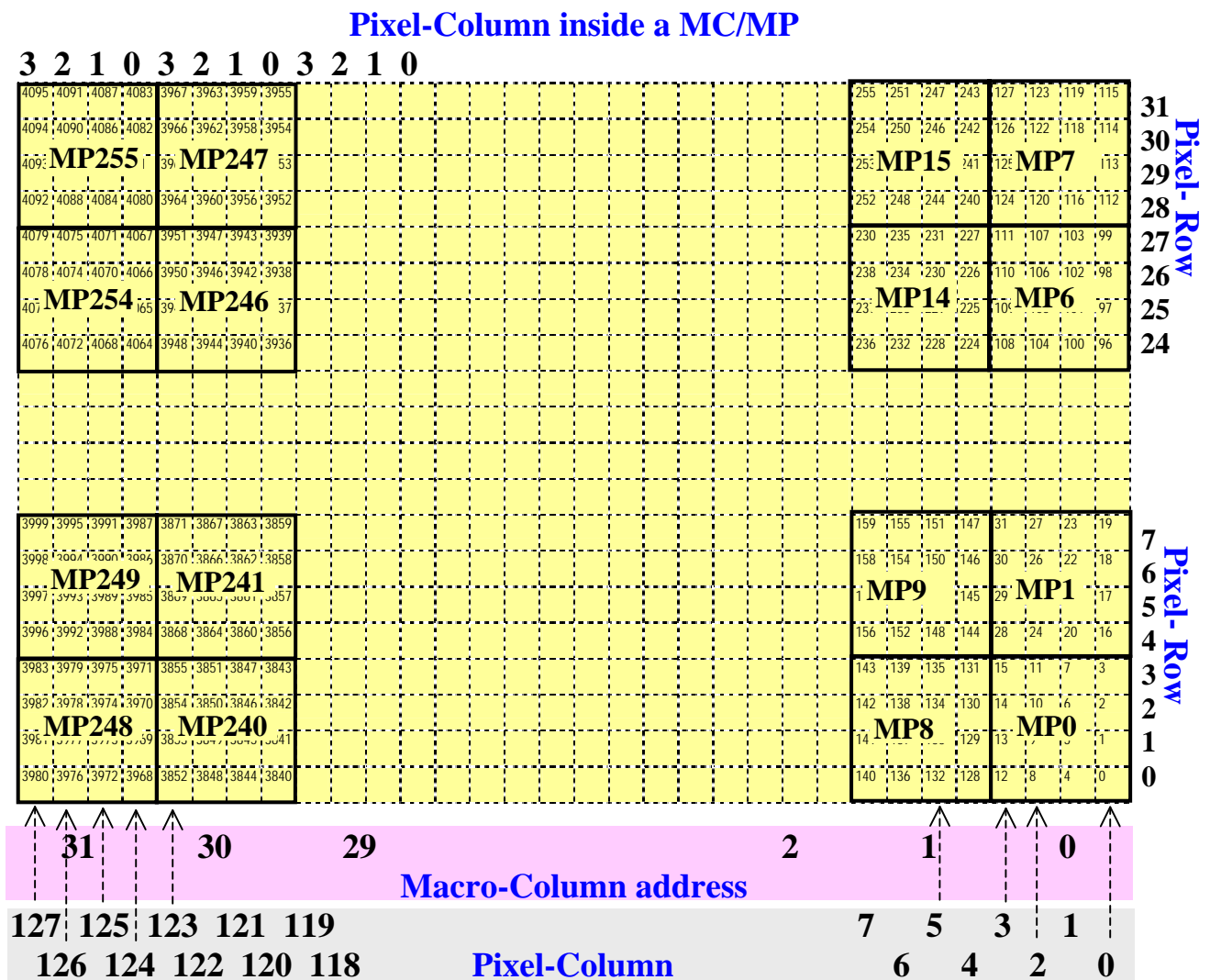


Figure 2: 4096-pixel Matrix

## 2.2 The Output Format

Here follows the 21-bit output format of the readout information. Each hit-information includes the address coordinates and the **time-stamp (TS)** it was switched on:

<Data-Valid><Pixel-Row><Pixel-Column-within-MP><Macro-Column-Address><Time-Stamp>

- **Data-Valid** is a 1-bit signal: it validates the rest of the data,
- **Pixel-Row** is a 5-bit bus: 0 to 31 as the row moves from bottom to top,
- **Pixel-Column-within-MP** is a 2-bit bus: 0 to 3 as the MP column moves from right to left,
- **Macro-Column-Address** is a 5-bit bus: 0 to 31 as MC moves from right to left,
- **Time-Stamp** is a 8-bit time mark: 0 to 255 incremented upon BC rising edge.

## 2.3 Block Hierarchy

The entire readout circuit is composed of the following blocks:

- Readout Circuit----- top level
  - o Time-Stamp----- second level
  - o 4 × Barrel----- second level
  - o Barrel-Final----- second level
  - o Latch-Enable----- second level
  - o MC-Address-Decoder----- second level
  - o Matrix of 4096 pixels: Matrix-Dummy----- second level
  - o Matrix of 4096 pixels: Matrix-Custom----- second level
    - 256 MP----- third level
  - o Slow-Control----- second level
  - o 4 × Sparsifier----- second level
  - o Sparsifier-OUT----- second level

## 2.4 Block Sketch

The entire readout circuit is divided into the following logical blocks of figure 3. All the circuits are described along this document but the **Custom-Matrix**. This latter is logically equivalent to the **Dummy-Matrix** and, for the readout viewpoint, behaves in the same way.

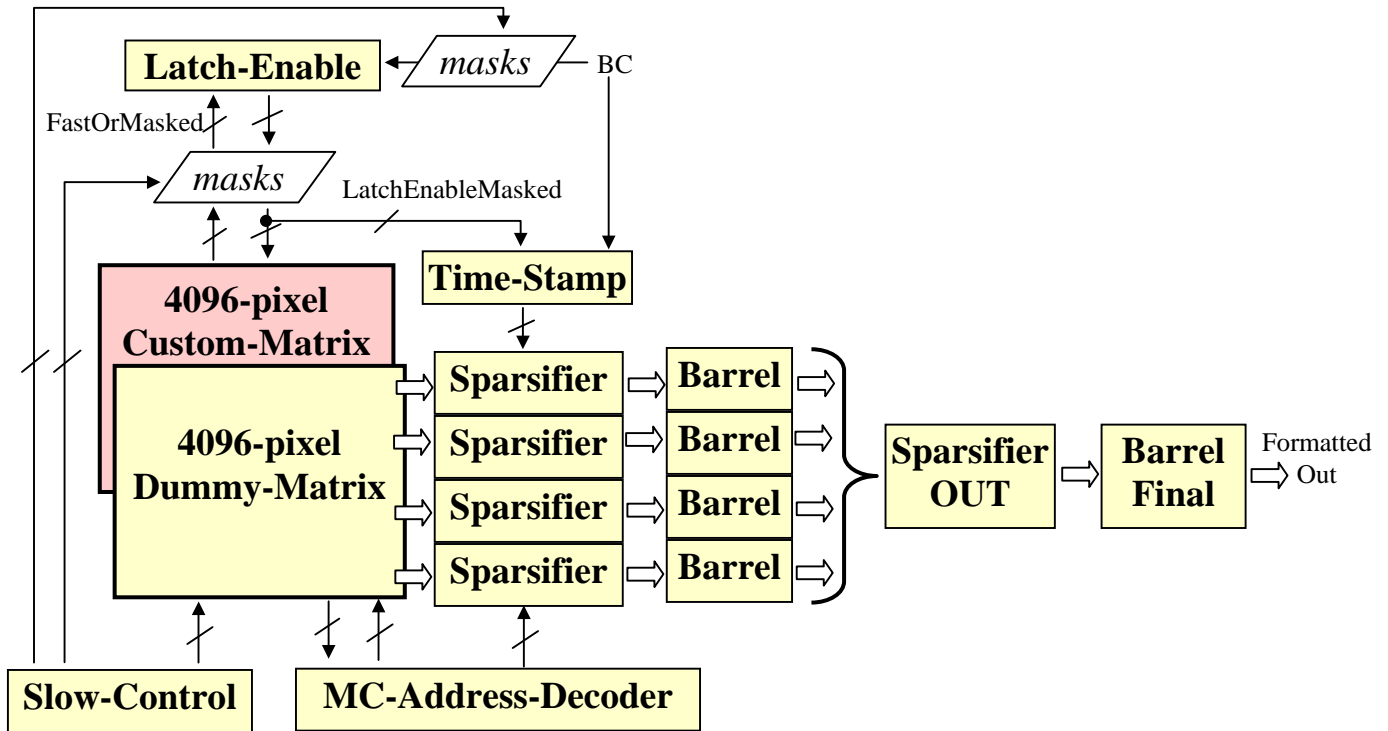


Figure 3: Readout Hierarchy

## 2.5 Matrix sweep

By looking at the figures 2 and 3, the matrix (**Dummy** or **Custom**) is swept as follows:

- the matrix is always swept from left to right and only the frozen **MPs** are considered,
- the 4 **Sparsifiers** work in parallel and cope with 8 rows of pixels each, out of the 32 rows,
- at any time one only column of pixels is considered thus, is not possible that different **Sparsifiers** read different columns at the same time,
- at any clock time, at most, 8 hits per **Sparsifier** can be read, leading to 32 hits altogether. Nevertheless, if this is the case, the successive **Barrels** force very soon the **Stand\_By** condition as if a whole column is lit, this is a consequence of very high (local) occupancies which are non compatible with the output port that outputs one hit only at a time. The event of local high occupancy can be handled as long as the average hit-rate is smaller that the throughput, see paragraph 3.3,
- the **Sparsifiers** have a depth of 32 but, taking into account that at any clock cycle up to 8 hit per **Sparsifier** might be loaded, the **Stand\_By** condition must be forced in advance when even though the queues are not be completely full,
- the **Stand\_By** condition can be forced both by the 4 **Barrels** or and by the **Barrel\_OUT**.

### 3. Readout

The figure shows the I/O ports of the entire circuit plus **SEED\_VECTOR** and **TEST** ports used only for design and simulation purposes: they will not be present on the final ASIC.

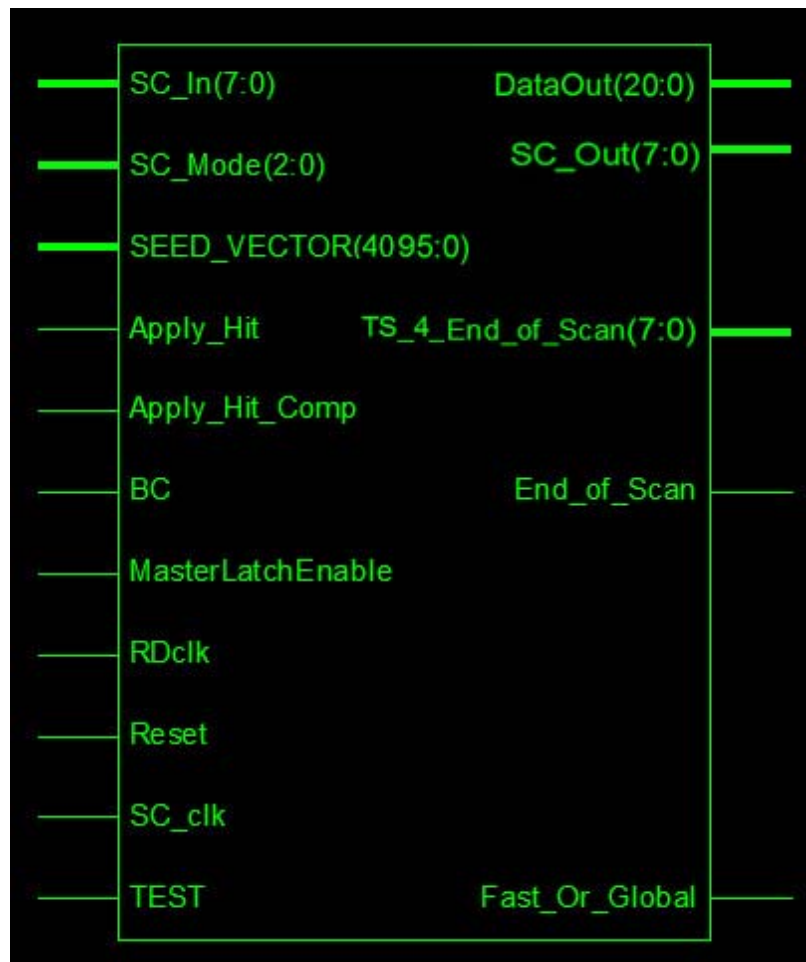


Figure 4: Readout

Thus, for this circuit the 4096-bit **SEED\_VECTOR** and the **TEST** ports are not real, they have been used only to stimulate the whole readout architecture. Here is a list of all the I/O ports:

- **SC\_In** is an 8-bit port used to load the internal registers, depending on the **SC\_Mode** value, at the rising edge of **SC\_clk**,
- **SC\_Mode** is a 3-bit port that specifies the required slow-control operation. During a slow-control phase several operations may be requested:
  - o a dummy-pixels load, in **digital-mode**, to configure the dummy matrix. It requires 512 **SCclk** periods as it loads 8-bits at a time as internal hits,
  - o a **MP-mask** load to select the **MP** that must be really seen from others which must be masked. It requires 32 **SCclk** periods as it loads 8-bits at a time,
  - o a **ROW-mask** load to select the **ROW** that must be really seen from others which must be masked. It requires 4 **SCclk** periods as it loads 8-bits at a time,
  - o an operating-mode selection to enable either the **custom-mode** or the **digital-mode**. It requires one **SCclk** period as it loads just a 1-bit register,

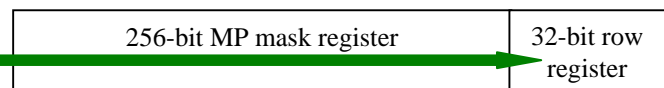


- a scramble operation on the dummy-pixels (sum, not, shift). It requires one **SCclk** period as it is a no-load operation,
- a **Soft\_Reset** to reset the time-stamp counter. It requires one **SCclk** period as it is a no-load operation.
- a **time-stamp-read** to see the time-stamp counter value. It requires one **SCclk** period as it is a no-load operation.

All these operations are synchronized on the rising edge of the slow-control clock, **SC\_clk**, by following a proper scheme. Here is a summary of the operations that can be selected via the **SC\_Mode** slow-control port:

- SC\_Mode** = 000 → load 4096 **Dummy-Pixels**. First MP address ranges from 0 to 15, second from 16 to 31, and so on up to 256<sup>th</sup> MP that ranges from 4080 to 4095. . At each **SC\_clk** period a half MP is configured, 512 periods for 256 MPs. This is only valid in **digital-mode**,
- SC\_Mode** = 001 → load 256 + 32 **Mask** bits. When loading the masks the **Row-masks** must be placed first then follow the **MP-masks**. Altogether 4 periods are required to load the **Row-masks** and other 32 periods to load the 256 **MP-masks**. Please note that as all the masks are cascaded, if a number of **SC\_clk** loading periods smaller than 36 is used, the masks are partially loaded. However, if only a subset of masks is required, a number of **SC\_clk** loading periods smaller than 36 could be enough. For example if only Mask(2) and Mask(10) are required, only one **SC\_clk** loading period is enough as they belong to the first 8-bit set of loaded masks. During normal running mode, **MP-mask(i)** masks MP(i).

the cascaded register is filled up 8-bits at a time



- SC\_Mode** = 010 → load the **Actual\_Dummy** register. This enables the **custom-mode** when is low and the **digital-mode** when is high,
- SC\_Mode** = 011 → read the **Time-Stamp** counter that is posted on **SC\_Out** port.
- SC\_Mode** = 100 → **Soft\_Reset** request to reset only the time-stamp counter inside the **Time-Stamp**,
- SC\_Mode** = 101 → 1-bit rolling shift on the **Dummy-Pixels** ( $N^{\text{th}} \rightarrow N^{\text{th}}+1$ ,  $255^{\text{th}} \rightarrow 0^{\text{th}}$ ). This is valid only in **digital-mode**,
- SC\_Mode** = 110 → add 1 to each MP 16-bit configuration. Thus, if a MP is configured with 16 bits, then this configuration is added to 0x"0001" to get a new configuration. This is valid only in **digital-mode**,
- SC\_Mode** = 111 → this is used as the normal running mode once either the dummy-pixels have been loaded in **digital-mode** or the matrix of MAPS is connected in the **custom-mode**,

The **SC\_Mode**="010" is crucial because it sets the operating mode. The **SC\_Mode**="111" is used as the normal run operation after the slow-control phase,

- **SEED\_VECTOR** is a 4096-bit vector that provides the 256 MPs with a 256 different stimuli composed of 16 bits each. The stimuli are read from a text file, whose single lines correspond to a coded single stimulus, and are applied whenever a **TEST** rising edge is provided. Thus, if  $n$  **TEST** pulses are provided,  $256 \times n$  lines of the file of stimuli are read, and  $256 \times 16$ -bit patterns are provided to the 256 MPs  $n$  times. These stimuli can be seen or not depending on the MPs' **Latch-Enable** status. In fact, if this is low, it means that the MP is frozen and blind to new hits,
- **Apply\_Hit** and **Apply\_Hit\_Comp** are signals, considered only for **digital-mode**, which force the stored dummy pixels to be copied into mirror registers (called **Latches** even if hardwired with FFs) to be read out as they were output latches of the pixel sensors. The process activates on the rising edge of **SC\_clk**. Thus, when the **Apply\_Hit/Apply\_Hit\_Comp**="10" configuration is seen at the **SC\_clk** rising edge, the **Dummy-Pixels** (in their 0/1 status) are copied to the 256 **Latches**, while the "01" configuration of **Apply\_Hit/Apply\_Hit\_Comp** indicates that a complemented version (1→0 and 0→1) of the **Dummy-Pixels** must be stored on the **Latches**. Once the "10" or "01" **Apply\_Hit/Apply\_Hit\_Comp** configurations are applied, the readout logic starts if it was standing or continues its matrix-sweep if it was on running. Once the whole matrix is swept, the read process stops. Then, a "00" configuration is required to let the column sweeping process starts again from the left-most column. This allows for avoiding immediate re-readout of the hits,
- **BC** is the Bunch-Xing signal, considered asynchronous, which forces an internal time-stamp register into being copied as a time-mark for any of the MPs that contain at least one hit (MP's FastOr is high). This is carried out after the BC is synchronized to the global readout clock **RDclk**. The **BC** signal is masked by the **SC\_Mode**=0,1,2,4; in other words, during slow-control phases, the **BC** external signal is not seen,
- **MasterLatchEnable** is an asynchronous signal inserted with an *and* logic function along with the internal 256 MPs' **Latch-Enable** signals and with the 256 **Mask** registers. In other words, eventually, the **Latch-Enable** signals that enter the MPs are given by:

$$\text{Latch-Enable}(i) \leftarrow \text{LatchEnableNMatrix}(i) \text{ and } \text{Mask}(i) \text{ and } \text{MasterLatchEnable}$$

where **Mask**(i) are the 16 masks loaded via slow-control and **LatchEnableNMatrix**(i) are the 16 signals provided by the readout control unit for the 16 MPs. These latter signals are synchronized with the rising edge of **SC\_clk**. Masked MPs have **Mask**(i) = 0. If **MasterLatchEnable** is low all the **Latch-Enable**(i) are forced to '0' and the MPs to which they belong are frozen and kept blind until **MasterLatchEnable** goes high again,

- **RDclk** is the 40 MHz global clock,
- the three ports with the prefix **SC\_**, (**SC\_Mode**, **SC\_In** and **SC\_clk**) stand for slow-control inputs and are used to configure the circuit,
- **Reset** is a hard-reset that resets all the internal registers to a predefined state as soon as it is seen and synchronized via **RDclk**: basically 0 for the registers and 1 for the masks, **digital-mode** for operating mode. It must last high for more that one **SC\_clk** and **RDclk** period to take effect,
- **SC\_clk** is an up to 40-MHz slow-control clock. It is asynchronous with respect to **RDclk** when it is used to load the internal registers as the **Dummy-Pixels** in **digital-mode** or the masks **Mask**(i) in both modes. When a given configuration of hits has been loaded into the **Dummy-Pixels**, and they have to be copied into the **Latches** via **Apply Hit/Apply Hit Comp** couple, the **SC\_clk** clock must be run with the same frequency of the **RDclk** clock. This is why, in **digital-mode**, the dummy matrix is updated on the rising edge

of **SC\_clk** while it is frozen, read and reset on the rising edge of **RDclk**. Using two different clock frequencies may lead to unpredicted states. Let's say that **SC\_clk** frequency must be not lower than **RDclk** during **digital-mode** matrix readout. In **custom-mode** instead, the **SC\_clk** may be run at any frequency during configuration.

- **TEST** is an asynchronous non-real port used to specify when the **SEED\_VECTOR** must be read and its values must generate the simulated hits, in **digital-mode**,
- **SC\_Out** is an 8-bit port used to either shift out the **dummy-pixels** when **SC\_Mode** = "000" or to readout the **time-stamp** internal value of the BC-counter when **SC\_Mode** = "011", at the rising edge of **SC\_clk**,
- **DataOut** is a 21-bit port. It is the output data bus with the format described in session 2.2,
- **End\_of\_Scan** is a single bit that indicates the end of sweeping phase. In other words, a pulse on this port confirms that a total sweep over the columns of pixels, from left to right, has finished,
- **TS\_4\_End\_of\_scan** is an 8-bit port used to output the **time-stamp** counter value related to the **End\_of\_Scan**. It should be noted that this port, along with **End\_of\_Scan**, can provide data before the internal queuing system has been emptied. In more detail, these ports indicate that a given sweep with a given TS is finished and all the data related to earlier TSs are for sure closed. However, the data can still be into the barrels. As the entire queuing system can hold up to 160 hits, the system can take up to 160 clock periods to output the data. Thus, if the occupancy is very high (not a great physics sense), these ports do not provide a significant contribute. Conversely, if the queuing system is not congested, the data exit very soon once the matrix is swept and the above ports make their sense,
- **Fast\_Or\_Global** is a single bit that indicates the status of the MPs' **FastOr** output pins. In particular it is a global *nor* logic function of the 256 **FastOr** signals after the masking operation. By following the above description for MasterLatchEnable,

**Fast\_Or\_Global** <= *nor*<sub>(0-to-255)</sub> [**FastOr**(i) and **Mask**(i)].

### 3.1 Design Bugs

---

**N.B.** Due to a design error the `Fast_Or_Global` is only sensitive to the first 64 `FastORs<63:0>` instead of all 256.

**N.B.** Due to design specification misunderstanding the `MasterLatchEnable` not only masks the 256 `LatchEnable` signals that enter the MPs, but also masks the correspondent input ports of the `Time-Stamp` and `MC-Address-Decoder` circuits.

**N.B.** Due to a design error the `MC_Address_Decoder`, in case there is one or more masked MP, stops onto the MCs to which these MPs belong to, at any sweep. Thus, as any MC read out lasts 5 `RDclk` periods, the readout time is increased of 5 `RDclk` periods for any MC that includes masked MPs. This delays the output of data and the dead-time of the pixels even though any individual hit is not lost.

In particular, to foresee the time when a given hit is output to the ASIC's port, depending on the BC and on the masks, the following scheme must be adopted. In case there are some masked MPs, the internal readout logic scans the MCs even without BC signal, consuming 5 `RDclk` periods per MC that contains masked MPs. At the time the BC is internally synchronized, i.e. two `RDclk` periods after it raises on the external pin, the scanning is on a given MC. Then, from the next MC to the right the logic is capable to put the hits in output. For example, if at a given time the readout scans `MC=20` when BC is internally synchronized, first hits on output will belong to `MC=19` if there are hits on that MC, or in any case will be the first ones with MC index lower than 20.



### 3.3 Internal Registers

For its operation modes the circuit has some internal registers that must be loaded via the slow-control.

REGISTER	# OF BITS	OPERATING MODE	NOTES
Dummy-Pixels	4096	digital	- Dummy MAPS, default at '0' at <b>Reset</b> = '1', - Loaded with <b>SC_Mode</b> = "000", 8-bits at a time - Scrambled with <b>SC_Mode</b> = "101" or "110"
Latches	4096	digital	- Dummy MAPS' output latches, default at '0' at Reset - Updated on rising edge of <b>SC_clk</b> when <b>Apply_Hit/Apply_Hit_Comp</b> = "10", "01"
MP-Mask	256	both	- no-mask = '1', default at '1' at <b>Reset</b> = '1', - Loaded with <b>SC_Mode</b> = "001", 8-bits at a time
Row-Mask	32	both	- no-mask = '1', default at '1' at <b>Reset</b> = '1', - Loaded with <b>SC_Mode</b> = "001", 8-bits at a time
Dummy_Actual	1	both	- '1' = <b>digital-mode</b> , default at <b>Reset</b> = '1', - '0' = <b>custom-mode</b> , - Loaded with <b>SC_Mode</b> = "010", <b>SC_In</b> (1)

### 3.4 Stand\_By

The entire circuit, whenever is not able to store the data provided by the **Sparsifier** or **Sparsifier-OUT** circuits, stops the sweeping of the pixel columns. This happens when the number of free location in the 4 **Barrel** or **Barrel-Final** circuits is smaller than the number of data that are going to be stored. In such a situation the readout stops for 32 (**Barrel/Barrel-Final** depth) **RDclk** periods until at least the queuing system of the **Barrel-Final** circuits is empty for sure.

In this design the **Stand\_By** condition can occur if at least one of the following two different events occur:

- the average hit-rate of the whole **Matrix** (lit pixels over unit time) is slightly higher that the throughput (1 hit per **RDclk** cycle) and only the **Barrel-Final** is over-loaded. This means that each single **Barrel** can afford the rate over ¼ of the **Matrix** and the following inequality is satisfied:

$$\frac{1}{4} * \text{average-hit-rate} < \text{throughput (i.e. 1 hit per RDclk cycle)} < \text{average-hit-rate}$$

In this case is only the **Barrel-Final** that forces the **Stand\_By** condition,

- the average hit-rate is sufficiently higher that the throughput (1 hit per **RDclk** cycle) so that any of the **Barrels** is over-loaded. This means that neither of the **Barrels** can afford the rate over ¼ of the **Matrix** and the following inequality is satisfied:

$$\frac{1}{4} * \text{average-hit-rate} > \text{throughput (i.e. 1 hit per RDclk cycle)}$$

In this case is only the first **Barrel** that reaches the full condition and forces the **Stand\_By** signal.

However, for the **Readout** viewpoint, no matter who interrupted the scan of the **Matrix** via a **Stand-By** request. The scan is halted for 32 **RDclk** periods.

### 3.5 Latency

After a **BC** signal is detected by a **RDclk** rising edge, this is synchronized by next two **RDclk** periods and, if some hits are present, the entire readout architecture takes at least 8 **RDclk** periods to out the first valid data. In other words, a latency of 8 clock periods can be retained as the minimum time before a valid output is provided, after the internally synchronized **BC** signal. Thus, the whole latency of the **DataOut** port, with respect to the **BC** signal, is 10 clock periods as the logic takes 2 periods to synchronize the **BC**. Two considerations must be said. First, other one-to-three periods have to be added depending on which column inside the **MP** contains hits, 3 periods for the rightmost and 0 for the leftmost. Second, according to the third N.B consideration explained in pag. 11, any **MC** that contains masked **MP** introduces 5 clock periods as delay to the output data time.

### 3.6 Simulations

Below follow two plots corresponding to the two operating modes. The plots have just a graphical purpose and, of course, are not to be clearly understood.

Fig. 7 shows the circuit operating in the **custom-mode**. By following the picture, from left to right, it can be seen a slow-control stage where the internal registers are loaded. After a given time, some **TEST** pulses are provided, then **BC** signal starts and, consequently, the **DataOut** bus begins to out valid data. Three **End\_of\_Scan** pulses are also visible.

Fig. 8 shows the circuit operating in the **digital-mode**. By following the picture, from left to right, it can be seen a slow-control stage that here lasts a longer time as all the internal registers correspondent to the **Dummy-Pixels** that have to be loaded. After a give time, some **BC** edges are provided (here does not make sense to provide the **TEST** pulses) and, consequently, the **DataOut** bus begins to out valid data.

Three **End\_of\_Scan** pulses are also visible. On the right hand side of the picture there is a **soft-reset** request, by means of the **SC\_Mode** value 4. In Fig. 8 the two clocks, **RDclk** and **SC\_clk** have the same frequency.

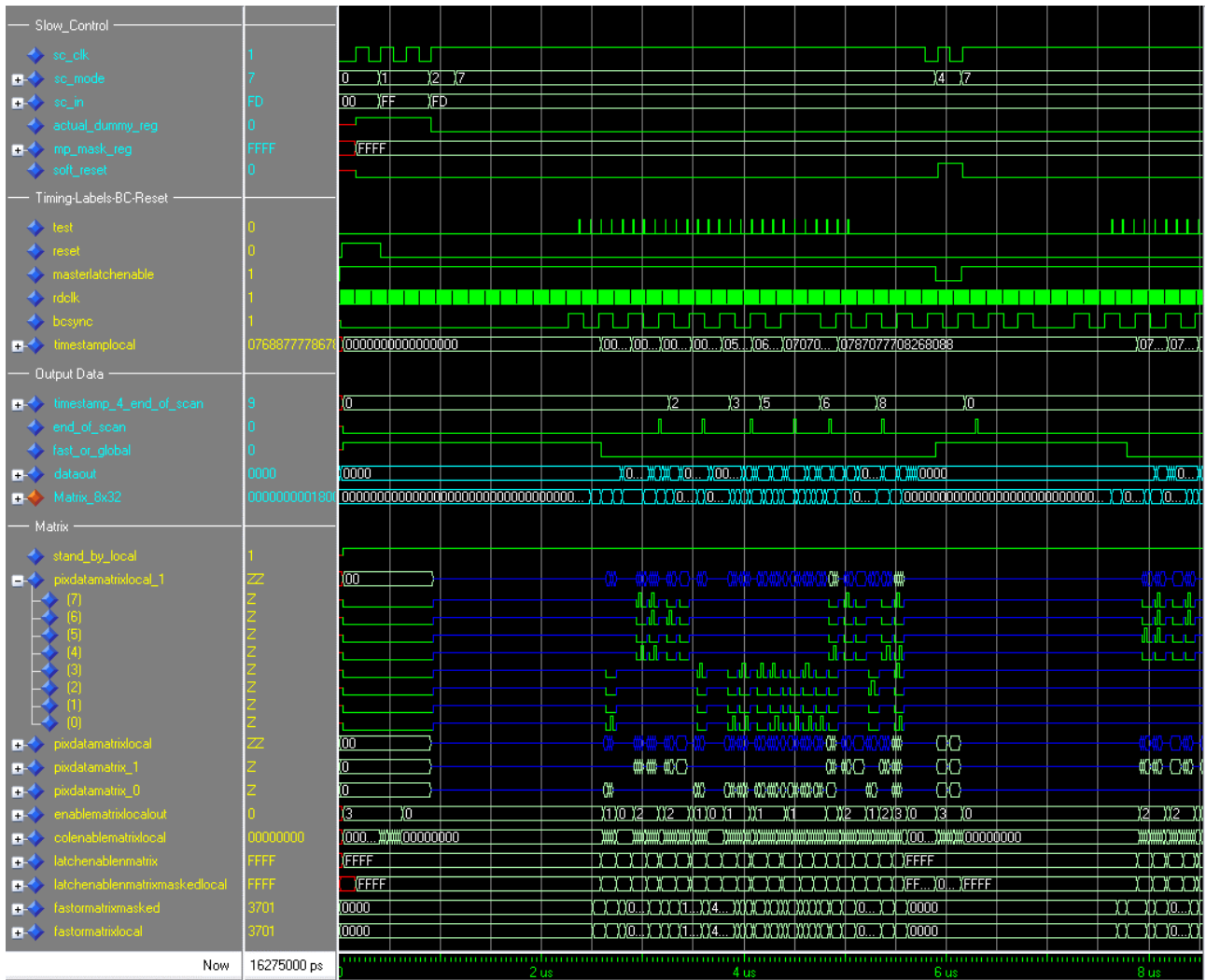


Figure 7: Simulation of the entire circuit in the custom-mode



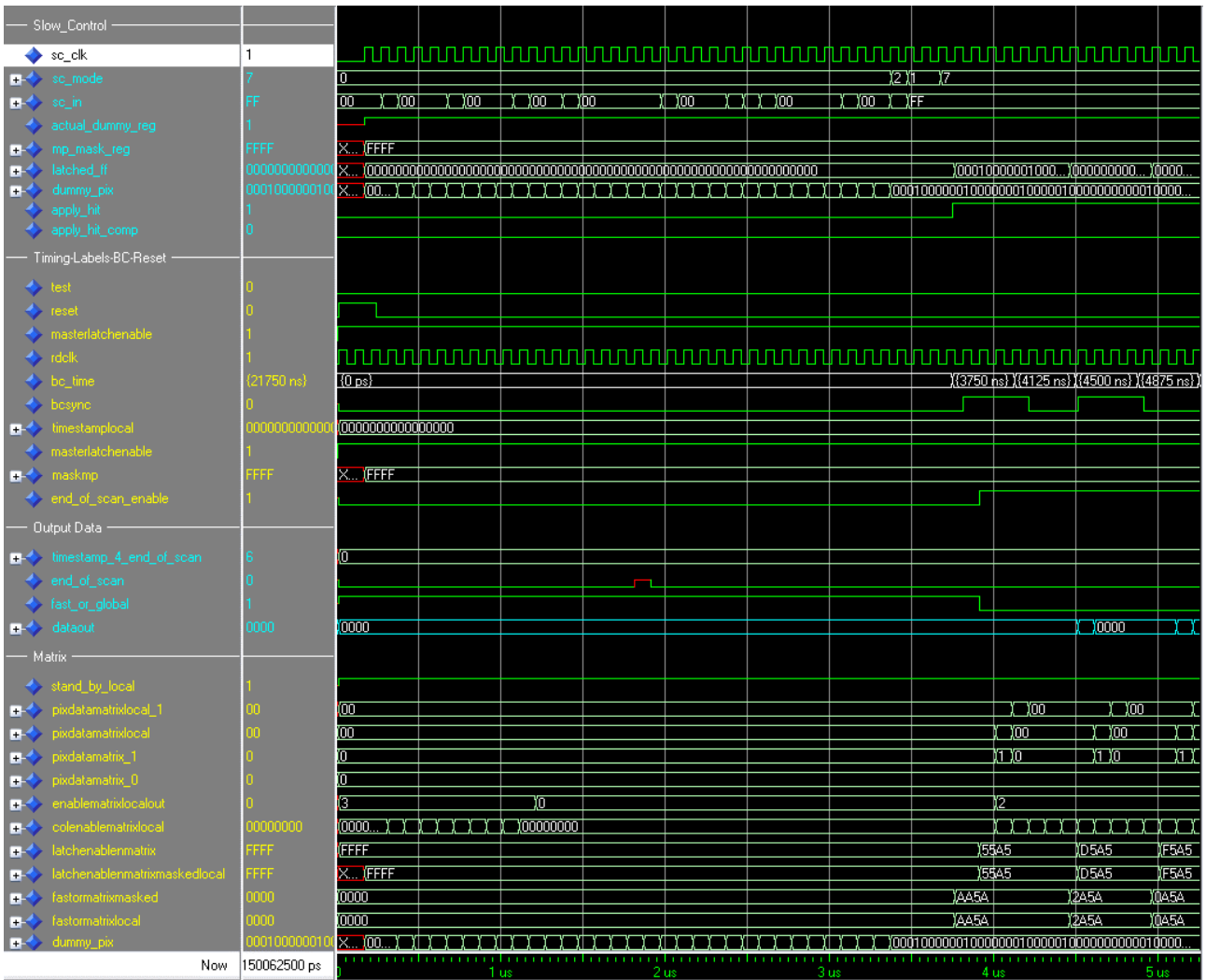


Figure 8: Simulation of the entire circuit in the **digital-mode**

## 4. Time-Stamp

The figure shows the I/O ports of the **Time-Stamp** circuits.

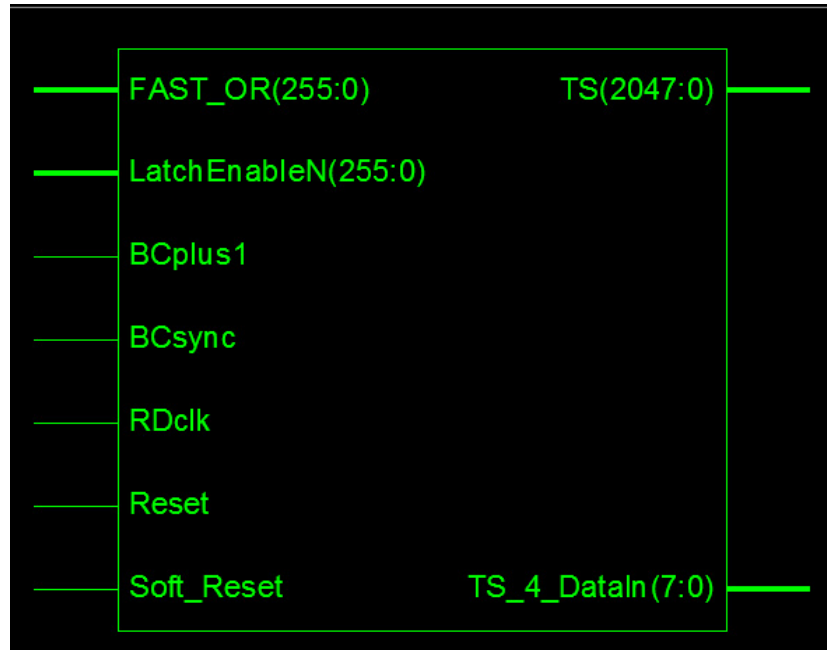
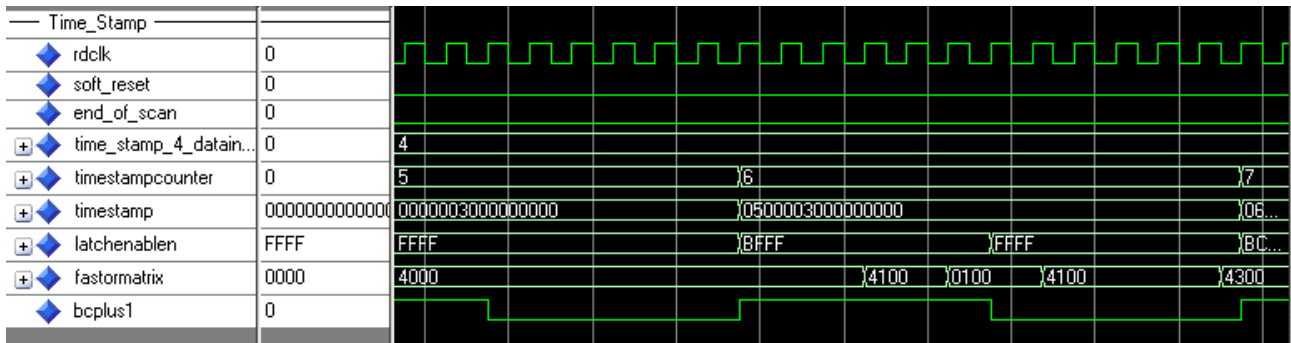


Figure 9: Time-Stamp

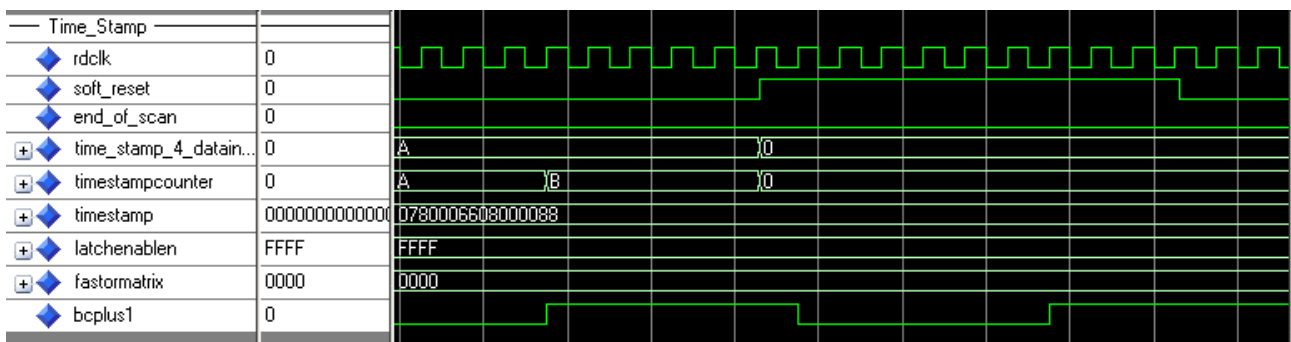
Here follows a description of the I/O ports:

- **FAST\_OR** is a 256-bit port that reads the relative **FastOrMatrix Matrix** output, after being masked depending on the internal 256 MP-mask registers,
- **LatchEnableN** is a 256-bit port that reads the relative **LatchEnableNMatrix Latch-Enable** output, after being masked depending on the internal 256 MP-mask registers and the external **MasterLatchEnable** port,
- **Bcplus1** is the bunch-Xing signal described in the **Latch-Enable**,
- **BCsyn** is the bunch-Xing signal described in the **Latch-Enable**,
- **RDclk** is the external clock described in the **Readout**,
- **Reset** is the external reset described in the **Readout**,
- **Soft\_Reset** is an external requested reset to reset the **time-stamp** register down to 0,
- **TS** is a 2048-bit output port that provides the status of the time-mark of all the 256 MPs (256 by 8-bits of the time-mark),
- **TS\_4\_dataIn** is an 8-bit port that provides the proper time-mark related to the **End\_of\_Scan** signal.



**Figure 10: Simulation of the Time-Stamp**

Fig. 10 shows the circuit simulation. The **TimeStamp**, **Time\_Stamp\_4\_dataIn** and **End\_of\_Scan** signals are valid immediately after **Bcplus1**. Also the internal **Time-Stamp** counter is shown and it can be seen that it updates on the **Bcplus1** rising edge.



**Figure 11: Simulation of the Time-Stamp**

Fig. 11 shows the simulation when a **Soft\_Reset** occurs: the internal **time-stamp** counter is reset to 0.

#### 4.1 Latency

After a **BCsyn / Bcplus1** couple “10” is detected by a **RDclk** rising edge, the **time-stamp** counter, the **TimeStamp** and the **Time\_Stamp\_4\_DataIn** ports are updated. Total latency from the asynchronous **BC** is 2 **RDclk** periods after **BC** is detected. The latency is 2 periods.

## 5. Barrel

The figure shows the I/O ports of the **Barrel** circuits. This circuit provides a queue for the output data. As the entire architecture reads out at most one valid 21-bit word at a time, i.e. at a **RDclk** cycle, in case more than one hit is read in parallel from the matrix, the exceeding hits must be temporarily held into a FIFO-like memory. This memory is a barrel that can be written with 1 to 8 24-bit words, and can be read one location at a time. The barrel depth is 32: all in all it has 32 locations of 24-bit words even though just a subset of the overall bits are used. Each **Barrel** reads only the data that were originated from two MRs and, eventually, 4 barrels are designed to work in parallel to read out up to 32 pixel rows at a time as shown in figure 3.

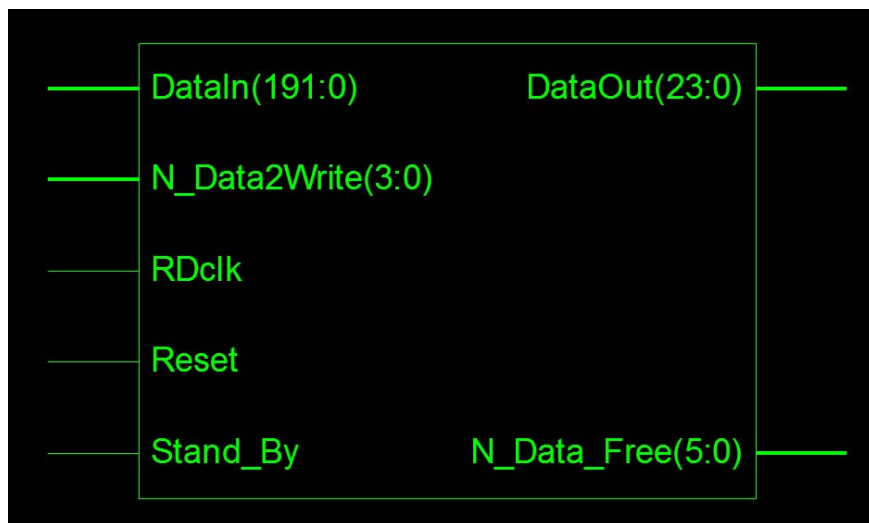


Figure 12: Barrel

Here follows a description of the I/O ports:

- **DataIn** is a 192-bit port that reads the related **DataIn** output of the **Sparsifier**. 192 bits are to be considered the maximum width of the valid data. In fact, it is 24-bit times 8 hits. Here only 21-bits out of 24 are really used (the synthesizer removed unused registers),
- **N\_Data2Write** is a 4-bit port that reads the number of valid data to pack together into the barrel. The **N\_Data2Write** is provided by the readout control unit,
- **RDclk** is the external clock described in the **Readout**,
- **Reset** is the external reset described in the **Readout**,
- **Stand\_By** is signal to freeze the scan of the matrix,
- **N\_Data\_Free** is a 6-bit port that send to output the number of free locations of the **Barrel**,
- **DataOut** is a 24-bit port. It is the output data bus with the following format:

<6-bit-X><Pixel-Row><Pixel-Column-within-MP><Macro-Column-Address><Time Stamp>

- Pixel Row is a 3-bit bus (only 2 MRs): 0 to 7 as the row moves from bottom to top,
- Pixel Column within MP is a 2-bit bus: 0 to 3 as the MP column moves from right to left,
- Macro Column Address is a 5-bit bus: 0 to 31 as MC moves from right to left,
- Time Stamp is a 8-bit time mark: 0 to 255 incremented upon BC rising edge,
- XX are unused bits.

Only 20 out of 24 bits are used to which one extra DataValid bit will be added via the Readout control unit along with the 2-bit row-extended address to account for the **Barrel**



## 6. Barrel-Final

The figure shows the I/O ports of the **Barrel-Final** circuit. This circuit provides a queue for the output data that come from the 4 parallel **Barrels** through the 4 **Sparsifier** and the **Sparsifier-Out** circuits as shown in figure 3. As this architecture reads out at most one valid 21-bit word at a time, i.e. at a **RDclk** cycle, in case more than one hit is read in parallel from the matrix, the exceeding hits must be temporarily held into a FIFO-like memory. This memory is a barrel that can be written with 1 to 4 24-bit words (4 **Barrels**), and can be read one location at a time. The barrel depth is 32: all in all it has 32 locations of 24-bit words even though just a subset of the overall bits are used.

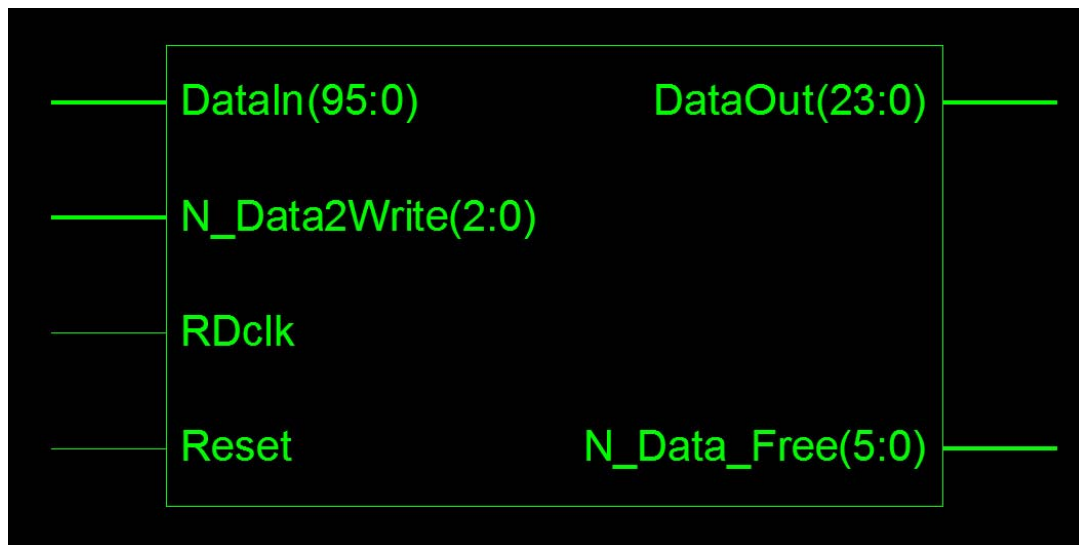


Figure 14: Barrel-Final

Here follows a description of the I/O ports:

- **DataIn** is a 96-bit port that reads the relative **DataIn** output of the **Sparsifier-Out**. 96 bits are to be considered the maximum width of the valid data. In fact, it is 24-bit times 4 hits. Here only 21-bits out of 24 are really used: the synthesizer removed unused registers,
- **N\_Data2Write** is a 3-bit port that reads the number of valid data to pack together into the barrel. The **N\_Data2Write** is provided by the readout control unit,
- **RDclk** is the external clock described in the **Readout**,
- **Reset** is the external reset described in the **Readout**,
- **N\_Data\_Free** is a 6-bit port that send to output the number of free locations of the barrel,
- **DataOut** is a 24-bit port. It is the output data bus with the following format:

<4-bit-XX><Pixel-Row><Pixel-Column-within-MP><Macro-Column-Address><Time Stamp>

- Pixel Row is a 5-bit bus: 0 to 31 as the row moves from bottom to top,
- Pixel Column within MP is a 2-bit bus: 0 to 3 as the MP column moves from right to left,
- Macro Column Address is a 5-bit bus: 0 to 31 as MC moves from right to left,
- Time Stamp is a 8-bit time mark: 0 to 255 incremented upon BC rising edge,
- XX are unused bits.

Only 20 out of 24 bits are used to which one extra DataValid bit will be added via the Readout control unit.



## 7. Latch-Enable

The figure shows the I/O ports of the **Latch-Enable** circuits.



Figure 16: Latch-Enable

Here follows a description of the I/O ports:

- **COL\_ENA** is a 128-bit port that reads the relative **COL\_ENA** port of the **MC-Address-Decoder**,
- **FAST\_OR** is a 256-bit port that reads the relative **FAST\_OR** output of the **Matrix**, after being masked depending on the internal 256 Mask registers,
- **OUT\_EN** is a 8-bit port to either enable the readout of a given MP or its reset cycle. It is high when the **COL\_ENA** has one subsection of 4 bits at one of the following values: “0001”-“0010”-“0100”-“1000”-“1001”,
- **BC** is the external Bunch-Xing signal described in the **Readout**,
- **RDclk** is the external clock described in the **Readout**,
- **Reset** is the external reset described in the **Readout**,
- **LATCH\_ENABLE** is a 256-bit output port that freezes those MPs that have **FAST\_OR** high, once the **BC** rising edge is detected. The port is also masked depending on the **MP-mask** register,
- **BCsync** is the Bunch-Xing signal **BC** after being synchronized to the global read clock **RDclk**,
- **Bcplus1** is the same as **BCsyn** after being synchronized and delayed one clock cycle **RDclk**. Together with **BCsyn** is used to detect the rising edge of **BC**.

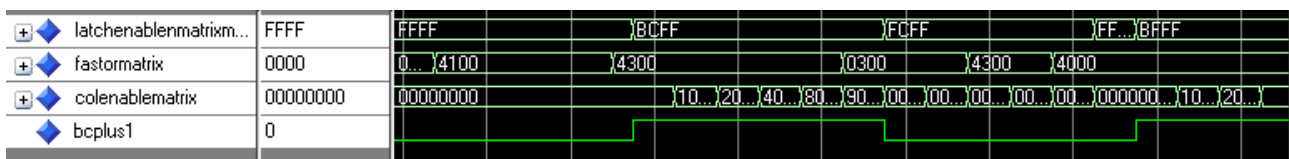


Figure 17: Simulation of the Latch-Enable



## 7.1 Latency

After the **Bcplus1** rising edge occurs, the **BCsync** is updated at the first **RDclk** rising edge. **BCplus1** and **LATCH\_ENABLE** ports is rather updated 2 periods later. The latency is 2 periods.

## 8. MC-Address-Decoder

The figure shows the I/O ports of the **MC-Address-Decoder** circuits. This provides the address of the column of pixels while the matrix readout is ongoing. It stops only over the MCs that have at least one hit. The readout of a MC lasts 5 **RDclk** periods, 4 to read out the columns inside the MC and 1 to reset the MPs just read. It can last more periods if the readout enters a **Stand\_By** condition.

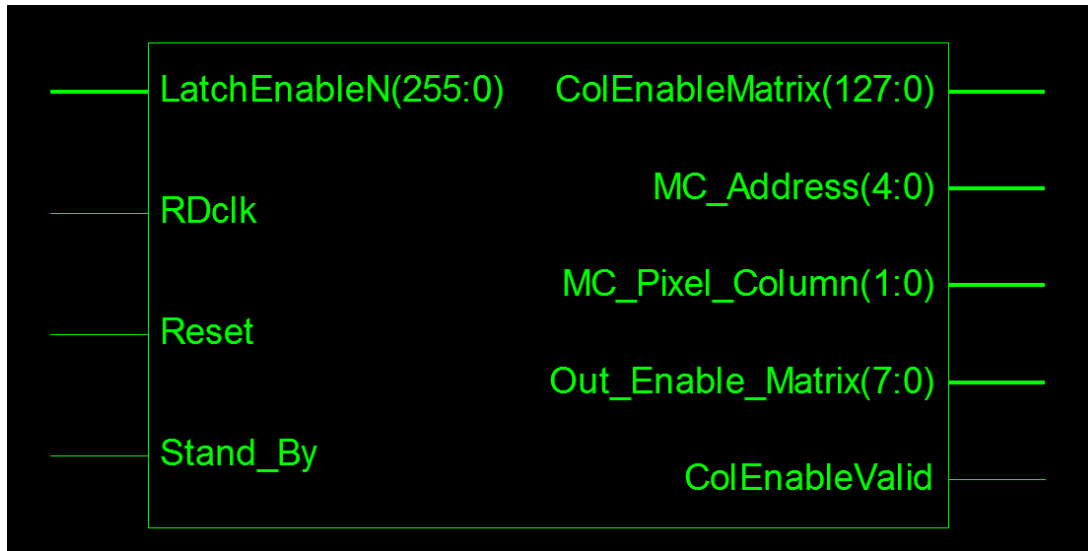
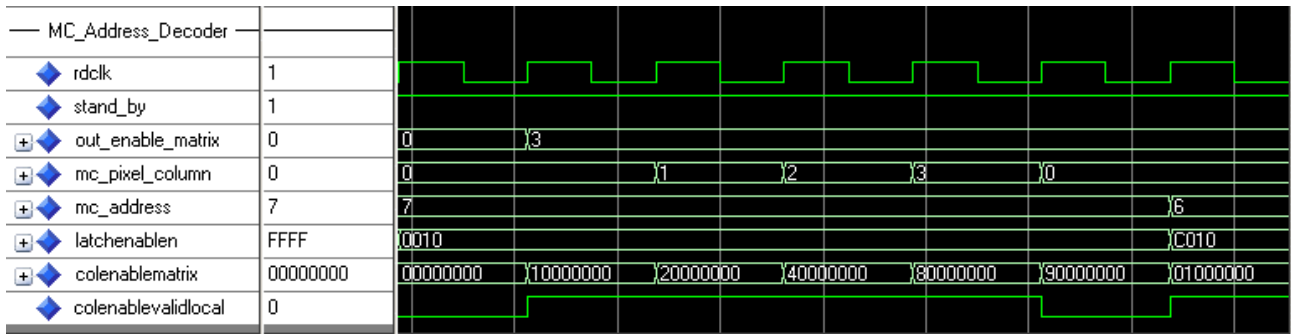


Figure 18: MC-Address-Decoder

Here follows a description of the I/O ports:

- **LatchEnableN** is a 256-bit port that reports the freeze status of those MPs which have **FAST\_OR** high, once the **BC** rising edge is detected. The port is also masked depending by the internal 256 MP-mask register,
- **RDclk** is the external clock described in the **Readout**,
- **Reset** is the external reset described in the **Readout**,
- **Stand\_By** is signal to freeze the scan of the matrix,
- **ColEnableMatrix** is a 128-bit port that enables the scan of the matrix. For each MC, it assumes the following states in the following order: “0001”-“0010”-“0100”-“1000”-“1001”. The first 4 states enable the reading of the single pixel columns of the MC, from right to left. The last state (“1001”) is the reset of the MPs, which belong to that MC, that have been previously frozen,
- **MC\_Address** is a 4-bit port that holds the address code of the MC, ranging from 31 to 0 as the MC address moves from left to right,
- **MC\_Pixel\_Column** is a 2-bit port that holds the address code of the column of pixels inside the active MC, ranging from 3 to 0 as the pixel column address moves from left to right,
- **Out\_Enable\_Matrix** is a 8-bit port that enables either the readout of a given MP or its reset cycle. It is high when the **ColEnableMatrix** has one subsection of 4 bits at one of the following values: “0001”-“0010”-“0100”-“1000”-“1001”,
- **ColEnableValid** is a signal that indicates if the readout of a given MP is ongoing or not. It is high when **ColEnableMatrix** is “0001”-“0010”-“0100”; it is low wherever else,



**Figure 19: Simulation of the MC-Address-Decoder**

### 8.1 Latency

After the **LatchEnableN** is updated at the next **RDclk** rising edge the address ports is also updated. The latency is just 1 period.

## 9. Dummy-Matrix

The figure shows the I/O ports of the **Dummy-Matrix** circuits.

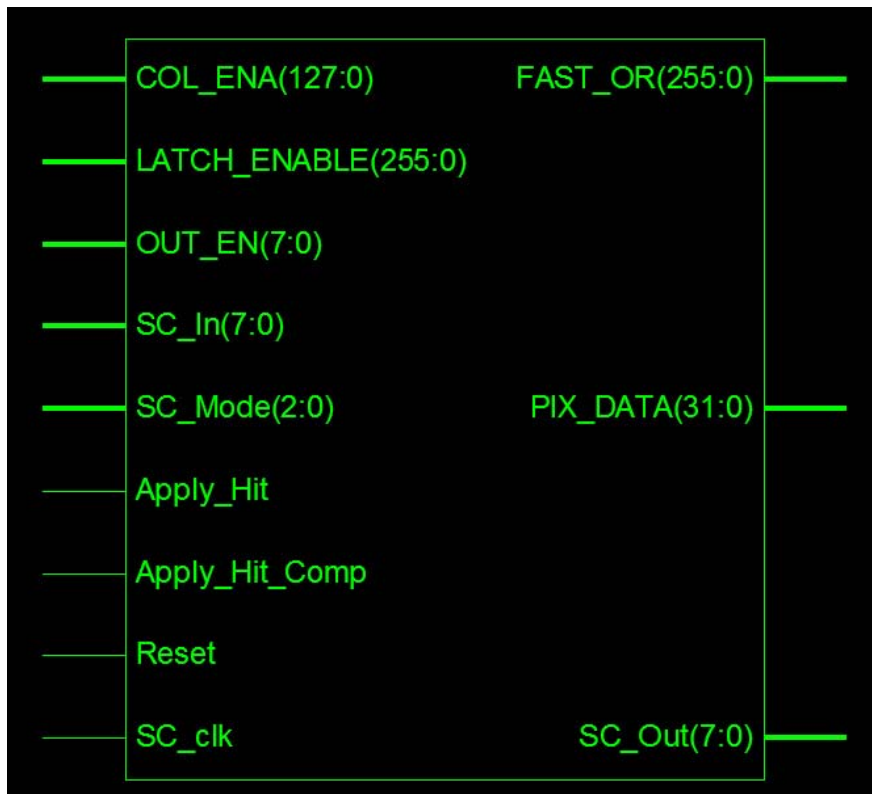


Figure 20: Dummy-Matrix

Here follows a description of the I/O ports:

- **COL\_ENA** is a 128-bit port that reads the relative **COL\_ENA** port of the **MC-Address-Decoder**,
- **LATCH\_ENABLE** is a 256-bit port that reads the relative **LATCH\_ENABLE** port of the **Latch Enable**,
- **OUT\_EN** is the 8-bit port that reads the relative **Out\_Enable\_Matrix** port of the **MC-Address-Decoder**,
- **SC\_In** is the 8-bit external slow-control port described in the **Readout**,
- **SC\_Mode** is the 3-bit external slow-control port described in the **Readout**,
- **SC\_clk** is the external slow-control port described in the **Readout**,
- **Apply\_Hit/Apply\_Hit\_Comp** are the external ports described in the **Readout**,
- **Reset** is the external reset described in the **Readout**,
- **SC\_clk** is the external clock described in the **Readout**,
- **FAST\_OR** is a 256-bit port that provides the *or* signals of the 256 MPs. If one of these is '1' it means that at least one pixel of the MP has a hit. The port is reset back to '0', MP by MP, depending on the **LATCH\_ENABLE** bits, when **COL\_ENA** assumes a given configuration (...1001...),
- **PIX\_DATA** is a 32-bit port that provides the hit configuration of a single column of the whole matrix. If both MPs of the selected MC are to be read out, this port can send up to 32 hits at a time. If one of the two MPs is not to be read out, the corresponding pins of the port are masked to '0' as there were no hits,

- **SC\_Out** is an 8-bit port used to either shift out the **Dummy-Pixels** when **SC\_Mode** = “000” or to read out the **Time-Stamp** internal value when **SC\_Mode** = “011”, at the rising edge of **SC\_clk**. The port goes directly to the output of the chip.

The custom-designed matrix has the same behavior of the **Dummy-Matrix**, except that it is not synchronized with **SC\_clk**, is not reset with **Reset** port and does not see **Apply\_Hit/Apply\_Hit\_Comp** couple nor the slow-control ports. It only shares the **LATCH\_ENABLE**, the **COL\_ENA**, the **OUT\_EN** and provides the corresponding **PIX\_DATA** and **FAST\_OR** output ports.

The **Dummy-Matrix** implementation on the silicon shares the same area as the readout circuit while the custom-designed matrix of MAPS occupies its own silicon area as it is the real sensor of the whole device.

### **9.1 Latency**

After the input ports are updated on the rising edge of **SC\_clk**, the **PIX\_DATA** and **FAST\_OR** output ports are updated asynchronously. There is no extra latency.

## 10. Sparsifier

The figure shows the I/O ports of the **Sparsifier** circuits.

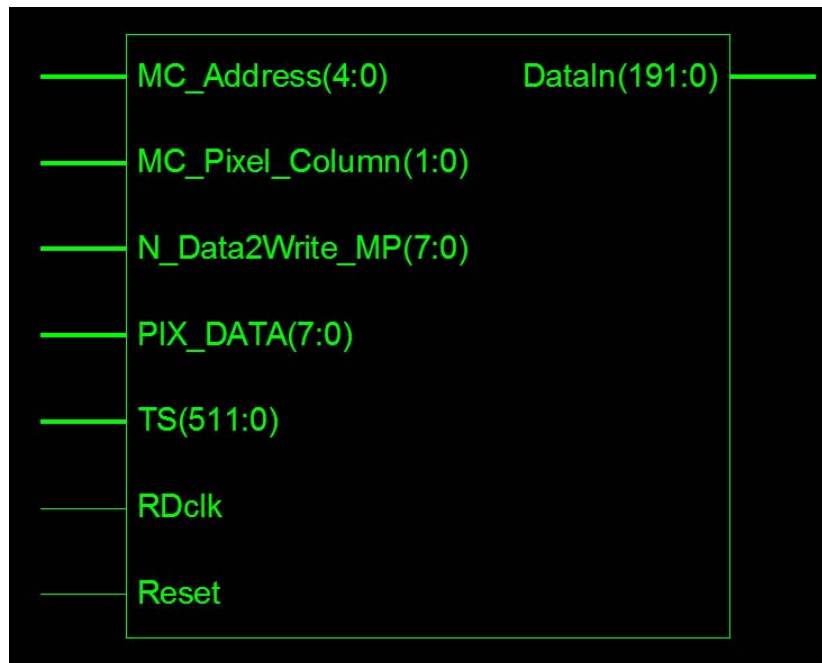


Figure 21: Sparsifier

Here follows a description of the I/O ports:

- **MC\_Address** is a 5-bit port that reads the relative port of the **MC-Address-Decoder**,
- **MC\_Pixel\_Column** is a 2-bit port that reads the relative port of the **MC-Address-Decoder**,
- **N\_DATA2Write\_MP** is an 8-bit port that indicates how many hits have to be considered at the same time as they all belong to the same column of pixels. The information of these hits are then sent to the **Barrel** through the **DataIn** output
- **PIX\_DATA** is the 8-bit port that specifies the hit configuration of the column of pixels.
- **TS** is a 512-bit port that reads the relative port of the **Time-Stamp**,
- **RDclk** is the external clock described in the **Readout**,
- **Reset** is the external reset described in the **Readout**,
- **DataIn** is the output 192-bit port that combines the information of the hits with the **time-stamp** into an up to 192 bits. In most cases the bits are redundant but, in case 8 hits are read within the same column of pixels, after being added with the 8-bit **time-stamp** each, they all are used. It should be said that both the **DataIn** port is dimensioned for 24-bit words while, at the moment, only 20 bits are used per words. This is in order to face future bigger matrixes.

### 10.1 Latency

After the input ports are updated, two **RDclk** cycles later the **DataIn** port is also updated. The latency is 2 periods.

## 11. Sparsifier-OUT

---

The figure shows the I/O ports of the **Sparsifier-OUT** circuit. This circuit is combinatorial.

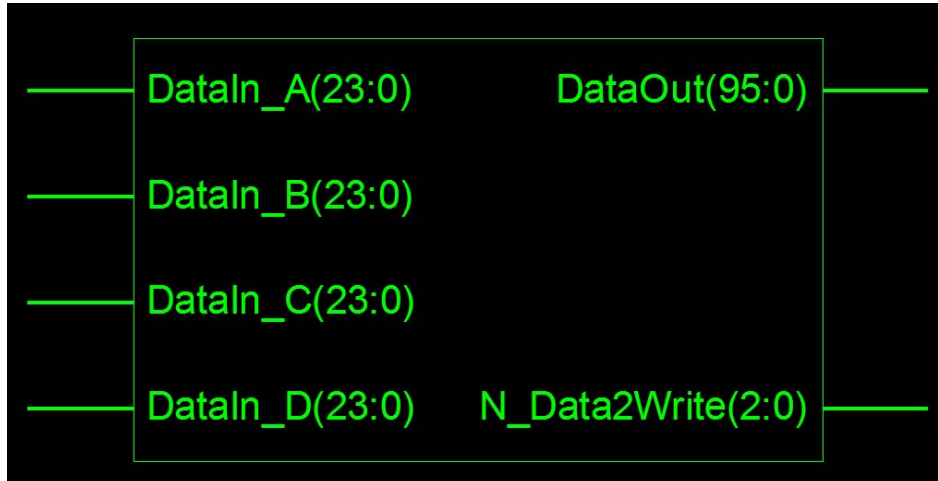


Figure 22: Sparsifier-OUT

Here follows a description of the I/O ports:

- **DataIn\_A/B/C/D** are 4 24-bit input ports that read the output data from the 4 barrels,
- **N\_DATA2Write** is a 3-bit output port that indicates how many hits have to be considered at the same time. The information of these hits are then sent to the **Barrel-Final** through the **DataOut** output port,
- **DataOut** is the output 96-bit port that combines the information of the hits coming from the **DataIn\_A/B/C/D** ports.

### 11.1 Latency

After the input ports are updated, two **RDclk** cycles later the **DataIn\_A/B/C/D** ports are also updated. There is no latency as it is combinatorial.

## 12. Slow Control

The figure shows the I/O ports of the **Slow-Control** circuits.

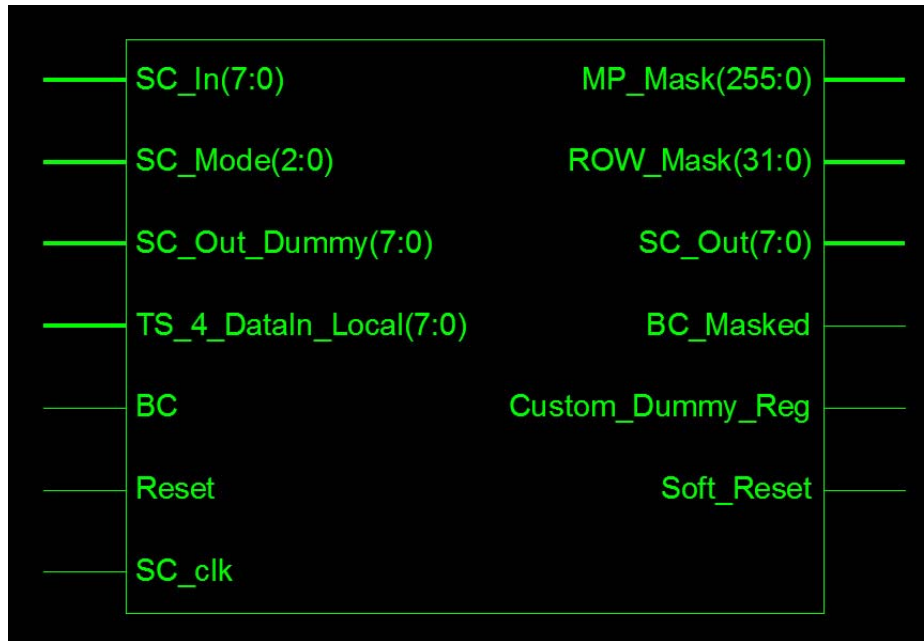


Figure 23: Slow-Control

Here follows a description of the I/O ports:

- **SC\_In** is the 8-bit external slow-control port described in the **Readout**,
- **SC\_Mode** is the 3-bit external slow-control port described in the **Readout**,
- **SC\_Out\_Dummy** is an 8-bit port used to either shift out the **Dummy-Pixels** when **SC\_Mode** = "000" or to read out the **Time-Stamp** internal value when **SC\_Mode** = "011", at the rising edge of **SC\_clk**. This port here goes directly to the output port **SC\_Out** when **SC\_Mode** = "000",
- **TS\_4\_DataIn\_Local** is the 8-bit input port that enters the updated **time-stamp** counter. When **SC\_Mode**="011" this port is saved and copied to **SC\_In** on the rising edge of **SC\_clk**,
- **BC** is the external Bunch-Xing signal described in the **Readout**,
- **Reset** is the external reset described in the **Readout**,
- **SC\_clk** is the external slow-control port described in the **Readout**,
- **MP\_Mask** is a 256-bit port used to mask the 256 MPs. This port masks the **LatchEnableNMatrix** and the **FastOrMatrix** of the **Readout**. The masks are loaded with **SC\_Mode**="001" in 32 **SC\_clk** periods. The default masks at **Reset**='1' are all-at-'1',
- **ROW\_Mask** is a 32-bit port used to mask the 32 rows of pixels. The masks are loaded with **SC\_Mode**="001" in 4 **SC\_clk** periods. The default masks at **Reset**='1' are all-at-'1',
- **SC\_Out** is the external port1 described in the **Readout**,
- **BC\_Masked** is a masked **BC** signal when **SC\_Mode** = 0,1,2 or 4,
- **Custom\_Dummy\_reg** is 1-bit register that indicated the operating mode: 0 for **custom-mode** and 1 for **digital-mode**. It is loaded with **SC\_Mode**="010", by copying **SC\_In(1)**, in one **SC\_clk** period. The default is 1 for **digital-mode** at **Reset**='1',
- **Soft\_Reset** is a signal used to reset only the **time-stamp** counter in the **Time-Stamp**. It is requested with **SC\_Mode**="100" in one **SC\_clk** period. After that, the **SC\_Mode** has to be set again to "111" to normal running operation. It is suggested to not force a **Soft\_Reset** during the scan of valid hits in order to avoid unpredictable output data.



## 12.1 Latency

After the input ports are updated, one **SC\_clk** cycle later the output ports are also updated. The latency is 1 period.

## 13. DEBUG

Test, BC and Output Data saved as formatted text files for debugging purposes.

### Time\_TEST\_actual.txt only for custom operating mode

.....
Test_Time = 1437.5ns
Test_Time = 1543.75ns
Test_Time = 1650ns
Test_Time = 1756.25ns
Test_Time = 1862.5ns
Test_Time = 1968.75ns
Test_Time = 2075ns
.....

### Time\_BC\_dummy.txt & Time\_BC\_actual.txt for digital and custom operating modes

.....
BC_Time = 8550ns
BC_Time = 9300ns
BC_Time = 10050ns
BC_Time = 10950ns
BC_Time = 11700ns
BC_Time = 12450ns
BC_Time = 13350ns
.....

### Time\_DataOut\_dummy.txt & Time\_DataOut\_actual.txt for digital and custom operating modes

Clock Time	Binary Format Out	Hexadecimal Format Out	Decimal Format Out
.....	.....	.....	.....
RDclk_Time = 7637.5ns	Bin = 010110001001	Hex = 1589	Dec = 5513
RDclk_Time = 7662.5ns	Bin = 001110001001	Hex = 1389	Dec = 5001
RDclk_Time = 7687.5ns	Bin = 000110001001	Hex = 1189	Dec = 4489
RDclk_Time = 7787.5ns	Bin = 101011111010	Hex = 1AFA	Dec = 6906
RDclk_Time = 7812.5ns	Bin = 111101111010	Hex = 1F7A	Dec = 8058
RDclk_Time = 7887.5ns	Bin = 000001100000	Hex = 1060	Dec = 4192
RDclk_Time = 8012.5ns	Bin = 110001011110	Hex = 1C5E	Dec = 7262
RDclk_Time = 8037.5ns	Bin = 010011011100	Hex = 14DC	Dec = 5340
RDclk_Time = 137.5ns	Bin = 111000110001	Hex = 1E31	Dec = 7729
RDclk_Time = 8162.5ns	Bin = 000000110001	Hex = 1031	Dec = 4145
RDclk_Time = 8187.5ns	Bin = 000010111111	Hex = 10BF	Dec = 4287
RDclk_Time = 8287.5ns	Bin = 001010010001	Hex = 1291	Dec = 4753
RDclk_Time = 8312.5ns	Bin = 011100010001	Hex = 1711	Dec = 5905
RDclk_Time = 8387.5ns	Bin = 111000000010	Hex = 1E02	Dec = 7682
RDclk_Time = 8412.5ns	Bin = 110000000010	Hex = 1C02	Dec = 7170
RDclk_Time = 8437.5ns	Bin = 101000000010	Hex = 1A02	Dec = 6658
RDclk_Time = 8462.5ns	Bin = 100000000010	Hex = 1802	Dec = 6146
.....	.....	.....	.....

## 14. APSEL4D Layout

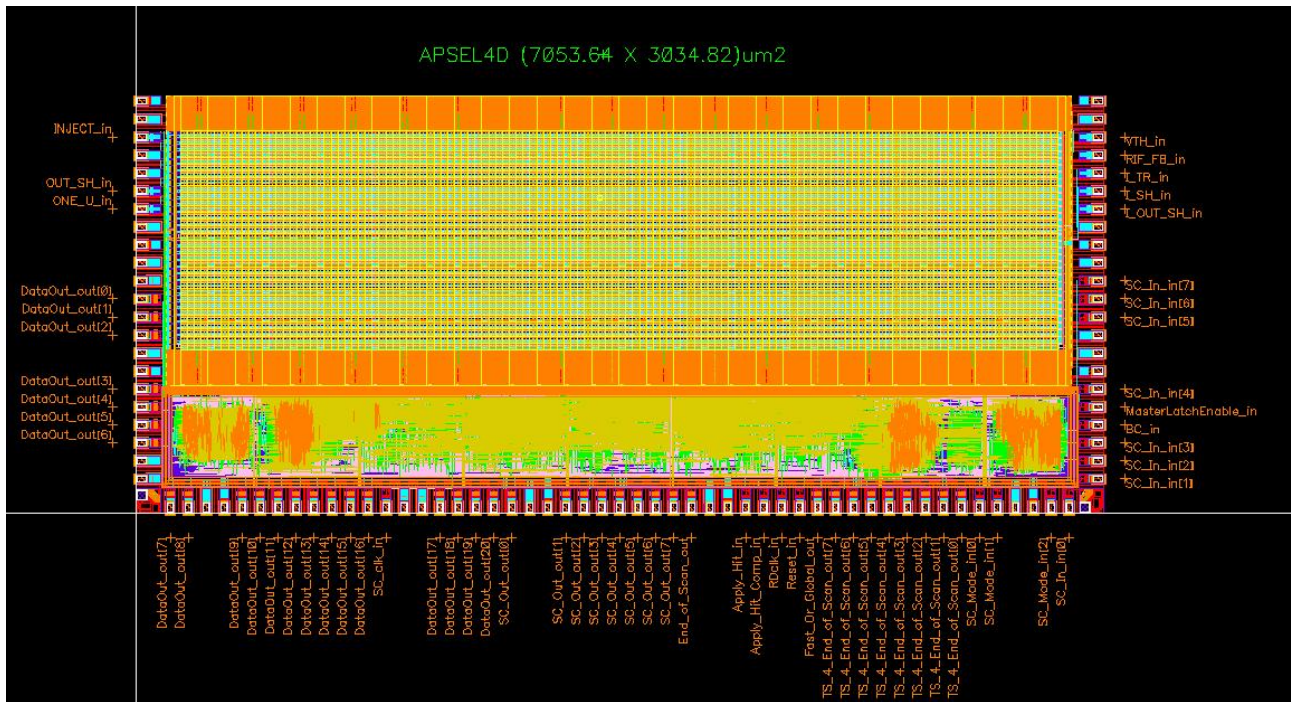


Figure 24: Apssel4D Layout

This is the picture of APSEL4D ASIC, designed with STM 130nm 6M CMOS Technology  
 The whole layout dimension is: **7053.64μm × 3034.84μm**  
 The pitch of the pads is: **131.2μm** on the Left, Bottom and Right sides

## 15. APSEL4D Pinout

<b>22 Left Pads (top-2-bottom)</b>	<b>51 Bottom Pads (left-2-right)</b>	<b>51 Bottom Pads (left-2-right)</b>	<b>22 Right Pads (bottom-2-top)</b>
0 - AGND	22 - Data_Out(7)	48 - SC_Out(5)	73 - SC_In(1)
1 - AVDD	23 - Data_Out(8)	49 - SC_Out(6)	74 - SC_In(2)
2 - INJECT	24 - VDD_CORE	50 - SC_Out(7)	75 - SC_In(3)
3 - SUB	25 - VSS_CORE	51 - End_of_Scan	76 - BC
4 - Vdd_Analog	26 - Data_Out(9)	52 - VDD_CORE	77 - MasterLatchEnable
5 - OUT_SH	27 - Data_Out(10)	53 - VSS_CORE	78 - SC_In(4)
6 - ONE_U	28 - Data_Out(11)	54 - Apply_Hit	79 - AGND
7 - DVDD	29 - Data_Out(12)	55 - Apply_Hit_Comp	80 - AVDD
8 - DGND	30 - Data_Out(13)	56 - RDclk	81 - Vdd_Analog
9 - GND_PERY	31 - Data_Out(14)	57 - Reset	82 - SC_In(5)
10 - VDD_PERY	32 - Data_Out(15)	58 - Fast_Or_Global	83 - SC_In(6)
11 - Data_Out(0)	33 - Data_Out(16)	59 - TS_4_End_of_Scan(7)	84 - SC_In(7)
12 - Data_Out(1)	34 - SC_clk	60 - TS_4_End_of_Scan(6)	85 - SUB
13 - Data_Out(2)	35 - VDD_PERY	61 - TS_4_End_of_Scan(5)	86 - DGND
14 - AVDD	36 - GND_PERY	62 - TS_4_End_of_Scan(4)	87 - DVDD
15 - AGND	37 - Data_Out(17)	63 - TS_4_End_of_Scan(3)	88 - I_OUT_SH
16 - Data_Out(3)	38 - Data_Out(18)	64 - TS_4_End_of_Scan(2)	89 - I_SH
17 - Data_Out(4)	39 - Data_Out(19)	65 - TS_4_End_of_Scan(1)	90 - I_TR
18 - Data_Out(5)	40 - Data_Out(20)	66 - TS_4_End_of_Scan(0)	91 - RIF_FB
19 - Data_Out(6)	41 - SC_Out(0)	67 - Scan_Mode(0)	92 - VTH
20 - GND_PERY	42 - VDD_CORE	68 - Scan_Mode(1)	93 - AVDD
21 - VDD_PERY	43 - VSS_CORE	69 - VDD_CORE	94 - AGND
	44 - SC_Out(1)	70 - VSS_CORE	
	45 - SC_Out(2)	71 - Scan_Mode(2)	
	46 - SC_Out(3)	72 - SC_In(0)	
	47 - SC_Out(4)		

## 16. Test Results

---

**This page is intentionally left blank**

## 17. INDEX

### Index of Chapters:

1. Notes for the readers .....	2
1.2 Version .....	2
2. Introduction .....	3
2.1 The matrix organization .....	4
2.2 The Output Format .....	5
2.3 Block Hierarchy .....	6
2.4 Block Sketch .....	7
2.5 Matrix sweep .....	7
3. Readout .....	8
3.1 Design Bugs .....	12
3.2 Configuration Steps .....	13
3.3 Internal Registers .....	14
3.4 Stand_By .....	14
3.5 Latency .....	15
3.6 Simulations .....	15
4. Time-Stamp .....	18
4.1 Latency .....	19
5. Barrel .....	20
5.1 Latency .....	21
6. Barrel-Final .....	22
6.1 Latency .....	23
7. Latch-Enable .....	24
7.1 Latency .....	25
8. MC-Address-Decoder .....	26
8.1 Latency .....	27
9. Dummy-Matrix .....	28
9.1 Latency .....	29
10. Sparsifier .....	30
10.1 Latency .....	30
11. Sparsifier-OUT .....	31
11.1 Latency .....	31
12. Slow Control .....	32
12.1 Latency .....	33
13. DEBUG .....	34
14. APSEL4D Layout .....	35
15. APSEL4D Pinout .....	36
16. Test Results .....	37
Index of Chapters: .....	38
Index of Figures: .....	39

## Index of Figures:

Figure 1: Apsel4D operating modes: custom-mode and digital-mode .....	3
Figure 2: 4096-pixel Matrix .....	4
Figure 3: Readout Hierarchy .....	7
Figure 4: Readout .....	8
Figure 5: Simulation of the APSEL4D configuration in custom-mode .....	13
Figure 6: Simulation of the APSEL4D configuration in digital-mode .....	13
Figure 7: Simulation of the entire circuit in the custom-mode .....	16
Figure 8: Simulation of the entire circuit in the digital-mode .....	17
Figure 9: Time-Stamp .....	18
Figure 10: Simulation of the Time-Stamp .....	19
Figure 11: Simulation of the Time-Stamp .....	19
Figure 12: Barrel .....	20
Figure 13: Simulation of the Barrel .....	21
Figure 14: Barrel-Final .....	22
Figure 15: Simulation of the Barrel-Final .....	23
Figure 16: Latch-Enable .....	24
Figure 17: Simulation of the Latch-Enable .....	24
Figure 18: MC-Address-Decoder .....	26
Figure 19: Simulation of the MC-Address-Decoder .....	27
Figure 20: Dummy-Matrix .....	28
Figure 21: Sparsifier .....	30
Figure 22: Sparsifier-OUT .....	31
Figure 23: Slow-Control .....	32
Figure 24: Apsel4D Layout .....	35