



Readout architectures for Pixel detectors

Roland Horisberger

Paul Scherrer Institut, CH-5232 Villigen, Switzerland

Abstract

The use of pixel detectors in high rate collider experiments requires the recording and readout of large amounts of data. The architectural choices for managing these large data flows are reviewed and illustrated on specific pixel projects. © 2001 Elsevier Science B.V. All rights reserved.

Keywords: Pixel detectors; Vertex detectors

1. Introduction

The tracking detectors of particle physics experiments at the Large Hadron Collider (LHC) will be confronted with an unprecedented increase in track rate and event complexity. At design luminosity of $10^{34} \text{ cm}^{-2} \text{ s}^{-1}$ the expected track rate is in the order of 3×10^{10} tracks per second, which is approximately a factor of 10^6 larger than the rate for a typical LEP experiment. For a pixel vertex detectors with a typical distance of 5–10 cm from the interaction region, this results in several Terabits per second of data, that need to be recorded, buffered and eventually readout. The readout architecture of a pixel chip has to organise and manage this enormous data flow with minimal deadtime and data losses. Starting with the sensor signal the following signal processing steps have to be performed:

- (i) Amplification.
- (ii) Hit decision.

- (iii) Store hits.
- (iv) Select hits of triggered events for further readout.
- (v) Find and readout trigger selected hits.

For a high rate pixel detector the dominant part of the circuit effort goes into the steps (iii), (iv) and (v). Table 1 shows size and the number of transistors of the Pixel Unit Cells (PUC) for various experiments with different readout architectures.

The use of CCD devices [1] as particle tracking detectors represents a very special type of pixel detector that is not mentioned in Table 1. These detectors do not perform a hit discrimination at the pixel level, but read out all pixel data to a off-detector electronics where all the data processing steps are performed. The pixel size can be kept rather small ($22 \mu\text{m} \times 22 \mu\text{m}$) since it contains no signal processing circuits that reduce the data volume. So far this limits its use for low rate collider experiments [2] only. However, there are plans for future projects [3] to improve the rate capability by placing a signal processing CMOS

E-mail address: roland.horisberger@psi.ch
(R. Horisberger).

Table 1

Size, area and approximate number of transistors of the Pixel Unit Cells (PUC) for various experiments

Experiment	$z \times r\phi$ (μm)	Area (μm^2)	Trans./PUC
ATLAS	400×50	20,000	250
ALICE	425×50	21,250	1500
BTeV/Fpix	400×50	20,000	550
CMS	150×150	22,500	125
DELPHI	330×330	108,000	40
LHC1/ Ω 3	500×50	25,000	400

chip to each CCD in a column-parallel architecture.

2. Architectures

The choice of a certain readout architecture in a pixel chip is strongly influenced by the track hit rate, the trigger latency and the trigger selectivity, which in turn defines the readout rate. If many of the data processing functions (i)–(v) are performed locally in the pixel, then the data volume to be read out will be strongly reduced. In case of the DELPHI pixel detector [4] the track rates of LEP are still sufficiently low, such that only the steps (i), (ii) and (iii) need to be done inside the PUC. Since the trigger latency is smaller than the LEP bunch crossing frequency a triggered readout is still possible. This changes, however, if the time between bunch crossings is smaller than the trigger latency. At this point the readout data rate from the PUC becomes equal to the track hit rate. A pixel chip that reads out all of the impinging pixel hits is often referred to as “*data push architecture*”, which allows to participate in the first level trigger decision. It enables to form a secondary vertex trigger in order to select events containing heavy flavour particles or τ leptons.

The FPIX pixel chip [5] for the BTeV experiment is based on this architecture and is planned to contribute to the first level trigger. This is possible since the particle fluences at the TEVATRON are smaller than at the LHC and since the fixed target geometry of the experiment allows to mount many optical readout fibres.

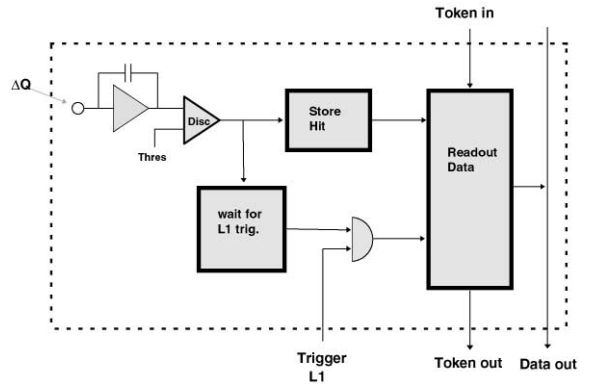


Fig. 1. Generic pixel unit cell performing all signal processing steps (i)–(v).

If one increases the track rates towards the LHC levels of $20\text{--}40\text{ MHz/cm}^2$ and considers the geometric constraints in a hermetic general purpose experiment, one realises, that further data reduction is required at the PUC or pixel chip level. This means that electronic circuits need to be implemented that perform the necessary signal processing steps described earlier. Fig. 1 shows the principle circuit realisation of all data reduction steps (i)–(v) inside a generic PUC. In this type of architecture only trigger verified data leaves the pixel unit cell. This architecture has been the basis for a series of pioneering Ω pixel chips [6].

Typical for these chips is the use of an analog timer delay that allows to perform a coincidence with the trigger signal and thus identifies the pixel hits to be read out (step (iv)). In later versions of this chip a trim mechanism [7] was implemented in order to obtain a small pixel-to-pixel dispersion of the analog delay. The “*timer architecture*” has the advantage of a minimal bus traffic along the pixel column, since only trigger verified data are leaving the pixel for readout. This allows potentially a pixel system with a very low hit threshold.

The mechanism to identify a specific bunch crossing number is normally referred to as timestamp mechanism and can be realised in several ways. In case of an analog delay the delay time is susceptible to radiation-induced changes of the transistor parameters. This could be prevented by use of a digital delay in form of a digital counter. It

requires, however, the distribution of a bunch crossing clock over the whole chip to every pixel, which implies a sizeable power dissipation. For LHC conditions this can easily result in $\approx 10 \mu\text{W}$ per pixel even without counting the power dissipation of the counter in the pixel itself.¹

A crucial step in the evolution of readout architectures was to move the timestamp mechanism from the pixel to the column periphery. This allows to reduce the total number of the timestamp circuits on a pixel chip by an factor of 10–20 without loss of functionality. The power dissipation and the total number of transistors per chip is such considerably reduced. Once the timestamp circuits are placed outside the PUC at the column periphery an association mechanism is required that provides a clear link between a hit PUC and its partner timestamp circuit in the periphery.

The first architecture that followed this route that was originally developed by the LBL group for the SSC and later adapted for the ATLAS pixel detector at LHC [8]. The crucial link mechanism consists of a pointer number (3-bit) that tells the PUC to which timestamp circuit in the periphery it belongs to. The PUC latches this number into a 3-bit memory cell and uses its contents to react for a call to readout, in case the associated timestamp in the periphery requires this. Since the PUC is linked to the timestamp circuit via a pointer, this architecture could be referred to as a “*timestamp pointer architecture*”. Since all pointers and data signals are transmitted along the pixel column bus one has considerable bus activities. These irregular signals can potentially cross couple to the pixel amplifiers and could eventually require a higher comparator threshold.

A different approach to associate a pixel hit with its timestamp mechanism in the chip periphery has been pursued by the ATLAS pixel group as the “*Front End-A*” readout chip [9]. This architecture is based on the idea to place the address (7-bit) of a PUC hit onto a 7-bit wide shiftregister that runs

with the LHC bunch crossing frequency of 40 MHz. With each clock cycle the hit address is transported from pixel to pixel towards the column periphery. When the hit address reaches the periphery, it is attached to one of several available timestamp counters, which then will count the remaining clock cycles for the correct trigger latency. This architecture could be described as “*conveyor belt architecture*” since pixel hits are transported uniformly down the column towards the chip periphery. Although the continuous clocking of the shiftregisters could eventually generate crosstalk to the pixel amplifiers, it is a crosstalk that is very regular and therefore can be adapted by the comparator threshold.

An important consideration in the conception of a deadtime free pixel architecture is the problem of the L1 occupancy. This quantity is the probability for a pixel cell to be hit during its L1 trigger latency time. It is the product of the track fluence rate, the area of the pixel unit cell and the L1 trigger delay time. For a pixel vertex detector in pp collision at LHC the L1 occupancy can reach 5% or more, which normally leads to data losses at the same level. Without additional circuitry this problem can be tackled by either making the pixel area smaller, or by a reduction of the L1 time period for which a pixel unit cell is blocked by a hit. Another possibility is to provide a multi-hit capability to each PUC which normally increases the pixel complexity and therefore its size. Fig. 2 shows the data loss of a PUC as a function of the

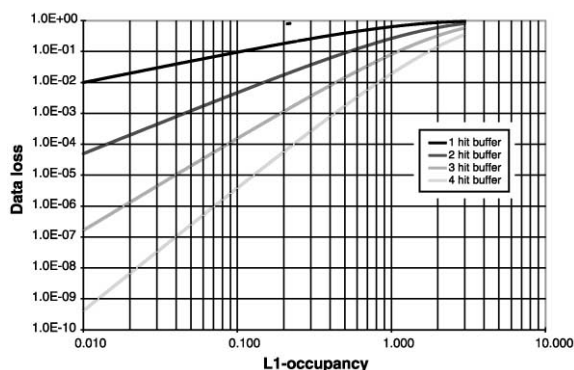


Fig. 2. Pixel data loss versus L1 occupancy for different scenarios of multi-hit buffers in the pixel unit cell.

¹ Estimated power dissipation for a digital supply voltage of 5 V. For deep submicron technologies with smaller supply voltages, the corresponding value will be of course smaller, however, its relative contribution to the overall power dissipation of the chip remains unchanged.

L1 occupancy and for different lengths of its multi-hit buffer. It can be noted that even for the pessimistic case of doubling the pixel area with a second hit buffer, which increases the pixel rate, one still gains in the data loss rate by more than an order of magnitude.

If a second hit capability is implemented into the pixel unit cell it will add more transistors and therefore increase the pixel area. This has negative effects like reducing the position resolution. Since the extra circuitry for a second pixel hit is only activated with the probability of the L1 occupancy, it is rather poorly utilised. This problem can be avoided by introducing a pool of second hit buffers at the column periphery that can be used much more efficiently.

The use of additional hit buffers in the column periphery requires a fast transfer mechanism that copies pixel hits as quick as possible to the periphery buffers. This in turn frees the PUC for a new hit and at the same time allows the L1 trigger verification of the transferred hits in the column periphery. Although the PUC still only stores one hit at a time in this architecture, it reduces the L1 latency problem by a large factor, since the pixel is not blocked for the complete L1 trigger delay time but only for the short time period that is required to copy the hits to the periphery buffers. This concept is the basis of the “*column drain architecture*” in the CMS pixel chip [10] and in a similar way also of the “*Front End-B*” architecture [11] chosen for the ATLAS pixel project. Although both architectures share the same concept of transferring pixel hits to the periphery for trigger verification they differ, however, in details of the utilised mechanisms.

In the CMS pixel chip the transfer of pixel hits to the periphery buffers is done by a column drain mechanism, that is capable of recording a new timestamp while still copying the pixel hits of the previous timestamp to their periphery buffers. This double hit capability at the column level reduces the data loss for this data transfer mechanism to the per mille level. Since, on the average, the column drain mechanism is active only for a fraction of the time ($< 10\%$) it does not contribute significantly to the overall power consumption of the readout chip. In addition this power dissipa-

tion can be kept small by concentrating the timestamp mechanism in the column periphery and therefore reducing the load capacitances of the timestamp bus. The association of the different timestamp buffers and their corresponding data buffers, containing the pixel hits, is done by a pointer segmentation of the two buffer systems that are operating concurrently. Details on the architecture and the statistical occupancy of the buffers are described elsewhere [12].

In the final ATLAS pixel chip, the timestamp number (7-bit) is distributed over the complete chip to all pixels. In case of a pixel hit this number is sent along with the pixel address to the periphery, where the hit gets buffered for L1 trigger verification. The availability of the timestamp at the pixel level is actually also used for a simple pulse height digitisation. This is achieved by recording the timestamp number for the trailing edge of the “*time over threshold*” signal from the comparator. The direct tagging of the pixel hits with the timestamp numbers (leading and trailing edge) has the advantage that multiple hits during the fast column scan should not result in any data loss. More details about the buffer depth and performance of this architecture can be found in Ref. [11].

It has been mentioned earlier, that the introduction of a multi-hit buffer in the PUC reduces the L1 latency problem by a large factor. The ALICE1 chip [13] has chosen a 2-hit buffer solution together with a L1 trigger verification at the pixel level, which implies a distribution of the trigger signal to all pixels. This architecture is performing all signal processing steps inside the PUC in a similar way to the generic pixel shown in Fig. 1. An interesting feature in this chip is a new digital trigger verification mechanism that replaces the analog delays used in the earlier series of Ω pixel chips [6]. This mechanism is based on a chip wide distributed periodic time signature (8-bit), that in case of a pixel hit is stored in the PUC, and later checked for unique recurrence. The period of this time signature is exactly the L1 latency, which allows the pixel to find its coincidence with the distributed trigger signal. In addition this PUC has even implemented a 4 deep hit buffering, that allows a readout depending on the L2 trigger

decision. The implementation of all these functionalities into the PUC results of course in an increased number of transistors per pixel as can be seen in Table 1. Due to the use of an deep submicron technology, however, it has been possible to stay with an pixel area that is quite similar to the other pixel projects. The high integration density of this CMOS technology has even permitted to include an additional feature, that allows to pool together 8 pixels to 1 super-pixel. This is planned to be used for a RICH readout application which requires a deeper event buffering [14] before readout.

As mentioned before, there is a common characteristic of the ATLAS and CMS pixel architectures, that they are transferring the pixel hits as quickly as possible to the periphery, where they are stored and wait for the first level trigger verification. If a pixel system must contribute for instance to a first level vertex trigger, this data buffering has to be omitted and the data stream must be read out directly. The FPIX pixel chips [15] for the BTeV experiment are designed for this purpose, although they are compatible with a triggered operation mode in future as well. In order to organise a deadtime free transfer of the pixel hits, a tagging system with 4 timestamps per column is used. The timestamp management at the column periphery is able to activate, for each timestamp, two command lines that control the activity status (idle, reset, output, write) of the pixel unit cells in this column.

3. Summary and conclusions

The readout architectures of pixel systems have the task to organise and handle the enormous amount of data that are typically generated in future high rate experiments. It is clear, that the choice of the architecture should be adapted to the specific experimental conditions (rates, orbit gaps, etc.) and its requested tasks, like the participation

in the first level trigger. The integration density of the used CMOS technology has a strong influence on the architectural choices. The availability of several metal layers allows designers in the future to develop connection intensive architectures, that for instance could perform a cluster analysis and therefore reduce the amount of data to be read out even further.

References

- [1] C.J.S. Damerell, *Rev. Scientific Instrum.* 69 (1998) 1549.
- [2] K. Abe et al., *Nucl. Instr. and Meth. A* 400 (1997) 287.
- [3] A.R. Gillman, *Proc Vertex 2000 Workshop*, *Nucl. Instr. and Meth.*, to be published.
- [4] P. Delpierre, J.J. Jaeger, *Nucl. Instr. and Meth. A* 305 (1991) 627.
- [5] D. Christain et al., *Nucl. Instr. and Meth. A* 435 (1999) 144.
- [6] E.H.M. Heijne et al., CERN RD-19 Collaboration, First Operation of a 72k Element Hybrid Silicon Micropattern Pixel Detector Array, CERN/ECP 94-1, March 1994.
- [7] E. Heijne et al., *Nucl. Instr. and Meth. A* 383 (1996) 55.
- [8] M. Wright, J. Millaud, D. Nygren, LBL-32912, October 1992.
- [9] The FE-A prototyping program, ATLAS Pixel Detector, ATLAS TDR 11, CERN/LHCC/98-13, 31 May 1998, pp. 107–126.
- [10] CMS Tracker Project, Technical Design Report, CMS TDR 5, CERN/LHCC 98-6, 15 April 1998.
- [11] The FE-B prototyping program, ATLAS Pixel Detector, ATLAS TDR 11, CERN/LHCC/98-13, 31 May 1998, pp. 126–142.
- [12] D. Kotlinski, The CMS Pixel Detector, Fifth Conference on Position Sensitive Detectors, London, Sept. 1999, *Nucl. Instr. and Meth.*, to appear.
- [13] K. Wyllie et al., A pixel readout chip for tracking at ALICE and particle identification at LHCb, Fifth Workshop on Electronics for LHC experiments CERN 99-09, CERN/LHCC/99-33, 29 October 1999, pp. 93–97.
- [14] LHCb Technical Proposal, CERN/LHCC 98-4, February 1998.
- [15] A. Mekkaoui et al., FPIX1: an Advanced Pixel Readout Chip, Fifth Workshop on Electronics for LHC experiments CERN 99-09, CERN/LHCC/99-33, 29 October 1999, pp. 98–102.