Macro Pixel ASIC (MPA): The readout ASIC for the pixel-strip (PS) module of the CMS outer tracker at HL-LHC

D. Ceresa [a*]**, A. Marchioro**[a]**, K. Kloukinas**[a]**, J. Kaplon**[a]**, W. Bialas**[a]**, V. Re**[b]**, G. Traversi**[b]**, L. Gaioni**[b]**, and L. Ratti**[b]

[a]*CERN,*
CH-1211, Geneve 23, Switzerland
[b]*University of Bergamo,*
24044 Dalmine (BG), and
INFN Sezione di Pavia,
27100 Pavia, Italy
E-mail: `Davide.Ceresa@cern.ch`

ABSTRACT: The CMS tracker at HL-LHC is required to provide prompt information on particles with high transverse momentum to the central Level 1 trigger. For this purpose, the innermost part of the outer tracker is based on a combination of a pixelated sensor with a short strip sensor, the so-called Pixel-Strip module (PS). The readout of these sensors is carried out by distinct ASICs, the Strip Sensor ASIC (SSA), for the strip layer, and the Macro Pixel ASIC (MPA) for the pixel layer. The processing of the data directly on the front-end module represents a design challenge due to the large data volume (30720 pixels and 1920 strips per module) and the limited power budget. This is the reason why several studies have been carried out to find the best compromise between ASICs performance and power consumption. This paper describes the current status of the MPA ASIC development where the logic for generating prompt information on particles with high transverse momentum is implemented. An overview of the readout method is presented with particular attention on the cluster reduction, position encoding and momentum discrimination logic. Concerning the architectural studies, a software test bench capable of reading physics Monte-Carlo generated events has been developed and used to validate the MPA design and to evaluate the MPA performance. The MPA-Light is scheduled to be submitted for fabrication this year and will include the full analog functions and a part of the digital logic of the final version in order to qualify the chosen VLSI technology for the analog front-end, the module assembly and the low voltage digital supply.

KEYWORDS: Digital electronic circuits; Data reduction methods; Pixelated detectors and associated VLSI electronics.

---

*Corresponding author.

Contents

---

1. The Pixel-Strip Module

The higher luminosity for the Phase-II upgrade of LHC entails new challenges in the design of the CMS silicon outer tracker [1]. The higher granularity needed to keep the occupancy level at a few percent and the requirement of having a good estimation of the z-coordinate of the hit gives rise to the need of pixelated sensors. Furthermore, to keep the Level 1 (L1) trigger rate at an acceptable level (500 kHz to 1 MHz) requires the capability to perform quick recognition of particles with high transverse momentum (p_T) [2]. A particle traversing a set of two sensors spaced by about 2 millimeters at an almost perpendicular direction to the plane of the sensors generates a so called "stub". The stub is the elementary primitive to build a vector of these high momentum particles. The stub finding is based on the concept that a low p_T track bends more in the 3.8 T magnetic field of CMS than a high p_T track, and it uses the distance between hits from the same track in these two sensors to discriminate between them.

The requirements mentioned above together with the limited power and material budget drive the development of the tracker modules: in order to generate stubs each module is composed by two sensor layers, the first of which is a pixelated sensor to ensure the high granularity, while the other is a strip sensor to limit the power consumed by the readout ASICs and to reduce the number of electrical lines on the hybrid. This module is called Pixel-Strip (PS) Module [3].

1.1 Module Architecture

This new stub finding module accommodates a strip sensor and a pixelated sensor, covering an area of approximately 5 cm x 10 cm, and is mounted on a mechanical assembly providing support and cooling, as shown in figure 1. This module can be placed with different orientations in the outer CMS tracker: in the barrel layers, the beam is parallel to the z-axis of the module, while in the end cap layers, it is parallel to the y-axis of the module and in both configurations the x-axis stays on the r-phi plane. Along the x-axis, the dimensions of the strips and pixels length is $100\,\mu$m, while along the z-axis it is 2.5 cm for the strips and 1.5 mm for the pixels. Consequently, the strip sensor is segmented into 2 x 960 strips while the pixel sensor is segmented into 32 x 960 pixels.

Readout ASICs   The strip sensor is read out from 16 Short Strip ASICs (SSA). Wirebonds provide the connectivity to a high-density substrate carrying the ASICs that are bump-bonded onto it. The pixelated sensor instead is read out by 16 Macro Pixel ASICs (MPA) distributed in two rows bump-bonded on it. Wirebonds connect the MPA periphery to the same substrate carrying the SSA, hence realizing the top-to-bottom connectivity [4].

From a functional point of view, the SSA processes the sensor strip signals and sends the hit information to the MPA at each bunch crossing. The latter stores the full event (pixel and strip hits) and correlates the hits from the two sensors to generate the stubs. Upon reception of a Level 1
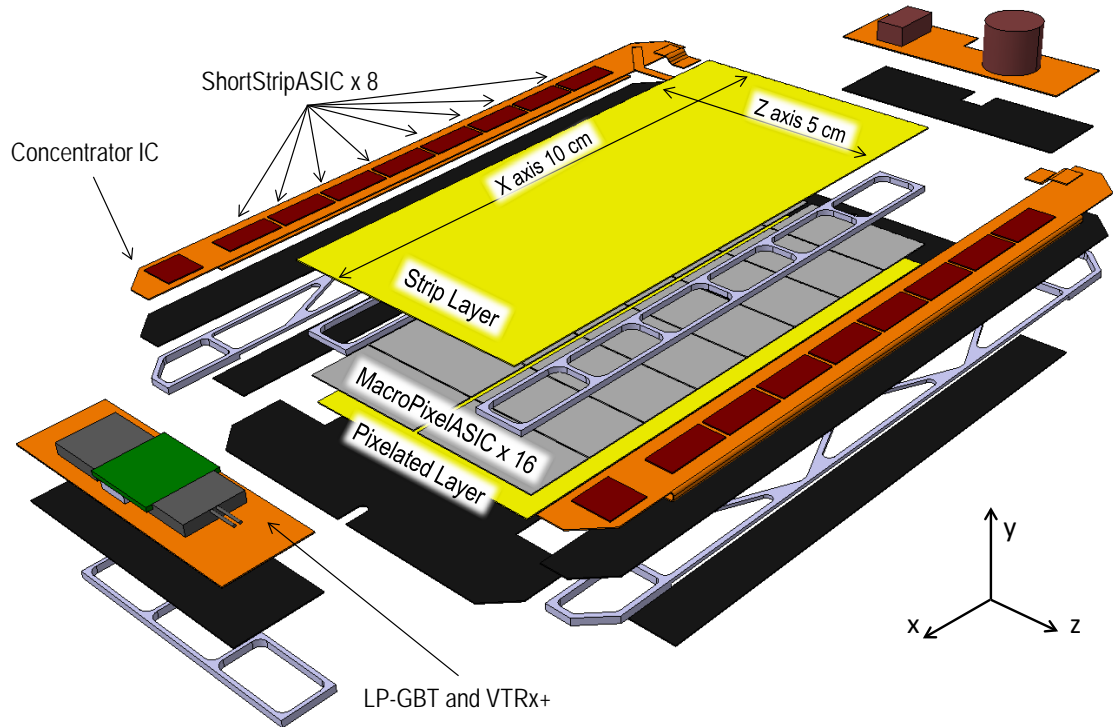

Figure 1. Pixel-Strip (PS) module exploded view. The stack consists of (bottom to top) a cooling plate (black), a pixel sensor (yellow), a layer of 16 MPAs (grey), two Al-CF sensor spacers (ligth blue), 2 Front-End hybrids (orange) housing the SSA (red) and the Concentrator IC (red), 2 service hybrids (orange) housing the optical link (green) and the DCDC converters (brown) and a short-strip detector (yellow).

trigger, i.e. after the Level 1 trigger latency, the MPA sends the stored complete event information to the readout back-end electronics. The generated stubs are instead sent out at each bunch crossing to the trigger back-end electronics with a latency, due to the stub finding process, which should not exceed 250 ns (10 bunch crossings). The MPA output data does not reach directly the CMS back-end: another ASIC, the Concentrator IC (CIC), aggregates the data from the 16 MPAs on each module and sends them to the Low Power GigaBit Transceiver (LP-GBT) which transmits a serial stream to the CMS back-end through an optical link transceiver (VTRx+).

This paper illustrates the design and the optimization concerning the Macro Pixel ASIC with a particular attention to the logic responsible for the discrimination of the high momentum particles, i.e. the so called stub finding.

1.2 Macro Pixel ASIC

The large size of the pixelated sensor requires the use of 16 MPA ASICs for reading out a single sensor. Every chip connects to 120 x 16 pixels and it is composed by a pixel matrix region of 12 mm x 24 mm and a periphery region of about 2 mm that resides on one edge of the chip as shown in figure 2 (top). The MPA logic processes at each bunch crossing the data from the pixel front-end and from the SSA through three functional blocks as illustrated in figure 2 (bottom):
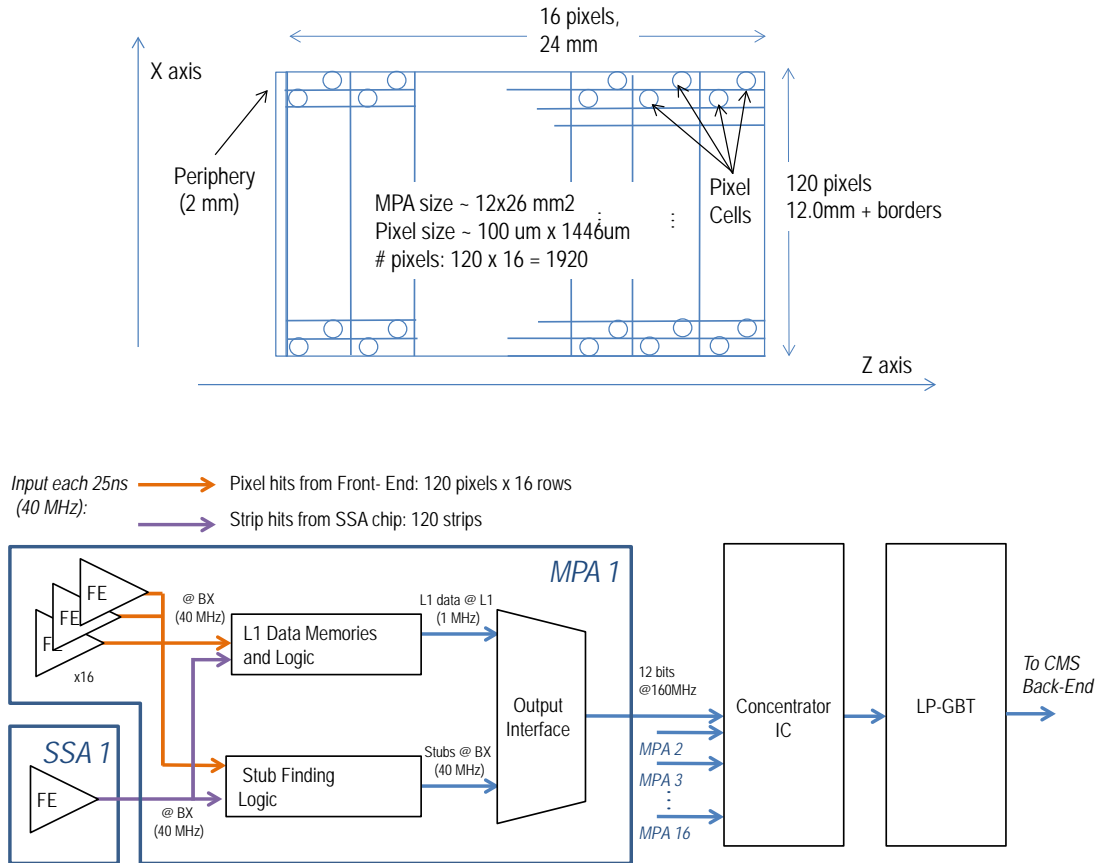

Figure 2. Top: Macro Pixel ASIC scheme with dimensions. Bottom: Pixel-Strip module block diagram with Macro Pixel ASIC data paths.

- The L1 data block stores the full event information and sends them out if receives a L1 trigger. It stores the hit information from the two front-end (Pixel and Strip) without any data reduction in the L1 Memories for the duration of the L1 latency. Upon arrival of a L1 trigger, the event is processed by the L1 Data Logic which encodes the position of each cluster and its width.

- The stub finding logic receives the same input of the L1 data block synchronously with the 40 MHz bunch crossing frequency and looks for coincidences within a narrow geometrical angle between pixel and strip clusters in order to find and encode the stubs.

- The Output Interface organizes data from the two previous blocks and transmits them to the Concentrator IC at a frequency of at least 160 Mbit/s. This paper does not detail this module since the communication protocol is still under development.

MPA Power Budget The most challenging constraint of the MPA development is the limited power budget available to carry out the complex functions described above. Considering an allowed power consumption of 200-220 mW per MPA, a rough power allocation has been done as follow:

- The analog front-end should consume $\frac{30\mu A}{channel}$ which, including also some bias structures, corresponds approximately to 1/3 of the total available power.

- Another 1/3 of the power has been allocated to the L1 memory, which will support the L1 latency longer than $12.5\,\mu$s. It is expected that this consumes about 60-80 mW.

- The remaining power is allocated for the operation of the three logic blocks described above as well as the clock distribution and the I/O.

Due to the large area of the chip, another non-negligible contribution to the power consumption is the data transport on-chip, i.e. the power needed to move the data from the pixel array to the periphery and from the pixel cells to the memories. This is the reason why the floorplan of the ASIC becomes of fundamental importance and in the following section not only the stub finding algorithms will be described but also the spatial placement of the different blocks will be covered. Further, also the power consumed by the clock distribution is affected by the large size of the pixel array and requires additional studies.

2. Design Development

To minimize the power, the most challenging block is the stub finding logic. This is due to the relatively high operational frequency (40 MHz) and to the numerous functionalities included. Hence, the next paragraph reports a detailed description of the architecture developed for the stub finding logic which is summarized in figure 3.

Afterwards, two other power consuming components of the MPA design are introduced: the L1 Data Path and the clock distribution.
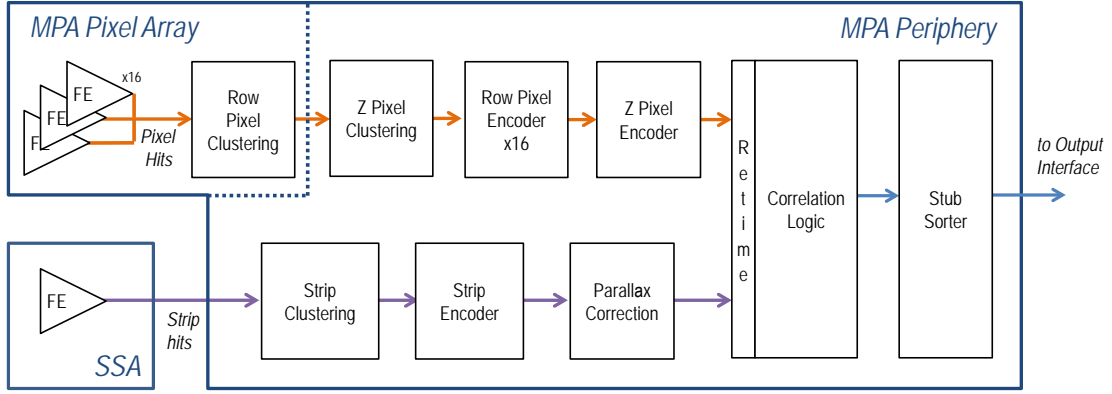
Figure 3. Stub finding logic block diagram.

2.1 Stub Finding Logic

The stub finding logic provides the high p_T information to the Level 1 Trigger by sending out coordinates of the stubs which have been found and some information related to an estimation of the transverse momentum. A stub is summarized with the coordinates of the point of incidence and with the bending angle in the r-phi plane (x-coordinate) of the particle. The stub finding algorithm discriminates the particles based on the bending value: if it is lower than a given threshold the particle is accepted and the information about the corresponding stub is sent to the L1 trigger. Pixel and strip ASICs implement binary readout, so the input of the stub finding logic is a binary matrix of 120 x 16 pixels and a binary vector of 120 x 1 strips, while the output are the encoded position and bending of the stubs found. In order to generate this information, the input data is first processed to extract interesting clusters: on the r-phi plane (pitch = 100 μm), clusters wider than a programmable threshold are rejected, while the centroids of accepted clusters are calculated. On the z-axis (pitch ∼ 1.5 mm), clusters wider than 2 pixels are rejected, while for 2-pixels wide clusters the coordinate of the pixel closer to the periphery is chosen as centroid. In a second processing stage, the centroid coordinates are encoded and a programmable offset is applied to the strip centroids depending on their module coordinates. This offset corrects the parallax error generated by approximating a cylindrical geometry with sensors that are actually planar strips. By using such planar sensor, different positions on the sensor in the r-phi plane correspond to different distances from the vertex. The core of the stub finding logic is the correlation between the two sensor layers: the correlation logic compares every pair of pixel and strip centroid and accepts the centroid pair if the difference between their x-coordinates is within a given programmable range. Before transferring to the output interface, the found stubs undergo a sorting process step. The next paragraphs detail the logic functions just described, comparing them with the architecture described in [5] which is the starting point of this development.

Pixel Clustering   For data reduction, the first operation on the Pixel matrix presented in [5] is the Column OR-ing, i.e the logic OR of the hits of each pixel column. By using this technique, large clusters from low p_T-particles, secondaries or combinatorials can hide good clusters from high p_T-particles decreasing the efficiency of the whole architecture. Several solutions have been compared

– 5 –

and the row pixel clustering technique has been chosen: it consists in doing the cluster elimination and the centroid extraction at pixel level without OR-ing the pixel columns. This solution avoids also the transmission of large clusters along the 24 mm long pixel matrix decreasing the amount of data moved to the periphery and the power consumption. On the other hand, the cost of this local clustering is an increased logic circuitry per pixel and additional interconnects along the row of pixels. This structure has been adopted for the x-axis clustering, while the clustering along the z-axis is carried out in the periphery.

Pixel Centroid Encoding Excluding the Column OR-ing and doing the Pixel Clustering as described in the previous paragraph, the periphery receives a large amount of lines: 180 x 16 instead of 120 x 1. Consequently, the next operation is the centroid position encoding which requires one MEPHISTO encoder [6] per row for the x-coordinates and one Priority Encoder for the whole matrix for the z-coordinates. The MEPHISTO structure ensures very low power consumption but it requires two bunch crossings to encode 4 coordinates because it is limited in processing to 2 coordinates per clock cycle. The Priority encoder instead encodes up to a maximum of 8 coordinates in one bunch crossing. Therefore the Pixel Centroid encoding requires a total of 3 latency cycles and it encodes up to 8 centroid coordinates.

Strip data processing The pixel data processing requires more design effort to minimize the power consumption, while the strip data processing requires a design trade-off between latency and power optimizations since the strip data acquisition and transmission from the SSA to the MPA requires at least 3 bunch crossings. Therefore, the strip clustering module finds in one clock cycle the centroids which are then encoded from a priority encoder with the same architecture of the z pixel encoder. Even if this encoder consumes more power, it encodes 8 strip centroids in one cycle, without increasing the latency. Having the strip centroid position already encoded, the parallax correction offsets the coordinate of the centroid by a programmable shift which is in the range of $+/-400\,\mu$m and has a precision of $+/-50\,\mu$m. Two offsets can be set, one for the first half of the strips, the other for the second half.

Correlation Logic A high p_T-particle in two closely spaced sensors generates centroids within a known range, called window. Exploiting this property, the correlation logic computes the x-position difference between pixel and strip centroid and if it is within the defined window it generates a stub. The stub position is defined as the pixel centroid xz-position while the stub bending is defined as the difference between the centroid x-coordinates. Since the encoders accept up to 8 pixel and strip centroids, 48 cells process each possible combination of pixel and strip centroids but the correlation logic limits the output to 2 stubs per pixel centroid. The last step of the stub finding logic orders the stubs giving priority to the lower row and transfers them to the output interface.

2.2 L1 Data path

The L1 Data path, summarized in figure 4, is divided in two functions: event storing and event processing. The L1 memory stores one event per bunch crossing and after the L1 latency, if it receives a L1 trigger signal, it sends the event to the processing, if not, it discards the event. The processing extracts the cluster width of the strip and pixel clusters and encodes the position of the first pixel or strip of the clusters. From the power consumption point of view, the most critical part is
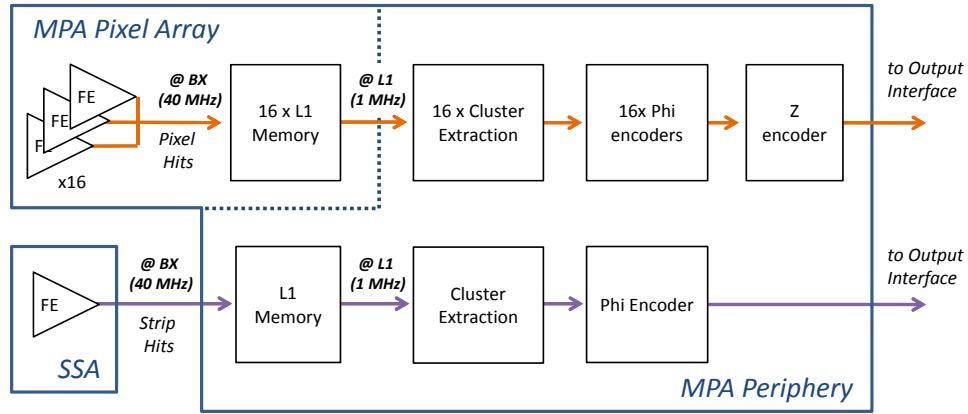
Figure 4. L1 Data Path block diagram.

the event storing because of the continues memory write cycles at 40 MHz, while the probabilistic memory read operations and the consecutive event processing works at an average frequency of 500 kHz or 1 MHZ (L1 average frequency). L1 memories are circular memories which store every event for the L1 latency; in particular one memory per pixel matrix row is foreseen plus one for the strip data, resulting in 17 L1 memories in total. The word size is 128 bits and the depth is defined from the L1 latency: up to a L1 latency of 12.5 μs, a 512 words memory is sufficient, while for longer latency (20/25 μs) a 1024 words memory is necessary. In the latter case, memory power consumption is clearly critical and increases to 1/3 of the power available. A possible solution to decrease this value, would be to move the event processing before the event storing but the cost in terms of power of running event processing at 40 MHz instead of 1 MHz must be carefully evaluated and it is currently under study.

2.3 Clock distribution

The large dimensions of the ASIC require a careful study for the clock distribution tree. A simple architecture based on clock tree implementation per column would consume more than 40 mW due to the 120 columns. The proposed solution, shown in figure 5, is a row distribution architecture where a global column buffer distributes the clock along the central column of the pixel matrix (24 mm) and one clock line per row distributes the clock to the 120 pixel cells in the row (12 mm). This solution decreases strongly the number of clock buffers: from 1 row buffer and 120 column buffers to just one column buffer and 16 row buffers. Furthermore, every row buffer in the row distribution scheme drives a load capacitance of around 2.7 pF, while every column buffer in the column distribution scheme drives a load capacitance of around 4.8 pF. Consequently, the power consumption decreases by about 85% (considering CMOS driver) and the most critical point is the large load capacitance ($\sim$17 pF) of the column buffer which can be managed introducing repeaters along the column (the same problem would be faced with the row buffer in the column distribution scheme).

Another important specification for the clock distribution is the maximum clock skew which must be shorter than 1 ns. The proposed scheme with standard CMOS driver and with 3 repeaters
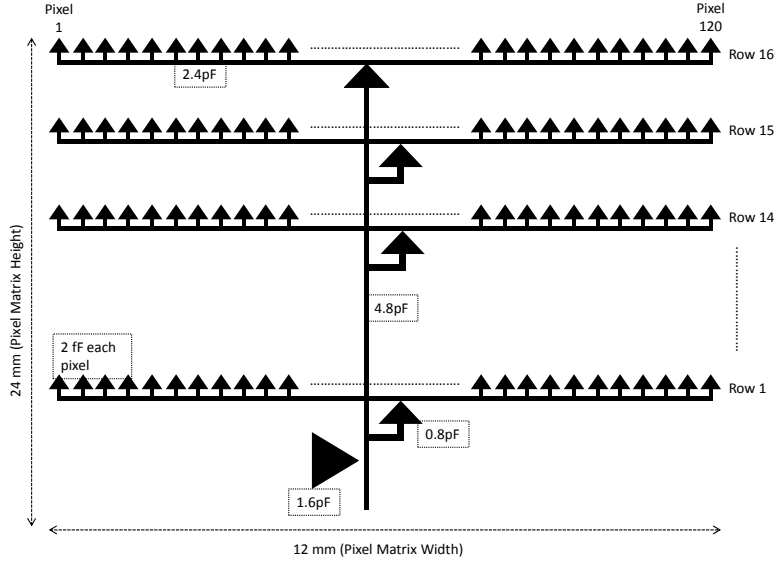
Figure 5. Row Distribution clock scheme with load capacitance for each stage

along the clock column fulfills this requirement with a power consumption lower than 5 mW.

In addition to the clock tree architecture, different buffer structures can further decrease the power consumption. In particular low-swing driver, low-voltage driver and charge redistribution buffers are currently evaluated.

3. MPA Monte Carlo Simulation

The proposed architecture is being validated by injecting randomly generated events in the MPA ASIC functional model at the Register Transfer Level (RTL) and at the synthetised gate level. The results of the verification with random input helps designers to improve the model and to find bugs, but does not provide any information about the performance or about the switching activity (directly related to the power consumption) of the chosen architecture. Consequently, also Monte Carlo generated events [7] have been used to evaluate the performance of the Macro Pixel ASIC architecture described in the previous section.

3.1 Simulation method

Monte-Carlo (MC) programs for computer simulation of complex interactions in high-energy particle collisions provide event samples for the entire CMS Tracker. These MC events contain information of all the particles with a p_T larger than 0.1GeV/c like p_T, impact parameter and hits positions. A script translates the format of the MC generated events into input files adapted for the MPA Verilog model and into files which contain the expected output stubs. Running simulations with the MC input files and comparing the obtained output with the expected output, the designer can evaluate all aspects of the performances of the MPA architecture and also the limitations introduced from the limited bandwidth in the module or from the module to the CMS back-end.
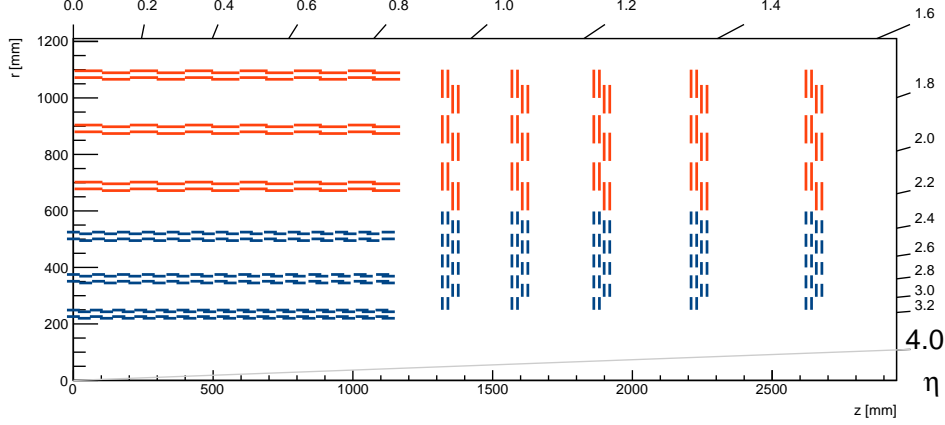
Figure 6. CMS outer tracker geometry used for simulation. Blue lines are the PS modules, while red lines are 2S modules [1].

3.2 Efficiency analysis

By using the proposed new tracker geometry [8] shown in figure 6, the MPA simulation with MC generated events provides an estimation of the MPA performance in terms of stub finding efficiency in the HL-LHC environment. This efficiency is defined as the percentage of particles with p_T larger than 2 GeV/c which the model detects. The results for the first barrel layer are shown in figure 7: the worst case performance corresponds to the densest environment, which is found in the Layer 1 located at approximately 23 cm from the beam line, where the total stub finding efficiency is around 88.5% (red points in figure 7). The stub finding efficiency increases to around 95% in the center of the tracker while it decreases at large z values due to geometrical inefficiencies stemming from the
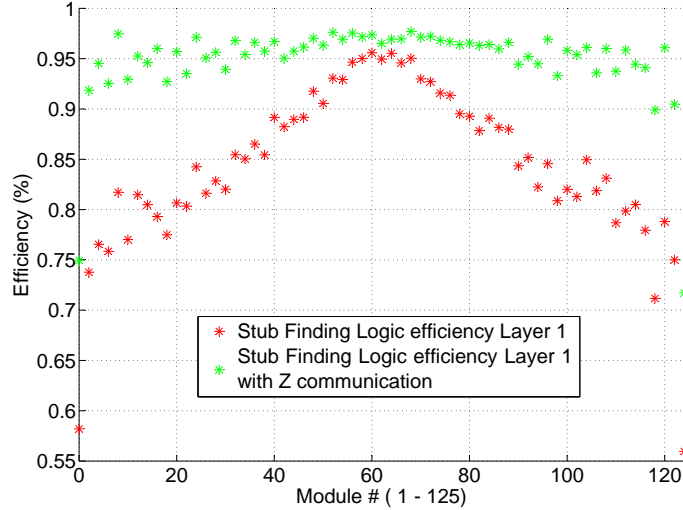

Figure 7. Stub finding logic efficiency respect to the module number in Layer 1. Module 63 is located at z = 0. Higher and lower module numbers correspond to positions with larger absolute z values, with a maximum z of +/- 1100 mm. The lower efficiencies of module 1 and 125 are artifacts due to the absence of the end-caps in the simulation.

absence of z-communication between the two chip rows in the PS module. This limitation makes
impossible to detect the particles crossing two different chip rows in the bottom and top sensors
and this phenomenon causes the largest inefficiency. Several solutions are being evaluated to solve
this problem, because by solving it the efficiency distribution in z becomes almost flat, providing
an overall efficiency of around 96% for the first barrel layer (green points in figure 7).

A promising alternative to the introduction of the z-communication is the tilted tracker lay-
out [9] which allows the PS module in the barrel layer to be always almost perpendicular to the
particles. Consequently, the number of particles crossing two different chip rows in the bottom and
top sensors will be minimized and the expected efficiency will be the same as the module located
at Z=0.

This analysis allows also the study of the parameters space of the MPA model. Several simula-
tions with different parameters as correlation window and cluster width provides the comparisons
between the different configurations. A simple example is reported in figure 8, where the simula-
tion is repeated with different correlation window dimensions and, decreasing the window size, the
total layer efficiency decreases by about 2%.

Another interesting result verified in the simulation is the amount of fake stubs which are
generated by non-interesting particles such as low p_T particles, secondaries or combinatorials:
only between 5-10% of the stubs generated by the MPA model corresponds actually to a high-p_T
particle. This large amount of "fake" stubs is filtered out by L1 central trigger in the CMS Back-End
which excludes them correlating the stubs from different tracker layers. However, all the generated
stubs need to be transmitted from the MPA to the CMS Back-End before the filtering. Using
as reference the diagram in figure 3, the bandwidth bottlenecks are the communication between
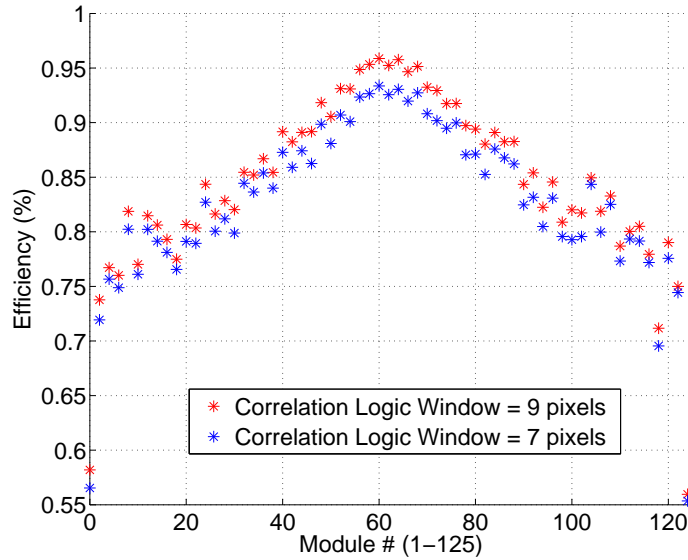

Figure 8. Stub finding logic efficiency respect to the module number in Layer 1. Red points represents the stub finding Logic efficiency with a correlation logic window of 9 pixels, while blue points represents the same efficiency with a correlation logic window of 7 pixels.

256 MPA and Concentrator IC and the Optical Link between the LP-GBT and the CMS Back-End.
257 Consequently, in addition to the MPA inefficiencies shown before, also the limited bandwidth for
258 stub transmission can introduce inefficiencies in the stub finding process which will be evaluated
259 and minimized in the next steps of the PS module development.

4. MPA-Light Demonstrator

261 A first prototype of the Macro Pixel ASIC is under development in a 65 nm Low Power CMOS
262 technology and it is called MPA-Light. It consists in a reduced size MPA with a pixel array of
263 16 x 3 pixels, instead of 120 x 16 pixels. It integrates bump-bond pads for sensor connections and
264 wirebond pads for hybrid connections. The size of the single pixel will be 100 x 1446 μm like
265 the final MPA. The principle purposes of this ASIC are to prototype and qualify the analog Front-
266 End circuitry described in the next paragraph, to facilitate the development of the sensor and to
267 understand and solve the technical aspects of the module assembly.

5. Analog Front-End

269 In both MPA and MPA-Light, the full analog chain, shown in figure 9, is composed by a pream-
270 plifier, a shaper and a discriminator. The preamplifier is built with a buffered cascode loaded with
271 a degenerated PMOS cascode current source and enclosed with a Krummenacher feedback [10]
272 providing leakage compensation for the n–on-p+ silicon sensors up to 200 nA. An extra current
273 source directly supplying the input transistor provides the extra boosting of the bandwidth and the
274 minimization of the noise contribution from the active loads.
275 The second stage working as an amplifier/integrator and the threshold interface are built with
276 a differential folded cascode loaded with resistors. The common threshold for the discriminator
277 is provided by high impedance current source mirroring the output current from an 8-bit mutual
278 DAC and sourcing it to one of the load resistors which produces a DC voltage imbalance. The local
279 per-pixel 5-bit DAC is connected to the second load resistor which provides the equalization of the
280 discriminators offset spread.
281 The two stage comparator consists of a folded cascode differential amplifier with swing limiter
282 preventing saturation of this stage due to DC threshold voltage at the input, followed by a differen-
283 tial to single ended stage which also provides hysteresis. The overall current consumed by the front
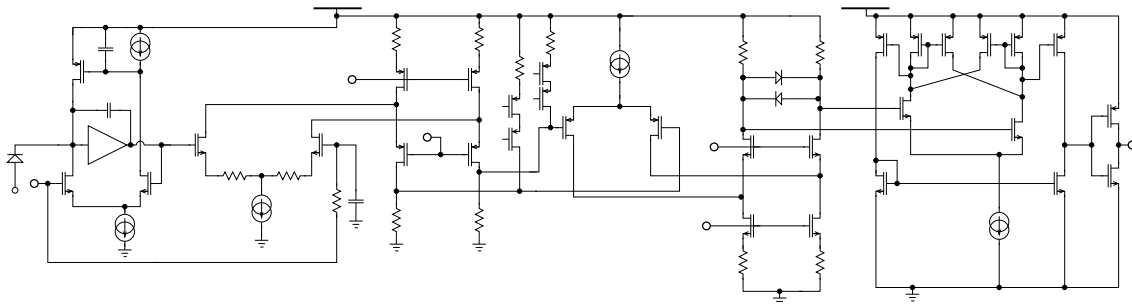

Figure 9. MPA Full Analog chain schematic with preamplifier, shaper and discriminator.
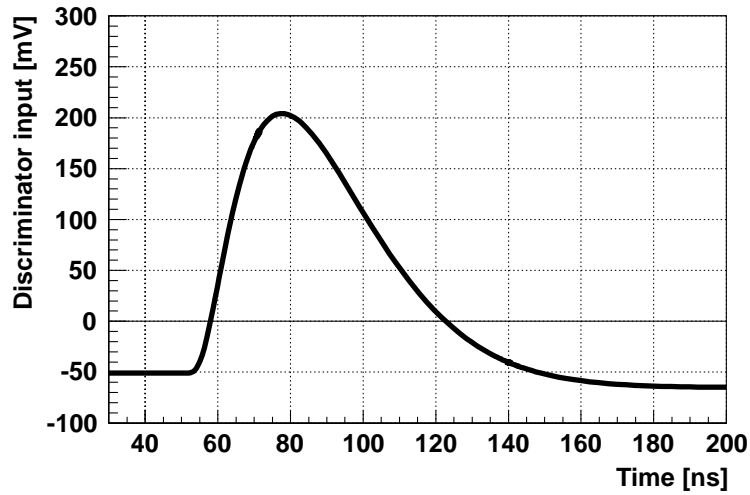
Figure 10. Discriminator input for a 2.5 fC signal with the threshold set at 0.5 fC

end is below 30 μA for nominal bias condition (input transistor biased with 16 μA). The pulse gain, as shown in figure 10, of the front end amplifier seen at the discriminator input is about 105 mV/fC (post-extraction simulation) and the peaking time of the amplified pulse from the detector is around 24 ns, which limits the time walk of the front end channel (preamplifier/shaper/discriminator) below 14 ns for signals ranging between 0.6-12 fC with the discriminator threshold set to 0.5 fC. The simulated noise for an expected input capacitance of around 500 fF (280 fF detector capacitance + 55 fF bonding pad + 160 fF ESD) and the worst case detector leakage current (50 nA) is around 200 e^- ENC.

6. Conclusion

Several solutions for the implementation of the stub finding in the Macro Pixel ASIC have been studied and the most promising in terms of efficiency and power dissipation has been developed in detail: the pixel clustering carried out at pixel level and the encoding of the centroid avoid hit masking problems and allow data transfer power optimization.

The described model has been verified and simulated in the high-luminosity LHC environment, allowing an estimation of the performance and an understanding on the causes of inefficiency. The largest one is caused by the impossibility to detect the particles crossing two different chip rows in the bottom and top sensors. Z-communication and tilted tracker layout are the possible solutions which would raise the efficiency to 96% in the first barrel layer (worst case).

A first reduced size prototype of the MPA is being designed to help the development of the final ASIC, with particular attention to the analog Front-End, as well as the development of the sensor and of the module assembly.

References

[1] D. Abbaneo, *Upgrade of the CMS Tracker with tracking trigger*, 2011 *JINST* 6 C12065.

[2] N. Pozzobon, *Development of a Level 1 Track Trigger for the CMS Experiment at the High-Luminosity LHC*, *Nucl. Instrum. Meth.* A732 (2013) 151.

[3] D. Abbaneo et al., *A hybrid module architecture for a prompt momentum discriminating tracker at HL-LHC*, 2012 *JINST* 7 C09001.

[4] G. Blanchot et al., *Hybrid circuit prototypes for the CMS Tracker upgrade front-end electronics*, 2013 *JINST* 8 C12033.

[5] A. Marchioro, *A hybrid module architecture for a prompt momentum discriminating tracker at SLHC*, PoS *(Vertex 2011)* 037.

[6] P. Fischer et al., *MEPHISTO – a 128-channel front end chip with real time data sparsification and multi-hit capability*, *Nucl. Instrum. Meth.*, A431 (1999) 134.

[7] S.Viret, *https://sviret.web.cern.ch/*

[8] S. Mersi et al., *CMS Tracker Layout Studies for HL-LHC*, *TIPP 2011 - Technology and Instrumentation in Particle Physics*, Physics Procedia, 37 (2012) 1070-1078.

[9] G.Bianchi, *tkLayout: a design tool for innovative silicon tracking detectors* , 2014 *JINST* 9 C093054.

[10] F. Krummenacher, *Pixel detectors with local intelligence: an IC designer point of view*, *Nucl. Instrum. Meth.*, A305 (1991) 527.