

Doctoral Dissertation
Doctoral Program in Electronic Enginering (31.th cycle)

Ultra high-density Hybrid Pixel Sensors for the detection of charged particles

Designing a Hybrid Pixel Detector for High Energy Physics experiments

Andrea Paternò
* * * * * *

Supervisor
Prof. Rivetti Angelo, Supervisor

Politecnico di Torino
March 22, 2019

I hereby declare that, the contents and organization of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

. . . . . .

Andrea Paternò
Turin, March 22, 2019

Summary

The LHC complex at CERN is the world's high energy particle accelerator and collider. By accelerating two beams of protons at near-light speed, it allows the experiments placed along its circular path to study the results of proton-proton collisions. The standard model of physics provides a detailed account of the collisions products and their production rates, and by studying the results of the experiments, it is possible to assess whether the model is correct, or if new theories must be put forward in order to explain the observations.

At the core of the experiments at CERN lies a tracker: a silicon detector capable, thanks to its layered structure, to reconstruct the path of the particles generated by the collisions as they escape the interaction point.

The next upgrade of the LHC complex will increase its luminosity, and, with it, the amount of radiation produced and the number of proton-proton collisions per beam crossing. The innermost layers of the trackers, composed of hybrid pixel chips, must therefore be replaced with new electronics capable of sustaining unprecedented radiation doses, and keep track of a much higher number of particles.

The original contribution of this thesis is the design of new digital architectures for hybrid pixel detectors. The new readout chips must be capable of recording particles at rates of 750 MHz for trigger latencies above 12.5 μs. The expected radiation doses that the chips have to withstand before replacement are in the order of 500 Mrad. The pixels' area has been scaled to 50×50 um^2, 8 times smaller than that of the current chips, in order to enhance the spatial granularity of the detector.

A variety of buffering and readout schemes have been proposed, implemented, and analyzed with respect to their performances. The efficiency has been assessed with realistic inputs by a verification environment. The proposed solutions rely on the advantages brought about by hierarchically grouping pixels together in the matrix, and share the buffering and control logic in order to achieve higher efficiencies while reducing the power consumption and area occupancy.

The architectures designed have been implemented in 3 main projects which share the common goal of exploiting the CMOS 65nm technology for the creation of highly integrated radiation-hard electronics capable of satisfying the requirements for the HL-LHC inner tracker.

The CHIPIX65 collaboration by INFN developed a 64×64 pixel chip demonstrator,

including many IP blocks and 2 Analog Front-Ends. The author designed the chip's readout scheme and buffering architecture, using 4×4 Pixel Regions, whose shared logic achieved > 99 % efficiency with a power consumption of 7.5 μW/pixel. The chip has been verified to be working after 600 Mrad of Total Ionizing Dose, and its performances characterized with a variety of sensors.

The second pixel chip has been developed in the context of the RD53 collaboration. The RD53A chip is a half-scale prototype with a 400×192 pixel matrix, integrating 3 different Analog Front-Ends, and 2 digital architecture, including a re-engineered version of the one proposed in CHIPIX65. The much bigger pixel matrix posed several integration issues, which had to be solved to ensure timing consistency across the matrix, and to allow for fast event readout. The chips have been tested with 2 DAQ setups, a variety of sensors and test beams, all showing the chip's functionalities to 500 Mrad irradiation.

The RD53 collaboration is currently working for the final full-chip prototypes for the CMS and ATLAS experiments. A thorough architectural performance assessment has been done with the new physics simulations available, in order to converge into a common digital architecture for the final chip.

Acknowledgements

Sono passati più di 3 anni dall'inizio di questo percorso pieno di sfide, scoperte, notti insonni, ed entusiasmo. E' stato un percorso arricchente, alla fine del quale posso affermare con sicurezza di avere molti strumenti in più sia sotto il profilo professionale, che quello personale.

La persona che ha reso tutto questo possibile, che mi ha seguito sin dal primo nostro scambio di email ormai 5 anni fa, e alla quale va tutta la mia gratitudine, è il mio tutor, Angelo.

Un ringraziamento speciale va a Lino, che è stato un punto di riferimento fondamentale per la mia attività di ricerca, supportandomi con costanza ed attenzione, ed ha rappresentato per me un esempio impeccabile di laboriosità, nella vita professionale e non.

Ai miei compagni di viaggio, Ennio e Luca, devo non solo confronti costruttivi e appassionati, ma anche tanta pazienza per i miei pranzi vegetariani e le soddisfazioni di un buon calice alla fine di produttive giornate di lavoro.

Tra i moltissimi colleghi coi quali ho avuto discussioni tecniche molto costruttive, non posso non citare Flavio, Roberto e Tomasz. Sono molto onorato di aver lavorato con voi, che avete reso la mia esperienza in RD53 emozionante, accogliente, e profondamente significativa.

La mia famiglia è sempre stata per me un sostegno imprescindibile, nonostante stia cominciando a perdere il conto dei passi percorsi lontano. La porto dentro con me in ogni cosa che faccio, e la sola sicurezza di poter contare su di loro, mi da forza in tutte le scelte importanti.

Ai miei amici, che hanno la pazienza di sopportarmi; a Casa Cappiello, il mio porto sicuro; ed alla mia sezione scout, che mi permette di crescere anche da grande.

A Cecilia, che ho incontrato lungo il cammino, ed ai nuovi tracciati che solcheremo insieme.

A Grazia, Massimo,
Greta.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Pixel Detectors are widely used in several applications, ranging from X-ray detectors to digital cameras. They are also used in research, and undertake a key role in the trackers for High Energy Physics experiments: pixel sensors can, in fact, be used to detect incident radiation.

In Pixel Detectors, the surface is regularly divided in fixed-size blocks (the pixels), which sense the energy deposited there by the photon or charged particle. The positional information of the recorded event can be derived from the address of the pixel in the matrix. The energy measurement, can, if necessary, be implemented with several methodologies.

For imaging applications, the intensity is usually calculated by counting the number of times the pixel has been hit by a photon during an exposure period. The pixels must only keep track of a *hit* signal, and count how many times this signal rises in the chosen period. In other cases, such as in Particle Physics Experiments, individual particles must be detected and their tracks be reconstructed.

1.1 High Energy Physics Experiments

High Energy Physics is the branch of physics that studies the nature of the particles that constitute matter and radiation. HEP experiments mainly deal with breaking up these matter constituents, by accelerating particles (in *Particle Accelerators*) and colliding them with another beam of particles, or a fixed target.

In the Large Hadron Collider (LHC) at CERN, two counter-rotating beams of protons are employed. Protons are subatomic particles which, together with the neutrons, constitute the atoms' nuclei. Protons are charged particles, and, being so, they're subjected to the electromagnetic force. Their mass is $0.938\,\mathrm{GeV\,c^{-2}}$ at rest, but can considerably increase if the particle is accelerated to near-light speed.

Deep in the 27-km-long underground accelerator, two beams of protons are accelerated up to 99.9999991% of the speed of light, and are then maintained at this energy

while rotating, a beam clockwise and the other counterclockwise, in 2 adjacent vacuum pipes. At such speed, protons have an energy of 7 TeV, which correspond to a Lorentz factor of about 7460. This has several implications, such as that the particles experience time in a very different way, passing 7460 times more slowly for the particles than it does for us observers.

The proton beams The orbits the beams rotate in are very close, although separate. In special points inside the detector, the proton beams are forced to collide almost "head-on", in order to study their interactions. The average size of the proton beams as they approach the interaction point in the ATLAS Experiment is shown in Fig. 1.1.

The beams are not made of a continuous stream of particles, but are instead segmented into *bunches* containing about a thousand of particles, confined in a cylindrical volume of a few centimeters lengths, spaced 7.5 m from the next bunch. Since their speed is very close to the speed of light, the protons cover that length in just 25 ns, which is, it follows, the time between every bunch crossing.

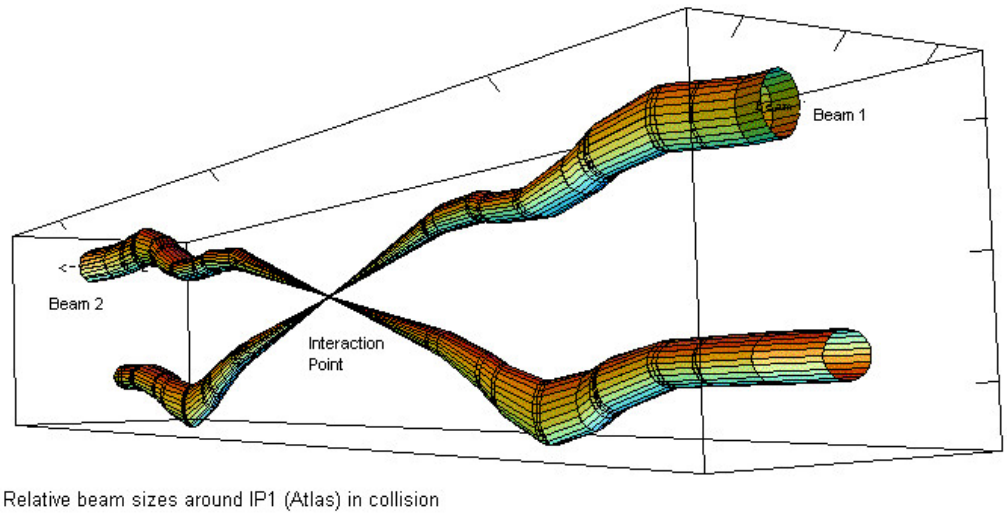

Figure 1.1: The 2 proton beams of the LHC accelerator overlap in the 2 general purpose experiments placed along its path.

Most of the protons pass unaffected at each crossing: only a fraction of the total number of protons in the bunches collide. The number of collisions per bunch crossing (event pile-up) depend on the density of the bunches, the amount of bunch overlap in the collision site, and other related parameters. Such quantities are usually condensed in a general parameter called *luminosity*, which is a measurement of the experiment's ability to constraint the protons through a given space in a given time, and, thus, is measured in $\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$.

The luminosity can also be interpreted as the proportionality factor between the number of events per second in the detector, and the related process cross section ω_p [46]. In other words, the luminosity of the experiment relates the probability that an event occurs (its cross section) with the number of events actually occurring in the experiment, considering the experiment's parameters.

$$\frac{dR}{dt} = \mathscr{L} \cdot \omega_p \tag{1.1}$$

Cross sections are measured in barns (1 barn = 10^{-24} cm^2), and, for proton-proton collisions at 7 TeV is approximately 110 mb. Cross sections for rarer events, such as the Higgs boson production in the experiment, can be as low as 50 fb (femto barns). With such low cross sections, it is important to have high luminosities in order to make these rare events observable.

These observations highlight how the luminosity of a collider is a critical parameter. In the LHC, the nominal luminosity is 1×10^{34} cm^{-2} s^{-1}, but work is constantly underway in order to increase it.

The collisions and their products When protons collide, the total energy at the collision point will be equal to the sum of the individual beam energies. This is called "center-of-mass energy", in the order of the tens of TeV for the LHC proton-proton collisions. This energy is thereafter converted back into matter, according to Einstein's famous $E = mc^2$, and kinetic energy.

Although unstable particles generated by the collisions are usually short-lived, the high speed they move at effectively increases their lifetime. This, however, is not sufficient to make most of the decay products reach the detector: they can decay again in other subproducts. We therefore speak of a *primary* vertex, where the original proton-proton interaction has happened, and a *secondary* vertex, where a particle generated by this first interaction decays. In Fig. 1.2, an event in the ALICE detector can be seen to produce a secondary and tertiary vertex in a particle track.

In order to study the trajectory of the particles produced in the primary or secondary vertex, the detectors are wrapped cylindrically around these collision sites, in a layered structure which covers about ∠178/∠180. The remaining degrees are covered by endcap structures places at the extremes of the cylindrical structure. This means that very few particles generated in the collisions manage to escape undetected.

Quantities under measurement The focus of the experiments is to characterize the particles which traverse the detectors, in order to infer the point they originated from, and, in case the point corresponds to a secondary or tertiary vertex, track the particle origin back to the primary vertex.

The identification of a particle and the measurement of its energy requires a thorough study of the particles' trajectories. To this end, the detectors are usually immersed in a magnetic field, which, by bending the charged particles, can distinguish them from

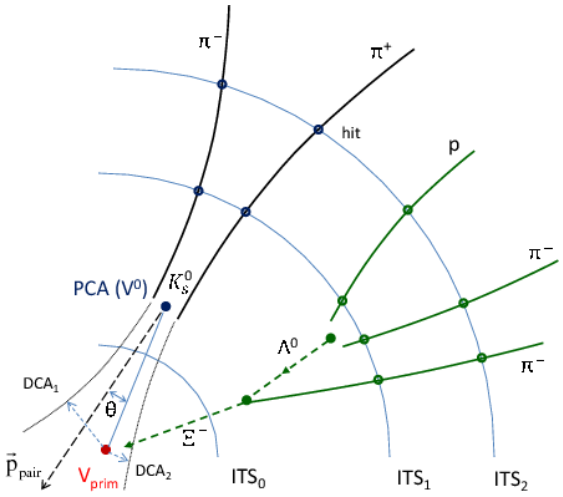

Figure 1.2: Secondary vertices are the result of the decay of the particles originated in the primary interaction point. [2]

non-charged ones, and (if they are charged) measure how much their trajectories are bent, as this allows to reconstruct the particle's momentum (mass times velocity).

Energy is measured by *calorimeters*. Calorimeters are structures capable of evaluating the particles' energy by effectively stop them, and measuring how much energy they release in being stopped.

1.1.1 The LHC Complex

In the LHC complex, there are 4 main experiments: the ATLAS and CMS experiments, which are composed of 2 general-purpose detectors, and the LHCb and ALICE experiments. In the following, we will detail the structure of ATLAS and CMS, the 2 experiments for whom the chips described in this thesis were designed.

1.1.1.1 CMS

The Compact Muon Solenoid (CMS) experiment at CERN is a general purpose detector, focused on the study of matter interaction at the TeV[1] scale, and to discover and characterize the Higgs Boson.

It comprises 5 different layers: the tracker, the electromagnetic calorimeter, the hadronic calorimeter, the magnet and the muon chambers. The experience gained by previous High Energy Physics experiments, like the UA1 experiment, in fact, clearly highlighted the need for a powerful muon triggering and reconstruction system, which in turn requires a high magnetic field. [24]

[1] 10 TeV roughly corresponds to 1.6 µJ

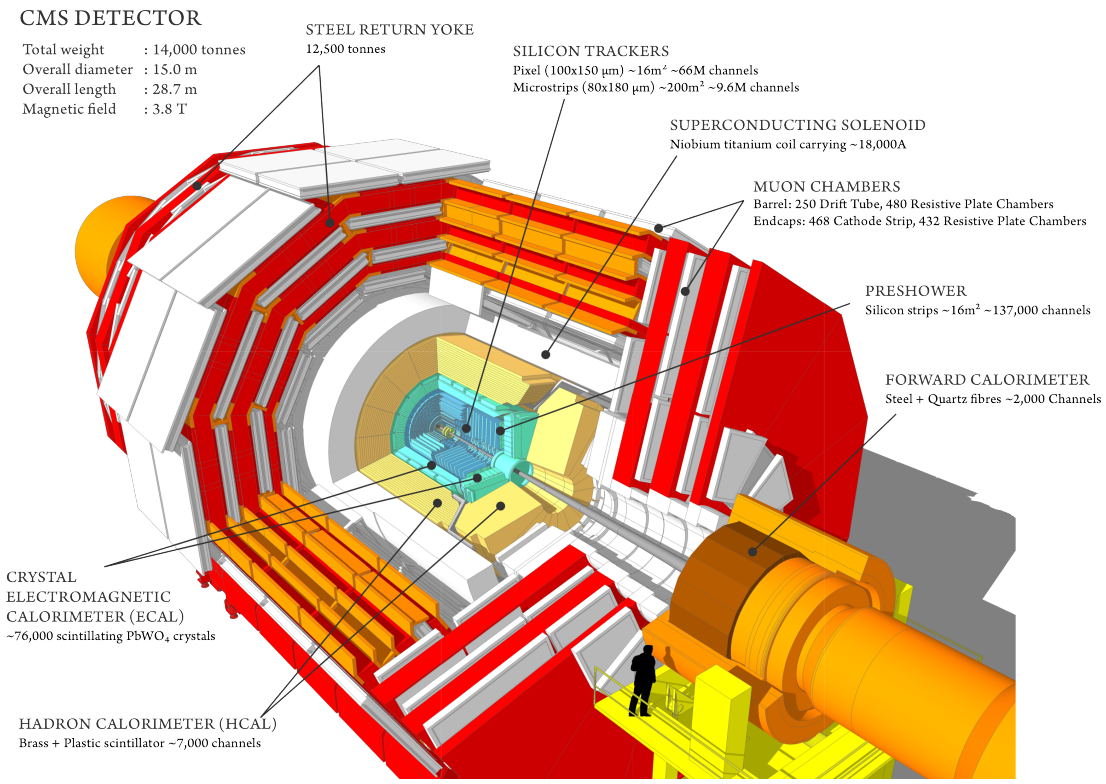

Figure 1.3: The CMS experiment features a compact design, in which the bulky solenoidal magnet is intertwined with the detectors.

This magnetic field is generated by a massive magnet, which uses almost 20 kA of superconducting current to generate a 4 T magnetic field. In order to sustain and guide the generated magnetic field, the muon chambers are interleaved with a 3-layer return yoke. [19]

1.1.1.2 ATLAS

ATLAS stands for A Toroidal LHC ApparatuS, and is the largest of the detector experiments of the LHC. It shares the same objectives of CMS, but its layout is very different.

The Inner Detector, consisting of the Pixel Detector, the Semi-Conductor Tracker (a Strip detector), and the Transition Radiation Detector, reconstructs the tracks of the charged particles, giving hints on the particles' type and momentum. It is surrounded by a solenoidal magnet which generates the appropriate magnetic field for operation.

Outside the solenoidal magnet lie the Electromagnetic and Hadronic Calorimeters, which work in a way similar to those of CMS, the toroid magnets and the muon detector system.

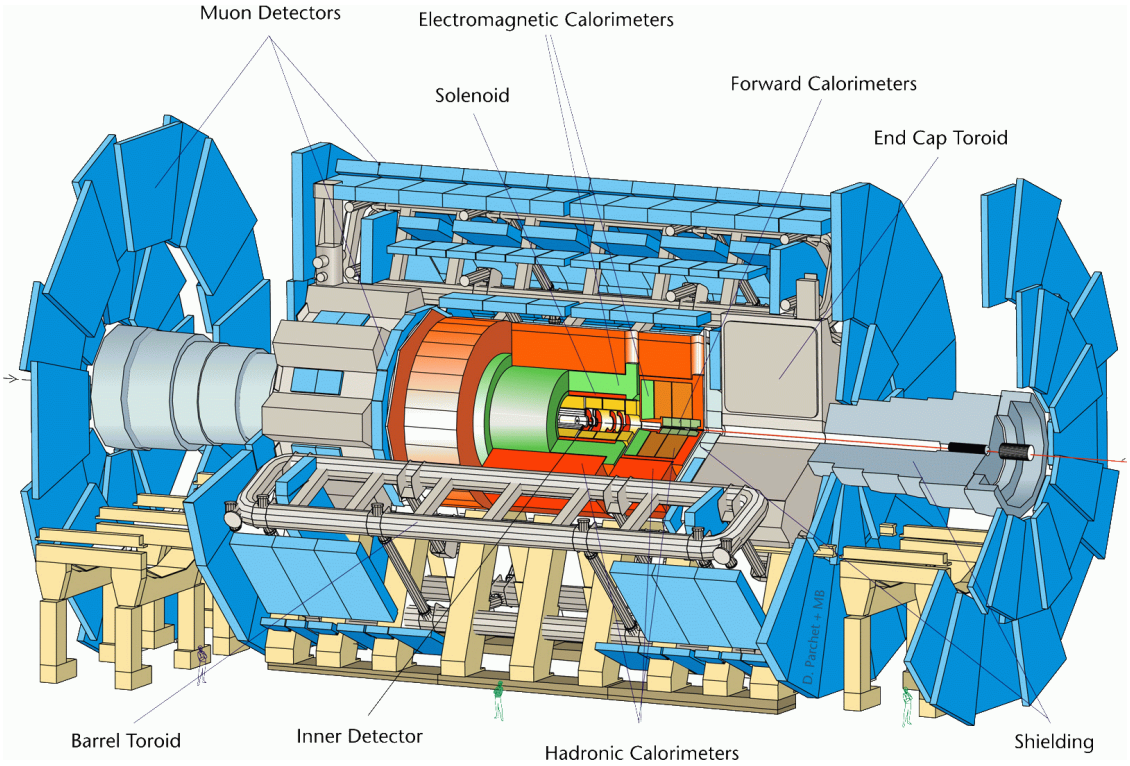

Figure 1.4: The ATLAS experiment features the largest of the detectors in the LHC path.

1.2 Trackers

The trackers are the part of the detectors which allows the reconstruction of particle track vertices. In the experiments, in order to map the particles' trajectories with great accuracy, they consist of several layers (usually 3 or more) placed at known, small, distances from the interaction point.

The track is reconstructed by appropriate algorithms that combine bi-dimensional information on the interaction of particles in all the layers. Fig. 1.5 shows the arrangement of the detector in the CMS Tracker: it can be seen how the Pixel Detector, which gives the maximum position resolution, is located very close to the interaction point. [93]

The need for very high accuracy in the track reconstruction is motivated by the physics the experiments needs to observe: the most interesting events at the LHC are likely to contain groups of particles called b-jets. A b-jet is a narrow cone of hadrons originating by the hadronization process of a b-quark. As b-jets possess a set of features that make them unique with respect to jets originating from hadronization of other quarks, it is possible to identify them in particle detectors. The feasibility of this measurement was made possible by the precision achievable with silicon detectors.

In fact, in order to allow an efficient identification (a process called *tagging*) of these

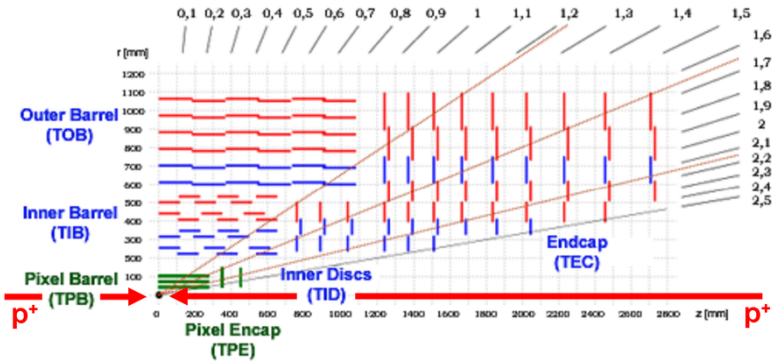

Figure 1.5: The CMS tracker is a layered structure, featuring many types of detectors. [44]

jets, as well as of other objects, the track reconstruction must extend as closely as possible towards the primary interaction vertex. Pattern recognition at the extremely high particle fluxes at these small distances requires the innermost tracking layers to be composed of silicon devices delivering true space point information with high resolution.

Each tracker layer surrounds the interaction point cilindrically at a precise distance, and can measure the position and energy (to some extent) of the charged particles that pass through it. The algorithms use these spatial coordinates and information about the applied electromagnetic field.

1.2.1 Modules

The tracker layers are composed of modular units, to which the sensing devices are connected via bond wires. Each module distributes the power supply, readout control and clock signals to the chips, and collects the physics data from them.

Connectivity to and from the modules is performed via Kapton cables, which run through the barrel to an outer region of the detector, where the control chips and electro-optical converters for optical signal transmission are located.

The modules are attached to cooling frames, with the cooling tubes being an integral part of the mechanical structure, and which represent one of the key contributors to the material budget for the pixel detector.

Matter inside the detector, in fact, needs to be as little as possible: as the particles generated in the collision escape the interaction point and traverse the detector, they are deflected by many small-angle scatters. The deflection is mainly caused by the matter in the detector through Coulomb interaction of the charged particle with the nuclei and strong interaction.

In the current LHC detectors, the scattering angle for a 1 GeV particle is $\angle 0.1$. It follows that in order to enhance the secondary vertices reconstruction, much efforts have to be put into lowering the material in the detector to a minimum. By reducing

the power consumption of the pixel modules, the cooling frame size can be reduced and so can the scattering angle. [51, 90]

1.2.2 Sensing devices

Given the precision and processing resources needed, trackers nowadays consist of silicon detectors, which have long been used for radiation detection [34]. The specific type of silicon detector for each layer, however, depends on the layer's position: the closer it is to the interaction point, the more accurate it needs be. The first layers are usually composed of costly but precise pixel detectors, while the outer layers, where the particle fluency is much lower (it decreases with the square of the distance), are made of cheaper strip detectors.

Strip (and micro-strip) detectors are made of arrays of long, thin diode strips on 300 µm thick wafers, with a strip pitch of 25 µm to 50 µm. This arrangement only allows for a one-dimensional detection of a particle, and a 2-plane configuration is needed to discern the full particle track position.

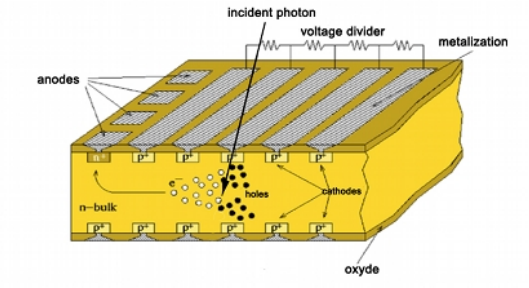

Figure 1.6: Microstrips are capable of measuring the particles' position in 1D.

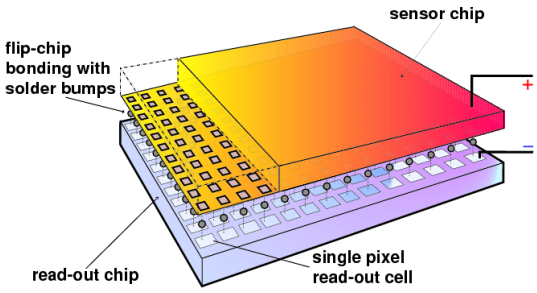

Figure 1.7: Pixel sensors can map the particles' passing point in a 2-D matrix. [98]

If the sensing elements are square, with sizes in the order of $1\,\mathrm{mm}^2$ or more, the detector is called a Pad detector. In this case the number of channels per chip are usually in the order of 10^2, and connection to electronics can be performed via wire bonding.

If, instead, the size of the sensing elements is smaller, one speaks of a Pixel detector. These detectors can have 10^4 or more channels, and a 2-dimensional connectivity as dense as the pixels themselves becomes necessary.

It follows that fewer planes of pixel detectors than strip detectors are needed to measure track trajectories. Furthermore, pixel detectors offer much better two-track separation than strip detectors since for a given (square) area a pixel detector has the square of the number of elements of an equal area strip detector with the same pitch [50].

An example of strip and pixel detectors are shown in Fig. 1.6 and Fig. 1.7 respectively.

1.2.3 Trigger

A critical requirement for the tracking detectors consists in the capability of buffering the information regarding the particle hits for a certain time period, called *Trigger latency*. High Energy Physics experiments, in fact, usually look for very rare decays, and in order to study them in a reasonable time, they need to have a very high rate of interactions, approaching 10^9 Hz. As a comparison, the Higgs Boson was expected to be produced at rates that vary between 10^{-1} and 10^{-2}. This means that the experiments need to have a minimum selectivity of about 10^{-10}.

This selection is made by *Triggering systems*, deeply embedded in the detector design, which take inputs from calorimeter and muon chambers in order to reduce the output bandwidth of the Trackers, which would otherwise be orders of magnitude greater than technically sustainable. The trigger systems use a set of algorithms to define which bunch crossings produced interesting events, but, given the finite computational power and resources available, the process takes some computational time in order to produce a result. This computational time, together with the time needed for its propagation to every module in the tracker, gives the *trigger latency*. The tracker must thus hold information regarding the hits it registered for this period of time, before discarding them if not selected, or send them out if so.

CMS Trigger CMS relies on a two-level trigger system. The Level-1 (L1) Trigger is implemented in hardware (ASICs and FPGAs) and serves to reduce the data rate from the 40 MHz of the LHC bunch crossing rate down to 100 kHz. Trigger signals must be generated by the hardware within 4 µs, a limit largely determined by the limited length of the silicon tracker's readout pipeline. The full detector is read out on receipt of a Level-1 Accept (L1A) signal, and events are built. The High-Level Trigger (HLT) is implemented in software and reduces the rate to about 1 kHz. [96]

ATLAS Trigger The ATLAS Trigger system uses information from the Calorimeters and Muon chambers in order to send a L1 Accept signal to the pixels, in a way similar to the CMS detector.

Although designed to sustain a Trigger Rate of 100 kHz, during the first runs (Run 1) of the LHC, the effective rate was intentionally limited to 65 kHz due to readout settings of some ATLAS subdetectors. The trigger system has a latency of 2.5 µs. [4, 102]

1.3 LHC Roadmap

The evolution of the pixel chips used in the detectors closely followed that of the experiments they had to be used in. In fact, the chip requirements became ever more stringent with each experiment improvement.

The LHC Roadmap, shown here in Fig. 1.8, highlights the main changes in the LHC complex and its experiments.

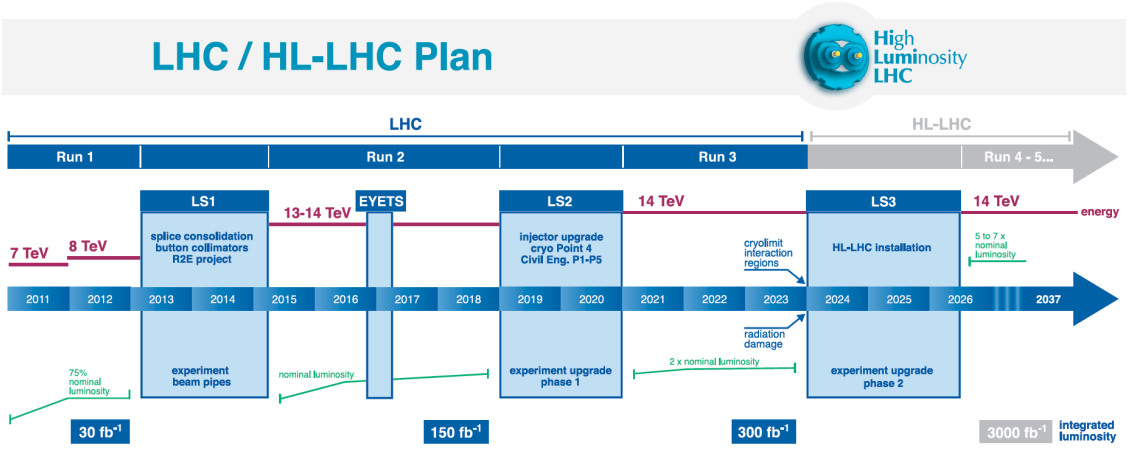

Figure 1.8: The LHC upgrade plan.

Run 1 The main experiments started collecting physics data in 2011, with the start of the Run 1. At the time, the center-of-mass energy reached by the machine was 7 TeV, later increased to 8 TeV near the end of Run 1 in 2012. The Bunch Crossing (BX) frequency though nominally of 40 MHz, during Run 1 has been always lower, as the bunch spacing during regular running was never less than 50 ns. The average and maximum number of collision per Bunch Crossings increased yearly: from 2011 to 2012, the average increased from 9 to 21, while the maximum from 16 to 34, as shown in Fig. 1.9 and Fig. 1.10. [22]

In between Run 1 and Run 2, the LHC underwent a series of upgrades, needed to increase the luminosity and reach the nominal bunch crossing frequency.

Run 2 In Run 2, between 2015 and 2018, the LHC machine reached the nominal luminosity, with 13 TeV to 14 TeV center-of-mass energy. The average event pile-up during this run, as computed by the CMS and Atlas experiments (see Fig. 1.11 and Fig. 1.12), averaged 34 proton-proton collisions per bunch crossing, with peaks reaching 60-70 collisions in 2017-2018.

Years 2019-2020 will see a major upgrade of the LHC machine, in order to experiments, called Phase 1 Upgrade, in order to prepare them to reach a luminosity of $2 \times 10^{34}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$.

Run 3 The increase in luminosity will, according to Eq. 1.1, increase the event rate correspondingly. At 14 TeV, and event pile-up of around 40.

The detectors are currently being upgraded in order to sustain the increase particle rate and radiation load.

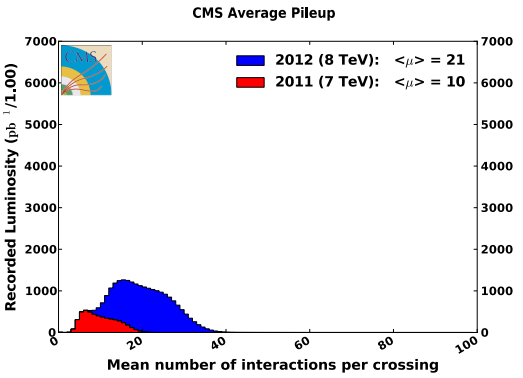

Figure 1.9: Event pile-up measured by the CMS experiment during Run 1. [80]

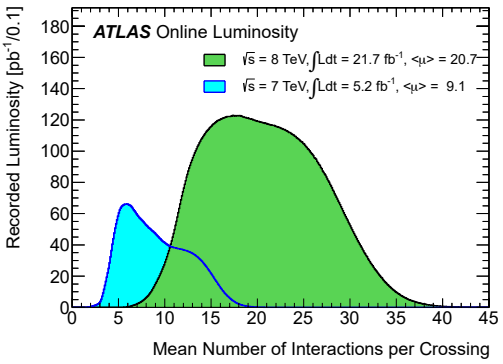

Figure 1.10: Event pile-up measured by the ATLAS experiment during Run 1. [79]

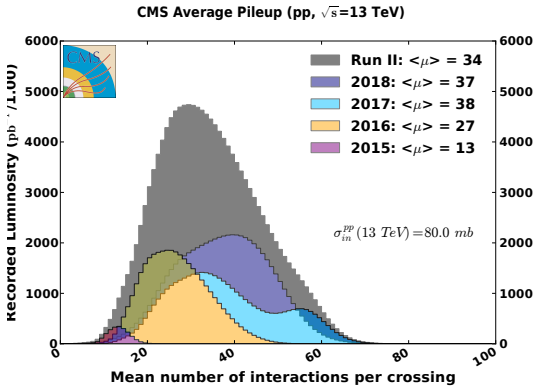

Figure 1.11: Event pile-up measured by the CMS experiment during Run 2. [80]

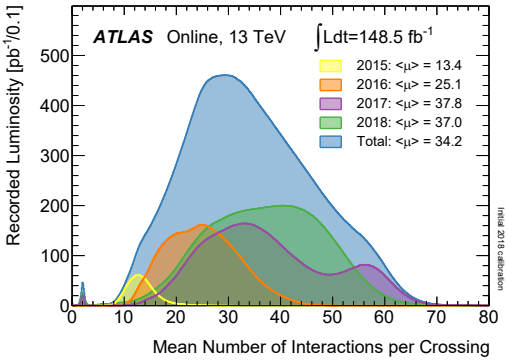

Figure 1.12: Event pile-up measured by the ATLAS experiment during Run 2. [78]

1.4 The Phase-2 Upgrade

After Run 3 the statistical gain in running the accelerator without a significant luminosity increase will become marginal: it is estimated that the time necessary to halve the statistical error of a given measurement, will be more than ten years.

Thus, a decisive increase in the LHC luminosity after 2020 is needed, and will be implemented in the upgrade called Phase 2 Upgrade, which will make the LHC reach a luminosity of $10 \times 10^{34}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$, which will translate into an event pile-up of 200.

The new luminosity of the HL-LHC will change key design parameters regarding the experiments' trackers, which will reflect in new specifications to be supported by the pixel detectors' sensors and electronics. The ones most pertinent to this work are detailed below.

Trigger CMS will maintain its two-level triggering strategy, although the entire trigger and DAQ system will be replaced, to allow a maximum L1A rate of 750 kHz, and a latency of 12.5 µs (or 500 LHC bunch crossings). In addition, the L1 trigger will, for the first time, include tracking information and high-granularity calorimeter information. [20]

ATLAS, instead, will upgrade its triggering system to employ a 2-level trigger: an L0 trigger pre-selects the events for readout, while an L1 trigger will later either confirm or reject this pre-selection. The expected rate and latency for L0 is of 1 MHz and 6 µs, while L1 would have a latency of 24 µs (to be added to the L0 latency, yielding 30 µs), at a rate of 400 kHz.

Level-0 is based heavily on the system which will already be built for an earlier upgrade of ATLAS between LHC Runs 2 and 3, while Level-1 will consist entirely of new hardware. The Level-0 trigger will run on coarse-grain inputs from the calorimeter and muon systems. The Level-1 trigger adds in finer-grained calorimeter information as well as tracking to the trigger decision.

The increased trigger latency translates into an increase in the event buffering requirements in the pixel chips in order to maintain a > 99 % efficiency.

Hit rate The Phase-2 Upgrade will also see an increase in the number of tracker layers in the CMS experiment, and a change in their configuration and radius. The comparison between the current inner layer at CMS and the proposed new scheme is shown in Fig. 1.13.

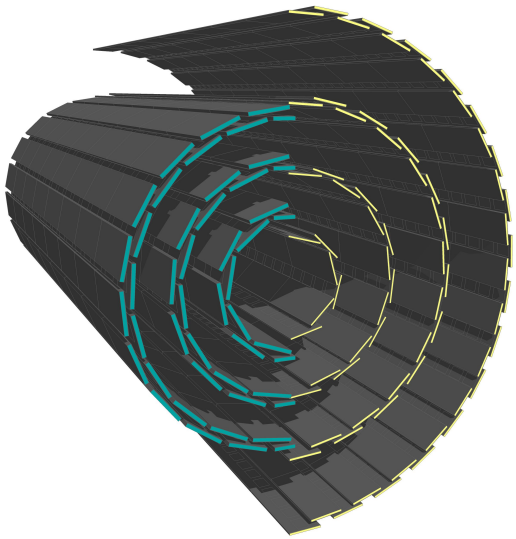

Figure 1.13: The CMS Pixel Detector is divided in 3 layers (to be upgraded to 4), with every layer composed of multiple modules.

In the new configuration, the innermost layer of the CMS tracker will move closer to the beam pipe, reducing the current distance of 4.4 cm down to 2.9 cm. The HL-LHC beam pipe radius will also decrease with respect to the current one, going from 30 mm to 22.5 mm.

These changes will improve the vertex resolution capability, but also drastically increase the charged particle rate, which will reach 750 MHz cm^{-2} in the first tracker layer. [21]

Material budget With the new layers and forward disks, the pixel detector's area almost doubles, but the overall material budget in the tracker is decreased in the new proposal, in order to reduce the particle scattering.

The reduction in material will be carried out by moving passive materials (like electronic connections) out of tracking volume, and adopting lighter mechanical and CO2 cooling supports.

In order to reduce the cooling infrastructure, the pixel detector's power consumption must decrease to below 500 mA cm^{-2}, which, for 50 µm pitch pixels, becomes about 10 µW/pixel, to be equally divided between analog and digital logic.

Granularity The granularity of the detector needs to be improved in order to enhance radiation tolerance and track separation capabilities at the new barrel distance from the beam pipe. The pixel pitch identified for the new detector is 50 µm, with pixels in the 25 µm × 100 µm or 50 µm × 50 µm form factors. By assuming 4-pixel clusters, the particle rate transforms into 3 GHz cm^{-2} single pixel hit rate.

Radiation tolerance The vicinity to the beam line will not only translate in an increased particle rate, but also a more intense radiation dose administered to the chips. In fact, the target integrated luminosity of 3000 fb corresponds to a hadron fluence of 2×10^{16} n_{eq} cm^{-2} and 1 Grad at 3 cm from the interaction region, where the innermost tracking layer is located.

Such a ionizing dose has to be supported by both sensor and readout electronics, without significant changes in the performances.

Chip Size As the manufacturing process of silicon-based integrated circuits evolve, fabrication laboratories are able to produce wider silicon wafers. This is a very important result for manufacturers, as larger wafers have bigger throughput, which means a lower cost per die.

Larger wafers also represent a big opportunity for customers, as they are able to submit bigger designs, while at the same time roughly reducing the cost quadratically with respect to area increase.

In HEP experiments, the size of the pixel chips has been steadily increasing with time, in order to reduce both the cost and the number of edge pixels. Edge pixels, in

fact, are often troublesome as they usually need different biases and configuration in order to achieve performance comparable to the other pixels in the ROC. Bigger chips also have an important advantage in hybrid solutions, as the cost of the flip-chip scales more with the number of chips than the total area covered.

As the total area of the pixel detectors will increase, the goal for next-generation ATLAS and CMS chips is to have pixel matrices of 20 cm to 16.4 cm and 22 cm to 19.2 cm respectively.

Chapter 2

Hybrid Pixel Read-Out Chips

The idea that silicon detectors might be able to be of help in the trackers for particle colliders has been around from the 60s, in hopes that this kind of apparatus could replace bubble chambers, whose data rate capability was low. This, however, was not possible at that time, as the technology was not mature enough.

In the late 70s, some trials with a layered structure hosting closely spaced silicon diodes proved that this Proof-of-Concept would work in experiments. Each diode could measure the total charge left by the passing particles and could therefore approximately deduce how many particles passed through each diode.

The first generation of microstrip detectors was introduced in the mid-80s, as testified by the 1984 CERN experiment WA82, which featured a microstrip sensor with 512 channels individually connected to hybrid amplifiers. This kind of detectors allowed much more precise measurements of particle track parameters.

In that same year it was first proposed that pixelated imaging sensors could be used, by appropriately bump-bonding it to a semiconductor diode array, to detect and track X-rays. The proposed scheme was simple, which allows integration with a small pixel size, but could not support triggers nor bandwidths faster than 1 kHz: limitations incompatible with the particle physics experiments' requirements.

It was only in the first half of the 90s that the development of hybrid pixel chips came to a full start, with efforts of both the Superconducting Super Collider (until it was dismissed in 1994) and the Large Hadron Collider experiments.

The R&D program from CERN/LHC gave birth to the project with eventually developed pixel chips actually used in particle physics experiments, specifically fixed-target experiments using heavy ions. The reason behind this choice are both the tight forward cone typical of the fixed-target experiments, which made the access to the detector setu easily accessible, and the fact that heavy ion collision typically yield many particles in their final state, and thus may benefit the most from the pixel detectors' capacity in tracks reconstruction.

This chip is the OmegaD pixel chip, used in the CERN experiment WA94. The chip

consisted of a 1024 pixel matrix with $75\,\mu m^2 \times 500\,\mu m^2$ pixels. The sensors are connected to an analog preamplifier, which is followed by an asynchronous comparator and, ultimately, a digital delay line which is used for trigger matching. The chip had a power consumption of 30 µW per pixel, and had an excellent single pixel performance (noise below $100e^-$ rms), although it featured a very wide threshold variation between individual channels (in the order of $500e^-$ rms).

However, this represented a critical step in the making of pixel chips for HEP experiments, and paved the way for future detectors, especially the trackers in LHC experiments. [91]

2.1 Structure of modern HPDs

Readout Chips for HEP experiments, just as CCDs and other imaging chips, typically separate the part of the chip devoted to charge/light detection (the pixel matrix) and that needed for overall chip operation and communication (the periphery). The exact structure of, and functions assigned to the matrix and periphery have changed with time, as technology advancements allowed for higher logic density. The efforts to push the chip capabilities further called for ever more integration of some functions directly into the matrix, offloading the periphery.

The active matrix, which for hybrid chips is bump-bonded to a compatible segmented sensor, contains the signal processing logic needed to measure the charge that traversed a pixel, and assign fine timestamps and precise positions to it. In order to minimize the noise and thus have the best Signal-to-Noise Ratio, the analog electronics that amplify and shapes the signal from the sensor is placed very close to it: directly underneath the bump pad.

The charge can be then either measured in loco or down in the periphery. This decision usually depends on the amount of information to be stored, the pixel size, and the technology node, which tells us the cell density we can achieve. It should be mentioned that some chips, like the ones for biomedical imaging, are not interested in measuring the charge of the particles traversing the sensor, but instead in the count of the charged particles per unit of time. These are referred to as Counting Chips, but they are not the focus of this thesis.

Others, instead, featured low-resolution digitization via flash ADCs. Recently, however, most of the chips feature fully-digital operation, using innovative charge measurement techniques.

Some early chips, using older technology nodes (250nm CMOS), featured analog charge readout, often implemented with a sampling capacitor in the pixels which transferred the charges down to the periphery by using a chain of multiplexers and control logic. [10] In order to fully exploit the advantages of a digital readout, however, chips steadily moved to peripheral digitization of charges [97], and pixel digitization [18].

A trigger-based filtering system is often embedded in detectors for HEP experiments, including a one or two level triggering scheme. This means, in order to reduce the bandwidths in the system, the ROCs have to be able to store the data for the trigger latency and filter out the unselected events.

In this section we will explore the main parts of a hybrid pixel chip, defining the terms later to be used in this thesis.

2.1.1 Sensor

The sensor is the part of the detector chip where it is possible to measure the radiation interaction with matter, and that delivers the measurement signal to the readout electronics. The signal is produced by ionization in the sensor material, which is usually silicon, although approaches involving diamonds are under scrutiny.

Before silicon was introduced in the 1960s, the detection principle involved gas-filled ionization chambers. But, given that the energy required for gas ionization is 20 eV, instead of the 3.6 eV needed for silicon crystal ionization, silicon-based detectors have a much greater energy resolution and thus rapidly became the detector flavor of choice for particle physics experiments.

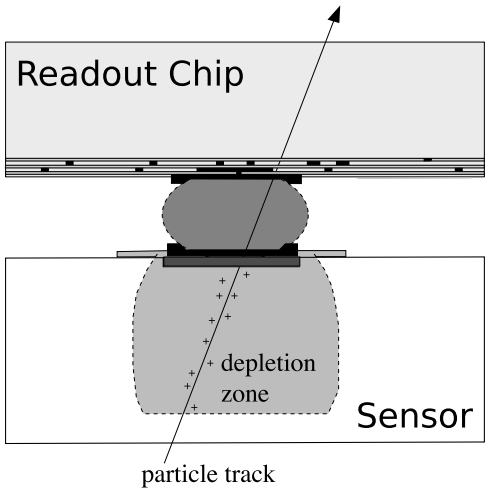

Figure 2.1: Hybrid pixel chips have a separate silicon sensor, in which a depletion region is used to collect electron-hole pairs generated by a passing charged particle. [91]

Silicon is nowadays widely used, because its electric properties are well understood, and the radiation-induced degradation well characterized, modeled and predicted. In hybrid chips, the sensor is connected to the readout electronics chip via solder bumps, which connect the sensor pixel to the readout pixel, as shown in Fig. 2.1. Both this and another integration solution, called Monolithic, are treated in this section.

2.1.1.1 Sensor orientation

When discussing the pixel position and size clusters, the cilindrical reference system of the pixel barrel is often used. In particular, the z axis points to the particle beam, while θ refers to angles in the y direction (polar coordinates), and ϕ to angles in the x direction (azimutal coordinates). In high energy physics experiments, the pseudorapidity η is often used in place of θ, as, approximately, particle production is constant as a function of (pseudo-)rapidity. The two are related by the formula:

$$\eta \equiv -\ln\left[\tan\left(\frac{\theta}{2}\right)\right] \tag{2.1}$$

It should be noted that, given the orientation of the strong solenoidal magnetic field in the detector, the charges in the silicon are affected by the Lorentz force in the ϕ direction, and thus widens clusters in x direction. In the barrel geometry the azimutal component is often small ($|\phi| < \angle 30$), while the polar one can be much higher:

- $\theta = \frac{pi}{2} \Rightarrow \eta = 0$
- $\theta = \frac{pi}{4} \Rightarrow \eta \approx 0.88$
- $\theta = \frac{pi}{8} \Rightarrow \eta \approx 1.6$
- $\theta = \frac{pi}{16} \Rightarrow \eta \approx 2.3$

2.1.1.2 Planar and 3D sensors

There are 2 main arrangements for silicon sensors: planar sensors have long been used in imaging and particle detection, while 3d sensors have been recently introduced as their feasibility has only recently become technically possible.

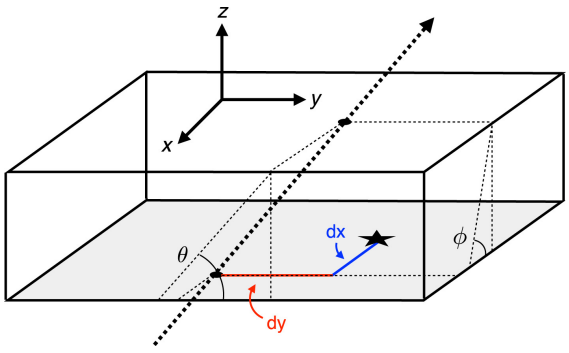

Figure 2.2: The reference system in a Pixel Chip with respect to the Detector's coordinates

Planar sensors A planar silicon-based pixel sensor is essentially a reversely biased pn-diode with a highly segmented cathode or anode. They are usually structured so that a strong backside bias is applied, and thus a large depletion zone is created. When a charged particle traverses this depletion zone, it induces the formation of electron-hole pairs. Thanks to the strong electric field, the pairs that don't immediately recombinate drift to the opposing electrodes, inducing an electric current.

This current is therefore proportional to: the amount of electron-hole pairs, defects in the lattice (charge trapping), and the weighting field. The latter, pictured in an example in Fig. 2.3, is different from the electric field that causes the drift of the charge carriers, as it refers specifically to the single electrodes. It can be obtained by applying unit potential only to the electrode under consideration.

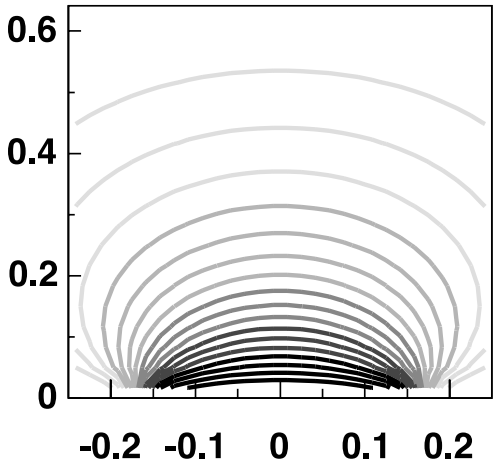

Figure 2.3: Drawing of the weighting field for a single electrode wide 1/3 of the wafer thickness. [91]

As the carriers induce a signal from the very moment when they start moving, and not only when they reach the electrodes, a small bipolar signal is induced in the neighboring pixels as the charge moves towards the corresponding electrodes. Moreover, the distribution of the weighting field causes most of the signal to be induced in the last part of the drift path, while the carriers drifting towards the backplane do not contribute significantly to the signal itself. These effects are collectively referred to as *small pixel effects*.

The signal induced to the pixels is also affected by two effects known as charge sharing and cross-talk. Charge sharing, as the name suggests, refers to the signal being distributed across several pixels, as pictured in Fig. 2.4. It is mainly due to effect of the position and angle of the track with respect to the sensor surface, but can also be strongly influenced by the electromagnetic field. A controlled charge sharing is usually desirable, as it allows to improve spatial resolution, but if it is too large, it may decrease

the signal in each pixel below the threshold, thus making part of, or the whole, cluster undetectable.

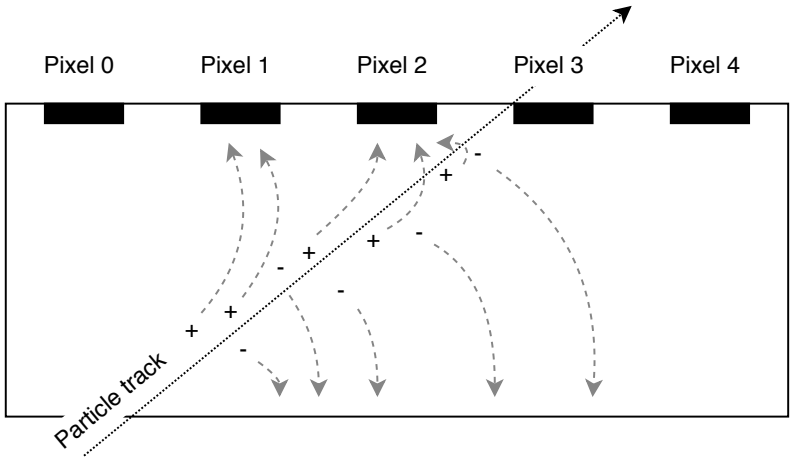

Figure 2.4: Charge sharing is due to the thickness of the detection volume, which distributed the charges originated from a track to multiple pixels

It should be noted that the high-intensity magnetic fields of the detectors is strong enough to directly influence the electron-hole pairs in the silicon, which therefore deviate from the electric field lines by the so-called *Lorentz angle* θ_L. The Lorentz effect also contributes to charge sharing.

Cross-talk, instead, is due to interpixel capacitance, which is roughly proportional to the pixel's perimeter and strongly dependent on the form factor and pixel pattern. As the input capacitance of the preamplifier which will process the sensor signal is usually at least one order of magnitude higher than the worst case interpixel capacitance, the crosstalk is often below 5%. [91, 86]

3D sensors Recent technological advances in the VLSI-MEMs context allowed for new geometries to be implemented in silicon devices. One of the most successful of these is represented by 3D sensor structures, in which the electrodes run orthogonally to the sensor surface, and, consequentially, the electron-hole pairs drift parallel to the surface.

3D sensors overcome some of the limitations of conventional planar sensors, in particular in high-radiation environments or applications which require a large active/inactive area ratio. These include detectors which operate close to the particle beam, such as pixel detectors in HEP experiments.

In fact, the three-dimensional arrangement of the electrodes allow a much faster charge collection (1 ns to 2 ns), and needs a much lower depletion voltage (10 V), because the charge colletion path is about 6 times lower than it is for planar solutions.

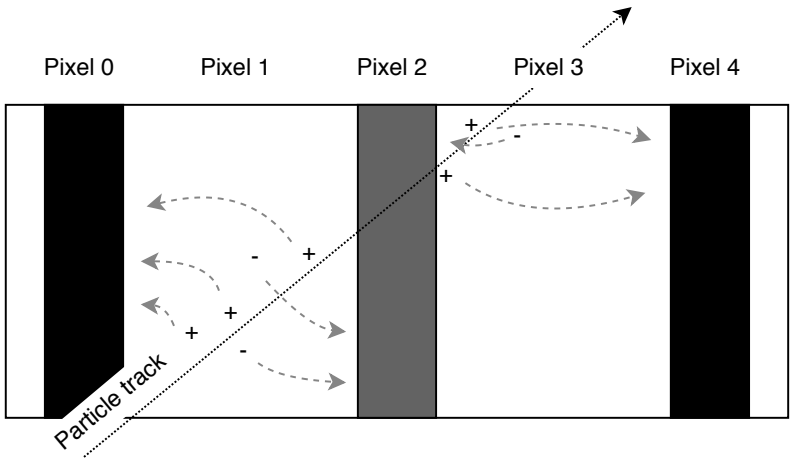

Figure 2.5: The mean collection path is much lower in 3D sensor with respect to planar sensors.

The new arrangement allows the edges of the sensor to become collection electrodes themselves, and this extends the active area of the sensor to micrometers from the sensor edge. This avoids inhomogeneous fields and surface leakage currents which occur in the planar configuration. However, the vertical electrodes introduce spots where there could be no charge collection, and that increases the inactive area.

2.1.1.3 Hybrid and Monolithic sensors

Pixel detectors can also be classified by the way the sensor and the readout parts are implemented and connected to each other.

Sensors need a wide depletion region for operation, which is more readily achieved in high-resistivity silicon wafers. Fast CMOS electronics, in contrast, require a low-resistivity medium. This incompatibility can be overcome by separating the design of the sensors from that of the readout electronics, provided that some kind of connection can be performed.

Hybrid Chips Flip-chip technology has been introduced commercially in the 60s, when it started to replace wire-bonding techniques where interconnection density and performance is key. Wire bonding, in fact, require a much larger area to be implemented, and introduce performance problems related to inductance and capacitance associated with bond wires. The structure of a flipchip process is shown in Fig. 2.6, where the bumps are used to interconnect a substrate with an Integrated Circuit.

For this reason, hybrid chips, mating sensor and readout electronics together, are the most common type of pixel detectors currently in use in HEP experiments.

Resistivity, however, is not the only key difference in the wafer production process.

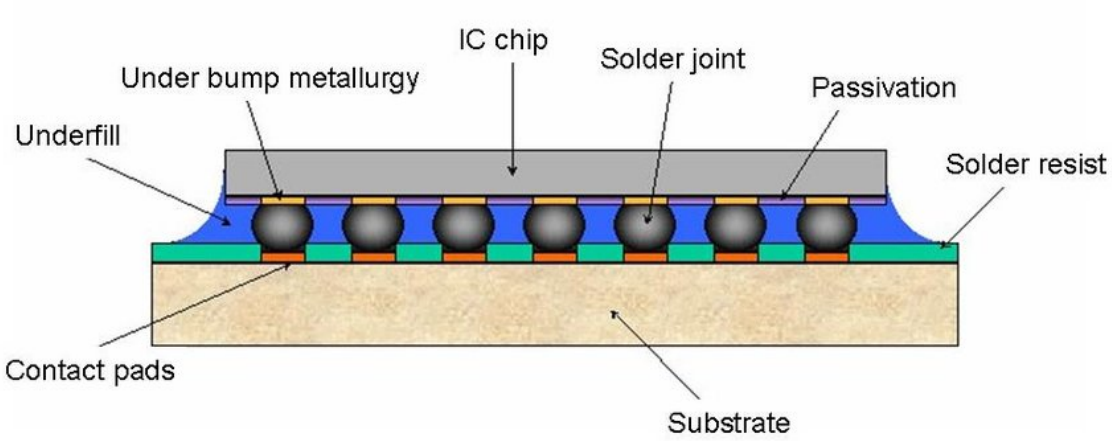

Figure 2.6: Flipchip technology allows a dense and precise interconnection between a chip and another chip or PCB board. [104]

An increasing issue for deep submicron technologies is the production yield, ever reducing as the number of transistors per square centimeter and the total chip size increases. Comparable yields can be obtained for sensor sizes an order of magnitude higher. Thus, it is not surprising that a hybrid pixel detector module usually connects together several readout chips with a single sensor [90, p. 9].

The costs associated with the production of hybrid chips are high, as they require the manufacturing of 2 different silicon wafers, and their interconnection via bump-bonding, which, although consolidated, is a costly procedure. These costs are however compensated by the performance of the system, which can optimize both sensor and readout at the same time.

Monolithic chips The complexity and costs involved in the production of hybrid pixel chips has moved the community into research for an integrated sensor/readout solution. As introduced in the previous paragraph, the main hurdle in accomplishing such a goal lies in the differences in resistivity requirements, which call for a trade-off in the performances.

Although monolithic pixel sensors have first been proposed and realized by Kenney et al. in the early 90s using high-resistivity substrates [53, 94], as technology matured implementation on both low and high resistivity bulks emerged. Two promising ones are the DEPFET and the MAPS, which are illustrated in Fig. 2.7 and Fig. 2.8 respectively.

DEPFET stands for Depleted P-Channel Field Effect Transistor. A DEPFET sensing device is a p-channel transistor located on a low-doped n-type substrate. Low-doped means highly resistive, and this, in turn, allows for a full depletion by applying a sufficiently high negative voltage to the backside sensor contact. The depleted bulk is the sensitive volume in which electron-hole pairs created by the incident radiation are separated by the electric field. While the holes move to the negatively biased backplane, the

electrons are collected in the local potential minimum below the channel of the transistor. This structure is called *internal gate* as it increases the channel charge density by induction. As a consequence, the transistor drain current, increased by a quantity proportional to the charge accumulated in the internal gate, can later be quantified and provide a measurement of the charge deposited by the passing particle. The internal gate represents a local minimum for the electric field: the charges trapped there can only be removed via a dedicated clear contact which can provide a runaway path. [15]

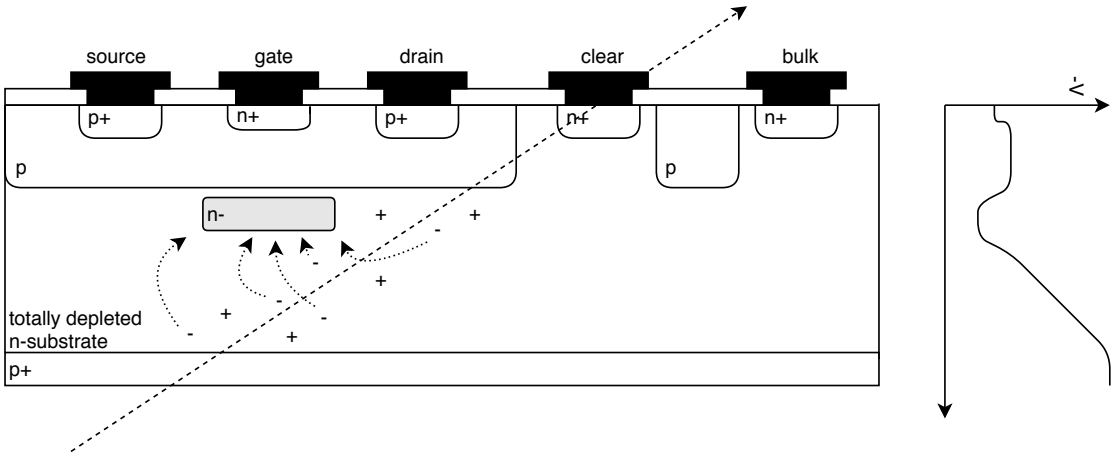

Figure 2.7: DEPFET chips use highly resistive substrates, typical of sensors, in order to easily deplete the sensing material. The internal gate, in gray, is formed by the drifting electrons generated by the passing particle ionization. On the right, the electric potential for the electrons.

MAPS, instead, stands for Monolithic Active Pixel Sensors, and is a device based on a low-resistivity bulk, the standard substrate for VLSI technologies. The depletion region realizable on such a substrate is shallow and consequently the charge collection efficiency poor. This limitation is overcome with a special structure, which takes advantage of the substrate structure of VLSI technologies, which feature twin (p and n) tubs, implanted in lightly doped, p-epitaxial silicon[1]. The depletion zone where the electron-hole pairs are produced is made of the junction existing between the n-well and the p-type epitaxial layer. The electrons produced by the radiation in the epitaxial layer diffuse towards the n-well diode contacts, which can be as many as 4 per pixel, to reduce charge sharing to neighboring pixels.

[1]This p-epi layer is grown on a highly doped p^{++} substrate

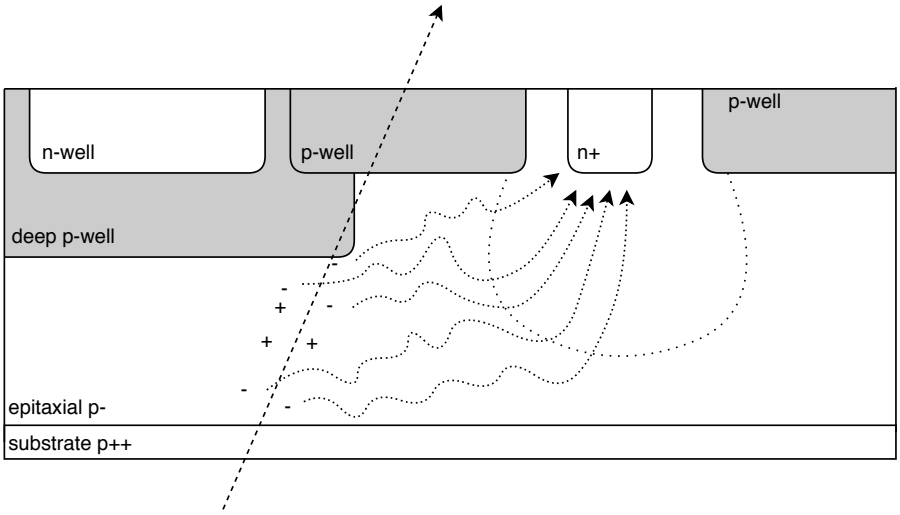

Figure 2.8: MAPS chips use low-resistance substrates, typical of VLSI technologies, which allow complex logic integration. Proper depletion is possible only close to the collecting diodes, thus, the electrons drift only in the last part of their path.

2.1.2 Pixel matrix

In hybrid chips, the sensor is connected, via bump-bonding, to the pixel matrix. The (active) pixel matrix senses the charges deposited on the sensors thanks to the presence of an analog front-end and readout circuitry. The matrix is made of logic pixels of the same area of the pixels in the sensor. In fact, even if the form factor of the sensor pixels is different than those of the readout chip, it is still possible to connect them via a suitable routing layer.

The logic in the pixel matrix of hybrid chips varies according to the chip requirements, but several key elements are nearly always present in some form or another:

1. Bump Pad
2. Charge Sensitive Preamplifier
3. Feedback circuit, with leakage compensation
4. Shaper
5. Discriminator
6. Test Charge Insertion circuitry
7. Control and Readout

The signal processing circuitry (CSA, Feedback circuit, shaper, discriminator, and TCI) are usually part of a single analog macro called the Analog Front-End (AFE). In

order to operate, the AFE needs a set of bias voltages and currents, which are usually generated in the Chip Periphery and then distributed to the whole matrix. This removes the need for dedicated bias cells in the pixels, and at the same time ensures that the whole matrix works at the same operating points. The pixels arrangement in an orderly matrix allow such biases to be distributed in vertical bias lines, starting at the bottom and propagating all the way to the top.

As for the readout, early ROCs feature whole analog operations, with the charge collected by the pixel being sent out of the chip for external digitization via analog multiplexers. As more innovative technology nodes were used, charge digitization was first moved inside the chip, in the periphery, and ultimately in the pixels themselves.

The kind of charge storage strongly depends on the application and operating requirements, and also influences the way readout control is implemented. If digitization is performed in the periphery, the digital control is minimal: the pixels make the periphery aware that it has a charge information it needs to transfer, and activates, when requested, the analog multiplexer for the shared bus. If digitization is performed in the pixels, instead, they usually store also the timestamp information for a smart readout.

Digitization in the pixels can be either performed by means of analog-to-digital converters (ADCs), or with a technique known as Time-Over-Threshold (ToT). The ToT technique consists in the implementation of a constant current discharge in the feedback circuit of the CSA, so that its output produces a saw-tooth-shaped signal with width proportional to the input charge. By measuring this pulse with a digital counter, a digitized charge information is available.

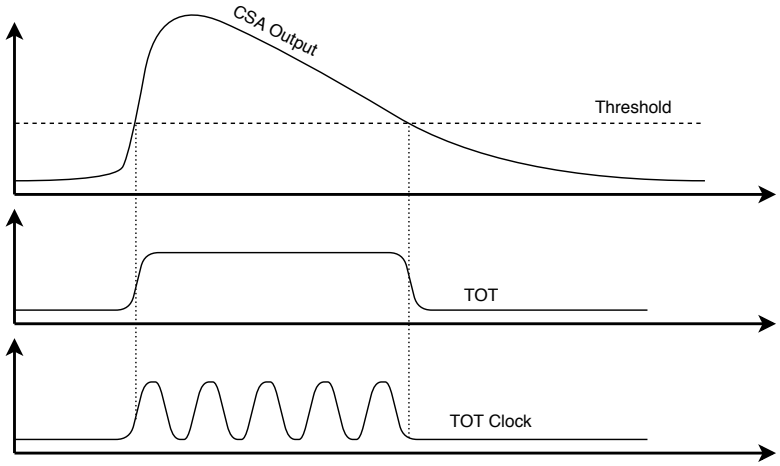

Figure 2.9: The Time-over-Threshold technique translates the charge measurement into a time measurement thanks to a constant discharge current. By using the ToT signal to gate a counter clock, this information can be easily digitized.

The pixels should also be able to retain some form of configuration. It is important

to be able to mask noisy pixels, or to select which ones to inject test charges to, and, if the discriminator uses a local DAC for threshold fine tuning, these DAC bits as well.

Moreover, they can be grouped in Pixel Regions, and the Pixel Regions themselves can be grouped again to form Pixel Cores. [38] Pixel grouping can be an efficient way to share logic and thus achieve higher performances and lower power consumption. Be they in the single pixels, Pixel Regions, or Pixel Cores, new generation ROCs usually feature data buffers in the matrix, in order to cope with the increasing hit rates and limit the bandwidth to the periphery. In triggered systems, the buffers can be quite bulky.

Both triggered and triggerless modes, however, will need some form of data propagation from the matrix to the Chip Periphery. Although various implementations are possible, a simple yet commonly used solution involves a data bus shared between all the pixels in a column. If Pixel Regions or Cores are used, the data bus can be shared in a Pixel Region Column or Core Column. A dedicated control logic should decide which pixel/Pixel Region/Pixel Core propagates its output, while the others wait for their turn. The shared bus is usually implemented via multiplexers as the tri-state buffers' power consumption makes them nonviable.

2.1.3 Periphery

The chip periphery contains the chip configuration and chip-wise logic. Among this, we can mention:

1. Output data serializer
2. Trigger and timestamp distribution
3. Pixel Matrix control logic
4. Data buffering and Event Reconstruction

As previously discussed, new generation chips usually store the charge information inside the active matrix. In order to allow for event reconstruction, then, a timestamp information, identifying the bunch crossing the charges belong to, must be attached to them. This timestamp is generated in a counter in the Chip Periphery, and usually encoded using Gray before being propagated to the columns, to save bit flips (power) and avoid intermediate spurious timestamps.

In a chip with triggered operation support, a trigger timestamp, equal to the timestamp minus the trigger latency, must also be propagated to the active area (See App. D). This is due to the fact that since the timestamp is usually encoded in Gray, there is no easy way to perform arithmetic computations on it, especially under the strict power and area constraints enforced in the pixels. It follows that the trigger timestamp needs itself to be converted in Gray, so that the pixels need only compare the timestamps of the buffered charges with that of the trigger timestamp to check for selected events.

The periphery also contains all the analog macros devoted to the generation and distribution of bias voltages and currents for the Analog Front-Ends, and the drivers and receivers for the chip I/O. In this regard, it should also be noted that the I/O pads extends beyond the sensor to allow for connectivity via bump-bonding techniques.

2.1.3.1 Readout mechanism

When laying out the chip structure, it is important to define the readout and triggering mode beforehand, as the chosen communication protocol must be supported by both the pixel matrix units and the periphery. In particular, the periphery should replicate a readout block for every column which takes care of the column readout by propagating the triggers, along with the trigger timestamp, to allow the pixels to check if they have a corresponding hit.

If a single40 MHz clock is used, as is often the case to avoid Clock-Domain-Crossing hurdles, to save power, and have a reasonable timing contraint for column-wide propagation of signals, the readout of a hit information in a column takes25 ns times the average number of pixels hit in a column (occupancy).

Triggers, however, can arrive at any time: the system must be capable of handling a second trigger while the data from the first one is being downloaded. Systems specification, indeed, often include a requirement for many more triggers to arrive consecutively (as many as 8 or 16).

2.1.3.2 Trigger latency evaluation

A simple triggering system assigns a timestamp to each recorded event, and selects a stored event for download if a trigger signal arrives after the trigger latency. This displacement in time can be evaluated in different ways, but all the following implementations require a minimal Finite State Machine.

It should be noted that the Trigger Matching operation should not only send out the triggered data, but also invalidate all the entries not selected by the trigger. If any of these events persist in the buffer, in fact, they could be later be associated with another trigger event. This problem, which is a kind of aliasing, is due to the limited number of timestamps which can be expressed with the Timestamp bits: they must be able to distinguish events at least until the trigger latency, but it is also very important not to use more bits than necessary, to keep the area down.

If T bits are used for the timestamp, then, it means that an event which happens at time A will have the same timestamp as an event which happens at time $A + 2^T$. The aliasing problem is illustrated in Fig. 2.10.

2.1.4 Configuration

There are usually 2 kinds of configuration in a Pixel Chip: a chip-wise configuration, and a per-pixel configuration.

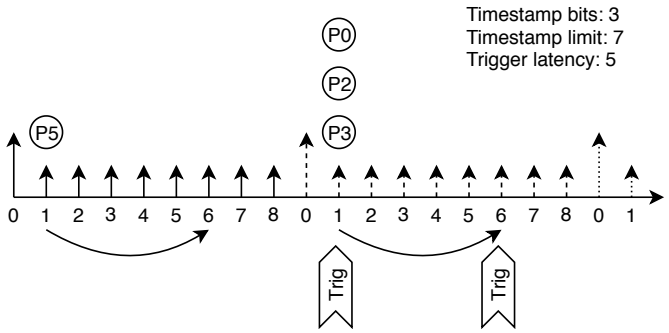

Figure 2.10: An example of aliasing. The different arrow style represent different timestamp windows, in which the timestamps are repeated even though they refer to different events. Should the event at solid-arrow time 1 not be cleared, it would be downloaded alongside the event at dashed-arrow time 1 when the trigger comes at dashed-arrow time 6.

The pixels' configuration depend on a number of factors, such as the readout mode, but usually always involve some test injection configuration bits, and a masking bit. Pixels can be injected with test charges, simulated with a dedicated analog or digital line, in order to test the analog front-ends and the digital logic even before a sensor is attached. These tests are very useful as it is possible to simulate the injection of a range of charges very precisely.

It can be very useful to select the pixels you want to inject the charges to, and therefore a pixel configuration bit is dedicated to this feature.

Another feature virtually always required is the possibility to mask some pixels: mismatches or radiation damage can make some pixels "noisy". Noisy pixels are pixels whose output is not correlated with the input charge, or very weakly so. For this reason, such pixels are selected (selection depends on the threshold voltage setting), and masked out. Masking can either be analog (by shutting of the analog-front-end), or digital. A pixel configuration bit is used to mark the pixels which need to be masked out.

Analog Front Ends usually need a way to compensate threshold mismatches inside the Pixel Matrix. If this compensation is performed via a Digital-to-Analog Converter, the digital bits used to configure the compensation voltage (or current) also need to be stored in the pixels. The number of bits used for such compensation is usually 3 or 4.

The configuration of the matrix should involve the possibility of both programming single pixels and whole clusters of pixels together. The reason behind this second requirement is that it is desirable to re-program often the whole matrix with a single command.

Another class of configuration parameters is dedicated to the configuration of peripheral components, such as the bias current or voltage DACs' settings, the generic chip-wise configuration bits (triggered/triggerless mode, ToT/binary output, trigger latency, serializer settings).

In contrast to pixel configuration bits, which are rarely SEU-protected, as the area requirements for the pixels is very stringent, the peripheral configuration bits are usually triplicated.

2.2 Examples of front-end ASICs for HPDs

In this section, we will go through the main hybrid pixel chips developed for HEP experiments and other applications. In this review, the chips are organized according to their readout type, which, in turn, depends on the application the chip has been designed for. The classification is made in 4 macro categories: analog readout, binary readout, digitized charge readout, and counting chips.

2.2.1 Counting Chips

A particular class of chips is not interested in the measurement of the energy deposited by the particles, but instead on the number of detections, in a certain time frame. Such chips are often used in medical applications, or, in general, in X-ray imaging.

Medipix2 Medipix2 chip is a pixel readout chip consisting of a matrix of 256×256 pixels, each working in single photon counting mode. Each pixel cell has a square area of $55\,\mu m \times 55\,\mu m$, and is connected to a sensor counterpart via bump bonding.

Every pixel features a preamplifier, whose feedback network provides leakage compensation for the sensor, and a shift register. Depending on the matrix shutter mode, the shift register can be used as a pseudo-random counter, which is increased if the preamplifier output fall within a predefined energy window (a condition verified by using 2 threshold comparators). If the shutter is closed, the shift register is instead used to download the data of every pixel in the column.

Each cell also has an 8 bit configuration register which allows masking, test-enabling, and threshold trimming with 3 bits of resolution for each discriminator.

The chip is designed and manufactured in a 6-metal 250nm CMOS technology node, a great improvement over its predecessor, Medipix1, developed in $1\,\mu m$ CMOS, with $170\,\mu m \times 170\,\mu m$ pixels. A summary of the characteristics of Medipix2 can be found in Table. 2.1. [58]

Medipix3 The Medipix3 chip is a hybrid pixel detector readout chip working in Single Photon Counting Mode. Like its predecessor, Medipix2, it features a matrix of 256×256 pixels, with $55\,\mu m$ pitch.

The new design has been scaled to a 8-metal CMOS 130nm technology, in order to accommodate for more features.

Medipix3 has 2 discriminators per pixel, just as in Medipix2, but also employs 2 counters, that can be programmed in multiple ways: if the *Sequential* readout mode is

Readout chip	Medipix2
Submission	1998
Application	X-ray imaging - photon counting
Technology	CMOS 250nm
Radiation Hardness	0.3 Mrad
Readout	LVDS
Chip size	16 mm × 14 mm
Pixel size	55 µm × 55 µm
Pixel matrix	256 × 256
Max counting rate	1 MHz/pixel
Power consumption	8 µW/pixel
Chip dissipation	500 mW

Table 2.1: Summary of the Medipix2 chip

selected, each discriminator increments one counter, while in the *Continuous* mode, the lower threshold discriminator increments one counter while the other can be readout.

The counters can be used in *Single Pixel* mode, where each pixel works in photon counting mode independently, or in *Charge Summing* mode, to reduce the spectral distorsion arising from charge sharing.

The chip also supports connectivity to 110 µm pitch sensors via the *Spectroscopic* mode. In this mode, the 4 55 µm ROC pixels' functionalities become available to the 110 µm sensor one, allowing up to 8 energy thresholds and counters.

Medipix3 is the first large scale mixed-mode chip in CMOS 130nm to be available for the High Energy Physics community. It was submitted in 2005, with the first tests in March 2006. [6, 7, 5]

Eiger The Eiger ROC was developed for single photon counting detectors for synchrotron radiation. It was designed for multiple-chip integration, in order to connect 8 chips together to a 38 mm × 77 mm monolithic silicon sensor.

Its 75 µm × 75 µm pixels feature configurable, double counters for continuous readout. The matrix is readout in frame-mode, where every row is sequentially selected, and its contents sent to the periphery via a parallel bus. Instead of transmitting the signal with CMOS levels, the transfer is made with 10 µA current steps.

The chip has a dead-time ±3 µs dead time between each frame, due to the time necessary to reset the buffering and counters. [81, 28]

Readout chip	Medipix3
Submission	September 2005
Application	X-ray imaging - photon counting
Technology	CMOS 130nm
Readout	8× LVDS → 1.6 Gbps
Chip size	17.3 mm × 14.1 mm
Pixel size	55 µm × 55 µm
Pixel matrix	256 × 256
Readout time	<500 µs
Power consumption	8 µW to 15 µW/pixel
Chip dissipation	∼1 W

Table 2.2: Summary of the Medipix3 chip

Readout chip	Eiger
Submission	circa 2010
Application	X-ray imaging - photon counting
Technology	CMOS 250nm
Radiation Hardness	∼6 Mrad
Chip size	19.3 mm × 20 mm
Pixel size	75 µm × 75 µm
Pixel matrix	256 × 256
Readout time	∼50 µs
Power consumption	-
Chip dissipation	-

Table 2.3: Summary of the Eiger chip

Samsung An interesting chip by Samsung for photon counting application [55] features a 128 × 128 pixel matrix.

Developed in CMOS 130nm, the chip has a pixel pitch of 60 µm, with each pixel capable of performing 3-bins energy binning thanks to 2 thresholds and 3 separate 15-bit counters. This means that the chip is capable of performing *color* X-ray imaging.

The data in the pixels is read via column buses, which download the rows sequentially before sending the data off-chip.

Readout chip	Samsung
Submission	circa 2011
Application	X-ray imaging - photon counting
Technology	CMOS 130nm
Radiation Hardness	No
Chip size	8.8 mm × 8.8 mm
Pixel size	60 µm × 60 µm
Pixel matrix	128 × 128
Power consumption	4.6 µW

Table 2.4: Summary of the Samsung chip

2.2.2 Analog Charge Readout

Among the chips which feature analog readout are the chips used for the first Runs of the CMS experiment. These include the **PSI43** chip, a precursor of the PSI46V2 chip used in the CMS experiment during Run 1. [32, 92]

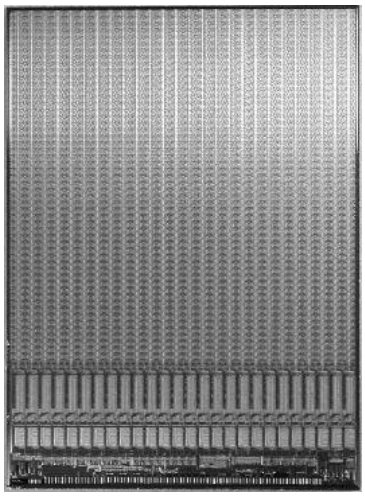

Figure 2.11: The PSI43 pixel chip.

PSI43 The PSI43 ROC, pictured in Fig. 2.11, features an active area of 53 pixel rows and 52 columns, which, for 150 µm × 150 µm pixels, make a total area of 8 mm × 8 mm . Pixel Unit Cells (which contain the pixel logic) are arranged in double columns: pairs of stacked PUCs, mirrored horizontally. This allowed for a net separation of the analog and digital blocks both at the microscopic level (PUC) and macroscopic level (matrix-wise). The chip, therefore, contains 26 double columns.[9]

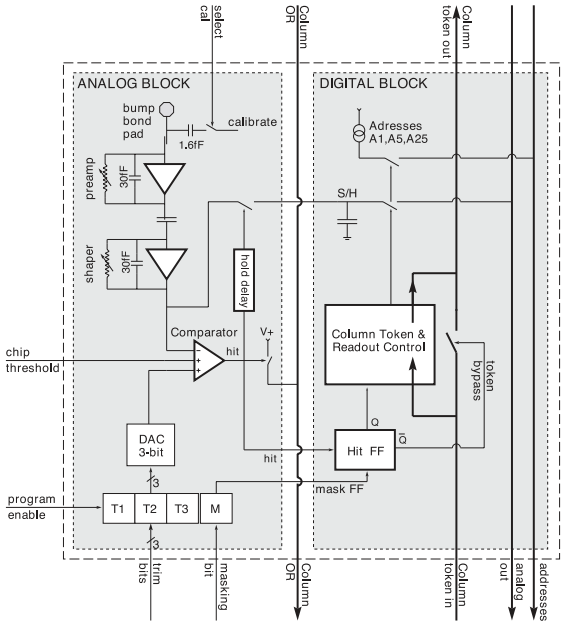

Fig. 2. Schematic layout of pixel unit cell.

Figure 2.12: The PSI43 Pixel Unit Cell. [9]

The Pixel Unit Cell in the PSI43 prototype performed hit recognition via a threshold comparator which uses both a global threshold and a pixel threshold trimming DAC to correct for pixel variations. The threshold comparator produces a *hit* signal, which is propagated to the chip periphery and is locally latched in a Flip Flop. The *hit* signal also enables the sample and hold circuitry: the charge signal is stored locally, and will later be read out by the periphery along with the pixel's address.

Every double column has an independent readout, which starts at the bottom of the odd column, and ends at the bottom of the even one. Hits are readout one event at a time: if a pixel in a double column is hit, the single Pixel Unit Cell is frozen till its information has been readout, while the other pixels in the double column can temporarily store another event, which will be read out after the current one. In other words, the double column support only one pending column drain: this source of data loss is therefore named *column busy*. Another, greater, source of data loss is given by a double-column setup time which prevents the processing of 2 consecutive events in a double-column (a loss source called *CD setup*).

The charge information drained by the double-columns is stored in the chip periphery, awaiting trigger. If an event is triggered, the chip uses its open collector output to send , for each hit pixel, an analogue conversion of the pixel address, followed by the sampled charge level.

The PSI43 chip used a radiation-hard technology called DMILL, which stays for *Durci Mixte sur Isolant Logico-Linéaire*. The device density and connectivity of this 0.8 µm SOI Bi-CMOS process with two metal layers is low compared to other commercial technological nodes, but the radiation hardness was expected to make chips tolerate more than 10 Mrad of Total Ionizing Dose.

The overall efficiency of the PSI43 prototype for beam particles was found to drop from 98% to 94% for highest beam intensities which is lower than expected from data-losses alone. A summary of the performance of the PSI43 chip is shown in Tab. 2.5.

Readout chip	PSI43
Submission	2002
Application	CMS Run 1
Technology	DMILL
Radiation Hardness	10 Mrad
Readout	40 MHz Analog
Chip size	10.8 mm × 8 mm
Pixel size	150 µm × 150 µm
Pixel matrix	53 × 52
Trigger rate	30 kHz
Trigger latency	3.2 µs
Particle rate	12 MHz cm^{-2} to 22 MHz cm^{-2}
Inefficiency	2% - 4%
Power consumption	40 µW/pixel
Chip dissipation	500 mW

Table 2.5: Summary of the PSI43 pixel chip

PSI46 In order to improve performance and yield and to reduce costs, the PSI43 design has been later migrated to a commercial process with much smaller feature size (250nm): the PSI46 chip. By featuring a device density about three times higher than that of the DMILL prototype, the number of pixels had been increased by 50% while keeping the same size of the active area.

The pixels, thus, have been scaled from 150 µm × 150 µm to 100 µm × 150 µm. The new technology allowed for an increase in the number of metal lines increased from 2 to 5, finer trace pitches, higher intrinsic device speed and an higher production yield. In order to minimize design risks, the PSI46 follows closely the architecture of the PSI43. Both PSI43 and PSI46 have an active area of about 8 mm × 8 mm organized in 26 double columns, but the lower pixel height make the PSI46 chip have 80 pixel rows (the PSI43 chip had 53).[31]

The increased transistor density achievable in this technology allowed for 2 major improvements over the PSI43 design: the two main data loss sources, the *CD setup* and the *column busy*, have been overcome. Moreover, the data buffers in the periphery have been increased by almost 50%. The PSI46 chip has therefore a much higher chip efficiency, supposedly < 2% even in the innermost CMS tracker layers.

Readout chip	PSI46
Submission	August 2003
Application	CMS Run 1
Technology	CMOS 250nm
Radiation Hardness	25 Mrad
Readout	40 MHz Analog
Chip size	9.8 mm^2 × 8 mm^2
Pixel size	100 µm × 150 µm
Pixel matrix	80 × 52
Trigger rate	30 kHz
Trigger latency	3.2 µs
Particle rate	25 MHz cm^{-2}
Inefficiency	<2%
Chip dissipation	120 mW

Table 2.6: Summary of the PSI46 pixel chip

The PSI46 chip came in 2 more iterations, the PSI46V2 and PSI46V2.1, which can tolerate a hit rate of 120 MHz cm^{-2}, although with losses in the order of 3.5%. [52, 89, 42]

Monch The Monch [29, 82] pixel chip by PSI is a charge integrating Read-Out Chip for high resolution, low noise and high dynamic range applications.

The very high spatial resolution is achieved with 25 µm pitch pixels, which however also makes logic integration and bump bonding challenging.

The chip has an analog readout, with the signal from the sensor passing through a preamplifier and a correlated double sampling stage before arriving to a set of storage capacitors.

The pixels' readout is parallelized over 32 groups (supercolumns) of 25 columns and 200 rows each. Each supercolumn of 5000 pixels is readout by a 32 MHz ADC, with a maximum frame rate capability by design of about 6 kHz.

Readout chip	Monch
Submission	circa 2013
Application	High resolution X-ray imaging
Technology	CMOS 110nm
Radiation Hardness	No
Readout	Differential 6 kHz (frame rate)
Chip size	$10\,\mathrm{mm}^2 \times 10\,\mathrm{mm}^2$
Pixel size	$25\,\mu\mathrm{m} \times 25\,\mu\mathrm{m}$
Pixel matrix	400×400
Max rate	100 photons $\mathrm{mm}^{-2}\,\mathrm{s}^{-1}$

Table 2.7: Summary of the Monch pixel chip

2.2.3 Digitized Charge Readout

The type of readout on which the works of this thesis are based has been extensively adopted in recent chips for HEP experiments. The digitization of the charges in the chips allow for a much faster and precise data transmission. In the following, 2 chips developed for the CMS and ATLAS first runs are described, along with 2 general purpose chips with charge discretization capability, derived from the Medipix experience.

PSI46dig The first CMS chip to introduce digitization of charge is the PSI46dig chip, designed in 250nm CMOS technology, the same of its predecessor PSI46v2, but with 6 metal layers instead of 5. The design has also been reviewed in a way to resist to over 250 Mrad of TID.

The Pixel Unit Cell of the PSI46dig has been improved in order to suppor faster double column readout, but elsewise is very similar to that of PSI46v2. The hits are stored in a sample and hold circuit in the PUCs, but in the new ROC, they are digitized in the Double Column periphery when triggered to read to 8-bit values. [97]

The PSI46dig chip was developed for the 2017 CMS pixel detector upgrade, supporting 25ns timing resolution and sub-pixel spatial resolution through charge interpolation, while also reducing the data losses drastically through increased readout speeds and deeper data buffers. The ROC is expected to lose $< 3.8\%$ of data with particle rates of 580 MHz cm^{-2}, while only 1.6% with particle rates of 150 MHz cm^{-2}.

The pixel size has not changed from its predecessors: 100 µm × 150 µm. The PSI46dig design was submitted in 2013. [42]

Readout chip	PSI46dig
Submission	2013
Application	CMS Phase 1 Upgrade (Run 3)
Technology	CMOS 250nm
Readout	160Mbit/s LVDS
Chip size	10.2 mm × 8 mm
Pixel size	100 µm × 150 µm
Pixel matrix	80 × 52
Trigger rate	30 kHz
Trigger latency	3.2 µs
Particle rate	150 MHz cm^{-2}
Inefficiency	<1.6%
Chip dissipation	120 mW

Table 2.8: Summary of the PSI46dig pixel chip

FE-I3 The ATLAS experiment began developing its pixel chips since the second half of the 1990s. The first prototypes were produced in 0.8 µm technologies: the FE-A and FE-C chips used CMOS technology, while the FE-B chip BiCMOS technology. These were submitted for production in 1998. The subsequent chip was developed using DMILL technology, merging concepts from FE-A/B/C into a common layout. Due to very low yields, however, development moved to another radiation-hard technology, which would have led to FE-H, but the chip was never submitted because of large cost increase.

The phase-out of traditional radiation-hard technologies pushed the collaboration towards deep submicron technologies: work started with a commercial 250nm CMOS process with a radiation-tolerant layout. A major design effort was initiated in September 2000. Three versions, FE-I1/2/3, were eventually produced, with the final chip (FE-I3) available in late 2003.

The FE-A column readout architecture used a shift register to transport the hit address to the bottom of the chip. Hits were associated with the level 1 trigger (L1) by counting the number of clock cycles needed for the hit to reach the bottom of the column. This mechanism was however later replaced in favor of a timestamp-based approach, which was eventually implemented in FE-I3.

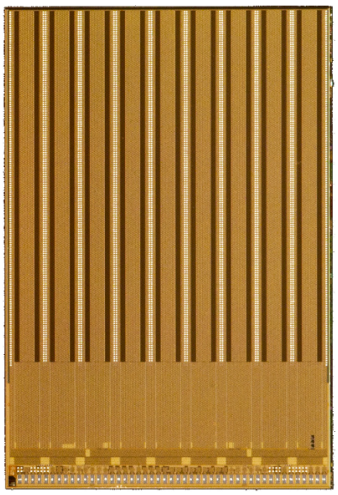

Figure 2.13: The FE-I3 pixel chip. [37]

In the new approach, the chip periphery propagates a gray-encoded timestamp to the PUCs, which would latch it in local latches, using purely combinatorial logic, when the front-end comparator rises (leading edge) and falls (trailing edge), as shown in Fig. 2.14. The LE and TE timestamps of a PUC pair are then sent to the chip periphery after the TE pulse with a priority mechanism that selects PUCs with data starting from the top row.

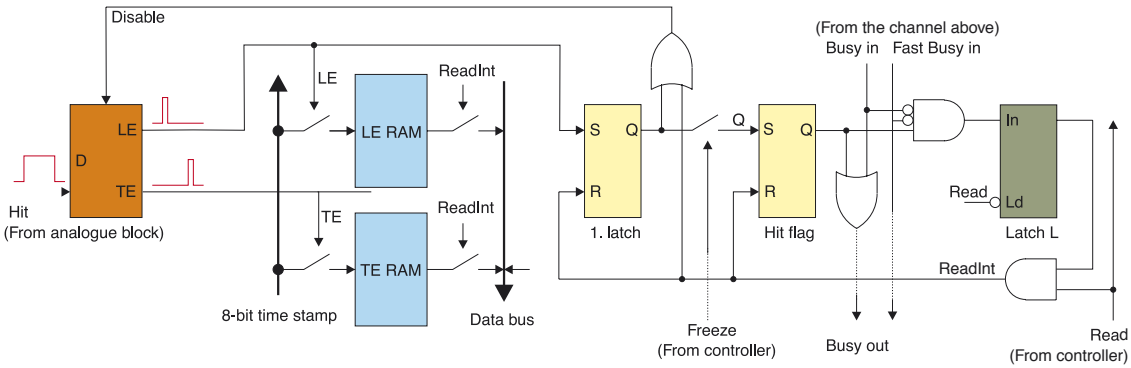

Figure 2.14: The FE-I3 pixel logic. [74]

The topmost cell with a hit transfers its data to the bus, inhibiting all the cells below.

When the cell readout is over, it releases the priority encoder bus and subsequent hits are selected and put on the readout bus. The ToT computation is performed in the chip periphery by subtracting the trailing edge timestamp from the leading edge one. [1, 74, 37]

Readout chip	FE-I3
Submission	2003
Application	ATLAS Run 1 Pixel Detector
Technology	CMOS 250nm
Radiation Hardness	100 Mrad
Readout	digital, differential
Chip size	11 mm × 7.4 mm
Pixel size	50 µm × 400 µm
Pixel matrix	160 × 18
Trigger rate	65 kHz
Trigger latency	3.2 µs
Inefficiency	3%
Power consumption	40 µW/pixel
Chip dissipation	140 mW

Table 2.9: Summary of the FE-I3 pixel chip

Timepix The Timepix chip was developed for the Time Projection Chamber of the International Linear Collider. Its development is heavily based on that of the Medipix2 chip described before, including its general organization, floorplan, and technology, but included more functionalities and flexibility. Such measures were taken in order for the new chip to be almost completely compatible with the Medipix2 readouts.

Among the novel features introduced by the Timepix chip, is the possibility of operating the pixels in 2 additional modes: the ToT mode, and the Time-Of-Arrival mode. In order to allow these additional measurements, an external reference clock is propagated to the pixels. In both the new modes, the pixel counter is clocked by this reference clock: in ToT mode, for the duration of the Hit signal; in TOA mode, since the discriminator output rises and until the matrix is readout. The external reference clock is distributed within 50ns to the whole matrix, with alternated phase between columns.

Another significant change has been implemented in the analog part of the pixel, and consists in the removal of the second discriminator, in order to achieve single

threshold operation, with 4 bits threshold trimming. [59]

Readout chip	Timepix
Submission	2006
Application	TPC-GEM
Technology	CMOS 250nm
Radiation Hardness	0.25 Mrad
Readout	LVDS
Chip size	16 mm × 14 mm
Pixel size	55 µm × 55 µm
Pixel matrix	256 × 256
Max counting rate	1 MHz/pixel
TOA resolution	25 ns
Pixel deadtime	ToT +300 µs
Power consumption	13.5 µW/pixel
Chip dissipation	450 mW

Table 2.10: Summary of the Timepix chip

Timepix3 Timepix3 represents an evolution of the Timepix chip, of which it shared the general floorplan and pixel size. The main innovations lie in improvements in the ToT and TOA evaluation, the introduction of zero-suppression and on-chip power pulsing.

In this new chip, developed in a 8-metal 130nm CMOS technology, the pixels can compute the ToT and the TOA at the same time, as they use different counters: the ToT has a dedicated 10-bit counter driven by a40 MHz clock, while the TOA has a 18-bit counter with 4 bits in high timing resolution which use a Voltage-Controlled-Oscillator (VCO) to achieve a640 MHz clock.

The VCO is not dedicated to a single pixel, but is instead shared between 8 pixels, in what has been called a *Superpixel* (shown in Fig. 2.15). In this new hierarchical entity lie also the enhanced readout logic, and an event FIFO. This improvements allow the chip to implement a sparse readout, instead of the full-frame of the previous Timepix chip.

The *Superpixel* FIFO receive the data from one pixel at time, via a selection performed by an internal token ring. Pixel data is shifted from a selected pixel into a deserializer in the SP, and written into a buffer for readout. The buffer has storage capacity

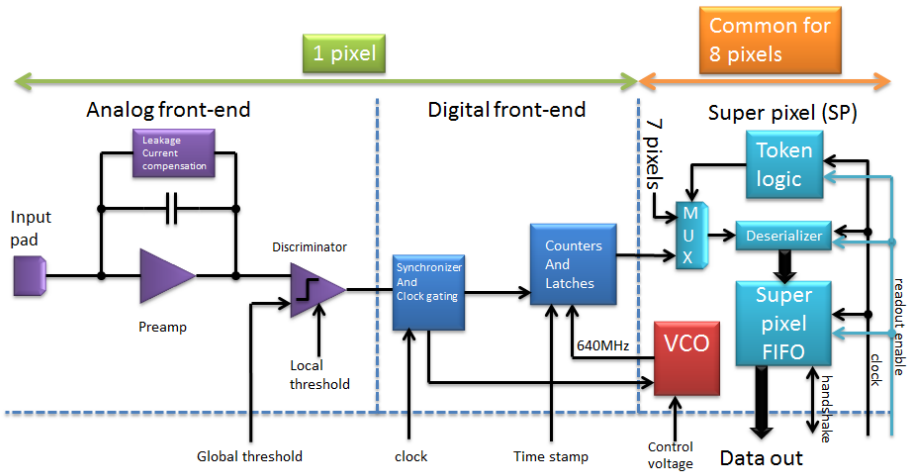

Figure 2.15: The Timepix3 Superpixel. [77]

for two events, which allows continuous acquisition with a pixel deadtime of only475 ns, much lower than the300 µs of Timepix.

This chip represents a great step forward in the functionalities and logic complexity in modern Read Out chips. A summary of its characteristics is found in Tab. 2.11. [77]

Readout chip	Timepix3
Submission	2013
Application	General purpose
Technology	8-metal CMOS 130nm
Readout	8 x SLVS → 5.12 Gbps
Chip size	16 mm × 14 mm
Pixel size	55 µm × 55 µm
Pixel matrix	256 × 256
Radiation hardness	No
TOA resolution	1.625 ns
Design hit rate	80 MHit/s
Pixel deadtime	ToT + 475 ns
Chip dissipation	$< 1\,\mathrm{W\,cm^{-2}}$

Table 2.11: Summary of the Timepix3 chip

2.2.4 Binary Readout

In some cases, the charge information is less relevant than the efficiency, especially at very high hit rates. In order to save on bandwidth and memory requirements, the binary readout chips only save a binary information regarding the hit pixels.

ALICE1LHCB The chip used for the Phase-0 Upgrade of the tracker for the ALICE experiment at CERN is called ALICE1LHCB. It has been developed to meet the experiment's requirements, such as dual trigger scheme support, and fine particle position resolution (∼12 µm).

The chip has been fabricated in a 6-metal 250nm CMOS technology, and contain a 256 × 32 pixel matrix, with each pixel measuring 425 µm × 50 µm.

The chip, whose total area amounts to 14 mm × 15 mm, uses a 1.8 V power supply, and consumes a maximum of 800 mW.

The pixels perform trigger matching by temporarily storing the output of the analog discriminator in a 2-row digital delay unit for the duration of the trigger latency: if the trigger arrives after this latency, the coincidence is registered in a 4-row event FIFO. The event FIFO performs de-randomisation and is used as a shift register to trasmit the hit data down to the Chip Periphery.

The chip can be used in a specially devised *LHCb* mode, in which 8 pixels in a column are configured in a *superpixel*. This superpixel, whose area is much greater (425 µm × 400 µm), combines the output of the single front-ends, and connects together the 16 digital delay units and 16 of the event FIFOs. Such measures allow the chip to also meet the requirements for the LHCb experiment. [3, 49, 27]

Readout chip	ALICE1LHCB
Submission	2003
Application	ALICE and LHCb Phase-0
Technology	6-metal CMOS 250nm
Readout	10 MHz
Chip size	14 mm × 15 mm
Pixel size	425 µm × 50 µm
Pixel matrix	256 × 32
Radiation hardness	500 krad
Power consumption	60 µW/pixel
Chip dissipation	800 mW

Table 2.12: Summary of the ALICE1LHCB chip

Velopix The Velopix chip is related to the Medipix/Timepix families of chips, and it shares the same floorplan and the technological node of Timepix3 (130nm CMOS), but its application is much different from the other chips. It was developed for the LHCb Vertex Detector upgrade, a hybrid pixel detector, which introduces 2 significant additional requirements to the design: it must be able to sustain a hit rate of up to 900 Mhits/s, with more than 16 Gbit/s of output bandwidth, and the radiation levels may reach an integrated 400 Mrad over its lifetime.

The LHCb detector, however, does not need to readout the charges detected by the pixels, and they are only computed inside the pixels to implement an additional, digital threshold. The timing resolution is also lower than the nominal requirement for the Timepix3 ASIC, and is equal to the LHC bunch crossing period (25 ns). In practice, the *Superpixel*, containing the digital logic for 8 pixels, checks every 25 ns if any of the 8 pixels is hit, and, if so, saves the hitmap along with the timestamp, which is propagated from the periphery. The scheme for this new Superpixel is displayed in Fig. 2.16.

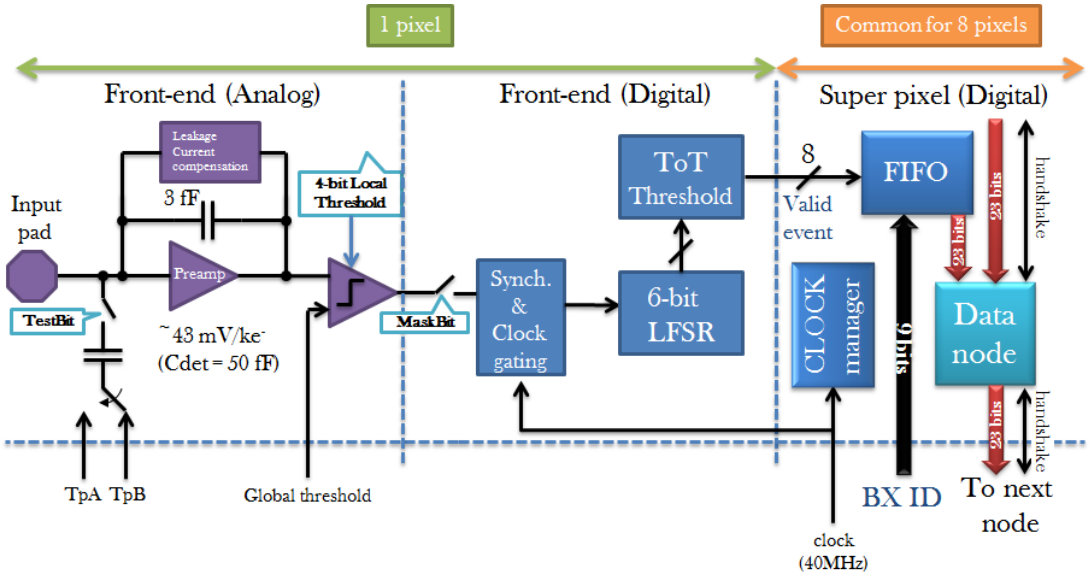

Figure 2.16: The Velopix Superpixel. [75]

The hitmap and timestamps are written to the *Superpixel* FIFO, which is read from a readout controller (a *Node*), in order to orderly ripple it down to the periphery. From there, the packets are immediately driven off-chip (data-driven readout).

A summary of the Velopix characteristics is in Tab. 2.13. [75]

Readout chip	Velopix
Submission	May 2016 [76]
Application	LHCb 2018 VeLo upgrade
Technology	8-metal CMOS 130nm
Readout	4 x SLVS → 20.48 Gbps
Pixel size	55 µm × 55 µm
Pixel matrix	256 × 256
Radiation hardness	40 Mrad
Time resolution	25 ns
Design hit rate	900 MHit/s
Pixel deadtime	ToT + 475 ns
Chip dissipation	<1.5 W cm^{-2}

Table 2.13: Summary of the Velopix chip

2.3 Recent HPDs for HEP experiments

In this section, we'll review 2 of the most recent pixel chips for HEP experiments: the chip developed for the innermost tracker layer of the CMS and ATLAS experiments in Run 3. The study of these chips, in particular, laid the basis for the works of this thesis and the innovations proposed.

2.3.1 PROC600

The Phase 1 upgrade of the CMS detector will modify the arrangement and number of the tracker layers: the upgraded pixel detector will consist of 4 cylindrical layers and 6 forward disks, instead of the current 3 layers and 4 disks. The radius of the innermost barrel layer is reduced to 3 cm, 1.2 cm closer to the beam line with respect to the current one.

For this reason, a new pixel chip has been developed by the PSI institute in order to meet the new requirements. The PSI46dig chip, in fact, will be suitable for the disks and outer layers (from the 2nd), but its maximum supported hit rate is not sufficient for the foreseen 580 MHz cm^{-2} for the innermost one.

The new PROC600 pixel chip is strongly based on the layout of the PSI46dig, but introduces several improvements in the readout and the periphery, while the analog front-end has been left unchanged.

Among the innovations, there is the possibility to continuously drain the columns without needing to reset the buffer, and an improved column drain mechanism that

increases the number of possible pending drains up to 7, and reads out 4 pixels at a time in order to increase the speed.

The PROC600 was finalized in the late 2010s. [95]

Readout chip	PROC600
Submission	2010
Application	CMS Phase 1 Upgrade (Run 3)
Technology	CMOS 250nm
Radiation Hardness	120 Mrad
Chip size	10.5 mm × 8 mm
Pixel size	100 µm × 150 µm
Pixel matrix	80 × 52
Particle rate	600 MHz cm^{-2}
Inefficiency	<2%

Table 2.14: Summary of the PROC600 pixel chip

2.3.2 FE-I4

The FE-I4 chip has been developed by the ATLAS collaboration, and introduces a new organization of the active matrix, grouping pixels together in *Pixel Regions*. The columns are arranged in a mirrored fashion, so that the analog parts lie at the side of a Double Column, while the digital part in the middle.

The FE-I4 chip was designed in 130nm CMOS technology. It is made of 80 × 336 pixels and features a reduced pixel size of 50 µm × 250 µm with respect to the FE-I3 chip. A new digital architecture is introduced, in which hit memories are distributed across the pixel array in the *Pixel Regions*: a group of 4 pixels that shares 5 common latency calculation and triggering units.

The timestamp is saved in the shared logic, but each pixel individually calculates its own 4-bit ToT. The pixels also feature a programmable digital threshold which allows to associate the timestamp of hits with lower TOTs (susceptible to time walk) to that of hits with higher TOTs in the same Pixel Region. Moreover, every event records not only the timestamp and Pixel Region's TOTs, but also a binary hit information regarding the 4 neighbor pixels.

The event information is only sent down to the Chip Periphery if a Level-1 Trigger selects the corresponding timestamp. A maximum of 16 consecutive triggers is supported, as the trigger have an associated 4-bit id.

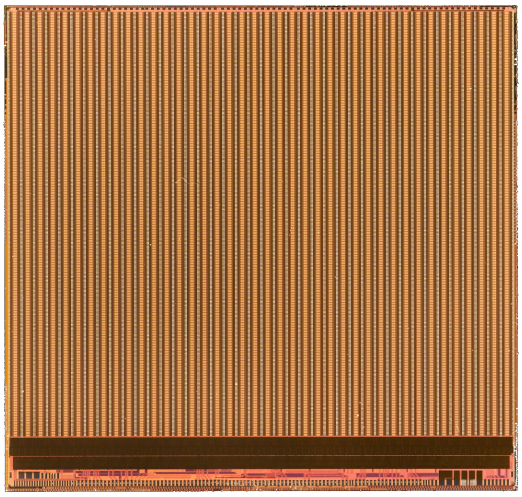

Figure 2.17: The FE-I4 pixel chip. [37]

Once at the periphery, the 20-bit data words (4× 4-bit TOTs and 4-bit neighbor hits) are formatted in order to form 24-bit packets that contain the information regarding 2 vertically adjacent pixels. This scheme has been shown to provide a 24% data bandwidth reduction.

Simulations proved that this architecture, made feasible by the finer technology node, can sustain much higher hit rates that its predecessor, with increased efficiency. [45]

Further development over FE-I4 led to the FE65-P2, the first test chip to introduce the concept of "analog island", a different arrangement of the analog and digital parts in the pixel matrix. While the FE-I4, as many other chips used by both CMS and ATLAS experiments, used a Double Column arrangement, by changing the form factor of the analog front-ends, it is possible to surround the front-ends by digital logic. Special precautions have to be adopted to minimize the chance of cross-talks, such as by using a two deep N-wells to shield the devices, but in this way the Pixel Region logic can extend also horizontally. [13, 61]

These concepts have been later introduced also in the CHIPIX65 and RD53 chips.

2.4 A ReadOut Chip for the Phase-2 Upgrade

None of the proposed pixel chips meet all the requirements for the HL-LHC detectors, although some of them come close to the specifications in specific aspects.

Chip Size The biggest chip for HEP experiments developed, the FE-I4, reaches 16.8 cm × 20 cm, an area very similar to that proposed for the next-generation ROCs.

This was accomplished by taking into account a multitude of factors during design,

Readout chip	FE-I4
Submission	2010
Application	ATLAS Phase 1 Upgrade (Run 3)
Technology	CMOS 130nm
Radiation Hardness	$\geq$ 200 Mrad
Readout	160Mbit/s LVDS
Chip size	20 mm $\times$ 19 mm
Pixel size	50 µm $\times$ 250 µm
Pixel matrix	336 $\times$ 80
Trigger latency	6.4 µs
Hit rate	400 MHz cm^{-2}
Inefficiency	<2%
Power consumption	6.6 µW/pixel

Table 2.15: Summary of the FE-I4 pixel chip

and therefore this chip's architecture can provide a good starting point for future chips, in particular in regards to signals propagation and matrix organization.

Pixel pitch The FE-I4, however, presents much bigger pixels with respect to the 50 µm $\times$ 50 µm specification. Velopix, instead, presents a very similar pitch, and achieved a 256 $\times$ 256 matrix by employing an improved matrix hierarchical structure to perform complex readout. The 50 µm pitch can be achieved with both 50 µm $\times$ 50 µm and 100 µm $\times$ 25 µm form factors. Physics simulations are currently being investigated for the best choice in this regard.

Hit rate capability Although the bunch crossing frequency will remain 40 MHz, the event pile-up will definitely increase, along with the pixel hit rate in the ROC, which rises to 3 GHz cm^{-2} in a 50 µm $\times$ 50 µm sensor matrix. None of the proposed chips is capable of sustaining this rate, as the most performant pixel chip in this regard is the PROC600, which can support up to 600 MHz cm^{-2}.

Trigger support The trigger latencies of the experiments will increase for both the CMS detector (up to 12.5 µs to 20 µs) and ATLAS (a double triggering scheme with 6 µs and 30 µs latencies). The increased trigger latency means that the events buffered in the chips will need more time before they can be readout or rejected. This, in turn, means that the event buffering capabilities would need to be increased, as, the longer

the buffering time, the more hits will need to be buffered. The buffering capabilities will need to be assessed later via analytical methods and simulations, as none of the proposed chips come close to this requirements.

Logic density The most scaled CMOS node used in the reviewed chips is the CMOS 110nm of Monch, which however features analog readout in a triggerless, frame-based readout. Many of the prototypes in CMOS 130nm allowed much greater logic integration in comparison to their predecessors, although none can address the buffering requirements for the HL-LHC chips.

This led the community to focus on smaller technological nodes, such as CMOS 65nm, for the next-generation chips. The CMOS 65nm is a very mature technology, supported by a large number of foundries which provide complete digital libraries and detailed models.

Radiation hardness The expected Total Ionizing Dose that the chips would have to sustain in order to avoid early replacement during operation is of 500 Mrad. By contrast, the radiation hardness of Velopix is limited to 40 Mrad, while the FE-I4 has been shown to be functioning until 200 Mrad.

The effects of ionizing and non-ionizing radiation to silicon devices is more thoroughly treated in App. B, but it has been shown that the 65nm CMOS node may be enough radiation hard to be used in the experiments.

Layout and digital precautions to be used include the avoidance of minimum size transistors, the use of slow corners to model the degradation in the propagation delay of logic gates. In addition, and the implementation of logic redundancy and techniques such as ”trickle” configuration can be used to limit the damage of Single Event Upsets, in particular in configuration registers. [40]

Charge resolution App. C highlights how 4 bits of charge resolution may represent the best tradeoff between buffering capabilities and position resolution achievable, with no significant gain in increasing the number of ToT bits, save for debug purposes and sensor qualification. In particular, the choice of $\mathrm{ToT}_{\mathrm{HALF}} = \mathrm{ToT}_{\mathrm{MIP}}$ in linear encoding, is the best for track separation, positional resolution, and particle identification.

Efficiency Efficiency should be as high as possible, for clusters and single hits. Most of the relevant positional information lies in the edge pixels of the clusters, with low charge, while for track separation and particle mass information, high charges are more relevant. As all this information is of interest, it is important to store as much of the charge information as possible.

At high hit rates, it is possible for a pixel to be hit while it is still processing a previous event. This case is referred to as in-pixel pile-up, and causes the charge of

the second hit to be summed to that of the preceding one, instead of being recognized distinctively.

This form of inefficiency can be tackled by reducing the ToT duration, which is controlled by the Analog Front-End feedback discharge current. The reduction in ToT pulse width must be accompanied by faster ToT clock frequencies, in order to maintain the charge resolution. Frequencies above 40 MHz can therefore have a great impact in reducing the pixels' dead-time and thus increase their hit rate capability.

Summary Tab. 2.16 summarizes the requirements for the Phase-2 Upgrade ROCs. A national and international R&D has been proposed in order to address the lack of a chip which can satisfy these requirements with novel solutions, which are investigated as part of this thesis. A relevant innovation brought about by this research consists in an improvement over current zero-suppression techniques (a review of whose is presented in App. E), which help.

Phase-2 Upgrade Requirement		Closest existing solution	
Radiation Hardness	500 Mrad	FE-I4	$\geq$ 200 Mrad
Pixel size	50 µm × 50 µm	Velopix	55 µm × 55 µm
Pixel matrix area	∼20 cm × 20 cm	FE-I4	16.8 cm × 20 cm
Trigger latency	12.5 µs	FE-I4	6.4 µs
Hit rate	3 GHz cm^{-2}	PROC600	600 MHz cm^{-2}
Inefficiency	<1%		
Power consumption	<10 µW/pixel		

Table 2.16: Summary of the requirements for the Phase 2 Upgrade and the chips whose performance approach them.

Chapter 3

CHIPIX65

In the framework of the efforts for the development of a Pixel Chip capable to sustain the hit rates of the inner barrel of experiments at HL-LHC, the CHIPIX65 was founded to start to develop the key technological base.

Among the improvements needed on the available solutions, there was:

1. Increase of radiation hardness
2. Increase in rate capability
3. Higher granularity
4. Power reduction
5. High trigger rate
6. High output data rate

The new pixel chips should guarantee to work after 10 years of operation with up to $3\,\mathrm{GHz\,cm^{-2}}$ of input rate and 1 Grad ionizing dose. The requirement on the trigger rate and the output data rate are interdependent, as the latter is proportional to the first. The analog performance should also be very low noise, since all possible sensors types considered for HL-LHC have to feature high radiation hardness and are characterized by small signal, running with a low threshold in order to be highly efficient.

Following a workshop held at CERN in November 2012 to collect the experience from experts in the field from CMS/ATLAS and other pixel projects/VLSI activities, the CMOS 65nm technology was chosen as the most promising for a new generation of pixel chips. This led to the constitution in 2013 of the RD53 Collaboration with seventeen funding institutes, including INFN Bari, Milano, Padova, Pavia, Perugia, Pisa and Torino.

The CHIPIX65 project was approved by INFN National Scientific Committee 5 as Call project in October 2013, and counts about 35 members experts in the field, of which 20 are actual IC designers, constituting a substantial fraction of INFN expertise on microelectronics. [26]

This chapter provides a description of the chip and its components, with a focus on the digital architecture and integration challenges, which have been overcome as part of this work of thesis.

Target Requirements The goal of the CHIPIX65 Collaboration was the development of a Pixel Chip which could meet the following requirements:

Parameter	Value
Pixel size	50 µm × 50 µm
Pixel matrix	64 µm × 64 µm
Particle Rate	500 to 750 MHz cm^{-2}
Trigger Latency	12.5 - 20 µs
Trigger Frequency	750 MHz - 1 GHz
SEU Tolerance	Control Logic
Power Consumption	10 µW / pixel
Efficiency	99%

A common assumption is that, with the current sensor technology, a particle would leave, on average, a 4-pixel cluster in the detector. Thus, the requirement for the pixel ROCs became that they would need to sustain a pixel hit rate of 2 GHz cm^{-2} (for a 500 MHz cm^{-2} particle rate), or 3 GHz cm^{-2} (for a 750 kHz cm^{-2} particle rate)

Along with these requirements, a set of testing features had to be present, such as a Triggerless mode, in which every hit recorded by the chip would be downloaded immediately after buffering; a digital injection mode, used to test the digital logic; and many IP blocks used to directly characterize the performance of the chip.

3.1 IP Blocks

The collaboration experts developed several IP Blocks needed for a possible Pixel Chip in CMOS 65nm. Among these, some key blocks are a Bandgap reference, a DAC, a Serializer and Deserializer, a DICE RAM, and an ADC.

3.1.1 Bandgap

A key component that ensures correct chip operation independent on Process, supply Voltage and Temperature (PVT) variations, is the BandGap Reference (BGR). The BGR provides a stable DC voltage, which for this application must also be able to be non sensitive to radiation dose.

Bandgap references take their name from the way they generate the output signal. By using two p-n junctions operated with different currents, it is possible to extract a current which is Proportional To Absolute Temperature (PTAT). Conversely, the voltage across a diode operated at constant current is Complementary To Absolute Temperature (CTAT).

It is thus possible to balance these effects out by using the CTAT voltage on one of the PTAT diodes, or another one driven by the PTAT current, in order to produce a voltage which is independent on the temperature.

Such structures, however, show a clear sensitivity to TID and TDD. To overcome these problems, new solutions solely based on MOSFETs, and thus avoiding the parasitic PNP bipolar structures available with CMOS technologies. Three different prototypes were submitted, in order to assess the one with the best characteristics: one is based on a classical BJT design, one is instead based on a P-N diode, and another one on N-MOSFETs biased in weak inversion region.

The prototypes were then irradiated in order to study the dependency of the output voltage with respect to the Total Ionization Dose (TID). Results show a strong variation of the output voltage in the BGRs based on bipolar and diode designs already at doses of hundreds of krad. This behaviour is due to an increase of the leakage current probably caused by the charges trapped in the field oxide above the p-n junction of the devices. A limited recovery can be observed after the annealing process. A modest variation, about 1.1%, is instead detectable in bandgap circuits based on N-MOSFET, which was thus identified as the best candidate for the chip. [100]

3.1.2 DAC

Digital to Analog Converters (DACs) are extensively used in Pixel Chips to generate the correct current or voltage references or biases needed to operate other analog blocks.

Current steering DACs are based on an array of matched current sources that are switched to the output, acting as a summing node. Two main architectures are possible: the binary weighted architecture and the unary decoded one. A combination of them is also feasible, in what is called a segmented architecture.

In the binary weighted scheme, every switch connects to the output a current source that is twice as large as the next least significant bit, thus the digital input word directly controls the switches. This architecture is relatively simple since it can be driven by digital words directly and thus no decoding logic is needed. It also has the advantage of being realizable in a small silicon area. The main drawback is its sensitivity to device mismatches, which implies a large Differential Non Linearity (DNL) error.

In the unary decoded architecture, the weighted current sources give way to identical unit current sources, where each one with its own switch is addressed separately using a thermometer decoder. This architectures guarantees an improvement of the DNL, as process variations are limited on identical devices. The main drawback is its

higher complexity for the additional decoder and the area and power increase for the presence of a switch for every unary current source.

Given that for this pixel chip it was desirable to have as much flexibility as possible, a 10-bit DAC was implemented, by using a segmented architecture: the two least significant bits are implemented with binary weighted current sources, while the eight most significant bits are implemented with unary decoded current cells.

This solution represents a trade-off between power consumption, area and DNL (Differential Non Linearity) optimisation. The design was also optimized for a Least Significant Bit (LSB) of 100 nA, but it is also possible to use different reference currents with only a marginal degradation of performance in terms of DNL and INL. The radiation hardness of this structure was improved by the avoidance of minimum size transistors.

10 prototypes (with a total of 20 DACs) were tested, by using an Ethernet controller and a stable reference voltage. By using this same reference current for all the DACs, the I_{LSB} was first evaluated and compared with the nominal one, which resulted in a variation of 0.74 nA (100.74 nA over 100.00 nA), with a standard variation of 1.31 nA. The Differential Non Linearity and Integral non Linearity were also measured and compared with the ones obtained via Monte-Carlo simulations (500 points) performed during the design. In table 3 the results are summarized while, in figure 5, the INL curves for MC simulations and test results are displayed and show the good agreement between simulations and tests. [87]

3.1.3 Serializer

Radiation tolerant 2GBps Serializer (SER) and Deserializer (DES) devices have also been developed for a reliable communication in a radiation harsh environment. The IP Blocks have been developed with a 20-bit work support. A Data-Strobe signal tells the logic when the Serializer is able to process a new input data word, while a Data-Valid signal tells the Deserializer when the readout stream can be sampled. These signals are structured so as they can be used as read and write clocks for input and output buffer FIFOs. The radiation hardness design is based on triple redundancy.

The first test chip integrates a radiation-tolerant testbench and two SER-DES pairs. The testbench is needed because of the limitations in pad availability that did not allow the direct access to the parallel ports. The first pair's clock is connected to a Current Model Logic (CML) receiver pad developed at CERN capable of sustaining a rate of 2 GHz, the one of the second chip via a standard CMOS pad, which can only support slower clocks. The test bench generates data packets for the SER devices, made of a Header used for channel synchronisation and by a programmable number of data words generated in a pseudo random generator.

Tests can therefore be performed by checking channel synchronization and comparing expected data with the received one. Control and status registers and error counters are accessible through a serial interface.

First prototypes of the SER and DES devices have been produced in early 2015.

3.1.4 ADC

The chip also embeds an accurate ADC, needed for monitoring the level of slow input signals. The designed ADC is based on the integrating dual-slope architecture, capable of supporting up to 16 inputs (via multiplexing). The ADC supports digitization with 12 bit resolution over an input range of 1 V, with a conversion rate is 5 kSample/s.

First, integration of the input signal if performed by converting the input voltage into a current, by means of a linear transconductor. This current is used to charge a 70 pF Metal Oxide Metal (MOM) integration capacitance C_{int} for 2^12 clock cycles.

The output code is obtained by counting the clock cycles needed to discharge the capacitance at a constant current, obtained using the full scale voltage as input of the same linear transconductor. [64]

3.1.5 DICE RAM

As the buffer requirements for the chip were high, the design of a radiation-hard RAM was also included in the project. The chosen design is a DICE memory, which employs circuit-level design techniques to prevent SEU.

A DICE memory contains duplicated data, such that the bit state is encoded in 2 homologous nodes. If a single particle is affecting the voltage of only one of the homologous nodes in a DICE, then the cell will not exhibit a SEU. The SRAM cell has been extensively simulated in worst cases, including RC parasitics, and with fault injection to simulate single events: the results demonstrate a very good tolerance to SEE.

A higher robustness can be achieved by physically separating homologous nodes of the single DICE, to a distance that prevents a single particle to affect both of them: for example, by interleaving two (or more) DICE elements.

The first design has been submitted in October 2014 and thereafter promptly tested.

3.2 The Analog Front-Ends

The CHIPIX65 Collaboration provided for the design of 2 different Analog Front-Ends: a synchronous one, developed by INFN Torino, and an asynchronous one, developed by INFN Pavia/Bergamo. The teams' efforts were put in the design of a full Front-End chain (preamplifier, discriminator, signal processing), with very fast peaking time, low noise solutions, and different Time-over-Threshold (ToT) measurement methods.

Both architectures require the conversion of the signal from the sensor into a voltage by means of a Charge Sensitive preAmplifier (CSA). Continuous charge reset in the

preamplifier is achieved through a Krummenacher[56] stage: this was specifically chosen for its capability to compensate for the expected radiation-induced increase up to 50 nA in the sensor leakage current during the experiment.

3.2.1 The synchronous FE

The Synchronous Front-End features a single stage Charge Sensitive Amplifier (CSA) with a Krummenacher feedback AC coupled to a synchronous discriminator composed of a Differential Amplifier (DA) and a positive feedback latch. A scheme of this Front-End is shown in Fig. 3.1.

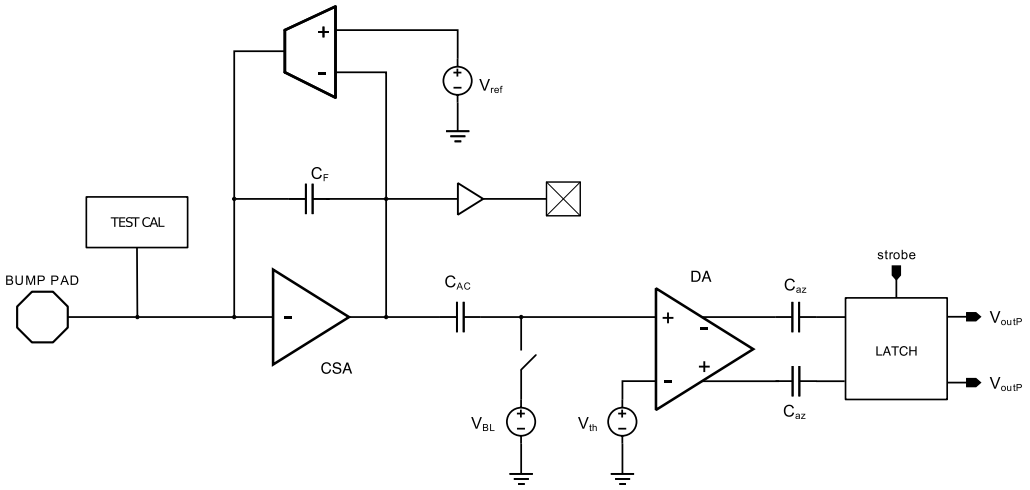

Figure 3.1: Schematic of the Synchronous Front-End. [65]

The Charge Sensitive Amplifier (CSA) implemented as a single-ended, high open-loop gain inverting amplifier with capacitive feedback. It contains a telescopic cascode stage with current splitting to minimize noise contributions, and a source follower that improves the driving strength. The CSA input node is also connected to a calibration circuit, which can be used to inject test charges of known values. Furthermore, test capacitors have been added to mimic different values of pixel sensor capacitance.

The Krummenacher feedback is designed to provide both the sensor leakage current compensation and the constant current discharge of the feedback capacitor: the larger the current the faster the preamplifier signal returns to the baseline. As a reference, a 10 nA current results in a 400 ns-long signal for an input charge of 10 keV, which is reduced to around 100 ns for a 40 nA current. Two capacitors, equal to 2.5 and 4 fF respectively, have been included in order to implement different gain values in the Krummenacher feedback. They can also be selected together, in a way to provide a 6.5 fF equivalent capacitor.

In addition, a calibration circuit featuring an injection capacitance of around 8 fF has been designed to provide an input charge in the desired interval, which is 1 keV to 30 keV.

In Deep Sub-Micron technologies the mismatch effects cannot be considered as negligible. Because of this, the output baseline of the CSA can be subject to quite large fluctuations (an effect quantifiable in tens of mV) between different channels. In order to filter out such behavior, the Differential Amplifier (DA) is AC coupled with the CSA. No other signal shaper is present, as the signal is already triangularly shaped.

The DA compares the input signal with the threshold voltage, while also providing a further small gain. Transistor mismatch results in an offset of the DA output voltage between pixels. These further mismatches are compensated using internal capacitors in a procedure known as "Auto-zeroing", which requires a compensation phase of around 100 ns every 100 µs. This mechanism implements a local threshold trimming without using a dedicated DAC. The advantage of this approach lies in the ease of implementation, as the front-ends do not have to be tuned individually, but the procedure can be applied to the whole matrix at once.

The compensated signal from the DA finally arrives at the discrete-time voltage comparator. This has been implemented as a positive feedback latch stage, which performs the comparison and generates the discriminator output. This stage has been designed to minimize mismatch effects causing a dynamic offset resulting in an additional threshold dispersion.

The latch can be turned into a local oscillator up to 800 MHz using an asynchronous logic feedback loop. The latter includes a current-starved delay line which is used to tune the oscillation frequency by changing the value of a dedicated bias current.

An external clock signal is required to periodically enable/reset positive feedback in the latch, thus introducing synchronous Front-End operations. Depending on the comparator decision, two differential outputs settle to rail-to-rail complementary logic levels at each clock cycle and the hit generation becomes synchronized with the external clock. This technique allows to implement fast signal digitization using the time-over-threshold mode. [65]

3.2.2 The asynchronous FE

The asynchronous front-end design is based on a linear comparison of the shaped input signal with the threshold. The signal from the sensor is converted to a voltage by means of a charge sensitive amplifier, continuously reset by means of a Krummenacher stage, that is capable of compensating for the detector leakage current, expected to increase significantly during the experiment. The scheme for this Front-End is shown in Fig. 3.2.

The signal at the preamplifier output is fed to a threshold discriminator, turning the signal amplitude into a time interval or ToT, time over threshold. The threshold discriminator is based on a low power transimpedance amplifier for fast switching operation.

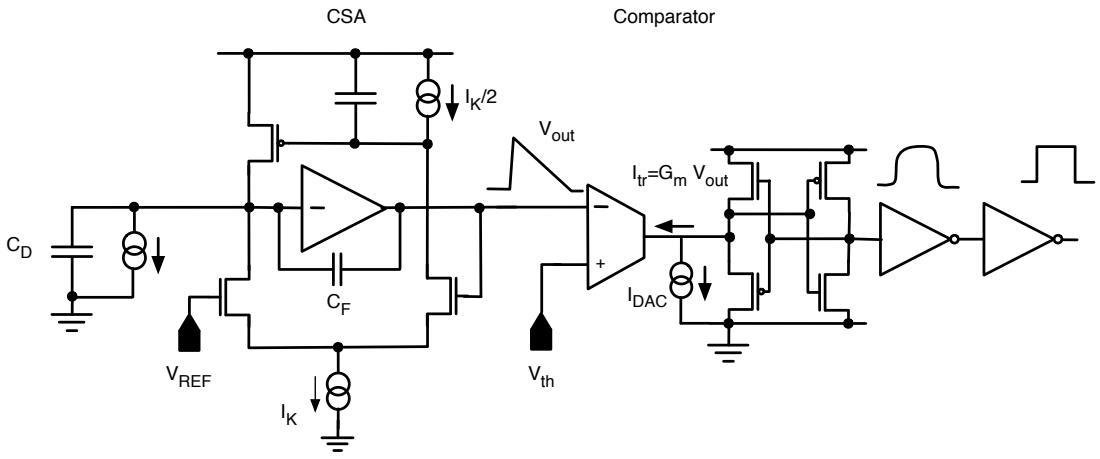

Figure 3.2: Schematic of the Asynchronous Front-End. [83]

[30]

Given the triangular shape of the preamplifier response, featuring a very fast leading edge and a constant slope return to baseline, a linear relationship between amplitude (or input charge) and ToT is expected.

The threshold discriminator output is used as a gate signal (through the AND gate) for the ToT clock, which is fed to the 5 bit ToT counter for time to digital conversion. The threshold dispersion is addressed by means of a local threshold compensation circuit baed on a 4-bit current steering DAC. The power consumption per channel is slightly smaller than 5 µW.

The front-end circuit is designed to comply with a maximum input signal of 30 keV and features an output dynamic range around 450 mV, a charge sensitivity of about 90 mV fC^{-1} and an equivalent noise charge (ENC) of 114 electrons rms for a sensor capacitance CD=100 fF.

A time walk not exceeding 25 ns is achieved in circuit simulations with a threshold of 700 electrons and signals 1000 electron in amplitude or larger. With a current I_K = 12.5 nA in the Krummenacher network and a preamplifier feedback capacitance of about 10 fF, the maximum expected ToT is 400 ns. Therefore, an 80 MHz ToT clock is needed to take advantage of the 5-bit counter full scale. [84, 83]

3.3 Architectural studies

The first studies for the overall digital architecture of the CHIPIX65 chip can be found in [71]. In this small section I will recap the status of those studies, the design choices made, and how the development steered from some of them.

3.3.1 Pixel Matrix organization

The first studies for the CHIPIX65 architecture focused on the feasibility of both single pixel and pixel region architectures. The single pixel architecture used physical simulation files provided by the Simulation Group at INFN Torino, in order to feed the chip with a realistic input pattern. The efficiency was measured in terms of hits lost due to buffer overflow.

Although it was possible to obtain the required 99% efficiency, the area needed for such implementation was too big to fit in the specified 50 µm × 50 µm. The research moved to the study of feasibility of Pixel Region architectures, already implemented in some precursor chips, like the FE-I4.

The first proposed Pixel Region implementations stored all the hits in a 4 × 4 pixel area in a common buffer, which could sustain a 2 GHz cm^{-2} hit rate, with an area occupancy in the Pixel Region of 74%. In particular, a 4 × 4 Pixel Region would need 16 buffer rows in order to display event losses <0.1%, as shown in Fig. 3.3.

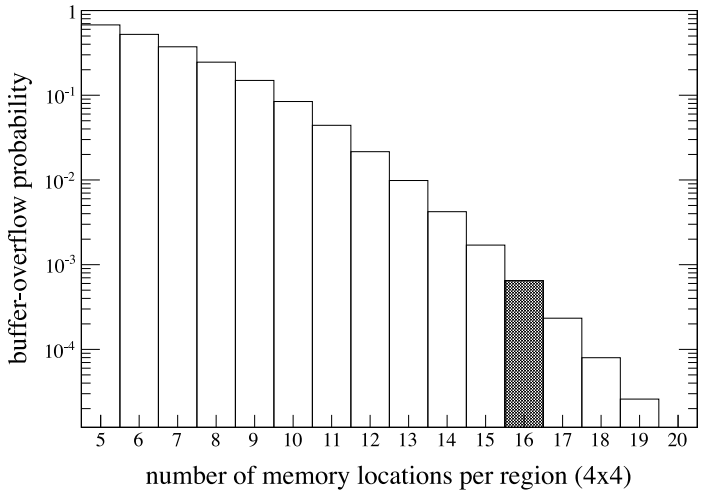

Figure 3.3: CHIPIX Pixel Region event buffer depth versus efficiency

The design, however, had to be updated in light of the increased hit rate specification (which became 3 GHz cm^{-2}). Moreover, the advent of the verification environment, later described, provided the hit patterns which allowed further optimization of the digital architecture.

3.3.2 ToT storage

Originally, the CHIPIX65 architecture used a straightforward way to encode the positional information of a hit in the Pixel Region: it assigned a number of ToT *slots* in the buffer rows equal to the number of pixels the Pixel Region is made up of. In this way,

there's a 1-to-1 mapping between the pixels and the corresponding ToT slot in the buffer. Such a mapping scheme was already used in the FE-I4 Pixel Region implementation, in that case with a 2×2 pixel arrangement.

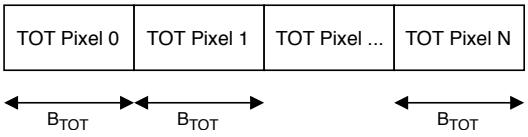

Figure 3.4: Data word for the Distributed ToT mapping scheme

It follows that for a Pixel Region composed of N pixels, the number of bits needed to encode the overall positional and charge information is equal to $N \times B_{\mathrm{ToT}}$ with B_{ToT} the number of ToT bits.

Simulations, however, showed how, of an entire Pixel Region event, only a fraction of the pixels were hit. In particular, according to the results presented in Tab. 3.1, it was found out that Pixel Region events containing more than 8 pixels hit were negligible.

No. of pixels	Probability
1	38%
2	26%
3	16%
4	11%
5	5%
6	2%
7	0.9%
8	0.1%
9	0%

Table 3.1: Pixel Region occupancy simulation

It was proposed that a way of zero-suppressing the events in the matrix could lower the area occupancy dramatically.

In particular, the original contribution of this thesis has been focused on the optimization of a hitmap-based approach to this zero-suppression: by attaching a N-bit sized hitmap, the number of stored TOTs can be diminished to $M < N$, and storing only those TOTs that are different than 0 (as a form of zero-suppression).

An immediate downside of this approach is that, should there be more than M pixels hit in a bunch crossing, a part of the charge information in the region would be lost.

This mapping is schematized in Fig 3.5.

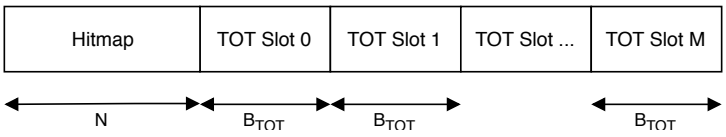

Figure 3.5: Data word for the Hitmap ToT mapping scheme

It is clear that the hitmap-based approach is convenient if:

$$N + M * B_{\mathrm{ToT}} < N * B_{\mathrm{ToT}} \rightarrow M < N * \frac{B_{\mathrm{ToT}} - 1}{B_{\mathrm{ToT}}} \rightarrow B_{\mathrm{ToT}} > \frac{N}{N - M} \tag{3.1}$$

The gain in area is higher if the number of bits per ToT (B_{ToT}) is higher, or if fewer ToT slots are used. In the CHIPIX65 implementation, the number of ToT bits is 5, while the number of ToT slots which would assure a charge information storage efficiency higher than 99% is 6.

This means that the proposed hitmap scheme allows a save in the number of memory elements equal to 34 bits over 80 bits (42%), which far exceeds the area overhead due to the ToT compressor, which is later described in detail.

3.3.3 Readout scheme

As part of the first studies on the CHIPIX65 architecture, a particular triggering and readout scheme was selected, based on its implementation simplicity and high efficiency. A block diagram showing the main signals is shown in Fig. 3.6.

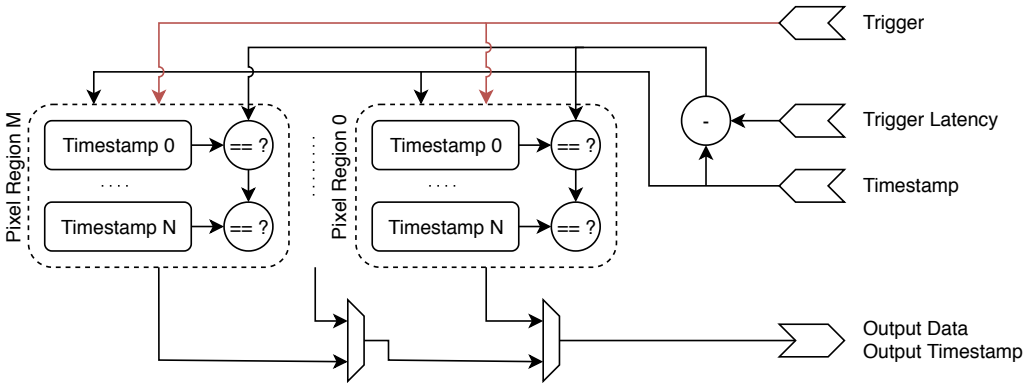

Figure 3.6: Buffer implementation for a Free fall readout.

The selected readout mode allows the Pixel Regions to send out the triggered information whenever ready. In this ”free fall” readout, however, it is not straightforward to

distinguish between information coming from adjacent, or however very close, triggers. This is due to the fact that the only arbitration in the shared bus depends on the output status of the preceding Pixel Regions (stop if any higher Pixel Regions is preempting the channel, otherwise send the triggered info).

As there is no guarantee that information coming from different triggers is not mixed up during readout, it becomes necessary to send out also the timestamp information. This is, of course, not problematic for the triggerless mode.

A simple workaround for this problem involves a desynchronization between the trigger signals that reach the chip and the actual ones that are sent to the Core Columns. If there is control over the trigger timestamp, one can buffer the incoming triggers in a FIFO, and serve one of them at a time. The updated block diagram can be seen in Fig. 3.7.

By buffering the trigger timestamps, the sequence would involve:

1. Polling the trigger fifo
2. Send the trigger timestamp along with the trigger signal to the matrix
3. Receive the charge information of the triggered event
4. When done receiving data, pull another event from the fifo

The drawback of this approach is that, if the trigger timestamp bus is used in this way, it cannot be used for both trigger matching and event clear. It is important, in fact, to remove events which are not triggered, as if they persist in the memory (becoming *stale* events), they may be erroneously triggered later. To avoid the aliasing problem, the event clear check is performed by comparing the event timestamp with the peripheral timestamp.

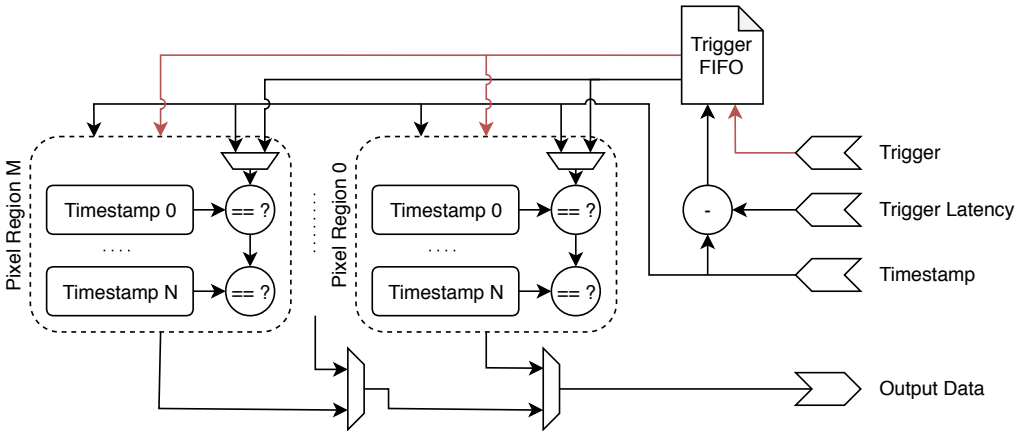

Figure 3.7: Buffer implementatino for a Detached Free fall readout.

This may cause events to remain in the Pixel Region buffer for longer than needed[1], which increases the inefficiency.

It should also be noted that, if the timestamp comparator is not duplicated, the trigger matching operation would inhibit validation of the events. This is another potential source of inefficiency, as it allows some entries to remain in the buffer when they should be cleared out, and thus manifest themselves, if erroneously triggered later on, as "ghost" hits. As simulations showed this ghost hit rate to be negligible, this implementation was retained for the chip.

3.4 Digital architecture

The digital architecture of the CHIPIX65 chip stemmed from the architectural studies and first implementation attempts, but was completed and integrated as part of this work. By trying to take advantage of the increased computing power that can be placed inside the matrix, design efforts have been put towards the development of a smart digital architecture for the pixel matrix. [73]

A significant part of the efforts were put in the customization of the Verification Environment in order to accurately measure the efficiency of the chip and test all the features supported.

3.4.1 Verification Environment

Developed in the context of RD53, a powerful UVM verification tool was available in its early stages and has been tested with CHIPIX65: VEPIX53. The verification environment provided means for random hits generation, random trigger generation, and a reference model to compare the DUT against.

In the early development phases, the VEPIX53 environment generated hits based on a number of parameters. These include the particle type (charged particle, jets, loopers, background hits), sensor's pixel pitch and thickness, particle hit rate and deposited charge range, cross-talk probability, and so on. [60]

The hit generation runs in parallel with the trigger generation, which can in turn be customized on the trigger latency and the average trigger rate.

The reference model describes an ideal pixel chip at high level modeling, and thus is capable of performing the buffering and trigger matching operations with 100% efficiency. The model has been soon refined in order to model the hit losses due to the Front-Ends deadtime, and with it a categorization of hit losses: it was possible to distinguish hits lost due to the pixel deadtime, and those lost due to other loss sources.

[1] In particular, this happens if a full Gray encoding is used and if $TL < 2^T$, with TL being the Trigger latency in clock cycles, and T the number of bits used to encode the timestamp

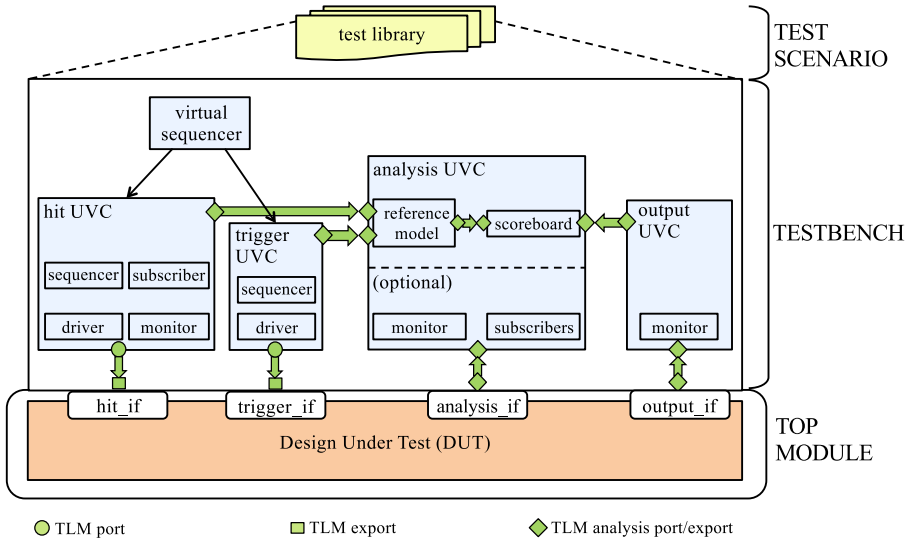

Figure 3.8: Block diagram of the VEPIX53 verification environment. [60]

Further improvements in the reference model were not available during the development of CHIPIX65, and the design has been iteratively improved by testing 2 DUTs: the Pixel Region itself, for the Pixel Region optimization, and then the whole active matrix in order to characterize and optimize the peripheral data flow.

3.4.2 Pixel Region

The Pixel Region architecture was separated in 2 communicating elements: the Pixel Logic, and the Shared Logic. The Pixel Logic contains all the structures needed to communicate with the Analog Front-End, including the ToT generation logic, and the configuration latches and driving logic. The shared logic, instead, is made up of the central Shared Buffer, the trigger matching comparators, and the output stage. A scheme is shown in Fig. 3.9.

As the Front-Ends are different, the Pixel Regions themselves come in 2 different *flavors*: one optimized for the Synchronous FE, and another optimized for the Asynchronous FE.

3.4.2.1 Pixel Logic

As the chip embeds 2 different Front-Ends, the Pixel Logic is slightly different in the 2 cases. The 2 Pixel Logic variants, in fact, mostly differ in the configuration bits and the ToT generation procedure, keeping intact the interface with the Shared Logic.

The charge information is, in fact, computed in the Pixel Logic using 5-bit ripple counters. The ToT Clock typically is the 40 MHz clock distributed to the matrix. The only exception is in the Synchronous FE, when the Fast mode is enabled.

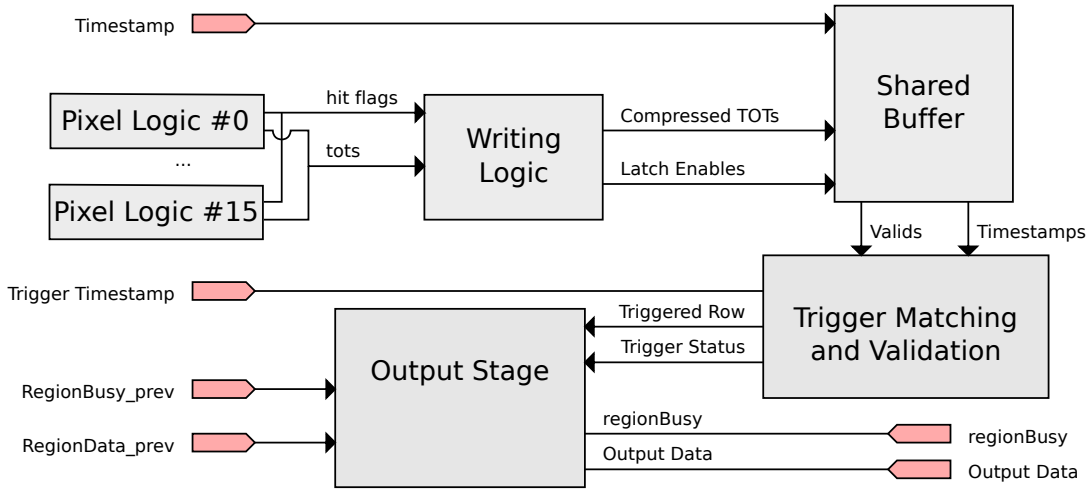

Figure 3.9: Block Diagram of the CHIPIX65 Pixel Region

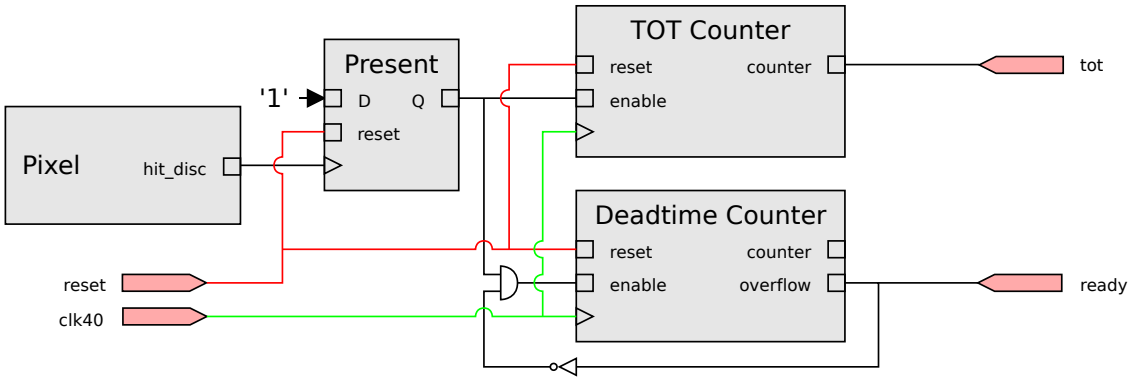

Figure 3.10: Block Diagram of the CHIPIX65 Pixel Logic. This module is replicated for every pixel in the Pixel Region. Not shown, the configuration logic.

This clock is gated in the pixels depending on the value of the discriminator, and then propagated to the ToT counter. The ToT counter is not the only one in the pixels: they also feature another counter, which is used for writing synchronization, called *Deadtime counter*.

Another key component in the Pixel Logic is the PCR: the Pixel Configuration Register and its logic. The PCR contain the bits needed for the operation of the Front-Ends, and other generic configuration, as the masking bit which can be used to ”silence” noisy hits.

The PCR bits are triplicated using Triple Modular Redundancy to reduce the effects of Single Event Upsets: for a bit to change its value, at least 2 SEUs must happen on the three-bit structure. The PCR configuration is controlled by the peripheral logic, which sends the address of the pixel to be configured, the data and the enable signals for the

configuration.

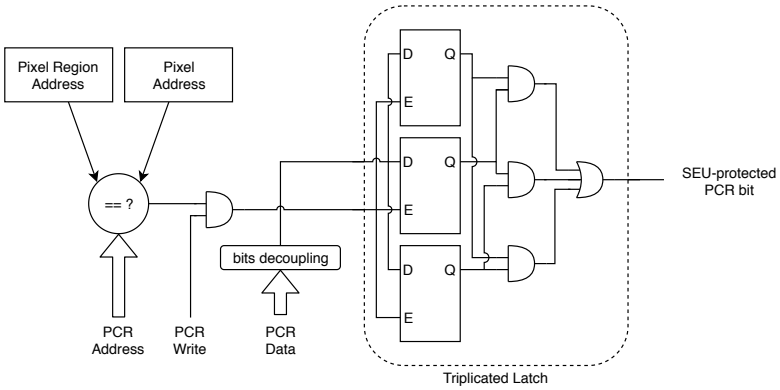

Figure 3.11: Block Diagram of the CHIPIX65 Pixel Configuration. The PCR address and data buses, and the write enable signal are propagated from the periphery. Shown, a triplicated latch in a PCR register.

Synchronous FE The Synchronous FE generates its own ToT clock, which can also operate at high speed, but, because of the latch structure, the counter should only sensitive to the rising edge of the clock. A special logic is needed to convert the rail-to-rail output of the Front-End latch into a proper signal.

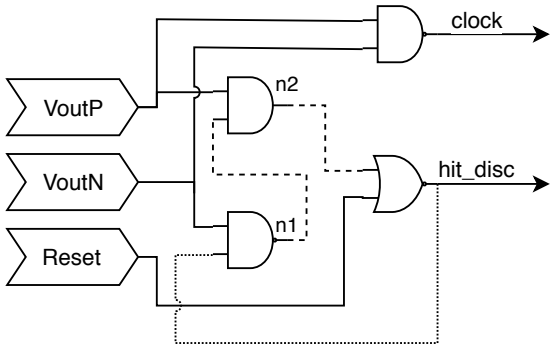

Figure 3.12: Schematic for the Synchronous FE output interpretation. This logic is embedded in the Pixel Logic of the Synchronous FE flavor.

Fig. 3.12 shows how the output signal and the generated clock are generated from the Synchronous FE outputs (VoutP and VoutN). The generated clock can be fed back to the Front-End as the driving clock in order to perform fast operations. In this case, the clock goes through a programmable delay line, whose delay is controllable via a bias current. The clock feedback is necessary to operate the Front-End in fast mode, in order

to perform the threshold comparison at the same rate as the ToT count. When not used in fast mode, the input clock of the Front-End is tied to the Pixel Region clock, in order to check the signal input sychronously with respect to the bunch crossings.

The front-end necessitates of the following configuration bits:

- Mask → Digitally mask the pixel from being read out
- Cal En → Enable test charge injection on this pixel
- Fast En → Enable Fast ToT counting on this pixel
- Sel C2F → Enable the 2.5 fF capacitance in the Front-End
- Sel C4F → Enable the 4 fF capacitance in the Front-End

Asynchronous FE The Asynchronous FE, instead, has to rely on the Pixel Region's clock, which is considerably lower in speed, but can use both edges.

The front-end necessitates of the following configuration bits:

- Mask → Digitally mask the pixel from being read out
- Cal En → Enable test charge injection on this pixel
- GAIN_SEL[1:0] → Gain selection for this pixel
- TDAC[3:0] → Threshold trimming DAC bits

The Asynchronous FE provides a binary discriminator output, and thus requires no signal processing in order to be used by the digital logic.

Operation The functionality of the Pixel Logic is highlighted in Fig. 3.13, a SPICE simulation where the key signals have been selected. Fig. 3.13 details the operations the Pixel Logic of the Synchronous Front-End performs when a hit is registered. Similar operations are performed by the Asynchronous Front-End Pixel Logic.

A hit is injected at the mark number 1, when the signal of the Front-End starts to rise. The VoutP and VoutN signals coming from the Synchronous Front-End are used to determine when the hit is present, and in this case the *hitPresent* signal rises. This enables the clock in the Pixel Logic at mark 2, along with the ToT counter operation. When the Front-End signal goes below the threshold at mark 3, the ToT count is stopped.

The deadtime counter reached the fixed deadtime at mark 4, thus triggering the compression of the TOTs until mark 5, when it resets the Pixel Logic. The pixel is ready to accept a new hit at mark 6.

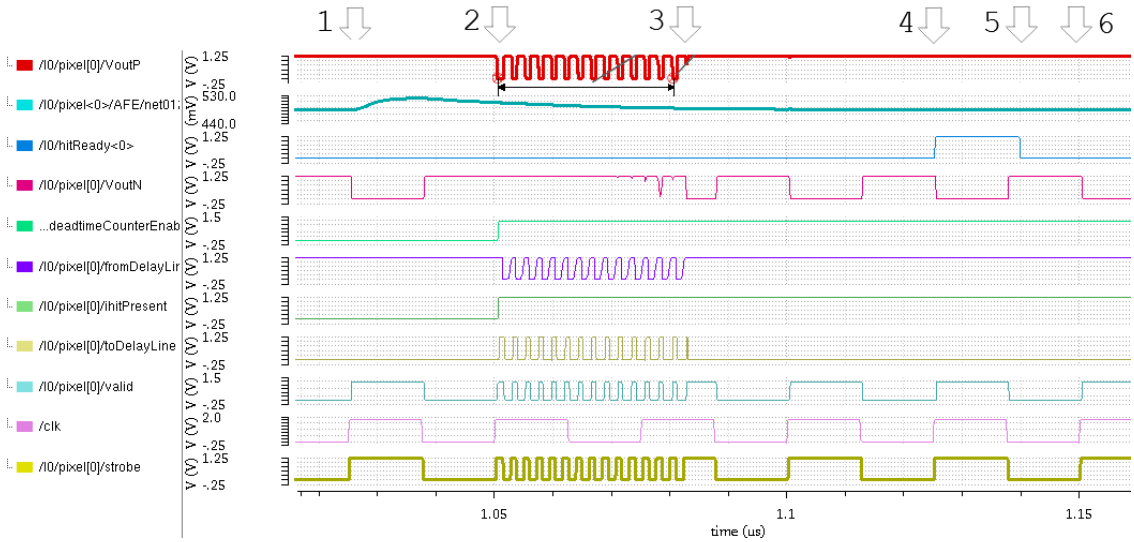

Figure 3.13: Spice-level simulation of the Pixel Logic of the CHIPIX65 Synchrous FE

3.4.2.2 Writing Logic

This chip is the first to study the implementation of an innovative solution to save resources in the Pixel Matrix by performing zero-suppression in the Pixel Regions themselves.

The simulations available during the design operation showed that, in a 4×4 Pixel Region, a Hitmap buffering scheme would be optimal for the charge storage: the centralized buffer would write a 16-bit hitmap for the event followed by a lower number of TOTs.

Considerable area and complexity is introduced in the Writing Logic by the *ToT Compressor*, which selects a subset of the total number of pixels whose TOTs will be saved.

The same internally generated hits in the verification environment helped define the number of TOTs to be saved per event. The occupancy measurements in the 4×4 Pixel Regions showed that more than 99% of the time, less than 7 pixels are hit per event. The distribution is shown in Figure 3.14.

The association between the ToT from one of the 16 pixels to one of the 6 ToT slots can be thought (and implemented) in several ways. It was necessary to choose and design a *compression* architecture with the least area footprint, and an acceptable timing.

Direct-mapping Zero Suppressor A straightforward compression algorithm would implement a direct multiple association between the pixels and the slots. One can picture it as a sequential check of every pixel status (hit, not hit), and, if hit, its assignment

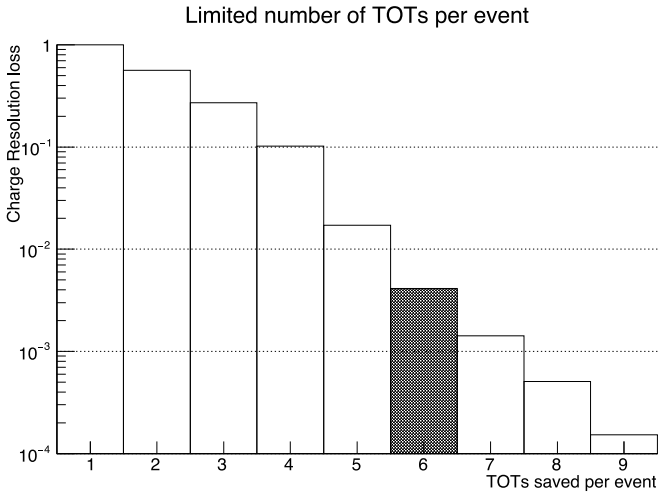

Figure 3.14: CHIPIX Pixel Region ToT slots number versus efficiency

to the first available slot. It would work by iteratively check if the i-th pixel is hit, associate it to the j-th slot, where j is the slot index, and then increment j itself.

This combinatorial loop can be written in the following pseudo-code:

1. Initialize j=0, i=0
2. For each pixel i
   (a) Check if pixel i is hit. If not, **continue** the loop.
   (b) Multiplex the ToT of pixel i to the ToT slot j.
   (c) Increment the counter j.

Any synchronous version of this algorithm was discarded a-priori, in order to avoid the overhead due to the Flip Flops and the logic redundancy. The efforts were instead focused on an asynchronous implementation, which would supposedly have a lighter area footprint, although more problematic with respect to the timing requirements.

The key part of this compression scheme lies in the *Assignment Table*, which keeps track of the pixels which have been assigned to the slots. This table allows to check whether there is a free slot, according to a priority queue. The *Assignment Table* for this architecture is schematized in Fig. 3.15.

By trying to assign the pixels to the first slot available, it is clear that certain assignments are not useful and can thus be discarded. These are represented by the assignment of pixels with index i to slots with index greater than i, which correspond to the colored boxes in Fig. 3.15.

The scheme in Fig. 3.15 does not show an important mechanism key for the functionality of the scheme: when pixel i checks the first available slot, it must do so by checking only if the pixels whose index is lower than i have been assigned to the slots. In other words, the *full* flag for the slot must depend on which pixel is checking it:

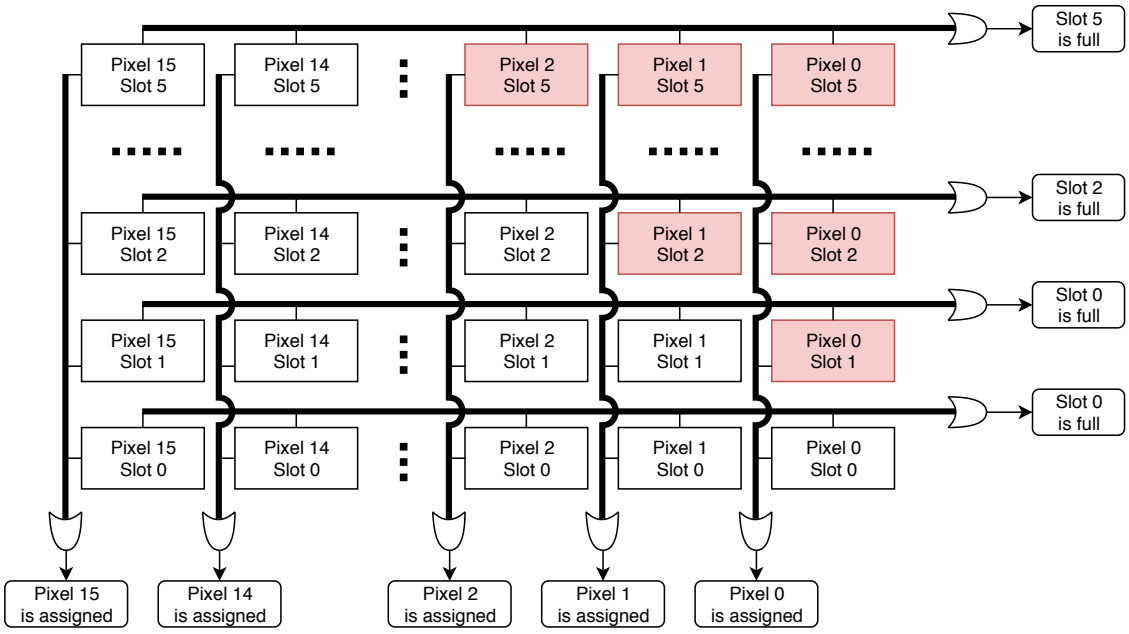

Figure 3.15: Block Diagram of the Assignment Table of the CHIPIX65 ToT Zero-suppressor. The colored blocks indicate the *impossible* assignments. For example, Pixel 0 will never be assigned to Slot 1, because, if hit, it would only be assigned to Slot 0.

$$\text{Full}^{\text{Slot } j}_{\text{Pixel } i} = \text{Assigned}^{\text{Slot } j}_{\text{Pixel } i\text{-}1} \vee \text{Assigned}^{\text{Slot } j}_{\text{Pixel } i\text{-}2} \vee \ldots \vee \text{Assigned}^{\text{Slot } j}_{\text{Pixel } 0} \tag{3.2}$$

It follows that, in order to assign the Slot j to the Pixel i, it must be:

$$\text{Assigned}^{\text{Slot } j}_{\text{Pixel } i} = \neg\text{Full}^{\text{Slot } j}_{\text{Pixel } i} \wedge \text{Full}^{\text{Slot } j\text{-}1}_{\text{Pixel } i} \wedge \text{Full}^{\text{Slot } j\text{-}2}_{\text{Pixel } i} \wedge \ldots \wedge \text{Full}^{\text{Slot } 0}_{\text{Pixel } i} \tag{3.3}$$

That is, that the current slot was not assigned to any of the preceding pixels, and that every preceding slot was instead assigned to a pixel. If this wasn't the case, and the check was to be performed on every pixel assignment and not only on the preceding ones in the priority queue, it would be impossible to reach a stable state, as the assignment of a pixel *i+1* in a slot would modify the slot availability list for the preceding pixel *i*, and so on.

The practical implementation for the first pixels and slots is shown in Fig. 3.16 and Fig. 3.17.

This implementation ends up with a one-hot vector $\text{Assigned}^{\text{Slot } j}$ where the index of the bit set to 1 is the index of the Pixel assigned to the Slot itself. Six 16-to-1 5-bit one-hot multiplexers are therefore needed to map the TOTs correctly.

Deadtime counters One of the problems of the compression approach lies in the fact that the compression can only be performed when the pixels have evaluated their

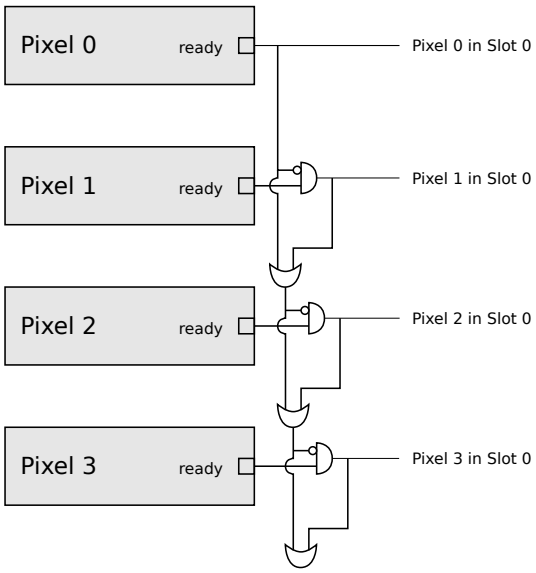

Figure 3.16: Schematic for the CHIPIX65 Zero-suppressor assignment of the first 3 pixels to the 1st ToT slot

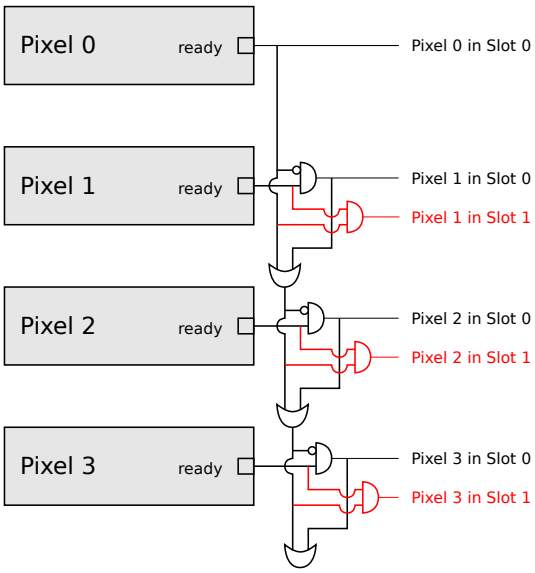

Figure 3.17: Schematic for the CHIPIX65 Zero-suppressor assignment of the first 3 pixels to the 1st and 2nd ToT slots

tots. The ToT computation, however, takes as long as the ToT value itself: it cannot be known a-priori.

This means that the compression must start at the first time it is guaranteed that the TOTs are ready, that is, after the upper limit of the ToT computation time. This value is of 16 clock cycles if the Front-Ends are operated at the 80 MHz clock, but can be reduced if the Synchronous Front-End is operated in Fast Mode. At a reasonable and reliable generated clock frequency of 240 MHz, 6 times the Pixel Region clock, the computation would take 6 40 MHz clock cycles at most.

Therefore, a programmable solution which could select either a 16 (High deadtime) or 6 (Low Deadtime) clock cycles timer was chosen. Every pixel would have an associated *Deadtime Counter*, which preempts the Pixel for the whole duration of the timer, and afterwards propagates a *Ready* signal.

The collection of the *Ready* signals makes the *Hitmap* which is then fed to the Compressor and the data word for the buffer.

This approach has 2 main drawbacks:

1. The 4-bit *Deadtime counter* makes for additional 64 flip flops to be added to the architecture
2. The timer introduces a deadtime (thence the name), to the single pixels, which discard any hits which arrive before the timer has elapsed.

The effect of this pixel deadtime was evaluated with the internal hits generated by the verification environment. The results of the simulation, shown in Fig. 3.18, indicate

that the Fast Mode allows for losses of about 1%.

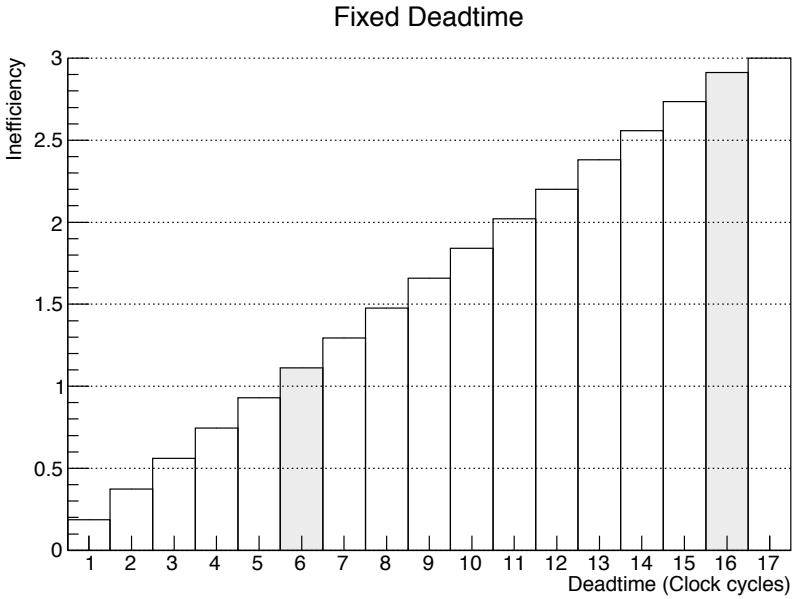

Figure 3.18: CHIPIX65 Pixel fixed deadtime versus inefficiency

It should be noted that the fixed deadtime postpones the time when the data are written to the shared buffer, and thus, the event timestamp: the timestamp recorded for an event will be the timestamp when the hit arrives plus the fixed deadtime. This offset needs to be compensated in the trigger timestamp in order to correctly select the appropriate entries.

3.4.2.3 Buffer and Trigger Matching

The compressed data from the pixel is eventually stored in the shared buffer. The buffer rows, however, also contain the key components for the trigger matching logic. In fact, the structure of the data word in the 16-rows shared buffer is:

1. 9-bit Timestamp
2. 1-bit Valid
3. 16-bit Hitmap
4. 6 x 5-bit TOTs

Once the compressed TOTs are ready, they form, along with the hitmap, the input data word for every row in the buffer. The writing in the rows is signaled by an enable signal driven by a row counter, which is incremented at every buffer writing operation.

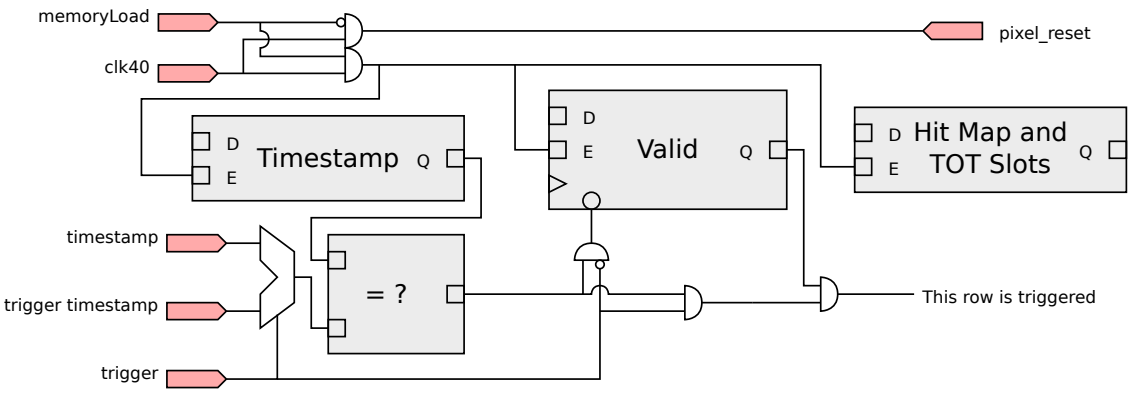

Figure 3.19: Block Diagram of a CHIPIX65 Pixel Region Shared buffer row

A scheme of the implementation in the Pixel Region is shown in Fig. 3.19, along with the control signals for a reference buffer row. In the figure, *memoryLoad* is a one-hot writing selection signal.

The *valid* Flip Flop implements a simple Finite State Machine, with 2 associated states: *Empty* and *Full*.

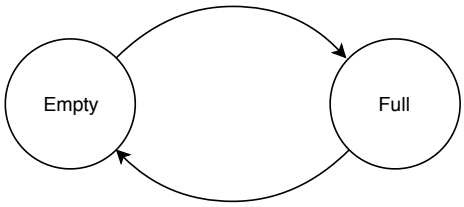

Figure 3.20: Statechart of a CHIPIX65 Pixel Region Shared buffer row

These states encode for the fact that the row contains still valid charge information and timestamp. If the row is *Full*, then, the comparator can be activated to match the saved timestamp against the trigger timestamp coming from the periphery. The status of the row can be reverted back to *Empty* if no trigger arrives after the trigger latency.

The drawback of this single-FF FSM is that the readout procedure usually takes more than 1 clock cycle, and thus, as the row needs to be freed, the triggered data must be buffered in the output stage waiting for readout. As the readout scheme adopted waits until the whole event has been readout before sending the next trigger, the triggered data buffered in the output stage is guaranteed not to get overwritten during new triggers.

The event buffer also supports a triggerless mode, which sends the event out for readout as soon as they are recorded. If this mode is enabled, all the incoming events are saved in the first buffer row. This is compatible with the low hit rates which are

expected to be used in a triggerless mode, as the buses could not sustain a high bandwidth anyway. The trigger matching comparators are inactivated, and the row is immediately selected for readout, bypassing the comparators' logic. The output then follows the usual Free-fall readout scheme.

3.4.2.4 Output Logic

The Output Logic is a simple Finite State Machine which drives the shared bus, ensuring proper communication between the Pixel Regions and the Periphery. As the selected readout scheme for this chip was of Free-fall type, the implementation is straightforward and is summarized in Fig. 3.21.

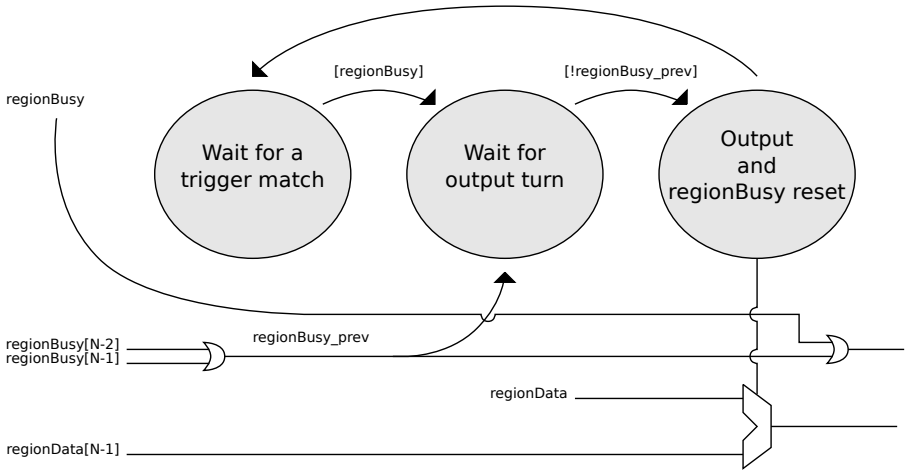

Figure 3.21: Statechart of the CHIPIX65 Pixel Region output FSM

The Finite State Machine stays idle until an event in its buffer is triggered. Thereafter, it awaits for its turn to drive the shared bus, by checking the output state of the preceding Pixel Region. When the bus is free, the Pixel Region propagates its output for one clock cycle and then goes back to the idle state.

3.4.3 Periphery

The data flow in the periphery can be divided in 2 main components: one, replicated for every column of Pixel Regions (Macro Column), handles the communication to and from the Pixel Regions in a column; another, centralizes the data coming from the Pixel Regions and sends them to the serializer (Dispatcher).

In the Chip Periphery lies the timestamp counter: a 9-bit triplicated Full-Adder counter, which restarts from 0 at the overflow. This counter generates the timestamp which is distributed to the whole pixel matrix, through appropriate buffers.

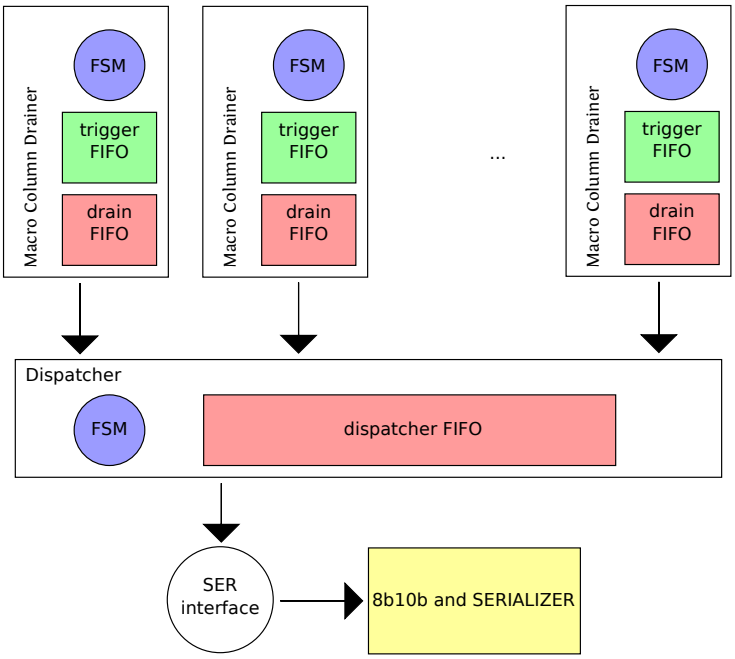

Figure 3.22: Block diagram of the CHIPIX65 Peripheral data flow

Alongside the timestamp counter are a series of subtractors: they are used to generate the timestamp of the triggered data, for the High Deadtime and Low Deadtime cases.

The 3 timestamps (bunch crossing, trigger with high deadtime and trigger with low deadtime) are then encoded in Gray.

3.4.3.1 Data Flow

The Macro Column Drainer is the block which is responsible of the communication to and from the Pixel Regions in a column. It contains two FIFOs: the trigger FIFO, key to the implementation of the Detached-trigger architecture; and a data FIFO, which is used to buffer the events from the Pixel Regions before merging them in the central buffer.

When a trigger arrives at the chip, it is propagated to the Macro Column Drainers where it latches the current trigger timestamp in the FIFO.

A dedicated Finite State Machine, represented in Fig. 3.23, checks the empty flag of the trigger FIFO: if it contains a valid trigger (that is, the trigger FIFO is not empty), it sends the trigger, then awaits one clock cycle for the readout to be processed, and thereafter it waits until all the data has been read (that is, until the busy flag from the Pixel Regions stays high). Once the readout of the event is done, the FSM goes back to the idle state, ready to process a new trigger.

The data being read from the Pixel Regions fills up the data FIFO, where the trigger

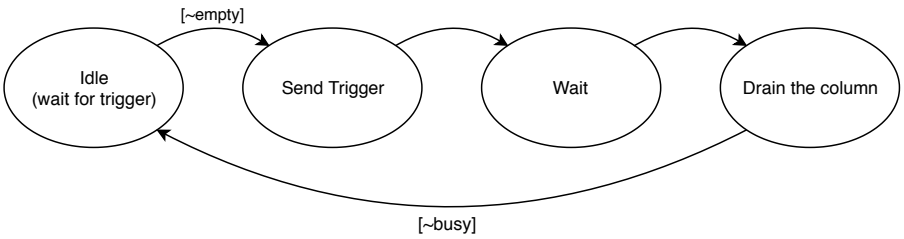

Figure 3.23: Statechart of the CHIPIX65 Macro Column Drainer

timestamp associated with the data is also appended to the charge data itself.

Another important component in the peripheral data flow is the *Dispatcher*: it iterates through all the Macro Column Data FIFOs in order to check their *empty* flags, read their entries and write them in a central FIFO, attaching also the Macro Column index to allow for correct decoding, and pausing, of course, if the receiver FIFO is full. In order to avoid congestion on noisy pixels or columns, the *Dispatcher* reads a maximum of 5 words from a *MacroColumnDrainer* FIFO, before switching to the next non-empty one.

The data word saved in the *Dispatcher* FIFO is thus composed of the following fields:

Bits	Field
63:60	Macro Column Address
59:50	Timestamp
49:46	Pixel Region Address
45:30	Hit Map
29:0	TOTs

3.4.3.2 Data Serialization

The *Dispatcher* FIFO is, in turn, read by an external logic which interfaces it with the Serializer.

The output data format chosen for the chip is a 8b10b encoding: a line code which maps 8-bit words into 10-bit symbols to achieve DC-balance while providing enough state changes to allow clock recovery. This also helps to reduce the demand for the lower bandwidth limit of the channel necessary to transfer the signal.

If there is not data to be read from the *Dispatcher* FIFO, the encoder is fed with a special filling word, which is used for synchronization on the receiver end.

Otherwise, the 64-bit word from the FIFO is split into 16-bit chunks and then packed into a frame consisting of a *Start Of Packet* word, followed by the data. The frame is then fed to a double 8b10b encoder, which processes 16-bit words into 20-bit symbols.

A special mode devised especially for synchronization between sender and receiver ends can be enabled by configuration. In this mode, 2 16-bit synchronization packets

are sent continuously to the double 8b10b encoder, until the synchronization mode is disabled.

All these special words are Control symbols in the 8b10b encoding, activated by a special flag in the encoder. The control symbols within 8b/10b are 10b symbols that are valid sequences of bits (no more than six 1s or 0s) but do not have a corresponding 8b data byte. They are used for low-level control functions. Here follows the list of the control words encoding the various functions implemented in the Chipix protocol.

Special Symbol	Binary value
Start Of Packet	16'b0011110000111100
Idle/Fill	16'b0011110000111100
Sync (1)	16'b1011110010111100
Sync (2)	16'b0001110000111100

3.5 Integration

The chip has been integrated using a top-down digital-on-top approach. The floorplan has been designed in order to optimize the routing of the various interconnections.

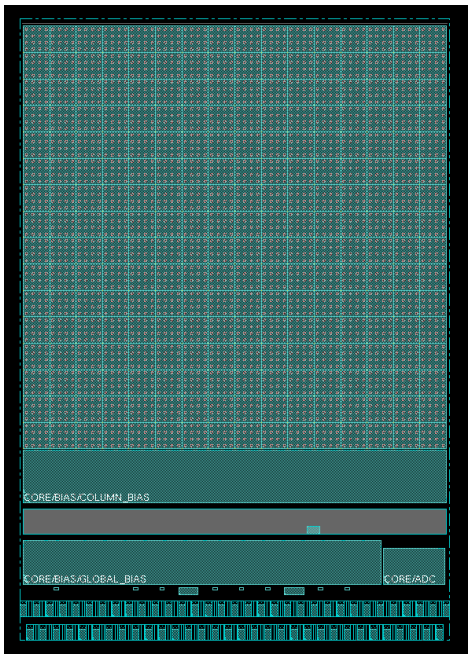

Figure 3.24: Placed view of the CHIPIX65 chip

The top chip floorplan, shown in Fig. 3.24, defines the physical regions assigned to the 16x16 Pixel Region matrix, the Column Bias cells, immediately under the matrix, followed by the Digital Periphery, the Global Bias cells and the ADC. At the bottom lie the padframe and the driving cells. [67]

3.5.1 Pixel Region

The Pixel Region floorplan has been optimized to increase the digital area availability while still protecting the analog biases and front-ends as much as possible. An Analog-island approach[61] has been followed, thus creating, in a 4x4 Pixel Region, 4 different islands, each connected to 4 Analog Front Ends, as shown in Fig. 3.25. [72]

In order to reduce the amount of digital noise in the neighborhood of an analog island, in an effort to reduce the coupling of noise into the analog signals, the placement of the configuration logic has been constrained around the analog islands. The configuration logic is not clocked, as it's made up of latches, and is active only when the bit is being written. Given that the bits are triplicated, a pixel's configuration can last much longer and doesn't need to be refreshed often: it can be considered as silent logic. The placement of these cells around an analog island is displayed on Fig. 3.26.

In Fig. 3.27, the top routing layers are shown. The vertical metal layers distribute both the power signals and the biases for the analog islands. The biases, in particular, are duplicated for the analog island and each line serves a column of Front-Ends only. The horizontal routing distribute the power in a grid-like fashion to ease the routing.

The end result of the placement of the synthesized design is shown in Fig. 3.28, where the Front-Ends are also visible. The routing layers were hidden for clarity.

The following Tab. 3.2 recaps the main characteristics of the placed and routed Pixel Regions in CHIPIX65. The power estimates for both the Pixel Region flavors are of about 7.5 µW/pixel during normal triggered operation, with the 40 MHz ToT clock.

Summary	Pixel Region (Sync)	Pixel Region (Async)
M2 wire length	35.4 nm	34.0 nm
M3 wire length	52.2 nm	52.4 nm
M4 wire length	39.8 nm	36.6 nm
M5 wire length	10.7 nm	10.4 nm
M6 wire length	29.6 nm	30.2 nm
Core density	97.8%	97.8%
Pure gate density	72%	69.4%

Table 3.2: Wire lengths and density of the CHIPIX65 Pixel Regions

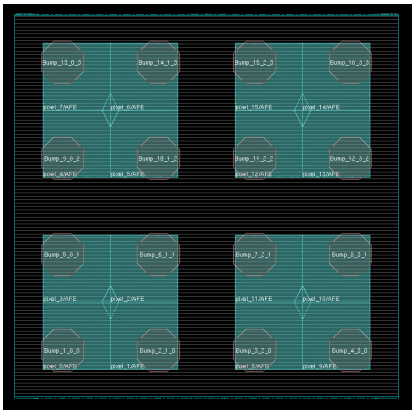

Figure 3.25: Floorplan of the CHIPIX65 Pixel Region, showing Analog Islands and sensor bumps.

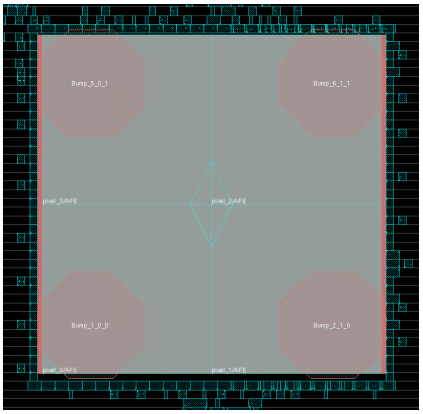

Figure 3.26: Placement of the configuration logic in the CHIPIX65 Analog Island neighborhood

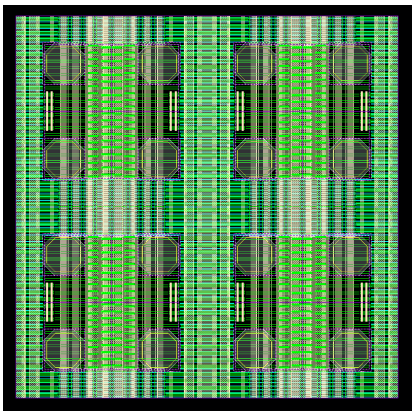

Figure 3.27: Routed view of the CHIPIX65 Pixel Region

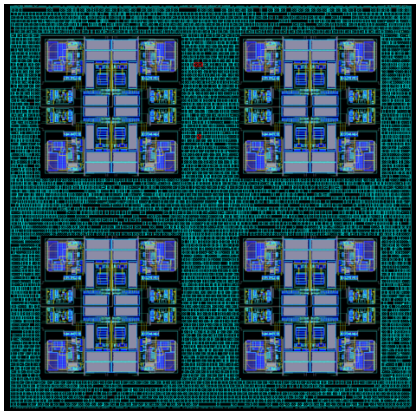

Figure 3.28: Placed view of the CHIPIX65 Pixel Region

3.5.2 Matrix

The matrix has been assembled by replicating the Pixel Region, but there are 2 different Front-End (and, thus, Pixel Region) flavors: the matrix was therefore split into a first half, featuring the synchronous front-end, and a second, which features the asynchonous one. The dichotomy is evident when looking at the layout of the matrix, shown in Fig. 3.29.

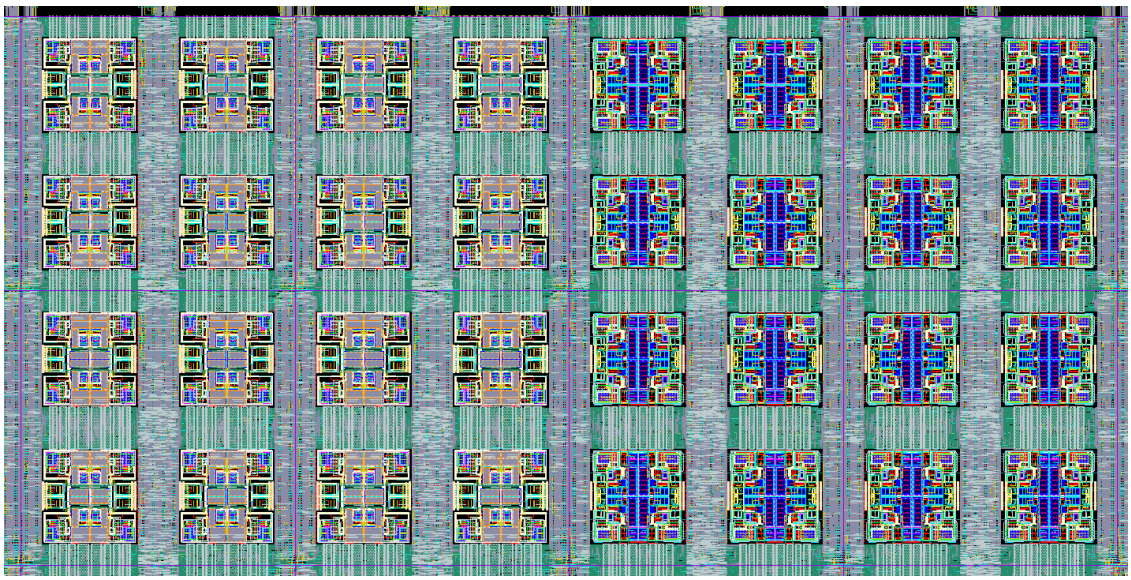

Figure 3.29: Layout view of the CHIPIX65 Pixel Matrix. A focus on the crossover region from the Synchronous FE half, to the Asynchronous FE half.

Pixel Region Address Although Pixel Regions in a column are identical, they still need a way to encode their position in the column. The Pixel Region address is vital to the configuration and readout processes to correctly configure the intended pixel, and reconstruct the pixel address in readout. In the synthesis and place and route of CHIPIX65, the Pixel Region address is an empty blackbox, which is assigned a value for simulation purposes, but whose real content in standard-cell is assigned on full-matrix elaboration via a SKIL script.

Column Timing In order to allow a proper propagation of the signals in the column, certain optimizations had to be made. In particular, one should note that the timing constraints for a Pixel Regions are hard to define... as the preceding block, and the subsequent one are not known a priori: they are, in fact, the same Pixel Region under constraint.

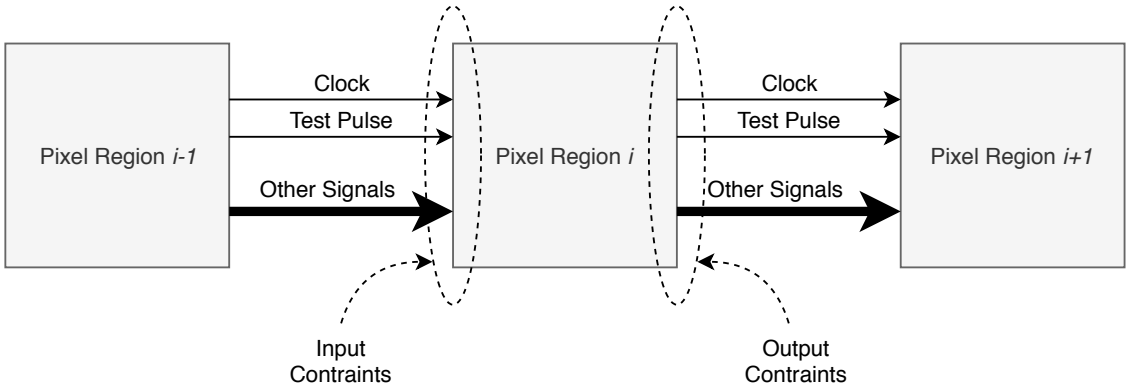

Figure 3.30: Timing constraints for the CHIPIX65 Pixel Regions

Fig. 3.30 shows this situation: the Pixel Region input and output constraints depend on the way the Pixel Region itself is eventually implemented. In order to break this loop, the I/O blocks were defined and placed by hand. In order to optimize, in particular, the synchronization of the Pixel Regions and that of the analog injection signal, the propagation of 2 signals was designed by hand:

- Clock (40 MHz)
- TESTP (Injection)

The structure identified for this is a buffer fork, shown in Fig. 3.31: a dual buffer structure is placed at the bottom of the Pixel Region. The input pin is directly connected to a large buffer (in this example, BUF16). This buffer will drive both another buffer for internal propagation (BUF1), and the next Pixel Region. In this way, it is possible to know for sure the driving capability of the input pin (which is roughly that of BUF16), and the output load (roughly equal to the BUF16 input capacitance).

A large buffer (BUF16, in the example, is a buffer capable of driving 16 cells) typically has a very fast response, and is thus the chosen buffer size for a fast signal propagation.

However, large buffers (large cells in general) typically also have a very high power consumption. If there is no need for fast signal propagation, then, it is also possible

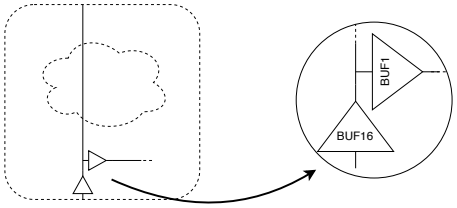

Figure 3.31: A hardcoded signal fork has been used in the CHIPIX65 Pixel Region in order to correctly constraint the timing in the Pixel Region column.

to retain the fork structure with a smaller buffer, in order to properly insert in the constraints file the proper value also in this case.

3.5.3 Periphery and Full-Chip analysis

As the Chip Periphery is not subject to the same constraints as the Pixel Region, the synthesis and place and route operations required no special optimization, apart for the constraints on the timing of the signals to the columns.

Every column must, in fact, receive the signals as synchronous with each other as possible. The area to be placed is much higher than that of the Pixel Region, thus, even if the requirements are somewhat lighter in the periphery, the Place and Route phase take very long. The final layout of the Chip Periphery, showing also the analog blocks and the padframe, is shown in Fig. 3.32.

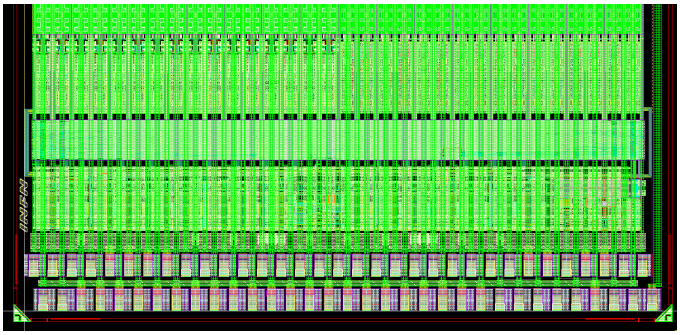

Figure 3.32: Layout view of the CHIPIX65 Periphery

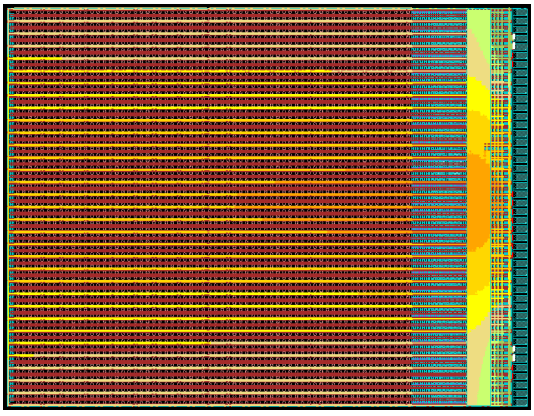

Figure 3.33: Power analysis view for the CHIPIX65 chip, highlighting the IR drop impact. Green regions indicate a VDD level of 1.194 V to 1.2 V, Red of 1.188 V. More than 75% of the chip lies in the 1.19 V to 1.193 V region.

The timing of the assembled matrix and its connection to the Chip Periphery has been tested via multi-corner post-PNR simulations, using SDF timing files.

The power analysis was also key to detect the IR drop along the column, which was found to be acceptable. A screenshot of this analysis is shown in Fig. 3.33.

3.6 DAQ Setup and Tests

After the chip has been sent to manufacturing, the design team helped the DAQ team in the customization of a DAQ setup already available at INFN Torino, in order to make it compatible with the CHIPIX65 output data scheme and mode. The chips came back from the foundry in September 2016, and soon afterwards they have been wire-bonded to a custom PCB test board.

The single-chip PCB board communicates with a FPGA development board powered by a Xilinx Kintex-7 via the FMC connector. Communication to and from the FPGA board happens via a custom Ethernet/UDP protocol. The PC uses a Labview^{TM} Virtual Instrument in order to manage the tests and elaborate the results. A portion of the DAQ interface during data acquisition is shown in Fig. 3.34.

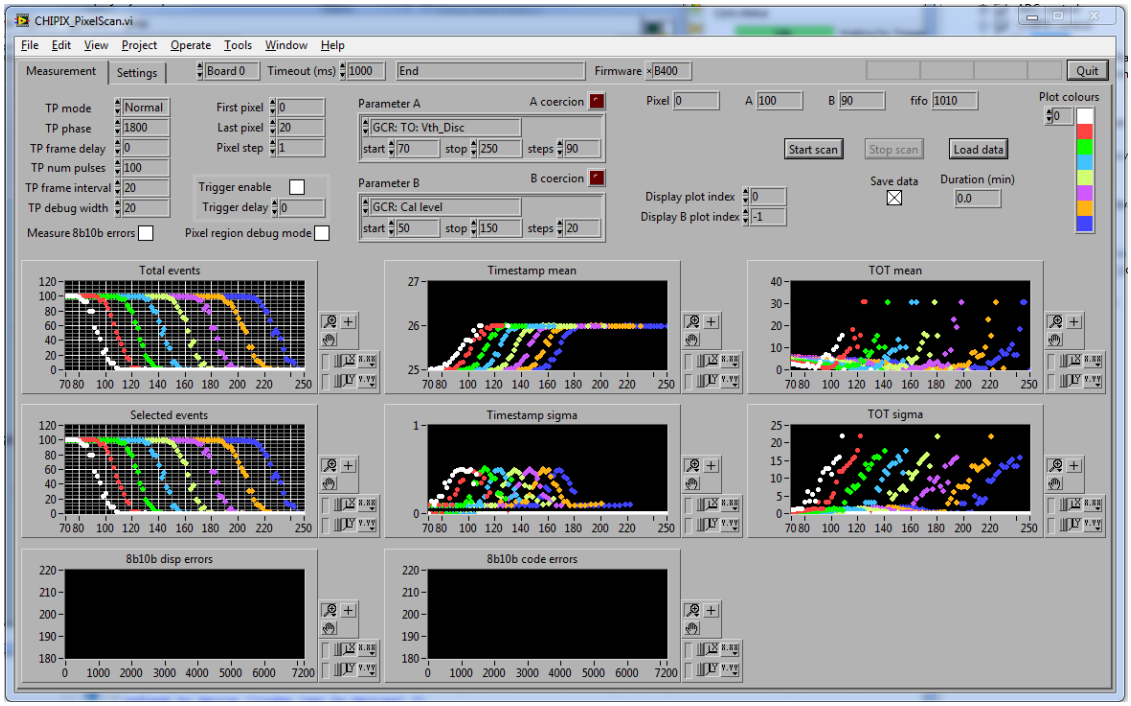

Figure 3.34: The CHIPIX65 DAQ interface, showing a data acquisition.

The tests have been performed at INFN Bari, Bergamo and Torino. Moreover, irradiation campaigns have been performed at INFN Padova and CERN PH/ESE facilities,

with the electronics always biased at nominal operating conditions and under continuous charge scans.

3.6.1 IP Blocks

Extensive validation has been performed against the predicted performance of the IP Blocks designed for the prototype. A key IP Block descripted in this Chapter is the Digital-to-Analog Converter, the element which defines the operation point of many other blocks in the chip. Its performance was found to be in full agreement with the SPICE simulations, as shown in Fig. 3.36.

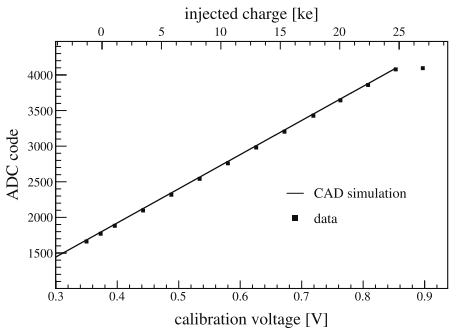

Figure 3.35: Linearity test of the CHIPIX65 ADC

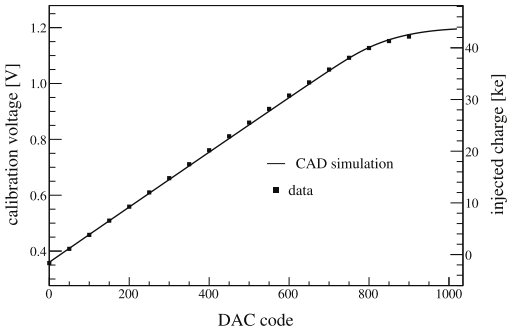

Figure 3.36: Linearity test of the CHIPIX65 DAC

Although the chip features an analog output which can be used to assert the value of the analog biases generated by the ADC, an internal monitoring DAC was also embedded in the design to provide for an internal test point, not affected by signal offsets and distortion by the output pads impedance. Fig. 3.35 also shows that the measurements are in agreement with the predictions.

3.6.2 Front-Ends

The performance of the Front-Ends, in terms of noise and minimum threshold, can be asserted via 2 types of tests: the calibration voltage scan, and the threshold voltage scan.

The calibration voltage scan consists in fixing the threshold value, and scanning (increasing the calibration voltage from a minimum value to a maximum or vice versa) through various steps. The number of steps determines the granularity of the scan, in a way such that more steps increase the accuracy of the scan. At every calibration voltage step, a predetermined number of events are injected. The response of the discriminator of the front-end are then recorded (hit/no-hit), and the number of recorded hits is associated to the step. Once the scan is complete, the plotted result will be an S-curve

(as, in practice, it is the deconvolution of a step-function with the gaussian noise of the FE), as at low calibration voltage levels the Front-Ends will not fire, but will start to do so when the calibration voltage will become comparable with the threshold voltage. At calibration voltage levels much higher than the threshold voltage, the Front-End fire every time.

The threshold voltage scan works in a similar way: the calibration voltage level is fixed, but the threshold voltage is scanned from a high/low point to a low/high one. When the threshold becomes very low, and comparable with the noise, the Front-Ends can fire even if there's no real charge injection. The presence of noisy hits can be determined by efficiency greater than 1, or by hit timestamps not corresponding to the timestamp of the injected charges.

3.6.2.1 Synchronous FE

The effective threshold of the synchronous FE was tested via a Calibration signal scan. The auto-zeroing procedure sends an auto-zeroing pulse of 75 ns every 100 µs. 100 injection pulses are sent to the pixels, and the number of hits are then counted off-chip. The hits per calibration voltage plot is also fitted with a sigmoid error function, which provides the means for extracting the curve's mean and sigma values, as shown in Fig. 3.37.

The Calibration scan was also used to assess the residual latch dynamic offset: some results are shown in Fig. 3.38. The offset was found to be of about 100 electrons RMS, in good agreement with CAD simulations (about 70 electrons RMS). The results demonstrates that the auto-zeroing process is working correctly.

A Threshold scan was instead key in the definition of the minimum threshold, in this case found to be of about 250 electrons, in full agreement with the predicted CAD simulation results. It is clear by Fig. 3.39 how for thresholds lower than this value, the pixel starts to fire from noise fluctuations.

The ToT linearity was also assessed by changing the injected charge value. It should be noted that the threshold charge value must be added again during the analysis as it is inherently subtracted in a ToT calculation. The linearity results are shown in Fig. 3.40 for a slot ToT clock. The fast ToT clock introduces an error in the order of tens of %, as expected by CAD simulations, and are due to mismatches in the analog part. [70]

After a 600 Mrad irradiation at −20 °C, the chip is still functional, and the increase of the threshold dispersion is below 10%, while the ENC shows a 10% increase.

3.6.2.2 Asynchronous FE

The analysis of the Asynchronous Front-End need a calibration procedure for the threshold setting. Before this *threshold tuning* operation, a threshold dispersion of around 400 electrons rms has been obtained. After tuning the threshold dispersion is reduced to 45 electrons rms. From circuit simulations, the main contributions to the dispersion

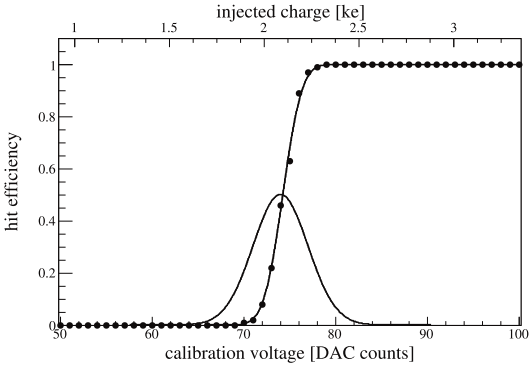

Figure 3.37: Calibration test for the CHIPIX65 Synchronous Front-End

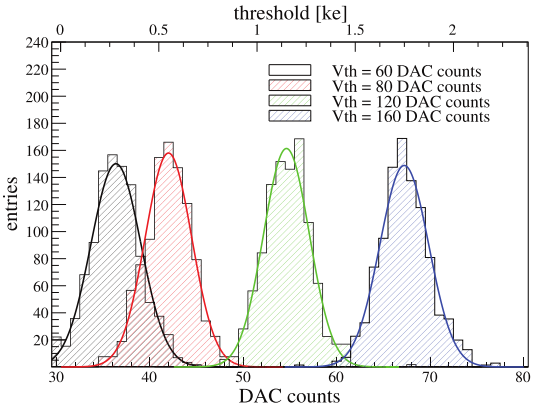

Figure 3.38: Residual Latch dynamic offset test for the CHIPIX65 Synchronous Front-End

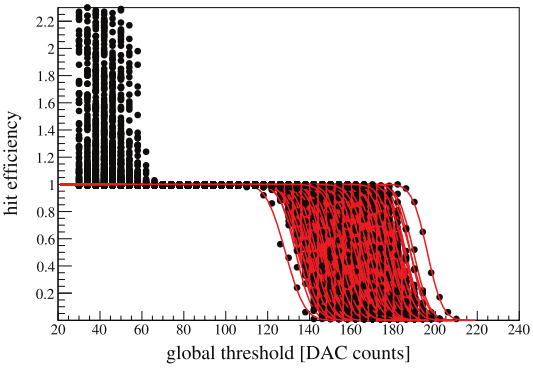

Figure 3.39: Threshold scan on the CHIPIX65 Synchronous Front-End

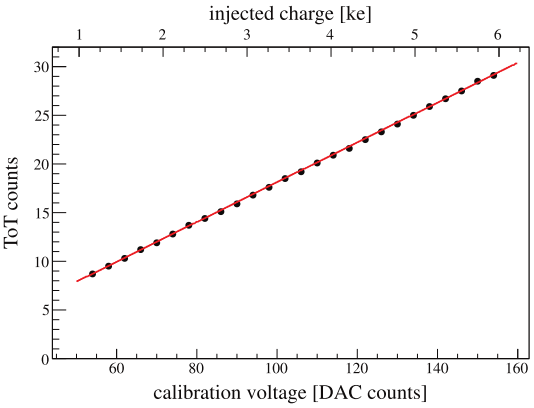

Figure 3.40: ToT linearity test on the CHIPIX65 Synchronous Front-End

are the differential pair and load transistors of the comparator input stage, contributing to 60% of the total threshold dispersion.

The threshold dispersion is significantly affected by radiation. A non-negligible increase, close to 50%, is already detectable at the first irradiation step, 0.1 Mrad, and increases up to 160 electrons rms at 1 Mrad TID. If the front-ends are re-tuned, the dispersion lowers down to 65 electrons rms. At 630 Mrad, the re-tuning procedure yields a final threshold dispersion equal to 150 electrons rms.

By fitting the hit efficiency curves it is possible to assess the noise performance of the analog front end. A mean ENC of 98 electrons rms has been measured when no sensor is connected to the Front-End.

The ToT linearity has also been studied, and the results show a very good linearity (with an integral nonlinearity equal to 2%) of the ToT for input charges larger than 2000 electrons. The nonlinear behavior foreseen for injected charges lower than 2000 electrons is in fairly good agreement with the experimental ToT data. [36]

3.6.2.3 Sensor Tests

The CHIPIX65 chip has been tested with a 3D sensor from FBK, made with a single-sided process. The 3D sensors employed came in 50 µm × 50 µm and 25 µm × 100 µm flavors. Fig. 3.41 shows a picture of the diced sensor bump-bonded on the chip.

After the bump bonding, and a proper verification of the sensors' electrical connectivity to the chips, the tests proceeded with a measurement of the average noise per pixel, extracted by performing S-curves as a function of the reverse sensor bias. In this paragraph, the tests of the Synchronous FE will be explored.

As expected, the 25 µm × 100 µm sensors yielded the lowest noise, as their capacitance is slightly lower than the other form factor. This noise is of about 105 e^-, as can be seen in Fig. 3.42

The ToT linerity equation $Q = a * \mathrm{ToT} + b$ has been completed with the characterization of the front-end with the sensor. By using a 80 MHz ToT count, and a krummenacher feedback counter that makes a 6ke charge last 90 ns, the gain was found to be of 700e^-/TOT, while the offset of 265e^-. The threshold was set to 800e−, as to produce a 1 Hz noise hit rate, much below the expected 50 kHz.

The tests have been performed with 3 different sources: Barium, Americium214, Strontium90. The results are shown in Fig. 3.43, Fig. 3.44 and Fig. 3.45.

The Barium source releases 32 keV γ rays, equivalent to 8800e^-, and the sensor recorded a Poissonian curve with mean value 8250e^-. The Americium-241 source, instead, releases 59.5 keV γ rays, equivalent to 16.5e^-, and the sensor recorded a charge distribution with mean value 14550e^-. The Strontium-90 tests, finally, resulted in a charge distribution with mean value 10690e^-. [68]

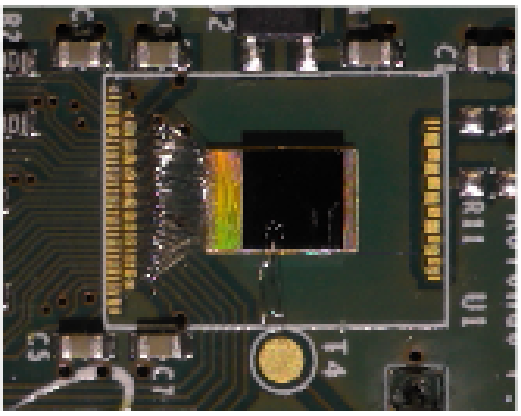

Figure 3.41: The CHIPIX65 chip bonded to a FBK 3D sensor on the test board.

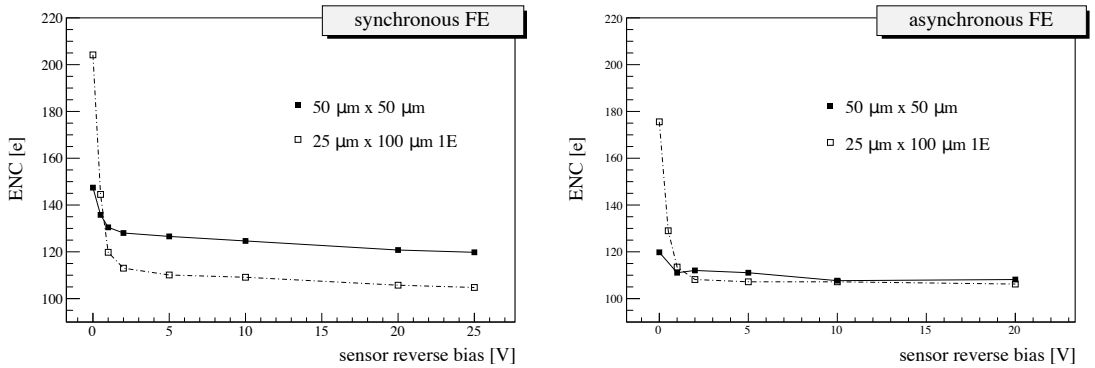

Figure 3.42: Noise tests on the Synchronous FE in the CHIPIX65 chip bonded with a FBK 3D sensor. [69]

3.6.3 Digital

The digital performances were already known before the fabrication due to the extensive simulations. As the main changes introduced by the manufacturing process influence the timing of the digital part, which cannot be measured in the Chipix chip due to the lack of test structures, the main results are represented by the functionality checks performed on the chip.

The chip was found to be working post-fabrication, which represents a first step in the assessment of non-evident routing errors. The digital features and options introduced in the chip were found to be working as expected in all cases. The chip was also tested at various irradiation doses, and found to be fully functional up until 600 Mrad. [69]

No functional bug has been discovered, and the chips work as expected.

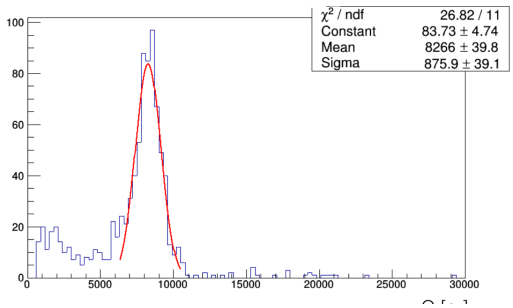

Figure 3.43: Tests with a Barium X-ray source on the Synchronous FE in the CHIPIX65 chip bonded with a FBK 50 µm × 50 µm 3D sensor.

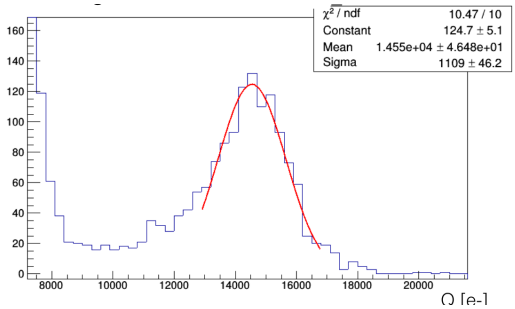

Figure 3.44: Tests with an Americium-241 X-ray source on the Synchronous FE in the CHIPIX65 chip bonded with a FBK 50 µm × 50 µm 3D sensor.

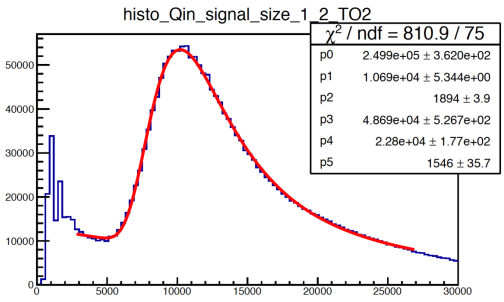

Figure 3.45: Tests with a Strontium-90 X-ray source on the Synchronous FE in the CHIPIX65 chip bonded with a FBK 50 µm × 50 µm 3D sensor.

3.7 Summary and Future Work

The chip presented in this chapter represented a major success in the validation of the 65nm CMOS technology for future detectors, meeting most of the requirements and containing a large number of testing features used to characterize both analog and digital logic, with a number of sensors attached. In this regard, the CHIPIX65 project proved to be of fundamental importance in the development of future 65nm CMOS pixel chips.

The performance of the digital logic in simulations yielded a 0.04% event loss due to buffer overflows, and a 0.4% charge information loss due to the limited number of ToT slots. The tests performed with injections or in test beams confirm the results of the simulations, allowing a thorough characterization of the Analog Front-Ends and the sensors attached.

At the same time, however, some solutions adopted for the chip are not scalable enough for use in a full-scale demonstrator, and had to be revised. These are detailed below.

Deadtime The digital performance has a relevant limitation determined by the fixed deadtime, which introduce a single-pixel loss of about 3% with a 40 MHz ToT clock, which however can be brought down to 0.4% and lower if a Fast ToT clock of about 320 MHz is used. The suboptimal solution of the temporary inactivation of the pixels introduce losses uncorrelated with the effective deposited charge. Much effort has been put into the research of an appropriate methodology to remove this limitation, in order to further reduce the buffering inefficiencies.

Readout mode The readout mode, which can sporadically introduce ghost hits, can also be improved. This solution, appropriate for a small scale demonstrator, is not very suitable for a full scale one, as the readout of the column would introduce large windows in which the triggers have to be buffered: the trigger FIFO would increase, as would the time the triggered hits have to be retained in the Pixel Region buffer, possibly going through overwriting in the process. For a full scale demonstrator, a readout mode based on trigger tags would probably be more appropriate.

Power The CHIPIX65 power consumption is fairly high, about 7.5 MW limit per pixel. This meets the first CHIPIX65 specifications, but as more detailed specs for the Phase 2 Pixel Upgrade arrived, it became clear that a significant effort has to be put into the power optimization of the logic.

The clock gating strategy could be improved significantly, without sacrificing the timing performance of the Pixel Region. In general, the Pixel Region can be restructured in a way to reduce the timing constraints, which can result in bigger buffers and cells, and thus can dramatically increase the power consumption of both sequential blocks and the clock network. Such strategies are adopted in the CBA architecture of RD53A.

Chapter 4

RD53A

The RD53 collaboration was established in 2013 to design a hybrid pixel readout chip for the high rate and radiation expected in the ATLAS and CMS phase 2 upgrades. But the chip in only the the end result of a process envisioned to produce all the elements and building blocks required for future chips. [23, 25]

The first large scale prototype of the collaboration is RD53A, intended to demonstrate in a large format IC the suitability of the chosen 65nm CMOS technology (including radiation tolerance), the stable low threshold operation, and the high hit and trigger rate capabilities, required for HL-LHC upgrades. The chip itself is not intended to be a final production IC for use by the experiments, and thus containts design variations for testing purposes. [41]

The RD53A integrated circuit specifications were approved in Fall 2015 after review by the ATLAS, CMS, and RD53 collaborations, and the design operations began in Spring 2016.

The author of this thesis was primarily involved in the core design team, consisting of about 10 experts in the field, with specific tasks on the overall chip and matrix integration, and pixel region architecture definitions and optimizations. Such tasks include the development of the Analog Front-Ends, Analog IP Blocks, I/O Blocks, Serial Powering, high-performance digital architecture, Chip Integration practices and Verification.

Chip Size and Matrix structure The height of the RD53A chip is constrained by the available space on the shared reticle submission. The engineering run, in fact, has been shared with the design of another chip of the CMS experiment (the MPA chip) in order to optimise the cost of the submission: for this reason the size of RD53A was limited to about one half of the final chip for LHC experiments.

The RD53A pixel matrix is 400 pixels wide by 192 pixel tall. Production chips are expected to increase the number of rows and remove the top row of test pads, therefore the power and bias distribution have been designed for a larger number of rows, up to 384.

The pixel matrix has been organized into Pixel Cores of 8×8 pixels, whose internal

structure depends on the Pixel Region implementation.

The matrix features 3 front-ends (Synchronous, Linear, Differential) and 2 Pixel Region architectures (a Distributed Buffering Architecture and a Centralized Buffering Architecture). The scope of this chapter will be focused on the digital architectures (in particular the CBA, developed by the author) and the interface to the Synchronous FE. [35]

Chip requirements overview The RD53A pixel chip specifications evolved with time to reflect a number of requests made from the CMS and ATLAS experiments. Because of this, some of the design parameters had to be changed during the chip development. The final requirements for the pixel chip are listed below.

Parameter	Specification
Hit loss	$\leq 1\%$
Trigger rate	1 MHz
Trigger latency	12.5 µs
Radiation tolerance	500 Mrad
Temperature range	−40 °C to 40 °C
Current consumption per pixel	<8 µA
Hit rate	75 kHz/pixel
Hit charge resolution	600 e^-
Hit charge dynamic range	≥ 4 bits
Pixel pitch	50 µm

The chip must also host a series of features needed for both the final chip and the testability of the current test chip and its components.

- Default Configuration mode for all the pixels
- Pixel masking
- Calibration signal injection
- Output Hit Or
- Matrix BX clock synchronization

4.1 The Analog Front-Ends

Although the original plans for the chip allowed for 4 different Front-Ends to be embedded, one of the designs has been dropped, so that the chip eventually contains three

different front end designs to allow detailed performance comparisons. The designs are not variations of a common design, but implement significantly different Front-End concepts.

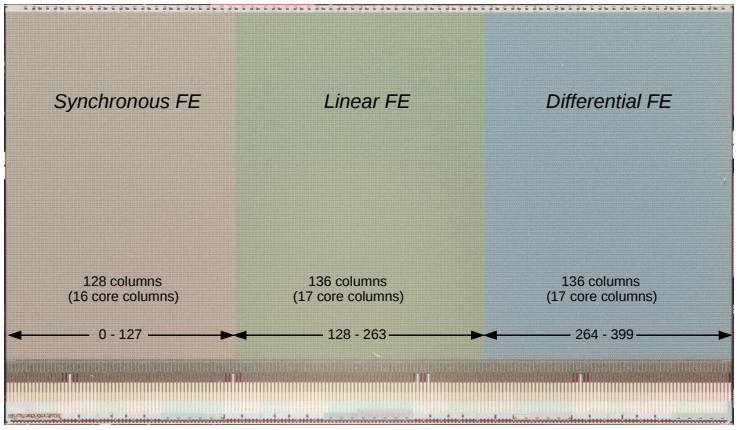

Figure 4.1: The RD53A chip features 3 different flavors of Front-Ends with 2 different digital architectures. [39]

The Front-Ends are identified as Differential, Linear and Synchronous. The Differential FE uses a differential gain stage in front of the discriminator and implements a threshold by unbalancing the two branches. The Linear FE implements a linear pulse amplification in front of the discriminator, which compares the pulse to a threshold voltage. The Synchronous FE uses a baseline ”auto-zeroing” scheme that requires periodic acquisition of a baseline instead of pixel-by-pixel 360 threshold trimming.

The Linear FE and the Synchronous FE are improved versions of the FE developed in the context of the CHIPIX65 collaboration.

The designs, however, share some common constraints and features, the main being the layout area and the bump bond pads, which are the same for all designs, making them easily interchangeable on the pixel matrix layout, and the calibration injection, key for direct performance comparisons. The bias distribution also follows the same organization for all 3 flavors.

The first 16 Pixel Core columns (with each Core Column consisting of 8 pixel columns) have been assigned to the Sync FE, the following 17 to the Linear FE, and the final 17 to the Differential FE. The improvements apported to the design of the single Front-Ends from the CHIPIX65 experience do not affect their output interface with the digital logic: for this reason, here we will not go further in their description.

4.2 The Digital Architectures

Considerable part of the work for this thesis has been put into the development of the digital architectures for this chip, with a particular focus on the CHIPIX65 successor: the CBA architecture. However, the digital design team worked together in the research of the architecture with the best performance, and thus many design solutions have been shared during the design phases.

In this section, I will briefly describe the organization of the DBA architecture, and some common solutions, such as the tag-based readout derived from the FE65_P2 experience, which were adopted in order to increase the compatibility with both architectures with the readout and reduce the design efforts.

4.2.1 Distributed Buffering Architecture

The DBA features 1x4 Pixel Regions, which share a common distributed buffer, where the charge is saved in per-pixel memories, and the timestamp buffer is centralized. This scheme, also used in the FE65_P2 and FE-I4 chips, had inspired the first version of the CHIPIX65 architecture, although here on a smaller scale.

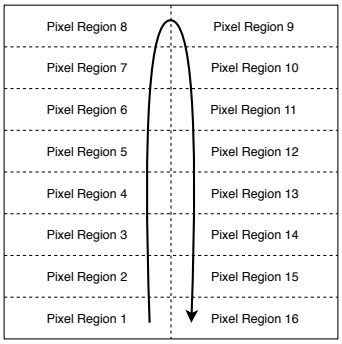

Figure 4.2: Position of the DBA Pixel Regions in the RD53A Pixel Core

The 8×8 Pixel Core, then, contains 16 Pixel Regions, arranged as a reverse U shape, as shown in Fig. 4.2. This allows easy connectivity of the readout token, which tells the Pixel Regions when it's their turn to drive the output data bus.

Pixel Region The Pixel Region in the DBA architecture store all the charge associated with a region-wise hit, along with the zeroes. A general structure for this architecture is displayed in Fig. 4.3.

The ToT memories and the Timestamp memory are in 2 different hierarchical entities: the Pixel Logic and the Pixel Region Logic. The Timestamp memory keeps track of the buffer location to be assigned to the next event to be recorded. The ToT memories

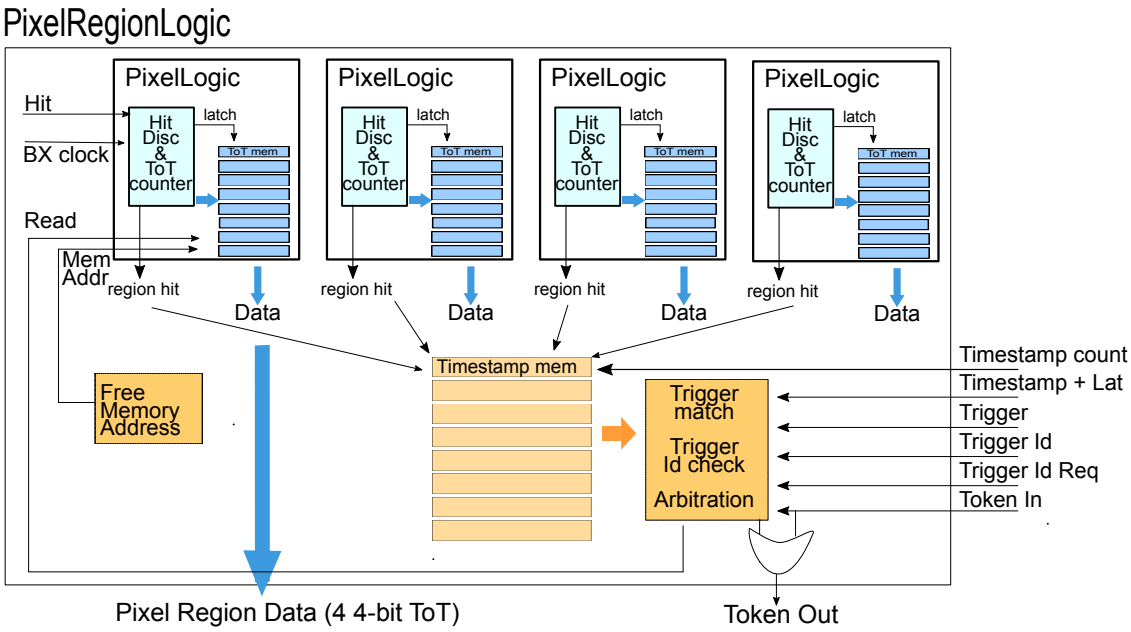

Figure 4.3: Block Diagram of the DBA Pixel Region in RD53A

are instead distributed throughout the regions, so that every pixel has a dedicated ToT buffer, which can be easily routed in the Pixel Logic neighborhood.

When a pixel is hit, it saves in a dedicated memory the address from the shared timestamp buffer: the ToT memory will, in fact, follow the shared one in the association between the buffer row and the event. This means that if an event timestamp is to be written in the row N of the shared buffer, the ToT of all the pixels hit will be written in the row N of the ToT buffer. The pixels which are not hit will, instead, write a ToT=0 code in the corresponding row.

It is fundamental that the pixels save the row address of the shared buffer as it is not known when the ToT computation will end, and therefore the row address might change in the mean time if another particle strike the Pixel Region.

Pixel Logic As the architecture follows from the original implementation of the FE-I4 Pixel Region, some of the inner signals maintained their historical names.

The binary output of the FE discriminator is fed to a 2-FF queue. In this way, it is possible to automatically generate a Leading-Edge pulse, fired one clock cycle after the recognition of the hit, and a Trailing-Edge pulse, fired one clock cycle after the hit signal becomes low again. These 2 signals are used to trigger 2 writing operations in the pixel: the shared memory row address is written on the Leading Edge, while the ToT is written at the Falling Edge in the row addressed by the index saved in the Leading Edge. A schematic of this organization can be found in Fig. 4.4.

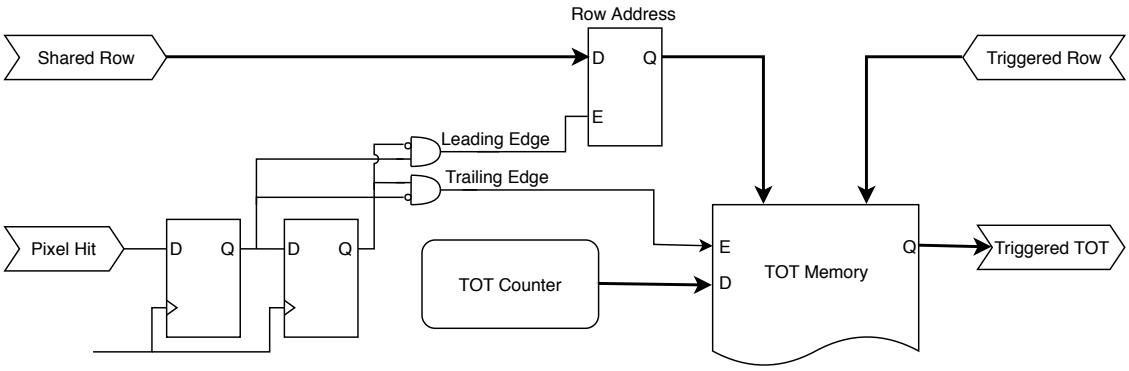

Figure 4.4: Block Diagram of the Pixel Logic in the RD53A DBA Pixel Region

Buffer The central buffer in the DBA architecture stores the timestamp of the events. The row selection is made via a dedicated combinational logic which selects the first available row in the buffer. This is the address that's propagated to the pixels as well, in order to align the writing of the TOTs.

4.2.2 Common Trigger and Readout logic

As introduced before, the RD53A implements a tag-based readout, already used successfully in the FE65-P2 pixel chip. This readout mechanism involves the association of a trigger tag to the triggered events. This tag can be used to identify the triggered events in a way that allows for ordered readout.

When propagating a trigger signal, the tag to be attached to this event is also propagated to the pixels. The triggered events must save this readout tag information in a separate field, or in that of the timestamp, as it is no longer needed.

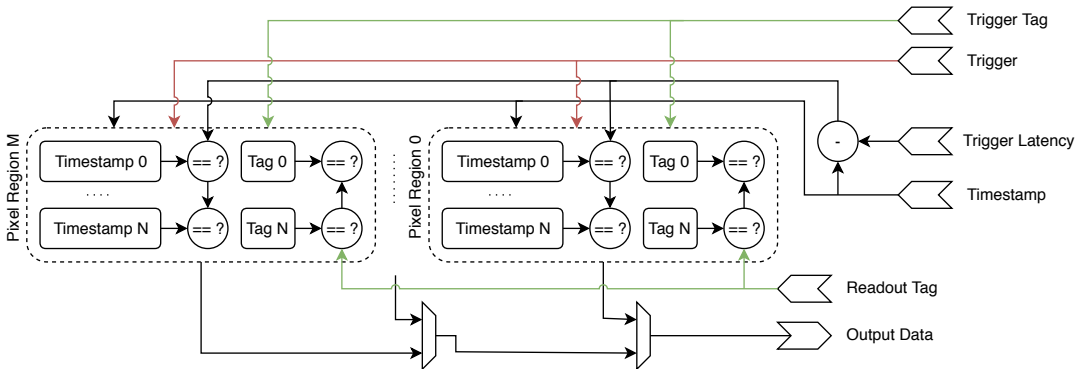

Figure 4.5: Schematic view of the buffer cells in a Pixel Region supporting a tag-based readout

The triggering and readout phases are then fully detached by the propagation, on a

separate bus, of the readout tag. If the readout tag matches the saved tag, the event is selected for readout.

In order to implement this readout, the 2-FF Finite State Machine used in the CHIPIX65 Pixel Region buffer is not sufficient: the readout requires 4 states.

With the introduction of the *Triggered* state, the row can switch to a state where it can await readout.

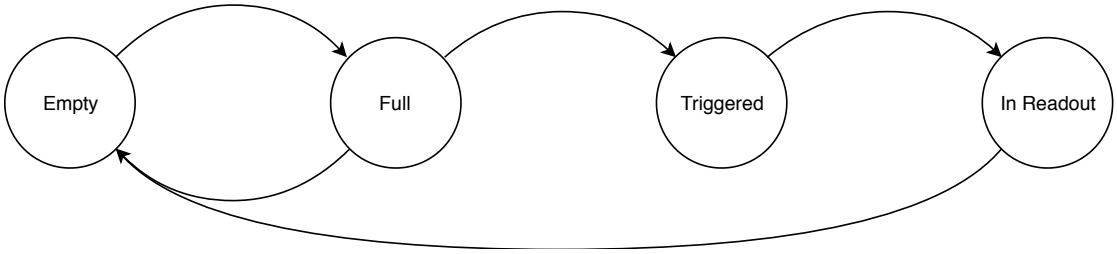

Figure 4.6: Statechart of a buffer row FSM supporting a tag-based readout

When the trigger timestamp equals the saved timestamp and, at the same time, the trigger signal is high, the state of the memory cell can switch from the *Full* state to the *Triggered* state. During this transition, the trigger tag associated with the trigger must be saved in the row. In the RD53A implementation, the tag is written in the timestamp memory of the row, which is not needed anymore, as the event has already been triggered.

The *Triggered* state makes the event row insensitive to the trigger timestamp, but enables a comparison of the saved trigger tag with the readout-requested trigger tag. When the two match, the FSM can switch to the last state: *In Readout*.

When in the *In Readout* state, the row simply awaits for the readout token to pass to the current Pixel Region (which happens when all the preceding Pixel Regions have already sent their output). When the token arrives at the Pixel Region, the row *In Readout* drives the shared bus. The periphery will thereafter acknowledge the readout with a *Read* signal, which will mark the transition of the row FSM from the *In Readout* state back to the *Idle* state.

Summing up, the single memory cells feature a 2-FF FSM, encoding for the 4 states:

- Empty → The row is currently empty
- Data → The row is waiting for the trigger latency to elapse
- Triggered → The row contains triggered data, and is waiting for its tag to be selected
- Tag readout → The row's tag is being readout, and the row is waiting its turn

4.3 The CBA Digital Architecture

The CBA digital architecture's heritage lies in the CHIPIX65 project, and was developed from that substrate and concept. It presents major improvements over the CHIPIX65 implementation, starting from the problem of the fixed deadtime.

The compression of the ToT must take place only after all the TOTs are computed. For this reason, as the number of clock cycles needed for the compression are not known, the compression stage must be postponed after the maximum possible ToT computation time, which is a function of the ToT clock frequency and the number of ToT bits.

In principle, it would be possible to postpone the compression to the moment when the last pixel is done computing its ToT, but this means that the compressed ToT would be written at a unknown time after the hit. This uncertainty has to be corrected, for example by subtracting the value of the ToT (in 40 MHz counts) from the timestamp. The problem with this approach is that the timestamp is propagated in Gray in order to save power, and the only way to operate on Gray numbers is by transform them in their binary counterparts, operate on them, and transform them back in Gray. These operations require a non-negligible amount of combinational logic to be implemented, which may be problematic for the area they would need. It is reasonable, therefore, to wait a specific, maximum number of clock cycle before compressing the TOTs.

If the pixel-specific deadtime wants to be excluded, then the synchronization stage has to be delegated to a separate component, which has been called a *Staging buffer*, which keeps track of the pixels hit, their TOTs, and the time elapsed from the hit timestamp.

4.3.1 Pixel Region

The CBA Pixel Region has the same number of pixels as the CHIPIX65 Pixel Region, although with the new improvements on the Verification Environment, the optimal form-factor between 4×4 and 2×8 has been studied, along with their implications in terms of routing congestion.

In order to maintain compatibility with the other digital architecture, the triggering and readout mechanism of the CBA has been adapted to implement the same tag-based solution, which also assured that only minimal changes would be needed in the Chip Periphery to interface the peripheral logic to the CBA.

The work on the Pixel Region has started by using the DBA code as a template, and replacing the single components with the CHIPIX65 counterparts: this allowed to maintain consistency on the naming and port conventions, and to make the whole architecture more human readable. In the core, the 16 DBA Pixel Regions were replaced by 4 CBA ones. The pixel logic was modified by integrating the necessary parts needed to operate the CBA logic, and the ToT buffers were removed. At the same time, the latency memories were expanded to make them host also the hitmaps and the TOTs. The

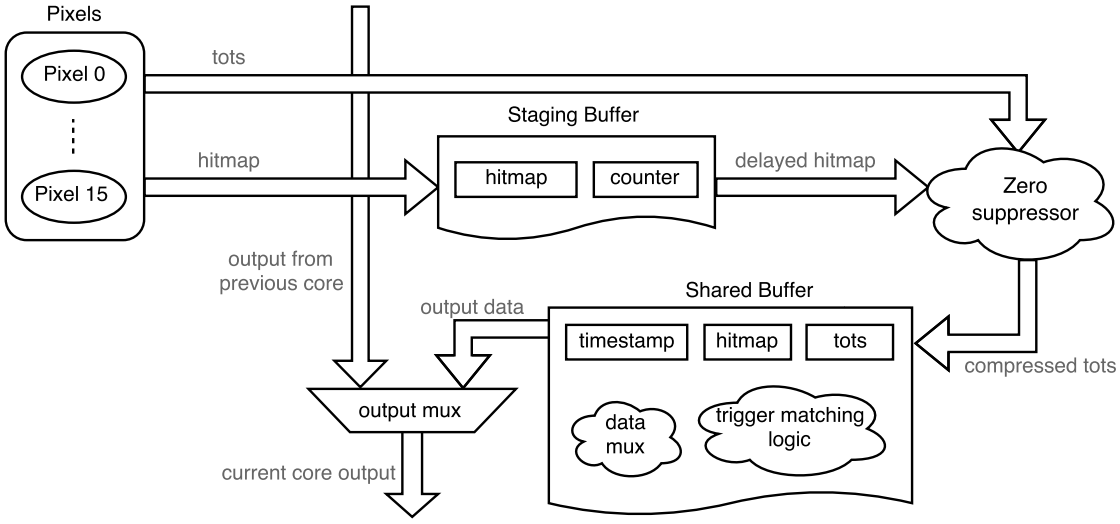

Figure 4.7: Block Diagram of the Centralized Buffer Architecture in RD53A

ToT compression module was also integrated and connected in the Pixel Region, and the output data format and logic was modified to support the larger data bus. A data re-builder was designed in order to make the simulation environment support the new data format.

The work started with the integration of the Differential Front-End, with uses, along with the Linear Front-End, a binary output. The integration of the Synchronous Front-End highlighted that the logic necessary for its integration is quite power hungry. Better modeling of the Front-End clocks, along with smarter clock gatings lowered the power consumption, but it remained higher than for the other 2. After the decision on the Digital Architecture to Front-End assignment, work has focused on the implementation of the Synchronous FE.

At first, it was proposed to use a high number of ToT bits (about 6), as it could be useful for sensor characterization. The CBA offered a great advantage in this regard, but hit reconstruction studies highlighted how, for track reconstructions, it was sufficient to store the pixel charges in 4 bits. In order to ease the design, the minimum requirements were first considered, with the possibility to consider increasing this number in the future.

4.3.1.1 Pixel Logic

The Pixel Logic in the CBA architecture is very similar to the one of CHIPIX65: it multiplexes the clocks for the Synchronous FE, evaluates the ToT via a dedicated counter, and implements the basic FSM needed for operation.

As explained before, the revised architecture does not feature a deadtime counter, so it is ready to process a new hit immediately after it has handled the preceding one. The main advantage of this solution is that, if a fast ToT clock is used, the deadtime of

the pixels can be cut down to a 3-4 40 MHz clock cycles. If a 40 MHz clock is used for ToT computation, the staging time would instead necessarily be 16 clock cycles.

In order to free the pixel to make it available for the processing of new hits, it is clear that the ToT computed by the counter must be saved someplace, so that the counter can evaluate the ToT of the new signal. In Fig. 4.8 this situation is shown clearly: a first hit arrives at time 2. It processes its ToT until time 10, and another hit arrives at time 15. However, the first ToT is not used until time 18: if the counter is not freed in time, it cannot evaluate the ToT of the second particle, and, at the same time, if it is not stored somewhere, the ToT of the first particle will be lost.

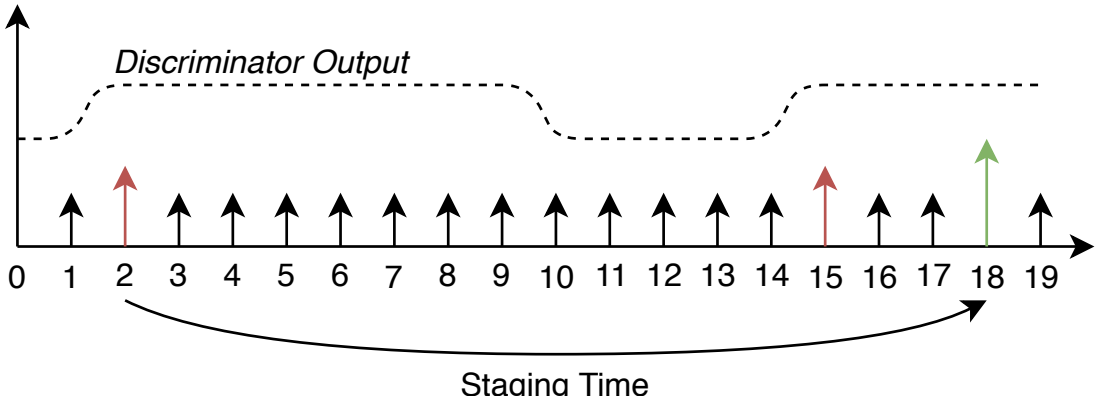

Figure 4.8: The Pixel's ToT is processed after a Staging Time in RD53A's CBA

In practice, the quantity of interest is the probability of a second particle to strike the pixel after the ToT computation of a first particle, but before the elapsing of the staging time. Put in other words, the probability of a pixel to be hit in a period of time equal to the staging time minus the average duration of a ToT computation. This is the same quantity as the losses due to the fixed deadtime in the pixels, and can be evaluated with Eq. A.1. By considering a 16 clock cycles staging time, and a variable ToT time, the probability is higher than 1% if the ToT of the first particle is smaller than 12, as shown in Fig. 4.9. Of course, in these considerations we're omitting the in-pixel pile-up, which is the situation in which a new particle strikes as the first is in ToT evaluation.

A bunch of latches has therefore to be added in the Pixel Logic: the Pixel FSM has to copy the value of the ToT counter to these latches before the pixel is reset. The ToT compression will then sample the ToT in the latches, and not that of the counters.

4.3.1.2 Staging Buffer

The staging buffer is one of the main innovations in this Pixel Region architecture. It postpones the ToT compression of a predefined time. Instead of a pipeline approach, which would require a huge number of Flip Flops, the staging buffer is composed of

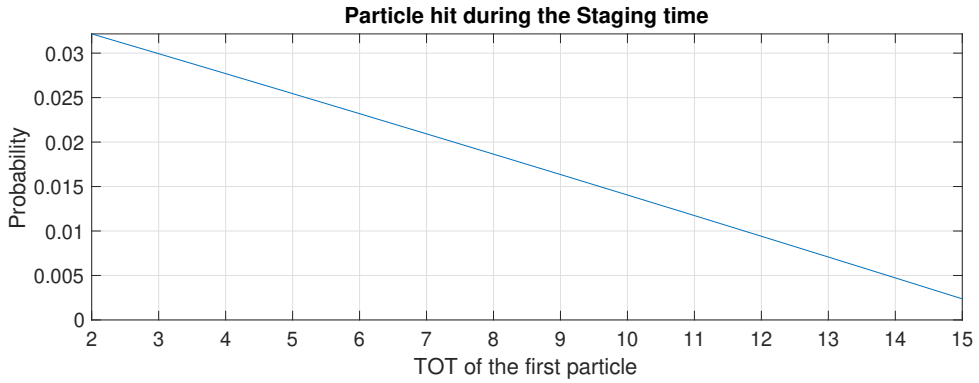

Figure 4.9: Probability of a Pixel to be hit during the staging time, for a 95 kHz pixel hit rate, 16 Clock Cycles Staging time.

a set of rows, each comprising a bunch of latches for the hitmap, and an associated counter for the staging time evaluation.

When any pixel of the Pixel Region is hit, the Staging Buffer saves the Hitmap of the event in a row, and starts its counter. The counter is also used to check for the row's states: if zero, the row is idle; otherwise, it's full.

The depth of the Staging Buffer is to be evaluated via Eq. A.1. It depends, of course, on the hit rate of the Pixel Region which, in turn, depends on its form factor. Preliminary analyses, shown in Fig. 4.10, highlight how 3 rows should assure event losses below 0.1% for most of the maximum ToT computation times in a 4×4 Pixel Region.

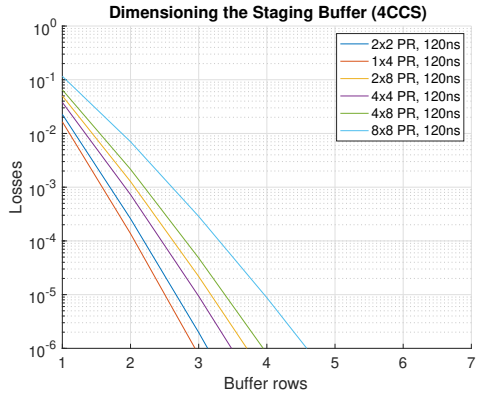

Figure 4.10: Losses related to Staging Buffer overflow for different Pixel Region sizes, minimum staging time, using internally generated hits in RD53A. The 4×4 Pixel Regions needs 3 Staging Buffer rows to have $< 0.1\%$ losses.

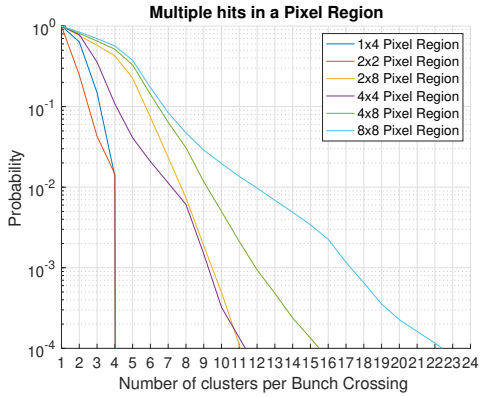

Figure 4.11: Probability of cluster sizes for different Pixel Region sizes, using internally generated hits in RD53A. The 4×4 Pixel Regions needs 8 ToT Slots rows to have $< 1\%$ losses.

Therefore, a Staging Buffer depth of 3 rows has been chosen. The output of the staging buffer consists in the delayed hit map, and a *valid* signal which can be used by subsequent blocks to trigger the sampling of delayed hit map.

4.3.1.3 ToT Compressor

The block in charge of the ToT compression is triggered by the *valid* signal from the staging buffer.

It reads the delayed hit map and the TOTs of the pixels corresponding in the hitmap itself, before going through the same compressor implementation from CHIPIX65. This compressor, in fact, was shown to be adapt for the application, and has therefore been implemented with only minor changes.

The number of TOTs stored has been increased to 8, as new analyses, depicted in Fig. 4.11, show that this number of TOTs can reduce the charge information loss to values below 1%.

4.3.2 Clock gating structure

In order to save as much power consumption as possible, the clock network has been re-analyzed from scratch, and a new hierarchical approach to clock gating implemented.

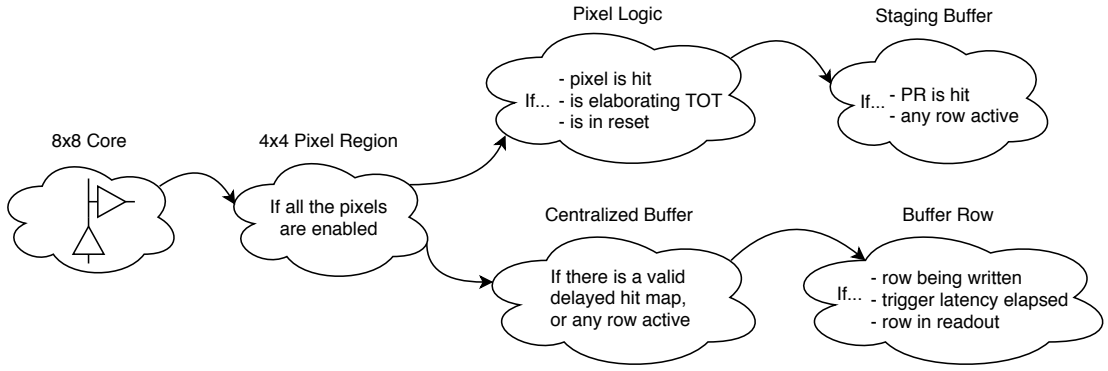

Figure 4.12: Clock gating hierarchy in RD53A's CBA Core

The 8×8 Pixel Core does not have any clock gating, as it is very unlikely for a whole Pixel Core to be inactive during operation. Conversely, the Pixel Regions can be inactivated if all of the Pixel Region's pixels are inactive, and thus the Pixel Region cannot process any hit.

Inside the Pixel Regions, both the Pixel Logics and the buffers can be gated. In particular, the Pixel Logic is mostly inactive during chip operation, which is why it is very important to gate its clock when it is masked or is not processing any hit.

The 2 buffers, staging and central, must be gated separately as their functionalities are very different. The Staging Buffer uses counters for operation, and thus is frequently active: the clock gating strategy which yielded the best results apply the gating to the single rows, and not to the whole structure as it is unlikely for the whole buffer not to have any counter active.

The central buffer, instead, is only active when any of its rows changes state. As there is no counter inside, but only latches and Flip Flops whose value change infrequently, the gating strategy applies both to the single rows, and to the whole buffer. This is especially important in big buffers such as the CBA one, in which it was clear how the row-by-row gating strongly impacted the gating performance.

4.3.3 Periphery

The periphery is organized so as to have 2 main blocks, which communicate directly with the Core Column and among themselves. They are the Column Read Control, replicated per Core Column, which handles the trigger tag buffering and readout from the Core Column; and the Data Concentrator, which buffers and sorts the data per trigger tag.

The Data Concentrator was designed to accept DBA packets, hence another block was required, which could split up CBA packets into DBA-like ones, effectively simulating the readout of a DBA column. This block was called *Output Adapter*.

4.3.3.1 Column Read Control

The Column Readout Control is controlled by a Finite State Machine which has 3 states: START, WAIT and DATA.

The START state is the idle state in which the FSM waits until new triggers arrive. If a new trigger arrives, the FSM switches to the WAIT state. The WAIT state is used to relax the timing constraints on the readout: the time needed for the readout of the Core Columns data packets is programmable, and can be progressively relaxed if the radiation effects slow down the digital logic. Conversely, it can be fastened up if the blocks behave correctly, in order to increase the bandwidth of the bus. After this WAIT state, in which the Cores maintain a steady output in the bus, the FSM switches to the DATA state, which triggers the sampling of the data in the bus into the Data Concentrator.

The common peripheral trigger logic keeps track of the trigger tags sent chip-wise. There are 32 possible trigger tags, which are sufficient for the chip operation at the foreseen trigger rate. Every Column Read Control block keeps track of the trigger tag it is reading out via a dedicated counter, which stops once it reaches the global counter (that is, when there are no more trigger tags to read).

Among the control signals to the Column Read Control block (the global trigger tag, the Token signal from the Core Columns, etc.), there is a Ready signal, which is originated from the Data Concentrator, in order to tell the Column Read Control whether

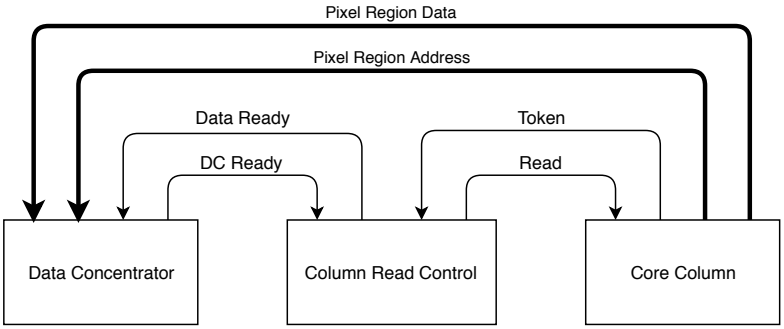

Figure 4.13: Main peripheral connectivity between Data Concentrator, Column Read Control and Core Column in RD53A's DBA Cores periphery.

it is currently capable of dealing with any further data from the corresponding Core Column. This message will be exploited by the Output Adapter to temporarily stall the readout.

4.3.3.2 Output Adapter

The Output Adapter block is used to convert the CBA packets into DBA equivalent ones. In doing so, it must be able to serve in different clock cycles the DBA-like packets to the Data Concentrator, as it is not capable of handling more than 1 packet at a time.

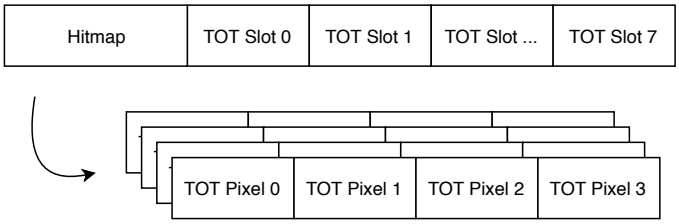

Figure 4.14: Packet translation between CBA and DBA data words

The Output Adapter has a dedicated FSM, which is used to replicate the behavior needed for operation. The goal of this implementation is to use the Output Adapter as an intermediary between the Data Concentrator, the Column Read Control, and the Core Column.

The Output Adapter intercepts the data from the Core Column and stores it in an internal FIFO. While it is splitting the CBA packet into the DBA counterparts, it stalls the readout of the Column Read Control by controlling its input Ready flag. The DBA-like packets are sent to the Data Concentrator along with a signal equivalent to the Data Ready produced by the Column Read Control.

Fig. 4.16 shows the Finite State Machine used for the Output Adapter implementation. Normally, the FSM is in the *IDLE* state, but when a packet arrives from the Core

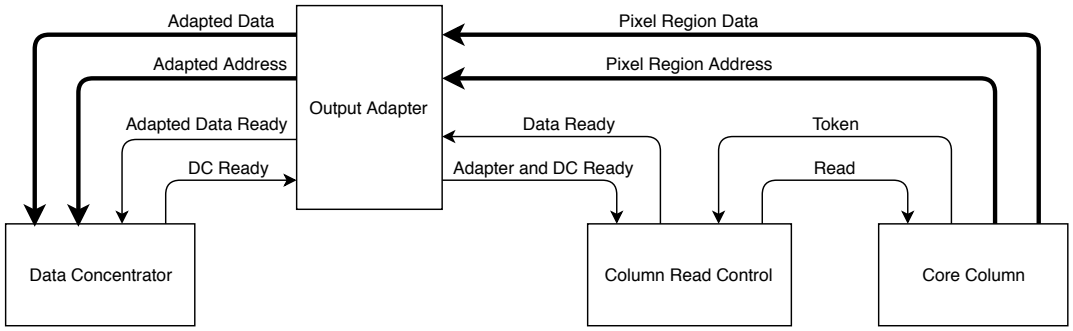

Figure 4.15: Main peripheral connectivity between Data Concentrator, Column Read Control, Core Column and Output Adapter in RD53A's CBA Cores periphery.

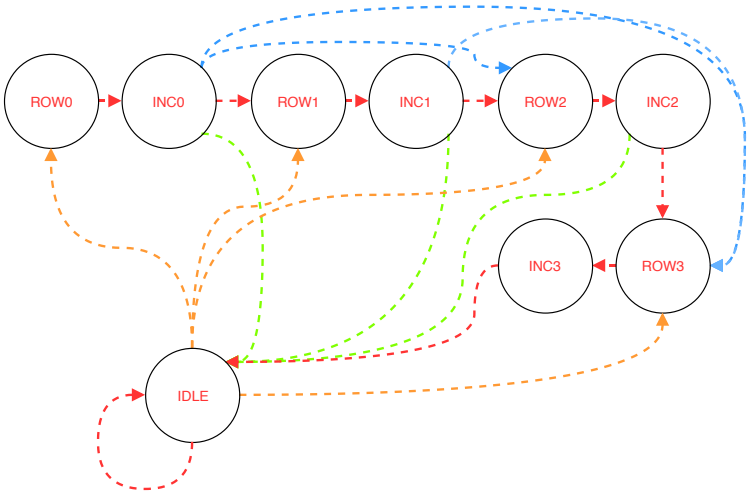

Figure 4.16: Statechart of the Output Adapter FSM

Column, it splits it combinatorially into the corresponding DBA equivalent. A CBA packet corresponds to 4 DBA ones: in order to suppress the zeroes, the *IDLE* state goes directly to the state corresponding to the first non-zero DBA-equivalent packet. Once it is processed, it goes to a temporary buffer state *INC*, which implements a pause needed by the Data Concentrator, and then switches to the next non-zero packet. When all the DBA packets have been sent, the FSM reverts to the *IDLE* state, and advances the FIFO read address in order to be ready to process the next CBA packet.

4.3.4 Verification

The CBA Pixel Core had to be modeled in the verification environment reference model for both the losses due the latency buffer overflow, and the staging buffer overflow. The simulations with the verification environment have been used to optimize the

design parameters, which are presented below in the final architecture version, along with the results the simulation with a 750 kHz cm^{-2} particle rate, 1 MHz trigger rate, and 12.5 µs trigger latency:

Parameter	Value
ToT bits	4
Latency Buffer depth	16
Staging Buffer depth	3
Maximum Staging time	32 CCs (5 bits)
ToT slots	8
Event loss	0.03%
Charge information loss	0.5%
Power	8.1 µW/pixel

4.4 Integration

The chip integration flow was inspired by the CHIPIX65 one, which followed a top-down digital-on-top approach. The chip floorplan is,however, much larger, as the matrix contains 192×400 pixels. The overall floorplan is shown in Fig. 4.1, while a functional view of the chip, with a closeup on the periphery, is shown in Fig. 4.17.

The wider chip with its load of analog blocks and digital areas represents a big challenge for the integration procedure, in particular with respect to the timing optimization of the digital areas.

Timing models In order to correctly characterize the behavior of the digital blocks after irradiations, a test chip called DRAD, developed in the context of the collaboration, had been submitted and its results analyzed. [14] In particular, the DRAD chip investigated a combination of 9 digital libraries made of different number of tracks (7, 9, 12, 18) and transistor threshold voltage flavors (low V_t, normal V_t, and high V_t).

In order to appropriately measure the timing performance degradation, the DRAD chip used delay chains and ring oscillator test structures on a set of pre-defined standard cells, which include:

- Inverters: INVD1 and INVD2
- NAND gates: ND2D1 and ND4D1
- NOR gates: NOR2D1 and NOR4D1

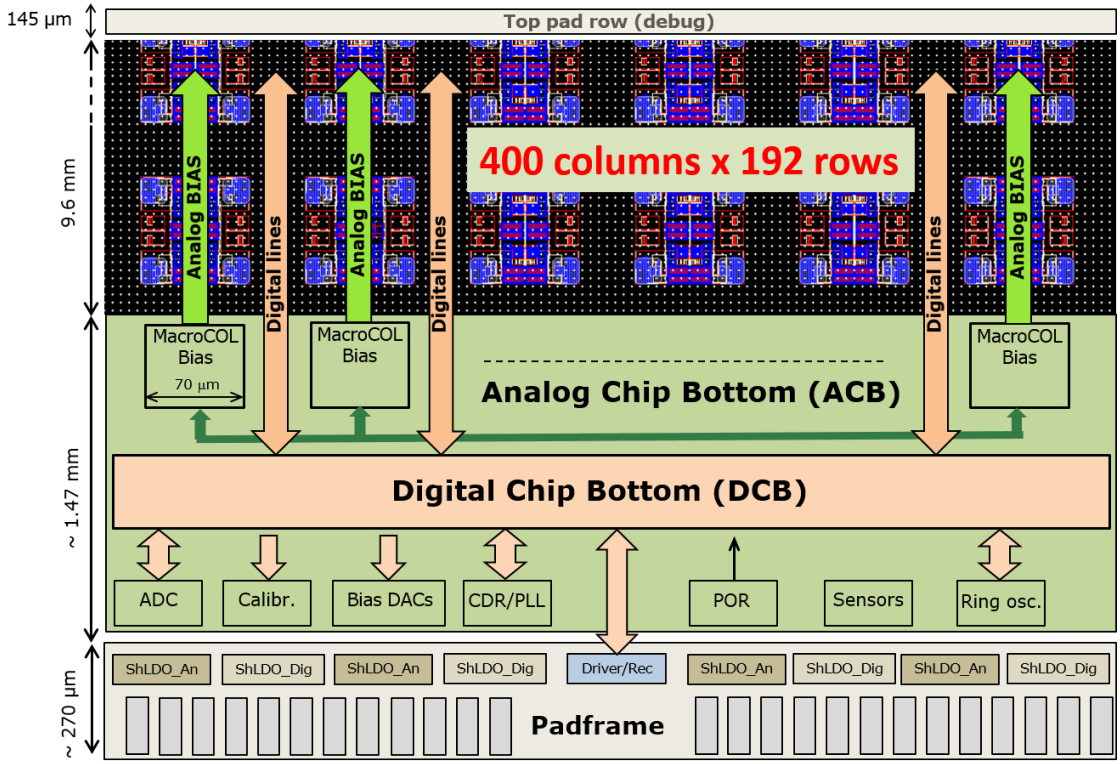

Figure 4.17: Block Diagram for the floorplan of the RD53A chip. [39]

- XOR gates: XOR2D1
- Clock buffers: CKBD1, CKBD4 and CKBD16
- Flip Flops: DFCNQD1
- Latches: LHCNQD1

The tests highlighted a degradation in the timing performances after 200MRad in the order of 10-15%, with a trend going from the worst case being High V_t, 7 tracks, and the best being Low V_t, 18 tracks. After 500MRad the trend remains, but goes from 40% worst case to 25% best case. The worst results have been obtained by the NOR gates, while the others were comparable.

4.4.1 Pixel Core

As explained earlier in this Chapter, the Pixel Core is the minimum synthesizable entity in the Pixel Matrix, and therefore all the synthesis and place and route optimization is performed on such block. The floorplan of the Pixel Core is similar to the one of CHIPIX65, but it comprises a 8×8 submatrix instead of a 4×4.

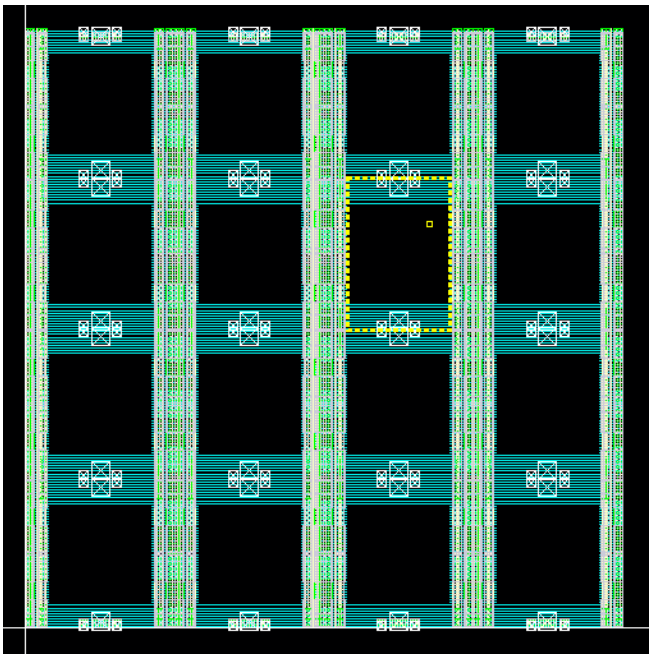

Figure 4.18: Floorplan of RD53A's Pixel Cores

In Fig. 4.18 the core is shown in its most basic floorplan: the analog islands can be seen, along with the vertical passways for the digital routing. Along these vertical pathways, the digital routing can be done from metal 1 to metal 6, while in the regions in between the analog islands, metal 6 is used to shield the analog bias lines passing on top, and thus digital routing can only be performed up until metal 5.

As metal 6 is a horizontal routing layer by default, this creates bottlenecks in the routing of the standard cells. In order to overcome this problem and ease the work of the routing engine, the placement of standard cells has been forbidden near the corners of the analog islands, and the direction of the routing layers has been inverted in the areas below the metal shields.

The I/O pins for the signals have been placed via a automated procedure at the bottom and top of the core, devised in a way to increase the space between the pins as much as possible (to avoid routing congestion) and ensure that when the Pixel Cores are abutted, the output pin is correctly connected to the corresponding input pin.

4.4.2 Matrix

The assembly of the RD53A matrix is implemented in a way very similar to that of the CHIPIX65 chip. The particularity of this chip, however, is that it features 3 different Front-Ends, along with their different bias lines and signals, and 2 digital architectures, with their different timing constraints and I/O pins.

One very smart idea, inherited from the FE65P2 chip, replaces the CHIPIX65 way of

assigning the Pixel Region address from the top through a black box in the Pixel Regions themselves, through a simple adder: the address is propagated from one Pixel Core to the next, but only after it's been increased of 1 unit. In this way, starting from the bottom, which assigns the first address, the following addresses are computed automatically through a network of static cells. This, although introduces a small routing and area overhead in the Cores, allows a drastic simplification of the address assignment.

4.4.2.1 Core Column timing closure

The CHPIX65 matrix was small enough as to not display any evidence of eccessive timing skew in the Pixel Regions due to the propagation delay of the signals. In RD53A, however, the matrix is much taller, which means that the signals need a much longer time in order to propagate through the buffers and lines from the bottom Core to the one at the top.

A novel way of correcting the timing skew in the Matrix has been proposed, stemming from the FE65P2 experience, and uses delay cells in order to introduce a delay in the signals inversely proportional to the Core distance from the bottom. A Propagation Delayer (a network of delay elements and multiplexers) has been devised, which introduces a fixed delay due to the muxes (called Offset), and a delay proportional to the Core Address (called Step).

With the standard cells identified, the contribution of each block depend on the timing corner used:

Delay	Minimum	Typical	Maximum
Offset	243 ps	391 ps	1.377 ps
Step	497 ps	732 ps	1.733 ps

This solution seems to be compatible with the model, and has a direct implication on the input delay of the cores, which becomes:

Corner	Minimum ID	Maximum ID
Minimum	2.5ns	6.5 ns
Typical	1.5 ns	6.5 ns
Maximum	1.3 ns	15 ns

This type of solution ensures that the performance degradation due to Total Ionizing Dose or manufacturing corners applies both to the signal propagation buffers and the compensation delays: they can balance out so that the effect of the first is much less impacting than it would otherwise be.

4.5 DAQ Setup and Tests

The RD53A chip was submitted in late August 2017. The first chips were ready for test in late January 2018.

There are 2 DAQ suites for the testing of RD53A: BDAQ53, and YARR. The first, available even before the chips were fabricated, has been extensively used for the first testing of the chips. Among the first tests, the digital scans were a complete success, and highlighted how the pixel matrix worked flawlessly for a series of multiple injections. The results are shown in Fig. 4.19.

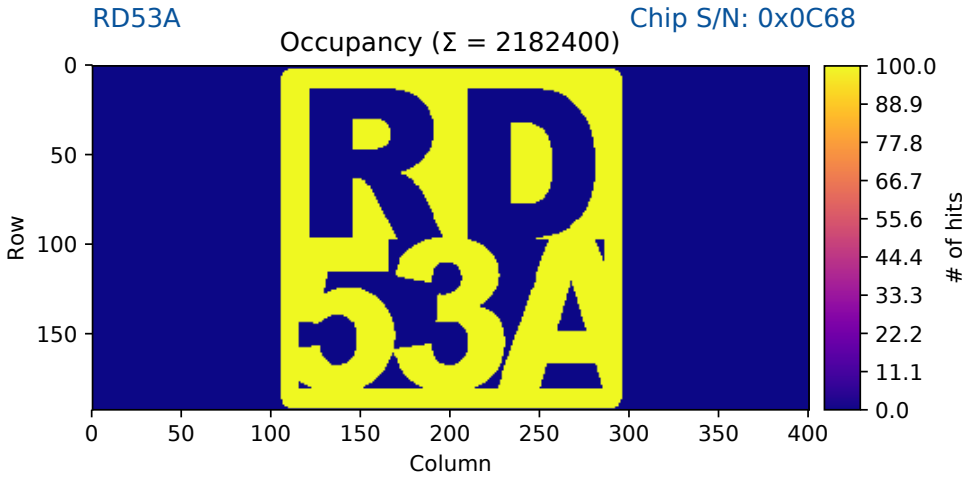

Figure 4.19: DAQ Occupancy scan of the RD53A Pixel Matrix masked to show the RD53A logo.

The tests of the RD53A reported great successes for the community, as the chip was found to be correctly working immediately, and the feature testing started soon afterwards. It was readily found that both buffering architectures work flawlessly, and that the cores are still functioning even beyond the 500 Mrad target.

4.5.1 Sensor Tests

The first sensors tests involved SOI3 sensors from the MPP (Max-Planck-Institut fuur Physik, Munich). Once the chips have been bonded, the obseverd yield was of about 70%. The batch produced 11 functioning chips, which confirmed that the whole matrix is fully functional.

The chips bonded with the sensors have been tested under a 2.5 GeV electron beam at ELSA, with the chips externally triggered by a scintillator. The direct measurement of the chip under the primary beam is shown in Fig. 4.20. Another measurement with the RD53A logo mask applied on the sensor backside is shown in Fig. 4.21.

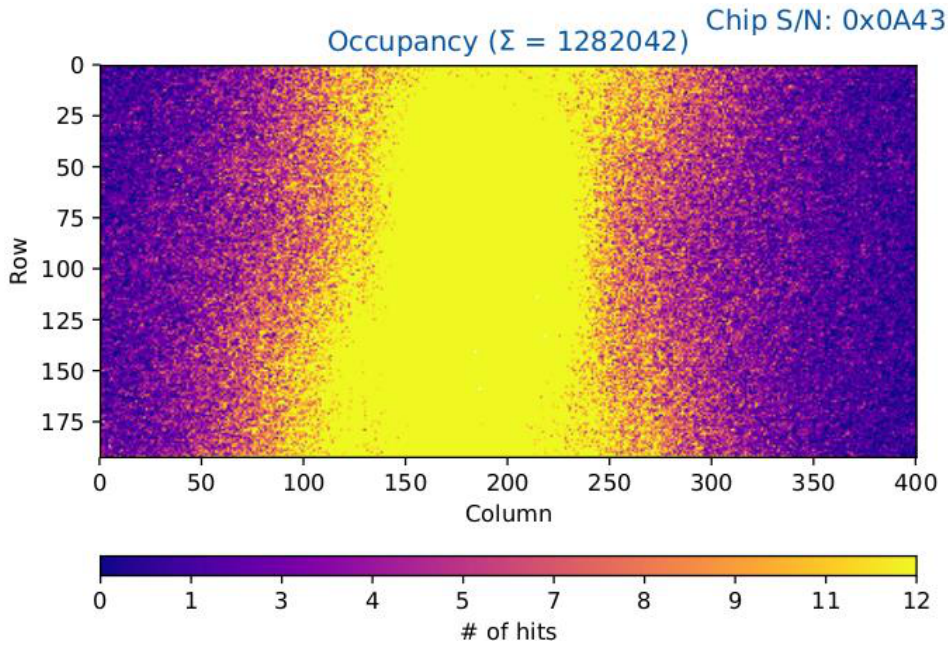

Figure 4.20: Sensor test of the RD53A chip at the ELSA accelerator. This Occupancy Scan shows the response of the RD53A pixels hit by the electron beam, and thus the position of the beam with respect to the chip.

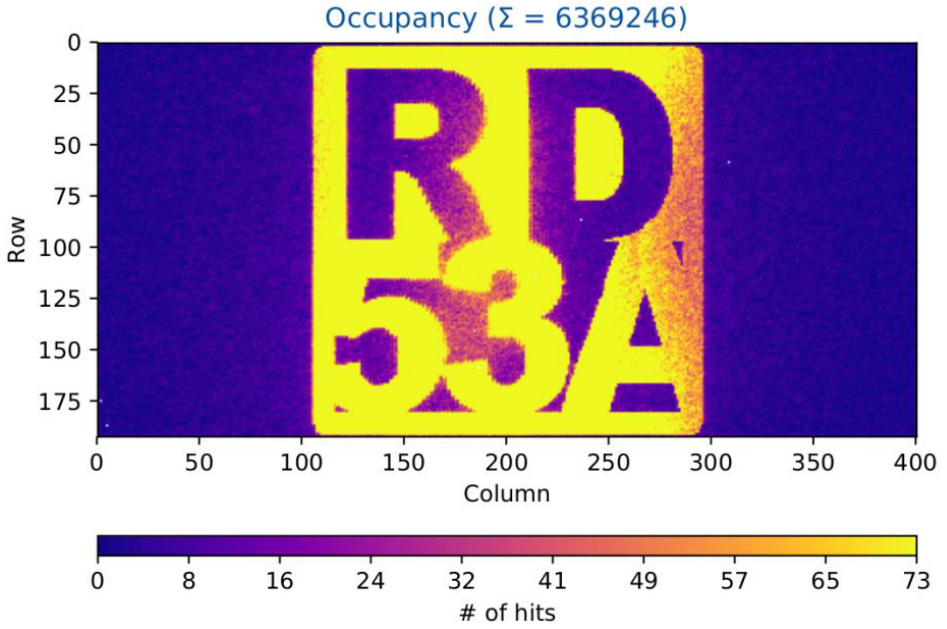

Figure 4.21: Sensor test of the RD53A chip at the ELSA accelerator, with a RD53A logo mask applied on the sensor backside.

4.5.2 Readout problems

During the testing of the chips, however, some readout oddities were reported. In particular, some of these were present only in the CBA architecture, and are due to small but impacting design bugs in the Chip Periphery. These bugs had not been found by the Verification Environment, as it did not contain the tests necessary to replicate the faulty behavior, and are only triggered in certain testing conditions: the threshold scans, and the full matrix injection tests.

4.5.2.1 Readout stuck

During the calibration scans of the Synchronous Front-End, it was discovered that as the injected signal voltage approached the threshold value, at some point the pixels would stop sending their output. The same situation was soon found to apply also to threshold scans, as the threshold started rising from low levels to approach the injected signal voltage. The problem did not depend on the pixel (or cluster of pixels) injected, but solely on the proximity of the injected charge and the threshold.

The problem was shown to be more common if the supply voltage to the chip was purposedly set to lower values (1.1 V against the nominal 1.2 V), and could be overcome if a reset was sent to the readout block, although this would mean losing all the events buffered in the readout chain. A final test showed how the readout stuck only applied to the single Core Column: subsequent injection with charges higher than the threshold in other Core Columns worked flawlessly.

All these clues pointed at the Output Adapter, where the culprit was found in the Adapter FSM: when the CBA packet is split into the corresponding DBA-like equivalents, the FSM advances till the first non-zero DBA packet.

The CBA pixel architecture expects the Synchronous FE to produce pulse with duration greater or equal to 25 ns whenever hit. This means that the CBA output packet will contain at least a ToT greater than 0.

The Output Adapter will transform the CBA Hitmap and ToT slots into the DBA equivalents by reverse assignment, and the FSM will look for any DBA equivalent whose ToT bits are different from zero in order to send them to the Data Concentrator.

The problem lies in the implementation of this behavior: the circuit expects a CBA packet with a valid ToT (greater than 0), if the corresponding pixel has been hit. If this doesn't happen, and the pixel ToT is 0, the circuit classifies it as a non-hit pixel. As it happens, the Synchronous FE in near-threshold situations can produce pulses with duration lower than 25ns: these will be recorded correctly in the hitmap, but with ToT equal to 0 in the corresponding slot.

If the CBA packet contains only non-hit pixels and pixels with ToT equal to 0, the FSM doesn't know which DBA equivalent to proceed to and gets stuck in the *IDLE* state. This is caused by a missing failsafe condition in the FSM implementation.

This problem could not be found in regular digital simulations as the model for the Synchronous FE could not replicate the kind of behavior thought to introduce the

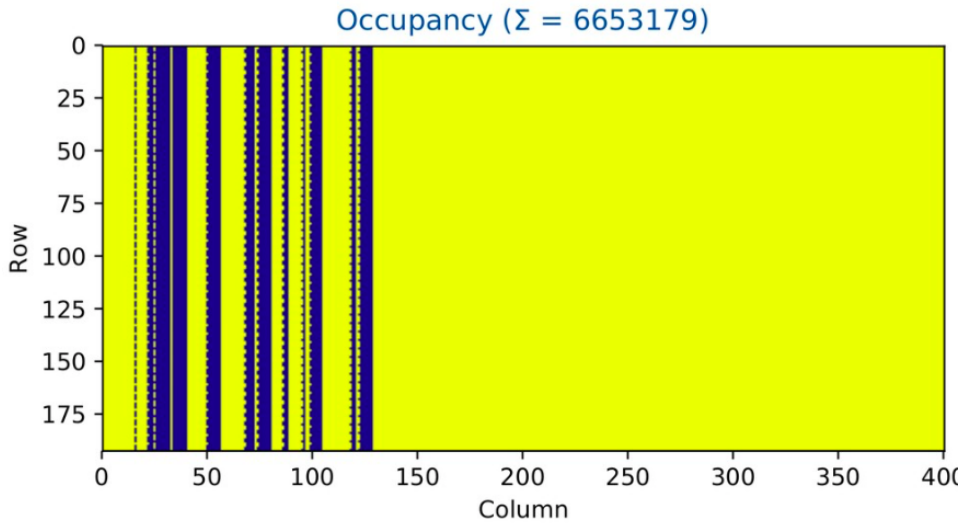

Figure 4.22: DAQ Occupancy scan of the RD53A chip showing some CBA Columns stuck

problem.

However, it was possible to replicate the problem with a particular setup: the FE was set to operate in Fast ToT mode, but the ToT clock was set to a very low value, lower then 40 MHz. With a digital injection (which works flawlessly as it can only produce pulses $\geq$25 ns), a ToT=1 pulse would, with such slow ToT clock, produce a ToT=0 in the pixel. This kind of packet could effectively stuck the readout.

The problem could not be found in mixed signal simulations either, as the problem lies in an error propagating from the pixel interface with the Pixel Core, to the Output Adapter. Such extensive mixed signal simulations are hardly feasible as they require an enormous amount of computing resources.

Although the culprit has been found in the FSM implementation, it was not possible to reliably replicate the faulty behavior in the pixels in simulations.

Analog tests In order to successfully and thoroughly check the performance of the Synchronous FE, it was fundamental to devise a readout scheme in which the impact of this bug was suppressed or minimized.

A particular readout strategy takes advantage of the positive effect of the reset, which unlocks any stuck state in the Core Column, and the fact that the stucking applies only to a single Core Column. In this proposed solution, which was applied to correctly characterize the Front-End, it is possible to rapidly scan the entire Core Columns without losing the data from any pixel.

The strategy can be summarized in the following steps:

- Select the Core Columns to scan

- Inject all (or a subset) of the pixels in one Pixel Region per Core Column
- Trigger the event for readout
- Wait until the data is readout, then reset the Chip Periphery
- Repeat from step 2 until all the pixels under scrutiny have been scanned

The key modification with respect to the scan strategy employed for the other architecture is that the pixels under scrutiny have the constraint of belonging to the same Pixel Region. The bug, in fact, applies to the entire Pixel Region: either a CBA Pixel Region packet is correctly processed, or it stucks the readout. The bug conformation does not allow for partial Pixel Region readout. By injecting one entire Pixel Regions per Core Column, the scan speed can be quite high (although not possibly as high as it is possible for the other architecture), without any loss of information.

4.5.2.2 Whole Matrix pixel loss

Another problem was found during whole-matrix digital injections, which showed the loss on some pixels in the CBA area. The losses seemed not to follow a clearly understandable pattern, but could be reliably reproduced with the exact same behavior.

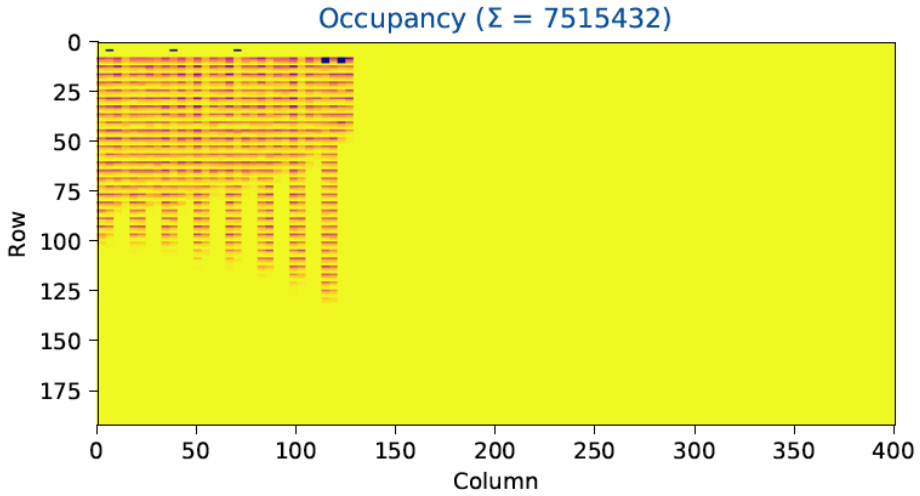

Figure 4.23: DAQ occupancy scan after whole matrix injections

Follow-up digital simulations were found to show reproduce the same behavior. Such tests were unfortunately not implemented by the verification team during the design, and thus this problem was not found until late.

The culprit was, again, found in the Output Adapter, in particular in the interface with the Data Concentrator. The implementation of this Adapter/DataConcentrator interface was developed along with the Data Concentrator designers, but unfortunately

a misinterpretation of the Data Concentrator Busy signal made possible to send, under certain conditions, a DBA packet to the Data Concentrator even when the Data Concentrator column FIFO is full.

The problem is exacerbated by a bug in the Data Concentrator FIFO, whose physical depth is of 16 rows, but only 15 are used in reality. Due to this bug, the Adapter only checks the Data Concentrator Busy flag only at the start of the CBA-to-DBA packet conversion, and ignoring any changes in the Data Concentrator Busy flag until the next CBA packet arrives: any DBA packet sent after the Busy flag rises, are therefore lost.

4.6 Summary and Future work

The experience in the RD53A design were key in the development of the proper 65nm skillset for HL-LHC pixel detectors. Many of the requirements were satisfied, along with many of the requested features.

The CBA digital architecture, in particular, integrated a whole new concept into the Pixel Matrix of a ROC for the first time, with the same performances as for the other architecture, but with a smaller area footprint. The next steps would be to tradeoff this unused area with more features, and bring the architecture to the RD53B chip.

Unfortunately, even at the moment of submission, the verification environment and verification procedures lacked the means to check some of the bugs which would later be discovered during operation. These bugs were promptly fixed as soon as they were first noticed, but there was no way to correct them in the chip.

The groundwork made for RD53A has still room for optimization in the power consumption, AFE integration, and digital performance. Such improvements have been integrated into the development for the RD53B chip.

Chapter 5

RD53B

The experience gained in RD53A was invested in the development of a final common chip for both the ATLAS and CMS pixel detectors. However, as the module characteristics of the 2 detectors were different, eventually it was decided to realize 2 chips with similar, although not identical features. The main differences are highlighted in the following table.

	RD53A	RD53B - CMS	RD53B - ATLAS
Matrix height	192	384	328
Matrix width	400	400	440
Trigger	L1 only	L0 with L1 support	

The duties of the author with respect of the RD53B chip development lied primarily in the final architectural studies and the digital chip integration, and have been thoroughly described in this Chapter.

5.1 Architectural studies

One significant part of the research for the final RD53B chip has been the placed into the digital architectures development in order to define their performances and take a decision on the one to be adopted for the chips.

In this context, the availability of new physics simulations of the foreseen CMS pixel detector, were key for a thorough architecture analysis. In this section, a variety of architectures is explored, along with some considerations regarding the area occupancy.

As can be seen in Fig. 5.1 and Fig. 5.2, the hit patterns and cluster sizes, which have a direct implication on the hit rates of Pixel Regions, varies largely, depending also in the part of the detector the module is placed in.

As the hit rate has a direct effect on the amount of data the Pixel Region has to buffer, some considerations have to be done on the possible ways to encode the data. The event buffer, in fact, can account for as much as half the total area in the Pixel Region, and this makes it one of the key parameter in the dimensioning the Pixel Region itself.

5.1.1 Distributed

The Distributed Buffering Architecture encodes the ToT of an event by employing a direct mapping between the ToT memory slots and the pixels in a Pixel Region. In this way, the non-hit pixels in an event are stored with a ToT code equal to 0.

Table 5.1 shows the number of bits per pixel achieved by the DBA according to various pixel regions form factors and sizes.

The total number of memory cells for the distributed architecture can be thus evaluated as:

$$B = (B_{\text{Timestamp}} + N_{\text{Pixels}} * b_{\text{ToT}}) * N_{\text{Buffer Rows}} \tag{5.1}$$

By using the minimum number of buffer rows needed to achieve $< 1\%$ losses, as computed in Table 5.1, it is possible to calculate the total number of memory cells for every possible pixel region size and form factor, as shown in Fig. 5.3. In this case, as in the following, the number of ToT bits is 4, and the number of Timestamp bits is 9.

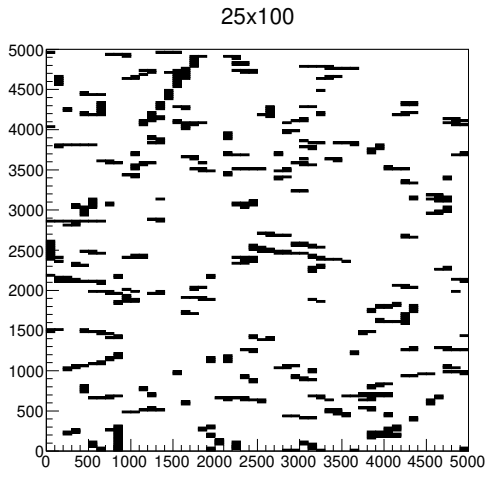

Figure 5.1: Superposition of 10 bunch crossing events at the center of barrel in the CMS simulation

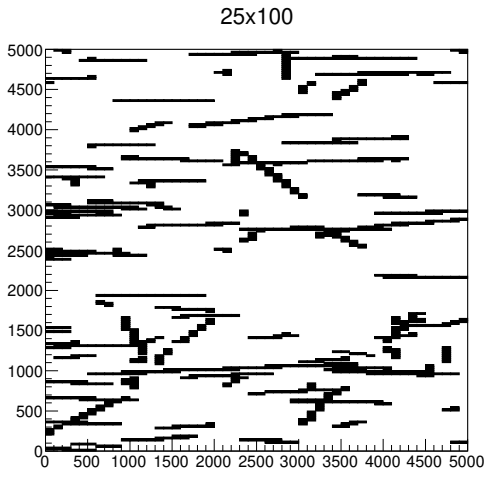

Figure 5.2: Superposition of 10 bunch crossing events at the edge of barrel in the CMS simulation

	Center of Barrel		Edges of Barrel	
	Hit Rate	Locations needed per pixel	Hit Rate	Locations needed per pixel
Pixel	93.94 kHz	4.00	84.90 kHz	4.00
2 × 2 PR	224.62 kHz	1.75	243.52 kHz	2.00
4 × 1 PR	312.43 kHz	2.25	304.94 kHz	2.25
1 × 4 PR	232.58 kHz	2.00	185.84 kHz	1.50
4 × 4 PR	602.73 kHz	0.94	542.15 kHz	0.81
8 × 2 PR	739.53 kHz	1.06	864.64 kHz	1.19
2 × 8 PR	619.04 kHz	0.94	428.40 kHz	0.69
8 × 4 PR	1089.44 kHz	0.72	1008.91 kHz	0.69
4 × 8 PR	1009.44 kHz	0.69	720.10 kHz	0.53
8 × 8 PR	1777.49 kHz	0.53	1294.34 kHz	0.41

Table 5.1: Hit rates and buffer locations needed by the Distributed buffering scheme in various Pixel Region sizes and form factors

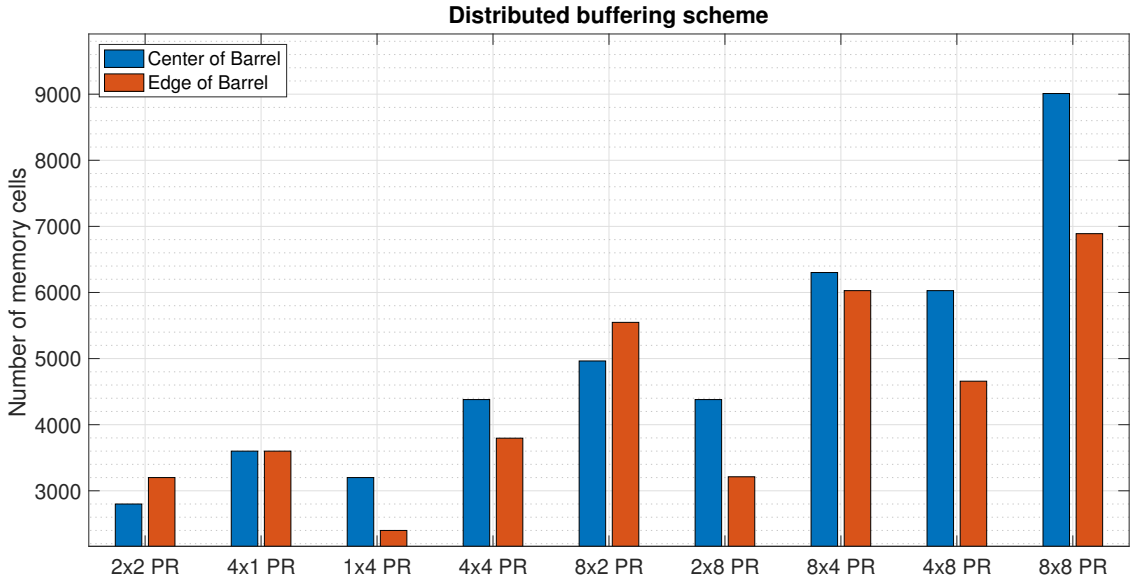

Figure 5.3: Number of memory elements needed to implement the Distributed buffering scheme in various Pixel Region sizes and form factors, repeated to fill a 8 × 8 Pixel Core

5.1.2 Hitmap

The Hitmap encoding scheme is used by the Centralized Buffer Architecture in the RD53A prototype, and uses a hitmap field to record the binary information on whether the corresponding pixel has been hit or not, and a number of ToT slots assigned to the hit pixels according to a priority queue.

The depth of the buffer is the same as for the distributed approach, as the event buffer depends on the Pixel Region hit rate, and not the number of pixels hit per event, which instead directly influences the number of ToT slots. By keeping a 1% efficiency threshold, the number of ToT slots needed can be evaluated via simulations. The results are shown in Fig. 5.4.

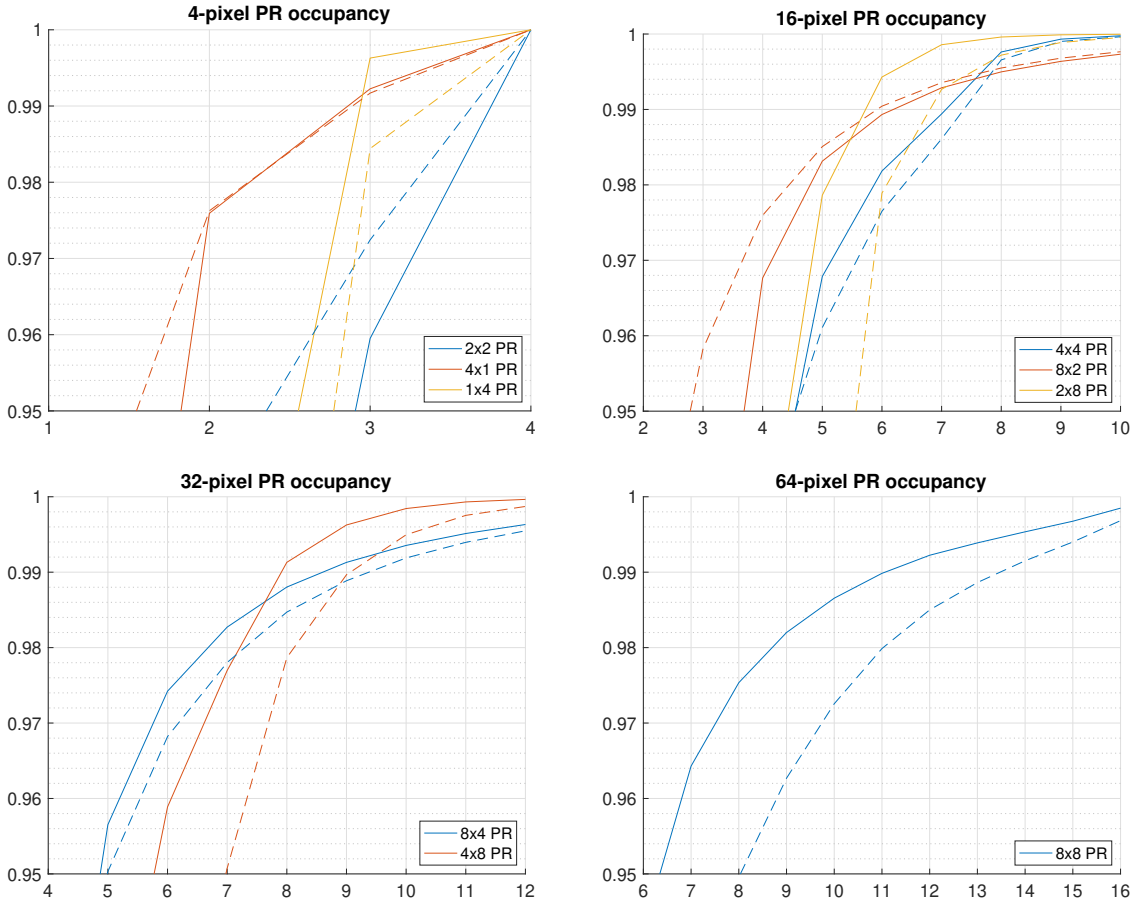

Figure 5.4: Efficiency versus Number of ToT Slots in various Pixel Region sizes and form factors

Table 5.2 shows the minimum numbers of ToT Slots to be used in Pixel Regions architectures if the ToT storage efficiency must be higher than 99%.

The total number of memory cells for the hitmap architecture can be evaluated as:

PR	2×2	4×1	1×4	4×4	8×2	2×8	8×4	4×8	8×8
Center	4	3	3	8	7	6	9	8	12
Edge	4	3	4	8	6	7	9	8	14

Table 5.2: Number of ToT Slots needed in various Pixel Region sizes and form factors at the Center and Edge of barrel

$$B = (B_{\text{Timestamp}} + N_{\text{Pixels}} + N_{\text{ToT Slots}} * b_{\text{ToT}}) * N_{\text{Buffer Rows}} \tag{5.2}$$

In order to evaluate the number of memory cells for this architecture, it is possible to employ the same principle as for Fig. 5.3: the number of buffer rows is extracted from Table 5.1, while the number of ToT Slots from Table 5.2. Fig. 5.5 is obtained by using Eq. 5.2.

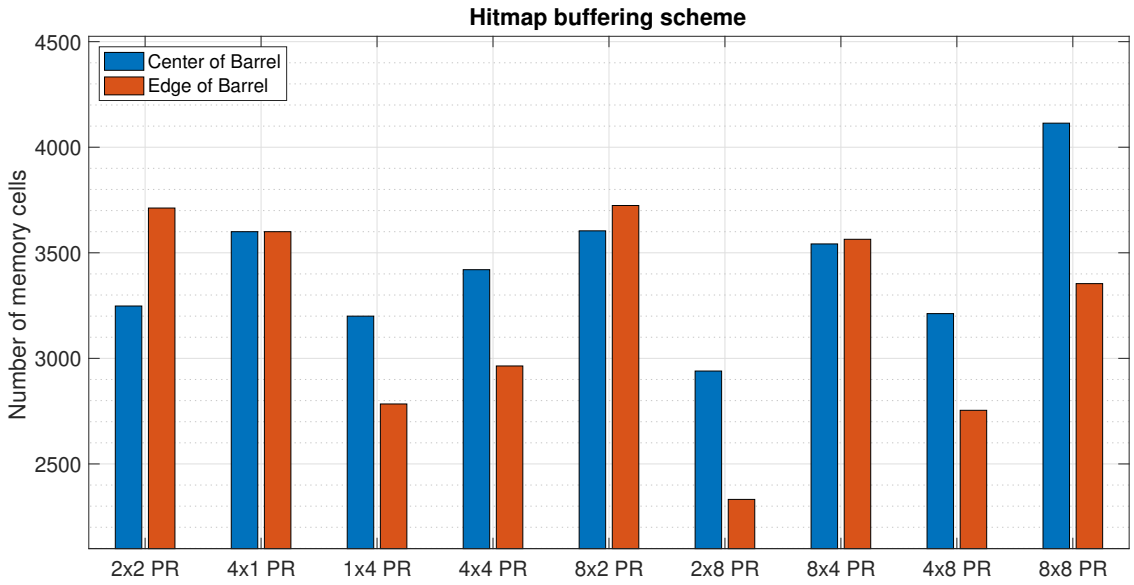

Figure 5.5: Number of memory elements needed to implement the Hitmap buffering scheme in various Pixel Region sizes and form factors, repeated to fill a 8×8 Pixel Core

The key aspect is this architecture, however, is not the number of latches, which is kept quite constant throughout the various options, but the depth of the buffer itself. The lower the number of buffer rows, in fact, the lower the number of timestamp comparators, row finite state machine, row writing and readout logic, and output data multiplexers. Another factor to be taken into account is that the necessary logic to *compress* the ToTs into the ToT Slots is not negligible in terms of area or timing requirements.

5.1.3 Pointers to Pixels

More complicated solutions involve the use of pointers in the memory. In this way, the pixel region can have 2 different buffer types: a central one, and a per-pixel one. If the memory holding the pointers is the central one, which points to the pixel buffers' locations, we are referring to a "Pointers to Pixels" architecture. The structure of such an approach is depicted in Fig. 5.6.

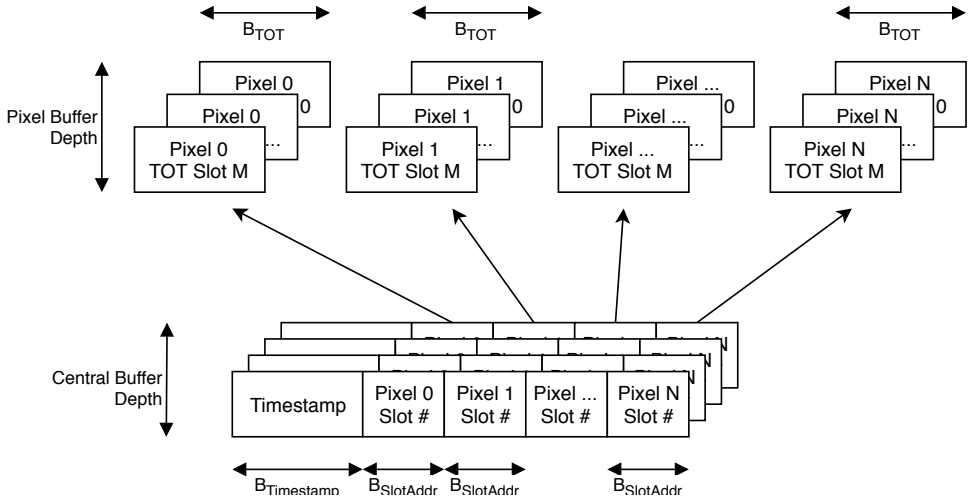

Figure 5.6: Scheme of the memory elements in the Pointer to Pixel buffering architecture

The total number of memory cells needed for this approach can be evaluated as:

$$B = \left(B_{\text{Timestamp}} + N_{\text{Pixels}} * B_{\text{Slot Address}}\right) * N_{\text{Central Buffer Rows}} + N_{\text{Pixels}} * B_{\text{ToT}} * N_{\text{Pixel Buffer Rows}}$$

$B_{\text{Slot Address}}$ is the base-2 logarithm of the number of ToT slots in the pixels. These, in turn, must be enough to keep losses low ($\leq$ 1%): this value can be extracted from Table 5.1. The number of memory cells for this architecture is shown if Fig. 5.7.

The logic that drives this architecture is not trivial: a two-way communication between the distributed and central memories is needed, in the form of a demultiplexer of the ToT slot for every pixel, and a quite complex multiplexing logic that selects the ToTs stored in the pixels for readout, when triggered. It should also be noted that the ToT Slot address in the central memory needs also to encode for an "empty" state, in case the corresponding pixel was not hit in the recorded bunch crossing. This is better schematized in Fig. 5.8.

Among the advantages, the fact that the pixel storage logic is independent from the central one, and does not cause any dead-time on it.

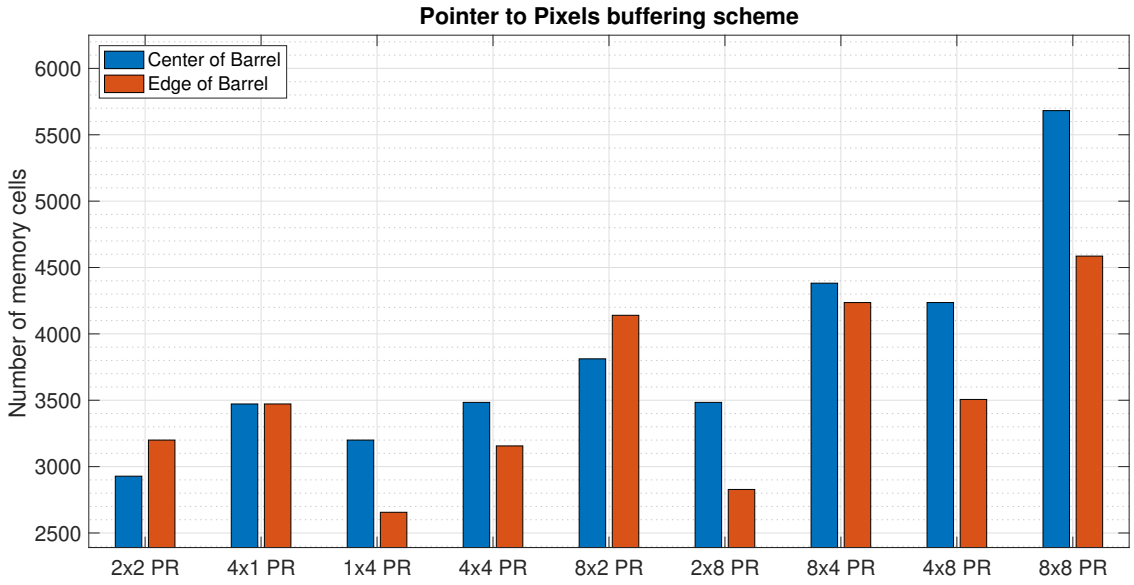

Figure 5.7: Number of memory elements needed to implement the Pointer to Pixel buffering scheme in various Pixel Region sizes and form factors, repeated to fill a 8×8 Pixel Core

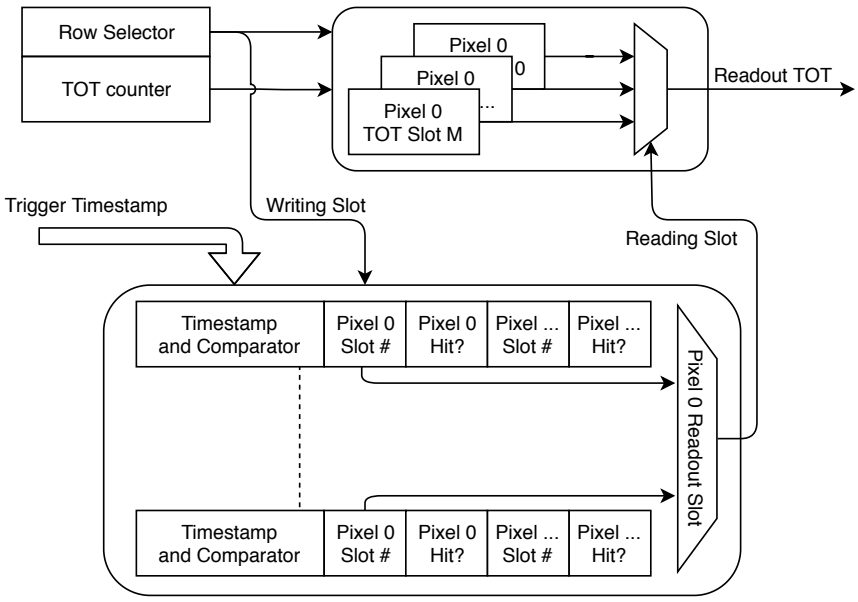

Figure 5.8: Block diagram of the Pointer To Pixel buffering scheme

5.1.4 Pointers to Shared

A variation of the ”Pointers to Pixels” is the ”Pointers to Shared”: an architecture that keeps distributed memories for the ToTs, which also embed a pointer to the central memory storing the timestamp information. Such scheme is depicted in Fig. 5.9.

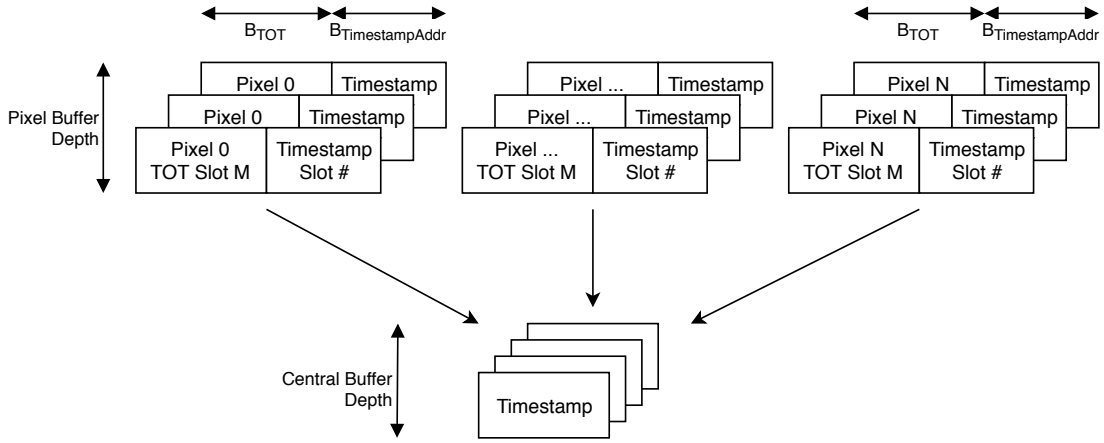

Figure 5.9: Scheme of the memory elements in the Pointer to Shared buffering architecture

One advantage of this variation is the reduced number of bits used as pointers. The total number of bits needed to encode the necessary information in this architecture is given by Eq. 5.3.

$$B = B_{\text{Timestamp}} * N_{\text{Central Buffer Rows}} + N_{\text{Pixels}} * \left(B_{\text{ToT}} + B_{\text{Central Address}}\right) * N_{\text{Pixel Buffer Rows}} \tag{5.3}$$

By comparing the pointer part with the ”Pointers to Pixels”, in order for this architecture to be more efficient, it must be:

$$N_{\text{Pixels}} * N_{\text{ToT Slots}} * \log_2 N_{\text{Central Rows}} < N_{\text{Central Rows}} * N_{\text{Pixels}} * \log_2 N_{\text{Pixel Rows}} \tag{5.4}$$

Given that the number of ToT Slots should be 4 in order to minimize the losses (< 1%), the equation becomes:

$$\log_2 N_{\text{Central Rows}} < N_{\text{Central Rows}} * \frac{1}{2} \tag{5.5}$$

This is always true except for $N_{\text{Central Rows}} = 3$. As $N_{\text{Central Rows}}$ is usually much greater, this architecture increases the bits efficiency. This is also highlighted in the comparison graph shown in Fig. 5.10.

In this architecture, the pixel event entries must be cleared alongside the central entry they refer to.

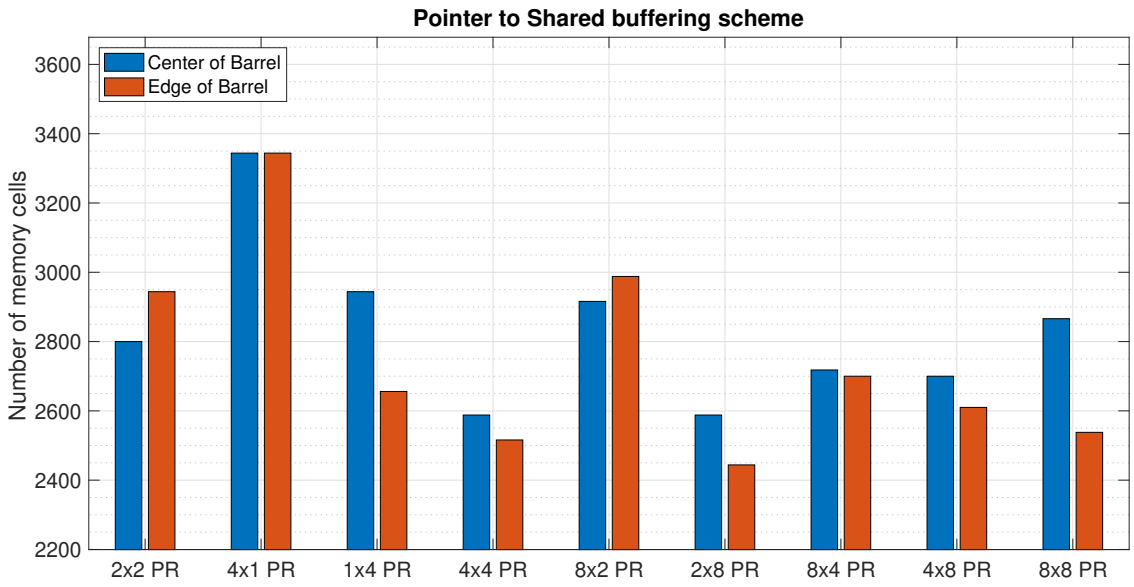

Figure 5.10: Number of memory elements needed to implement the Pointer to Shared buffering scheme in various Pixel Region sizes and form factors, repeated to fill a 8×8 Pixel Core

5.1.5 Architecture comparison

The plots in this section are useful for the detailed study of the Pixel Region sizing and shape, but for a proper comparison, the worst case of each Pixel Region must be chosen (between the one in the center of barrel or the edge of barrel), and a proper estimate of the additional logic must be added.

This additional logic must comprise:

- Driving logic
  - Writing row selector
  - Row finite state machine
- Reading logic
  - Timestamp comparator
  - ToT multiplexer

The writing row selector can either be implemented with a counter and a demultiplexer or combinatorially. As the counter solution is SEU-sensitive, if the timing path is not critical, the combinatorial solution is preferred. Its impact in cell number is approximately equal to 2 times the number of rows to be addressed.

The row finite state machine must encode for about 4 states. As this logic is sequential, it needs to take into account the clock tree propagation, and the flip flops themselves which are rather bulky. Its impact can be estimated in 20 latch-like cells.

The timestamp comparator is a combinatorial network, approximately comprising 3 times the number of timestamp bits.

The ToT multiplexer strongly depends on the type of architecture. In a first approximation, it can be estimated that its impact is 2 times the number of ToT bits, per ToT slot.

The additional logic can be superimposed to the memory cells themselves, and thus yield an estimate, in latch-like cells, of the total area of each buffer architecture. Fig. 5.11 details this total number of cells for the worst-case hit rate for every architecture and Pixel Region size. For direct comparison, the results were scaled in each Pixel Region in order to fill a 8×8 core.

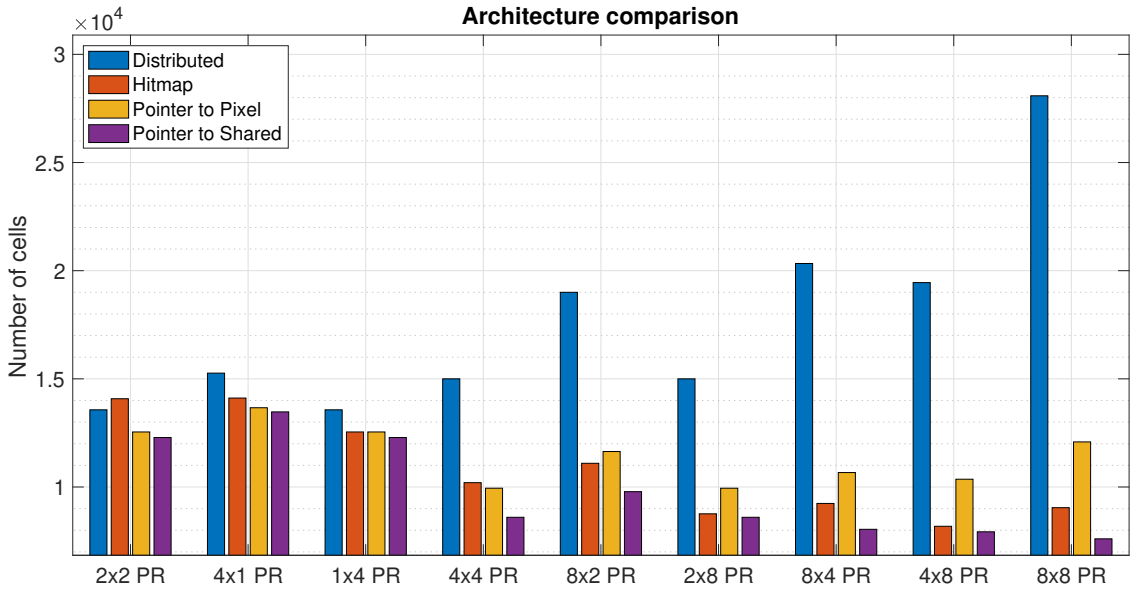

Figure 5.11: Estimated number of cells needed to implement the various buffering schemes for various pixel region sizes and form factors, repeated to fill a 8×8 core.

Fig. 5.11 gives a clear indication on which architecture gives the best result in terms of buffer area. Given that the results are scaled to fill a core, the search must be oriented on the architecture which gives the least area result. It is evident that the shared architectures yield a better result, especially when they can span across more and more pixels.

One aspect which is not easily inferable is the amount of routing that shared architectures require: this will be investigated by place and route trials. The simulations show how a 4×8 Pixel Region with Hitmap implementation can potentially be the best

solution for the pixel chip, as the pointer architecture may be tricky to debug, and may be less SEU-hard.

The CBA development in this chip has therefore focused on the 4x8 Pixel Region with Hitmap buffering. The contribution of the staging buffer to the number of cells, not highlighted in Fig. 5.11, will also be studied in detail.

5.2 An Improved CBA

Having defined the buffering scheme, the work of the author has focused on the logic, in order to further reduce the losses and power consumption. Moreover, as the CBA architecture had more available area, most of the new functionalities to be implemented in the Pixel Matrix have first been integrated in the CBA, and only then copied to the other.

The proposed new Pixel Region arrangement in a 4×8 structure featured the best performance/area ratio and tackled the issues regarding the charge information losses. The expansion to 32 pixels increased the foreseen Pixel Region hit rate from 600 kHz to 1 MHz: the Shared Buffer depth had to be therefore increased from 16 to 23 rows.

Among the innovative features there are an additional temporary ToT memory in the pixels, in order to make them capable of sustaining multiple hits during the staging time, and the support for the ATLAS-like trigger. The buffering procedure has also been modified in order to allow events to be written in 2 buffer rows, and thus double the number of available TOT slots.

5.2.1 ATLAS-like Trigger

Among the requests for the RD53B chip is the support of a double trigger scheme, needed by the ATLAS experiment. Its implementation needs some modification on the tag-based trigger readout. In the new use case, the chip waits for 2 triggers: a preselection trigger L0, and a confirmation trigger L1. In order for an event to be selected for readout, it must both be pre-selected and then confirmed.

The implementation of this scheme depends strongly on the readout operation. The simplest implementation involves the propagation of both L0 and L1 trigger signals to the Pixel Regions. The tag-based FSM provides an alternative way to implement the ATLAS trigger scheme. The periphery could propagate the L0 trigger to the Pixel Regions, and perform the L1 trigger matching in the periphery, by *clearing* the L0-triggered data in the matrix. In order to do so, the periphery must keep track of the timestamps and the assigned trigger tags, and then clear all the L1-untriggered events back to the *Empty* state, instead of reading them out. The event row FSM must be modified in order to support this transition as shown in Fig. 5.12.

This ”clear” feature also comes in handy in cases when an oversize event is being downloaded, and the user wants to stop the readout before it fills the peripheral buffers,

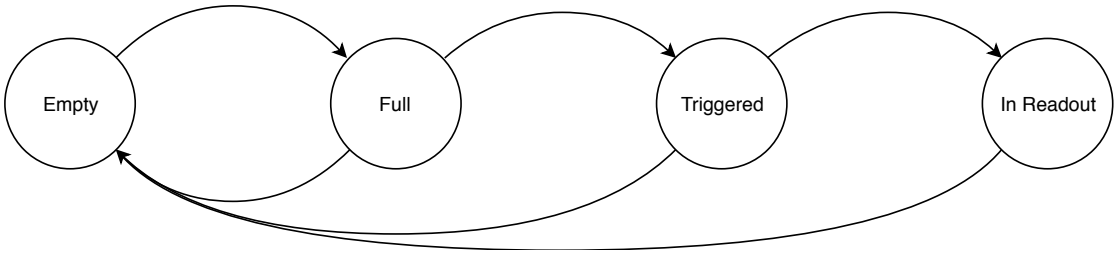

Figure 5.12: Statechart of the FSM needed to implement the ATLAS-like triggering scheme

preventing further events from being processed. For this reason, this implementation can also be implemented in the CMS chip at practically no cost, as an additional readout feature.

The buffering requirements of the ATLAS Triggering scheme is not trivial to evaluate. In order to do so, a high level simulator has been used to model a buffer with constant input rate, and 2 selection triggers. If the L0 trigger signal (simulated at a fixed rate) does not arrive after the L0 trigger latency, the row is dropped. Otherwise, the row is retained until the L1 trigger latency, when it is discarded. A similar structure modelling a single selection trigger has been used to validate the simulation results. The simulation results are shown in Fig. 5.13 for a buffer with an input rate of 1.1 kHz (Hit Rate of a 4×8 Pixel Region), and in Fig. 5.14 for a buffer with an input rate of 1.8 kHz (Hit Rate of a 8×8 Pixel Region).

The figure also picture the buffer usage evaluated statistically through Eq. A.1: the Poissonian distribution in the CMS case has a $\lambda = \mathrm{L1}_{\mathrm{Latency}} * \mathrm{Hit}_{\mathrm{Rate}}$, while for the ATLAS case it was:

$$\lambda = (\mathrm{L0}_{\mathrm{Latency}} + \mathrm{L1}_{\mathrm{Latency}} * \mathrm{L1}_{\mathrm{Probability}}) * \mathrm{Hit}_{\mathrm{Rate}} \tag{5.6}$$

with:

$$\mathrm{L1}_{\mathrm{Probability}} = \mathrm{L1}_{\mathrm{Rate}}/\mathrm{Hit}_{\mathrm{Rate}} \tag{5.7}$$

From Fig. 5.13 and Fig. 5.14, it is clear that the buffering requirements for the ATLAS Triggering scheme is much lower than that for the L1-only CMS one.

5.2.2 Dual line buffering

The simulations highlighted that the new 32-pixels Pixel Region would require at least 10 ToT slots in order to minimize the charge information loss below 1%. As this would impact every row of the now deeper shared buffer (increase to 22 rows), another solution was devised, which could, in fact, even outperform the increase in ToT slots per row.

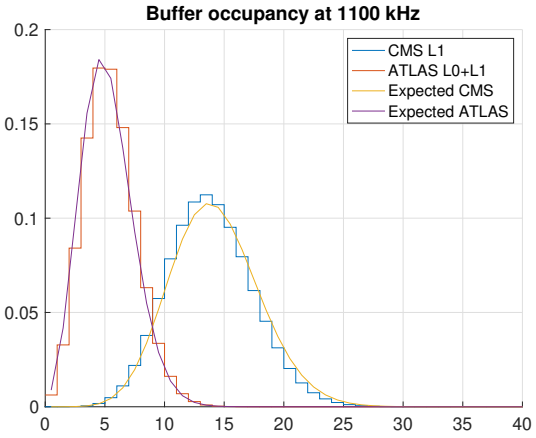

Figure 5.13: Buffer occupancy for a 4x8 Pixel Region hit rate

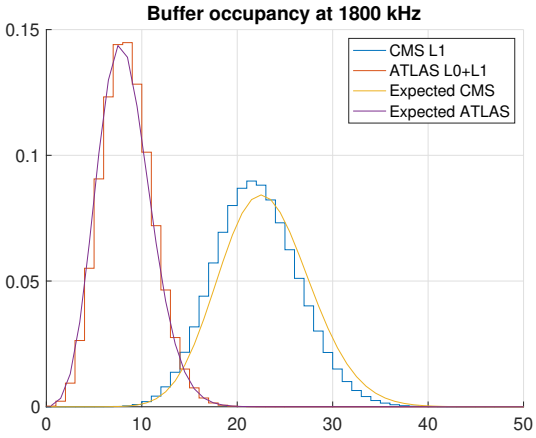

Figure 5.14: Buffer occupancy for a 8x8 Pixel Region hit rate

It was noted that only in 5% of the cases there are more than 6 pixels hit in the Pixel Region. If such 5% of the events could be buffered in 2 rows, instead of 1, the total amount of ToTs which could be stored would double, at the only cost of implementing the double line writing logic, along with the needed changes in the compressor.

If 5% of the events need 2 buffers to be buffered, the 22 rows would also need to increase by 5%, which means, to 23 rows. It was therefore decided to reduce the number of ToT slots per row to 6, and implement the possibility of writing an event in 2 buffer rows should it contain more than 6 pixels hit.

The selection of the row to write to, in RD53A, is made combinatorially by checking the state of the rows' FSMs. This means that the buffer is not operated in a circular mode, where each new event is written to the row index following the one of the preceding event. Enabling a double writing support to this structure cannot be done with the trivial implementation of writing to both row *i* and *i+1*, as there is no guarantee that row *i+1* is free. An example of such a situation is shown in Fig. 5.15, where event 5 is written 3 rows after that of event 4.

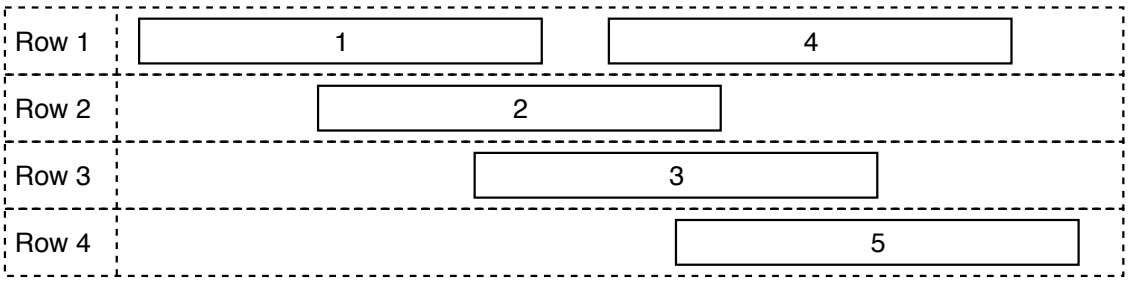

Figure 5.15: Example of non continuous row assignment in the buffer

For this reason, another approach was used: the priority queue encoding for the first available row was split in 2, serving odd and even rows respectively. Of the row indexes obtained by the 2 priority queues, the lower one is assigned to the first half of the event, while the higher is assigned, if needed, to the part containing the rest of the ToTs.

In this case, the risk of overwriting a valid line is removed. This solution has the other advantage of being almost completely parallelized, as the two priority queues are independent, and only the final, ordering, step depends on their output.

The readout logic needs no changes in order to support a readout on 2 buffer rows.

5.2.3 Compressor prototypes

The extension of the 4×4 Pixel Region to a 4×8 one was cause of a major restructuring in the compressor implementation. The direct-mapping implementation from CHIPIX65, in fact, was not scalable to the required size, as it occupies an area exceeding what was available. Moreover, the double row event support would now require not simply a mapping from 32 ToTs to 8 slots, but to 6 slots with the possibility to expand to 12.

In this subsection I will briefly describe the main solutions explored.

5.2.3.1 Cascade implementations

The first attempts to innovate the direct multiplexing implementation of RD53A stems from the research of a way to minimize the routing needed to perform the ToT multiplexing. The routing congestion, in fact, is the main factor that contributed to discard the direct multiplexing version.

In the cascade implementation, the multiplexers are arranged in a truncated pyramid, with each multiplexer taking as input the output of the 2 multiplexers preceding multiplexers in the layer above. This allows to consecutively converge the non-zero Tots to the corresponding total number of slots (for line 0 and line 1).

The number of multiplexing layers is equal to the number of pixels (N_p) minus the number of slots (N_s): it is not sufficient to compress to ($2N_s$) as it is desired to write only one line whenever sufficient. This full-scale cascade zero-suppression scheme is shown in Fig. 5.16.

In our case, the number of multiplexing layers is equal to $32-12 = 20$, plus a 6-layer $12 \rightarrow 6+6$ stage, for a total of 26 layers and 481 4-bit multiplexers: a number too great, as the area it occupies and the routing tracks it needs are unacceptable. Moreover, the sequential nature of this implementation make the compressor slow and thus hardly compatible with the single clock cycle (25ns) timing constraint.

Therefore, another approach has been investigated, which splits the pixels into chunks which can be compressed in parallel. In practice, every layer will compress a groups of $2N_s + 1$ TOTs into $2N_s$ slots, leaving the remaining $N_p\%(2N_s + 1)$ TOTs

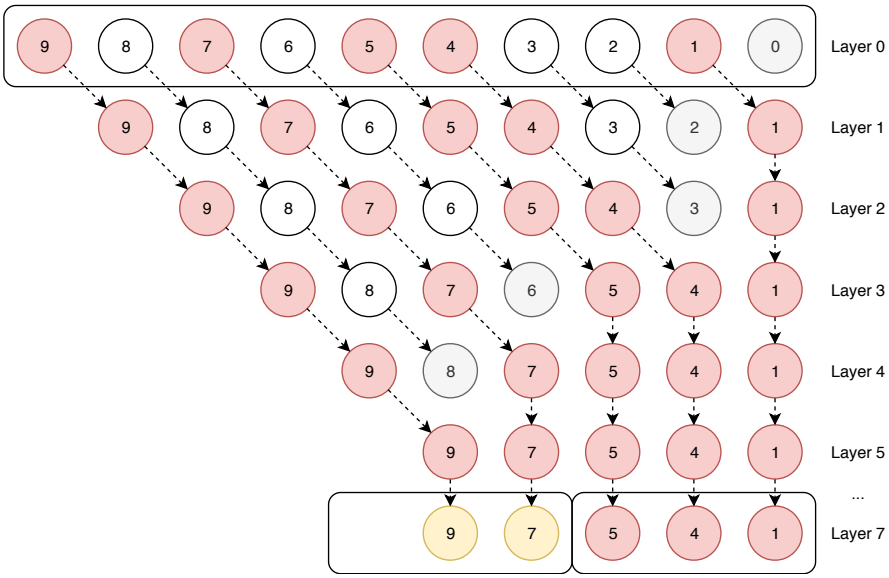

Figure 5.16: Scheme of $8 \rightarrow 3+3$ dual-row zero-suppression with a Cascade compressor

uncompressed, for the next layer. In alternative, the leftover TOTs can be assigned to the other compressors, which would therefore have more internal layers.

In this case, it is important that the compression to be performed in $2N_s+1 \rightarrow 2N_s$, as if it were to be $N_s + 1 \rightarrow N_s$, the compression would lose valid TOTs if there are more then N_s consecutive ones. Such a faulty compression is shown in Fig. 5.17.

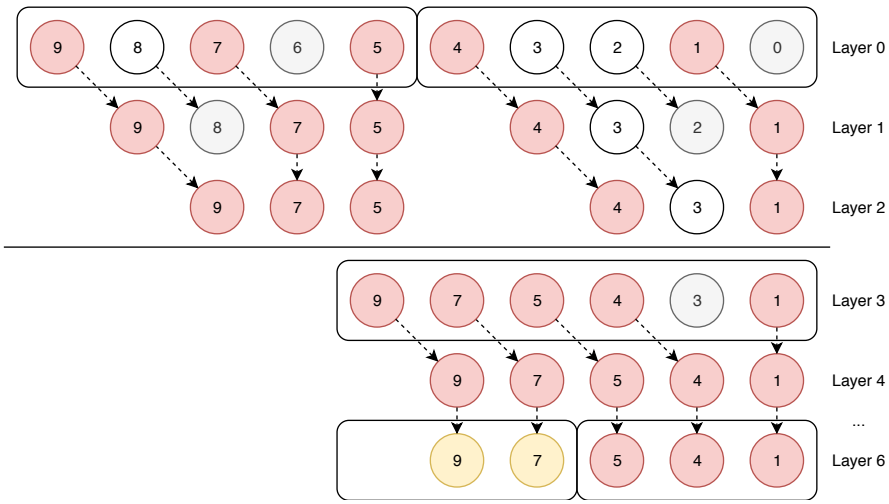

Figure 5.17: Scheme of $8 \rightarrow 3 + 3$ dual-row zero-suppression with a Parallel Cascade compressor

In our case, the correct implementation would consist of a cascade of:

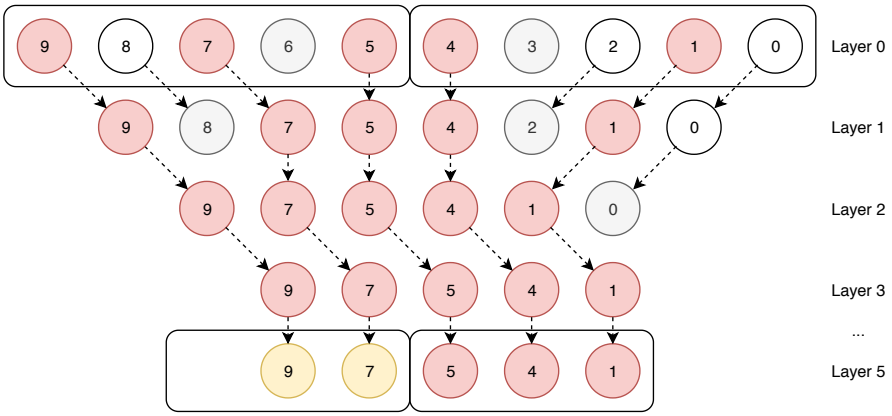

Figure 5.18: Scheme of $8 \rightarrow 3+3$ dual-row zero-suppression with a Converging Cascade compressor

1. $2 \times 16 \rightarrow 12$ compressors (4 layers, 108 muxes).
2. $1 \times 24 \rightarrow 12$ compressor (12 layers, 210 muxes)
3. $1 \times 12 \rightarrow 6 + 6$ compressor (6 layers, 51 muxes)

In total, the new arrangement would consist of 22 layers and 369 4-bit multiplexers. Again, this number is too high to be implemented in considerable area and routing tracks, although slightly more parallel than the previous implementation, with a save of 2 layers.

A further development of this concept takes advantage of the availability of 3-input multiplexers in the standard cell library. It may be possible, in fact, to perform the zero suppression not only from one side, but from both. Such parallel compression would reduce the number of layers, and by using a dedicated control logic, correctly handle the middle multiplexers, which may take data from 3 different units. The scheme of such implementation is shown in Fig. 5.18.

This arrangement results in a number of compressing and adjusting ($12 \rightarrow 6 + 6$) layers equal to half the number of inputs. In our case, it would be 16, much lower than the other alternatives. The number of cells required would be of 151 2-input muxes plus 120 3-input muxes. Although the total number of muxes is lower than for the previous cases, the increment in the control logic and complexity needed to operate this zero-suppressor made this implementation unfeasible in practice.

These results prompted an investigation in alternative zero-suppression techniques.

5.2.3.2 Windowed implementation

The main limitation of the cascade approach lies in the enormous number of multiplexers required, and the poor parallel predisposition.

Another approach, developed in the context of this thesis, consists in dividing the pixels in groups (*windows*) containing a number of pixels equal to the number of ToT slots.

Every window performs a N_s-to-N_s multiplexing, depending on the first window slot available. The availability of a window slot also depends on the assignments of the slots in the previous windows. In this way, the slots will already be in place, and a final OR between the output of every window can be performed to build the lines for the buffer. In order to support 2 lines, every window must be assigned a reference line, which depends on whether the previous window has filled all the slots available.

The N_s-to-N_s multiplexing may be implemented either with a direct multiplexing approach, or by the cascade one. The window size is restricted, and thus the direct multiplexing would be of little overhead compared to the great gain in time saving.

A further switch, signalling that the previous window has consumed the last remaining ToT slot, has to be propagated too, in order to prepare the second-line N_s slots. The same results can be obtained by using $2N_s$ windows instead of N_s, but the window size would then be too great for this approach to be worthwhile. A scheme of the windowed approach is presented (in a simple $10 \rightarrow 3+3$ case) in Fig. 5.19, with 2 cases: case 1 highlights a situation in which the implementation would work flawlessly; while case 2 illustrate a scenario in which the hard window constraint proves a strong limitation.

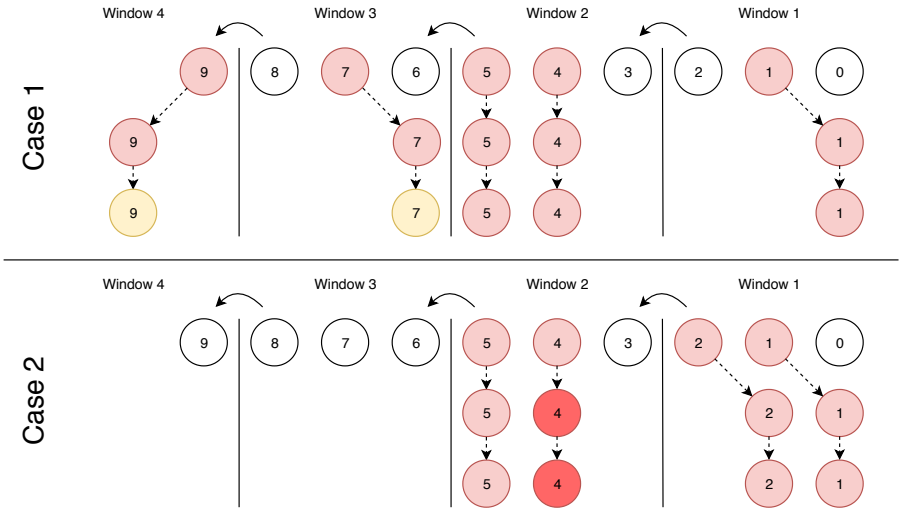

Figure 5.19: Scheme of $8 \rightarrow 3+3$ dual-row zero-suppression with a Hard Window compressor

In order to overcome this, windows could be changed from a ”hard” boundary to a ”soft” one, allowing TOTs to be assigned to slots of the preceding window, should it be empty. This would clearly have a big impact, as shown in the case 1 of Fig. 5.20, but at

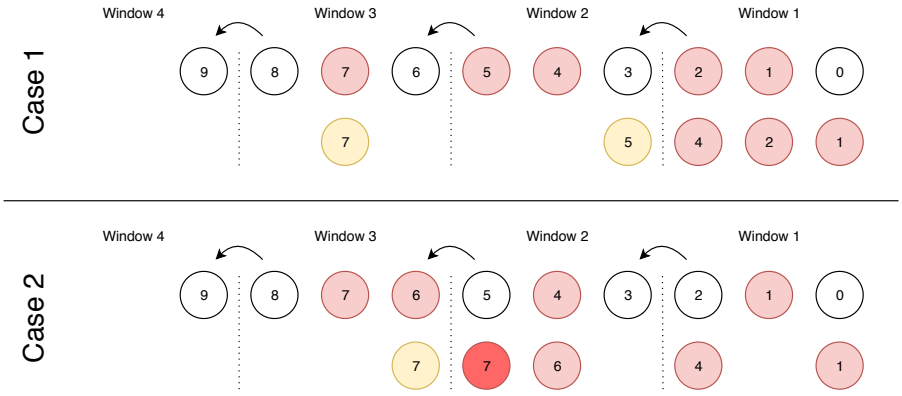

Figure 5.20: Scheme of $8 \rightarrow 3+3$ dual-row zero-suppression with a Soft Window compressor

the expense of a more complex network of muxes and wires.

The advantage of the hard windows, in fact, was that the routing is circumscribed between ToTs and slots in a group of pixels. Only at the end, a small routing overhead would be necessary to collect all the slots prepared. This had the advantage of reducing the connections between pixels physically distributed far away. In addition, this compressing architecture gets tricky when corner cases are considered. The case 2 of Fig. 5.20, shows one of them: one ToT changes the occupancy of the previous window, and thus its "full" flag. By doing so, it changes the line associated with the current window, and, thus, its own association. This kind of combinatorial loop is difficult to solve, and complex to handle.

Due to these problems, this architecture has been discarded, as the growing complexity surpassed its advantages.

5.2.3.3 Ripple implementation

The final implementation for the compression scheme dropped the window grouping, but instead resorted to single pixel multiplexing.

In the ripple zero-suppression scheme, each ToT can be forwarded to a local slot. This is achieved by having N_s 4-bit AND gates per ToT. The ToT-slot assignment is performed consecutively, with each assigned ToT increasing an index. The index is used by the following TOTs to perform its assignment.

In practice, the ToT-slot index assignment is not intrinsically connected with the ToT-slot value assignment. A scheme of this implementation is shown in Fig. 5.21.

The index propagation can use either one-hot or binary encoding. Whichever the choice, another line must propagate the ToT slots overflow signal, which can be used to

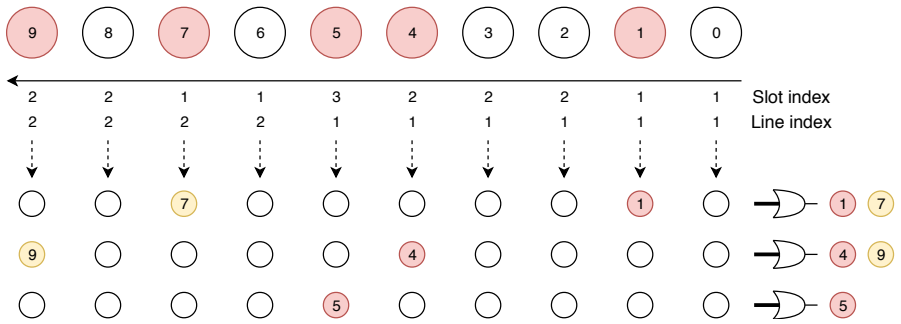

Figure 5.21: Scheme of $8 \rightarrow 3+3$ dual-row zero-suppression with a Ripple compressor

assign the following TOTs to the new line, and therefore restart the following assignments from slot 0.

This solution proved to be the best one in terms of area occupancy and routing required.

5.2.4 Dual pixel ToT memory

During a more accurate analysis of the simulation results, it came to light that the a significant number of charge mismatches (the number of pixels readout whose charge differed from the predicted one) was to be attributed to the overwriting of the pixels ToT memories during the staging time.

This inefficiency is only relevant at 40 MHz ToT operation, but has been addressed nonetheless in order to minimize the inefficiency sources.

Two main implementations were studied in order to provide an additional ToT memory in the pixel.

5.2.4.1 Dual row

The dual row implementation increases the number of ToT latches in the pixel by one, and associates a *full* flag to them. When the first ToT row is full, then, the new ToT is written to the second one.

However, the problem arises in how to tell the shared logic that the ToT was saved in the first or second local ToT memory. The hitmaps saved in the Staging Buffer, in fact, would have to be duplicated in order to support in-pixel addressing, but such solution was immediately discarded as the area overhead would have been too great.

The current implementation of the Staging Buffer only propagated the hitmap forward to the compressor. If it were to backpropagate it to the pixels, however, each pixel could know when it has been read. This can be used to clear the full flag from the ToT memory, and thus keep track of which ToT to send out to the compressor.

First row	Second row	ToT to propagate
Empty	Empty	None
Full	Empty	First row
Full	Full	First row
Empty	Full	Second row

When another hit is received from the pixel during the staging time, it is saved into the second row. When the first staging time elapses, then, its full flag is cleared, and thus when the second staging time elapses as well, the pixel knows it must propagate the ToT in the second row.

This solution suffers from situations in which a third hit arrives when the first ToT has been processed, and the second is waiting. In this case, the ToT would be written in the first slot, setting its full flag, and thus erroneously it propagate when the second staging time elapses, effectively inverting the TOTs to be readout between the second and third staging times.

Moreover, should an event in the staging buffer overflow, the corresponding pixels would not receive any read signal, provoking a misalignment between the 2 pixel ToT memories which would last indefinitely.

These incompatibilities may be overcome by preventing the writing of a third hit when the second is waiting, or through more complex handling logic, which would however further weigh on the architecture.

5.2.4.2 Single row

Another solution, which simplifies the problem and adds minimum overhead, does not use a second row of ToT latches to save the new charge information.

This single row solution stalls the Finite State Machine of the pixel whenever a second hit arrives during the staging time of a previous one, postponing the moment in which its ToT is written in the memory latch. In order to assess whether the ToT latch contains valid data, a full flag similar to the one used in the Dual row solution, is used.

When the pixel ToT is read, the full flag is reset, and the new ToT can be written. This mechanism suffers, however, from the staging buffer overflow problem described above. In order to overcome it, the staging buffer has been modified in order to propagate the Read signal to all the pixels when it is empty. This ”failsafe” mechanism automatically clears any leftover ToT which has not been processed due to the problem highlighted before.

5.3 Chip integration

Among the work performed by the author regarding the chip integration procedures, here we highlight the modelling of the Generic Analog Front-End, and some

considerations on the signal propagation along the Core Columns.

5.3.1 Generic Analog Front-End

In order to accelerate the integration of the final chip with the chosen front-end, while at the same time providing both digital architectures with a common Front-End reference against which to compare the performances, it was proposed to use a dummy Generic Analog Front-End (GAFE) with realistic sizes and characteristics.

The GAFE, however, was also used to optimize the size and placement position of the Analog Islands: during the place and route of the Pixel Cores in the RD53A design, it was clear that the Analog Quads (the collections of 4, mirrored, Analog Front-Ends, making up the single Analog Island) prevented the placement of standard cells in their vicinity.

This was caused by the DRC rules regarding the spacing of the wells in conjunction with the chosen offset for the standard cell rows.

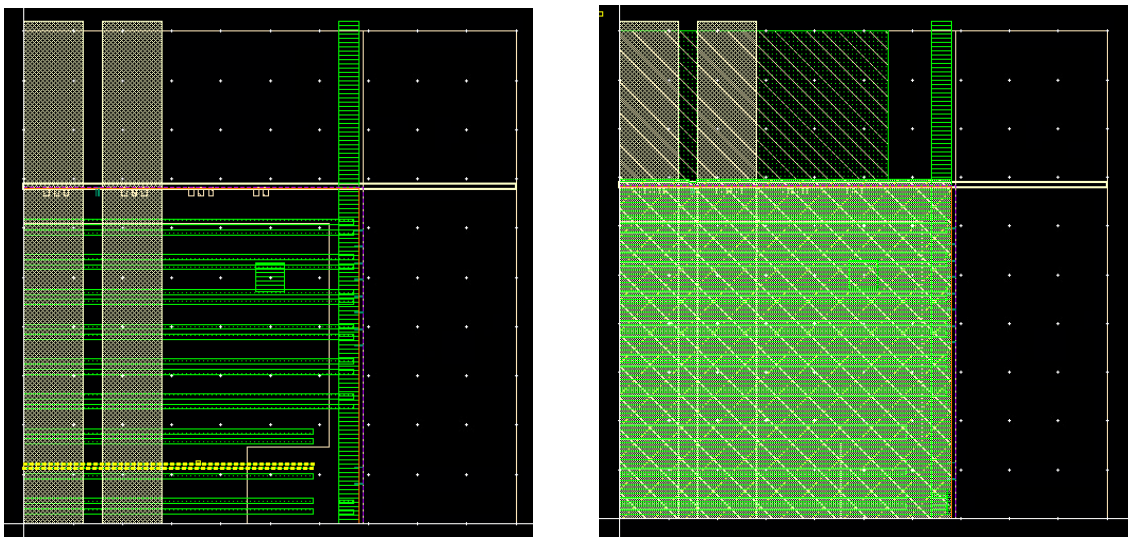

Figure 5.22: Layout of the Generic Analog Front-End

In Fig. 5.22, the main structures of the Generic Analog Front-End can be seen. The macro for the GAFE is a $50\,\mu m \times 50\,\mu m$ structure which contains, in its bottom left corner, the dummy front-end. The macro is larger than the actual front-end size in order to embed in it also the analog bias lines, shielding, and the Deep N-Wells needed by the Front-End.

In particular, on the left, Fig. 5.22 highlights the power distribution metal layers and a bias line. In the background, the white polygonal lines highlight the two Deep N-Wells: one dedicated to the Analog Front-End, and the other to the digital area.

The double DNW structure, already implemented in RD53A, guarantees maximum isolation, in order to protect the delicate Front-End logic from the substrate current spikes injected by the digital switches.

On the right, Fig. 5.22 shows the routing blockages for the GAFE, which also extend in the top region, as the bias lines will be placed there by the analog designers.

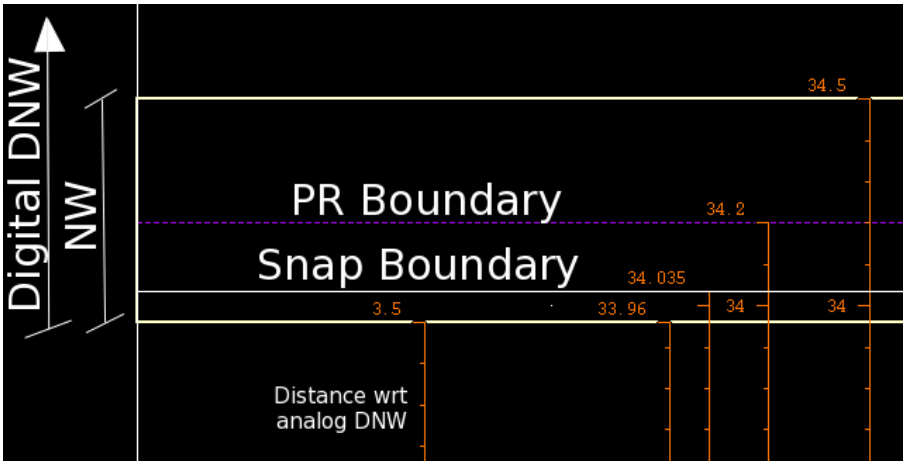

Figure 5.23: Detail of the main references and their positions in the design of the GAFE

In Fig. 5.23, the actual offsets and measurements for the Generic Analog Front-End are displayed. In the chosen technology library, a standard cell row has a height of 1.8 µm, the M1 power lines are 0.33 µm thick, and, of course, features a N-Well at one side for the PMOS transistors. In order to satisfy the DRC requirements and save area, every standard cell row is specular in the vertical dimension with respect to the adjacent ones: the N-Well is shared between the PMOS transistors of a row, and the PMOS transistors of the adjacent one.

In the designed GAFE structure, the closest standard cell row can start at 34.2 µm, a reference signed with the PR Boundary in Fig. 5.23. The PR Boundary shows where the center of the VDDD line would lie: by subtracting half its thickness, the VDDD metal line would begin at the height marked by the Snap Boundary.

The standard cells PMOS N-Well start 75 nm below that, and that is where a reference N-Well has been placed in the GAFE structure, along with the Deep N-Well for the digital part. The Analog Deep N-Well, in order to satisfy the DRC rules, must be spaced 3.5 µm from the digital DNW. This means that, in this configuration, the height for the Analog DNW is 30.46 µm, a size compatible with the Linear FE in RD53A.

Greater heights are discouraged, as they would shrink down the already narrow region in between analog islands, greatly hindering the routing of digital signals. At the same time, the 34.2 is an odd multiple of the standard cell row height (1.8), which assures that the situation would specularly apply to the bottom of the Analog Quad. This can be seen clearly in Fig. 5.24, where the bottom right corner of the Analog Quad

is shown to be perfectly integrated with the digital rows and the corresponding NWs and DNWs.

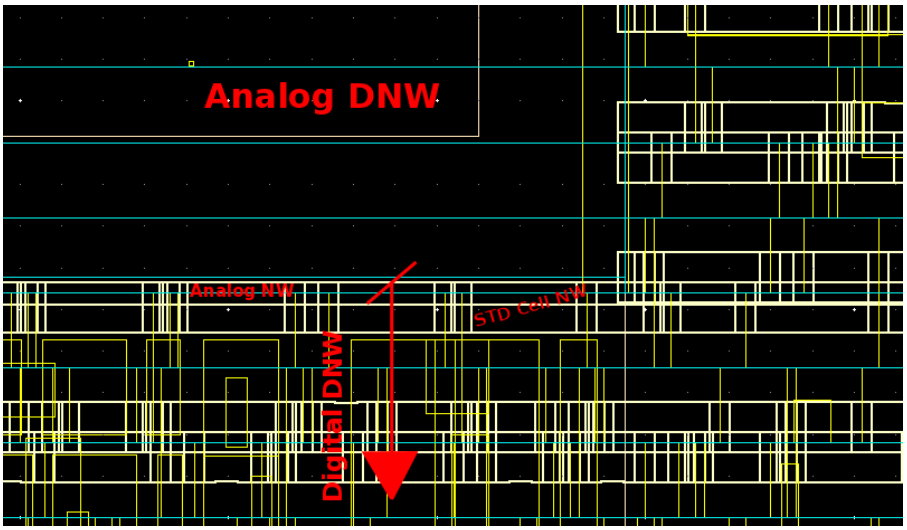

Figure 5.24: Detail of a GAFE island bottom right corner in a Pixel Core after place and route.

The positioning of the Analog Quads in the Pixel Core which optimizes the number of standard cells is, however, incompatible with the 50 µm × 50 µm pitch required for the bump placement, as 50 is not a multiple of the standard cell row height.

Due to this reason, in contrast to what has been done in RD53A, the bump for the connection with the sensor is not embedded in the Front-End view, but is instead superposed in the Pixel Core. In order to ensure connectivity of this bump with the Front-End input, the Analog Front-Ends must feature a small metal strip in the topmost layer: the actual bump will be placed somewhere along its length. The displacement has been measured to be between 0 and 0.8 µm.

5.3.2 Signal Propagation

When designing the communication buses and protocols in a pixel chip, it is key to consider the distance and travel times needed for the signals to propagate from the bottom of the matrix to the top, and the other way around.

Ideally, a pixel chip should properly balance the synchronous operations on its matrix: on one hand, it is desirable to reduce the timing uncertainty of the particle detection, one the other, a small skew in the clock helps decreasing the power consumption peaks on the rising and falling edges of the clock, smoothing them out.

Although a formal requirement exists for the average power consumption, it is particularly important to manage the power peaks in order to reduce the effects introduced

primarily by IR drop. The pixel chip, in fact, is usually powered only through power pads located at the periphery, and as the power lines needs to run through about 2 cm of silicon, it is important to both reduce the resistivity (by using the top metals, which are wider) and the peak current (by using decap cells and by skewing the clock). The IR drops could in fact possibly interfere with the delicate analog logic or with the digital states of the pixels.

It is therefore important to carefully control the timing distribution of the signals in the matrix, especially the clock, which synchronizes all the operations. In order to automatize as much as possible the assembly of the pixel matrix, it is undesirable to place dedicated buffer cells for the signals' propagation: these are, usually, placed inside the minimum synthesizable entity in the matrix, which, in our case, is the Pixel Region or the Pixel Core.

By extracting the propagation delay of a buffer cell, attached to a metal line which runs for the core height and having a load impedance of the buffer cell itself, it is possible to study the total effect of this delay. Fig. 5.25 shows these delays for 3 types of buffer cells: the fastest, and thus most power-hungry, is the *Clock* buffer cell, followed by a *Fast* cell, and a *Slow* cell.

By considering a 384-pixel column, which corresponds to 48 pixel cores, in a typical timing corner, the clock propagation from the bottom to the top would take 4 ns, while the fast signal 5 ns, and the slow signal as much as 12 ns. If the slow corner is under scrutiny, the delay would roughly double.

In order to compensate for the 4 ns skew, it is possible to employ a programmable signal delayer in the cores, as already done in the RD53A chip. The delayers could postpone the signals in the lower cores in a way to make them available to the logic roughly at the same time as they do at the top ones, but do also have a limit on the delay granularity because of both the power consumption of the individual delay cell, and the amount of area they take.

A signal delayer which compensate the clock signal every 8 cores (64 pixels), would result in a mean clock delay of 4 ns, with a skew of ± 0.65 ns, at the typical corner. This is shown in Fig. 5.26.

As the signal delayer's area is not negligible, it is hardly replicable to the other signals or buses. As the input delay requirements for the input ports has to be estimated, the compensated clock has to be compared against the propagation delay of uncompensated signals. Such comparison is shown in Fig. 5.27, where the signals are also delayed at the bottom in order to correctly arrive at the cores after the relative clock edge, with a 1 ns margin to compensate for any mismatch.

This study, performed on the buffer cells of the chosen CMOS technology, along with layout simulations, were important to properly modify the structure of the signal delayer proposed in RD53A and correctly choose the buffers which could minimize the delay on critical signals. Both chip-wise Static Timing Analysis (STA) and simulations performed on the placed and routed design with SDF timing files, were used to cross-check these design choices.

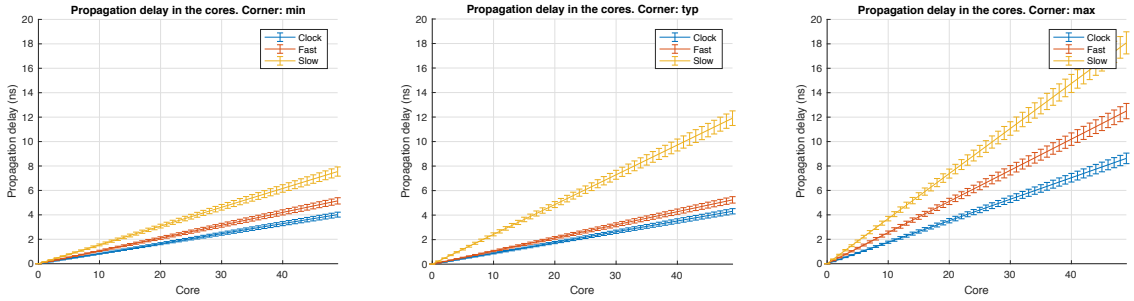

Figure 5.25: Propagation delays for 3 different buffer cells at various timing corners.

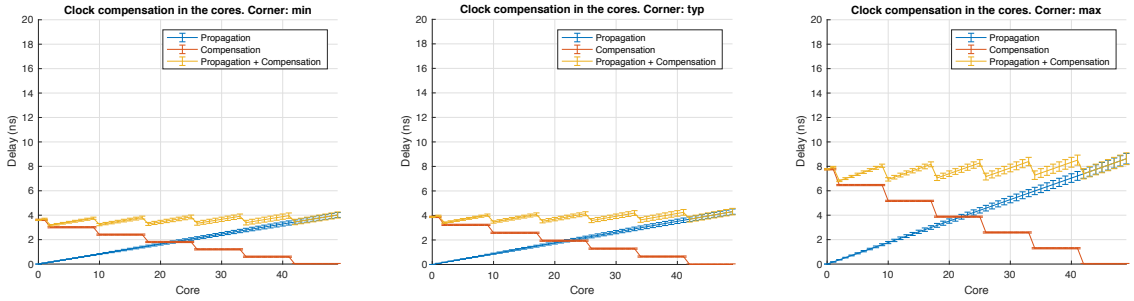

Figure 5.26: Effects of the clock compensation block on the clock arrival time in the Pixel Cores at various timing corners.

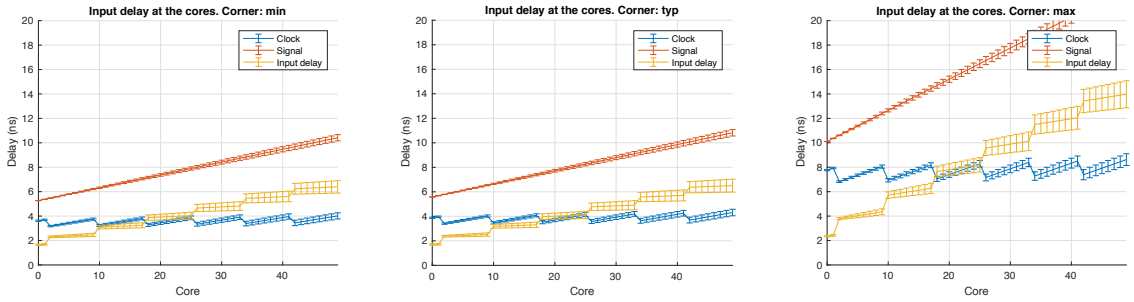

Figure 5.27: Comparison of clock and signal arrival times in the Pixel Cores for various timing corners.

5.4 Performance evaluation

The great customizability of the architecture allowed for an iterative selection of the best parameters, according to the results of the simulations using CMS data. Among the parameters varied for the analysis, there is:

- Pixel Region form factor (4×8 or 8×4)
- Latency Buffer depth
- Staging Buffer depth
- Pixel 2-ToT strategy
- Dual line events support
- Number of ToT slots

A quick analysis fixed some of these parameters early on, such as the support for dual line events, and the 6 number of ToT slots. The advantage of using the horizontal form factor, when using the 25 µm × 100 µm pixels, in the center of barrel, became immediately clear, which, together with the choice of 22 latency buffer rows, resulted in an inefficiency of about 0.07%, as shown in Fig. 5.28.

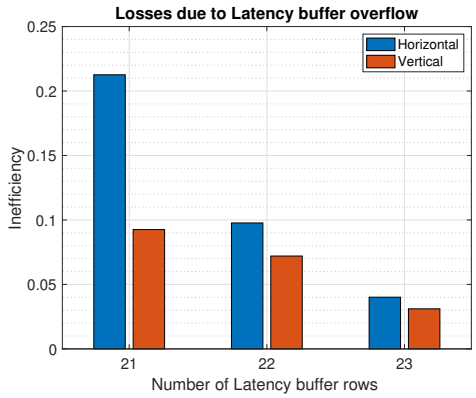

Figure 5.28: Event loss due to latency buffer overflow in the proposed RD53B architecture

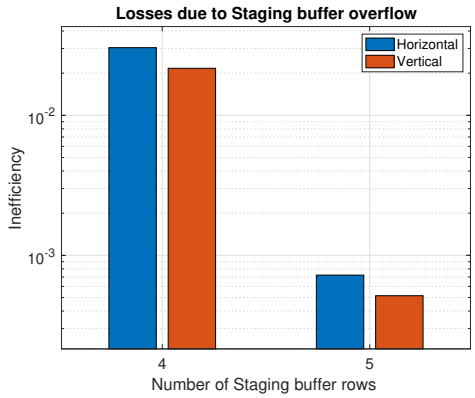

Figure 5.29: Event loss due to staging buffer overflow in the proposed RD53B architecture

The choice of retaining 4 staging buffer rows is justified as the inefficiency is tolerable, and the gain in increasing the number of rows is negligible, as demonstrated by Fig. 5.29.

Last, the double pixel ToT strategy highlighted how the great the impact that dual pixel ToT memories would be. When a single ToT latch is used, susceptible of overwriting, the losses would almost arrive at 1.5%. The double row approach would bring this number down to 0.2%, and the single row approach further down to 0.02%. The single row approach would also save are for a total of 1.4% on the total occupancy (from 88.4% down to 86.0%).

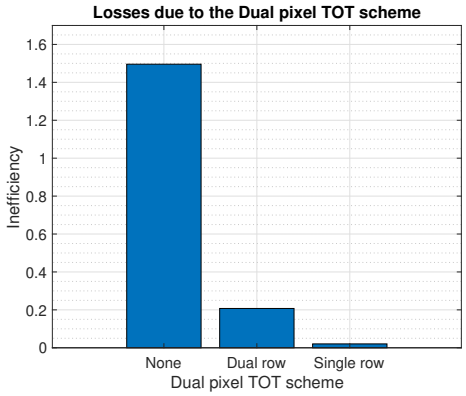

Figure 5.30: Charge information loss with various Dual pixel ToT memory implementations

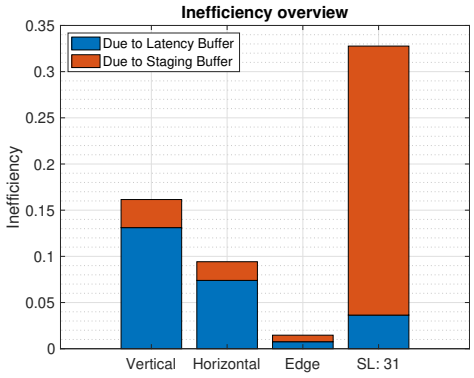

Figure 5.31: Overview of the event losses in particular cases

A final overview of the losses for these choices is shown in Fig. 5.31, which also contains the corner cases of a simulation with the horizontal form factor and edge of barrel position, and one with a purposedly set maximum staging time. At the end, the following parameter values were chosen:

- Dual line events support → Yes
- Number of ToT slots → 6
- Pixel Region form factor → 4×8
- Latency Buffer depth → 22
- Staging Buffer depth → 4
- Pixel 2-ToT strategy → Single row

The placed and routed design has an area occupancy of 88.3%, and the total power consumption averages 3.6 µW per pixel.

5.5 Concluding remarks

The CBA architecture proved the feasibility of a smarter logic in the Pixel Array, with important gains in terms of area occupancy and performance. Its main limitations derive from the intense routing required, which, in the particular floorplanning with the Analog Islands hindering core-wise connectivity, proves difficult to achieve if the standard cell number is too great.

Monte Carlo Dataset	Pixel Hitrate $MHz\,cm^{-2}$	Event loss (%)		Charge loss (%)	
		DBA	CBA	DBA	CBA
$25 \times 100\,\mu m$					
Layer 1 (center)	3470	0.4389	**0.2552**	-	0.0487
Layer 1 (edges)	334	0.1638	**0.0389**	-	0.1455
Layer 4 (center)	27	0	0	-	0
Layer 4 (edges)	29	0	0	-	0
Endcap first (inner)	110	**0.0007**	0.0035	-	0.0213
Endcap first (outer)	28	0	0	-	0.1292
Endcap last (inner)	59	**0**	0.0026	-	0
Endcap last (outer)	16	0	0	-	0
$50 \times 50\,\mu m$ pixels					
Layer 1 (center)	356	0.4664	**0.3467**	-	0.0845
Layer 1 (edges)	452	**0.1719**	0.4846	-	0.6091
$25 \times 100\,\mu m$ pixels, additions simulating extreme cases at Layer 1 (center)					
Clustered hits	494	**2.2846**	3.5663	-	0.2656
$1\,GHz\,cm^{-2}$ Noise	446	**2.5874**	10.972	-	0.4147

Table 5.3: Overview of the event and charge information losses in various simulation corners for both the final CBA and DBA buffering architectures

Tab. 5.3 presents the final results for the CBA architecture, in comparison with the other architecture developed in the context of the collaboration. The CBA architecture performs better in almost all the conditions, save the Endcap simulations (which, however, presents little statistical data and is thus less accurate), and the edge of barrel in Layer 1 when using 50×50 pixels.

The more available area in CBA allowed to add also triplication in the configuration bits for the pixels.

Chapter 6

Conclusions

The goal of this thesis was to assess the feasibility and radiation hardness of a digital architecture for HPD chips developed in 65nm CMOS. The greater logic density available in this technology node, in fact, gives room for more features and many improvements over the previous readout architectures.

This thesis focused on the methodology to assess the optimal buffering architecture in the Pixel Matrix, by employing a large number of parameters in the architecture definition and performing a multivariate analysis based on both statistical data and physical simulations. The presented architectures are the results of an iterative process, where the improvement of the simulation quality helped refine the architecture type and parameters.

Both the CHIPIX65 and RD53A digital architectures proved successful for HL-LHC applications, and presented practical solutions to the design problems arising for chips which have to withstand a very high pixel hit rate and radiation flux. These solutions and the methodology underneath their implementation can be used also in other applications where the power consumption, hit rate, and radiation hardness are a concern. In particular, the area-saving buffering schemes proposed are particularly adapt for designs which have to integrate a large number of functionalities, or have otherwise strong constraints on the floorplan.

6.1 Main findings

This research, which eventually led to the development of two Hybrid Pixel Detector chips called CHIPIX65 and RD53A , respectively with 64x64 and 400x192 pixels, and important contribution to a third one called RD53B (400x400 pixels) , allows to draw the following conclusions:

- the use of a more scaled 65nm CMOS technology, together with the study of a novel digital architecture, with resources shared among several pixels, has allowed to increase substantially the granularity of pixel detector for High Energy

Physics experiments, while maintaining a low power consumption and a high efficiency at extreme particle rates even at the high ration of radiation does

- the pixel size can be brought down to 50 µm × 50 µm and integrate an intelligent logic enabling to sustain pixel hit rates equal to 3 GHz cm^{-2} with trigger latencies of 12.5 µs and a power consumption <10 µW /pixel (50% analog and 50% digital electronics) even in extreme radiation conditions with a Total Ionization Dose up to 500 Mrad
- the key elements of the digital architecture developed in this thesis - a common central buffer shared up by up to 32 pixels and a smart zero-suppressing architecture - have proved to be successfully working allowing to reach fully efficient pixel detection (>99.5%)
- the optimization of the digital architecture, and in particular of the buffering scheme, is possible through a thorough statistical analysis, which can be refined using physical simulations of the incoming input signals (pixel hits generated by the charged particle).
- a complete verification environment, based on System Verilog, capable of discerning losses due to various sources in the design, is crucial to debug implementation flaws or corner cases which do not happen during normal operation simulations.

A summary of the three chips' performances is presented in Tab. 6.1. It should be noted, however, that the loss estimates for CHIPIX65 and RD53A have been performed via random hits generated internally by the verification environment. The analyses of the RD53B proposals, instead, used CMS physics simulation data, and the resulting losses were shown to be about 5 times higher both for events and charge information, with respect to the counterpart.

6.2 Limitations of the study

The development of the architectures presented in this thesis has been guided by the statistical data available at the time of design. The quality of this data has improved more recently, surpassing that of the cluster models used during the development of CHIPIX65.

This reliance of the design parameters on the simulations available allowed to continuously improve the parameters from one chip iteration to the next, but at the same time didn't allow a proper research into the architectural alternatives discarded at the beginning because though to be inappropriate

Had the advanced simulation data and verification methodologies been available earlier in the design process, the development might have proceeded differently, potentially uncovering solutions more apt for the application.

Readout chip	CHIPIX65	RD53A - CBA	RD53B - CBA
Submission	2016	2017	2019 (foreseen)
Application	ATLAS-CMS Phase-2 Upgrade		
Technology	CMOS 65nm		
Radiation Hardness	500 Mrad		
Readout	SLVS @ 320 Mbps	4 SLVS @ 2.56 Gbps	4 SLVS @ 2.56 Gbps
Pixel size	50 µm × 50 µm		
Pixel matrix	64 × 64	400 × 192	400 × 384 (CMS)
Chip size	10.2 mm × 8 mm	20.0 mm × 11.8 mm	TBD
Particle rate	750 MHz cm^{-2}		
Trigger rate	1 MHz		1 MHz L0: 1 MHz L1: 500 kHz
Trigger latency	12.5 µs		12.5 µs L0: 6 µs L1: 30 µs
Charge Inefficiency	<0.4%	<0.5%	<0.15%
Event Inefficiency	<0.04%	<0.03%	<0.3%
Power consumption	7.5 µW/pixel	8.1 µW/pixel	3.6 µW/pixel

Table 6.1: Summary of the chips developed as part of this thesis

Appendix A

Statistical analysis

Statistical analysis of the buffering requirements can be performed thanks to the assumption of the 75kHz pixel hit rates. But in order to achieve maximum precision, physical simulations have to be performed in order to assess the real event rates.

The simulations available for this work were performed using Physics CMS data, to assert the hit rates of single pixels, columns, or group of pixels (Pixel Regions). The key aspects considered are the number of pixels and the form factor. Given that simulations take into account the contribution of the Lorentz force, due to the background magnetic field, which spreads the clusters in the polar direction (indicated by the angle ϕ in the whole detector, or the x-coordinate in the modules, see Fig. 2.2), we should expect a lower hit rates for Pixel Regions elongated along x. However, if we consider the modules further away from the center of barrel towards the edges, particles will have greater pseudorapidity and hence elongate the clusters in the y direction. The worst case hit rate will be the design parameter driving the choice of the Pixel Region size and form factor.

In the center of barrel, where particles mostly traverse the sensor perpendicularly, as expected, the Lorentz effect is stronger and particle clusters are mostly elongated along x/ϕ, that is, in the direction of the ROC columns. This effect is shown in Fig A.1. In the edge of barrel, instead, the incident angle has a stronger influence, as shown in Fig A.2.

As particle generation happens at a constant rate and without time inter-dependency, it can be described by Poissonian statistics. If an event on average happens λ times during a period of time T, the probability that such event happens k times during the same period T is described by: [63]

$$P(k \text{ events in period T}) = \frac{\lambda^k e^{-\lambda}}{k!} \tag{A.1}$$

By applying this principle to our case, the formula above can be used to evaluate the Probability Density Function (PDF) of a pixel being hit k times during the trigger latency T. By integrating the PDF over the interval $[0, k]$, we obtain the Cumulative

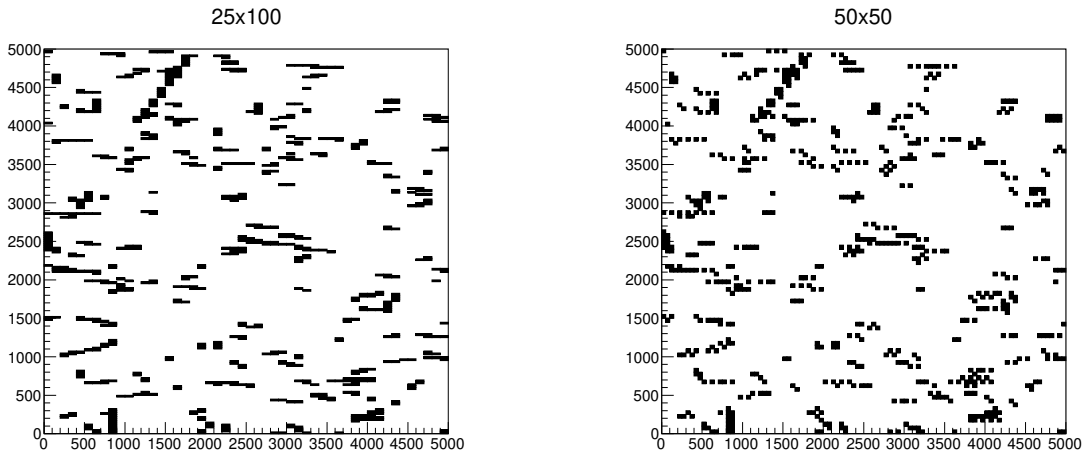

Figure A.1: Physics CMS simulations at the layer 1, center of barrel, showing the mapping between 25×100 pixels and 50×50 ones.

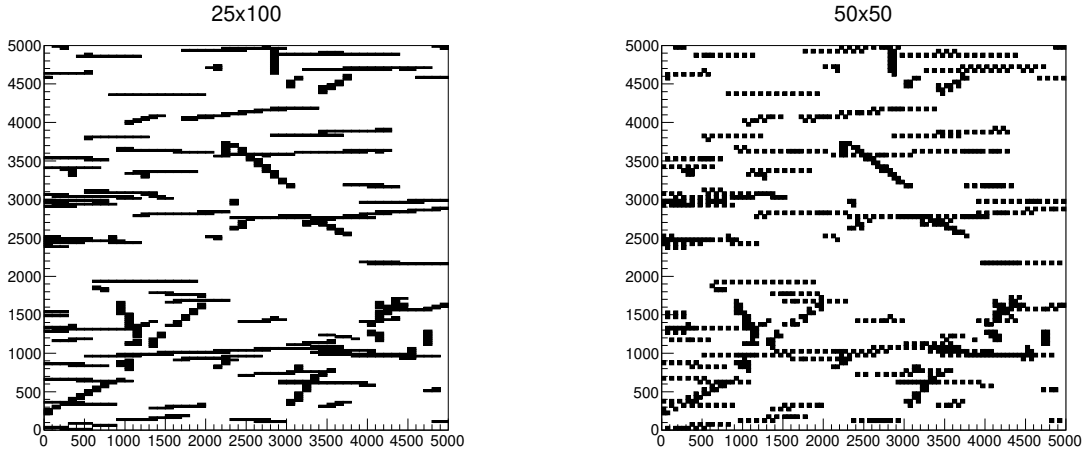

Figure A.2: Physics CMS simulations at the layer 1, edge of barrel, showing the mapping between 25×100 pixels and 50×50 ones.

Density Function (CDF), that is, the probability that a pixel is hit at most k times during the trigger latency, or, in other words, that the event buffer of the pixel will reach an occupancy of k rows. It follows that 1-CDF represents the probability of losing hits if the pixel buffer has k rows.

Buffering can be done at different stages. In early chips, buffering was performed at the periphery. Pixels would send their hit data downstream, where all the data coming from a pixel column would be buffered and matched against the trigger. With the advent of more dense technology node, buffering has been moved inside the Pixel Matrix, in

pixels or Pixel Regions.

In this Appendix, the feasibility of various buffering types is analyzed.

A.1 Single Pixel

The probability of a single pixel being hit during the trigger latency can be evaluated by applying Equation A.1 to this problem, by using:

$$\lambda = T_{\text{Trigger Latency}} \times F_{\text{Hit Rate}} = 12.5\,\mu\text{s} \times 75\,\text{kHz} = 0.9375 \tag{A.2}$$

Tab. A.1 shows the values of the PDF and CDF for a number of hits in the pixel from 0 to 5.

Hits	PDF	CDF	1-CDF
0	39.16%	39.16 %	60.84%
1	36.71%	75.87%	24.12%
2	17.21%	93.08 %	6.92%
3	5.38%	98.46%	1.64%
4	1.26%	99.72%	0.28%
5	0.23%	99.95%	0.05%

Table A.1: Probabilities of a single pixel to be hit multiple times during the trigger latency

In practice, the value of interest is 1-CDF(x), as it represents the probability of a pixel to be hit more than x times. It can be seen how, in an effort to keep all the sources of inefficiency below 1%, a pixel should be capable of buffering at least 4 events for a period of time equal to the trigger latency.

This kind of analysis has been used when the Physics data was not available. The simulated data, which is more precise, yield similar hit rates, which are documented in Fig. A.3, for the center of barrel, and Fig. A.4, for the edge of barrel. The average rate is 86 kHz in the first case, and 84 kHz in the latter.

By adjusting the probability distributions with this mean values, we find that the minimum number of rows that allow < 1 % losses is still 4, wth 0.5% of events lost.

In considering the buffer width, one should recall that every pixel should be able to retain all the necessary information:

1. ToT - 4 bits
2. Timestamp - 9 bits

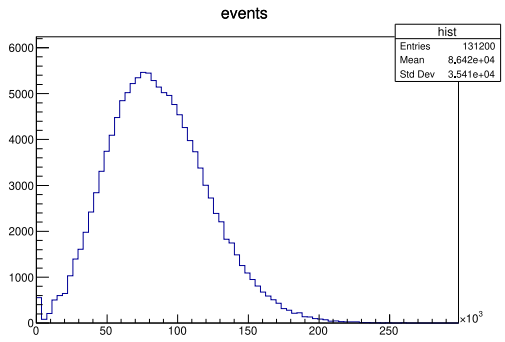

Figure A.3: Distribution of the pixels' hit rates at layer 1, center of barrel.

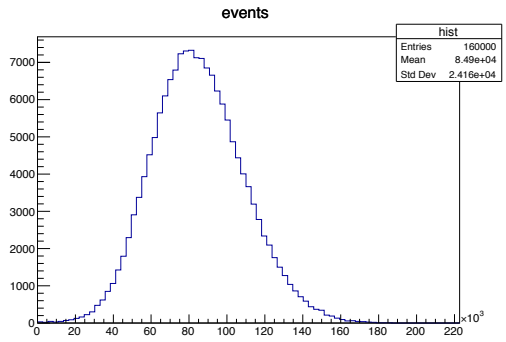

Figure A.4: Distribution of the pixels' hit rates at layer 1, edge of barrel.

In addition, the pixel should host all the logic for the trigger timestamp comparison, which contribute to even more area. These considerations represent strong evidence that single pixel buffering is not an optimal design choice in terms of area occupancy. One could think, instead, of dropping the buffering (and, thus, trigger matching) in the pixels in favor of direct transmission to the Chip Periphery.

A.2 Single Pixel Column

If a pixel column is taken under consideration, it would be wrong to assume that the probability of a single pixel to be hit in a bunch crossing would directly correlate to the probability of a number of pixels to be hit in a pixel column. That is because the hits are not uncorrelated with one another, as a single particle produce a cluster of hits on the sensor.

Simulations have to be performed to evaluate the correct estimate of the number of pixels hit in a bunch crossing in a pixel column. CMS Simulations with 25 µm × 100 µm pixels, in the center of barrel at the first layer yield a hit rate of almost 4 MHz, as shown in Fig. A.5.

This means that, on average, a pixel column receives a hit in any of its pixels every tenth clock cycle. Moreover, as shown in Fig. A.6, although most of the times there is only 1 pixel hit in the column, the probability of multiple hits is still high. This situation makes single pixel communication to the Chip Periphery not ideal, as the transfer rate would be very high, and, as the columns are tall, the power consumption on those lines would be high as well.

Moreover, both sorting and trigger matching would have to be performed at the End of Column. With over 50 buffer rows per single pixel column, this solution has been discarded.

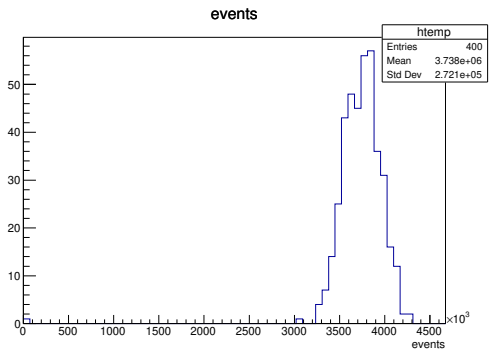

Figure A.5: Distribution of the columns' hit rates. The columns are 328-pixels tall.

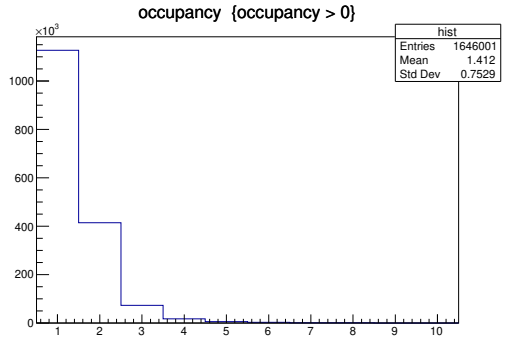

Figure A.6: Distribution of the occupancy of columns. The columns are 328-pixels tall.

A.3 Pixel Region

The solution adopted in this thesis employs another level of hierarchy in the Pixel Matrix: the Pixel Region. By unifying the storage for a group of pixels, in fact, it is possible to save key area in the event buffer. The rationale behind this approach lies in the fact that particles usually leave a cluster of pixel hits when they traverse the sensor, thanks to charge sharing, couplings and other effects. As hits in the pixels are therefore not truly independent, and usually appear at the same bunch crossing, the storage can be optimized by employing some kind of addressing scheme and sharing the memory.

The buffering scheme is important as it affects one of the most cumbersome components in the architecture: the latches/flip flops for storing the event information can easily take up the majority of the area in a Pixel Region. Apart from the number of memory elements, one must take into account the clock tree distribution to these cells, and the writing/reading logic and multiplexers which the architecture scheme implies. The buffer in a Pixel Region needs to store also the information regarding which pixels, of all the Pixel Region ones, have been hit and which have not.

The change in the hit rates according to the Pixel Region form factor and size is shown in Fig. A.7, where both the cases at the center of barrel and edge of barrel are depicted. On the right, the corresponding minimum event buffer depth which guarantees a ≤ 1 % event loss, is shown.

This work represents the basis for the analysis performed in Chap. 5.

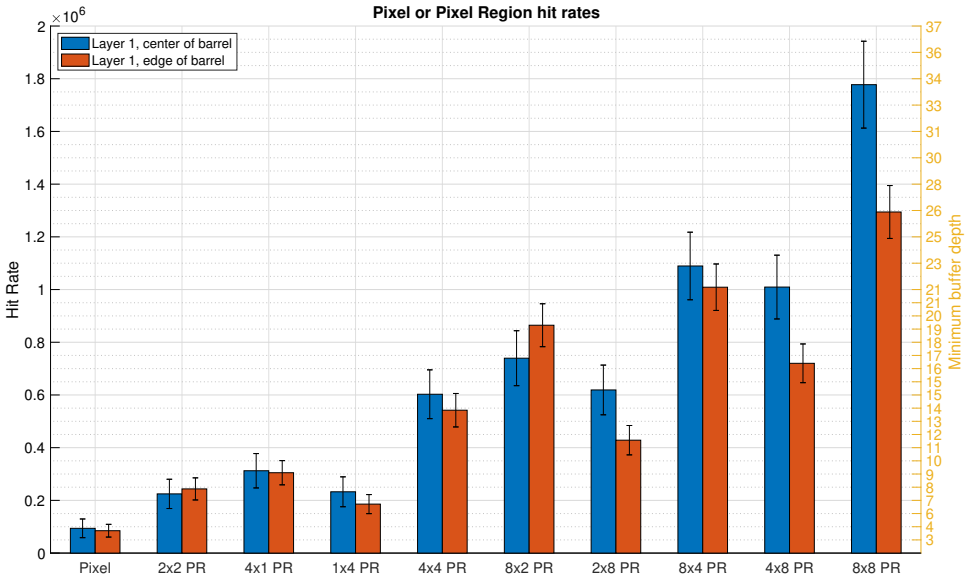

Figure A.7: Pixel Region hit rates for various sizes and form factors

Appendix B

Radiation tolerance

The high particle rate in the inner parts of a detector is responsible for non-negligible radiation damage to both sensor and electronics. This damage can be classified either as bulk effects, caused by displacement of crystal atoms, or surface effects, which affect dielectrics and silicon-dielectric interfaces.

Bulk damage appears when high energetic particles interact not only with the electronic cloud, but also with the atoms' nuclei, displacing them from their lattice position. In fact, the silicon in the wafers is in crystal form, and the induced crystal imperfections may be electrically active and thus change the electric properties of the material. The concentration of such defects in chips for HEP experiments is so high that after some years of operation will exceed the initial substrate doping concentration, effectively changing the bulk type. Other bulk damage will manifest as the increase of the leakage current, and charge trapping, which manifests in the depletion regions by reducing the signal generated by passing particles.

Surface effects, in contrast to bulk effects, are not due to irradiation but to ionization phenomena in the silicon. When ionizing radiation leads to the creation of electron-hole pairs in the oxide layer, most of the pairs recombine immediately. In the remaining ones, the electrons are rapidly driven to the positively biased electrode, while the holes slowly move in the direction of the electric field[1]. If the holes reach the silicon-oxide interface, they may be trapped there permanently because of deep hole traps, mainly due to interstitial oxygen. The accumulation of these charges leads to an increase in the transistor's flat band voltage. [91, 86]

Given that CMOS chips have a very high doping concentration (at least one order of magnitude higher than defect density at very high fluence), the primary radiation effects which have to be compensated in radiation-hard circuits are surface effects. [40]

The primary sensible oxide surfaces are those of the transistor gates, of Shallow Trench Isolation and of Gate Spacers. Although gate oxide charge would be the most

[1] The difference in speed reflects the difference in the electrons and holes mobility in SiO_2: the mobility of the holes is a factor of 6 lower than that of electrons.

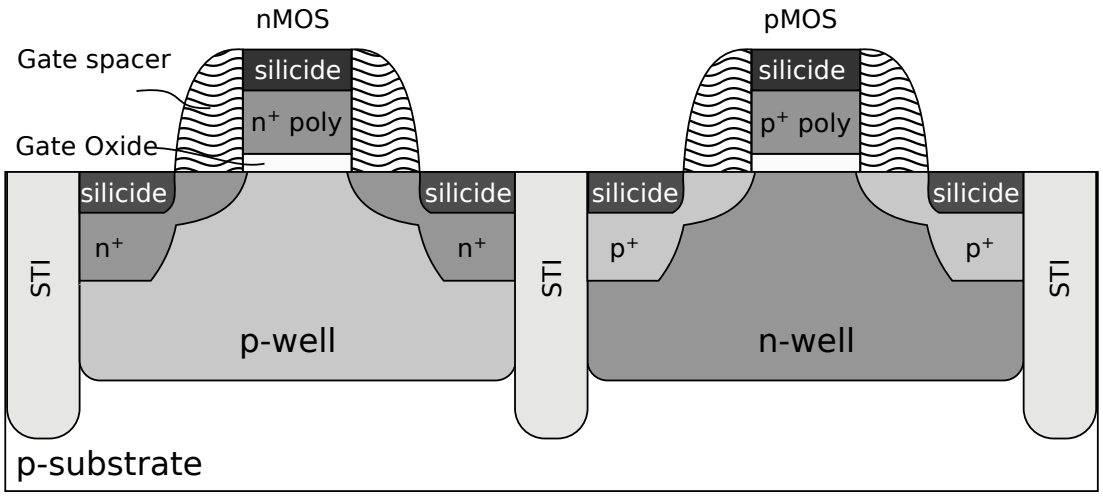

Figure B.1: Section of a MOS structure, highlighting the position of the main oxides.

problematic, effectively increasing the transistor threshold voltage, in 250nm CMOS processes, and below, the gate thickness is low enough (about 6nm) to allow for tunneling effects to eventually drive any collected holes away.

The leakage current created by these tunneling effects in the gate dielectrics represents the reason that drove the industry away from SiO_2 dielectrics in tech nodes of 45nm and below. STI and spacers, instead, are thick enough to store positive semistatic charge. This, however, can eventually migrate due to the strong electric fields involved, and reach the silicon-oxide interface. There, in virtue of other ricombinational processes, their effects is fairly limited, affecting mainly the effective width (W_{eff}) of PMOS devices and the effective length (L_{eff}) of PMOS devices twice as much as NMOS devices (RINCE and RISCE effects). [40]

Another common problem in high radiation environments is represented by instantaneous soft errors caused by energy loss by ionizing particles. These errors are called Single Event Upsets (SEUs). A single event upset typically takes the form of a bit flip in a memory cell due to the energy deposited by a particle in the control node of the cell itself. SEUs are parametrized by their cross-section: a measure of the number of bit flips per unit of fluence, expressed in cm^2/bit. The SEU cross-section is always below 1, so that the lower cross-section, the lower the sensitivity to SEUs. Cross-sections are expressed with respect to the Linear Energy Transfer (LET) of the particles, which represent the energy deposited by a ionizing particle to the material traversed per unit distance and is thus, in our case, a function of the energy of the particles themselves and the degree of the particle beam with respect to the surface. The SEU cross-section usually saturates with LET, with a minimum threshold and a knee.

It should be noted that various factors and design pratices for digital designs may help reducing the SEU cross-section of a circuit, apart from tweaking transistor sizes. For example, by employing deglitchers on, and close to, the the latch/ff control pins,

we reduce the amount of wire sensitive to energy deposits. Moving memory cells away from each other also reduce the probability of multiple bit flips, which for higher LET can be a non-negligible contribution to the total cross-section.

Radiation hardness of the CMOS 65nm technology The effects of the Total Ionizing Dose (TID) deeply affect the design of analog circuitry, but are also responsible for increased delays in digital standard cells. The choice of a deep (but not too deep) submicron technology node is key to overcome the gate oxide surface effects, while also allowing to increase the logic density. The 65nm technology, which typically has a gate oxide thickness of about 2nm, is the smallest features size CMOS node for which all manufacturers use SiO_2 gate dielectric. [38]

Bonacini et. al previously anticipated that, for 65nm devices, some design practices, as the increase of transistor widths, may limit the increase in leakage current in NMOS devices, and loss in drive current in PMOS devices. It appears that foundry-provided digital IP blocks may continue to work after irradiation, with an increase in static current dependent on the type of transistors employed in the design. They also proved that the SEU cross-section of the standard library register in the 65nm technology they studied is lower than equivalent blocks in other older technologies and that this measure is only marginally affected by the power supply voltage.

The test circuits employed also showed how the accumulated TID slows down digital blocks (in their case, a ring oscillator), most likely by reducing the driving current in PMOS devices. [11, 62]

Another chip submission in 65nm showed the same pattern: the lower the transistor size, the higher the degradation effects on the devices, especially PMOS transistors. Radiation campaigns indicated how after a TID of 200 Mrad, followed by ambient and high temperature annealing, the digital cells averaged an increased delay of 20%, while after 500 Mrad, of 100%. The same research showed a dependence of the radiation effects on the number of tracks and transistor threshold. Despite the increased radiation hardness of a greater number of metal routing tracks (12, 18), in order to contain the costs, a 9 metal tracks stack has been shown to be sufficient. [14].

By taking these delays into account, a modified digital cell library has been built that model the TID-induced behavior of the standard cells. [66]

In conclusion, the choice of a 65nm CMOS technology represents a good compromise between cost, performance, logic density, and radiation tolerance. It is sufficiently radiation hard to avoid the usage of redundant logic and ELT devices but in exceptional cases.

Appendix C

Position Resolution

Particle track identification demands for ever increased precision on 3D space point reconstruction in the detector. [40]

The charge cloud deposited on the sensor has a dependence on incident particle type and trajectory, and the detector electromagnetic fields. In order to keep the pixel occupancy down, and thus enhance the track identification, the granularity of the pixel sensor has to scale to the charge cloud deposit level. With current detectors, the granularity is even greater, which allows algorithms to be even more precise by evaluating the cluster centroid and decrease the fake rate during track reconstruction.

Particle hit information can either consist of binary information (pixel hit, or not hit), or contain a measurement of the charge left in every pixel. With a binary readout, where the only information coming from the pixel tracker are the event number, and the position of the hit pixels, the worst-case resolution is pitch/$\sqrt{12}$ (this is the standard deviation of a uniform distribution in [0, pitch]. It was shown that in current pixel detectors in HEP experiments, this upper limit is not reached, and this will never be the case until the pitch becomes so small as to be comparable to the distance between energy deposits in silicon [103]. The reason behind this is that when calculating the centroid, other info is usually not taken into account: for instance, the cluster shape.

C.1 Binary Readout

A binary readout is only capable of retrieving the hit/not-hit information of the pixels of a particular event. The effects of charge sharing allows to improve the particle position resolution, as the sensors have a non-zero depth.

For small η, the resolution in the y direction is between $0.3 \cdot \text{pitch}/\sqrt{12}$ and $0.9 \cdot \text{pitch}/\sqrt{12}$, depending on the azimutal component. Better resolution is possible with large ϕ, but this is not common in typical pixel detector layouts. For $\eta \approx 2$, instead, the upper and lower bounds become $0.85 \cdot \text{pitch}/\sqrt{12}$ and $0.6 \cdot \text{pitch}/\sqrt{12}$. Thus, the uncertainty decreases, but the average value increases.

In the azimutal direction, excluding the worst case corresponding to $\phi = 0$, which yields a resolution of $0.9 \cdot \text{pitch}/\sqrt{12}$, the resolution decreases as η increases, reaching $0.2 \cdot \text{pitch}/\sqrt{12}$ at $\eta = 2$.[103]

C.2 Charge Readout

If charge information is also available, instead, much better resolutions are achievable.

By omitting considerations on noise and cross-talk, a Neural Network trained for cluster recognition can achieve a resolution of $0.25 \cdot \text{pitch}/\sqrt{12}$ for $\eta = 2$, to $0.45 \cdot \text{pitch}/\sqrt{12}$ for $\eta = 0$ in the azimutal direction, and a resolution of $4.5 \cdot \text{pitch}/\sqrt{12}$ for various η in the polar direction.

But one of the main advantages in the availability of the charge information is the increased capability for multi-track cluster classification. The probability of recognizing a 2-particle cluster virtually becomes 1 if the full charge information is available and by using a properly trained Neural Network.

In addition to distinguishing the number of particles traversing a cluster, the ToT can be used to identify the particle type, which can be very important in the study for Long Lived Particles, or for identifying δ-rays.

Another important discussion point regards the digitization of the charge information. The number of ToT bits is directly related to the charge resolution, as are the extremes of the dynamic range. Non-linear encodings can help in reducing the number of charge codes and therefore of necessary bits, if some codes are more frequent, or relevant, than others.

Minimum ionizing particles (MIPs) are charged particles whose mean energy loss rate through matter is close to the minimum. The energy loss of a swift charged particle as it traverses matter is described by the Bethe-Block equation, which takes into account the interaction with the electrons of atoms in the material through ionization, leading to an energy loss of the traveling particle. Minimum ionization occurs when the kinetic energy of particles is at least twice larger than their rest mass.

Since the ionization losses of these particles are only weakly dependent on their momentum, it is generally accepted that a minimum ionizing particles produce an even distribution of free charge carriers along their paths. In silicon, a MIP averages a charge deposit of 80 electron-hole pairs per µm. In a pixel sensor of pixel size 50 µm × 50 µm and depth of 150 µm, this corresponds to a charge of about 2 fC. As typical pixels saturate at 50 fC of charge, 25 MIP usually represents the full scale of the detection.

As there is usually little information in charges detected above ToT_{MIP}, the midscale charge value ToT_{HALF} is usually set at a value close to ToT_{MIP}. But, as ToT_{MIP} varies for MIPs traversing the sensor at different ηs, choices often fall between $\text{ToT}_{\text{MIP@}\eta=0}$ and $\text{ToT}_{\text{MIP@}\eta=1}$.

By counting at 40 MHz, that is the Bunch Crossing Frequency, and thus the minimum reasonable clock to be distributed to the pixels, and with a 4-bit ToT linear encoding assumption, MIPs @ $\eta = 1$ average a pile-up inefficiency of about 0.6% for $\mathrm{ToT}_{\mathrm{MIP}@\eta=0}$, far below the 1.2% there would be for $\mathrm{ToT}_{\mathrm{MIP}@\eta=1}$. In a double edge clock counter is used, these values diminish to 0.2% and 0.5% respectively. To achieve 0.1% inefficiency, counting should be done at 150 MHz in the $\eta = 0$ case, and 270 MHz in the $\eta = 1$ case.

As the pratical goal is to use the minimum number a ToT bits to encode the charges, and still preserve as much information as possible, studies were performed to determine such a value.

As it happens, 4 bits of linearly-encoded ToT, with $\mathrm{ToT}_{\mathrm{HALF}}$ set between $\mathrm{ToT}_{\mathrm{MIP}@\eta=0}$ and $\mathrm{ToT}_{\mathrm{MIP}@\eta=1}$, assure a good separation power for multi-track clusters at $\eta = 1$ (1% below the maximum of 0.78), and a position resolution that is at most 12% more that the minimum resolution possible (achieved with infinite ToT bits). There is little to no gain for higher ToT values, although if $\mathrm{ToT}_{\mathrm{HALF}}$ shifts to $\mathrm{ToT}_{\mathrm{MIP}@\eta=1}$, the resolution can be improved to 4% and below more than the minimum.

TOTs, however, can be compressed as well, in order to take advantage of the non-uniformity of ToT distribution. Code efficiency (entropy) increases as the $\mathrm{ToT}_{\mathrm{HALF}}$ approaches $\mathrm{ToT}_{\mathrm{MIP}}$ as expected, but for a 4-bit ToT, at an appropriate $\mathrm{ToT}_{\mathrm{HALF}}$, the efficiency is almost 1 regardless of the compression method. Some small gain can be achieved in the ToT overflow avoidance, but nothing significant can be gained in code utilization. [17]

In conclusion, a scheme utilizing 4 bits, with a counter of 80 MHz or more, and an appropriately chosen $\mathrm{ToT}_{\mathrm{HALF}}$ can achieve the best performance/cost compromise.

Appendix D

Trigger Matching

In this Appendix, a quick review of various trigger matching implementations is presented. Moreover, some common options discussed, along with a few optimizations.

D.1 Implementation

Trigger matching can be performed in several ways, depending on where and how the timestamp counter is implemented.

Dedicated Timestamp Counter One of the most elementary solutions involves the use of a dedicated timestamp counter in the buffer rows of the pixels/Pixel Regions. If the comparator logic is asynchronous, the counter could be implemented as a ripple counter in order to reduce the area overhead and power consumption.

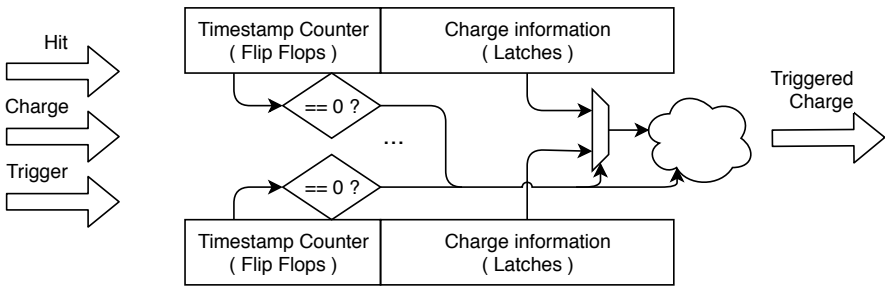

Figure D.1: Scheme of the Dedicated Timestamp Counter implementation for trigger matching in the Pixel/Pixel Region

Fig. D.1 also shows the prototype of the logic needed to perform the trigger timestamp comparison. In the figure, a descending counter was supposed: when an event

is recorded, the counter is set to the trigger latency value (in clock cycles), and programmed to decrease its value every clock cycle. When it reaches zero, it tells the output logic and propagates its charge information to it.

The trigger latency can be propagated to the Pixel Regions by a bus from the Chip Periphery. The evaluation in the Pixel Regions can be made by comparing the counter's value with this propagated trigger latency, if the counter is ascending, or by using the bus as the reset value of the counter, if it is descending.

One of the main drawbacks of this solution is the area occupancy and the power consumption. This solution, in fact, involves continuous switching of the flip flops implementing the timestamp counters.

Shared Timestamp Counter In an effort to reduce the area and power consumption, the ever-switching flip-flops of the counters could be replaced with latches which store the timestamp evaluated by a memory-wise timestamp counter. In this way, there is only one active counter, whose value can be latched upon event detection in a row, and thereafter waiting for the trigger.

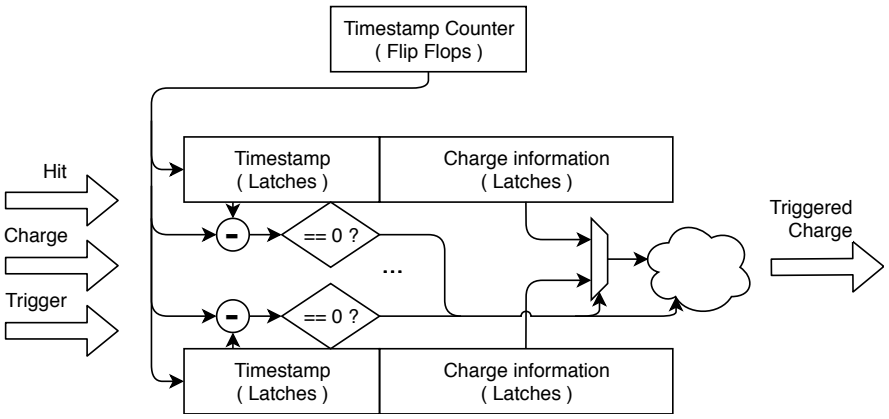

Figure D.2: Scheme of the Shared Timestamp Counter implementation for trigger matching in the Pixel/Pixel Region

Fig. D.2 shows an hypothetical implementation of such a scheme. If the counter is ascending, it must be set to overflow to 0 when it reaches the trigger latency in Click Cycles (CCs). Elsewise, if a descending counter is employed, it must be set to underflow to the trigger latency in CCs when it reaches 0. When an event is recorded, the current value of the timestamp counter is latched in the timestamp latches of the selected row. The implementation in the figure proposes the use of a subtractor and a zero check.

The counter type to be employed should be a synchronous one (based on Full Adders), in order to reduce the impact of Single Event Transients (SETs) and avoid timing problems.

Peripheral Timestamp Counter A variant from the previous solution consists in the relocation of the timestamp counter from the pixel region to the Chip Periphery. Such action would remove an energy-intensive block from being repeated multiple times in the matrix, and have it only once in the Periphery. This case is shown in Fig. D.3. Among the advantages are the timestamp consistency across the matrix (and, so, across the whole chip), and the additional SEU precautions which could be put in place, such as triplication, as the energetic and area impact would be negligible if compared to the implementation of such measures in the Pixel Regions. To further decrease the power consumption, the timestamp could be propagated in Gray code, which involves the switching of only one bit at a time per word.

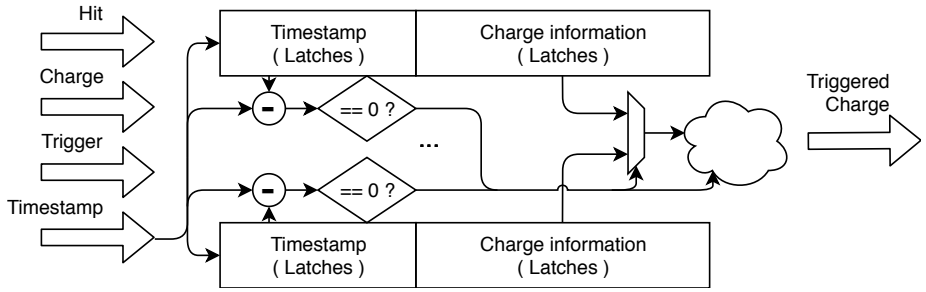

Figure D.3: Scheme of the Peripheral Timestamp Counter implementation for trigger matching in the Pixel/Pixel Region

For these reasons, this implementation is the most used in pixel chips, and is the one used in the ROCs described in this thesis.

In order to make the trigger latency programmable, it becomes necessary to propagate a second timestamp bus to the matrix, containing the timestamp of the triggered event. That is, essentially, the timestamp minus the trigger latency in CCs, as shown in Fig. D.4.

The comparison in the memory cells must therefore be made against this trigger timestamp, and not the actual BX timestamp used during the writing operation. This solution is more robust and allows for flexibility.

Single Timestamp bus The trigger timestamp, at the moment propagated as a separate bus, could be removed if the timestamp bus is used for both event buffering and trigger matching. In this case, the timestamp counter must be set to overflow after the trigger latency, in a way that if an event is saved with timestamp A, the next time that the timestamp is equal to A is will be when the trigger latency has elapsed. The trigger timestamp comparison would hence be done against the timestamp itself: if a trigger signal arrives, then the data is triggered; otherwise, discarded.

The main drawback of this solution is that a standard Gray encoding is designed to

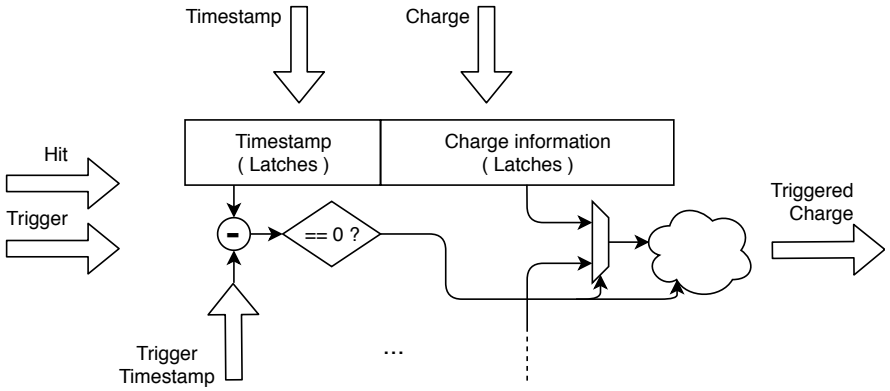

Figure D.4: Scheme of the Peripheral Timestamp Counter implementation for trigger matching with programmable latency in the Pixel/Pixel Region

map only power of 2 numbers, and would therefore not be suitable for any non power-of-2 trigger latencies.

However, the Gray code is not the only code which has the property of having unit Hamming-distance steps: an alternative would be a quasi-Gray encoding, which involves trimming the extremes from a Gray-code sequence. The Gray code, in fact, has the property of being composed (recursively) of 2 mirrored sequences, with the MSB switched in sign. By removing some entries from both extremes (or the middle), the resulting code would preserve the required property, while having the only requirement of being even in the number of entries.

This means that, by rounding up any odd trigger latency, it is possible to still use the Gray encoding with a programmable trigger latency and unique timestamp bus.

Appendix E

Zero suppression

Zero suppression is a data compression technique which consists in the removal of redundant zeroes from a number. Although usually introducing an overhead, the zero-suppression operation removes any non-coding cypher from a dataset. In a HEP experiment, this is a fundamental technique, as it is key to avoid the transmission of non-hit pixels in an event, and only sending the useful data out of the ROC.

The zero suppression compression rate is greater when the occupancy is low: this is the trend in next generation pixel detectors, where the pixel sizes are shrinking significantly with respect to the previous HEP detectors. As it takes a non-negligible amount of computing power and area, it has traditionally been implemented in the ROCs' periphery.

The first proposal for zero-suppression in HEP circuits goes back to the late 80s, with a zero-suppressing circuit for analog pulses. [12] In the ALICE Muon Tracker, for example, the ReadOut chip, in Towerjazz 180nm technology, works in rolling shutter mode, performs online cluster detection in the periphery and cluster-wise zero-suppression as shown in [33]. Another early example can be found in a CMOS ASIC for a MAPS detector, in which the use of sparse data scan in the periphery, in a fast readout architecture, reduces the detector data of a factor of 10 to 1000. [43, 48]. More complex and efficient solutions delegate zero-suppression to FPGAs, as in the case of a proposed cluster-based algorithm paper for LHCb. [16]

A simpler architecture proposal for LHCb upgrade started by studying the effects of zero-suppression and compression in the pixel matrix. By pairing pixels up, the architecture allowed for a saving of 21% in the data packets: when adjacent pixels are hit and are selected for readout, the Time of Arrival information is sent out only once. Further sorting and compression brings the total compression rate up to 30%. This zero-suppression technique in embedded in the chip architecture, as pixels are functionally divided in pairs, and these pairs are read out together. Only even pixels have both a ToA memory and a ToT memory, while odd pixels only have a ToT memory. In case an odd pixel is hit, it must pair up with an even neighbor for readout. [47]

Bibliography

[1] G et al. Aad. "ATLAS pixel detector electronics and sensors". In: *JINST* 3 (2008), P07007. URL: https://cds.cern.ch/record/1119279.

[2] Betty Bezverkhny Abelev et al. "Performance of the ALICE Experiment at the CERN LHC". In: *International Journal of Modern Physics* A29 (2014), p. 1430044. DOI: 10.1142/S0217751X14300440. arXiv: 1402.4476 [nucl-ex].

[3] G Anellia Wyllie et al. "Front-end pixel chips for tracking in ALICE and particle identification in LHCb". In: (Jan. 2002).

[4] "ATLAS trigger menu and performance in Run 1 and prospects for Run 2". In: *2013 IEEE Nuclear Science Symposium and Medical Imaging Conference (2013 NSS/MIC)*. IEEE, Oct. 2013. DOI: 10.1109/nssmic.2013.6829554. URL: https://doi.org/10.1109/nssmic.2013.6829554.

[5] R. Ballabriga et al. "Medipix3: A 64k pixel detector readout chip working in single photon counting mode with improved spectrometric performance". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 633 (2011). 11th International Workshop on Radiation Imaging Detectors (IWORID), S15–S18. ISSN: 0168-9002. DOI: https://doi.org/10.1016/j.nima.2010.06.108. URL: http://www.sciencedirect.com/science/article/pii/S0168900210012982.

[6] Rafael Ballabriga, Michael Campbell, and Xavier Llopart. "Asic developments for radiation imaging applications: The medipix and timepix family". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 878 (2018). Radiation Imaging Techniques and Applications, pp. 10–23. ISSN: 0168-9002. DOI: https://doi.org/10.1016/j.nima.2017.07.029. URL: http://www.sciencedirect.com/science/article/pii/S0168900217307714.

[7] R Ballabriga et al. "Characterization of the Medipix3 pixel readout chip". In: *Journal of Instrumentation* 6.01 (Jan. 2011), pp. C01052–C01052. DOI: 10.1088/1748-0221/6/01/c01052.

[8] William Balunas. "ATLAS Trigger and Data Acquisition Upgrades for High Luminosity LHC". In: Feb. 2017, p. 858. DOI: 10.22323/1.282.0858.

[9] M. Barbero et al. “Design and test of the CMS pixel readout chip”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 517.1 (2004), pp. 349–359. ISSN: 0168-9002. DOI: https://doi.org/10.1016/j.nima.2003.09.043. URL: http://www.sciencedirect.com/science/article/pii/S0168900203026391.

[10] Roland Baur. “Readout architecture of the CMS pixel detector”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 465.1 (2001). SPD2000, pp. 159–165. ISSN: 0168-9002. DOI: https://doi.org/10.1016/S0168-9002(01)00382-5. URL: http://www.sciencedirect.com/science/article/pii/S0168900201003825.

[11] S. Bonacini et al. “Characterization of a commercial 65 nm CMOS technology for SLHC applications”. In: *JINST* 7 (2012), P01015. DOI: 10.1088/1748-0221/7/01/P01015.

[12] F. Bourgeois. “Proposal for a fast, zero suppressing circuit for the digitization of analog pulses over long memory times”. In: *Nuclear Instruments and Methods in Physics Research* 219.1 (1984), pp. 153–159. ISSN: 0167-5087. DOI: https://doi.org/10.1016/0167-5087(84)90148-0. URL: http://www.sciencedirect.com/science/article/pii/0167508784901480.

[13] Rebecca Carney et al. “Results of FE65-P2 Pixel Readout Test Chip for High Luminosity LHC Upgrades”. In: *PoS* ICHEP2016 (2016), p. 272. DOI: 10.22323/1.282.0272.

[14] L.M. Jara Casas et al. “Characterization of radiation effects in 65 nm digital circuits with the DRAD digital radiation test chip”. In: *Journal of Instrumentation* 12.02 (2017), p. C02039. URL: http://stacks.iop.org/1748-0221/12/i=02/a=C02039.

[15] G. Cesura et al. “New pixel detector concepts based on junction field effect transistors on high resistivity silicon”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 377.2 (1996). Proceedings of the Seventh European Symposium on Semiconductor, pp. 521–528. ISSN: 0168-9002. DOI: https://doi.org/10.1016/0168-9002(96)00236-7. URL: http://www.sciencedirect.com/science/article/pii/0168900296002367.

[16] H. Chanal. “A zero-suppression algorithm for the readout electronics of the SciFi Tracker for the LHCb detector upgrade”. In: *Journal of Instrumentation* 11.02 (2016), p. C02073. URL: http://stacks.iop.org/1748-0221/11/i=02/a=C02073.

[17] Yitian Chen et al. "Optimal use of charge information for the HL-LHC pixel detector readout". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 902 (2018), pp. 197–210. ISSN: 0168-9002. DOI: https://doi.org/10.1016/j.nima.2018.01.091.

[18] D.C Christian et al. "Development of a pixel readout chip for BTeV". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 435.1 (1999), pp. 144–152. ISSN: 0168-9002. DOI: https://doi.org/10.1016/S0168-9002(99)00421-0. URL: http://www.sciencedirect.com/science/article/pii/S0168900299004210.

[19] The CMS Collaboration et al. "The CMS experiment at the CERN LHC". In: *Journal of Instrumentation* 3 (Aug. 2008), S08004. DOI: 10.1088/1748-0221/3/08/S08004.

[20] CMS Collaboration. *The Phase-2 Upgrade of the CMS L1 Trigger Interim Technical Design Report*. Tech. rep. CERN-LHCC-2017-013. CMS-TDR-017. This is the CMS Interim TDR devoted to the upgrade of the CMS L1 trigger in view of the HL-LHC running, as approved by the LHCC. Geneva: CERN, Sept. 2017. URL: https://cds.cern.ch/record/2283192.

[21] CMS Collaboration. *The Phase-2 Upgrade of the CMS Tracker*. Tech. rep. CERN-LHCC-2017-009. CMS-TDR-014. Geneva: CERN, June 2017. URL: https://cds.cern.ch/record/2272264.

[22] The CMS Collaboration. "The CMS trigger system". In: *Journal of Instrumentation* 12.01 (2017), P01020. URL: http://stacks.iop.org/1748-0221/12/i=01/a=P01020.

[23] Elia Conti et al. "Development of a Large Pixel Chip Demonstrator in RD53 for ATLAS and CMS Upgrades". In: *Topical Workshop on Electronics for Particle Physics*. Vol. TWEPP-17. SISSA, 2017, p. 005. DOI: 10.22323/1.313.0005. URL: https://pos.sissa.it/313/005/pdf.

[24] Michel Della Negra, Peter Jenni, and Tejinder S. Virdee. "The Construction of ATLAS and CMS". In: *Annual Review of Nuclear and Particle Science* 68.1 (2018), pp. 183–209. DOI: 10.1146/annurev-nucl-101917-021038.

[25] N. Demaria et al. "Recent progress of RD53 Collaboration towards next generation Pixel Read-Out Chip for HL-LHC". In: *8th International Workshop on Semiconductor Pixel Detectors for Particles and Imaging*. Vol. 11. 12. Sestri Levante, Italy, 2016, p. C12058. DOI: 10.1088/1748-0221/11/12/C12058. URL: http://lss.fnal.gov/archive/2016/conf/fermilab-conf-16-622-ppd.pdf.

[26] N. Demaria et al. "CHIPIX65: Developments on a new generation pixel readout ASIC in CMOS 65 nm for HEP experiments". In: *2015 6th International Workshop on Advances in Sensors and Interfaces (IWASI)*. June 2015, pp. 49–54. DOI: 10.1109/IWASI.2015.7184947.

[27] Roberto Dinapoli et al. "An analog front-end in standard 0.25um CMOS for silicon pixel detectors in ALICE and LHCb". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* (Jan. 2000).

[28] Roberto Dinapoli et al. "EIGER: Next generation single photon counting detector for X-ray applications". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 650.1 (2011). International Workshop on Semiconductor Pixel Detectors for Particles and Imaging 2010, pp. 79–83. ISSN: 0168-9002. DOI: https://doi.org/10.1016/j.nima.2010.12.005. URL: http://www.sciencedirect.com/science/article/pii/S0168900210027427.

[29] R Dinapoli et al. "MÖNCH, a small pitch, integrating hybrid pixel detector for X-ray applications". In: *Journal of Instrumentation* 9.05 (May 2014), pp. C05015–C05015. DOI: 10.1088/1748-0221/9/05/c05015. URL: https://doi.org/10.1088%2F1748-0221%2F9%2F05%2Fc05015.

[30] "Discriminators in 65 nm CMOS process for high granularity, high time resolution pixel detectors". In: *2013 IEEE Nuclear Science Symposium and Medical Imaging Conference (2013 NSS/MIC)*. IEEE, Oct. 2013. DOI: 10.1109/nssmic.2013.6829777. URL: https://doi.org/10.1109/nssmic.2013.6829777.

[31] W. Erdmann. "The 0.25um front-end for the CMS pixel detector". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 549.1 (2005). VERTEX 2003, pp. 153–156. ISSN: 0168-9002. DOI: https://doi.org/10.1016/j.nima.2005.04.044. URL: http://www.sciencedirect.com/science/article/pii/S0168900205009848.

[32] W. Erdmann. "The DMILL readout chip for the CMS pixel detector". In: *Semiconductor pixel detectors for particles and X-rays. Proceedings, International Workshop, PIXEL2002, Carmel, USA, September 9-12, 2002*. 2002. URL: http://www.slac.stanford.edu/econf/C020909/.

[33] C. Flouzat et al. "Zero suppression logic of the ALICE muon forward tracker pixel chip prototype PIXAM and associated readout electronics development". In: *JINST* 10.05 (2015), p. C05012. DOI: 10.1088/1748-0221/10/05/C05012.

[34] E.R. Fossum. "CMOS image sensors: electronic camera on a chip". In: vol. 44. Jan. 1995, pp. 17–25. ISBN: 0-7803-2700-4. DOI: 10.1109/IEDM.1995.497174.

[35] L. Gaioni et al. “Design of analog front-ends for the RD53 demonstrator chip”. In: *25th International Workshop on Vertex Detectors*. Vol. Vertex 2016. La Biodola, Italy: SISSA, 2017, p. 036. DOI: 10.22323/1.287.0036. URL: https://pos.sissa.it/archive/conferences/287/036/Vertex%5C%202016%5C_036.pdf.

[36] L. Gaioni et al. “Heavily Irradiated 65-nm Readout Chip With Asynchronous Channels for Future Pixel Detectors”. In: *IEEE Trans. Nucl. Sci.* 65.10 (2018), pp. 2699–2706. DOI: 10.1109/TNS.2018.2871245.

[37] M. Garcia-Sciveres. “ATLAS Experiment Pixel Detector Upgrades”. In: (2011). arXiv: 1109.4662 [physics.ins-det].

[38] M. Garcia-Sciveres, A. Mekkaoui, and D. Ganani. “Towards third generation pixel readout chips”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 731 (2013). PIXEL 2012, pp. 83–87. ISSN: 0168-9002. DOI: https://doi.org/10.1016/j.nima.2013.04.023. URL: http://www.sciencedirect.com/science/article/pii/S0168900213004191.

[39] Maurice Garcia-Sciveres. *The RD53A Integrated Circuit*. Tech. rep. CERN-RD53-PUB-17-001. Geneva: CERN, Oct. 2017. URL: https://cds.cern.ch/record/2287593.

[40] Maurice Garcia-Sciveres and Norbert Wermes. “A review of advances in pixel detectors for experiments with high rate and radiation”. In: *Reports on Progress in Physics* 81.6 (2018), p. 066101. URL: http://stacks.iop.org/0034-4885/81/i=6/a=066101.

[41] Mauricio Garcia-Sciveres. *RD53A Integrated Circuit Specifications*. Tech. rep. CERN-RD53-PUB-15-001. Geneva: CERN, Dec. 2015. URL: https://cds.cern.ch/record/2113263.

[42] J A Gray. “The CMS Phase-1 pixel detector”. In: *Journal of Instrumentation* 8.12 (2013), p. C12047. URL: http://stacks.iop.org/1748-0221/8/i=12/a=C12047.

[43] Ch Hu-Guo et al. “CMOS pixel sensor development: a fast read-out architecture with integrated zero suppression”. In: *Journal of Instrumentation* 4.04 (2009), P04012. URL: http://stacks.iop.org/1748-0221/4/i=04/a=P04012.

[44] V. Halyo, P. LeGresley, and P. Lujan. “Massively Parallel Computing and the Search for Jets and Black Holes at the LHC”. In: *Nucl. Instrum. Methods Phys. Res., A* 744.arXiv:1309.6275 (Sept. 2013), 54–60. 7 p. URL: http://cds.cern.ch/record/1603000.

[45] T. Hemperek et al. “Digital architecture of the new ATLAS pixel chip FE-I4”. In: *2009 IEEE Nuclear Science Symposium Conference Record (NSS/MIC)*. Oct. 2009, pp. 791–796. DOI: 10.1109/NSSMIC.2009.5402304.

[46] W. Herr and B. Muratori. "Concept of luminosity". In: *Intermediate accelerator physics. Proceedings, CERN Accelerator School, Zeuthen, Germany, September 15-26, 2003*. 2003, pp. 361–377. URL: http://doc.cern.ch/yellowrep/2006/2006-002/p361.pdf.

[47] S. Heuvelmans and M. Boerrigter. "A pixel read-out architecture implementing a two-stage token ring, zero suppression and compression". In: *JINST* 6 (2011), p. C01093. DOI: 10.1088/1748-0221/6/01/C01093.

[48] A. Himmi et al. "A Zero Suppression Micro-Circuit for Binary Readout CMOS Monolithic Sensors". In: *Proceedings, Topical Workshop on Electronics for Particle Physics (TWEPP09)*. CERN. CERN, 2009. DOI: 10.5170/CERN-2009-006.426.

[49] J Hunen et al. "Irradiation and SPS beam tests of the ALICE1LHCb pixel chip". In: (Oct. 2001). DOI: 10.5170/CERN-2001-005.85.

[50] Paul E. Karchin. "Use of pixel detectors in elementary particle physics". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 305.3 (1991), pp. 497–503. ISSN: 0168-9002. DOI: https://doi.org/10.1016/0168-9002(91)90148-J. URL: http://www.sciencedirect.com/science/article/pii/016890029190148J.

[51] V. Karimäki. "The CMS tracker system project". In: (1997). Ed. by M. Mannelli et al.

[52] H.Chr. Kästli et al. "Design and performance of the CMS pixel detector readout chip". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 565.1 (2006). Proceedings of the International Workshop on Semiconductor Pixel Detectors for Particles and Imaging, pp. 188–194. ISSN: 0168-9002. DOI: https://doi.org/10.1016/j.nima.2006.05.038. URL: http://www.sciencedirect.com/science/article/pii/S0168900206007674.

[53] Christopher J. Kenney et al. "A prototype monolithic pixel detector". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 342.1 (1994), pp. 59–77. ISSN: 0168-9002. DOI: https://doi.org/10.1016/0168-9002(94)91411-7. URL: http://www.sciencedirect.com/science/article/pii/0168900294914117.

[54] Vardan Khachatryan et al. "The CMS trigger system". In: *JINST* 12.01 (2017), P01020. DOI: 10.1088/1748-0221/12/01/P01020. arXiv: 1609.02366 [physics.ins-det].

[55] Hyun-Sik Kim et al. "A sampling-based 128x128 direct photon-counting X-ray image sensor with 3 energy bins and spatial resolution of 60um/pixel". In: *2012 IEEE International Solid-State Circuits Conference*. IEEE, Feb. 2012. DOI: 10.1109/isscc.2012.6176941. URL: https://doi.org/10.1109/isscc.2012.6176941.

[56] F. Krummenacher. "Pixel detectors with local intelligence: an IC designer point of view". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 305.3 (1991), pp. 527–532. ISSN: 0168-9002. DOI: https://doi.org/10.1016/0168-9002(91)90152-G. URL: http://www.sciencedirect.com/science/article/pii/016890029190152G.

[57] Volker Lindenstruth and Ivan Kisel. "Overview of trigger systems". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 535.1 (2004). Proceedings of the 10th International Vienna Conference on Instrumentation, pp. 48–56. ISSN: 0168-9002. DOI: https://doi.org/10.1016/j.nima.2004.07.267. URL: http://www.sciencedirect.com/science/article/pii/S0168900204015748.

[58] X. Llopart et al. "Medipix2: A 64-k pixel readout chip with 55-μm square elements working in single photon counting mode". In: *IEEE Transactions on Nuclear Science* 49.5 (Oct. 2002), pp. 2279–2283. DOI: 10.1109/tns.2002.803788. URL: https://doi.org/10.1109/tns.2002.803788.

[59] X. Llopart et al. "Timepix, a 65k programmable pixel readout chip for arrival time, energy and/or photon counting measurements". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 581.1 (2007). VCI 2007, pp. 485–494. ISSN: 0168-9002. DOI: https://doi.org/10.1016/j.nima.2007.08.079. URL: http://www.sciencedirect.com/science/article/pii/S0168900207017020.

[60] S. Marconi et al. "The RD53 Collaboration's SystemVerilog-UVM Simulation Framework and its General Applicability to Design of Advanced Pixel Readout Chips". In: *JINST* 9.10 (2014), P10005. DOI: 10.1088/1748-0221/9/10/P10005. arXiv: 1408.3232 [physics.ins-det].

[61] A Mekkaoui, M Garcia-Sciveres, and D Gnani. "Results of 65 nm pixel readout chip demonstrator array". In: *Journal of Instrumentation* 8.01 (Jan. 2013), pp. C01055–C01055. DOI: 10.1088/1748-0221/8/01/c01055.

[62] M. Menouni et al. "1-Grad total dose evaluation of 65 nm CMOS technology for the HL-LHC upgrades". In: *JINST* 10.05 (2015), p. C05009. DOI: 10.1088/1748-0221/10/05/C05009.

[63] E. Migliore. "CMS Pixel Detector design for HL-LHC". In: *Journal of Instrumentation* 11.12 (2016), p. C12061. URL: http://stacks.iop.org/1748-0221/11/i=12/a=C12061.

[64] E. Monteil et al. "A prototype of a new generation readout ASIC in 65nm CMOS for pixel detectors at HL-LHC". In: *Journal of Instrumentation* 11.12 (Dec. 2016), pp. C12044–C12044. DOI: 10.1088/1748-0221/11/12/c12044.

[65] E. Monteil et al. "A synchronous analog very front-end in 65 nm CMOS with local fast ToT encoding for pixel detectors at HL-LHC". In: *Journal of Instrumentation* 12.03 (Mar. 2017), pp. C03066–C03066. DOI: 10.1088/1748-0221/12/03/c03066.

[66] Aristeidis Nikolaou et al. "Extending a 65nm CMOS process design kit for high total ionizing dose effects". In: *Proceedings, 7th International Conference on Modern Circuits and Systems Technologies (MOCAST 2018): Thessaloniki, Greece, May 7-9, 2018*. 2018, p. 8376561. DOI: 10.1109/MOCAST.2018.8376561.

[67] Luca Pacher et al. "A Prototype of a New Generation Readout ASIC in 65 nm CMOS for Pixel Detectors at HL-LHC". In: *25th International Workshop on Vertex Detectors*. Vol. Vertex2016. La Biodola, Italy: SISSA, 2017, p. 054. DOI: 10.22323/1.287.0054. URL: https://pos.sissa.it/archive/conferences/287/054/Vertex%5C%202016%5C_054.pdf.

[68] Luca Pacher et al. "Results from CHIPIX-FE0, a small-scale prototype of a new generation pixel readout ASIC in 65 nm CMOS for HL-LHC". In: *Topical Workshop on Electronics for Particle Physics*. Vol. TWEPP-17. SISSA, 2017, p. 024. DOI: 10.22323/1.313.0024. URL: https://pos.sissa.it/313/024/pdf.

[69] Luca Pacher et al. "Results from CHIPIX-FE0, a Small-Scale Prototype of a New Generation Pixel Readout ASIC in 65 nm CMOS for HL-LHC". In: *Proceedings of Topical Workshop on Electronics for Particle Physics — PoS(TWEPP-17)*. Sissa Medialab, Mar. 2018. DOI: 10.22323/1.313.0024. URL: https://doi.org/10.22323/1.313.0024.

[70] S. Panati et al. "First measurements of a prototype of a new generation pixel readout ASIC in 65 nm CMOS for extreme rate HEP detectors at HL-LHC". In: *2016 IEEE Nuclear Science Symposium, Medical Imaging Conference and Room-Temperature Semiconductor Detector Workshop (NSS/MIC/RTSD)*. Oct. 2016, pp. 1–7. DOI: 10.1109/NSSMIC.2016.8069857.

[71] A. Paternó. "Design of Digital Readout Electronics for Pixel Detectors in 65nm CMOS Technology". Master's Thesis. Politecnico di Torino, 2015.

[72] Andrea Paternò et al. "A prototype of pixel readout ASIC in 65 nm CMOS technology for extreme hit rate detectors at HL-LHC". In: *Topical Workshop on Electronics for Particle Physics*. Vol. 12. 02. Karlsruhe, Germany, 2017, p. C02043. DOI: 10.1088/1748-0221/12/02/C02043.

[73] Andrea Paternò et al. "New development on digital architecture for efficient pixel readout ASIC at extreme hit rate for HEP detectors at HL-LHC". In: *2016 IEEE Nuclear Science Symposium and Medical Imaging Conference*. Strasbourg, France, 2016, p. 8069855. DOI: 10.1109/NSSMIC.2016.8069855.

[74] Ivan Perić et al. "The FEI3 readout chip for the ATLAS pixel detector". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 565.1 (2006). Proceedings of the International Workshop on Semiconductor Pixel Detectors for Particles and Imaging, pp. 178–187. ISSN: 0168-9002. DOI: https://doi.org/10.1016/j.nima.2006.05.032. URL: http://www.sciencedirect.com/science/article/pii/S0168900206007649.

[75] T. Poikela et al. "VeloPix: the pixel ASIC for the LHCb upgrade". In: *Journal of Instrumentation* 10.01 (Jan. 2015), pp. C01057–C01057. DOI: 10.1088/1748-0221/10/01/c01057.

[76] T Poikela et al. "The VeloPix ASIC". In: *Journal of Instrumentation* 12 (Jan. 2017), pp. C01070–C01070. DOI: 10.1088/1748-0221/12/01/C01070.

[77] T Poikela et al. "Timepix3: a 65K channel hybrid pixel readout chip with simultaneous ToA/ToT and sparse readout". In: *Journal of Instrumentation* 9.05 (May 2014), pp. C05013–C05013. DOI: 10.1088/1748-0221/9/05/c05013.

[78] *Public ATLAS Luminosity Results*. Mar. 17, 2019. URL: https://twiki.cern.ch/twiki/bin/view/AtlasPublic/LuminosityPublicResultsRun2.

[79] *Public CMS Luminosity Results*. Aug. 24, 2018. URL: https://twiki.cern.ch/twiki/bin/view/AtlasPublic/LuminosityPublicResults.

[80] *Public CMS Luminosity Results*. Mar. 19, 2019. URL: https://twiki.cern.ch/twiki/bin/view/CMSPublic/LumiPublicResults#Technical_details.

[81] V Radicci et al. "EIGER a new single photon counting detector for X-ray applications: performance of the chip". In: *Journal of Instrumentation* 7.02 (Feb. 2012), pp. C02019–C02019. DOI: 10.1088/1748-0221/7/02/c02019.

[82] M. Ramilli et al. "Measurements with MÖNCH, a 25 µm pixel pitch hybrid pixel detector". In: *Journal of Instrumentation* 12.01 (Jan. 2017), pp. C01071–C01071. DOI: 10.1088/1748-0221/12/01/c01071. URL: https://doi.org/10.1088%2F1748-0221%2F12%2F01%2Fc01071.

[83] L. Ratti et al. "An asynchronous front-end channel for pixel detectors at the HL-LHC experiment upgrades". In: *Proceedings, 2015 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC 2015): San Diego, California, United States*. 2016, p. 7581981. DOI: 10.1109/NSSMIC.2015.7581981.

[84] Lodovico Ratti et al. "65-nm CMOS front-end channel for pixel readout in the HL-LHC radiation environment". In: *IEEE Trans. Nucl. Sci.* 64.12 (2017), pp. 2922–2932. DOI: 10.1109/TNS.2017.2771506.

[85] John Richardson. "The ATLAS pixel front-end readout chips". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 473.1 (2001). Proceedings of the 9th International Workshop on Vertex Detectors, pp. 157–162. ISSN: 0168-9002. DOI: https://doi.org/10.1016/S0168-9002(01)01138-X. URL: http://www.sciencedirect.com/science/article/pii/S016890020101138X.

[86] Angelo Rivetti. *CMOS: Front-End Electronics for Radiation Sensors (Devices, Circuits, and Systems)*. CRC Press, 2015. ISBN: 9781466563100. URL: https://www.amazon.com/CMOS-Front-End-Electronics-Radiation-Circuits/dp/1466563109.

[87] G. De Robertis et al. "Design of a 10-bit segmented current-steering digital-to-analog converter in CMOS 65 nm technology for the bias of new generation readout chips in high radiation environment". In: *Journal of Instrumentation* 11.01 (Jan. 2016), pp. C01027–C01027. DOI: 10.1088/1748-0221/11/01/c01027. URL: https://doi.org/10.1088/1748-0221/11/01/c01027.

[88] Dave Robinson. "ATLAS Tracker and Pixel Operational Experience". In: *PoS* Vertex2016 (2017), p. 005. DOI: 10.22323/1.287.0005.

[89] T. Rohe et al. "Radiation hardness of CMS pixel barrel modules". In: *Nucl. Instrum. Meth.* A624 (2010), pp. 414–418. DOI: 10.1016/j.nima.2010.03.157. arXiv: 1001.0666 [physics.ins-det].

[90] Leonardo Rossi et al. *Pixel Detectors: From Fundamentals to Applications*. Springer, 2006. ISBN: 3540283323.

[91] Leonardo Rossi et al. *Pixel Detectors: From Fundamentals to Applications (Particle Acceleration and Detection)*. Springer, 2006. ISBN: 3540283323. URL: https://www.amazon.com/Pixel-Detectors-Fundamentals-Applications-Acceleration/dp/3540283323.

[92] Anirban Saha. "Phase 1 upgrade of the CMS pixel detector". In: *Journal of Instrumentation* 12.02 (2017), p. C02033. URL: http://stacks.iop.org/1748-0221/12/i=02/a=C02033.

[93] Albert M Sirunyan et al. "Precision measurement of the structure of the CMS inner tracking system using nuclear interactions". In: *JINST* 13.10 (2018), P10034. DOI: 10.1088/1748-0221/13/10/P10034. URL: http://cms-results.web.cern.ch/cms-results/public-results/publications/TRK-17-001.

[94] Walter Snoeys et al. “First beam test results from a monolithic silicon pixel detector”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 326.1 (1993), pp. 144–149. ISSN: 0168-9002. DOI: https://doi.org/10.1016/0168-9002(93)90344-H. URL: http://www.sciencedirect.com/science/article/pii/016890029390344H.

[95] Andrey Starodumov, P. Berger, and M. Meinhard. “High rate capability and radiation tolerance of the PROC600 readout chip for the CMS pixel detector”. In: *JINST* 12.01 (2017), p. C01078. DOI: 10.1088/1748-0221/12/01/C01078.

[96] A Tapper and Darin Acosta. *CMS Technical Design Report for the Level-1 Trigger Upgrade*. Tech. rep. CERN-LHCC-2013-011. CMS-TDR-12. June 2013. URL: https://cds.cern.ch/record/1556311.

[97] G. Tinti et al. “Similarities and differences of recent hybrid pixel detectors for X-ray and high energy physics developed at the Paul Scherrer Institut”. In: *Journal of Instrumentation* 10.04 (2015), p. C04043. URL: http://stacks.iop.org/1748-0221/10/i=04/a=C04043.

[98] L Tlustos and Erik H M Heijne. “Performance and limitations of high granularity single photon processing X-ray imaging detectors”. Presented on 1 Apr 2005. 2005. URL: https://cds.cern.ch/record/846447.

[99] Mia Tosi. *The CMS trigger in Run 2*. Tech. rep. CMS-CR-2017-340. Geneva: CERN, Oct. 2017. URL: https://cds.cern.ch/record/2290106.

[100] G. Traversi et al. “Characterization of bandgap reference circuits designed for high energy physics applications”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 824 (2016). Frontier Detectors for Frontier Physics: Proceedings of the 13th Pisa Meeting on Advanced Detectors, pp. 371–373. ISSN: 0168-9002. DOI: https://doi.org/10.1016/j.nima.2015.09.103. URL: http://www.sciencedirect.com/science/article/pii/S0168900215011924.

[101] R Turchetta et al. “A monolithic active pixel sensor for charged particle tracking and imaging using standard VLSI CMOS technology”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 458.3 (2001), pp. 677–689. ISSN: 0168-9002. DOI: https://doi.org/10.1016/S0168-9002(00)00893-7. URL: http://www.sciencedirect.com/science/article/pii/S0168900200008937.

[102] Tamara Vazquez Schroeder. “The ATLAS Trigger in Run 2: Design, Menu, and Performance”. In: *PoS* EPS-HEP2017 (2017), p. 525. DOI: 10.22323/1.314.0525.

[103] Fuyue Wang, Benjamin Nachman, and Maurice Garcia-Sciveres. "Ultimate position resolution of pixel clusters with binary readout for particle tracking". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 899 (2018), pp. 10–15. ISSN: 0168-9002. DOI: https://doi.org/10.1016/j.nima.2018.04.053. URL: http://www.sciencedirect.com/science/article/pii/S0168900218305709.

[104] Zhuqing Zhang and C.P. Wong. "Flip-Chip Underfill: Materials, Process and Reliability". In: Jan. 2009, pp. 307–337. ISBN: 978-0-387-78218-8. DOI: 10.1007/978-0-387-78219-5_9.

This Ph.D. thesis has been typeset by means of the TEX-system facilities. The typesetting engine was LuaLATEX. The document class was `toptesi`, by Claudio Beccari, with option `tipotesi=scudo`. This class is available in every up-to-date and complete TEX-system installation.