PROCEEDINGS OF SCIENCE

Efficiency and readout architectures for a large matrix of pixels

A. Gabrielli

INFN and University of Bologna

F.M. Giorgi*

INFN and University of Bologna

E-mail: giorgi@bo.infn.it

M. Villa

INFN and University of Bologna

We present a digital readout architecture for a silicon pixel matrix sensor. It has been developed to cope with high hit rates, above 1 MHz/mm^2 for matrices greater than 80K pixels. This technology can be implemented inside a silicon MAPS device (*Monolithic Active Pixel Sensor*): a high-resolution particle detector which integrates on the same bulk the sensor matrix and the CMOS logic for readout. The architecture proposed is based on three main concepts. In first place the readout of the hits is performed by activating one column at a time; all the fired pixels on the active column are read, sparsified and reset in parallel in one clock cycle. This implies the use of global signals across the sensor matrix, the consequent reduction of metal interconnections improves the active area while maintaining a high granularity (down to 40 μm of pixel pitch). Secondly, the activation for readout takes place only for those columns overlapping with a certain fired area, thus reducing the sweeping time of the whole matrix and reducing the pixel dead-time. Third, the sparsification (x-y address labeling of the hits) is performed with a lower granularity respect to single pixels, by addressing vertical zones of 8 pixels each. The fine-grain Y resolution is achieved by appending the zone pattern to the zone address of a hit. We show the benefits of this technique in presence of clusters.

The main features of the readout architecture are described, then we presents the results obtained with a simulation of the VHDL readout model.

9th International Conference on Large Scale Applications and Radiation Hardness of Semiconductor Detectors,RD09
September 30-October 2, 2009
Florence,Italy

*Speaker.

PoS(RD09)042

High resolution vertex detectors are exploiting by several years the silicon technology, for example hybrid pixel sensors have been used for the ATLAS [1] and CMS [2] trackers at LHC. However the high particle density foreseen in the innermost layers of future experiments, is unaffordable for present front-end architectures [3]. In a flavor factory like SuperB [4] the expected particle flux foreseen at 1 cm of radius is[1] 25M particle/(s $\cdot$ cm^2), and, foreseeing a clustering factor of 4, the hit rate raises to 100 MHz for 1 cm^2 of sensor area.

In this paper we present a digital readout architecture meant to operate on a 80K pixel sensor matrix capable to sustain the high rates expected with a high overall efficiency. The solution developed can be implemented on hybrid pixel sensors and MAPS devices (*Monolithic Active Pixel Sensors*), in both cases there is a CMOS digital logic directly connected to the sensor matrix. In the first solution the CMOS logic is realized as a different chip bump-bonded or vertically integrated on the sensor substrate. In a MAPS device, instead, the sensor and the logic can be built on the same silicon substrate and the readout logic is typically placed as a separate block beside the sensor array [5] [6]. Several research groups are also trying to exploit the new 3D technologies made available by some foundries in order to overcome the challenging conditions, in term of by material budget, resolution and readout speed, required to discover new physics [7] [8]. In this case the process consists in the realization of a monolithic chip made up by several 25μm-thinned silicon tiers interconnected by vias of the order of 1 μm.

1. The Matrix Organization

The matrix considered for our readout architecture is 320×256 pixel wide, for a total of about 81K pixels. This sensor array is divided into 4 smaller matrices (80×256), each one served by a dedicated and independent readout, see Fig. 1. With a 40 μm pitch, the total sensor area covers 1.31 cm^2, but it can reach 2 cm^2 in case of a 50 μm pitch.

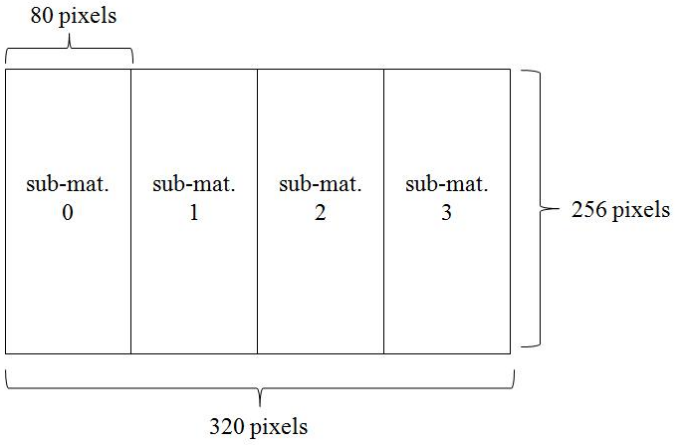


Figure 1: Matrix partitions.

The major problem with high density matrices is the interconnection of the readout block with pixels. Since the readout is typically situated beside the matrix, the total number of pixels

[1] Including a x5 security factor

scales with a quadratic growth with respect to the contact side of the matrix and the digital readout block. The consequent upper-bound in the matrix dimension is given by the limited interconnection density in the contact side of the two blocks.

In order to decrease the number of interconnections between sensor and readout, and hence to increase the matrix dimension, we introduced the concept of Macro Pixel, and active column.

The *Macro Pixel* (*MP*) is an independent group of pixels, with a private *fast-or* line connected to the readout logic. If the fast-or line is activated, it means that at least one of the pixels inside has been fired. In our specific case we considered a MP dimension of 2×8, hence the whole matrix results made up of 5120 MPs.

On the arrival of a Bunch Crossing Clock (*BC*) rising edge, the content of a fired MP is immediately[2] frozen by the logic, which means no more pixels can be turned on, even if a signal over threshold is detected (refer to Fig. 2). BC clock beats time in the experiment and it determines the time granularity of the events recorded, for this reason a counter modulo 256 has been implemented in the readout logic, incrementing on each BC positive edge. When a MP gets frozen, it is associated with the current value of the time counter. Thereafter it waits to be read, reset and reactivated. Each MP has a private freezing signal which is called *Latch Enable*. Timing information is recorded by the readout logic at the moment of a MP freezing.

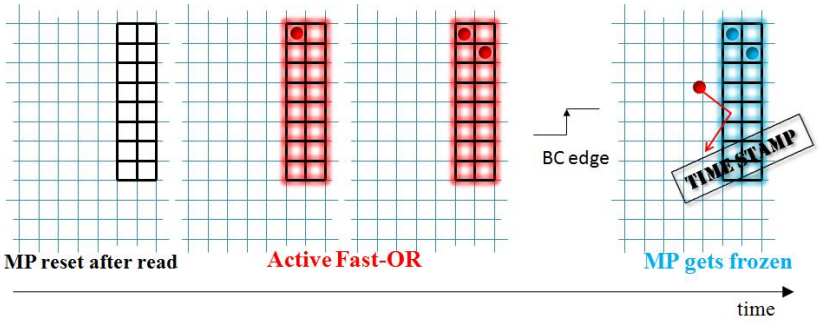


Figure 2: MP working phases.

The *Active Column*, instead, is a concept concerning the readout of the hits. The hits are read through a column-wide common bus, called *Pixel Data Bus*, shared among all the pixel columns. The active column of pixels, which is intended to drive the Pixel Data Bus, is selected by a decoded *Column Enable* bus; in addition there is an *Output Enable* signal selecting the MP rows that need to be read out. The intercept *Column Enable-Output Enable* determines which of the two columns of a MP is driving a segment of the *Pixel Data* bus (Fig. 3).

Once the content of a MP is read out, selecting the correct MP row and its two columns, the MP can be designed to automatically reset all the latches of its pixels. We introduced the *Output Enable* bus in order to be able to choose which MPs on the active column have to be read and reset. In case two MPs, belonging to different time stamps, are fired on the same columns, it is possible to read only the desired one leaving the other waiting for next sweep. This allows, for example, to read out only those MPs tagged with a given time stamp, permitting a time-wise sweep of the matrix.

[2]Fixed and low latency, dependent on read clock period

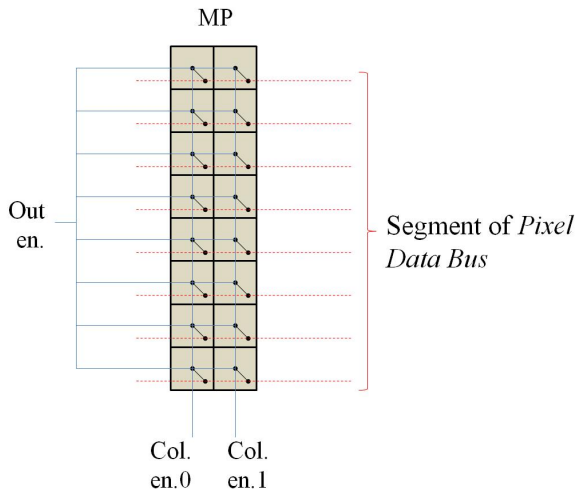


Figure 3: Macro Pixels interconnection scheme.

2. The Readout Logic

All the digital CMOS logic blocks for sparsification and readout of the hits are now described. As previously mentioned, the whole matrix has been divided into 4 vertical sub-matrices (80x256 pixels), each one driven by an independent readout instance. In this way we can exploit the maximum benefits of our architecture which is strongly vertically parallelized adding a further horizontal parallelism. A graphical representation of the components which make up a single readout instance is presented in Fig. 4.

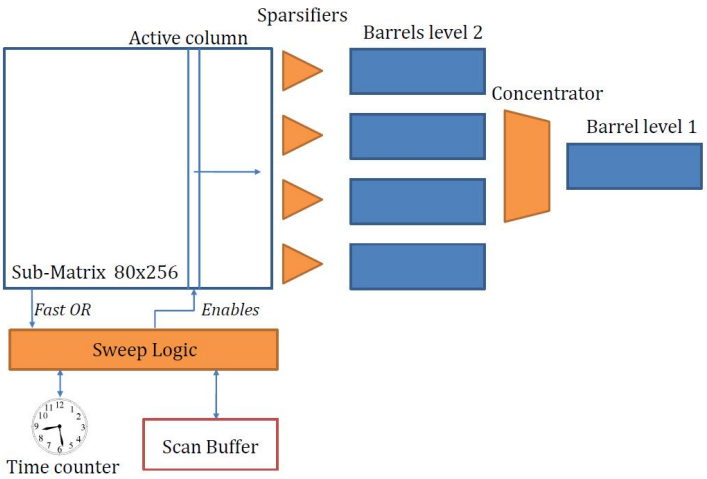


Figure 4: Readout block scheme. The figure does not represent the physical disposition of readout respect to the matrix.

In the top level structure of the chip readout, the instance is replicated 4 times and it is con-

nected to the 4 different sub-matrices. A common final stage is foreseen, running at higher frequencies, meant to drive a fast data bus. The high performance on matrix readout must be in deed supported by a broad-band bus capable to sustain the high data throughput generated.

2.1 The Sweeping logic

This component is responsible for the freezing of fired MPs, the time tagging of the hits, and the sweeping of the active column over the matrix.

Time tagging is provided with a counter modulo 256 incrementing at BC clock rate. When a BC positive edge arrives, all the MPs fired in the current time window are frozen, and a list of them is stored in the *Scan Buffer* together with the current time stamp. The buffer can store up to 8 scan-to-do lists, each one with an 8-bit time label.

The sweep logic pulls from the stack the most dated to-do list and informs the sparsifiers that a new scan is starting related to the corresponding time stamp. Afterwards it proceeds with the active column sweep over the listed MPs.

2.2 The Sparsifier

The current active column drives the *Pixel Data Bus* which is analyzed by the sparsifiers. Their task is to encode the space coordinates of the fired pixels into hit-words. Sparsified data is then stored in a formatted asymmetric FIFO called *Barrel* (ref. to next section).

The sparsifiers encode also the information about the beginning of a matrix scan. When a new scan starts, each sparsifier stores a special word containing the associated time stamp in its adjacent *Barrel*. These words are called SOS (*Start Of Scan*) and divide into bunches the hit-words cropped during different scans.

In the considered sub-matrix, we have 256 rows of pixels and thus a 256-bit wide *Pixel Data Bus*. The developed sparsifier has a 64-bit wide input bus, and it is able to process the whole of it in one clock cycle. In the proposed architecture 4 sparsifiers working in parallel are implemented to cover the full *Pixel Data Bus* (Fig 4).

To profit from possible clustering of hits, the sparsification is not done at the pixel level. The 64-bit sparsifier input bus is divided into 8-bit segments called *zones*. A fired pixel in a certain zone generates a hit-word containing information of the entire zone. A hit-word consists of the XY zone addresses plus the zone hit pattern (see Fig. 5).

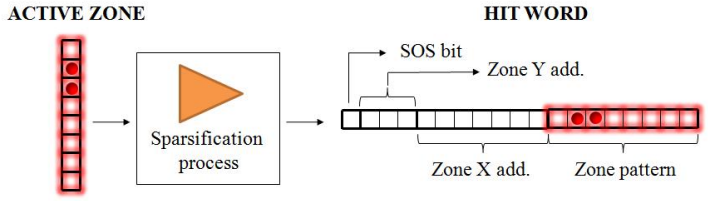


Figure 5: Zone sparsification diagram. When a *Start Of Scan* is encoded, the SOS_bit is set to 1 and the 8 least significant bits contains the *Time Stamp*; other bits are meaningless. Otherwise SOS_bit is 0 and the hit is coded as shown.

According to these, the format of the 19-bit generated hit-word is: SOS_bit + zone_Yaddress[2:0] + zone_Xaddress[6:0] + zone_pattern[7:0].

In principle, all the 8 zones connected to a sparsifier can present fired pixels, thus it was made possible to encode all of them in the same clock cycle.

This technique has been implemented foreseeing the presence of clustered patterns, allowing to reduce the total number of transmitted hits. Some calculations about the benefits brought are shown in next sections.

2.3 The Barrels

The barrels directly connected to the sparsifiers are called *Level 2 Barrels* (B2s) while those collecting data from a whole sub-matrix are called *Level 1 Barrels* (B1s).

The *Barrel* is basically an asymmetric FIFO buffer that can store up to 8 hit-words per clock cycle. Each hit-word refers to a 8-bit zone, then each B2 can store up to 64 fired pixels per clock cycle. Since the complexity of synthesized logic increases fast with the number of hits that can be stored simultaneously, the introduction of the zone technique extends the range of inspected rows of the sparsifiers and barrels with a consequent reduction of the total required components at a fixed fifo depth.

A tree of barrels has been designed, it is composed of 4 B2, driven by their respective sparsifiers, and 1 B1 collecting data from the whole sub-matrix. In between, a smart data concentrator controls the flux of data preserving the time sorting of the hits. In B1 the set of scanned hit is stored after a single leading *SOS word* containing the common *Time Stamp*. In addition 2 bits are added to every hit in order to encode the respective B2 source address.

The B2s have a depth of 8 hit-words, while the B1 can buffer up to 128 hit-words. The asymmetry is not only due to the 4 to 1 correspondence but also for the different emptying methods. B2s are data-through FIFOs, no hold condition on the output is foreseen. B1 outputs instead, for the adopted Round Robin algorithm, are kept in hold for 3*average emptying time, requiring more space for buffering. These depth values have been investigated in several simulations in order to find the optimal parameters.

2.4 The Final Concentrator

Each sub-matrix is provided with an independent and parallel readout instance as it is shown in Fig. 6. The *Final Concentrator* is the element that collects data from the 4 B1 instances in order to drive the output data bus with a proper data protocol.

The output data protocol is realized in order to preserve the time sorting of the hits and it implements a minimal data compression. The B1s are emptied with a Round Robin algorithm and a special *Header Word* is sent before switching to a new B1. In the header word are specified the Time Stamp of the following hits and their B1 source address. Following hits preserve the same B1 data formatting.

We need then a 21-bit word to sparsify a single fired pixel (ref. Fig. 5 + 2 bits of B2 address). Assuming the same header word strategy but a direct x-y pixel addressing, the same amount of information is carried in 16 bits only (1 SOS bit + 7 bit X address + 8 bit Y address).

We opted then for the zone technique mainly for the improvement that brings in case of clustered events. Let's consider the cluster factor of 4 introduced in the target hit-rate: If we suppose

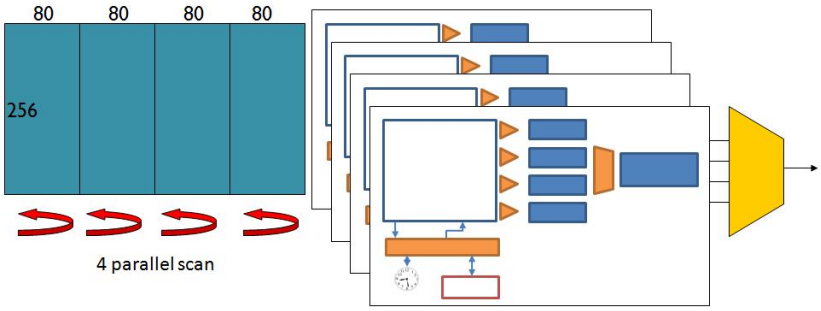


Figure 6: Full readout block scheme. 4 sub-matrices driven by 4 readout instances, and a common output stage for data transmission over a broad-band bus.

that the typical shape of a cluster is 2×2, there is a probability of 87.5% that we can use 2 words only (42 bits) to sparsify the whole event, and in the remaining 12.5% of cases we must use 4 words (84 bits). A weighted average returns a mean value of 46.2 bits per cluster. In a direct x-y sparsification technique instead, we would transmit a cluster at the cost of 64 bits. Thus we foresee that the zone technique will bring, in first place, the benefit of an average bandwidth saving of more than 25%.

3. Simulation Results

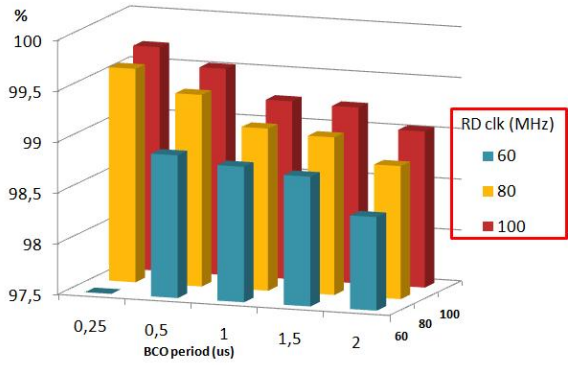
The architecture shown has been implemented with a synthesizable VHDL model. Test bench simulations have been carried out for model verification and fine adjustments of parameters.

An intensive simulation campaign was performed also in order to establish the efficiencies of the readout architecture. The main results of these tests are reported here.

First of all a test bench was set up for the evaluation of efficiencies concerning a single sub-matrix readout. A non-synthesizable Macro Pixel VHDL model was realized with random hit generation capability, adjustable rate and shape, and provided with built-in efficiency trackers. The sub-matrix model is a 2D array of MPs with parameterized dimensions.

A span of typical working conditions has been probed, ranging on realistic clock frequency intervals and hit rates. The results presented in Tab. 7b and plotted in Fig. 7a refer to a set of simulations carried out with a constant hit rate fixed to the target value of 100 MHz/cm^2. The efficiency values reported, refer only to the loss of hits due to MP freezing. The longer is the average freezing time, the lower the efficiency. These values do not take into account the possible inefficiencies of the sensor and it is supposed that each MP is ready to trigger right after the reset. Freezing inefficiency is then a factor of the total inefficiency caused directly, and only, by the readout algorithm and the matrix architecture. It represents then a good benchmark of how well the architecture is behaving regardless of all the other sources of inefficiency.

We varied the main read clock of the digital readout from a minimal value of 60 MHz to a Max value of 100 MHz, with a middle step of 80 MHz. At the same time we varied the BC clock period (time granularity) from 0.25 μs to 2 μs.



(a) Plot

99.7	99.5	99.3	99.2	99.0	**100**
99.6	99.4	99.1	99.1	98.8	**80**
97.5	98.9	98.8	98.8	98.4	**60**
0.25	**0.5**	**1**	**1.5**	**2**	

RDclk (MHz)

BC clock periods (μs)

(b) Table of values

Figure 7: Freezing efficiency plot. The efficiency drop in lower-left corner is due to Scan Buffer overflows. This implies no hit loss but a longer average sweeping time and a reduced time resolution for some events. Freezing efficiency results in %. 1 ms simulated at 100 MHz/cm^2, corresponding to more than 30 khit generated on a 80x256 sub-matrix, 40 μm pitch, no clustering.

A second campaign of simulations was intended to test the behavior of the entire chip, putting together 4 sub-matrices and 4 readout instances plus the data concentrator. The full 82-Kpixel matrix and the 4 independent instances of readout were simulated at a real time rendering factor of about 150 ns per second. For this simulations we imposed the usual hit rate of 100 MHz/cm^2 and we used a 66.6 MHz read clock and a 200 MHz fast clock for the output bus driving. At the same time we wanted to inspect the behavior of the whole infrastructure scaling the BC period down to hundreds of ns. Results are reported in Fig. 8.

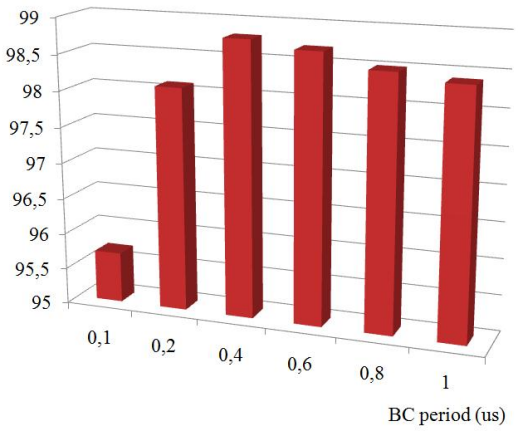


Figure 8: Freezing efficiency plot for the whole matrix. Efficiency drop at 100 ns is caused by Scan Buffer overflows, ref. to Fig. 7.

4. Conclusions

An innovative readout architecture for a wide matrix of pixels has been investigated (>80 Kpixels). Similar structures have already been implemented on silicon thanks to fruitful collaborations like SLIM5 and VIPIX [9] [10]. The chosen target conditions resemble those foreseen at the SuperB SVT layer 0. The project shown can process more that 50 Gpixels per second with a 50 MHz clock exploiting the great parallelization in both matrix dimensions, thus granting high efficiencies even with rates up to 100 MHz/cm^2. The proposed architecture has been modelled in synthesizable and parameterized VHDL, then it has been submitted to a wide-range series of tests exploiting an ad-hoc VHDL matrix model. We run several simulations with a Monte-Carlo hit generation for the evaluation of readout efficiencies. These tests, shown that readout algorithms introduce an inefficiency factor smaller than 2% in the nominal target conditions.

References

[1] R. Klingenberg on behalf of the ATLAS Pixel Collaboration. *The ATLAS pixel detector*. Nuc. Instr. and Meth. in Phys. Res. A - 2007, Volume 579, pp. 664-668.

[2] S. Schnetzer for the CMS Pixel Collaboration. *The CMS pixel detector*. Nuc. Instr. and Meth. in Phys. Res. A - 2003, Volume 501, pp. 100-105.

[3] H. Spieler. *Front-end electronics and trigger systems - Status and challenges*. Nuc. Instr. and Meth. in Phys. Res. A - 2007, Volume 581, pp. 65-79.

[4] SuperB Collaboration *A high luminosity asymmetric e+e- super flavor factory - Conceptual Design Report*. http://arxiv.org/abs/0709.0451v2

[5] A. Gabrielli for the SLIM5 Collaboration. *Proposal of a sparsification circuit for mixed-mode MAPS detectors*. Nuc. Instr. and Meth. in Phys. Res. A - 2008, Volume 596, pp. 93-95.

[6] G. Rizzo. *Recent development on CMOS monolithic active pixel sensors*. Nuc. Instr. and Meth. in Phys. Res. A - 2007, Volume 576, pp. 103-108.

[7] L. Gaioni et Al. *A 3D deep n-well CMOS MAPS for the ILC vertex detector*. Nuc. Instr. and Meth. in Phys. Res. A. doi:10.1016/j.nima.2009.09.041.

[8] R. Lipton. *3D-vertical integration of sensors and electronics*. Nuc. Instr. and Meth. in Phys. Res. A - 2007, Volume 579, pp. 690-694.

[9] A. Gabrielli for the SLIM5 Collaboration. *A 4096 pixel MAPS device with on chip data sparsification*. Nuc. Instr. and Meth. in Phys. Res. A - 2009, Volume 604, Issue 1-2, pp. 408-411.

[10] G. Rizzo for the SLIM5 Collaboration. *Development of deep N-well MAPS in a 130 nm CMOS technology and beam test results on a 4k-pixel matrix with digital sparsified readout*. IEEE NSS conference record 2008. (ISBN-978-1-4244-2714-7).