**Università di Pisa**

FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E NATURALI
Corso di Laurea Magistrale in Fisica

TESI DI LAUREA MAGISTRALE

# Development of a real-time tracking device for the LHCb Upgrade 1b

Candidato:
**Federico Lazzari**

Relatori:
**Prof. Giovanni Punzi**
**Dott. Riccardo Cenci**

**Anno Accademico 2016-2017**

# Contents

# Chapter 1

# Introduction

Past experiments on $b$- and $c$-physics have provided important contributions to the understanding of CP violation and to the determination of CKM matrix parameters. Current and future experiments, such as LHCb at the LHC collider and Belle II at SuperKEKB machine, have the potential to significantly improve our knowledge on CKM parameters thanks to an huge production of $b$- and $c$-hadrons (see Chapter 2). However, because of a small signal-to-background ratio for typical interesting processes and the limited bandwidth available for storing data, the adoption of powerful and very selective trigger systems is needed, particularly at hadron colliders. The most important discriminant for decay of $b$- and $c$-hadrons is their relatively long lifetime, that requires excellent tracking systems to discriminate interesting events from the huge background.

The LHCb experiment is going to increase its luminosity by a factor of 5 starting in the 2020 (see Chapter 3). In this new regime, it will adopt a full software trigger running on a large PC farm, to reconstruct all tracks produced in every LHC collision, occurring at a rate of 40 MHz. This is a large step forward from the current rate of tracked events (1 MHz)- in addition to the luminosity increase. On account of the significant CPU time required, it is not planned to perform the reconstruction of particles generated outside of the vertex detector ("downstream tracks"). While this covers most of the decays of $b$- and $c$-particles, not having access to this information limits efficiency for decay modes containing neutral hadrons and long lived particles ($K^0_\mathrm{S}$ and $\Lambda$). These include many interesting decays like $D^0 \to K^0_\mathrm{S} K^0_\mathrm{S}$ or $\Lambda^0_b \to 3\Lambda$. For this reason, while still maintaining a trigger strategy firmly based in software, the LHCb collaboration has recently put forward an Expression of Interest for a further upgrade of the detector [1] to include some specialized hardware devices that could operate a real-time reconstruction of some parts of the tracking, and to relieve of the computational burden from the CPU farm, thus allowing extending the reconstruction to the downstream part of the tracker and handling even higher beam intensities.

In this thesis I have to perform a study of the feasibility of performing a real-time reconstruction of downstream tracks at the earliest trigger level (the Event Builder), using a FPGA-based system organized according to the innovative "Artificial Retina" architecture.

The “Artificial Retina” approach (see Chapter 4) draws inspiration from the biological example of the organization of visual areas in the brain of mammals [2]. The parameter space of tracks is divided in cells, that are implemented as active computational elements, evaluating a numerical “excitation level” in a totally parallel way. The local maxima in the “excitation distribution” are also calculated by a fully parallel clustering process, that interpolates the response of adjacent cells to obtain good resolution performances while keeping the number of cells within manageable limits. A fully custom intelligent switching network provides large-volume, low-latency data distribution to the cells, exploiting to its fullest the wide bandwidth now made commercially available by the latest technological developments in optical-link based programmable digital devices (FPGAs).

This approach is new and currently still in an R&D stage. With my thesis work I have contributed to its development within the INFN technological project ‘RETINA’. When I started my thesis, only a low-speed demonstrator prototype was in existence, based on slow FPGA from previous generation (65 nm process), whose purpose was just to test that the overall logic of the device was correct and could operate as desired, with no requirements on speed (see Chapter 5).

The first goal of my thesis work was to investigate whether the ‘RETINA’ system really had the potential to process tracks with a speed sufficient for implementation of a realistic tracking system capable of operating at Level-1 of the LHC - this had never been attained before without the help of some form of time-multiplexing to reduce the rate. To this purpose, I have produced a new implementation of the system on current, much faster and bigger FPGAs (Stratix-V), connected by fast lines the same board, in order to simulate the real conditions of the final device. In order to take full advantage of the performance of this new hardware prototype, I had to re-design several parts of the previous existing firmware, both the switching network and the cell processors, for optimal performance and speed, using low-level hardware description languages (VHDL). This also included re-design the interface completely for use on a completely different board, a special custom-order board aimed at the development and test of new fast ASICs projects (see Chapter 6).

Testing the system with realistic simulated events in a “general-purpose” 6-layer tracking detector, I debugged and measured the throughput of the new system as a function of the occupancy of the tracker. In this way, I managed to produce in the lab a hardware prototype capable of processing events at a rate even higher that LHC event rate.

Another crucial parameter of the system is the latency. In order to work in the intended way, as a transparent device incorporated within the Event Builder, the latency of the system has to be limited to very few µs. This is another challenging requirement, as all currently existing designs require tens of µs. After performing an optimization of the internal pipeline of the device, I managed to reduce the latency to less than 1 µs.

Encouraged by these results, I proceeded to a higher-level study of the efficiency, ghost rate, and event rate of this system when applied to a generic bare-bone detector, and compared these with the performance of a software algorithm (see Chapter 7). The Retina showed to have similar efficiency and event rate higher of 1-2 order of magnitude with 10-20 tracks, but also an higher ghost rate. For reducing the ghost rate I studied also possible optimization of the system, introducing requirements on the track $\chi^2$ computed with a linearized fit. Modern FPGA have a large number of digital signal processors (DSP) capable of floating point operation suitable for the task, and my work demonstrate the need of a DSP stage to the final system.

Finally, I proceeded to an even higher-level study, tackling the real configuration of the LHCb downstream tracking detector (see Chapter 8). Given the complexity of the system, I aimed at reproducing the 2D section of the reconstruction in the most forward Scintillating Fiber detector. I also compared it with the performance of the traditional CPU-based reconstruction software, to have a first check that a reconstruction of sufficient quality is feasible, using an amount of hardware contained within reasonable practical limits. I performed both a preliminary study based on a home-made event generator, that did not include multiple scattering and the fringe magnetic field, and a more complete study based on the actual official simulation of the LHCb detector, interfaced through a custom piece of software to my own code. Both have been performed with realistic track occupancy as expected in the upcoming physics run of the LHC. In conclusion (see Chapter 9), while longer and more extensive studies are needed including all layers of the full 3D detector, my work demonstrates that a special-purpose processor based on the "Artificial Retina" approach can be built at a reasonable cost using FPGA devices. I implemented and tested an advanced prototype, and made a series of studies on the performances of the system applied to a real case, the tracking of downstream track at LHCb Upgrade. This is a significant step towards real-time tracking at HL-LHC, a methodology that will also open the possibility to trigger purely on long-lived neutrals, increasing significantly the acceptance for some channels and expanding our Physics reach.

# Chapter 2

# Motivations

The *CP* violation, i.e. the non-invariance of the weak interactions with respect to a combined charge-conjugation ($C$) and parity ($P$) transformation, dates back to year 1964, when this phenomenon was discovered through the observation of $K_{\mathrm{L}} \to \pi^+\pi^-$ decays [3]. One of the key features of our Universe is the cosmological baryon asymmetry of $\mathcal{O}(10^{-10})$. As was pointed out by Sakharov [4], the necessary conditions for the generation of such an asymmetry include also the requirement that elementary interactions violate *CP*. Model calculations of the baryon asymmetry indicate, however, that the *CP* violation present in the Standard Model (SM), *i.e.* the theory which currently describes better the nature at the smallest scale of fundamental interaction, seems to be too small to generate the observed asymmetry.

The understanding of *CP* violation, and therefore of flavour physics, is particularly interesting since New Physics (NP), *i.e.* physics lying beyond the SM, typically leads to new sources of flavour and *CP* violation. Following this direction, an important field of investigation is represented by flavor physics at accelerating machines, and in particular by the beauty and charm sectors. Over years, numerous experiments were dedicated to $b$ and $c$-hadron study, following different approaches. Two deeply different but complementary environments are represented by $B$-factories and by high energy hadron colliders. Both study *CP* invariance violation in bottom and charmed hadron physics by performing high precision measurements of *CP* violation, to increasingly constrain the theoretical uncertainties on SM and to search for NP.

## 2.1 *CP* violation

In 1964, the observation of neutral long-lived K mesons decay in both two and three pions states [3] showed that not all interactions in Nature are symmetric under *CP* transformation. The measurement of a $\mathcal{O}(10^{-3})$ branching fraction for the $K^0_L \to \pi^+\pi^-$ was the first evidence for *CP* violation in Nature. In particular, this is a manifestation of indirect *CP* violation, caused by the fact that the neutral kaon mass eigenstates, $K^0_L$ and $K^0_S$, are not eigenstates of the *CP* operator. This causes the small *CP*-even component of the $K^0_L$ state decay into the $\pi^+\pi^-$ final state.

After 30 years of series of experiments, in 1999 was established the first direct

*CP* violation evidence, still in neutral kaon states, by NA48 [5] and KTeV [6] collaborations. It directly concerns the decay amplitudes of two *CP* conjugate states, and confirms the theory for which the *CP* violation is an universal property of the weak interaction, proposed by Wolfenstein [7] in 1964 just after its first observation. Huge experimental efforts have been dedicated to extend the *CP* violation study on other systems than kaons, BaBar [8] and Belle [9] experiments observed for the first time the $B^0 \to J/\psi K$ decay-rate asymmetry, caused by the interference of decay amplitudes occurred with $B^0 - \overline{B}^0$ flavor mixing and the amplitude of the direct decays.

## 2.2 The CKM matrix

Since the first experimental evidence of *CP* violation in Nature, considerable efforts to describe it into a coherent theoretical environment have been performed. They significantly have contributed to build the SM, describing the electroweak interactions. In this framework, *CP*-violating effects originate from the charged-current interactions of quarks, having structure:

$$D \to UW^-,$$

where $D$ denotes down-quark flavors $(d, s, b)$, $U$ denotes up-type quark flavors $(u, c, t)$ and $W^-$ is the usual gauge boson. The electroweak states $(d', s', b')$ respectively of $d, s, b$ quarks are connected with their mass eigenstates $(d, s, b)$ through the following unitary transformation:

$$\begin{pmatrix} d' \\ s' \\ b' \end{pmatrix} = V_{CKM} \cdot \begin{pmatrix} d \\ s \\ b \end{pmatrix},$$

where $V_{CKM}$ is the unitary Cabibbo-Kobayashi-Maskawa (CKM) matrix [10, 11], which represent the generic "coupling strengths" $V_{UD}$ of the charged-current processes:

$$V_{CKM} = \begin{pmatrix} V_{ud} & V_{us} & V_{ub} \\ V_{cd} & V_{cs} & V_{cb} \\ V_{td} & V_{ts} & V_{tb} \end{pmatrix}.$$

Expressing the non-leptonic charged-current interaction Lagrangian in terms of the mass eigenstates $(d, s, b)$, we obtain:

$$\mathcal{L}^{CC}_{int} = -\frac{g_2}{\sqrt{2}} \begin{pmatrix} \bar{u}_L, & \bar{c}_L, & \bar{t}_L \end{pmatrix} \gamma^\mu V_{CKM} \begin{pmatrix} d_L \\ s_L \\ b_L \end{pmatrix} W^\dagger_\mu + h.c., \tag{2.1}$$

where $g_2$ is a coupling constant, and the $W^{(\dagger)}_\mu$ field corresponds to the charged $W$ bosons. Looking at the interaction vertices following from equation 2.1, we observe that the $V_{CKM}$ elements describe the generic strengths of the associated charged-current processes, as we have noted above.

In a vertex $D \to UW^-$, $CP$ transformation involves the replacement $V_{UD} \to V_{UD}^*$: $CP$ violation could therefore be accommodated in the SM through complex phases in the CKM matrix. As pointed by Kobayashi and Maskawa in 1973 [11], the parametrization of $V_{CKM}$ for three generations of quarks involves three Euler-type angles and one complex phase. However, further conditions have to be satisfied to observe $CP$-violating effects [12–14], related to quark mass hierarchy.

Using the related experimental informations together with the CKM unitarity condition, and assuming only three quark generations, we obtain the following values for the CKM matrix elements [15]:

$$|V_{CKM}| = \begin{pmatrix} 0.974334^{+0.000064}_{-0.000068} & 0.22508^{+0.00030}_{-0.00028} & 0.003715^{+0.000060}_{-0.000060} \\ 0.22494^{+0.00029}_{-0.00028} & 0.973471^{+0.000067}_{-0.000067} & 0.04181^{+0.00028}_{-0.00060} \\ 0.008575^{+0.000076}_{-0.000098} & 0.04108^{+0.00030}_{-0.00057} & 0.999119^{+0.000024}_{-0.000012} \end{pmatrix}.$$

Transitions within the same generation are governed by the CKM matrix elements of $\mathcal{O}(1)$, those between the first and the second generation are suppressed by CKM factors of $\mathcal{O}(10^{-1})$, those between the second and the third generation are suppressed by $\mathcal{O}(10^{-2})$, and transitions between the first and the third generation are suppressed by CKM factors of $\mathcal{O}(10^{-3})$.

To bring out the CKM matrix hierarchical structure, it is convenient to represent it in the so called “Wolfenstein parametrization” [16] as a function of a set of parameters $\lambda, A, \rho, \eta$:

$$V_{CKM} = \begin{pmatrix} 1 - \frac{1}{2}\lambda^2 & \lambda & A\lambda^3(\rho - i\eta) \\ -\lambda & 1 - \frac{1}{2}\lambda^2 & A\lambda^2 \\ A\lambda^3(\rho - i\eta) & -A\lambda^2 & 1 \end{pmatrix} + \mathcal{O}(\lambda^4),$$

The unitarity of the CKM matrix, which is described by the relationship:

$$V_{CKM}^\dagger \cdot V_{CKM} = V_{CKM} \cdot V_{CKM}^\dagger = I,$$

results into a set of 12 equations, consisting of 6 normalization and 6 orthogonality relations. The latter can be represented as 6 triangles in the complex plane, all having same area. However, only two of those are non-squashed triangles, having angles of same order of magnitude. They are defined by the relations:

$$\begin{aligned} \underbrace{V_{ud}V_{ub}^*}_{(\rho+i\eta)A\lambda^3} + \underbrace{V_{cd}V_{cb}^*}_{-A\lambda^3} + \underbrace{V_{td}V_{tb}^*}_{(1-\rho-i\eta)A\lambda^3} &= 0, \\ \underbrace{V_{ud}^*V_{td}}_{(1-\rho-i\eta)A\lambda^3} + \underbrace{V_{us}^*V_{ts}}_{-A\lambda^3} + \underbrace{V_{ub}^*V_{tb}}_{(\rho+i\eta)A\lambda^3} &= 0. \end{aligned}$$

At $\lambda^3$ level, the two orthogonality relations agree with each other, yelding:

$$[(\rho + i\eta) + (-1) + (1 - \rho - i\eta)]A\lambda^3 = 0. \tag{2.2}$$

Therefore, those two orthogonality relations describe the same triangle in the $(\rho, \eta)$ plane shown in Figure 2.1, which is usually referred to as the unitarity triangle of the CKM matrix. Angles of unitarity triangle are usually called $\alpha, \beta, \gamma$.

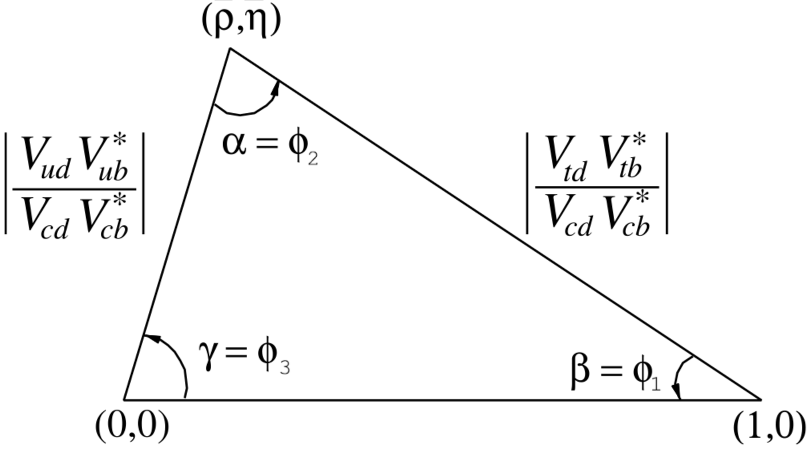
$(\bar{\rho},\bar{\eta})$
$\left|\frac{V_{ud}V_{ub}^*}{V_{cd}V_{cb}^*}\right|$
$\alpha = \phi_2$
$\left|\frac{V_{td}V_{tb}^*}{V_{cd}V_{cb}^*}\right|$
$\gamma = \phi_3$
$\beta = \phi_1$
(0,0)
(1,0)

Figure 2.1: Rescaled unitarity angle.

The parametrization of the CKM matrix is not unique; in particular, we can replace the $\rho$, $\eta$ parameters with a new set $(\bar{\rho}, \bar{\eta})$ in a such way to include in the equation 2.2 also terms $\mathcal{O}(\lambda^5)$, obtaining [17]:

$$[(\bar{\rho} + i\bar{\eta}) + (-1) + (1 - \bar{\rho} - i\bar{\eta})]A\lambda^3 + \mathcal{O}(\lambda^7) = 0,$$

where:

$$\bar{\rho} = \rho(1 - \frac{\lambda^2}{2}), \quad \bar{\eta} = \eta(1 - \frac{\lambda^2}{2}).$$

The CKM matrix has a great predictive potential on *CP* violating processes, and large experimental efforts have been performed to measure its parameters. Figure 2.2 shows the global fit of CKM parameters [15], in $(\bar{\rho}, \bar{\eta})$ plane, resulted by combining performed measurements.

The study of several, different physics processes have provided measurements of *CP* asymmetry in Nature, which are all contained within the uncertainties of CKM parameters. Nevertheless, to the present day they are still not measured with great precision, such as for the $\gamma$ parameter [15]:

$$\gamma = (72.1^{+5.4}_{-5.8})^\circ.$$

Much more, precise measurements of CKM parameters are required to seriously challenge the SM explanation of *CP* violation. This investigation represents a fundamental probe to validate at deeper scales of precision the SM predictions on observable physics processes, and to search for NP evidences.

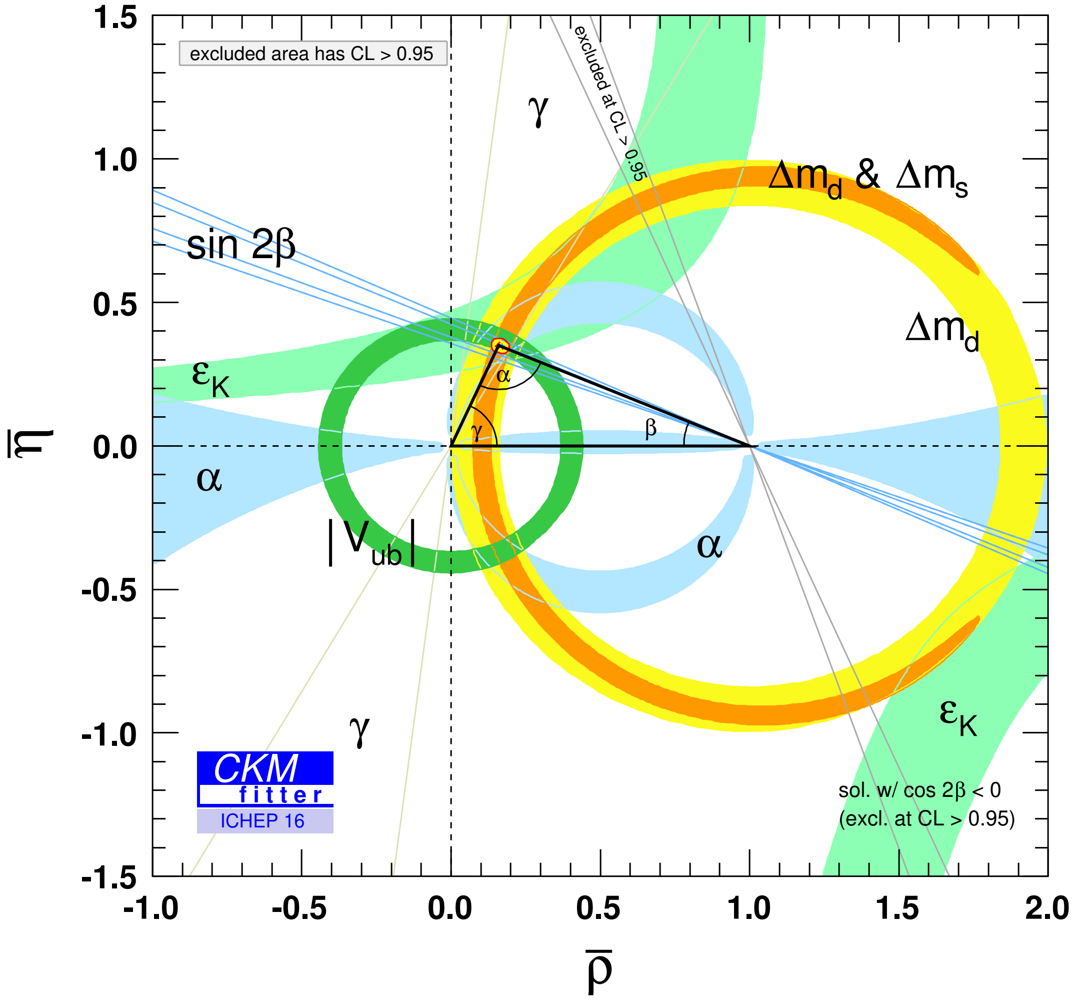

Figure 2.2: Global CKM fit in the $(\bar{\rho}, \bar{\eta})$ plane.

## 2.3 *CP* violation and heavy flavor physics

While *CP* violation might have a role in leptonic interactions as well, the most experimentally accessible field is that of quark interaction. In particular, due to its connection with the 3-generation structure of the matrix, the heavier quarks that are still able for form bound states (bottom and charm) play a central role. Luckily, the large mass of these quarks also helps in allowing some simplifying approximations in performing theoretical calculations of the relevant hadron dynamics. Past experiments on *b*- and *c*-physics have provided important contributions to the *CP* violation understanding, and to the determination of CKM matrix parameters.

At the same time, current and future experiments, such as LHCb at the LHC collider and Belle II at SuperKEKB machine, will be able to largely improve our knowledge on CKM parameters thanks to an huge production of $b$- and $c$-hadrons, resulting in a collection of very large samples of interesting physics processes.

The $b$-hadrons represent particularly interesting systems to study $CP$ violation. First, they contain the $b$-quark, belonging to the third quark generation and therefore characterized by the possibility to decay to quarks of both first and second generations of the first or second generation. This allows reaching larger $CP$ violation effects than in kaon systems. Moreover, the larger mass of the $b$-quark compared to the $s$-quark one makes kinematically available many decay modes, offering multiple experimental possibilities to study $CP$-violating observables. Even if having a smaller mass, charmed hadrons equally represent very interesting systems, and they are the only system in which up-type quark interactions can be studied, which might in principle have a separate dynamics from down-type quarks. For these reasons, flavor physics represents a particularly promising and interesting sector to deeply study $CP$ violation. However, the presence of multiple available channels results in small branching fractions of individual processes, and high statistic samples are therefore required.

## 2.4 Experimental considerations on flavor physics

Charmed hadron physics begun in lepton annihilation experiment in 1974, with the discovery of the $J/\psi$ resonance at SLAC experiment [18] and Brookhaven Laboratory [19]. After only three years, the $b$-hadrons physics dates its beginning in proton-nucleus collisions with the discovery of the $\Upsilon$ resonance, in 1977 at Fermilab laboratory [20]. Measurements on heavy flavor states followed in UA1 experiment [21] and in CDF I from 1992 to 1996 (as example, see [22, 23]). Much more significant contributions to $b$-quark physics came from $e^+e^-$ machines operating at the $\varUpsilon(4S)$ resonance (the so named $B$-factories machines), or at the $Z$ pole and more recently in hadronic machines, when the huge available cross section for production of heavy quarks started to be systematically exploited by means of new and improved experimental techniques.

### 2.4.1 The $B$-factories

$B$-factories are $e^+e^-$ colliders with asymmetric beam energies, producing $\varUpsilon(4S)$ resonances with 0.4-0.6 Lorentz boost. The $\varUpsilon(4S)$ meson decays more than 96% of times into $B\overline{B}$ pairs (where $B$ is $B^0$ or $B^+$) [24], which thanks to the beam asymmetry decay in vertices typically displaced by 200-300 μm. Exploiting the good spatial resolution of silicon detectors, this distance allows to determine the time-interval between the two decays with sufficient precision to measure time-dipendent $CP$-violating asymmetries. Operating at an energy calibrated to the $\varUpsilon(4S)$ production, just above the open beauty threshold, avoids the presence of fragmentation products and imposes kinematic constraints resulting in a background reduction. Pile-up events, that is multiple primary interactions in a single beam crossing, are typically absent and track multiplicity is typically not greater then $\sim 5$ tracks for event. However, cross-section of $B\overline{B}$ production is limited to just $\sigma(b\bar{b}) \sim 1\,\text{nb}$.

Past experiments installed at $B$-factories, such as BaBar [25] and Belle [26], successfully demonstrated the validity of this approach giving large contributions to heavy flavor physics understanding, such as the measurement of the $\beta$ angle of the unitarity triangle [27], shown in Figure 2.3 for the channel $B^0 \to \eta' K^0$. The Belle II experiment, at SuperKEKB $B$-factory, is currently being set up and is expected to begin data collection from 2019 [28].

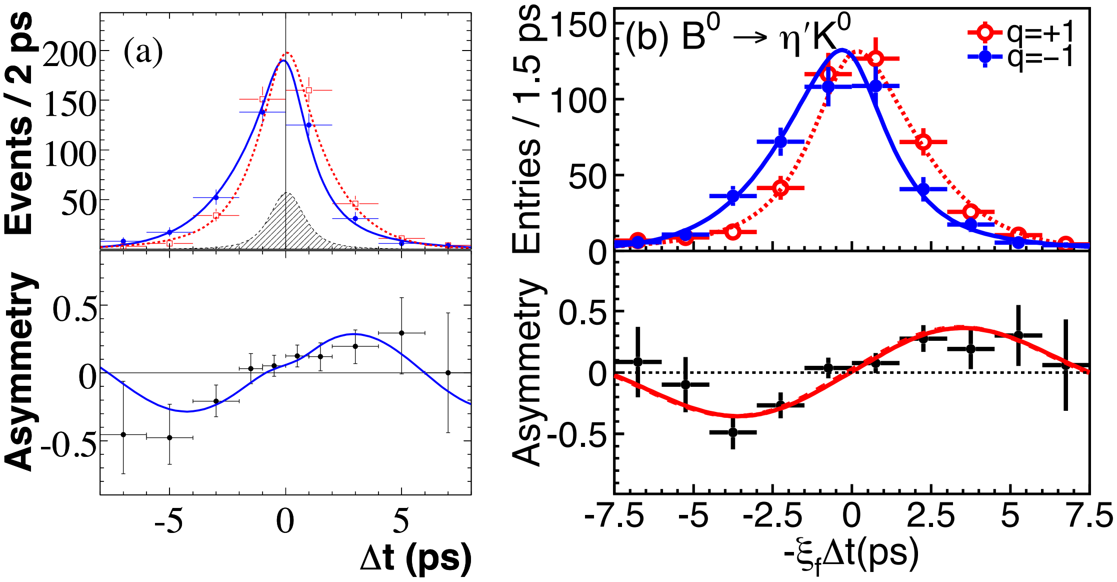

Figure 2.3: Measurement of $\Delta t$ and asymmetries distributions in the $B^0 \to \eta' K^0$ channel, performed by BaBar (a) and Belle (b) experiments. For BaBar, only $\eta' K^0_S$ mode is shown.

### 2.4.2 Flavor physics at hadron colliders

Hadron colliders have much larger cross-section for $b$- and $c$-quarks production. The dominant production process for $b$-hadrons is the non-resonant inclusive $b\bar{b}$ production, with typical values at Tevatron ($p\bar{p}$ collisions) and LHC ($pp$ collisions), integrated on the entire solid angle:

$$\sigma(p\bar{p} \to b\bar{b}X, \sqrt{s} = 1.96\,\text{TeV}) \sim 80\,\mu\text{b},$$
$$\sigma(pp \to b\bar{b}X, \sqrt{s} = 14\,\text{TeV}) \sim 500\,\mu\text{b},$$

where $\sqrt{s}$ represents the center-of-mass energy of the collision. These values must be compared with the typical $b\bar{b}$ cross-section production at B-factories, of

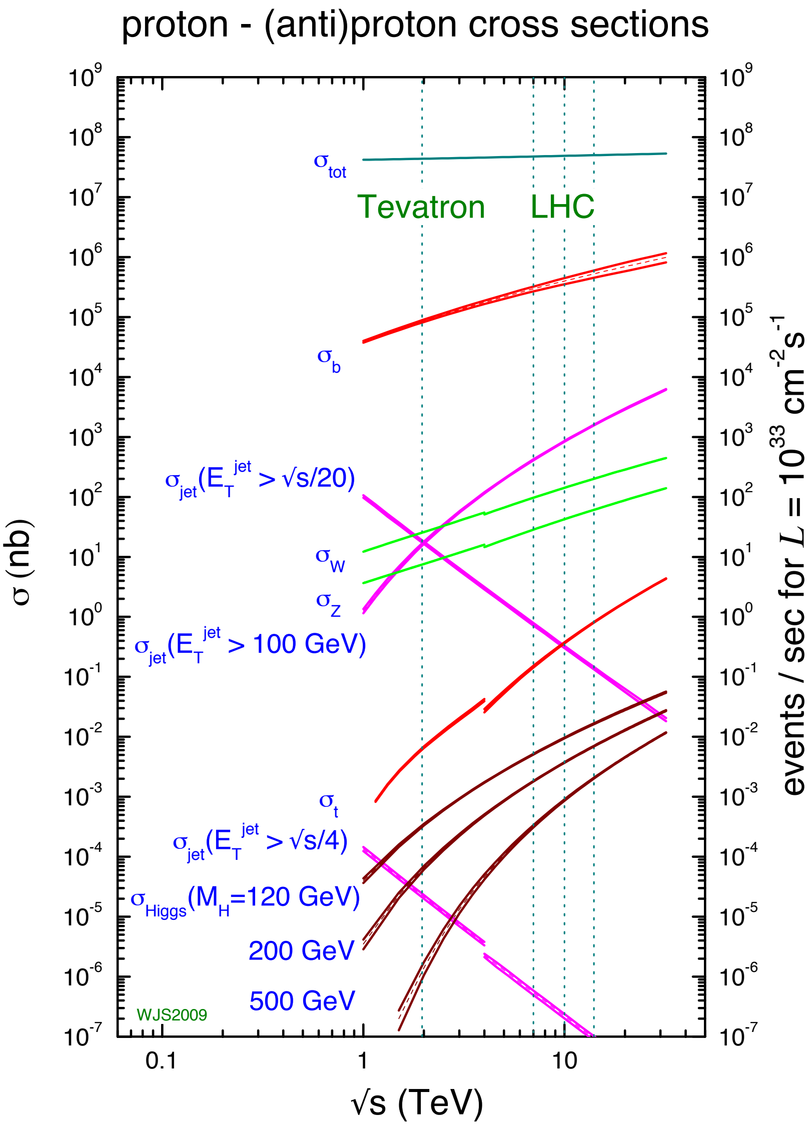

Figure 2.4: Cross-sections for processes at $pp$ and $p\bar{p}$ colliders, depending on machine center-of-mass energy $\sqrt{s}$. Discontinuities are caused by transitioning from $p\bar{p}$ to $pp$ collisions.

$\sigma(b\bar{b}) \sim 1\,\text{nb}$. Figure 2.4 reports the cross-sections trend for processes at $pp$ and $p\bar{p}$ colliders, depending on machine $\sqrt{s}$. The $\sqrt{s}$ energy available at hadron colliders allows the production of all $b$-hadrons species: $B^0$ and $B^+$ mesons, but also $B_s^0$, $B_c^+$ mesons and $b$-baryons; moreover the typical $\beta\gamma$ Lorentz boost of produced $b$-hadrons are larger compared to $B$-factories. This results in larger decay lengths, which allow probing shorter scales in heavy-flavor time-evolution. However, at hadron collisions the $b\bar{b}$ cross-section is about three order of magnitudes lower than hadron-hadron inelastic cross-section [29]:

$$\sigma(pp \text{ inelastic}, \sqrt{s} = 14\,\text{TeV}) \sim 100\,\text{mb},$$

resulting in high-suppressed signal-to-background ratio for typical interesting processes, for instance of the order $\mathcal{O}(10^{-9})$ for the $B^0 \to K\pi$ channel. Because of the limited bandwidth available for storing data, this makes it necessary tracker and trigger systems which operate in real-time, capable to discriminate interesting events from the huge light-quark background and therefore to select high-purity signal sample to store. Events in hadron colliders are also more complex than in $B$-factories, resulting in more difficult reconstruction of $b$-hadrons decays and requiring higher granularity detectors. Indeed, in most hard interactions only one constituent (valence or sea quark, or gluon) of the colliding hadron undergoes an hard-scattering against a constituent of the other colliding hadron: this is the leading interaction that may produce a $b\bar{b}$ pair. Others hadron constituents rearrange in color-neutral hadrons, which may have transverse momentum (i.e. momentum perpendicular to the beam pipe) sufficient to enter the detector acceptance, resulting in the so named underlying event. In the underlying event multiple hard-scattering interactions may occur between the partons consisting the same pair of colliding hadrons. Furthermore, $b$-hadron fragmentation process, that is the transition from a not observable single-state quark to an observable color-singlet hadron, results in a number of accompanying hadrons produced in the local region around the hadronizing quark. Fragmentation of all quarks and gluons in the event represent an important source of track multiplicity. Finally, when beams collide multiple hard interactions may occur between their hadrons, resulting in pile-up events. Each hard interaction introduces related fragmentation processes and underlying events.

Similar arguments are valid for charmed hadrons, although characterized by even higher production cross-section [30]:

$$\sigma(pp \to c\bar{c}X, \sqrt{s} = 14\,\text{TeV}) \approx 10\,\text{mb}.$$

### 2.4.3 Final considerations

$B$-factories and hadronic collider are both interesting facilities to study $CP$ invariance asymmetry in High Energy Physics (HEP) environment. The two approaches are complementary, with peculiar features that deeply differentiate them. $B$-factories are characterized by typical simple events to reconstruct, and small production cross-sections. Instead hadronic collisions allows to study a larger fraction of $b$-physics sector and ensure much greater production cross-section for interesting events, but events are much more complex and huge underlying background is present. We summarize $B$-factory and hadronic collider main parameters, concerning flavor-physics production, in Table 2.1. Cross sections of $b\bar{b}$ pair production are calculated within the detector acceptance [31, 32].

| | $e^+e^- \to \Upsilon(4S) \to B\overline{B}$ | $pp \to b\bar{b}X$ |
|---|---|---|
| accelerator | CESR, PEP-II, KEKB | LHC (Run I) |
| detector | CLEO, BABAR, Belle | ATLAS, CMS, LHCb |
| $\sigma(b\bar{b})$ | $\sim 1\,\text{nb}$ | $\sim 75 - 150\,\mu\text{b}$ |
| $\sigma(b\bar{b})/\sigma(\text{bck})$ | $\sim 0.25$ | $\sim 0.005$ |
| typycal $(b\bar{b})$ rate | $10\,\text{Hz}$ | $\sim 30 - 100\,\text{kHz}$ |
| flavors | $B^0$ (50%),<br>$B^+$ (50%) | $B^0$ (40%), $B^+$ (40%),<br>$B^0_s$ (10%), $B^+_c$ ($< 0.1$%),<br>$b$-baryons (10%) |
| boost $< \beta\gamma >$ | 0.06-0.6 | 1-10 |
| pile-up events | 0 | 1-20 |
| track multiplicity | $\sim 5$ | $\mathcal{O}(100)$ |

Table 2.1: B-factory and hadronic collider main parameters concerning flavor physics production.

# Chapter 3

# LHCb experiment

## 3.1 LHC

The Large Hadron Collider (LHC) is a proton-proton and heavy ion collider [33] located at the CERN laboratory, on Swiss-French state border. The LHC is installed in a 27 km long circular tunnel, about 100 m underground. Protons are extracted from hydrogen gas and their energy are gradually increased by a series of accelerator machines, shown in Figure 3.1. Extracted protons are first accelerated by the Linac 2 up to an energy of 50 MeV, then by the Booster up to an energy of 1.4 GeV. The Proton Synchrotron (PS) and Super Proton Synchrotron (SPS) respectively accelerate them to an energy of 25 GeV and 450 GeV. Finally protons are injected in the LHC.

In the LHC, two proton or ion beams circulate in opposite directions in two separate beam pipes. Beams are bent by more of 1,200 superconducting dipole magnets 15 m long, cooled at temperature of 1.9 K by 120 tons of superfluid helium, which generate a magnetic field of 8.3 T.

Beams collide in four point placed along the LHC ring, where the detectors of the four major LHC experiments are installed. ATLAS and CMS are general-purpose experiments, while ALICE and LHCb are specifically dedicated to heavy-ion and heavy-flavor physics respectively. Other three smaller experiments are installed, TOTEM for the measure of total $pp$ cross section, LHCf to study astroparticle physics, and MoEDAL to look for magnetic monopole.

Proton beams are split in bunches each one consisting of about $10^{11}$ protons, and are time-spaced for a multiple of 25 ns corresponding to a bunch-crossing rate up to 40 MHz. The maximum number of bunches per beam is 2808, so the average bunch-crossing rate is $\sim$ 30 MHz. The peak istantaneous luminosity of the LHC project design is of $\mathcal{L} = 10^{34}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$ at a center of mass energy $E_{cm} = 14\,\mathrm{TeV}$.

As shown in figure 3.2, all the design parameters will be achieved in 2021 during the Run 3. In 2021 LHCb will receive a major upgrade (LHCb Upgrade 1a). The tracking of the LHCb upgrade will be discussed in Section 3.4. In 2024 LHC will be upgraded for increase significantly its luminosity entering in the High-Luminosity Large Hadron Collider (HL-LHC) era. Also ATLAS and CMS will be upgraded

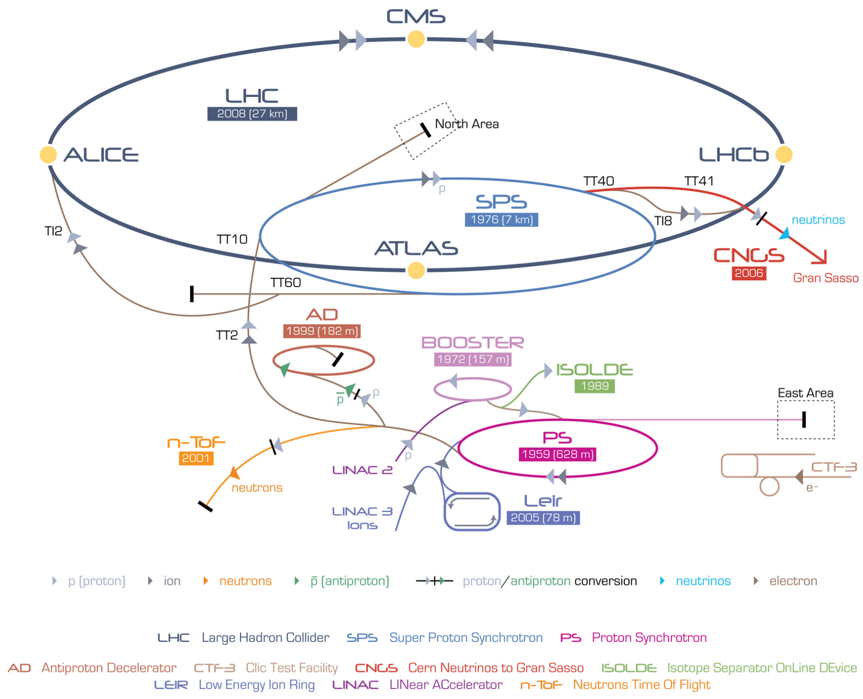

Figure 3.1: Cern Accelerator Complex.

to work in the new environment. In 2025 minor upgrade will be apply to LHCb (discussed in Section 3.5), and during Long Shutdown 4 LHCb will receive the LHCb Upgrade 2. Table 3.1 recaps LHC energies and luminosities for different runs. The schedule of the four major major LHC experiment are different, in this thesis I will use the schedule names of LHCb (Figure 3.3).

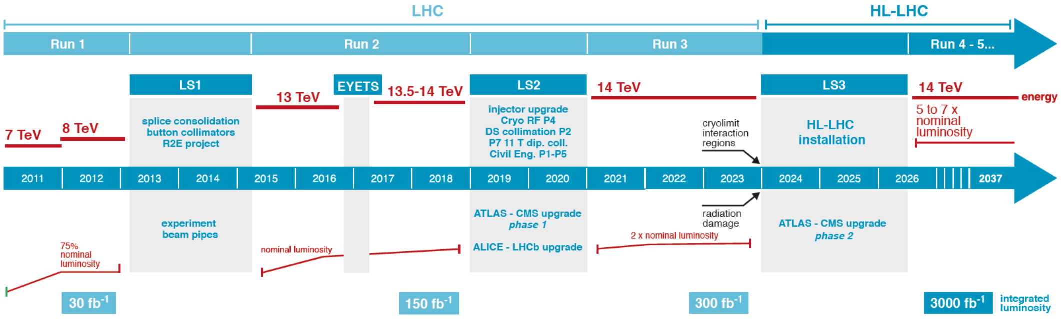

Figure 3.2: LHC schedule.

| LHC RUN | 2010–12<br>1 | 2015–18<br>2 | 2021–23<br>3 | 2026–29<br>4 | 2031-33<br>5 |
|---|---|---|---|---|---|
| $E_{cm}$ ( TeV) | $7-8$ | 13 | 14 | 14 | 14 |
| peak luminosity ( $\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$) | $7.7 \cdot 10^{33}$ | $1.7 \cdot 10^{34}$ | $2 \cdot 10^{34}$ | $7 \cdot 10^{34}$ | $7 \cdot 10^{34}$ |

Table 3.1: LHC parameters of $pp$ runs from 2010 to 2033.

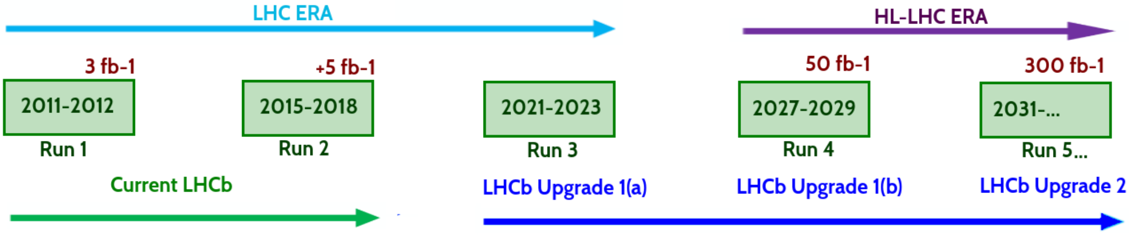

Figure 3.3: LHCb schedule.

## 3.2 LHCb and its Physics Program

The LHCb detector [34, 35] is a single-arm forward spectrometer covering the pseudorapidity range $2 < \eta < 5$, designed for the study of particles containing $b$- or $c$-quarks. The LHCb detector layout, shown in figure 3.4, is motivated by the fact that at high energies both $b$-hadrons are produced in the same forward or backward cone, as shown in figure 3.5.

The main goal of LHCb is find deviations from the SM prediction, these deviations are hints for NP. NP effect can be large in $b \to s$ transition, modifying the $B^0_s$ mixing phase $\phi_s$ measured from $B^0_s \to J/\psi\,\phi$ decays [36], or in channels dominated by other loop diagrams, like, for example, the very rare decay $B^0_s \to \mu^+\mu^-$ [37,38]. Another main goal is to perform a precise measurements of the CKM matrix elements. Therefore, the challenge of the future $b$ experiments is to widen the range of measured decays, reaching channels that are strongly suppressed in the SM, and to improve the precision of the measurements to achieve the necessary sensitivity to NP effects in loops. LHCb extends the $b$-physics results from the $B$-factories by studying decays of heavier $b$-hadrons, such as $B^0_s$ or $B^+_c$, which are copiously produced at the LHC.

To achieve these goals, LHCb detector includes a high-precision tracking system consisting of a silicon-strip vertex detector surrounding the $pp$ interaction region [39], a large-area silicon-strip detector located upstream of a dipole magnet with a bending power of about 4 Tm, and three stations of silicon-strip detectors and straw drift tubes [40] placed downstream of the magnet. Different types of charged hadrons are distinguished using information from two ring-imaging Cherenkov detectors [41]. Photons, electrons and hadrons are identified by a calorimeter system consisting of scintillating-pad and preshower detectors, an electromagnetic calorimeter and a

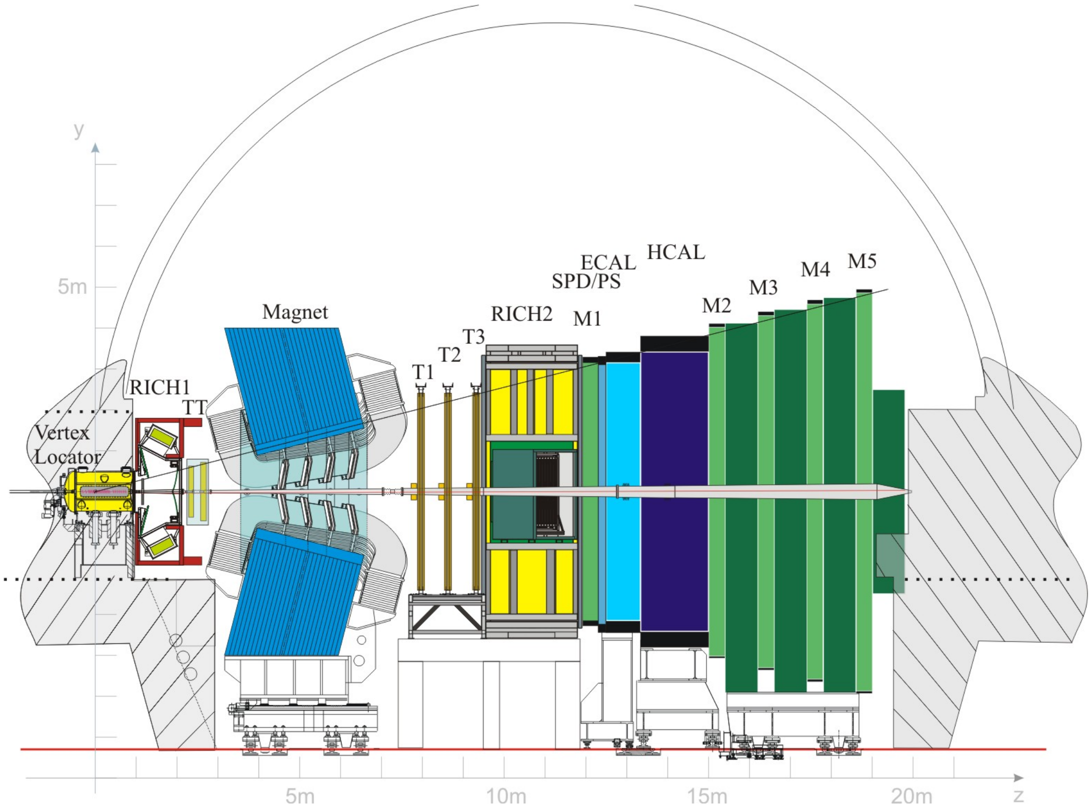

Figure 3.4: Layout of LHCb detector.

hadronic calorimeter. Muons are identified by a system composed of alternating layers of iron and multiwire proportional chambers [42]. LHCb adopts a right-handed coordinate system with $z$ coordinate along the beam, and $y$ coordinate along the vertical.

The nominal LHC luminosity value is reduced to $\mathcal{L} = 4 \cdot 10^{32}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$ in the LHCb intersection point. Lower luminosity is obtained by appropriately defocusing the beams by moving them apart transversely. This transverse separation is progressively modified during a fill, to keep the luminosity constant as the beam current decreases. The chosen luminosity value is optimized to obtain one or two inelastic interactions per bunch crossing according to trigger bandwidth, and for limit radiation damage.

During Run 1 LHCb collected $3\,\mathrm{fb}^{-1}$, and at the end of Run 2 is planned to collect an additional $5\,\mathrm{fb}^{-1}$. The LHCb collaboration already published $\sim 400$ papers, including the first evidence for the decay $B^0_s \to \mu^+\mu^-$ [43], studies of *CP* violation in various particle systems, and the observations of charmonium-pentaquark states in the $J/\psi\, p$ channel [44]. However many of the LHCb measurements remain limited by statistics, and other rare decay like $B^0 \to \mu^+\mu^-$ are not yet observed. LHCb will therefore undergo a major upgrade (Upgrade 1a) in the Long Shutdown 2 of LHC (see Section 3.4). The instantaneous luminosity will be increased to $\mathcal{L} = 2 \cdot 10^{33}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$, and the average number of primary $pp$ interactions per bunch crossing $\mu$ will be

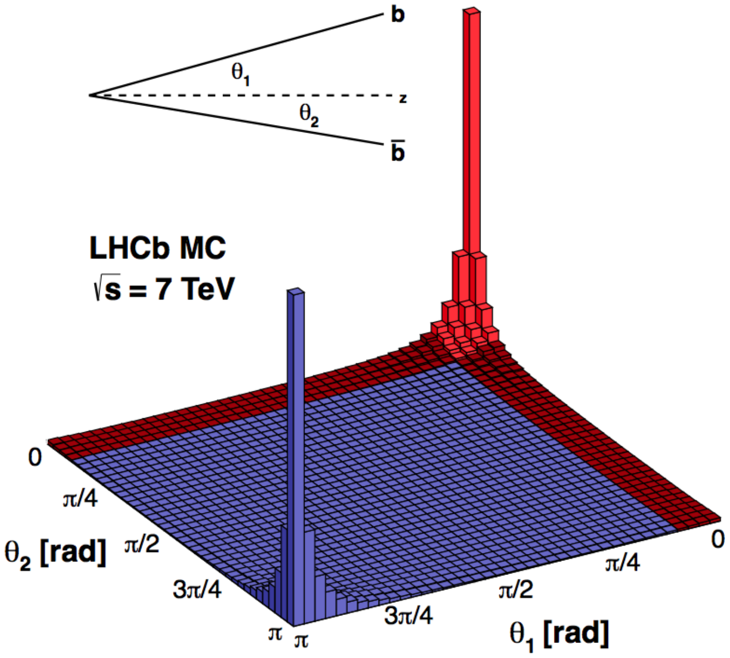

Figure 3.5: Angular correlation between $b$ and $\bar{b}$ quarks in $b\bar{b}$ pair production, simulated with PYTHIA event generator.

7.6. The goal is to collect $50\,\mathrm{fb}^{-1}$ of data during Run 3 and Run 4. With this amount of data the LHCb collaboration plans to increase the precision of many measurements and increase the sensitivities for many searches. Table 3.2 shows a list of key observables with the uncertainty prediction compared with current ones, both theoretical and experimental.

## 3.3 The current tracking at LHCb

The tracking system provides a measurement of momentum of charged particles, $p$, with a relative uncertainty that varies from 0.5% at 20 GeV/$c$ to 1.0% at 200 GeV/$c$. The minimum distance of a track to a primary vertex, the impact parameter, is measured with a resolution of $(15 + 29/p_{\mathrm{T}})$ µm, where $p_{\mathrm{T}}$ is the component of the momentum transverse to the beam, in GeV/$c$.

In the next sections I will describe the VErtex LOcator (VELO) detector used for reconstruct primary and secondary vertexes, the spectrometer composed by the Tracker Turicensis (TT) upstream a dipole magnet, the Inner Tracker (IT) and the Outer Tracker (OT) downstream the magnet. I will also describe the trigger system.

| Type | Observable | Current<br>precision | LHCb<br>2018 | **Upgrade**<br>($50\,\mathrm{fb}^{-1}$) | Theoretical<br>uncertainty |
|---|---|---|---|---|---|
| $B_s^0$ mixing | $2\beta_s\ (B_s^0 \to J/\psi\ \phi)$ | 0.10 [45] | 0.025 | 0.008 | $\sim 0.003$ |
| | $2\beta_s\ (B_s^0 \to J/\psi\ f_0(980))$ | 0.17 [46] | 0.045 | 0.014 | $\sim 0.01$ |
| | $A_{\mathrm{fs}}(B_s^0)$ | $6.4 \times 10^{-3}$ [47] | $0.6 \times 10^{-3}$ | $0.2 \times 10^{-3}$ | $0.03 \times 10^{-3}$ |
| Gluonic | $2\beta_s^{\mathrm{eff}}(B_s^0 \to \phi\phi)$ | – | 0.17 | 0.03 | 0.02 |
| penguin | $2\beta_s^{\mathrm{eff}}(B_s^0 \to K^{*0}\bar{K}^{*0})$ | – | 0.13 | 0.02 | $< 0.02$ |
| | $2\beta^{\mathrm{eff}}(B^0 \to \phi K_S^0)$ | 0.17 [47] | 0.30 | 0.05 | 0.02 |
| Right-handed | $2\beta_s^{\mathrm{eff}}(B_s^0 \to \phi\gamma)$ | – | 0.09 | 0.02 | $< 0.01$ |
| currents | $\tau^{\mathrm{eff}}(B_s^0 \to \phi\gamma)/\tau_{B_s^0}$ | – | 5 % | 1 % | 0.2 % |
| Electroweak | $S_3(B^0 \to K^{*0}\mu^+\mu^-; 1 < q^2 < 6\,\mathrm{GeV}^2/c^4)$ | 0.08 [48] | 0.025 | 0.008 | 0.02 |
| penguin | $s_0\,A_{\mathrm{FB}}(B^0 \to K^{*0}\mu^+\mu^-)$ | 25 % [48] | 6 % | 2 % | 7 % |
| | $A_{\mathrm{I}}(K\mu^+\mu^-; 1 < q^2 < 6\,\mathrm{GeV}^2/c^4)$ | 0.25 [49] | 0.08 | 0.025 | $\sim 0.02$ |
| | $\mathcal{B}(B^+ \to \pi^+\mu^+\mu^-)/\mathcal{B}(B^+ \to K^+\mu^+\mu^-)$ | 25 % [50] | 8 % | 2.5 % | $\sim 10\,\%$ |
| Higgs | $\mathcal{B}(B_s^0 \to \mu^+\mu^-)$ | $1.5 \times 10^{-9}$ [51] | $0.5 \times 10^{-9}$ | $0.15 \times 10^{-9}$ | $0.3 \times 10^{-9}$ |
| penguin | $\mathcal{B}(B^0 \to \mu^+\mu^-)/\mathcal{B}(B_s^0 \to \mu^+\mu^-)$ | – | $\sim 100\,\%$ | $\sim 35\,\%$ | $\sim 5\,\%$ |
| Unitarity | $\gamma\ (B \to D^{(*)}K^{(*)})$ | $\sim$ 10–12° [15, 52] | 4° | 0.9° | negligible |
| triangle | $\gamma\ (B_s^0 \to D_s K)$ | – | 11° | 2.0° | negligible |
| angles | $\beta\ (B^0 \to J/\psi\ K_S^0)$ | 0.8° [47] | 0.6° | 0.2° | negligible |
| Charm | $A_\Gamma$ | $2.3 \times 10^{-3}$ [47] | $0.40 \times 10^{-3}$ | $0.07 \times 10^{-3}$ | – |
| *CP* violation | $\Delta A_{CP}$ | $2.1 \times 10^{-3}$ [53] | $0.65 \times 10^{-3}$ | $0.12 \times 10^{-3}$ | – |

Table 3.2: Statistical sensitivities of the LHCb upgrade to key observables [54]. For each observable the current sensitivity is compared to that which will be achieved by LHCb before the upgrade, and that which will be achieved with $50\,\mathrm{fb}^{-1}$ by the upgraded experiment. Systematic uncertainties are expected to be non-negligible for the most precisely measured quantities.

### 3.3.1 VErtex LOcator

The VELO detector measures charged particle trajectories in the region closest to the interaction point. Its main purpose is to reconstruct primary and secondary vertexes with a spatial resolution smaller than typical decay lengths of $b$- and $c$-hadrons in LHCb ($c\tau \sim 0.01$ - $1\,\mathrm{cm}$), in order to discriminate between them. Therefore it plays a fundamental role for discriminating heavy flavors signals from the underlying background, especially at the High Level Trigger (see Section 3.3.6).

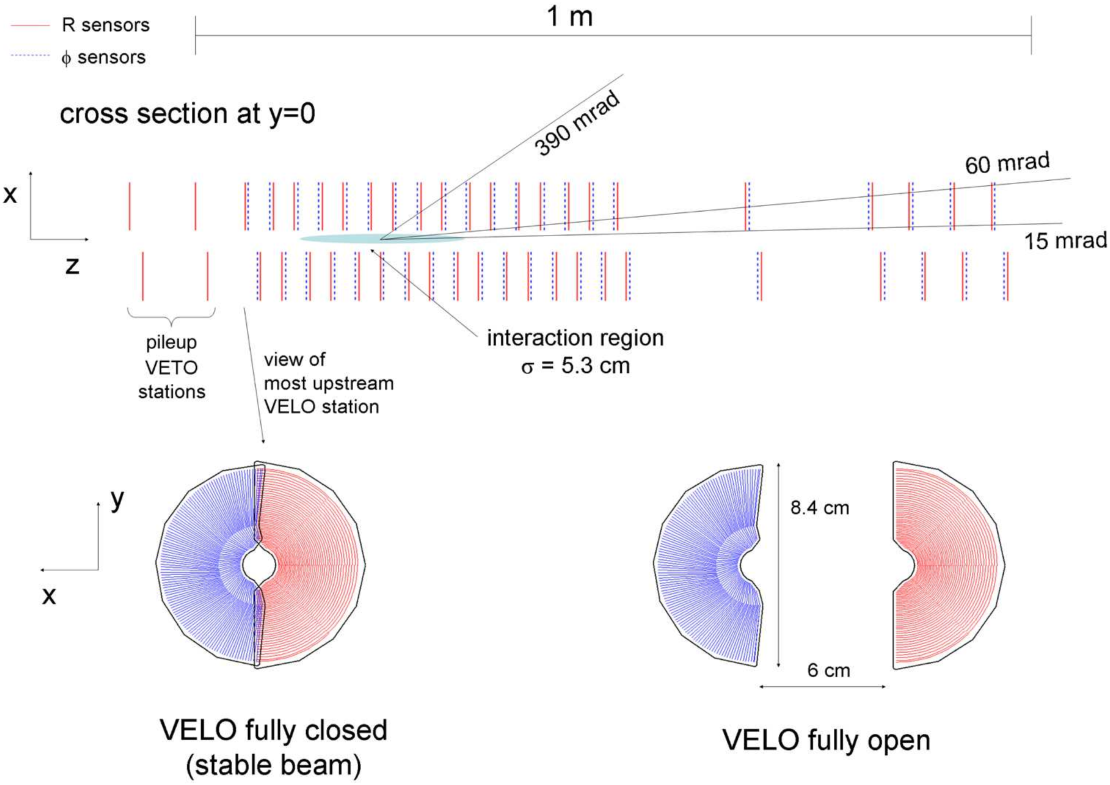

Figure 3.6: Representation of VELO detector, with a transverse view of a VELO station in closed and open configurations.

The VELO consists of 21 disk-shaped stations installed along the beam axis inside the beam pipe, both upstream ($z > 0\,\mathrm{cm}$) and downstream ($z < 0\,\mathrm{cm}$) of the nominal interaction point. Figure 3.6 shows the layout of the system. Stations placed at $z > 0\,\mathrm{cm}$ provide precise measurements of vertexes positions. While the stations at $z < 0\,\mathrm{cm}$ constitute the pile-up veto system, which provides position of primary vertices candidates along the beam-line and measures the total backward charged track multiplicity. The stations are made by two type of silicon strip sensors, the $r$ and $\phi$ sensors, arranged with radial and azimutal segmentation to measure $r$ and $\phi$ particle intersection coordinates. Each station is divided into two retractile halves, called modules, as shown in Figure 3.6. Each halves consists of both $r$ and $\phi$

sensors. VELO veto stations consist of $r$ sensors only. The retractile halves allow to move the sensors away from the beam, to do not damage silicon sensors during LHC injection phases, when VELO stations are “opened” and the sensors have a minimum distance of 30 mm from the beam axis, instead, when stable beams are circulating for data taking, station are “closed” and the sensors reach a minimum distance of 5 mm from the beam axis.

Both $r$ and $\phi$ sensors are centered around the nominal beam position, and are covering a region between 8 and 42 mm in radius. The $r$ sensors consist of semicircular, concentric strips with increasingly pitch from 38 $\mu$m at the innermost radius to 102 $\mu$m at the outermost radius. The $\phi$ sensors are subdivided in two concentric regions: the inner one covers a radius $r$ between 8 and 17.25 mm, the outer one covers $r$ between 17.25 and 42 mm with pitch linearly increasing from the center. $\phi$ sensors are designed with an angular tilt of +10° in the inner region and -20° in the outer region, respect to the radial direction; for adjacent sensors, the tilt is reversed. This layout is designed to improve pattern recognition and to better distinguish noise from genuine hits. Each VELO module is encased in a shielding box, to protect it from the radiofrequency electric field. The individual hit resolution of the sensors is strongly correlated to the sensor pitch and projected angle, that is the angle perpendicular to the strip direction. Raw hit resolution varies from $\approx$ 10 µm for smallest pitch to $\approx$ 25 µm for biggest pitch.

### 3.3.2 Tracker Turicensis

The TT uses silicon microstrip sensors, with a strip pitch of 183 µm. The sensors are 500 µm thick, 9.64 cm wide and 9.44 cm long. TT is located upstream the dipole magnet, and covers the full acceptance of the experiment ($\approx$ 300 mrad). It is designed for reconstructing low-momentum tracks that are swept out of the detector acceptance by the magnet. The TT consists of one tracking station subdivided in four layers in a $x$-$u$-$v$-$x$ arrangement, with vertical strips in first and last layers, and tilted strips by a stereo angle of -5° and of +5° in central layers. Each TT layer is subdivided in two half-modules, each consisting of seven silicon sensors. TT layout is shown in Figure 3.7. Single-hit resolution is of $\approx$ 50 µm.

### 3.3.3 The dipole magnet

The LHCb warm dipole magnet generates an integrated field, of about 4 Tm, manly along the vertical direction and between $z = 3$ and 8 m. A fringe field is present in the region where the tracking detectors are installed, between $z = 0$ and 10 m. The dipole magnet consists of two identical coils each one formed by 15 laminated low carbon-steel plates, 10 cm thick. The coils, weighting a total of 54 tons, are symmetrically installed in a iron yoke of 1500 tons. A magnet perspective view is proposed in Figure 3.8. Overall dimensions of the dipole magnet are of 1 m x 8 m x 5 m. The magnet dissipates an electric power of 4.2 MW, and the nominal current in conductor material is of 5.85 kA while the maximum permitted current is of 6.6 kA.

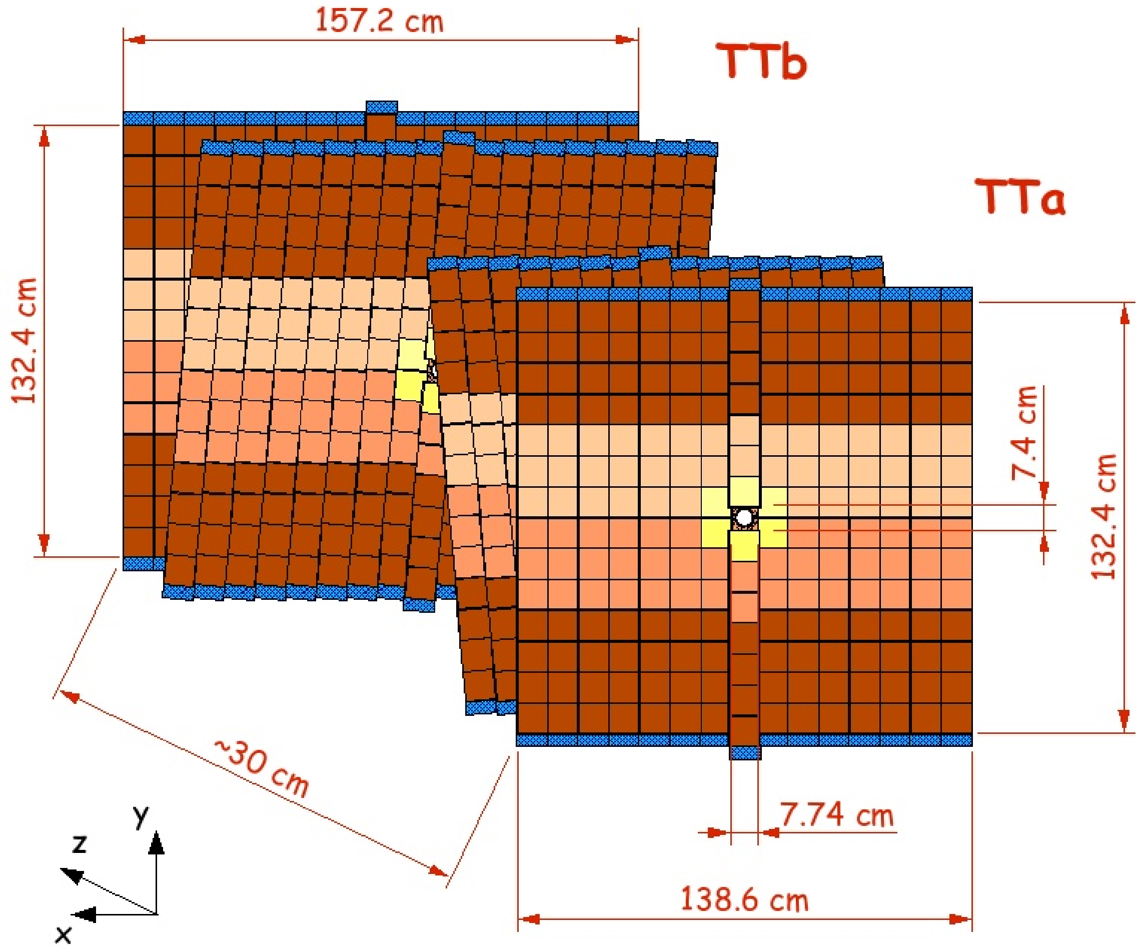

Figure 3.7: Layout of TT.

Current in the magnet, and therefore the field direction, are periodically inverted, to reduce systematic effect in precision measurements of *CP* asymmetries.

To provide a good particle momentum reconstruction, the magnetic field intensity must be known with great precision. An array of 180 Hall probes, calibrated to a relative precision of $10^{-4}$ on field intensity measurement, allow to achieve a field mapping with measurement precision of about $4 \cdot 10^{-4}$ in the entire tracking volume. Measured vertical component of this magnetic field, $B_y$, is shown in Figure 3.9.

### 3.3.4 Inner Tracker

The IT detectors is located downstream the dipole magnet, and it consists of 3 cross-shaped stations, it covers an acceptance of $\sim$ 150-200 mrad in the bending plane and of $\sim$ 40-60 mrad in the $yz$ plane. The IT reconstruct tracks that passed through

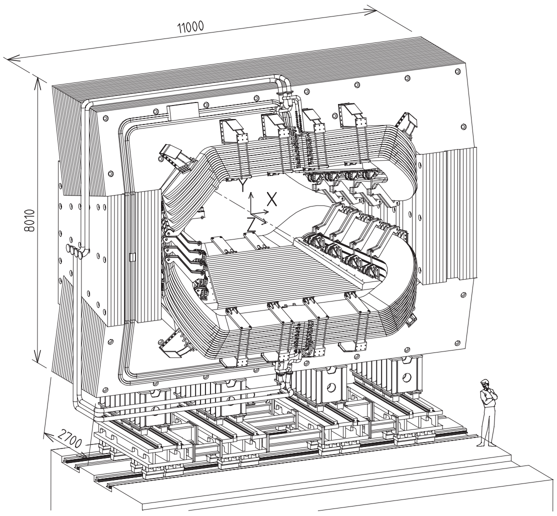

Figure 3.8: Perspective view of LHCb dipole magnet.

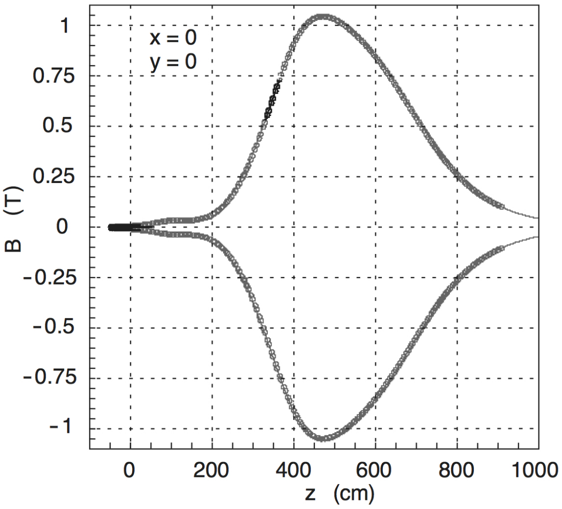

Figure 3.9: Measured $B_y$ component of LHCb magnetic field.

the magnetic field region lying near the beam axis. It uses the same microstrip sensors used in the TT. Like the TT, each IT station is subdivided in four layers in a *x*-*u*-*v*-*x* arrangement. Each IT layer consists of four subunits, positioned around the beam pipe, and each subunits includes seven modules. In the subunits above and below the beam pipe a module corresponds to one silicon sensor, while subunits on right and left have modules with two silicon sensors each one. IT layout is shown in Figure 3.10. Single-hit resolution of IT detectors is of $\approx 50\,\mu$m.

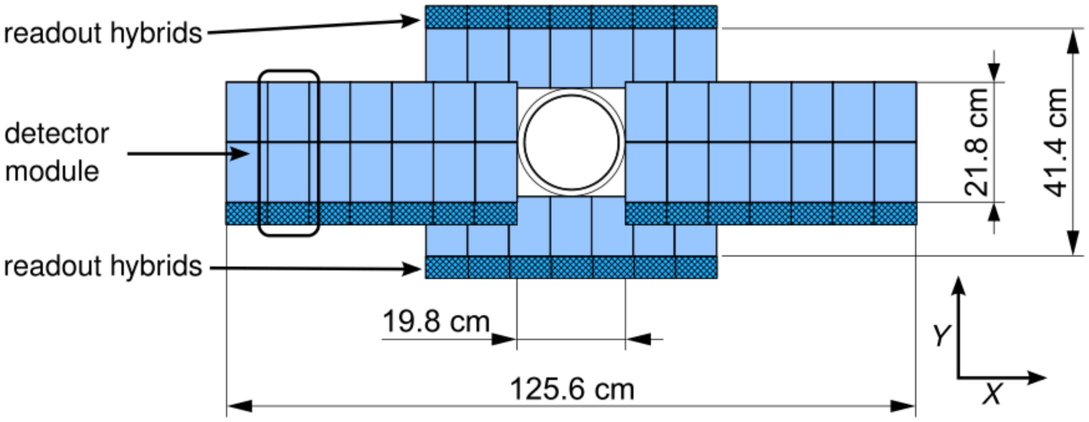

Figure 3.10: Layout of one IT layer.

### 3.3.5 Outer Tracker

The OT is used to measure track bending in the acceptance region not covered by the IT subdetector. The OT consist of three straw tubes stations, each station is located downstream a IT station, which together form a T-station. OT layout is shown in Figure 3.11. Each OT station is subdivided in four layers $x$-$u$-$v$-$x$. Each layer is subdivided in modules, consisting of 64 straw tubes. Straw tubes are filled with a mixture of 70% Ar and 30% $CO_2$, with a drift time up to 50 ns. The straw tubes allow to reconstruct tracks with a spatial resolution of $\approx 200\,\mu$m.

### 3.3.6 Trigger

The LHCb trigger was designed to select heavy-flavor decays from the huge light-quark background, sustaining the LHC bunch-crossing rate of 40 MHz and selecting up to 5 kHz of data to store [55]. Only a small fraction of events, about 15 kHz, contains a $b$-hadron decay with all final state particles emitted in the detector acceptance. The rate of “interesting” bottom hadron decays is even smaller, of a few Hz. Corresponding values for charmed hadrons are about 20 times larger. It is therefore crucial, for the trigger, to reject background as early as possible in the data flow.

The LHCb trigger is organized into two sequential stages, the L0 trigger and the High Level Trigger (HLT). This two-level structure helps coping with timing and selection requirements, with a fast and partial reconstruction at low level, followed by a more accurate and complex reconstruction at high level. The hardware-based L0 trigger operates synchronously with the bunch crossing. It uses information from calorimeter and muon detectors to reduce the 40 MHz bunch-crossing rate to below 1.1 MHz, which is the maximum value at which the whole detector can be read out by design. Then, the asynchronous software-based HLT performs a finer selection based on information from all detectors, and reduces rate to 5 kHz, after an upgrade

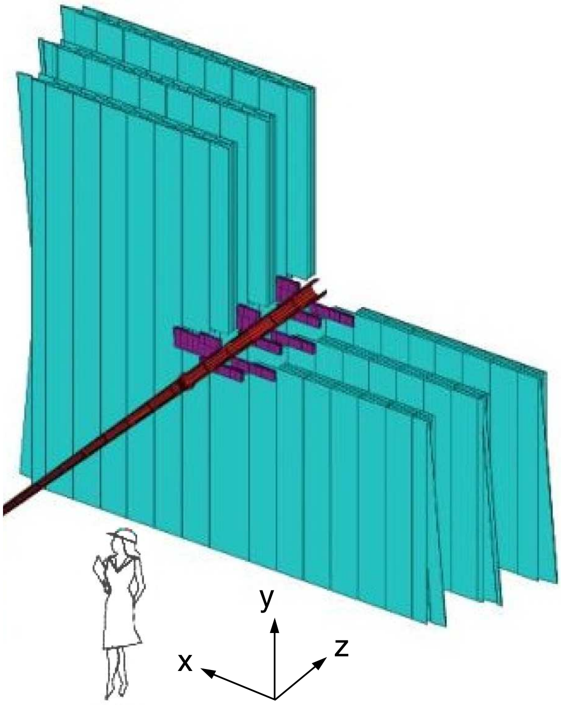

Figure 3.11: Layout of OT subdetector. The IT in purple.

to the storage in Run 2 the output rate was increased to 12 kHz. Figure 3.12 shows the LHCb trigger flow for Run 1 and Run 2, and typical event-accept rates for each stage.

### The L0 trigger

The task of L0 trigger is to reduce the event rate from 40 MHz (same as the bunch-crossing rate) to 1 MHz, that is the maximum rate at which the full detector can be read. Data from all detectors are stored in memory buffers consisting of an analog pipeline that is read out with a fixed latency of 4 μs. The L0 decision must be available within this fixed time, therefore the L0 trigger is entirely based on custom-built electronic boards, relying on parallelism and pipelining. At this stage, trigger requests can only involve simple and immediately available quantities, like those provided by calorimeter and muon detectors. The L0 trigger consists of three independent trigger decisions, the *L0 hadron*, the *L0 muon*, the *L0 calorimeter*. Each decision is combined with the others through a logic "or" in the L0 decision unit.

The L0 hadron trigger aims at collecting samples enriched in hadronic $c$- and $b$-particle decays. Final-state particles from such decays have on average higher transverse momenta than particles originated from light-quark processes, and this property helps in discriminating between signal and background.

The L0 muon trigger uses the information from the five muons stations, to identify

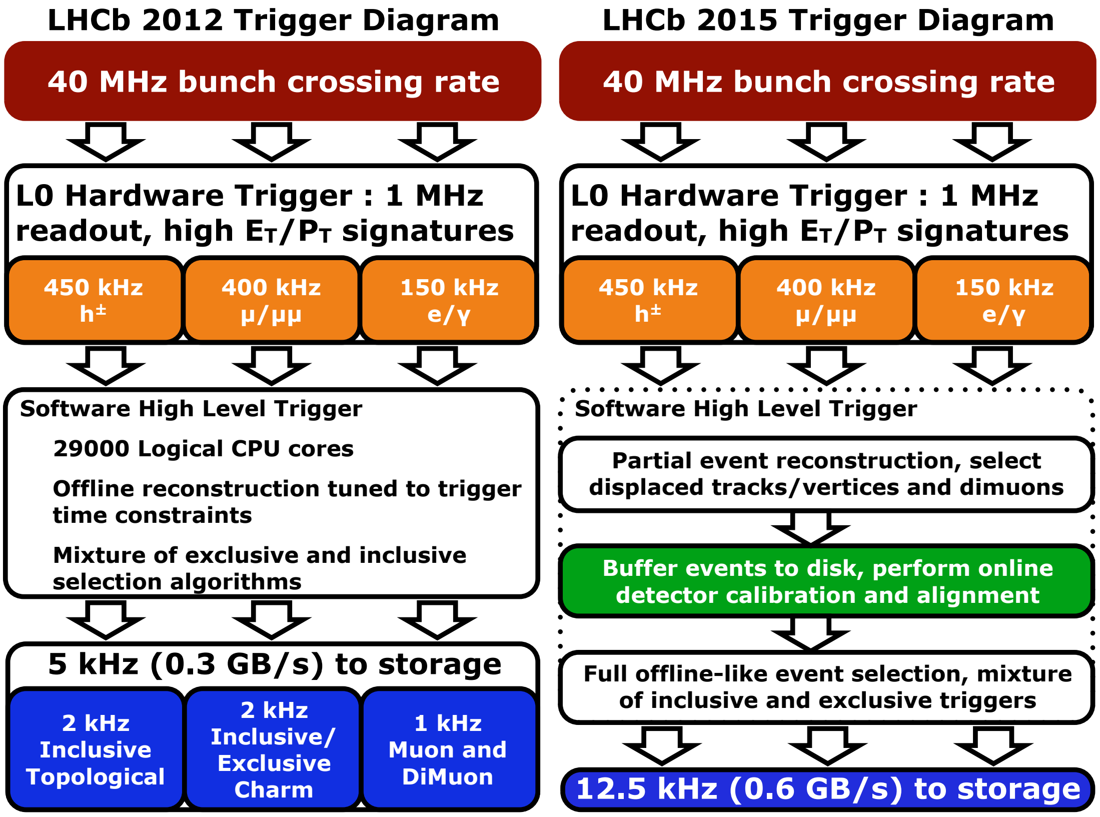
LHCb 2012 Trigger Diagram
40 MHz bunch crossing rate
L0 Hardware Trigger : 1 MHz readout, high E_T/P_T signatures
450 kHz h±
400 kHz μ/μμ
150 kHz e/γ
Software High Level Trigger
29000 Logical CPU cores
Offline reconstruction tuned to trigger time constraints
Mixture of exclusive and inclusive selection algorithms
5 kHz (0.3 GB/s) to storage
2 kHz Inclusive Topological
2 kHz Inclusive/ Exclusive Charm
1 kHz Muon and DiMuon
LHCb 2015 Trigger Diagram
40 MHz bunch crossing rate
L0 Hardware Trigger : 1 MHz readout, high E_T/P_T signatures
450 kHz h±
400 kHz μ/μμ
150 kHz e/γ
Software High Level Trigger
Partial event reconstruction, select displaced tracks/vertices and dimuons
Buffer events to disk, perform online detector calibration and alignment
Full offline-like event selection, mixture of inclusive and exclusive triggers
12.5 kHz (0.6 GB/s) to storage

Figure 3.12: Representation of LHCb trigger flow and typycal event-accept rates for each stage.

the most energetic muons. Once the two muons candidates with highest transverse momentum per quadrant of the muons detectors are identified, the trigger decision depends on two thresholds: one on the highest transverse momentum (L0 muon) and one on the product of the two highest transverse momenta (L0 dimuon).

The L0 calorimeter trigger uses the information from the electromagnetic calorimeter, the hadron calorimeter, the preshower detector, and the scintillator pad detector. It calculates the transverse energy $E_{\mathrm{T}}$ deposited in a cluster of 2x2 cells of the same size, for both the electromagnetic calorimeter and the hadron calorimeter. The transverse energy is combined with information on the number of hits on preshower and scintillator pad detectors to define three types of trigger candidates, photon, electron, and hadron.

### The High Level Trigger

Events accepted at L0 are transferred to the Event Filter Farm (EFF), an array of computers consisting of more than 15,000 commercial processors, for the HLT stage. The HLT is implemented through a C++ executable that runs on each processor of the farm, reconstructing and selecting events in a way similar to the offline processing. A substantial difference between online and offline algorithms is the time available to completely reconstruct a single event. The offline reconstruction requires almost 2 s per event in average, while the maximum time available for the online reconstruction is typically 50 ms, determined by the L0 event-accept rate (870 kHz in 2011) and the computing power of the farm.

The HLT consists of several trigger selections designed to collect specific events, in particular, *c* or *b*-hadron decays. Every trigger selection is specified by reconstruction algorithms and selection criteria that exploit the kinematic features of charged and neutral particles, the decay topology, and the particle identities. The HLT processing time is shared between two different levels, a first stage called HLT1 and a second stage HLT2. A partial event reconstruction is done in the first stage in order to reduce the event accept rate to 30 kHz, and a more complete event reconstruction follows in the second stage.

At the first level, tracks are reconstructed in the VELO and selected based on their probability to come from heavy-flavor decays, by determining their impact parameter with respect to the closest primary vertex. At the second level, a complete forward tracking (see following section) of all tracks reconstructed in the VELO is performed. Secondary vertex reconstruction is performed and requirements on decay length and mass are applied to reduce the event-accept rate to 5 kHz, at which events are stored. Several trigger selections, inclusive and exclusive, are available at this stage.

### 3.3.7 LHCb tracking

The LHCb tracking reconstruction is currently performed in stages [56]. First, tracks are reconstructed as straight lines using the R sensors of the VELO. Then, hits from the $\phi$ VELO sensors are added to these tracks. Two different algorithms are used to combine these VELO tracks with hits in the other tracking stations. The first method propagates VELO tracks through the magnetic field, and adds hits in the downstream tracking stations (forward). The second method finds straight track segment in the downstream tracking stations (track seeds) and then attempts to propagate them in the opposite direction, matching them to VELO tracks (backward). Finally, hits from the TT are added to the track to improve the momentum resolution and reject incorrect combinations of hits.
Within the LHCb tracking environment, tracks are classified as follows:

- a track reconstructed both in VELO and T-stations subdetectors is called “long track”;

- a track reconstructed both in VELO and TT subdetectors is called “upstream track”;

- a track reconstructed on TT and T-stations subdetectors is called “downstream track”;

- a track reconstructed on T-stations only is called “T-track”;
- a track reconstructed on VELO only is called “VELO-track”.

Figure 3.13 shows a representation of this track classification.

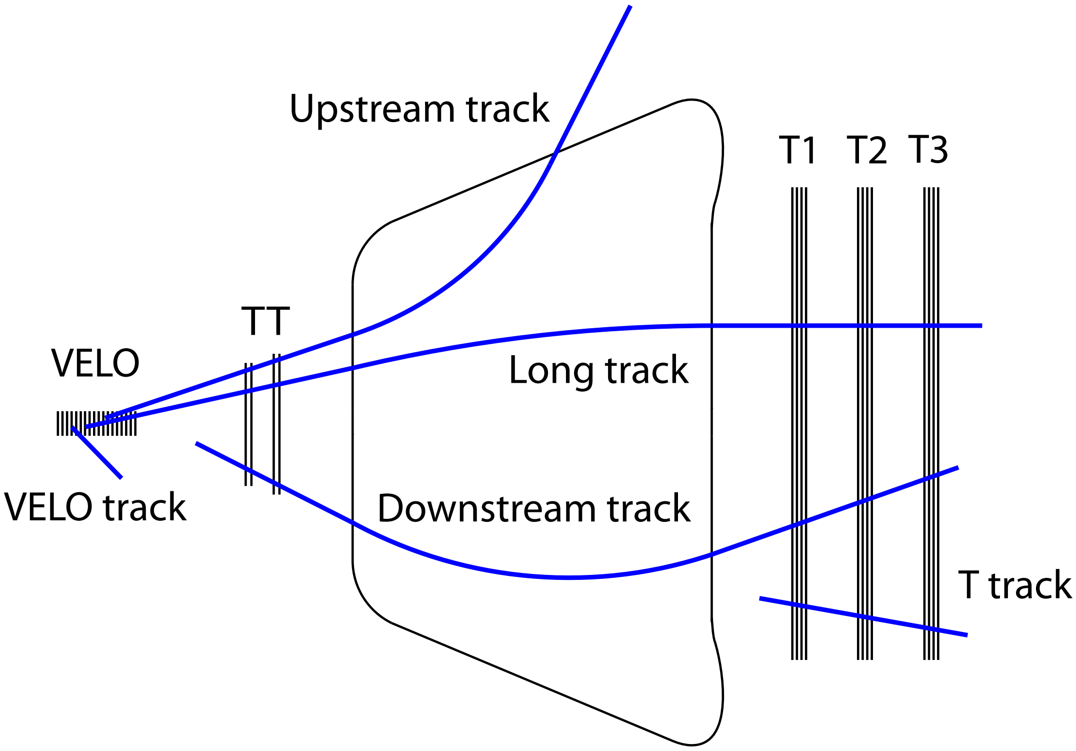

Figure 3.13: lhcb tracking scheme.

## 3.4 The tracking system in LHCb Upgrade 1a

With the intent of collect $50\,\mathrm{fb}^{-1}$ in Run 3 and Run 4, during the Long Shutdown 2 of the LHC collider $(2019-2020)$, the LHCb experiment will receive substantial upgrades concerning both detector and online systems [54]. After the upgrade, the readout rate will be 40 MHz instead of the current frequency of 1.1 MHz. This will allow a huge increase of data rate, leading to important improvements in annual signal yields, but will also enormously increase the demands on EFF and off-line processing. All upgrades must take into account the new experimental environment, with a center-of-mass energy of $\sqrt{s} = 14\,\mathrm{TeV}$ and an important increase of luminosity, set to $\mathcal{L} = 2\cdot 10^{33}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$ in order to reach the desired goal of $50\,\mathrm{fb}^{-1}$ collected. This results in a much higher track multiplicity then nowadays, and in an average number of primary $pp$ interactions per bunch crossing equal to $\mu = 7.6$ that will require new detectors with greater granularity to maintain a good track reconstruction performance.

The VELO and the TT will be replaced respectively by the VELOPIX and the Upstream Tracker (UT). The IT and the OT will be replaced by the Scintillating Fibre Tracker (SciFi). In addition, the amount of off-line processing that will be feasible for each event in these new conditions will be more limited than in the past, and the plan is to perform most, if not all, of the tracking reconstruction work within the HLT process in real-time. This further increases the demands on the EFF, and will require some compromise on reconstruction strategies, as it will be further discussed later.

### 3.4.1 VELOPIX detector

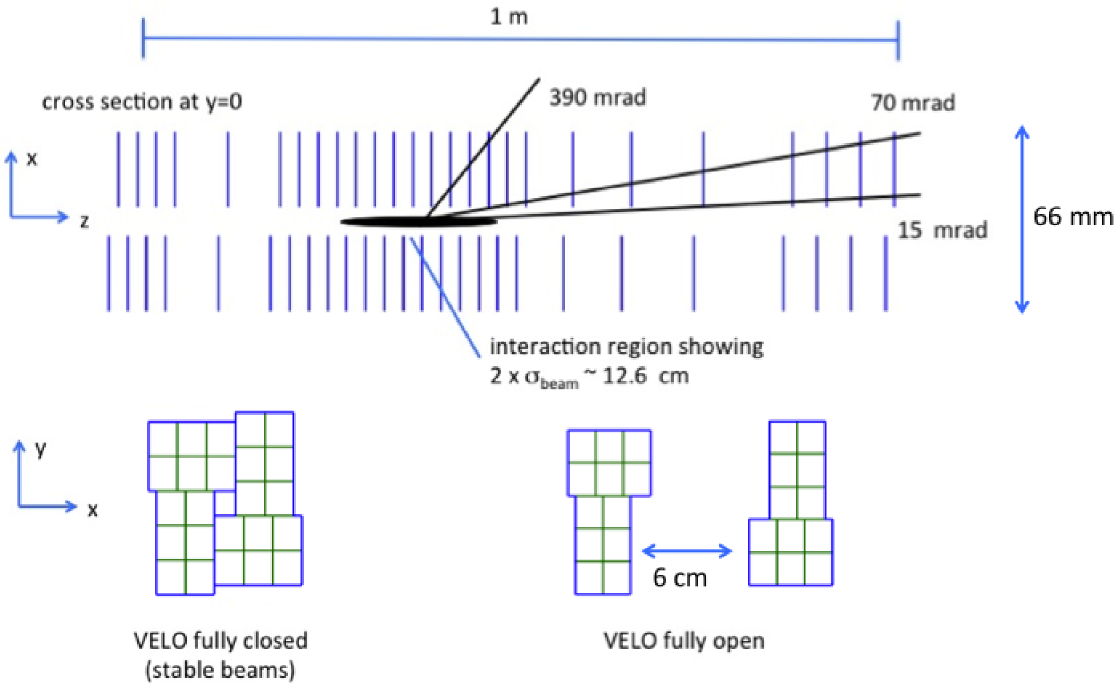

Figure 3.14: Layout of VELOPIX.

The VELO will be replaced by a dector based on silicon pixel technology (VELOPIX [57]). The VELOPIX consists of 26 tracking layers, as shown in Figure 3.14, and two of them are pile-up stations used to measure backward track multiplicity. Each station is subdivided in two modules, with the ability of distancing them from the beam axis such as for the current VELO detector. Each module contains four silicon sensors with an active area of 42.46 x 14.08 mm$^2$. The pixel sizes are 55 x 55 µm$^2$ and the entire VELOPIX detector includes about 41 M pixels. The inner radius of sensitive area from beam axis will be reduced from current $r = 8.2$ mm to less of

$r = 5.1\,\text{mm}$, to improve impact parameter resolution. The single hit resolution is expected to be about $12 - 15\,\mu\text{m}$ for both $x$ and $y$ coordinates.

### 3.4.2 Upstream Tracker

The current TT will be replaced by the UT [58], a new detector consisting of four planes of silicon micro-strips. With respect to the TT, UT planes use thinner sensors ($250\,\mu\text{m}$ *vs.* $500\,\mu\text{m}$) with finer segmentation in the central region ($95\,\mu\text{m}$ *vs.* $183\,\mu\text{m}$), and provide a larger acceptance coverage. UT planes are arranged in a $x$-$u$-$v$-$x$ configuration, with vertical strips in the first and last layers, and tilted strips by a stereo angle of -5° and of +5° in the central layers. Pitches and lengths of sensors vary depending on their position. Around the beam pipe, sensors with $95\,\mu\text{m}$ pitch and 5 cm long are used, while in central areas we have sensors with $95\,\mu\text{m}$ pitch and 10 cm long. Finally, more externally sensors with $190\,\mu\text{m}$ pitch and 10 cm long are used. Figure 3.15 shows the UT layout. Angular coverage of UT detector is of 314(248) mrad in the bending (non bending) plane.

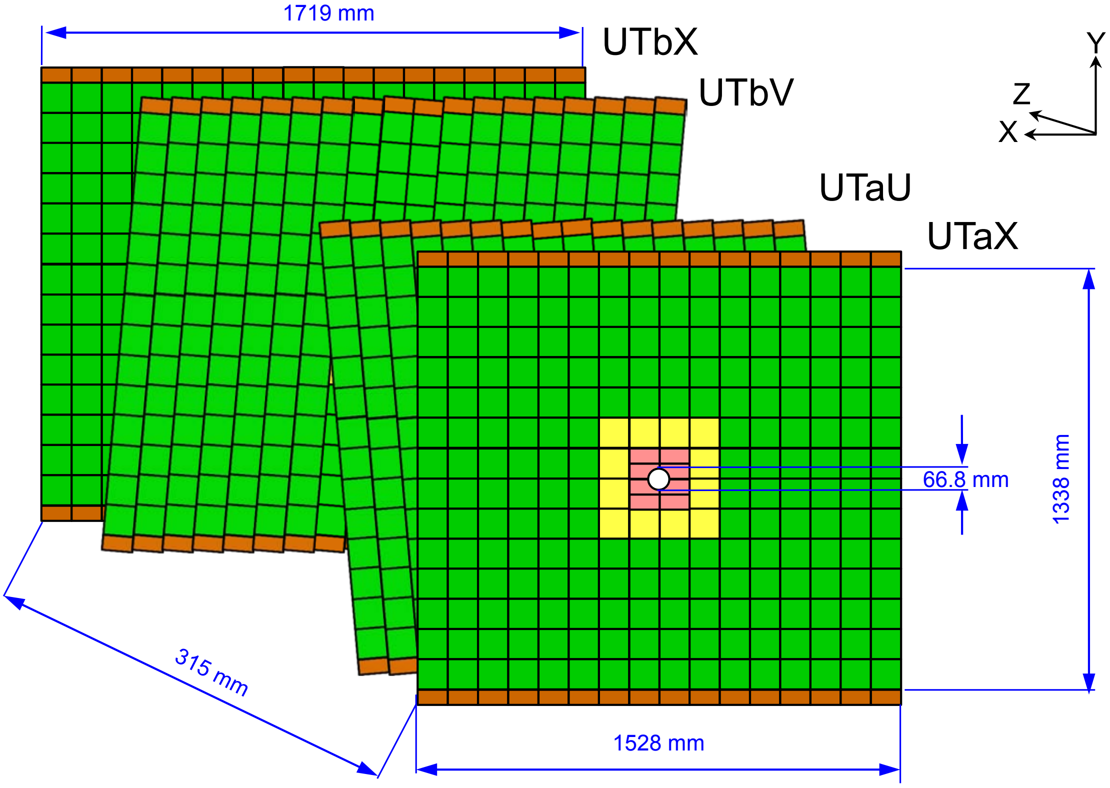

Figure 3.15: Layout of UT detector.

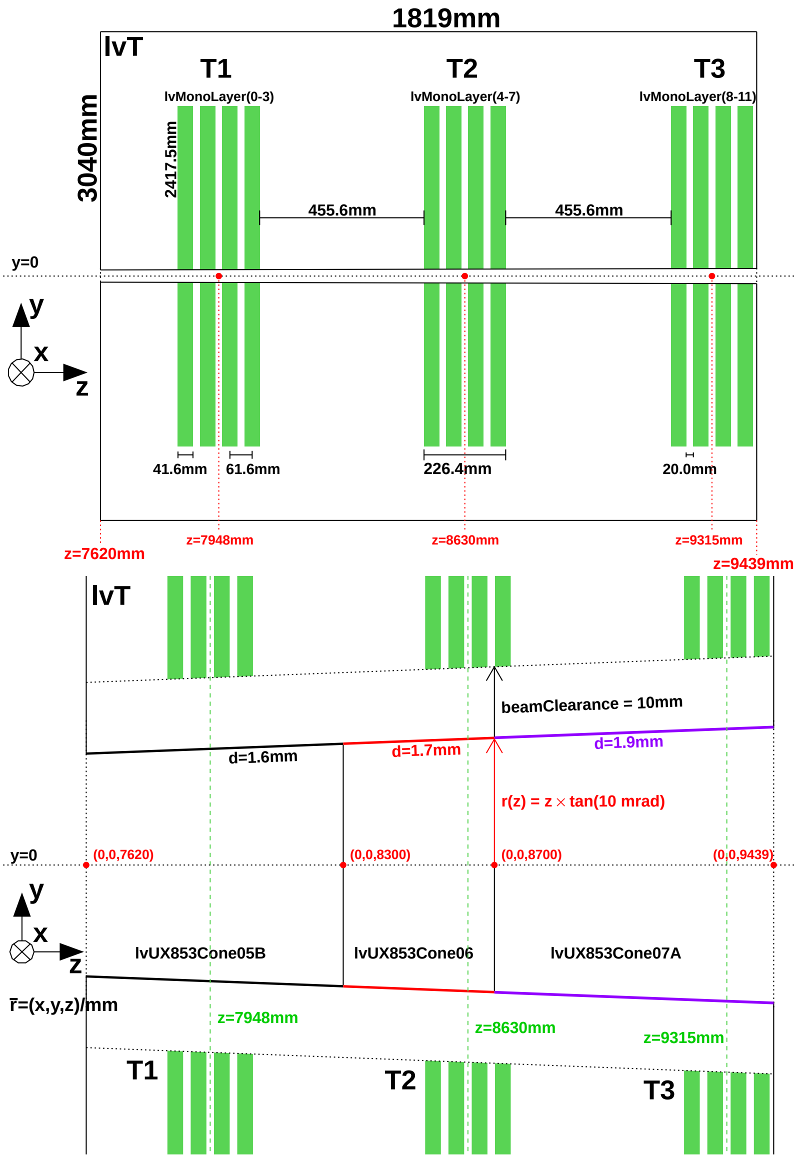

Figure 3.16: Arrangement of the SciFi within the tracker volume.

### 3.4.3 Scintillating Fibre Tracker

After the dipole magnet, the new SciFi will replace current IT and OT stations [58]. The SciFi (shown is Figure 3.16) is composed of 2.5 m long fibers read out by silicon photo-multipliers outside the detector acceptance. SciFi detector consists of three stations coinciding with the nominal positions of current OT stations. Each SciFi station includes 4 tracking layers arranged in a $x$-$u$-$v$-$x$ configuration, with $u$ and $v$ layers tilted respectively by -5° and of +5° respect to the vertical axis. The layer is broken into modules that are 5 m in height, with a width of 0.52 m, resulting in 12 modules per plane. There is a 3 mm gap between modules; the inefficiency due to geometrical gaps and single dead channels is expected to be 1%. There will be two basic types of modules, as shown in Figure 3.17: beam-pipe and non-beam-pipe modules. The beam-pipe modules will require special modifications to accommodate the beam-pipe and they will have six fibre layers. Due to lower irradiation received, non-beam-pipe modules with only five layer fibre mats. Scintillating fibers have circular cross-section and a total diameter of 0.25 mm. A fiber consists of a polymer core, with the addition of an organic fluorescent dye for about $\sim 1\%$ of the fiber weight. Light is produced by excitation of the polymer core, and propagates through the fiber by total internal reflection. The decay time of the scintillation light is $\approx 3\,\mathrm{ns}$; the propagation time of light along the fiber is $6\,\mathrm{ns/\,m}$. The simulated hit detection efficiency at the end of the lifetime of the detector is above 97.4%

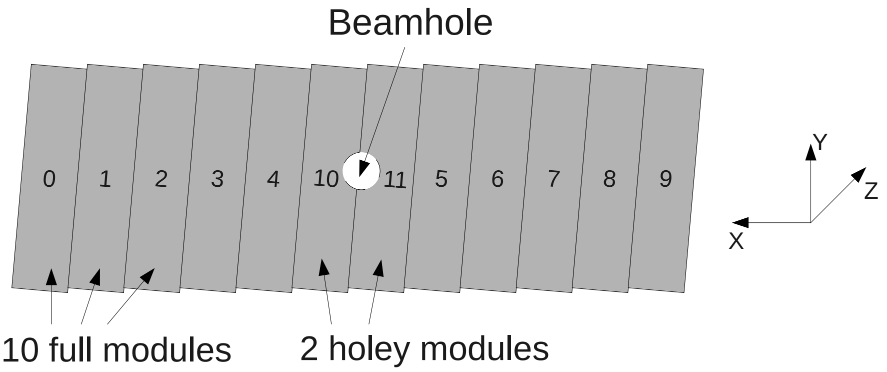

Figure 3.17: The structure of one layer is made up of 12 Fibre modules.

### 3.4.4 The Data acquisition in the Upgrade

The Trigger system and event reconstruction will be fully software based on the HLT [59]. Figure 3.18 shows the trigger scheme for LHCb Upgrade 1a. All needed hardware will be installed on the surface, and it will be consist of an Event-Builder (EB) and the EFF. The full system is visible in Figure 3.19.

#### The Event-Builder

The EB collects data from the detector, and sends data packets to the EFF. Data movement inside the EB is performed by commercial CPUs organized in 500 rack-mounted PC boxes [59]. Every node will mount a PCIe40 board for receiving data from the detector and two network interfaces, one connected to the EB network and one to the EFF. The PCIe40 is a custom board carrying an Altera Arria 10 Field Programmable Gate Array (FPGA) used for receiving and reformatting the data coming from the detector front-end. The board can be connected up to 48 optical links. Data are pushed by the PCIe40 FPGA into the main-memory of the EB PC.

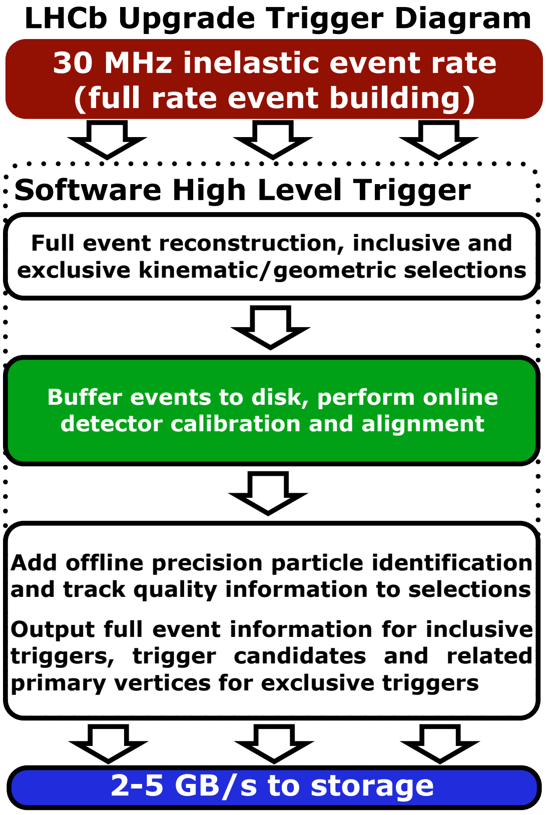
LHCb Upgrade Trigger Diagram
30 MHz inelastic event rate (full rate event building)
Software High Level Trigger
Full event reconstruction, inclusive and exclusive kinematic/geometric selections
Buffer events to disk, perform online detector calibration and alignment
Add offline precision particle identification and track quality information to selections
Output full event information for inclusive triggers, trigger candidates and related primary vertices for exclusive triggers
2-5 GB/s to storage

Figure 3.18: Representation of LHCb Upgrade 1a trigger flow and typycal event-accept rates for each stage.

Data from several bunch-crossings are grouped together into a multi-event fragment packet (MEP) to ensure efficient link-usage. A single node receives through the EB network all MEPs containing data from the same bunch crossing and builds the events merging all the packets together. Each EB node then sends the events to a sub-farm of the EFF, where the High Level Trigger will process them. Each node will send an event every $\sim 13\,\mu\text{s}$. Figure 3.20 shows the data-flow in the EB server.

### The Event Filter Farm

The EFF will be responsible for reducing the event-rate from the 30 MHz of colliding bunches to the accepted output rate of the storage. The baseline EFF will have 1000 servers running the HLT software. It is estimated that in 2020, using multicore CPU, a server will be able to run 400 HLT instances. Thus the maximum processing time allowed for each event in the EFF is 13 ms [59].

The trigger system will use track reconstruction algorithms similar to those currently used offline, but prioritized to reconstruct the most valuable tracks first, with more specialised track reconstruction algorithms only being used later in the

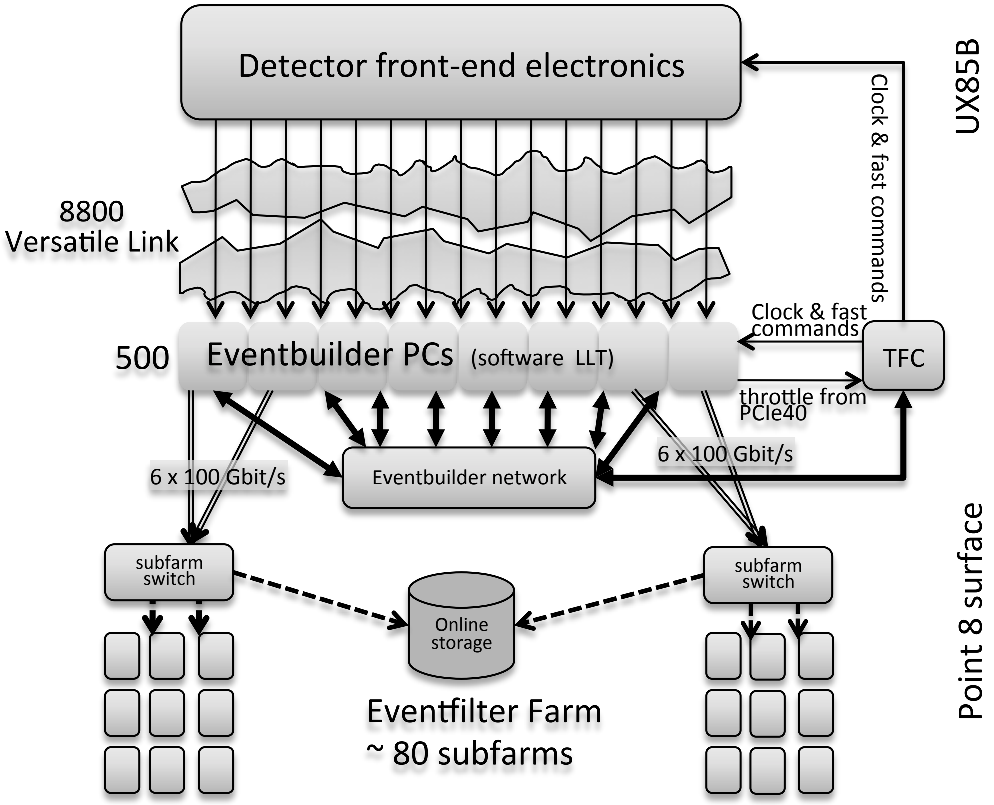

Figure 3.19: The architecture of LHCb Upgrade 1a readout-system.

process. Figure 3.21 shows a diagram of the track reconstruction sequence used in the trigger, as well as the main offline reconstruction sequence.

Track reconstruction in the trigger begins with execution of the full VELO tracking. Information from the UT sub-detector is then used to extend every VELO track which is consistent with a transverse momentum of at least 0.2 GeV/$c$. For the subset of tracks which were successfully extended, the charge and momentum is estimated. These tracks are then extended further by searching for hits consistent with $p_{\mathrm{T}} > 0.5\,\mathrm{GeV}/c$ in the SciFi sub-detector. The size of the search regions used to extend tracks in the SciFi are reduced by taking into account the charge and momentum measured in the UT.

Due to the limited storage bandwidth and size, only exclusive channels will be selected and recorded by the trigger system. Track not reconstructed by the HLT software will therefore not be reconstructable later. One notable example is the category of tracks named "downstream tracks", as they will require a significantly longer CPU time to reconstruct that what could be affordable in Run 3 (see Table 3.3). This is due to the lack of a starting seed in the VELO (see Figure 3.21 (right)). They will not therefore be generally available from Run 3 onward.

This thesis revolves around a project of making downstream tracks reconstructable and available to the HLT for triggering, even in the harsh conditions of rate and luminosity that LHCb will face in the future runs with a full readout of the detector at 40 MHz.

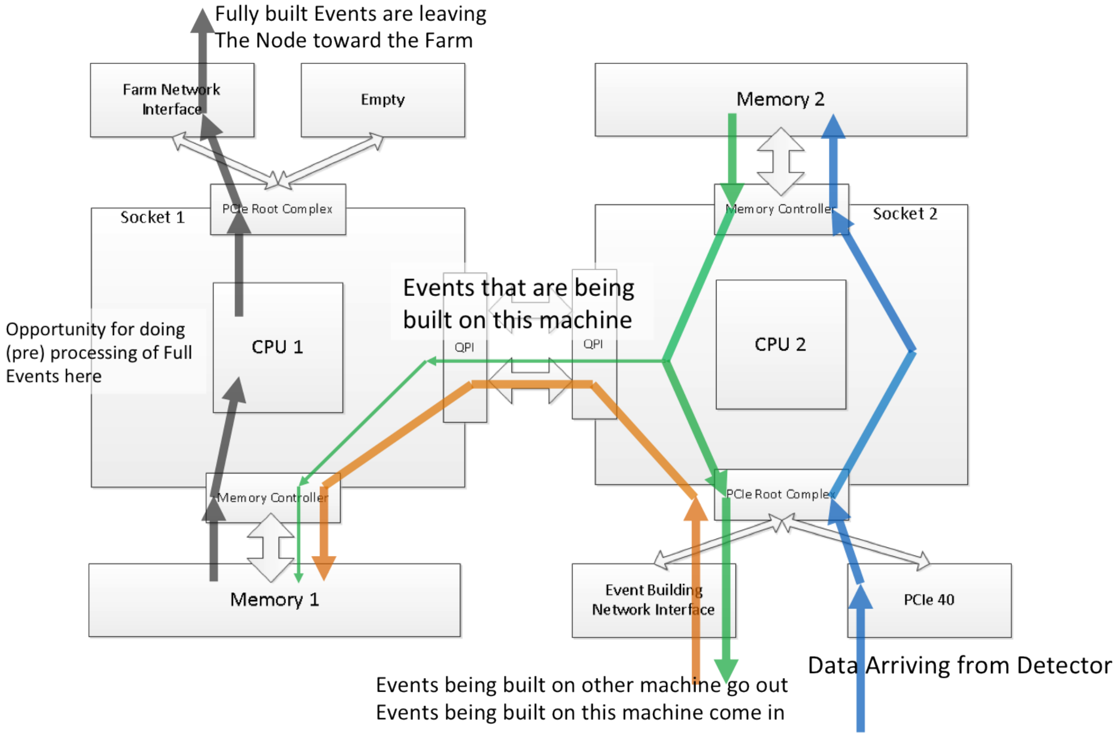

Figure 3.20: Data-flow in the Event-Builder server

| | time/event ( ms) | |
|---|---|---|
| | current | Upgrade 1a, Upgrade 1b |
| | $\nu = 2$ | $\nu = 7.6$ |
| T track reconstruction | 18 | 172 |
| matching | 8 | 100 |
| Downstream reconstruction<br>(T track + matching ) | $\sim 26$ | $\sim 272$ |

Table 3.3: Execution time of software downstream tracking [58, 60, 61]. The maximum processing time allowed for each event is 13 ms [59]

## 3.5 A "downstream tracker" for LHCb

The LHCb collaboration has recently published an Expression of Interest for a future upgrade program beyond the Run 3 [1]. This is motivated by the fact that many

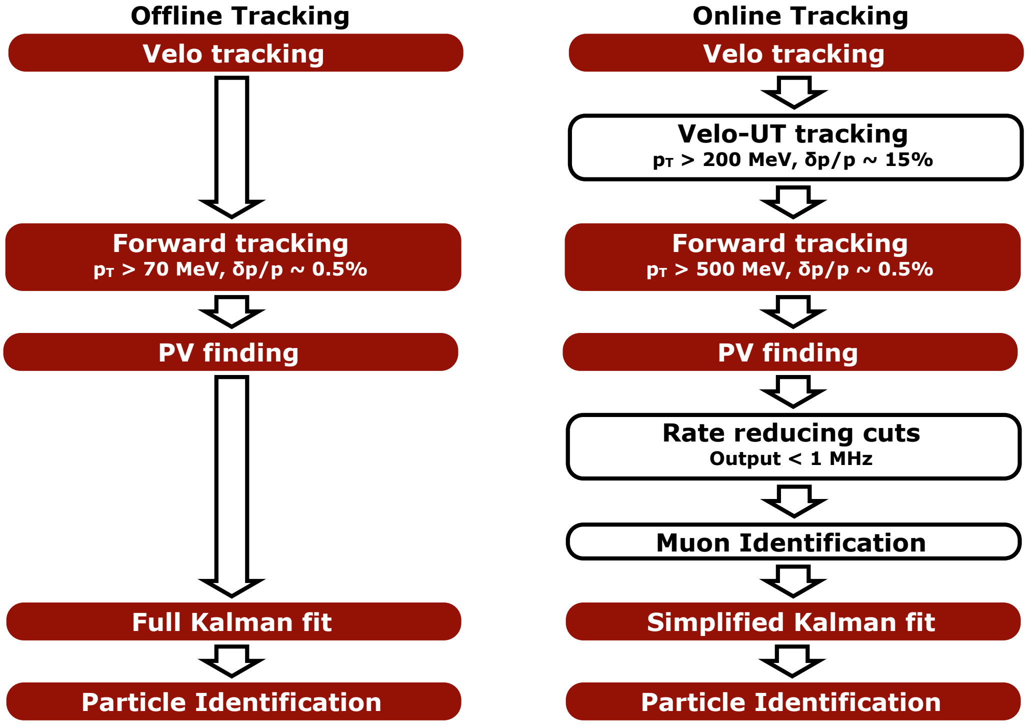

Figure 3.21: Track reconstruction sequences used (left) in the offline and (right) in the online trigger reconstruction. The offline reconstruction considers all VELO tracks for extension in the SciFi, whereas in the trigger information from the UT sub-detector is used to determine the charge and remove low $p_{\mathrm{T}}$ tracks before the Forward tracking. The use of the UT significantly reduces the execution time of the Forward tracking.

interesting physics measurements in its current program will still be limited by statistic by the end of the current program (end of Run 3), and, on the other end, continuation of data taking with the Run 3 detector will stop being attractive, on account of the excessive running time needed for a further significant increase of statistical precision. This becomes even less attractive starting from Run 5, when LHC will operate at even higher luminosities, approaching $10^{35}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$ (HL-LHC), thus making the LHCb Upgrade 1a detector using only a modest fraction of the available flow of data, that could in principle offer much greater physics possibilities, enabling many important observables to be measured with a precision unattainable at any other experiment, as concisely summarized in Table 3.4.

The plan set forth by the LHCb collaboration includes a first "consolidation" phase (Upgrade 1b) including modest improvements to the current scheme to be commissioned for a Run 4 at the same instantaneous luminosity of Run 3, followed by more extensive upgrades for a higher-luminosity phase ($\mathcal{L} = 2 \cdot 10^{34}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$) starting in Run 5, with the ultimate goal of collecting $300\,\mathrm{fb}^{-1}$.

A natural candidate for the consolidation phase is the realization of a specialized device capable of supplementing the Run 3 system with the capability of fully reconstructing all downstream tracks in every event, that would otherwise be lacking for the reasons explained in the previous sections. This is explicit discussed in the EoI [1] as an attractive addition to the existing system, because not having access to this information limits efficiency for decay modes with downstream tracks that cannot easily be triggered through another signature. An example is any channel containing a $K^0_\mathrm{S}$ meson and less than two prompt charged hadrons, like $B^0 \to \phi K^0_\mathrm{S}$, $B^0 \to J/\psi K^0_\mathrm{S}$, $D^0 \to K^0_\mathrm{S} K^0_\mathrm{S}$, $D^\pm_s \to K^0_\mathrm{S}\pi^\pm$, $K^0_\mathrm{S} \to \mu^+\mu^-$ *etc.* The same is true for decays involving $\Lambda$ baryons like $\Lambda^0_b \to \Lambda\mu^+\mu^-$, $\Lambda^0_b \to \Lambda\gamma$, or long-lived exotic particles. The study of these channels was already planned in the physic program of Upgrade 1a and Upgrade 2 as reported in Tables 3.2 and 3.4. The Downstream Tracker can increase the sensitivity for these channel by a factor from 2 to 10. Figure 3.22 shows the invariant mass distributions of $K^0_\mathrm{S}\pi^+\pi^-$ candidate events in LHCb, data corresponds to an integrated luminosity of $1\,\mathrm{fb}^{-1}$. Yields obtained from the fit are $845 \pm 28$ downstream decay $B^0 \to K^0_\mathrm{S}\pi^+\pi^-$, and $360 \pm 21$ long decay [62].

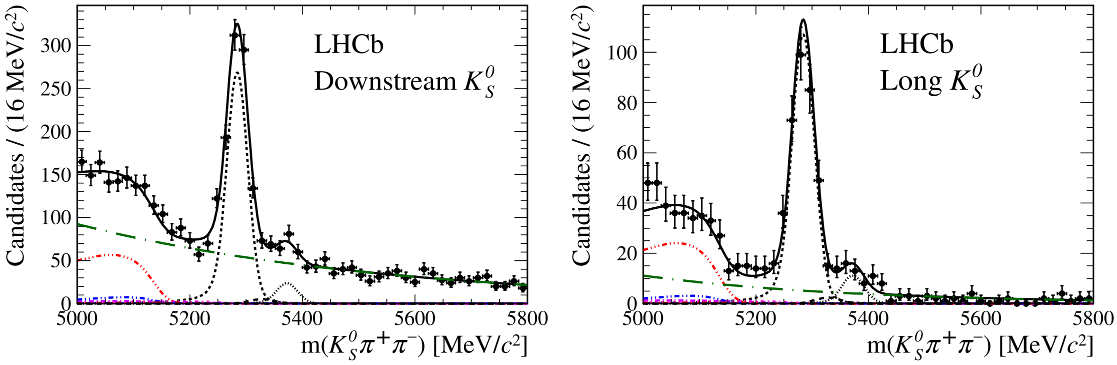

Figure 3.22: Invariant mass distributions of $K^0_\mathrm{S}\pi^+\pi^-$ candidate events in LHCb [62].

In order to be useful, a Downstream Tracker needs to provides tracks to the software trigger in parallel with all the rest of raw detector information in the event [63]. Also, it needs to do it with a fraction of the huge size, cost and power consumption that would otherwise be needed by an implementation based on conventional CPU technology, and must be able to seamlessly work within the existing Run 3 Data acquisition (DAQ) that has been described in previous sections. This is clearly a difficult task, but a worthwhile one, as it is a great opportunity not only to solve a specific well-defined physics need, but also to develop and test new and more advanced technologies of fast data reconstruction, that, as outlined in the previous chapter, will be more and more needed in future experiments at high intensities.

The rest of this thesis describes the work that I have contributed towards this ambitious goal; starting from testing the potential of a new parallel processing architecture named “Artificial Retina” (Chapter 4), verifying the achievable tracking and timing performance and amount of needed resources (Chapter 6), up to performing a first study of its actual utilization in reconstructing interesting $K_S$ modes, using an emulator interfaced with the full LHCb official simulation (Chapter 8).

| Topics and observables | Experimental reach | Remarks |
|---|---|---|
| **EW Penguins** | | |
| Global tests in many $b \to s\mu^+\mu^-$ modes with full set of precision observables; lepton universality tests; $b \to dl^+l^-$ studies | *e.g.* 440k $B^0 \to K^*\mu^+\mu^-$ & 70k $\mathit{\Lambda}_b^0 \to \mathit{\Lambda}\mu^+\mu^-$; Upgrade 2 $b \to d\mu^+\mu^- \approx$ Run-1 $b \to s\mu^+\mu^-$ sensitivity. | Upgrade 2 ECAL required for lepton universality tests. |
| **Photon polarisation** | | |
| $\mathcal{A}^{\Delta}$ in $B_s^0 \to \phi\gamma$; $B^0 \to K^*e^+e^-$; baryonic modes | Uncertainty on $\mathcal{A}^{\Delta} \approx 0.02$; $\sim 10k$ $\mathit{\Lambda}_b^0 \to \Lambda\gamma$, $\Xi_b \to \Xi\gamma$, $\Omega_b^- \to \Omega\gamma$ | Strongly dependent on performance of ECAL. |
| $b \to cl^-\bar{\nu}_l$ **lepton-universality tests** | | |
| Polarisation studies with $B \to D^{(*)}\tau^-\bar{\nu}_\tau$; $\tau^-/\mu^-$ ratios with $B_s^0$, $\mathit{\Lambda}_b^0$ and $B_c^+$ modes | *e.g.* 8M $B \to D^*\tau^-\bar{\nu}_\tau$, $\tau^- \to \mu^-\bar{\nu}_\mu\nu_\tau$ & $\sim 100k$ $\tau^- \to \pi^-\pi^+\pi^-(\pi^0)\nu_\tau$ | Additional sensitivity expected from low-$p$ tracking. |
| $B_s^0, B^0 \to \mu^+\mu^-$ | | |
| $R \equiv \mathcal{B}(B^0 \to \mu^+\mu^-)/\mathcal{B}(B_s^0 \to \mu^+\mu^-)$; | Uncertainty on $R \approx 20\%$ | |
| $\tau_{B_s^0 \to \mu^+\mu^-}$; $CP$ asymmetry | Uncertainty on $\tau_{B_s^0 \to \mu^+\mu^-} \approx 0.03\,\mathrm{ps}$ | |
| **LFV $\tau$ decays** | | |
| $\tau^- \to \mu^+\mu^-\mu^-$, $\tau^- \to h^+\mu^-\mu^-$, $\tau^- \to \phi\mu^-$ | Sensitive to $\tau^- \to \mu^+\mu^-\mu^-$ at $10^{-9}$ | Upgrade 2 ECAL valuable for background suppression. |
| **CKM tests** | | |
| $\gamma$ with $B^- \to DK^-$, $B_s^0 \to D_s^+K^-$ *etc.* | Uncertainty on $\gamma \approx 0.4°$ | Additional sensitivity expected in $CP$ observables from Upgrade 2 ECAL and low-$p$ tracking. |
| $\phi_s$ with $B_s^0 \to J/\psi K^+K^-$, $J/\psi\pi^+\pi^-$ | Uncertainty on $\phi_s \approx 3\,\mathrm{mrad}$ | |
| $\phi_s^{s\bar{s}s}$ with $B_s^0 \to \phi\phi$ | Uncertainty on $\phi_s^{s\bar{s}s} \approx 8\,\mathrm{mrad}$ | |
| $\Delta\Gamma_d/\Gamma_d$ | Uncertainty on $\Delta\Gamma_d/\Gamma_d \sim 10^{-3}$ | Approach SM value. |
| Semileptonic asymmetries $a_{\mathrm{sl}}^{d,s}$ | Uncertainties on $a_{\mathrm{sl}}^{d,s} \sim 10^{-4}$ | Approach SM value for $a_{\mathrm{sl}}^{d}$. |
| $V_{ub}/V_{cb}$ with $\Lambda_b^0$, $B_s^0$ and $B_c^+$ modes | *e.g.* $120k$ $B_c^+ \to D^0\mu^-\bar{\nu}_\mu$ | Significant gains achievable from thinning or removing RF-foil. |
| **Charm** | | |
| $CP$-violation studies with $D^0 \to h^+h^-$, $D^0 \to K_{\mathrm{S}}^0\pi^+\pi^-$ and $D^0 \to K^{\mp}\pi^{\pm}\pi^+\pi^-$ | *e.g.* $4 \times 10^9$ $D^0 \to K^+K^-$; Uncertainty on $A_\Gamma \sim 10^{-5}$ | Access $CP$ violation at SM values. |
| **Strange** | | |
| Rare decay searches | Sensitive to $K_{\mathrm{S}}^0 \to \mu^+\mu^-$ at $10^{-12}$ | Additional sensitivity possible with downstream trigger enhancements. |

Table 3.4: Summary of prospects for Upgrade 2 measurements of selected flavour observables [1].

# Chapter 4

# “Artificial Retina” approach to real time tracking

## 4.1 The “Artificial Retina” approach

The “Artificial Retina” architecture was proposed in 2000 [2] as a fast parallel track reconstruction system applicable to HEP experiments inspired by the mechanisms of vision in the natural brain. The mathematical aspects of the algorithm have some similarities with the “Hough transform” [64,65], a method already applied for finding lines in image processing. However, the crucial feature of the “artificial retina” is the design of a layout and an implementation with the potential of sustaining the event rate at HL-LHC experiments. Thanks to the exploitation of some structural ideas extracted from the current knowledge of the visual system of mammals, that allows them to recognize specific “patterns” in the incoming data with throughput and latency performances vastly superior to what has been achieved in artificial systems to date.

Compared to previous successful real-time tracking systems based on patterns stored in databases (like Associative Memories-based systems) developed for HEP experiments, one of the “Artificial Retina” distinctive element is the way to compare the stored patterns with the incoming detector informations. While other systems provide a binary response (“yes” or “no”) from the comparison with stored patterns, the “Artificial Retina” returns a response that continuously varies depending on the “distance” of the track from the patterns imitating the continuous neuron response to exciting stimuli. Interpolating the comparison responses from different patterns allows to obtain higher tracking performances with a reduced number of stored patterns. Another important feature suggested by the structure of natural neural system is a fully data-flow organization with a very high degree of parallelization, with careful avoidance of any sequential steps and wait states. This brings together a further feature known to exist in the natural vision, that is a peculiar organization of the overall system bandwidth: in traditional trigger systems the bandwidth is progressively reduced during processing, while in the “Artificial Retina” the bandwidth increases significantly, because multiple copies of the same data are allowed to be produced, shrinking down only at a later stage as shown in Figure 4.1. This approach has only recently become technically feasible due to the progress of telecommunication technology. A selective data distribution reduces the amount of required bandwidth for single device to a rate feasible in current devices.

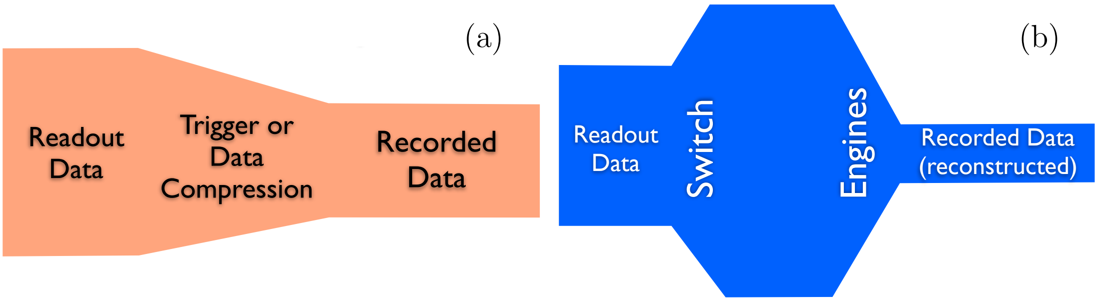

Figure 4.1: Bandwidth flow in a generic trigger system (a) and in the "Artificial Retina" architecture (b).

### 4.1.1 Basic Concepts

To describe the "Artificial Retina" architecture working principle, we consider the simplified case of straight tracks intersecting a few parallel planar detector layers. Given the coordinates of the hits, we want to estimate the parameters off all the tracks that generated them. If we consider only one transverse view, a track can be described by two parameters only. The tracks parameters are the coordinate of intersection of the track with the first and the last layer of the detector, we call them respectively $U$ and $V$. We divide the two-dimensional phase space in a grid consisting of cells, and label each cell with a pair of parameters $(U_i, V_j)$. Each cell corresponds to a mapped track. The coordinates of the intersection of a mapped track with the detector layers are $t_l(U_i, V_j)$ where $l$ is the layer number, that we call receptors.

For each event we compute the excitation level defined as:

$$R_{ij} = \sum_{lr} \exp\left(\frac{-d(x_r^{(l)}, t_l(U_i, V_j)^2}{2\sigma^2}\right)$$

where $d(x_r^{(l)}, t_l(U_i, V_j)$ is the Euclidean distance between the hit $x_r$ on layer $l$ and the receptor $t_l(U_i, V_j)$. The sum is extended to all hits present in all the layers and computed for all the cell. The parameter $\sigma$ can be adjusted to optimize the sharpness of the response of the receptor.

Pattern recognition can be reduced to finding cluster in this cell array. Information on the parameters of the tracks can be obtained from the position of each cluster in phase space. Since the response of each receptor is a smooth function of the coordinate of the hits, the excitation level can be used as a weight and the position of the cluster of the center of the cluster can be obtained by interpolation. In this way, the precision on track parameters that can be achieved is typically much better than the pitch of the grid. Plus, the needed computations can be performed in parallel over the array.

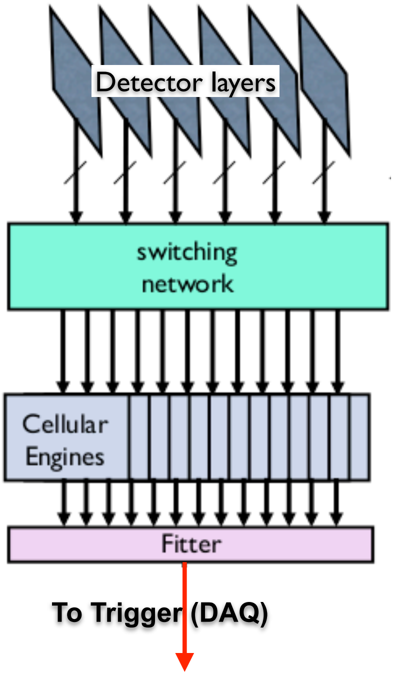

Figure 4.2: Integration of the "Artificial Retina" architecture in the DAQ system of a real experiment.

Figure 4.2 shows the block diagram of a generic "artificial retina" system. Each cell is implemented as an independent block of logic (Engine) that performs autonomously all the necessary operations. Hits flow from the detector into a custom switching network, that delivers each hit to all relevant Engines in parallel duplicating them as necessary. Local maxima are found in parallel in all Engines, with some limited exchange of information between neighbor Engines. The coordinates and excitation level of the local maxima, and the excitation level of their nearest neighbors are outputted sequentially. A final parallel linearized fitter stage extracts track parameters from the cluster informations. The reconstructed tracks are then made available to the trigger/DAQ system. The whole system works as a short asynchronous pipeline accepting an uninterrupted flow of events, that when implemented in modern programmable digital devices endowed with large internal bandwidths, can operate with very high throughputs and low latencies. This opens the possibility of operating the device transparently, so that it effectively appear to the rest of the DAQ as if tracks are coming out of the detector directly, making it particularly suitable to high-rate, high volume applications.

### 4.1.2 Application to LHCb Downstream Tracker

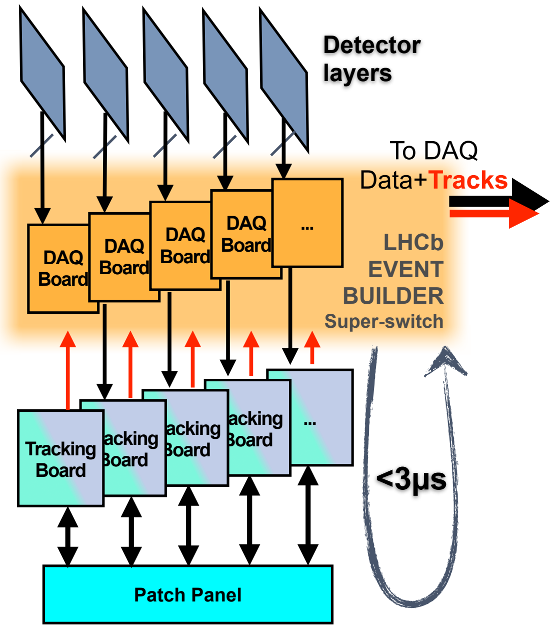

Figure 4.3: Integration in LHCb DAQ.

The Downstream Tracker project, as was explained in Section 3.5, must be integrated inside the EB of LHCb Upgrade 1b. As explained in Section 3.4.4 the EB is a cluster implemented as 500 PC. Gathering data from a such large number of nodes require an equally large number of devices to perform the switching function. Since the switching network only requires a small amount of the logic, using these devices only for this task is a waste of resources. Figure 4.3 shows a modified design conceived for the specific purpose of integrating the Downstream Tracker in the EB. In the new design a large number of individual Tracking Boards are aggregated to form the Downstream Tracker. The Tracking Boards carry out the function of the switching network and the Engines together. Every board is connected to a EB node from which it reads the hit. Each portion of the switching network delivers hits to all relevant Engines. A mesh network labeled Patch Panel in the figure allows to exchange hits between different Tracking Boards. Each Tracking Board returns a subsample of the reconstructed tracks to the EB node it is connected to. Those tracks are added to the raw data collected by that node, and from this point on the Event Building proceeds normally. As a results, the EFF will receive "raw data" that additionally contain fully reconstructed tracks, that appear as if having been produced by an additional virtual detector. This solution allows to use all the resource of the devices and the full-duplex capabilities of the inter-devices connection.

For not modify the EB specification the latency of the Downstream Tracker must be somewhat smaller than the latency of the EB ($\sim 13\,\mu$s). With the data currently at hand, a sensible objective seems to be a latency $< 3\,\mu$s.

We can implement this design in different ways. One option is to add a PCIe board with a large FPGA in a empty slot of the EB nodes. The Tracking Board could read the data from the PCIe40 through the PCIe bus. The reconstructed tracks can be returned to the EB through the same PCIe bus. This option uses the existing hardware of the EB adding only the Tracking Board and the Patch Panel resulting in a cheap system, but it requires that the EB nodes have a PCIe slot available. Due to the low number of PCIe line in actual CPU this option might turn out to be unfeasible. The DAQ system don't use all the optical links of the PCIe40 board. 24 optical links are used at half speed for data. This board can send data to Downstream Tracker through the 12 free links at full speed. The Downstream Tracker can thus work as a standalone system. It can return the reconstructed tracks to the EB through some dedicated additional EB nodes. This option is a more flexible solution but it requires more hardware to read data through the optical links, plus some modification of the PCIe40 firmware. At the time of this writing, the precise details of the EB implementation that we would need to know to make a definite decision are still subject to change, therefore we will keep both possibilities open for the time being. They have anyway negligible impact on the rest of the discussion in this thesis.

## 4.2 Implementation details

The architecture described in previous sections is flexible and largely scalable. Without significant loss of generality, we will in the following make reference straight tracks intersecting a few parallel detector layers. The tracks parameters that we use are the coordinate of intersection of the track with the first and the last layer of the detector, we call them respectively $U$ and $V$.

### 4.2.1 Mapping algorithm

For configuring the "Artificial Retina", the phase space of track parameters is divided into cells, which mimic the neurons connected to the receptive fields of the retina. The center of each cell corresponds to a specific track in the detector that intersects the layers in spatial points called receptors. A C++ piece of software code (Detector-Mapping) calculates the receptors for each cell, as shown in Figure 4.4. Not all hits are significant for each cell. A second step of the Detector-Mapping, also shown in Figure 4.4, groups contiguous cells. For a group of contiguous cells, where variations of track parameters are small, the corresponding receptors in the detector layers would belong to a limited area. The Detector-Mapping calculates which groups of cells are influenced by the hits recorded in a detector area. A hit influences a cell if its distance from the receptor is lower than the distance search. The distance search is a parameter of the "Artificial Retina", a typical value of the distance search is the pitch of cells grid.

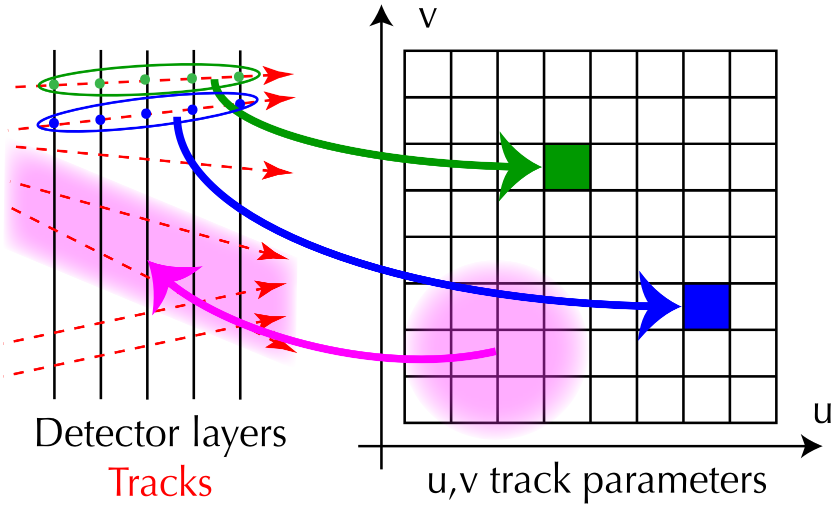

Figure 4.4: "Artificial Retina" mappings for tracks on a plane without magnetic field, where tracks can be described by two parameters $U$ and $V$. Magenta cloud shows a group of cell influenced by a specific detector area.

### 4.2.2 The switching network

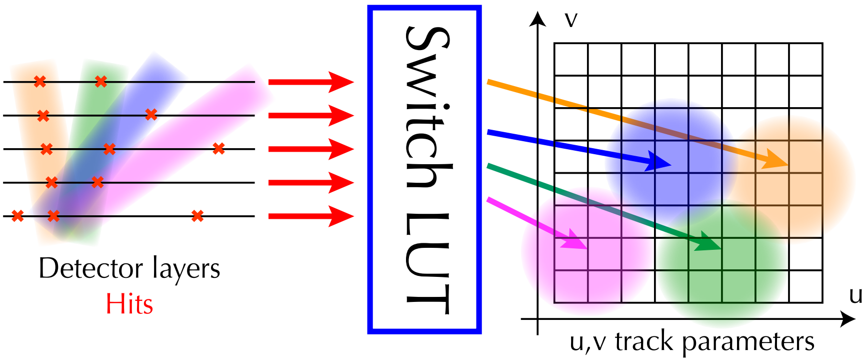

Figure 4.5: The switching network send hits to the influenced cell groups. Clouds shows cell group influenced by different detector areas.

To realize the "Artificial Retina" in practice, a crucial ingredient is a system for distributing in real time the hit informations coming from the detector layers to the array of cell. Given the high bandwidth of several Tbits/s, this is a nontrivial task.

The switching network is a intelligent delivery system, with embedded information allowing each hit to be delivered in parallel to all cell. The switching network use the information calculated in the mapping step to deliver each hit only to the influenced cell groups. The switching network can send the same hit to more cell groups, as shown in Figure 4.5. As shows in Figure 4.1, the network increase the global bandwidth making multiple copies of hits ($10 - 12$ copies), but it never sends a hit to all the engines, reducing the input bandwidth for them.

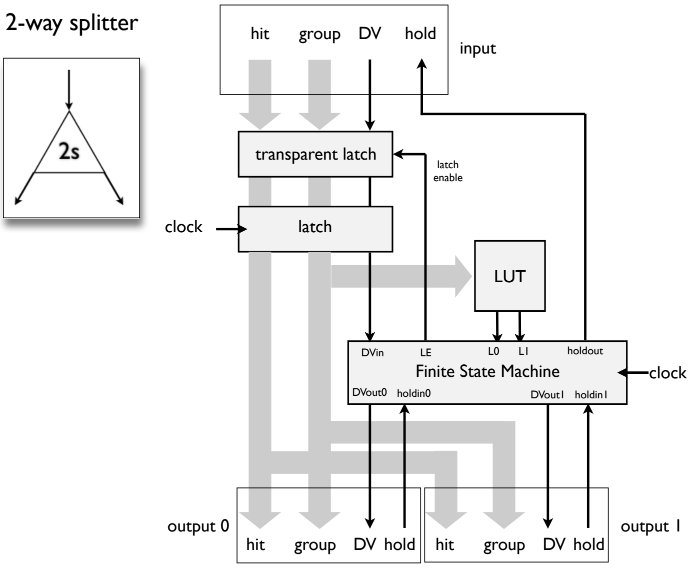

Figure 4.6: Splitter design developed in 2014. A finite state machine regulates the behavior of the splitter.

The basic components of the switching network are the splitter and the merger. The splitter (Figure 4.6) is a component with one input and two output. It searches the input data inside a lookup table (LUT). The LUT reports to which output the splitter must sends the data. The merger (Figure 4.7) is a component with two input and one output. It merges the two input channel in one.

The switching network can be assembled using basic 2-way dispatchers (2d) with two inputs and two outputs: data on any input can be redirected to any output using two splitters and two mergers, as shown in Figure 4.8.

The dispatchers work as a pipeline and the latency is proportional to n. This type of switching network can be easily scaled adding enough inputs to receive data from the readout modules and outputs to transfer data to all the devices used for the cellular engines. To implement a dispatcher with $2^n$ inputs/outputs, we need N 2-way dispatchers connected together, where

$$N(n) = 2N(n-1) + 2^{n-1}.$$

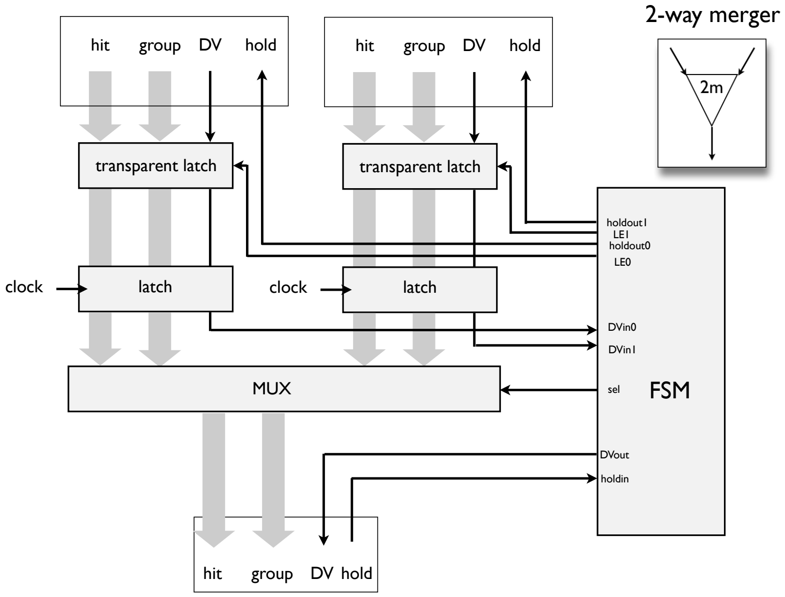

Figure 4.7: Merger design developed in 2014. A finite state machine regulates the behavior of the merger.

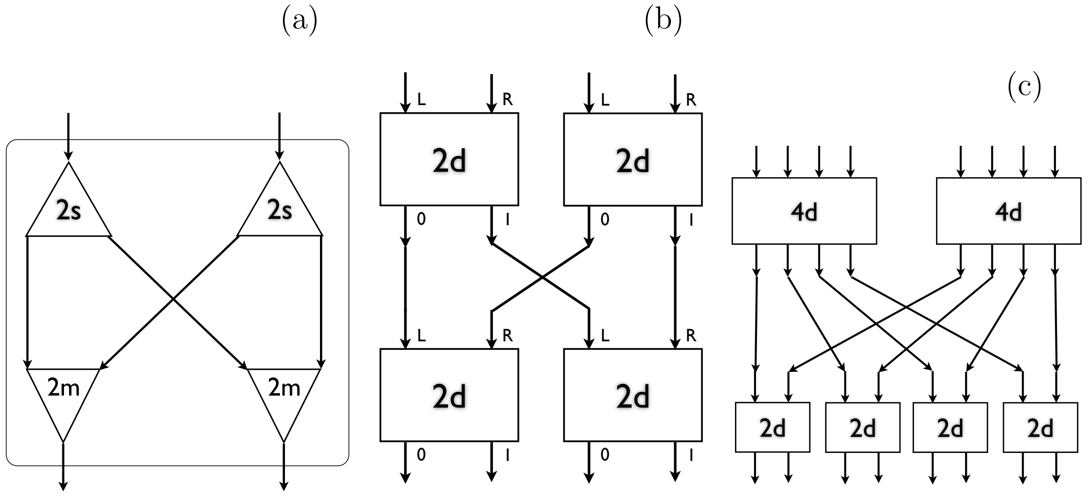

Figure 4.8: Schemes of dispatchers with 2(a), 4(b), and 8(c) inputs/outputs.

### 4.2.3 The processing Engine

The Engine is the hits processor of a cell. Figure 4.9 shows the engine functions. For each hit $x_l$ on layer $l$ the Engine calculates the Euclidean distance $d_l(x_l, t_l)$ from the cell receptor $t_l$ of that layer. Then calculates the weight $w$ of each hit. The weight is defines as:

$$h(x_i) = \begin{cases} 0 & \text{if } d_s < d_l(x_l, t_l) \\ \exp\left(\frac{-d_l(x_l,t_l)^2}{2\sigma^2}\right) & \text{if } d_l(x_l, t_l) < d_s \end{cases}$$

where $d_s$, the distance search, is a cutoff of the weight function, and $\sigma$ controls the width of the weight function. $d_s$ and $\sigma$ are parameters of the "Artificial Retina" architecture, that can be adjusted to optimize the sharpness of the response of the receptors, and to reduce the input bandwidth for the cells. Typical value of $d_s$ is the pitch of cells grid, and typical value of t$\sigma$ is half pitch of cells grid. The engines can process a hit for each clock cycle, and all the engines work in a fully parallel way. The Engine excitation level is the summation of the weight of each hit. When all the hits of a same event are processed, the Engines begin to identify the tracks.

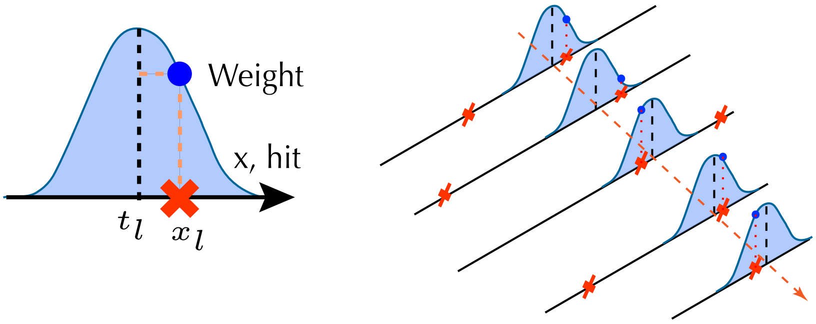

Figure 4.9: Engine calculates the weighted distance of the hits from its receptors.

Figure 4.10 shows the state of the art for the Engine [66]. The engine can elaborate a hit at each clock cycle in work in a fully pipelined way. The Engine reads the receptor from a LUT, and calculates the difference between the receptor and the coordinate of the hit. Dividing the difference by the $\sigma$ and computing the exponential are functions that require a large amount of logic and clock cycle. The Engine avoid these task searching in a LUT the final value of these operations.

Layers can be grouped in stations made of a doublet of layers (Section 5.3.1). Thus, Engine can have partial accumulators, one for every station; this solution allows to apply a threshold on each doublet's accumulator and to improve the ghost rate of the system. The last hit of a event is called End Event (EE), it indicate the end of the event end not carry hit coordinate. When a EE arrives, the Engine sum the doublet accumulators. If the sum is over a threshold, it is sent to the clustering module. Then the Engine processes a new event. The total latency of the Engine is 10 clock cycle [66].

Since a single device can't contain all the Engine, the Engines matrix must be distributed on more devices. However the clustering operations requires that every Engine communicate with the eight neighbor Engines. The Latency and bandwidth

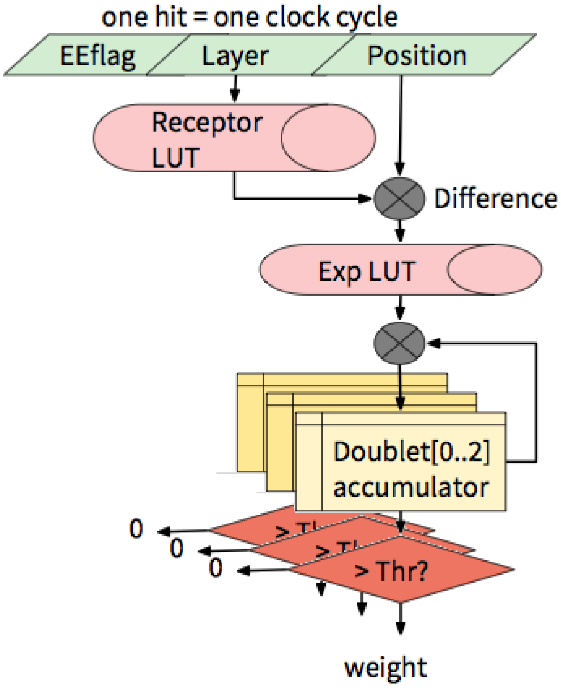

Figure 4.10: State of the art for the Engine.

of inter-devices connections are unsuited. The solution adopted is to surround the Engine matrix with a set of border Engine. A border Engine can calculate its excitation level but can't be a maximum. The cell associated to a border Engine is also associated to a regular Engine in an another devices.

A single logic component carry out the function of the Engine and clustering. Whatever the Engine can process a second event just after the first, without wait that the clustering has processed the first event.

### 4.2.4 Clustering

Tracks can be identified by looking for local maxima of the excitation level over the cells grid, shown in Figure 4.11. The Engines read the excitation level of the neighbor Engines inside a square $3 \times 3$. If the excitation level of the neighbor Engines is lower than the Engine excitation level, the Engine is flagged as local maximum. We can set a threshold level for avoid false positive maxima. For a track resolution similar to offline reconstruction the grid does not require a high granularity, because significantly better precision can be obtained by computing the centroid of the $3 \times 3$ excitation level cluster. Given the excitation level $R_{kl}$ of the Engines, the track parameters $\overline{u}$ and $\overline{v}$ can be calculated as:

$$\overline{u} = u_0 + \delta u \frac{\sum_{kl} k R_{kl}}{\sum_{kl} R_{kl}}$$

$$\overline{v} = v_0 + \delta v \frac{\sum_{kl} l R_{kl}}{\sum_{kl} R_{kl}}$$

with $k = i-1, i, i+1$ and $l = j-1, j, j+1$, where $u_0$ and $v_0$ are the track parameters of the cells grid origin , $\delta u$ and $\delta v$ are the pitch of the cells grid, $i$ and $j$ are the index of the local maximum Engine.

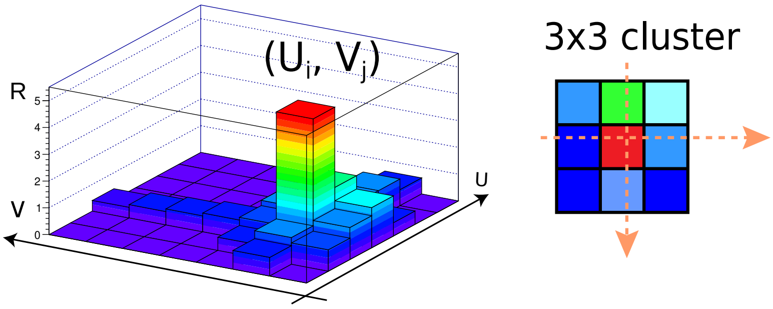

Figure 4.11: Cells excitation levels with a single maximum, and the $3 \times 3$ cluster around the maximum.

## 4.3 Goals of my work

The first goal of my work was to demonstrate that a tracking system based on "Artificial Retina" architecture had the potential to reconstruct the downstream tracks with a speed sufficient to operate at Level-1 of the LHCb, like proposed for the LHCb Downstream Tracker discussed in section 3.5.

To do this I had to produce a new implementation of the system prototype on current, much faster and bigger FPGAs and to optimize the design for achieve an input rate of 40 MHz. This is the subject of the next chapter.

# Chapter 5

# Hardware prototyping

## 5.1 Implementation Requirements

The "Artificial Retina" approach is conceived to be implemented in custom digital electronic circuit, as described in [2], due to the high parallelized architecture. Furthermore it requires an high bandwidth, as explained in 4.1. This can be seen more precisely by considering a quantitative example scenario Table 5.1. The total bandwidth is order of Tbits/s. The only way to achieve this bandwidth is via a large, custom built electronic system. Development costs and effort make application specific integrated circuits (ASICs) not an attractive choice for developing a prototype. Then we chose to use programmable logic devices, in particular FPGAs. A modern commercially available FPGA can reach an I/O bandwidth of several Tbits/s [67–70], through modern high-speed serial links (SerDes), actually exceeding the needs of most applications. Moreover, FPGA performances are still increasing at a steady pace, taking advantage of new silicon technology ($14 - 16$ nm), and increasing the number of logic elements (up to several millions). It is possible to implement our system on multiple board connected together trough optical fiber cables. If some further performances boost is needed (less than 100%), the firmware can even be semi-automatically transferred on ASIC devices for mass production at an additional cost.

| | |
|---|---|
| crossing frequency | 40 MHz |
| number of layers | 6 |
| number of hits per layer per crossing | 50 |
| number of bits per hits bandwidth | 15 bits |
| total hit bandwidth | 180 Gbits/s |
| maximum copies per hits | 12 |
| max total bandwidth | 2.2 Tbits/s |

Table 5.1: Bandwidth required in a possible scenario.

## 5.2 The Field Programmable Gate Array

Until the '70s, the developing of digital system was based on interconnected integrated circuit, at low-scale and medium-scale integration (1-100 logic gate per integrated circuit). More complex system required an higher number of components, and that led to expensive, low efficiency, and unreliable devices.

One solution was to design custom integrated circuit for a specific application, called ASIC. Another solution is to use a programmable logic devices (PLDs), that can implement any digital-logic function based on firmware-like information stored in non-volatile memory. This approach allows a very high flexibility in development phase, reducing cost and time of development, although the performance of the device are lower than ASIC.

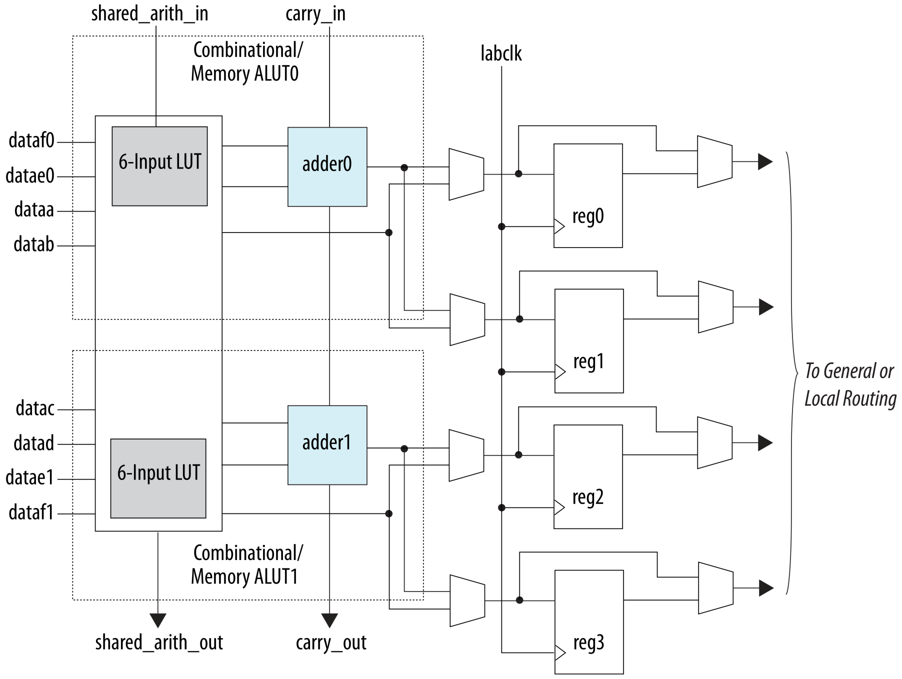

Figure 5.1: ALM High-Level Block Diagram for Stratix V Devices.

The most advanced PLDs are FPGAs. They are often used nowadays in building powerful electronic systems that need to be produced only in limited quantities and be easily reconfigurable - some examples are radars, medical CT scanners, advanced navigation and communication systems. The FPGAs contain an array of programmable logic blocks, and a hierarchy of reconfigurable interconnects that allow the blocks to be "wired together", like many logic gates that can be inter-wired in different configurations. In most FPGAs, logic blocks also include memory elements. Even if the basic concepts are the same, manufacturers use different name and organization for the internal components of the FPGAs. For this thesis I worked with Altera FPGAs, in particular the Stratix V series [71]. This choice is favored by the diffusion of these devices in the LHCb experiment. Their internal structure is described in detail below.

In Altera Stratix V the programmable logic blocks are called adaptive logic modules (ALMs). These modules can be configured to implement logic functions, arithmetic functions, and register functions. The basic components of an ALM are the LUT, full adders, D-type flip-flops, and multiplexers. The ALM for a Stratix V FPGA is shown in Figure 5.1. Ten ALMs are grouped together to form a logic array block (LAB). Each LAB contains also dedicated logic for driving control signals to its ALMs, a fast-local interconnect for ALMs in the same LAB, and a "direct-link" interconnect with the neighbor LABs, memory blocks, and digital signal processors (DSPs). The "direct link" connection feature minimizes the use of global interconnects, providing higher performance and flexibility. Figure 5.2 shows the internal structure of two LABs and the interconnection between them. Stratix V, as many modern FPGAs, also contain hard intellectual property (IP), devices build alongside the programmable fabric to include common high level functionalities. Having these common functions embedded into the silicon allows to save space when using them and gives those functions increased speed compared to building them from primitives using the ALMs. Examples of these IP include multipliers, generic DSP blocks, high speed serializer for I/O, and sometimes even complete embedded processors.

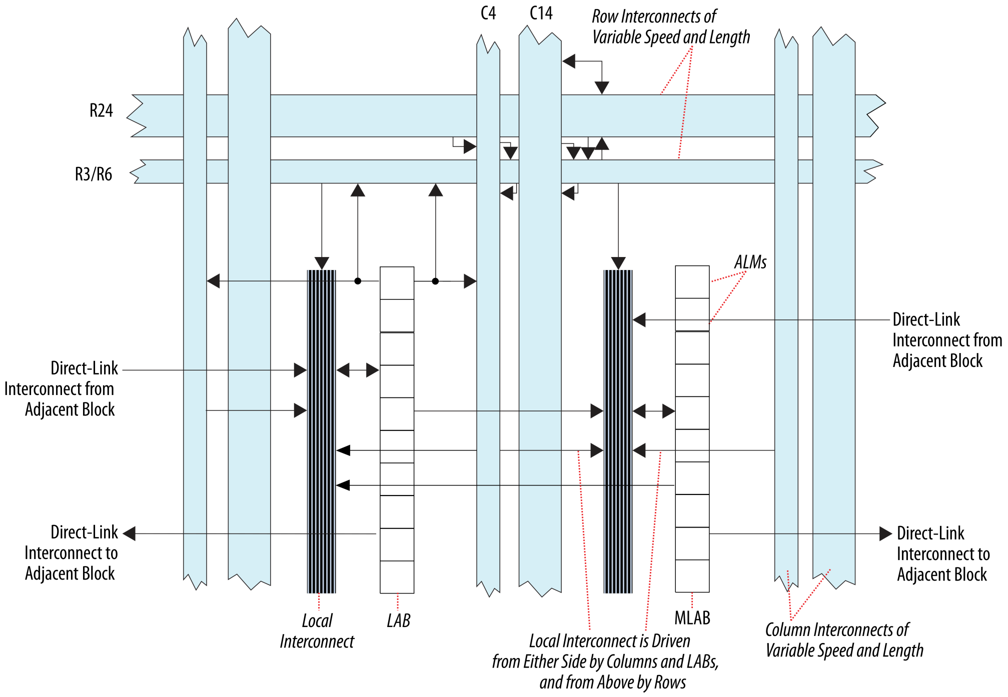
C4
C14
Row Interconnects of Variable Speed and Length
R24
R3/R6
ALMs
Direct-Link Interconnect from Adjacent Block
Direct-Link Interconnect from Adjacent Block
Direct-Link Interconnect to Adjacent Block
Direct-Link Interconnect to Adjacent Block
Local Interconnect
LAB
MLAB
Local Interconnect is Driven from Either Side by Columns and LABs, and from Above by Rows
Column Interconnects of Variable Speed and Length

Figure 5.2: Internal structure of two LABs and the interconnection between them.

## 5.3 State of the art

In 2015 part of LHCb Group in Pisa started the “RETINA Project”. This is a 3-years initiative supported by INFN-CNS5 and devoted to R&D for a track processor based on “Artificial Retina” architecture. At the beginning of my work a non-optimized functional prototype existed, showing that the logic functionality of the architecture worked, but with no requirements on speed.

### 5.3.1 The functional prototype

The functional prototype [72] was implemented using pre-existing boards with old-generation FPGAs developed for the DAQ of the NA62 experiment [73]. It was programmed to reconstruct tracks in the LHCb IT (Section 3.3.4) using all six $x$ layers. Because the layers are grouped in three station, we refer to the layers of a station as doublet. Figure 5.3 show the detector configuration, $U$ and $V$ are the parameters of the phase space, they represent respectively the hits on the first and last layer. A grid of 3,360 cells cover the track parameters space of a lateral IT subunit. Engine are distributed as 16 independent $16 \times 15$ matrix (border Engine included). Eight boards compose the prototype. The description of the boards is in the below section. Pair of boards are connected as shown in figure 5.4 to adapt them to the “Artificial Retina” architecture. One board was used for the switching network and one for the Engines.

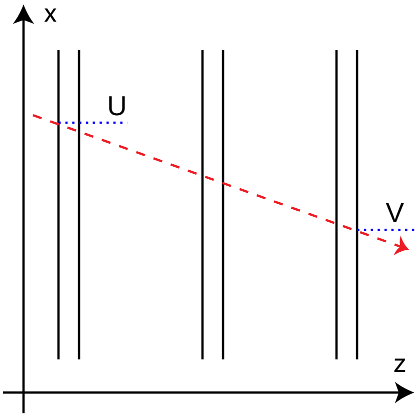

Figure 5.3: Detector configuration used for the functional prototype.

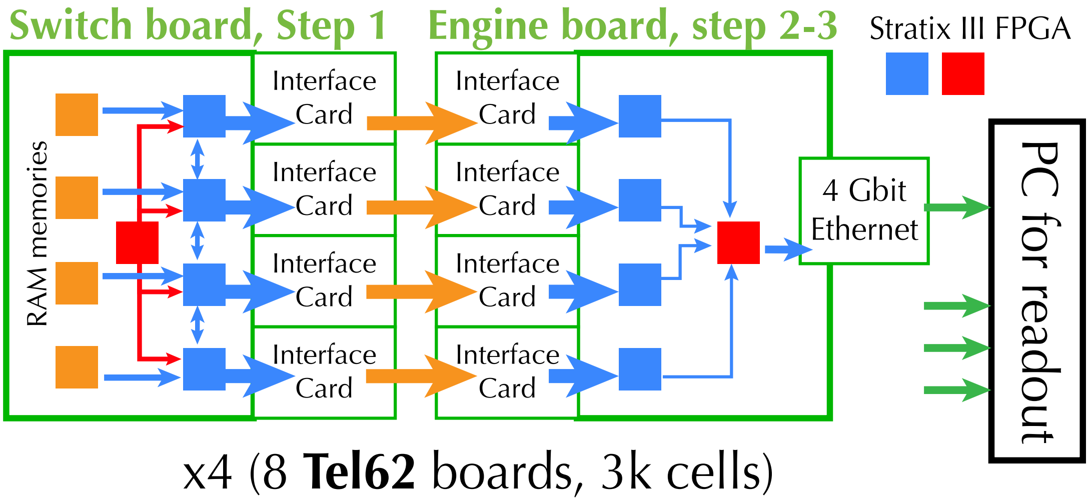

Figure 5.4: Diagram of the functional prototype made using TEL62 board.

Though high speed is not required for the functional prototype, the performances of the system was still measured. The maximum clock frequency of the functional prototype is 40 MHz for the switching network and 160 MHz for the Engine. If a component of the functional prototype is full and incapable of receiving any more hit, a back pressure mechanism halts the sending of hits to this component until it's capable of receiving more hits. The back pressure regulates the effective event rate of the system. Figure 5.5 shows the event rate achieved by the functional prototype at different tracks occupancy. The latency of the system is $< 2\,\mu s$.

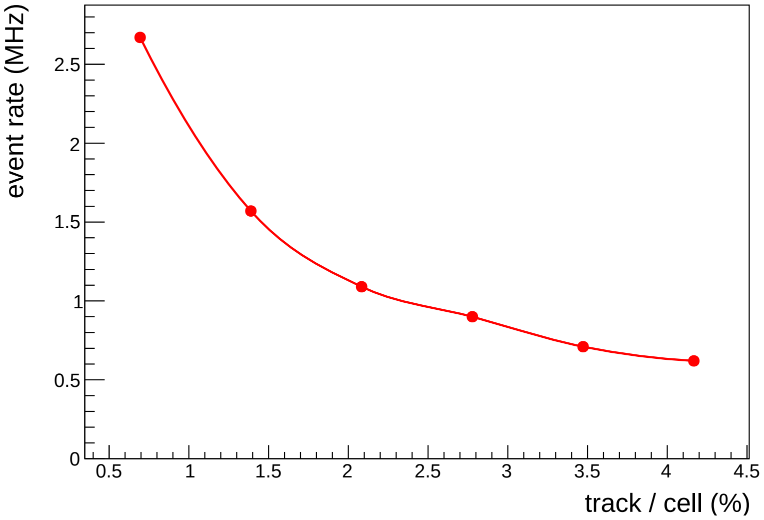

Figure 5.5: Functional prototype event rate as a function of the occupancy of the system.

## 5.4 The Board for the functional prototype

Here I will give some details about the TEL62 board [73] where the functional prototype has been implemented. The TEL62 board, shown in Figure 5.6, was designed by INFN Pisa for the data acquisition and trigger of the NA62 experiment. This board is provided with 5 Stratix III FPGAs. Four FPGAs, called Pre-Processing (PP), are dedicated to data processing. In the original design the PP device receives data through a connector that includes 4 32-bit channels and can host a mezzanine daughter-card. Each PP FPGA is also connected to the neighbors with 2 16-bit buses on each side. The fifth device, called SyncLink (SL), receives data from each PP through 2 32-bit buses and send them out through another connector where is installed a Gigabit Ethernet card. For each bus, we have additional lines that carry clock and various control signals. The maximum clock speed of the Stratix III FPGA is 350 MHz. An embedded PC (CCPC) grants the control of the board and provide access to the internal registries of FPGAs.

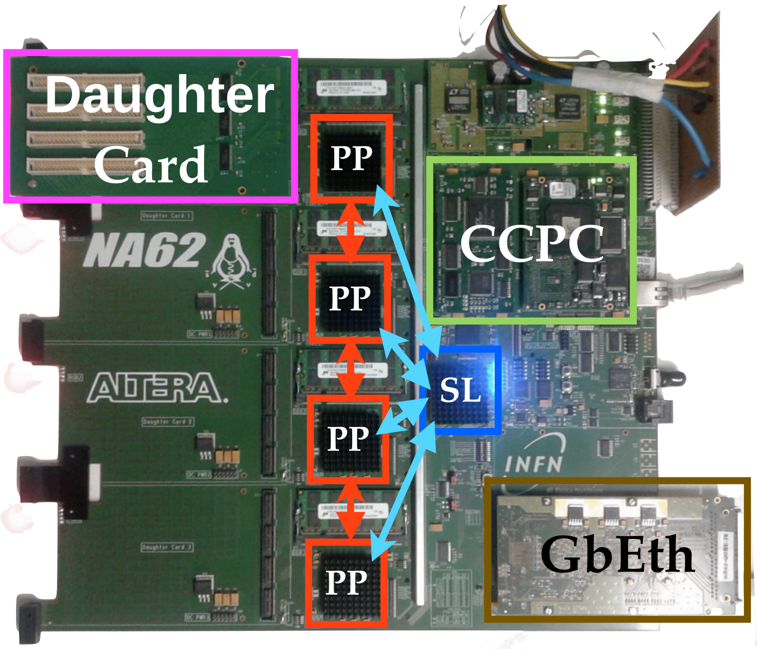

Figure 5.6: The TEL62 board.

The design of the functional prototype had to be adapted to the existing structure of TEL62. Therefore, two boards were connected together to build the prototype using simple interface cards, as shown in Figure 5.4. The connections of TEL62 where used to implement the switching network, to bring data to the engine board, and then out through the SL and the network card. Implementing the connections between all the PP devices of the switch board was particularly cumbersome, as it is shown in Figure 5.7.

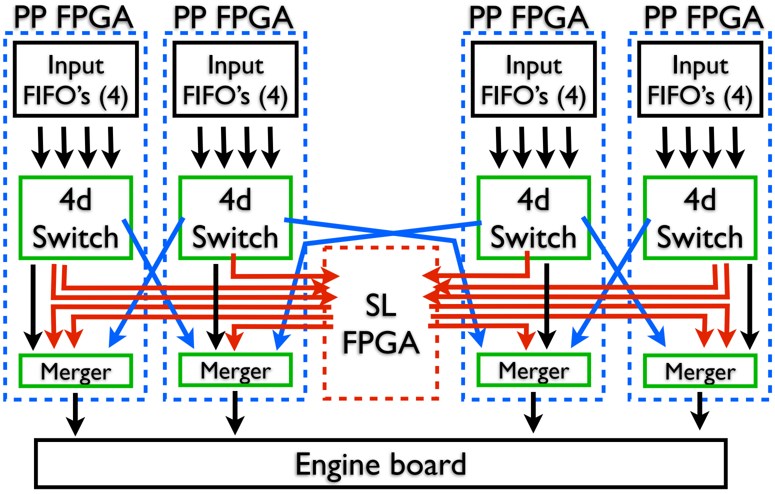

Figure 5.7: Connections between all the PP devices of the functional prototype switch board.

## 5.5 Hardware choice for the High-Speed Prototype

During this thesis I developed the first High-Speed prototype of the "Artificial retina" design, using current electronic devices. For this purpose, rather than designing a whole new board from scratch, it would be ideal to be able to use a pre-existing board, if one could be found providing both the power and the flexibility needed for our purposes. After considering a large number of candidate options, the choice fell on a commercially available board, called DN0237, designed and manufactured by a small company, aimed at ASIC prototyping. This board was chosen because it mounts two high-capability FPGAs and 96 external high speed serial lines (12.5 Gbits/s per line) providing a total I/O bandwidth toward the external world in excess of 1.2 Tbits/s. This is an unusually large bandwidth to be found on a single board, as most applications do not require it. Each chip has $\sim 5$ times more logic elements than the TEL62 FPGAs. These modern FPGAs allow to evaluate the speed performances of the system implemented on modern hardware that support a clock frequency twice as high as than TEL62. The high number of serial lines provide a total I/O bandwidth of 600 Gbits/s to each FPGA, allowing a fast connection between the two FPGAs or other external devices for testing the scalability of the system.

The basic diagram for the DN0237 board is shown in Figure 5.8. The main devices of the board are two FPGA Stratix V, called User FPGAs (UFPGAs), with about one million logic element each, and where the user can load his firmware. We made a custom-order for our board, in order to have it mount the largest existing chips of the family. The maximum clock speed of these FPGAs is 650 MHz. The two FPGAs are also directly connected through 161 LVDS lines inter-FPGA, allowing tests in which the two chips are tightly interacting, to behave nearly as a single device. Each device has 48 high-speed serializers accessible from outside through two daughter cards (I/O Board) that can host up to 12 QSFP optical transceiver. A Marvell Discovery microprocessor and another FPGA, called Config FPGA (CFPGA), allow to control the board, program the main devices, access their internal registries, and manage various other functions.

Given the larger amount of logic blocks available in the main FPGAs, it is surely feasible to port the design from two TEL62 boards to a single DN0237. Given the reduced device occupancy for the switch board in the functional prototype, it should also possible to implement the prototype in a single chip. In Chapter 6 I will describe in detail how I ported the design of the functional prototype to the DN0237 board, first using two devices, and then just a single one.

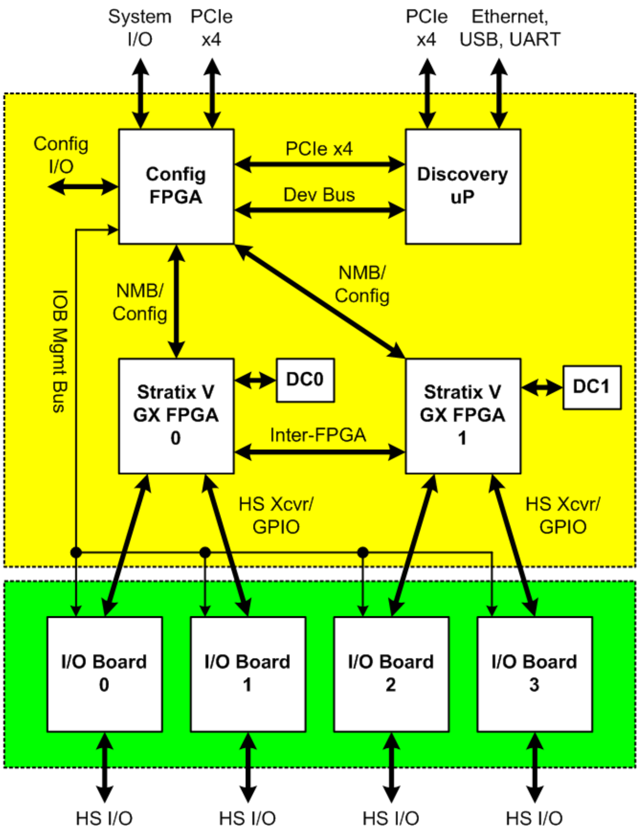

Figure 5.8: Basic Diagram for DN0237 Main Board showing inter-FPGA connections and serializers connected to I/O devices.

# Chapter 6

# Prototype implementation and performance evaluation

## 6.1 Building the High-Speed prototype

The first step of this work was to port the design of the functional prototype to the DN0237. This is a good starting point for the project, and allows a first assessment of the performance achievable with the current generation FPGA.

For the purpose of this discussion, it is necessary to explain how the "Artificial Retina" components are placed in the TEL62. Figure 6.1 shows the diagram of the components implemented on TEL62 boards. We can recognize three main parts: the board #0 with the switching network, the 8 interconnection channel between the boards, and the board #1 with the Engine. The interconnections between the boards use $18 \cdot 8 = 144$ lines. As discussed in Section 5.5 161 lines are available to connect the two FPGA on the board. Furthermore FPGAs have $\sim 5$ times more

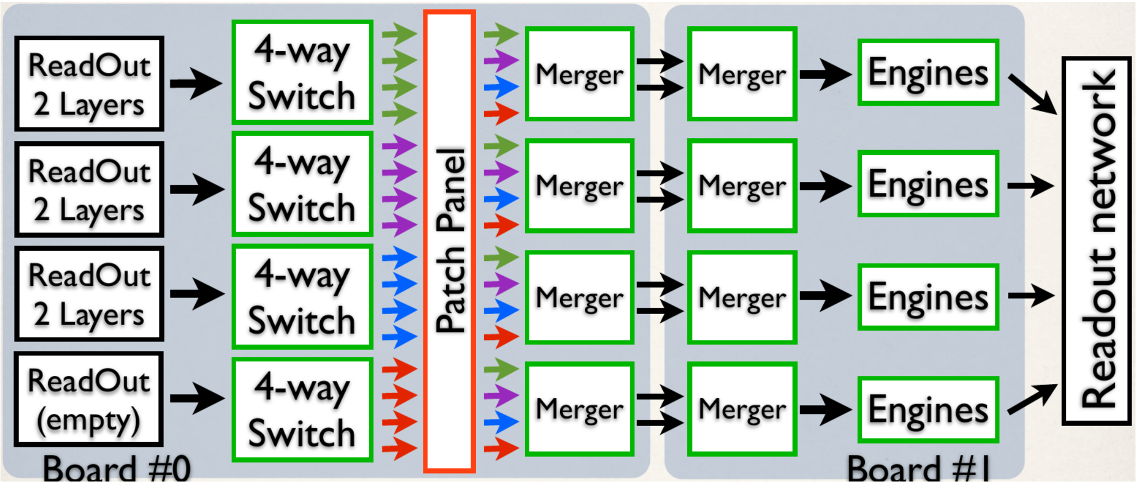

Figure 6.1: Diagram of the components implemented on TEL62 boards.

logic elements than the FPGAs in the TEL62 therefore I implemented the firmware of a whole Tel62 in a single FPGA of DN0237. Table 6.1 shows the ALMs utilization for the FPGA on both TEL62 and DN0237. During developing the firmware has to be compiled several times. This task require a lot of time ($> 4\,\text{hr}$). For this reason I compiled the firmware with $16 \times 15$ Engine matrix only a few times to verify that the firmware fit in the FPGA, for any other analysis I used just a $8 \times 8$ Engine matrix. Table 6.2 shows the ALMs utilization for the various FPGAs using $8 \times 8$ Engine matrix.

| | TEL62 | | DN0237 |
|---|---|---|---|
| | PP | SL | |
| Board #0 | 8% | 3% | 6% |
| Board #1 | 90% | | 70% |

Table 6.1: Comparison of ALMs utilization with $16 \times 15$ Engine matrix.

| | TEL62 | | DN0237 |
|---|---|---|---|
| | PP | SL | |
| Board #0 | 8% | 3% | 6% |
| Board #1 | 23% | | 17% |

Table 6.2: Comparison of ALMs utilization with $8 \times 8$ Engine matrix.

In the functional prototype the switching network is distributed over 4 PP FPGAs. The communication channel between these FPGAs can not reach a high speed (max 400 Mbits/s) and require to use the SL FPGA as a bridge. This is not an optimal solution for the "Artificial Retina" architecture. In the DN0237 I reproduce the structure of the functional prototype, but inserting the logic in a single FPGA allows to reach an higher speed.

The functional prototype use the TEL62 communication protocol (ECS). With this protocol, a PC write the hits in a memory inside the PP FPGAs of the board #0. The communication with the DN0237 use the NMB protocol, a proprietary protocol developed by the board manufacturer. I designed an interface between NMB and ECS. This interface allows to write and read register and memory of the ported design, using the C++ library provided by the board manufacturer, without modifying the communication protocol of each component.

I also write C++ and Python code, called RetinaSpy, to configure and control the High-Speed prototype. A software simulation described in Section 7.1 simulates every step of the "Artificial Retina". RetinaSpy can compare the High-Speed prototype output with the simulation output, checking that output is correct.

For testing the prototypes we need to provide input data at high speed, but the interface protocol is too slow. For getting fast input data we need to use local RAMs, but even 1 Gbytes of events would be processed in only 22 s, assuming an event rate of $\sim 1$ MHz. Therefore I stored hits from about 100 events onto RAMs inside the FPGA, and read these RAMs on a never-ending loop, so the system can run indefinitely. Indeed, I was able to run the prototype for days without errors.

### 6.1.1 Event rate

The event rate is the frequency at which the prototypes reconstruct the events. It is directly proportional to how many hits are distributed in the switching network and are processed in the Engines. The number of the hits is directly proportional to the number of the tracks. Increasing the number of the Engine, the number of hits processed on average by each Engine decreases. Adding more Engines requires a wider switching network, that will have more bandwidth. The number of the Engine is equal to the number of the cells. Thus we define the occupancy as the number of tracks divided by the number of cells. At occupancy 0%, there are no tracks hits, but there are still the End Events (EEs) that indicate the end of the event end not carry hit coordinate. The event rate at occupancy 0% is our upper limit, representing the behavior of the system when the detector is only sparsely populated. We can increase this limit only increasing the clock frequency or changing completely the data structure.

At each arrow in Figure 6.1 an event counter is implemented. The event counter count how many EEs transited. Every 256 EEs it sends a pulse to a output pin. With an oscilloscope is possible to view the pulse. I calculated the event rate by measuring the frequency of the pulses.

The clock frequency used for the functional prototype is 40 MHz for the board #0 and 160 MHz for the board #1. The first test was performed using for both FPGAs the same clock frequency used for the functional prototype board #1. Figure 6.2 and Table 6.3 report the event rate as a function of the occupancy for the functional prototype (TEL62 40/160 MHz) and the first test of the High-Speed prototype (DN0237 160/160 MHz). Increasing this clock by a factor 4 leads to a event rate gain about a factor 3. This gain does not reach the factor 4 because in the functional prototype the bottleneck was the board #0, actually in this configuration the bottleneck is the FPGA #1 of the High-Speed prototype.

The firmware was recompiled using the maximum clock frequency allowed. The second test was performed with FPGA #0 and FPGA #1 clock frequency set to 240 MHz and 340 MHz respectively. The event rate measures are visible in Figure 6.2 and Table 6.3 with the label "DN0237 240/340 MHz". The event rate is 6 times greater then the event rate of the functional prototype. The event rate scales properly with the clock frequency. The obtained number now for the first time touches the 40 MHz line as an upper limit. The bottlenecks of this configuration are the mergers before and after the interconnection between the FPGAs.

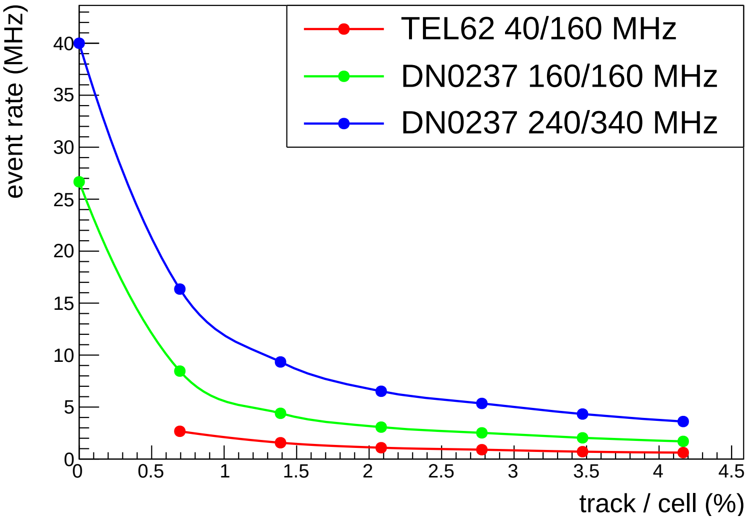

Figure 6.2: Comparison of event rate as a function of the occupancy of the system.

| track/cell (%) | event rate (MHz) | | |
|---|---|---|---|
| | TEL62<br>40/160 MHz | DN0237<br>160/160 MHz | DN0237<br>240/340 MHz |
| 0.00 | | 26.67 | 40.00 |
| 0.69 | 2.67 | 8.46 | 16.35 |
| 1.39 | 1.57 | 4.40 | 9.34 |
| 2.08 | 1.09 | 3.07 | 6.52 |
| 2.78 | 0.90 | 2.52 | 5.35 |
| 3.47 | 0.71 | 2.04 | 4.33 |
| 4.17 | 0.62 | 1.70 | 3.61 |

Table 6.3: Comparison of event rate as a function of the occupancy of the system.

### 6.1.2 Latency

Latency is a time delay between the input of a signal in a system and the response of this. We measure the latency of the components placed between two event counters as the time between the transit of the same EE in the two event counters. Indeed every event counter sends a pulse to its output pin for the same event. With an oscilloscope I measured the delay between two pulse generated by the desired event counters. When the input rate is higher than the event rate measured in Section 6.1.1, some buffers between the components are filled completely, and the latency reach a value depending only on the buffer size, has shown in Figure 6.3. Queueing theory predicts that the latency isn't constant for input rate lower than the maximum device throughput, while above this limit is determined simply by the total length of the internal buffers that get completely filled, and is therefore not a meaningful evaluation of what we want to know. I modified the loop-ram adding the possibility to select the input event rate. In this way I can measure the latency below the saturation point.

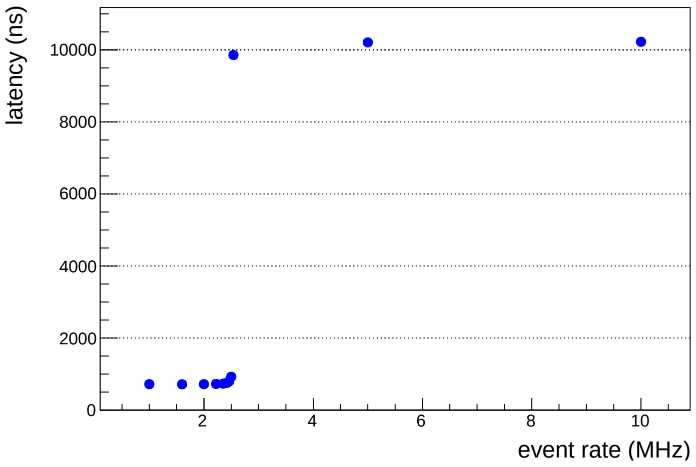

Figure 6.3: Total latency of the High-Speed prototype as a function of the event input rate.

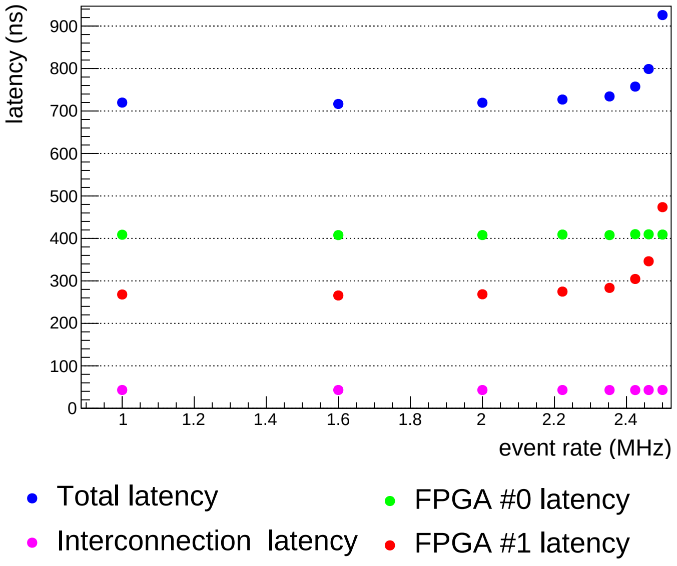

Figure 6.4: Latencies of the High-Speed prototype as a function of the event input rate.

Latency measurements were performed by setting the clock frequency of the two FPGA to 160 MHz. Measuring the latency with other clock settings is not necessary. Indeed the latency scale with clock frequency, and the proportional factor is the number of clock cycle used by the components for accomplish their functions. This number change only changing the logic of the components. The latency measures were performed at occupancy 2.78%. We chose this value because is the occupation level expected for the detector reproduced by the prototypes. We measure the latency of the FPGA #0, the interconnection between the FPGAs, and the FPGA #1. Figure 6.4 shows the latencies measured. The total latency of the High-Speed prototype is ∼ 720 ns. As discussed in Section 4.1.2 to integrate the "Artificial Retina" to a DAQ system the latency must be less than few µs. The measured latency is compatible with our purpose even after taking in consideration that the complete system will have further latency contributions, that we can prudently estimate to be an additional ∼ 1 µs.

## 6.2 Multi-channel Engines

As mentioned in Section 6.1.1, the bottlenecks of this design are the mergers before and after the interconnection between the FPGAs. Since the weigh calculated for each hit get added together, we can add the hits coming from the same readout channel to form a partial sum, then add the partial sums together. This allows to evade the bottleneck produced by merging the readout channel. The Engine developed for the functional prototype contains three partial accumulator (Section 4.2.3), but only has one input channel.

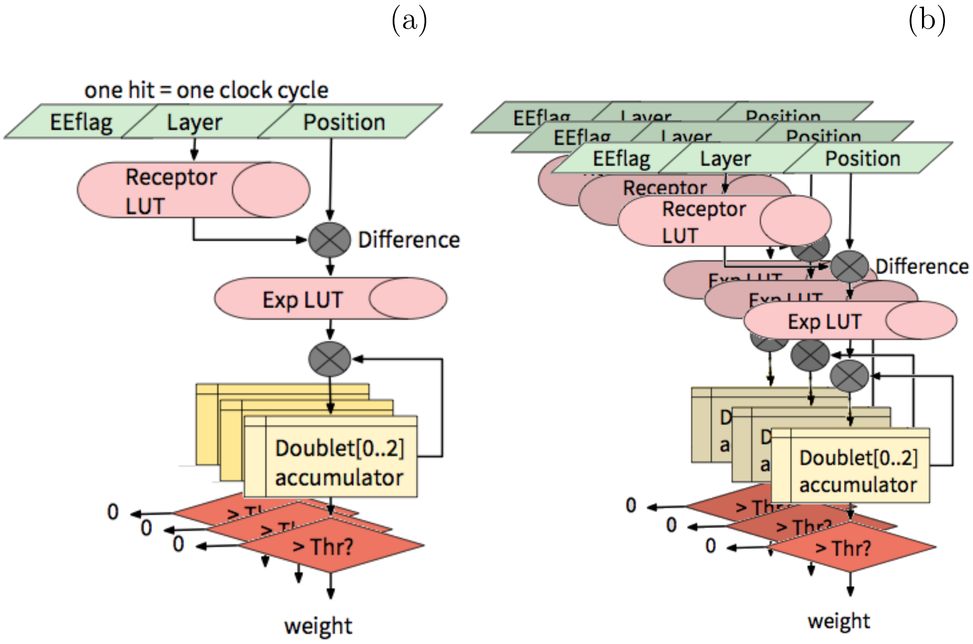

Figure 6.5: Engine developed for functional prototype (a) and Engine with multi-channel input (b).

I modified the Engine to have as many input channel as partial accumulator. I also rewrote its internal control logic. The control logic send a End Event signal when all the EE of the same event are delivered to each input channel.

Connecting the two FPGA with 3 channel for each Engine Matrix require $3 \cdot 4 \cdot 18 = 216$ lines. The DN0237 board does not have enough inter-FPGA lines. So I implemented dual-channel Engines and removed the mergers in the FPGA #1. The ALMs utilization for FPGA #1 is increased to 19% using $8 \times 8$ Engine matrix and to 77% using $16 \times 15$ Engine matrix.

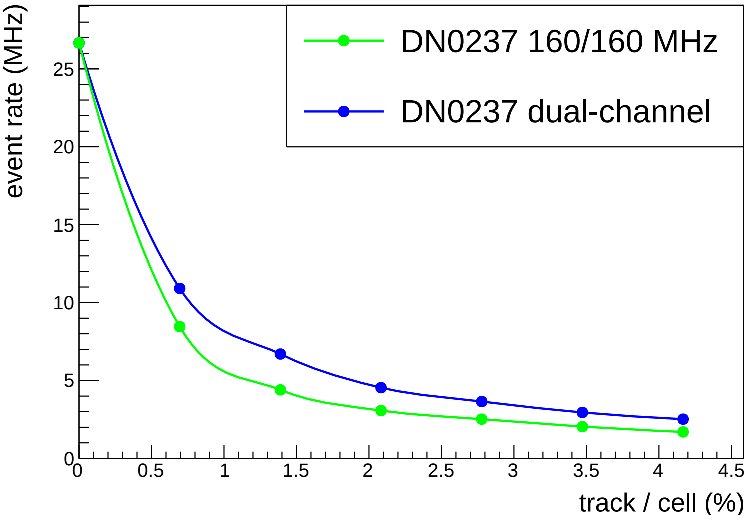

Figure 6.6: Comparison of event rate between the design with single-channel Engine and a design with dual-channel Engine.

| track/cell (%) | event rate (MHz)<br>DN0237<br>160/160 MHz<br>single-channel | <br>DN0237<br>160/160 MHz<br>dual-channel |
|---|---|---|
| 0.00 | 26.67 | 26.67 |
| 0.69 | 8.46 | 10.91 |
| 1.39 | 4.40 | 6.70 |
| 2.08 | 3.07 | 4.54 |
| 2.78 | 2.52 | 3.65 |
| 3.47 | 2.04 | 2.95 |
| 4.17 | 1.70 | 2.52 |

Table 6.4: Comparison of event rate between the design with single-channel Engine and a design with dual-channel Engine.

In spite of the doubling of the number of Engine input channels, the gain is limited because not having changed the switching network in FPGA #0, only one third of the hits are delivered to the new input channel. We can expect a gain of a factor 1.5. I tested this design with clocks at 160 MHz. Figure 6.6 and Table 6.4 show the event rate achieved with the new design, the measure reproduce the expected gain. The event rate at occupancy 0% doesn't change. This happens because, as discussed in Section 6.1.1, we can increase this event rate only increasing the clock frequency or changing completely the data structure.

## 6.3 Further optimization: monolithic implementation

We want to test the maximum performances of the prototype with the available hardware. In particular we want to implement triple-channel Engines and to remove the switching network components that are no longer needed. For connect the switching network to the Engines, there aren't enough inter-FPGA lines, so I changed to a new implementation, where switching network and Engines reside in the same device. While the final system can't be implemented in a single device, this test is still a good measurement of the maximum event rate achievable by the system because in the final, distributed architecture described in Section 4.1.2, the number of connections between FPGAs will not be a limitation.

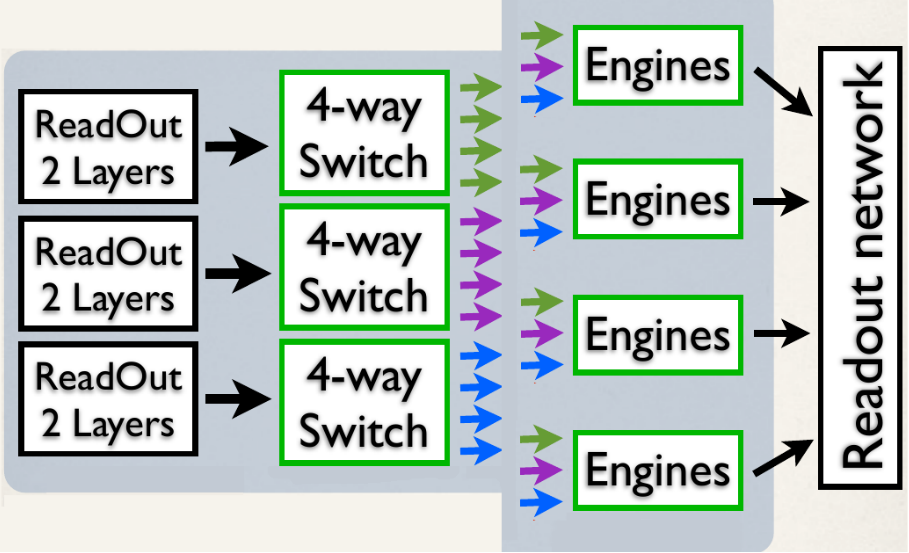

Figure 6.7: Diagram of the components implemented in the new optimized design.

Figure 6.7 shows the new design. This new design allows to get rid of a few components, thus saving logic: the empty ReadOut. a 4-way Switch, the Patch Panel, the eight merger. Engines were implemented with 3 input channels. The ALMs utilization by this design is 27% using 8 × 8 Engine matrix and 93% using

16 × 15 Engine matrix. The system proved to be working properly using a clock of 400 MHz for the switching network and the a clock of 280 MHz for the Engines. The event rate gain of this design with respect to the porting of the design described in Section 6.1 is $\sim 2$. Figure 6.8 and Table 6.5 show the measured event rate. An event rate of 30 MHz (expected for LHCb) can now be sustained up to a non-negligible device occupancy of 0.5%. The total latency of the System is also further decreased, reaching a very comfortable figure of 442.5 ns. It is worth mentioning that the best tracking processors designed for HEP have never attained latencies below the level of several microseconds, typically $10 - 20$ µs at best.

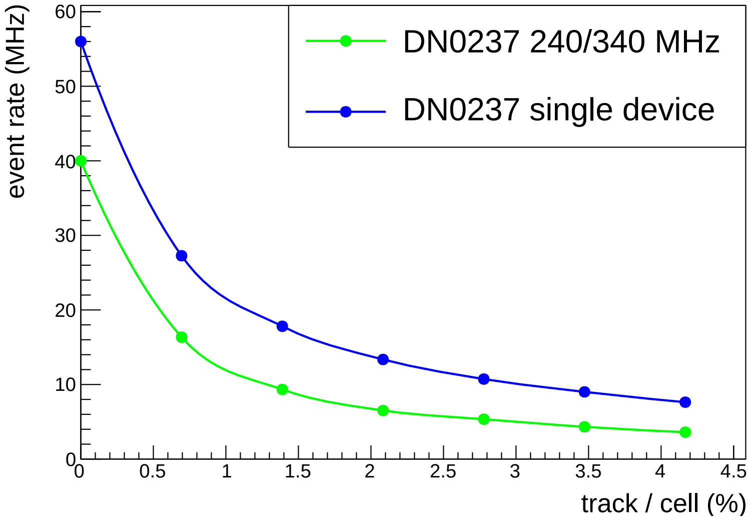

Figure 6.8: Comparison of event rate between the porting of the design described in Section 6.1 and the single device design.

## 6.4 Final remarks

I have assembled and tested the first high-speed physical prototype of a tracking processor based on the “Artificial Retina” concept, using commercially available components and boards. I measured throughput and latency, and found that it is possible to operate the system, on a 6-layer planar detector, with an event rate of 30 MHz and a latency $< 0.5$ µs, up to a detector occupancy of $\sim 0.5\%$. These results are extremely encouraging regarding the possibility of realizing a generic real-time tracking device operating at the LHC crossing rate based on this technology , and more specifically the LHCb downstream tracker we are aiming at.

It should be noted that these figures are subject to several further improvements. The current switch logic is not exploiting every clock cycle for data transfer, and it could conceivably be optimized to this aim, yielding a further increase in throughput.

| | event rate (MHz) | |
|---|---|---|
| track/cell (%) | DN0237 | DN0237 |
| | 240/340 MHz | 400/280 MHz |
| | | single device |
| 0.00 | 40.00 | 56.00 |
| 0.69 | 16.35 | 27.27 |
| 1.39 | 9.34 | 17.81 |
| 2.08 | 6.52 | 13.36 |
| 2.78 | 5.35 | 10.73 |
| 3.47 | 4.33 | 9.01 |
| 4.17 | 3.61 | 7.63 |

Table 6.5: Comparison of event rate between the porting of the design described in Section 6.1 and the single device design.

Further optimizations of the implementation logic are still conceivable. The FPGA devices we are using are current, but not the very top of the line, and not the highest speed grade; better chips are already on the marked today, and by the time the Downstream Tracker hardware will actually need to be bought, the components available will be still better than what we have now.

However, given that the current results already provide a strong indication of feasibility, rather than trying to pursue further optimizations, we will move on to tackle further relevant questions for the project in the following chapters, regarding track reconstruction quality, practical viability in comparison with alternative possibilities, and more LHCb-specific implementation issues.

# Chapter 7

# Tracking Performances

## 7.1 High-level utility software description

A software package of C++ utility programs have been developed over time in parallel with hardware prototypes. These programs generate the configuration files and simulate the "Artificial Retina" architecture. Figure 7.1 shows the general structure of this package. In Section 4.2.1 we described the Detector-Mapping functions. Detector-Mapping divides the phase space in cells and calculates track parameters associated to each cell. Using the same C++ classes of the detector simulator (described below), Detector-Mapping calculates the receptors of the cells. Checking which receptors are near than the distance search to specific areas of the detector, Detector-Mapping also writes the configuration file for the switching network. The track generator produces tracks according to predefined distribution and parameters, and can be forced to produce tracks in predefined regions of the parameters space. In our test, we generate tracks uniformly in the mapped phase space. The detector simulator calculates the hits of the tracks on the detector, taking into account resolution and efficiency of the detector. This software utilizes as input the detector geometry description and the generated tracks parameters.

The main program, "Artificial Retina" simulator, processes the hits, calculates the excitation level of each cell, and performs clusterization. It can run using floating point values or, as in the hardware case, using integer values. In the latter mode, it can produce the results expected from the device at the bit-level. The "Artificial Retina" simulator saves the output of each stage. These informations have been used to verify the correctness of the behavior of the hardware prototypes in the tests described previously. The "Artificial Retina" simulator also allows to process hits generated from an external source, thus allowing a great flexibility in the types of tests that can be performed.

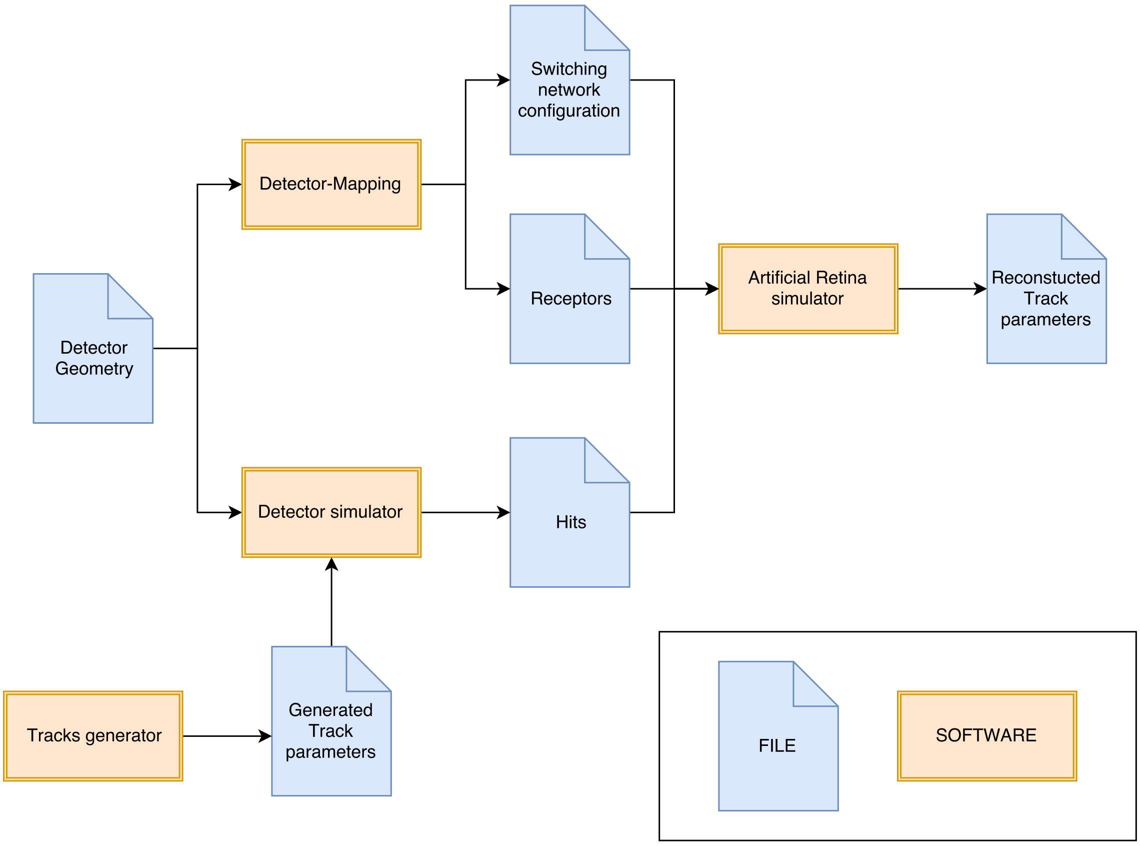
Switching network configuration
Detector-Mapping
Receptors
Artificial Retina simulator
Reconstucted Track parameters
Detector Geometry
Detector simulator
Hits
Tracks generator
Generated Track parameters
FILE
SOFTWARE

Figure 7.1: The “Artificial Retina” configuration and simulation flow.

## 7.2 Tracking Study Using the Simulation

The high-level software simulator is a much more practical instrument for studying the tracking performance parameters of our device than the hardware prototype, due to the much greater ease of setting up, recording and analyzing complex data sets. We want to study the tracking performances, like the efficiency and the ghost rate, *i.e.* the rate of the false positive.

For these studies we used a detector made by 3 equidistant layers (Figure 7.2), with a granularity of 10 bit (819 channel). The parameters of the phase space are the hits on the first layer ($U$) and on the last ($V$), and the tracks are uniformly distributed in the phase space. The “Artificial Retina” covers the track parameters space with a grid of 576 cells.

I extended the “Artificial Retina” simulation with the addition of a matching algorithm. This algorithm is based on a distance between tracks in the parameter space. We define the distance between a reconstructed and a generated track as:

$$d_{match} = \sqrt{(U_{rec} - U_{gen})^2 + (V_{rec} - V_{gen})^2}$$

where $U_{rec}$ and $V_{rec}$ are the parameters of the reconstructed track, and $U_{gen}$ and $V_{gen}$

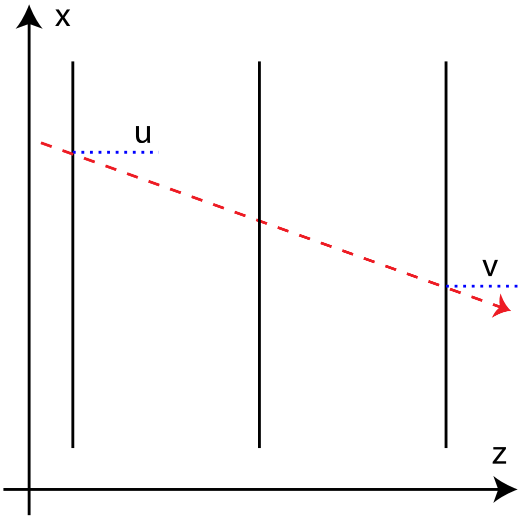

Figure 7.2: Minimal tracking detector.

are the parameters of the generated track. The reconstructed track parameters are calculated from the centroid of the cluster (Section 4.2.4).

Figure 7.3 shows the $d_{match}$ distribution for occupancy 0.17% and 0.7%. The peak on the left represents the correctly reconstructed tracks, the right part of the distribution being contributed by fake tracks produced by incorrect hit associations ("ghost tracks"). From these distributions it is possible to see how the performances of the "Artificial Retina" architecture depend from the occupancy of the system.

We define the efficiency and the ghost rate as:

$$\varepsilon = \frac{\#\, tracks\, matched}{\#\, tracks\, generated}$$

$$Ghost = 1 - \frac{\#\, tracks\, matched}{\#\, tracks\, reconstructed}$$

I implemented an algorithm for the track matching that searches, for each reconstructed track, if there is a generated track within a maximum distance. We chose as value for the maximum matching distance the minimum of the distance distribution shown in Figure 7.3 (b), in this case 1.25 pitch of the cells grid.

Whether or not a ghost rate at this level is acceptable will depend on the application, but it is not a negligible number. Moreover, if we look at occupancies beyond 1%, the Ghost fraction rapidly grows up to unacceptable levels - for instance, at occupancy 0.7% the ghost rate is $\sim 26\%$, at occupancy 1.4% the ghost rate is 54%. It can therefore be argued that the use of a final refitting stage after cluster finding, for which allowance was made in the block diagram of Figure 4.2, can actually be necessary, at least in some cases. We therefore performed a preliminary study of

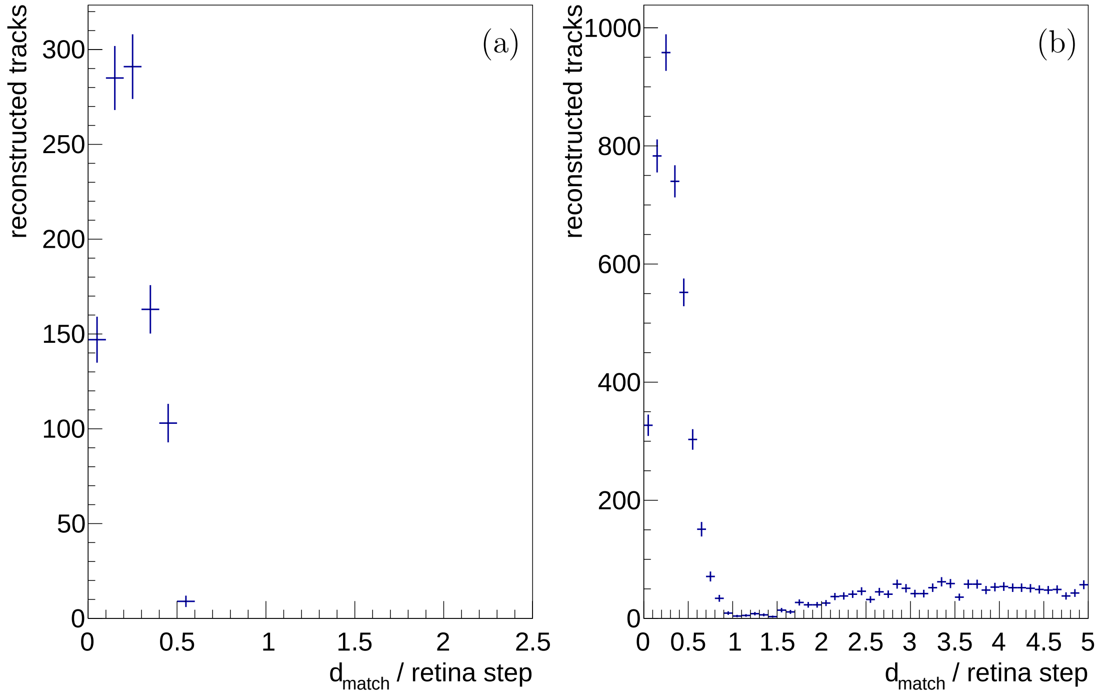

Figure 7.3: Distribution of the distance of the reconstructed tracks from the nearest generated track at occupancy 0.17% (a) and 0.7% (b). The distance is normalized to the pitch of the cells grid.

the benefits that might be expected from such fitting stage, as discussed in the next section.

## 7.3 The Linearized $\chi^2$ fit

To reduce the ghost rate we explored different solutions alternative to the centroid calculation, for example performing a fast fit for every local maximum found. In this case the Engine selects a hit for layer, for example the nearest to the receptor, then it performs a linearized fit. The $\chi^2$ of the fit can be used to discriminate real tracks from false positive.

The $\chi^2$ fit aims at recognizing real tracks from accidental combinations of hits [74]. Given that every candidate track is associated to an array on $n$ hits coordinates, the candidate track can be thought as a point within a set $\mathcal{C} \subset \mathbb{R}^n$. Not every point in $\mathcal{C}$ is equally likely to represent a track: only the vector $\overline{x}$ representing hits aligned along a track path have chances to represent real tracks. The set of all $\overline{x}$ aligned along a track path is the subset $\mathcal{T} \subset \mathcal{C}$. Suppose each track can be parameterized by a number $m < n$ of real parameters $\overline{p}$ (*e.g.* $U$, $V$), in the ideal case of perfect resolution, the set $\mathcal{T}$ reduce to a $m$-dimensional surface contained in $\mathcal{C}$, described by

$n$ parametric equations:

$$\overline{x} = \overline{x}(\overline{p})$$

that can also be cast in implicit form, yielding $n - m$ constraint equations:

$$f_i(\overline{x}) = 0 \quad i = 1, \ldots n - m$$

These constraint function can be determinated from the knowledge of the geometry of the detector. For each candidate track one evaluates the $f_i(\overline{x})$, and accepts the track if all $f_i(\overline{x})$'s are zero. The effect of finite resolution is to make $f_i(\overline{x})$'s slightly different from zero. The measure of this effect is given by the covariance matrix $F_{ij}$ of the $f_i(\overline{x})$, which can ba calculated at the first order from the covariance matrix of the coordinates $\overline{x}$. A $\chi^2$ can be formed as:

$$\chi^2 = \sum_{ij} f_i \cdot F_{ij}^{-1} f_j$$

A cut on this quantity can be used to select good tracks with any chosen efficiency. Its value is the same one would obtain for the minimum of the usual $\chi^2$ from a standard fitting procedure of the parameters $\overline{p}$. However, here we do not get any parameters value, that we don't need for track finding. $F_{ij}^{-1}$ is symmetrical, then it can be diagonalized, and the constraints redefinited accordingly:

$$F_{kl}^{-1} = M_{ki} \frac{\delta_{ij}}{\sigma_i^2} M_{jl}$$

$$\widetilde{f}_i = \frac{M_{ij} f_i}{\sigma_i}$$

$\chi^2$ expression can be simplified and rewrites as:

$$\chi^2 = \sum_i \widetilde{f}_i^2$$

That is a very simple expression that is fast to evaluate from the $\widetilde{f}_i$'s, requiring just $n - m$ multiplications and $n - m - 1$ sums.

We need to compute the values of the constraint functions $\widetilde{f}_i(\overline{x})$ for each candidate track. Generally speaking, they can be quite complicated functions, however, experience has shown that in vast majority of tracking problems they can be approximate with quite good precision by linear expansions about some convenient point $\overline{x}_0$:

$$\widetilde{f}_i \simeq \frac{\partial \widetilde{f}_i}{\partial \overline{x}} \cdot (\overline{x} - \overline{x}_0) = \overline{v}_i \cdot \overline{x} + c_i$$

Geometrically, this amounts to approximating $\mathcal{T}$ with its tangent hyperplane in $\overline{x}_0$, and $\overline{v}_i$'s are the vector orthogonal to the hypersurface in $\overline{x}_0$. The approximation works well when $\mathcal{T}$ is nearly flat. In general, in order to obtain a sufficient precision, it is necessary to segment $\mathcal{T}$ in several smaller region, and perform the expansion around the central point $\overline{x}_0$ of each of them.

All constant can be calculated numerically starting from a sample of vector $\overline{x}$ belonging to $\mathcal{T}$. The variance of any linear function $y(\overline{x}) = \overline{v} \cdot \overline{x} + c$ evaluated in this sample is given, to the first order, by:

$$\sigma_y^2 \simeq \overline{v} \cdot M \cdot \overline{v}$$

where $M$ is the covariance matrix of $\overline{x}$, estimated from the sample as:

$$M \simeq \frac{N}{N-1} \langle (x_i - \langle x_i \rangle)(x_j - \langle x_j \rangle) \rangle$$

If it append that $y = 0$ for all $\overline{x}$ in our sample, then $\overline{v} \cdot M \cdot \overline{v} = 0$, and since $M$, being a covariance matrix, has no negative eigenvalues, it follows that $\overline{v}$ is an eigenvector of $M$ with eigenvalue 0. We can find out the $\overline{v}_i$ taking any base of the Kernel of $M$. The corresponding constant $c_i$ are determined by imposing $\langle \widetilde{f_i} \rangle = 0$, that gives $c_i = -(\sum \overline{v}_i \cdot \overline{x})/N$.

The linearized $\chi^2$ fit was used in the CDF Silicon Vertex Tracker (SVT) [74] to perform track finding and fitting at high speed inside ASIC. This approach is very compatible with the FPGA implementation because modern FPGA have a high number of DSP to perform multiplications at high speed. In addition, due to the use of a fine subdivision of the parameter space in the retina matrix, the "Artificial Retina" approach already has a grid of expansion points conveniently available.

I implemented the linearized $\chi^2$ fit inside the existing simulation code. I assume each Engine to have the capability of storing the nearest hit to the corresponding receptor, a functionality extension that can be realized with a very small amount of additional logic. This approach is obviously an approximation depending on the occupancy. Indeed the nearest hit to the receptor can not be the hit of the generated track, if the many hits are near.

For testing the efficiency of the linearized $\chi^2$ fit, I chose some ghost rate values, then I tightened the cut on $\chi^2$ to reach these values, after that I measured the reconstruction efficiencies achieved. As comparison I also measured the efficiency using the centroid method: raising the threshold on the Engines excitation level, without cutting on $\chi^2$. Table 7.1 shows the results of this test, and that the linearized $\chi^2$ fit is able to reject ghosts with an efficiency greater than the simple threshold on the excitation level.

| | centroid | | linearized $\chi^2$ fit | |
|---|---|---|---|---|
| ghost rate | excitation level | $\varepsilon$ | $\chi^2$ | $\varepsilon$ |
| 26% | $> 550$ | 98.5% | no cut | 98.5% |
| $\sim$ 12% | $> 700$ | 65.9% | $< 1100$ | 90.0% |
| $\sim$ 1.5% | $> 1050$ | 2.7% | $< 8$ | 78.5% |

Table 7.1: Comparison of the efficiency achieved by the "Artificial Retina" discriminating track on excitation level of the centroid and on the linearized $\chi^2$ fit. Occupancy 0.7%

## 7.4 Power comparison with a traditional architecture.

The study of the previous section has shown that the tracking performance of the retina approach is strongly dependent on the desired level of efficiency and on the occupancy, that in turn depends on the size of the engine cell array. Obviously, given a sufficiently large array of cells, the system can always be made to perform as well as desired. Therefore, the previous considerations are insufficient to asses the promise of the approach unless a comparison is made with some other alternative methodology. We have therefore decided to perform a comparison with a similarly performing tracking system based on a standard commercial CPU. The results and the definitions introduced previously can now be used to define an "equivalent level of performance".

The other needed element for a comparison is a parameter comparing the cost/size/complexity of the two implementations being compared. This is much harder to achieve, given their heterogeneity and the existence of many possible terms of comparison. For our purposes of a first gross comparison with no pretense of precision, we have decided that a comparison based purely on equivalent electric power consumption was the best we could do. This is, after all, a quite reasonable parameter, considering that

a) in modern data processing system, this is the most important factor contributing to the total operating cost

b) it is in direct proportion with most other cost elements, like cooling, size of installation, power plant, etc.

For the purpose of this comparison, we have chosen a recent model of CPU, and wrote and optimized a piece of C++ code performing a tracking task as similar as possible to the task we were running on our prototype.

While in the "Artificial Retina" the Engines provide the hits sets to the fitter, the CPU needs to try all the hits combinations. In a generic $n$ layers case, complex optimizations to the CPU algorithm are possible, but by considering a simple 3 layer tracker, the mandatory procedure is to consider all the combination of 3 hits, one hit for layer. While this is not the most interesting, or representative case to study, we have therefore decided to use for this comparison a very simple detector. The detector is made by 3 equidistant layers, and was described at the begin of Section 7.2. At least, we know that this is a worst-case for our approach, that is intended to take greater advantage from the presence of multiple layers.

With the simulation described in Section 7.1 we have generated the tracks in the detector end simulated the hits on layers. Then we processed this set of hits with the "Artificial Retina" simulation. I wrote a C++ program to calculate the $\chi^2$ on every combination of 3 hits of the same set ("CPU fitter"). "CPU fitter" performs a linearized $\chi^2$ fit with exactly the same parameters used by the retina looping on every hits combination, if the $\chi^2$ of a combination is under a threshold it calculates the parameters of the track.

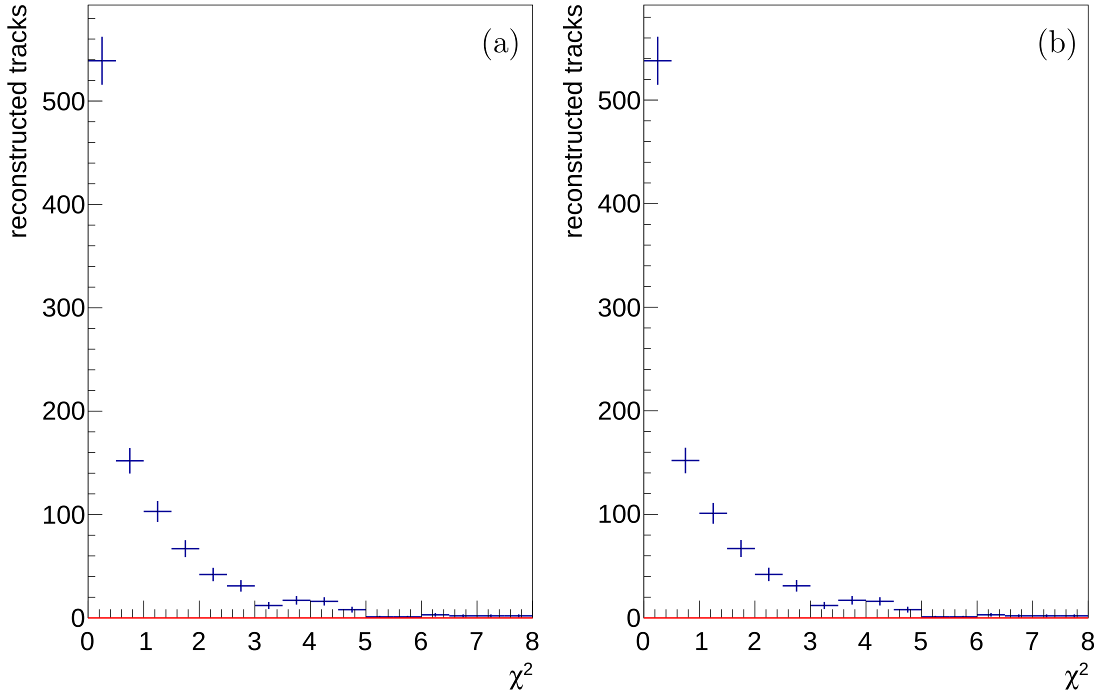

Figure 7.4: $\chi^2$ distribution of reconstructed tracks in events with 1 track generated of each event (occupancy 0.17%). Tracks reconstructed via "CPU fitter" (a) and "Artificial Retina" (b). Track matched in blue, ghost tracks in red.

| | $\varepsilon$ | ghost rate |
|---|---|---|
| "CPU fitter" | 99.8% | 0% |
| "Artificial Retina" | 99.5% | 0% |

Table 7.2: Efficiency and ghost rate cutting at $\chi^2 < 8$ for events with 1 tracks (occupancy 0.17%).

In the case where only one track is generated for each event (occupancy 0.17%), Figure 7.4 shows the $\chi^2$ distribution for "CPU fitter" and the "Artificial Retina". The distribution are the same offering a cross-check of the two implementations. Table 7.2 reports the efficiency and the ghost rate. The difference in efficiency for the two approach is negligible and amenable to some approximation of the "Artificial Retina".

In the case where four track are generated for each event (occupancy 0.7%), the $\chi^2$ distributions are not the same (Figure 7.5). Figure 7.6 shows that the $\chi^2$ distribution of track reconstructed with the "Artificial Retina" has a long tail. We know that the tail is composed by real tracks and ghost, and is related to how the "Artificial Retina" chose the set of hits. Table 7.3 show the efficiency and the ghost rate of the "Artificial Retina" with different cut on $\chi^2$, and compare these results with the efficiency and the ghost rate of the "CPU fitter". A tight cut on $\chi^2$ strongly reduce the "Artificial Retina" efficiency. A possible solution is that the Engine may store more hits per layer, and then calculating the $\chi^2$ over all the combinations of these hit, though this solution requires a larger amount of logic. However our primary goal is having a system that runs at an event rate close to 40 MHz, so this solution was not tested.

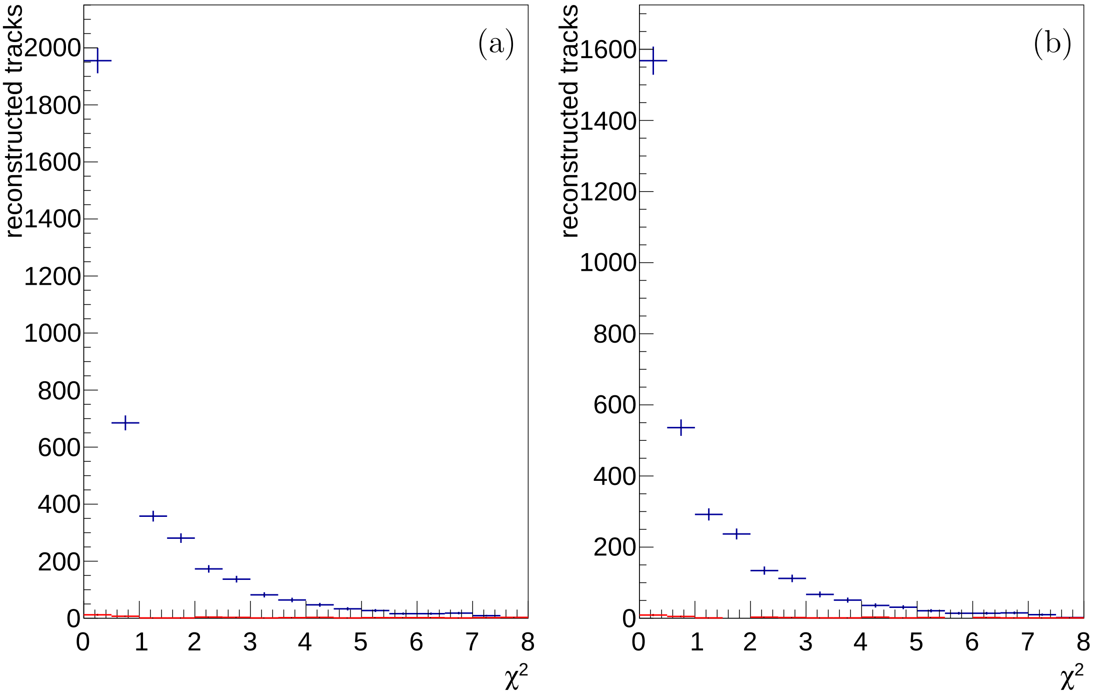

Figure 7.5: $\chi^2$ distribution of reconstructed tracks in events with 4 track generated of each event (occupancy 0.7%). Tracks reconstructed via "CPU fitter" (a) and "Artificial Retina" (b). Track matched in blue, ghost tracks in red.

| | $\chi^2$ | $\varepsilon$ | ghost rate |
|---|---|---|---|
| "CPU fitter" | $< 8$ | 97.5% | 1.4% |
| "Artificial Retina" | $< 8$ | 78.5% | 1.0% |
| "Artificial Retina" | $< 1100$ | 90.0% | 12% |
| "Artificial Retina" | no cut | 98.5% | 26% |

Table 7.3: Efficiency and ghost rate cutting at different $\chi^2$ for events with 4 tracks (occupancy 0.7%).

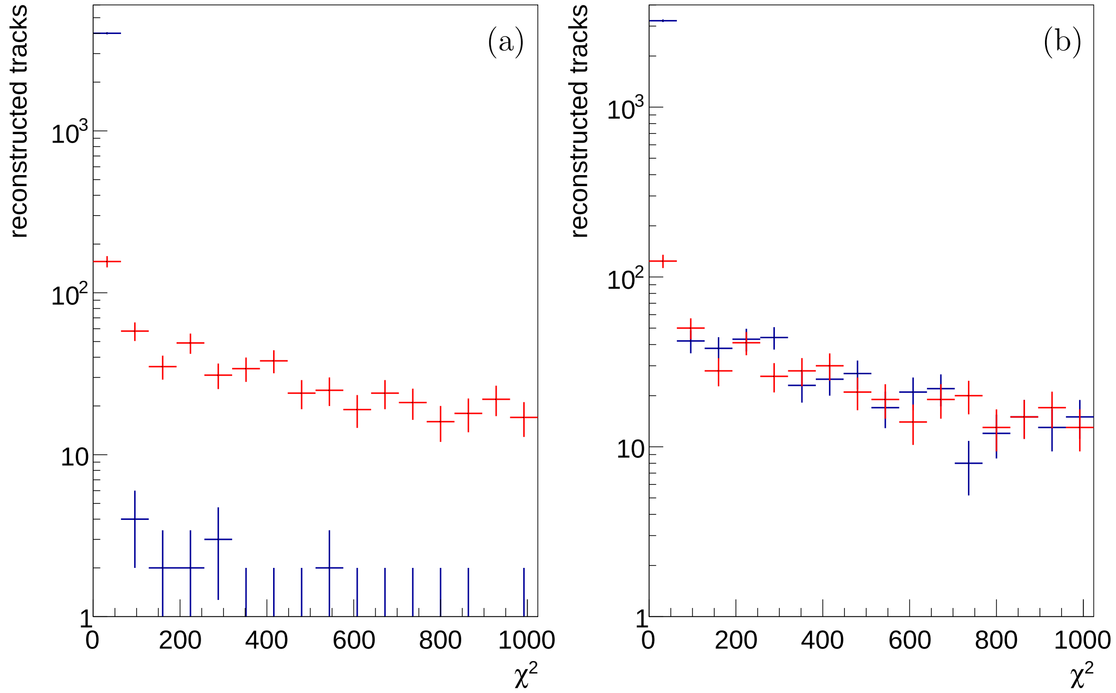

Figure 7.6: Tails in $\chi^2$ distribution of reconstructed tracks in events with 4 track generated of each event (occupancy 0.7%). Tracks reconstructed via "CPU fitter" (a) and "Artificial Retina" (b). Track matched in blue, ghost tracks in red.

Another comparison that is possible to perform is on the event rate of the two approach. "CPU fitter" is optimized for achieve the maximum speed: when I wrote this software I avoided to repeat operations in the loop cycle, I used shift registers instead of division, and enabled compiler optimizations. Furthermore "CPU fitter" was run on a Intel i7-6850K, one of the fastest processor in workstation class. Figure 7.7 shows the event rate as a function of the number of tracks per event. We notice that the scaling of the two approach are very different. At low occupancy "CPU fitter" is faster, instead the "Artificial Retina" is faster at high occupancy. This because the event rate of "CPU fitter" is inversely proportional to the number of combinations of hits, these increase with the number of tracks to the third power. The "Artificial Retina" approach permits to elaborate one hit per clock cycle, then the event rate is inversely proportional to the number of tracks. Fitting the event rate with the function:

$$f(x) = \mathrm{A} \cdot x^{\mathrm{E}}$$

the exponent measured is $-2.52$ for "CPU fitter" and $-0.69$ for the "Artificial Retina".

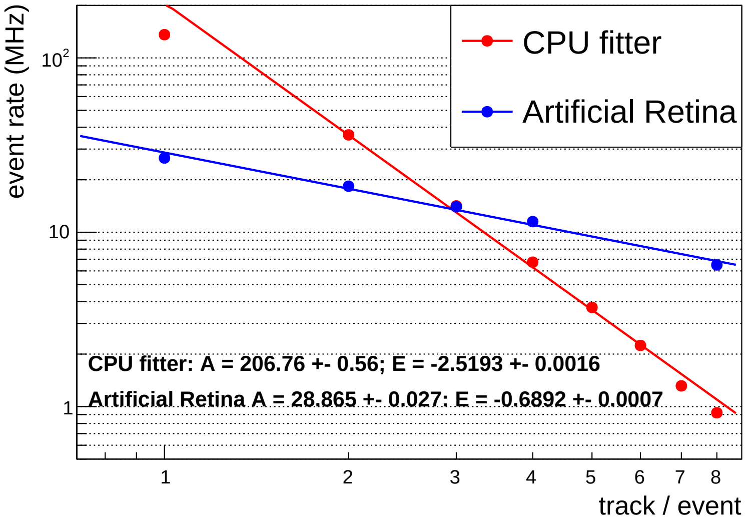

Figure 7.7: Event rate comparison between the “Artificial Retina” and “CPU fitter” running on a i7-6850K.

## 7.5 Final Remarks

In this chapter we have explored the potential of the “Artificial Retina” approach from the point of view of tracking performance. We understood that the results are significantly dependent on the precise operating conditions, and therefore need to be analyzed in detail for each specific case. However, some conclusions of general validity are:

- when compared with traditional CPUs, the “Artificial Retina” system seems to be advantageous for track throughput at equivalent power consumption, every time the track multiplicity is larger than very few tracks, and the advantage grows rapidly. This has been obtained in a setting that is quite unfavorable to the retina .
- The “Artificial Retina” provides however a good track purity only when the occupancy of the detector does not exceed $\sim 1\%$.
- The track purity can be improved by a second stage fitting, but only in a limited way, unless a larger amount of logic is devoted to solving the combinatorial than we have done in our simple tests.

It goes without saying that the latency performance of the “Artificial Retina” is way better than any other system, so it is definitely the only viable solution when tight latency constraints are present.

We conclude that the system we have been developing is highly likely to be a good solution for implementation of the LHCb Downstream Tracker; however, given the dependence on implementation details, a more specific study directly addressing the detector configuration of our choice is important in order to gain a greater level of confidence in its feasibility. This is the subject of the following chapter.

# Chapter 8

# Study of Tracking Performance in a Real Detector

In Chapter 7 I demonstrated that the tracking performances of the “Artificial Retina” approach depend significantly on the occupancy of the detector. Therefore, to determine realistic performance we must apply the system on a real case. The LHCb Upgrade SciFi Tracker has a geometry very similar to the one used for testing the functional prototype, so it is convenient to use this detector for our study. This application would be also very useful for LHCb Upgrade itself because, as described in 3.5, the reconstruction of tracks without external seed in this detector is not currently planned.

## 8.1 Application to Scintillating Fibre Tracker of the LHCb Upgrade

In this study I plan to reconstruct only the tracks projection onto $xz$ plane, described by the parameter $U$ and $V$ (Section 4.2) and only in a quadrant of the SciFi Tracker. As shown below, the quadrants can be considered independent to the reconstruction. To verify that I generated with the official LHCb simulation a set of minimum-bias events, and Figure 8.1 shows the track distribution over the phase space projected to $xz$ and $yz$ planes. Tracks that change quadrant when passing through the detector populate the regions where $x$ (or $y$) on layer 1 and 6 have opposite sign. From the distribution I estimate that the fraction of tracks that move from the upper quadrants to the lower quadrants (or vice versa) is only 1.4%, while tracks moving from the right quadrant to the left (or vice versa) are 4.3% of the total.

From 8.1 is also possible to see that tracks populate only a specific band of the phase-space plane. Furthermore the tracks are not uniformly distributed over the phase space, due to the forward detector geometry and the topology of physics events. Since system performances depend on the occupancy, we may have regions with different efficiency and ghost rate, and it may be not efficient to optimize the system configuration for the region with the highest occupancy. To avoid this

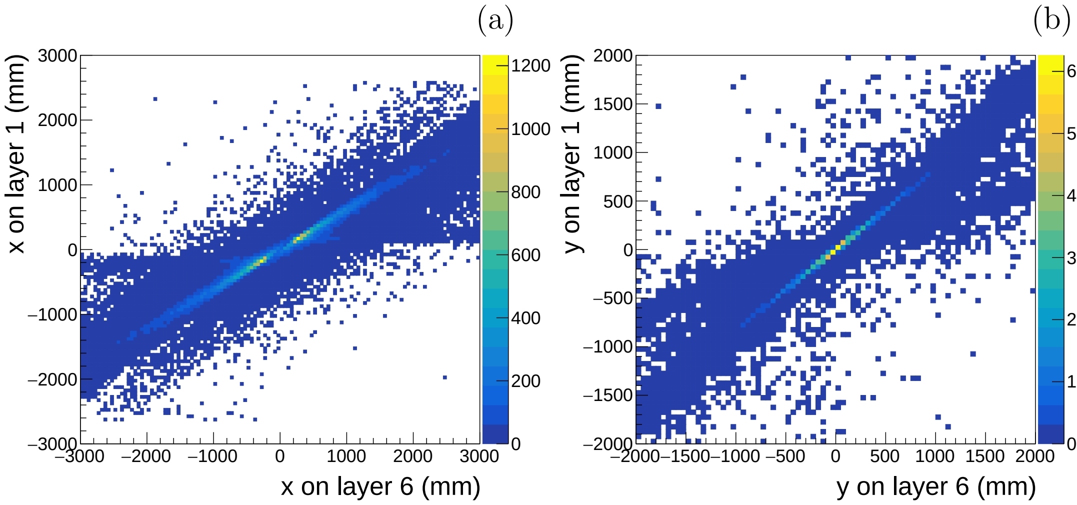

Figure 8.1: Tracks distribution over the phase space projection to $x - z$ (a) and $y - z$ (b) planes.

undesirable effect, we must uniformly distribute hits. This can be achieved defining a hit coordinate transformation from the real space of the detector to a transformed space. Detail of this transformation are described in Section 8.1.1.

Finally, to determine the maximum number of cell that can be implemented, I consider the same design proposed for integrating the Downstream Tracker into LHCb Upgrade EB, described in Section 4.1.2. Assuming to have one device for each EB node, 36 of them are available for each SciFi quadrant. Using FPGAs with 1M logic elements like the device in the DN0237 board. we can implement 588 engines in a single chip, for a total of $\sim 20,000$ cells. Dividing the transformed phase space in a grid of $158 \times 194$ cells, I cover the relevant band with 106 $14 \times 14$ Engine matrix, for a total of 20,776 Engine.

### 8.1.1 Load Balancing

To avoid significant variation in efficiency and ghost rate when tracks are not uniformly distributed, coordinates in the "Artificial Retina" phase space can be transformed with respect to the real one in the detector.

If we indicate with $(x_i, z_i)$ the coordinates of the track intersection with the detector layer $i$ placed at $z = z_i$, we introduce the hit distribution $f(x_i)$ on the $i$ plane. Figure 8.2 shows the hit distribution on plane 1 and 6. We fit the distribution with the function:

$$f(x_i) = \begin{cases} p_0 & \text{if } x_i < x_{i,min} \\ \frac{1}{p_1 \cdot x_i + p_2} & \text{if } x_i > x_{i,min} \end{cases} \tag{8.1}$$

where $p_0$, $p_1$, $p_2$ are parameters. The $f(x_i)$ is a discontinue function due to the fact that SciFi layers have a hole around the beam pipe and the distribution of hit are

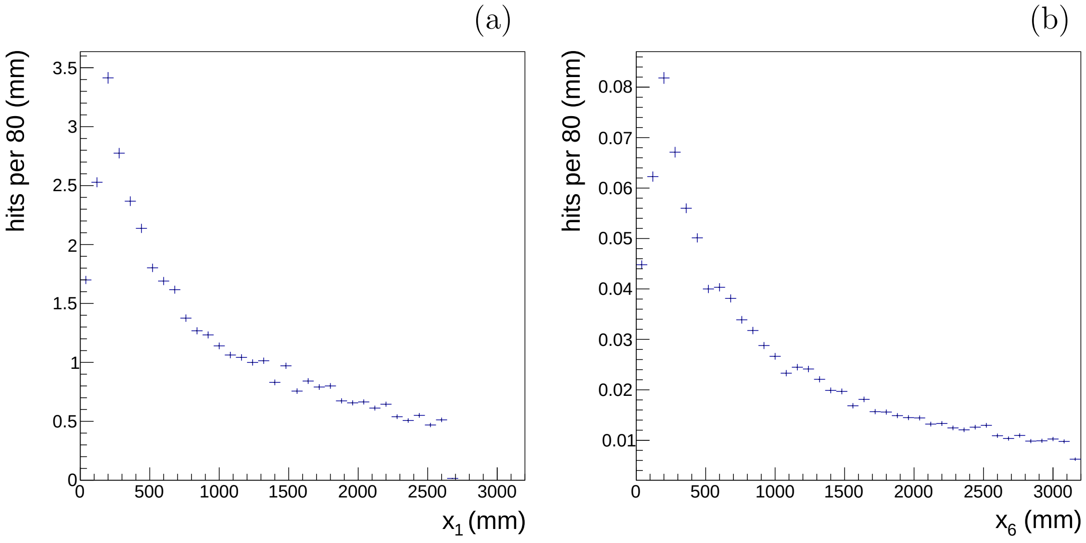

Figure 8.2: Hit distribution on plane 1 (a) and 6 (b).

discontinue at $x_i = 13cm$. Then we define the following coordinate transformation to obtain an uniform hit distribution $h(x_i)$:

$$h(x_i) = \begin{cases} N_1 \cdot \frac{x_i}{x_{i,min}} & \text{if } x_i < x_{i,min} \\ N_1 + \frac{1}{N_2} log \frac{p_1 \cdot x_i + p_2}{p_1 \cdot x_{i,min} + p_2} & \text{if } x_i > x_{i,min} \end{cases}$$

where $N_1$ and $N_2$ are normalization factors defined as follows:

$$N_2 = log \frac{p_1 \cdot x_{i,max} + p_2}{p_1 \cdot x_{i,min} + p_2}$$

$$N_1 = \frac{p_0}{N_2}$$

Because the hit distribution is not too different among the various layers, as shown in Figure 8.2, we use a single set of parameters for the transformation obtained fitting the sum of the distribution for all the layers. Figures 8.3 and 8.4 shows the hits and tracks distribution in the transformed coordinates. The tracks distribution shows clearly how the transformation made the track density more uniform along the highly-populated diagonal band.

## 8.2 Performance study with tracks from basic simulation

From the MC-simulated sample described before, we established that, on average, 50 tracks per quadrant cross the detector. Therefore, I process events with similar

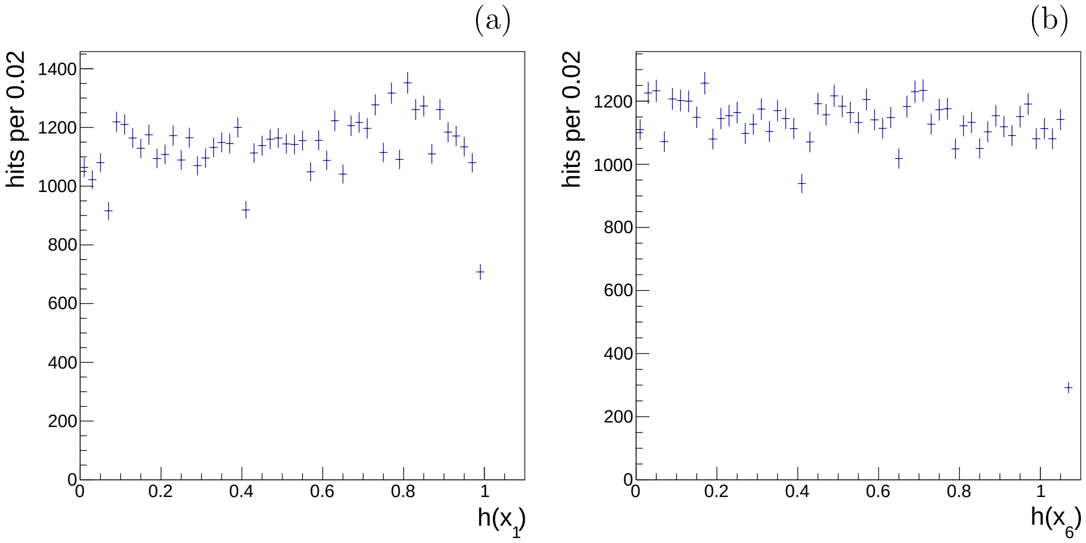

Figure 8.3: Transformed hit distribution on plane 1 (a) and 6 (b).

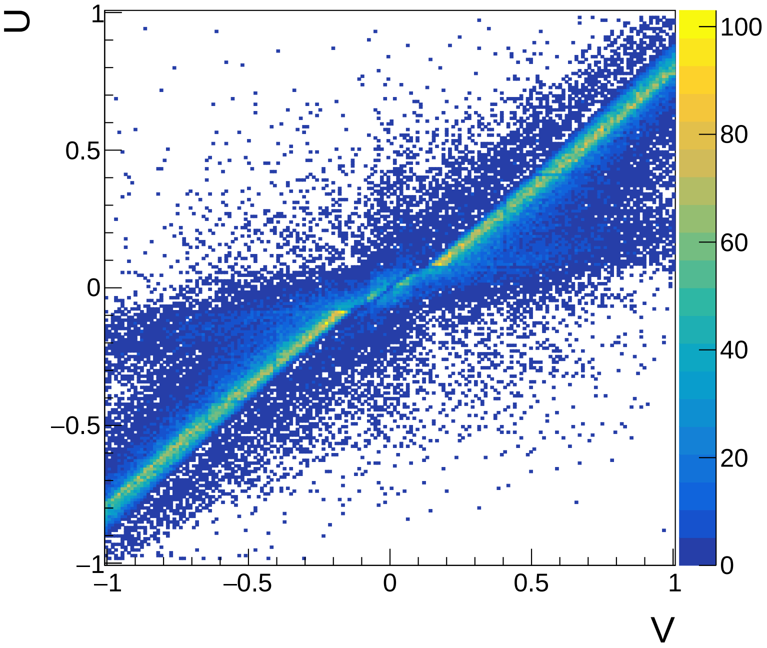

Figure 8.4: Tracks distribution in the transformed phase space.

occupancy using the our software simulation configured as the “Artificial Retina” system described above.

First, using the “Detector Simulator” I generated events with 50 tracks uniformly distributed inside the covered area of the transformed phase space. Figure 8.6 shows the excitation level of the Engines for one event and the position of the corresponding 50 generated tracks. Figure 8.7 shows the same distribution after applying a threshold on the doublets, as mentioned in Section 4.2.3. Because the Engines adds the hits using three separate accumulator, applying a threshold on them rejects false local maxima produced by hits from different tracks. In Figure 8.8 I have also added the position of tracks reconstructed with the centroid technique (Section 4.2.4) using the Engines with excitation level above the the global threshold. I optimized the global threshold looking at the distribution of excitation level for reconstructed tracks, shown in Figure 8.5, and found a value of 1100. The efficiency and ghost rate of this configuration are:

$$\varepsilon = 95.2\% \quad Ghost = 83.1\%$$

One limitation of the “Artificial Retina” architecture is the inability to reconstruct tracks in adjacent cells. Indeed by definition only one cell can be a local maximum, if two tracks are generated in adjacent cells, then the “Artificial Retina” reconstructs only one track. However it is not a problem if the reconstructed tracks are used as a seed for a next tracking stage or later was performed a fit. This inefficiency can be calculated from the simulation and it depends on the occupancy. I implemented a second matching algorithm that searches, for each reconstructed track, if there is a generated track within a maximum distance. We call the efficiency calculated with the second matching algorithm $\varepsilon_2$, and for this configuration it is:

$$\varepsilon_2 = 96.3\%$$

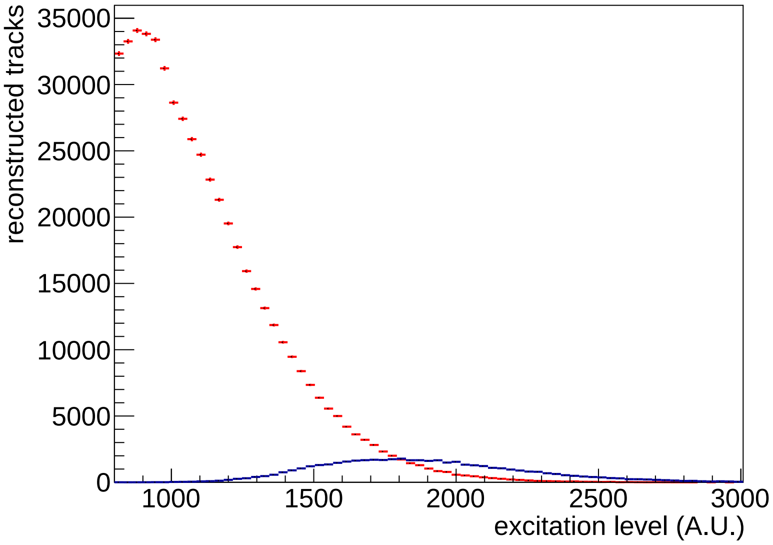

Figure 8.5: Distribution of reconstructed tracks excitation level. Track matched in blue, ghost tracks in red.

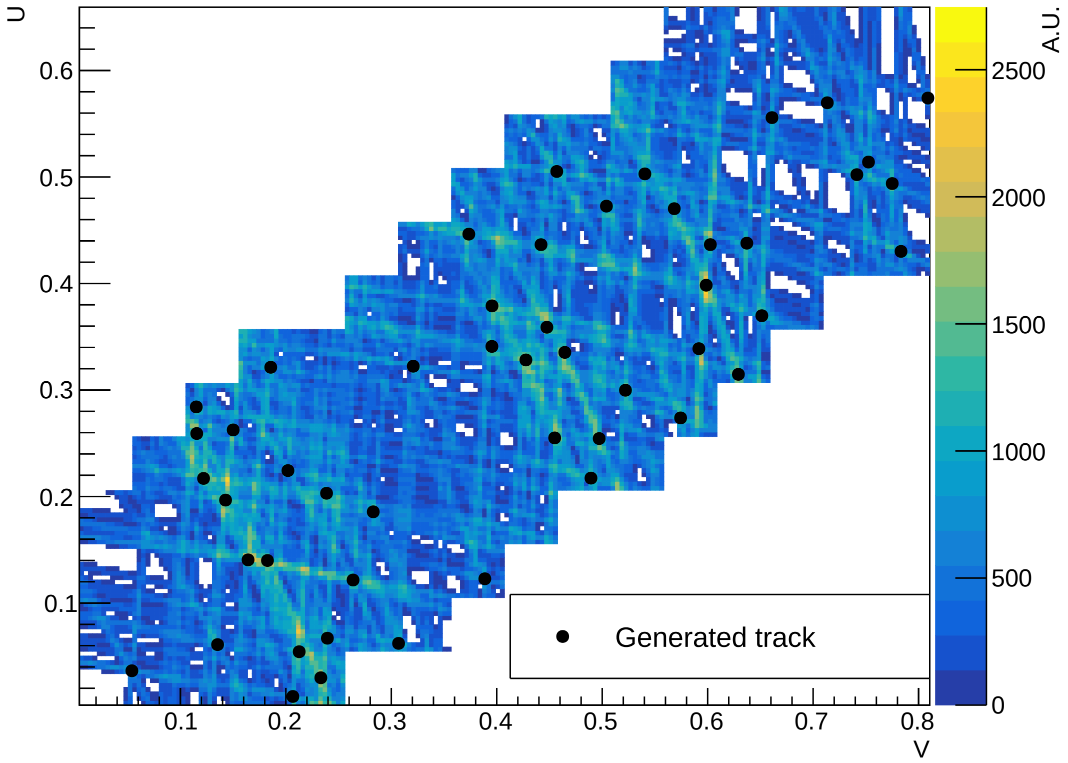

Figure 8.6: Engine excitation level distribution and generated tracks for a event.

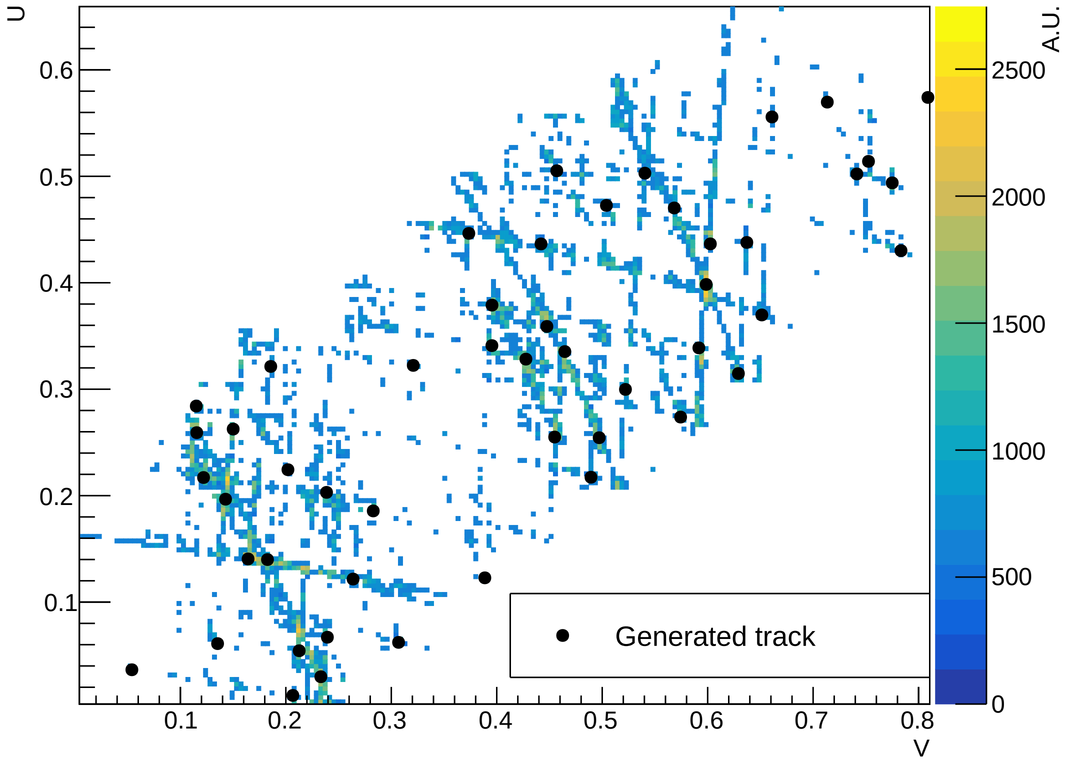

Figure 8.7: Engine excitation level distribution after that a threshold on the doublet was apply, and generated tracks for an event.

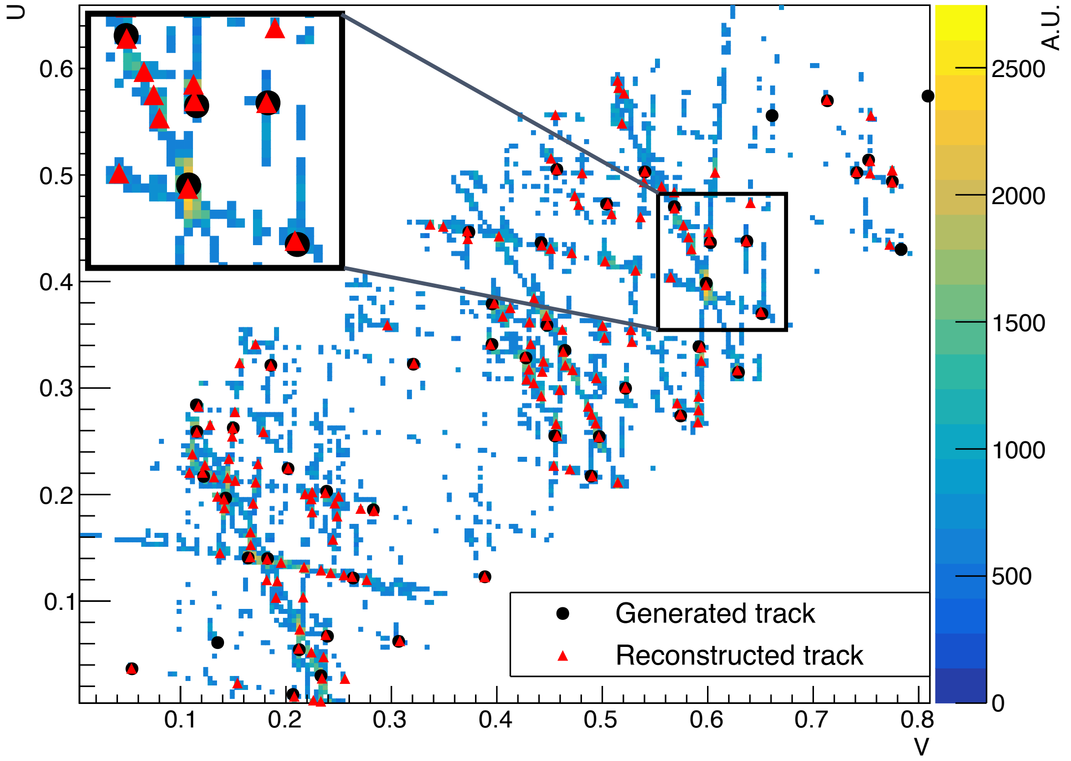

Figure 8.8: Engine excitation level distribution after that a threshold on the doublet was apply, generated tracks, and reconstructed tracks for an event..

Then, instead of the centroid technique, I cut on $\chi^2$ computed from the linearized $\chi^2$ fit, in order to reduce the ghost rate. In this case we must consider the effects of detector efficiency. Because the SciFi Tracker modules efficiency is 97.5%, the probability that the SciFi Tracker detects all the hits of the track is:

$$P(n_{hit}) = \binom{n_{layer}}{n_{hit}} {\varepsilon_{det}}^{n_{hit}} (1 - \varepsilon_{det})^{n_{layer} - n_{hit}}$$

$$P(n_{hit} = 6) = {\varepsilon_{det}}^{n_{hit}} = 0.975^6 = 85.9\%$$

with $n_{layer} = 6$, while the probability that 5 or more hits are detected is

$$P(n_{hit} \geq 5) = 99.1\%.$$

For this reason we require that a track must be reconstructed using hits from at least five layers. Figure 8.9 shows the $\chi^2$ distribution for reconstructed tracks with 6 and 5 hits. For tracks with 6 hits, the ghost rate is $\sim 37\%$, so we a cut on $\chi^2$ is not necessary. Instead for tracks with 5 hits, the rate is very high ($\sim 96\%$), then I need to apply a cut on $\chi^2$. Rejecting 5-hit tracks with $\chi^2$ higher than 1.5 and keeping all the 6-hit tracks, the effciencies and ghost rate are:

$$\varepsilon = 93.2\% \quad \varepsilon_2 = 94.3\% \quad Ghost = 49.1\%$$

This demonstrate that cutting on the $\chi^2$ we obtain much better performances with respect to the centroid technique. These results are satisfactory, therefore we move to study the system performances when tracks are fully simulated with the official software of LHCb, as described below.

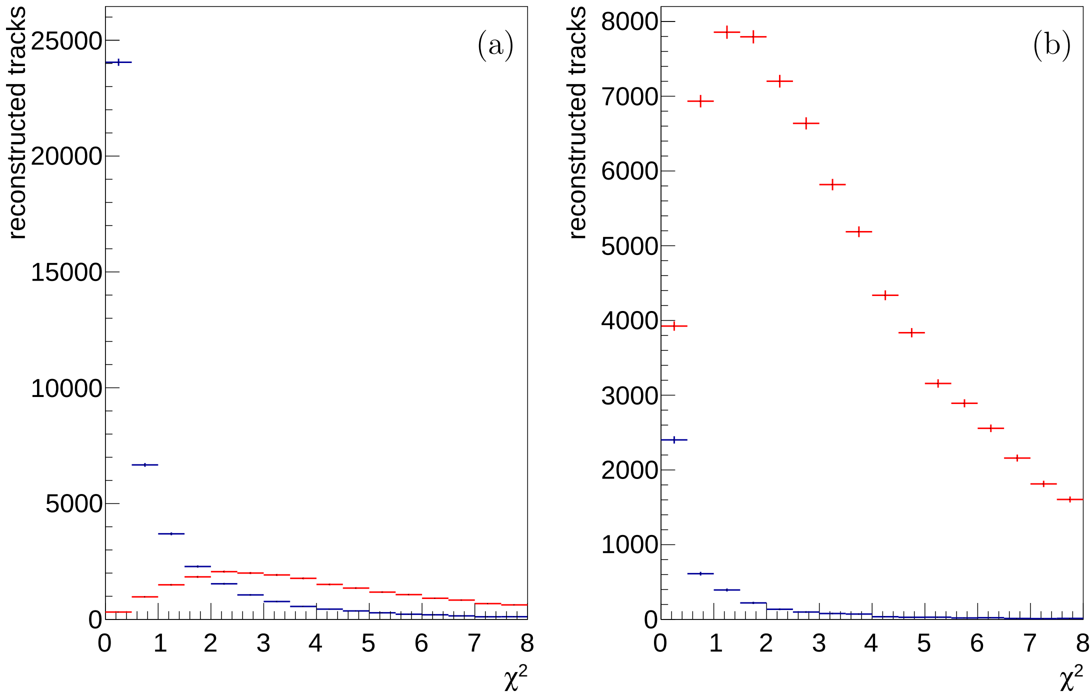

Figure 8.9: $\chi^2$ distribution for tracks reconstructed with hits on 6 layers (a) and 5 layers (b). Track matched in blue, ghost tracks in red.

## 8.3 Performance study with tracks from LHCb simulation

### 8.3.1 Interfacing with LHCb simulation

For more accurate results, I implemented an interface between the "Artificial Retina" simulation and the official LHCb simulation (Figure 8.10). First, I configured the official LHCb simulation to include magnetic fringe field, multiple scattering, secondary particles (*e.g.* delta rays), and hits clusterization. Then I add functionalities to the "Artificial Retina" utility software in order to pass tracks from our software to LHCb simulation, and SciFi hits from LHCb simulation to our software. In this way the LHCb simulation can replace the functionality of the Detector-Simulator: receptor or hits from specific tracks can be created by the LHCb simulation, and physics events fully simulated can be passed to the "Artificial Retina" simulator.

I had to slightly change the configuration of the "Artificial Retina" system, because the detector geometry in the LHCb simulation was updated with slightly shifted layers. Therefore I remapped the phase space, dividing the transformed phase space using a grid 182 × 194 cells, and covering the interested zone with 125 14 × 14 Engine matrix, for a total of 20,721 Engine. The total number of the engines is very similar to the previous configuration, even if the number of matrices is higher, due to having for some of those matrices only a fraction of the engines activated.

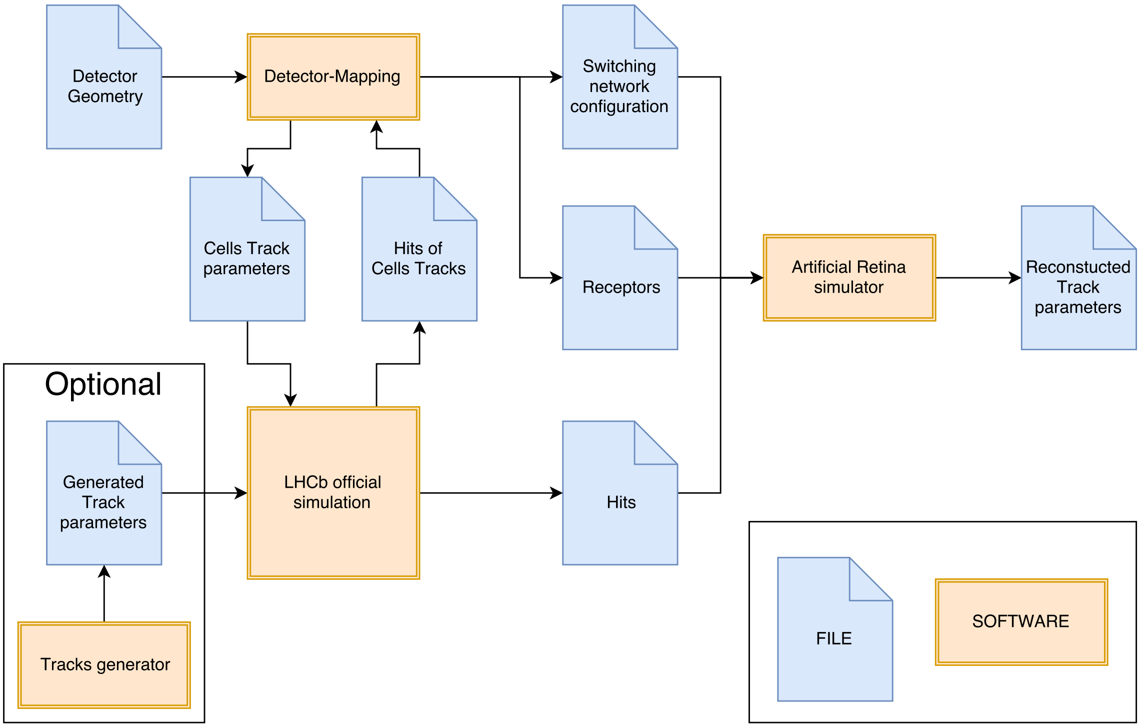

Figure 8.10: Integration of "Artificial Retina" software suite with LHCb official simulation.

### 8.3.2 Results

First I performed a test using events with 50 tracks distributed uniformly in the covered area of the transformed phase space. The same momentum was assigned to every track in the event. The purpose of this test is to evaluate the effects of the fringe magnetic field and of the multiple scattering on the efficiency. The tracks are generated with our track generator and simulated with the LHCb simulation. For this test I did not apply any cut on the $\chi^2$. Figure 8.11 shows the efficiency as a function of the track momentum. Comparing these results with previous ones, we can state that the fringe magnetic field and the multiple scattering do not have a significant effect on performances for tracks with $p > 5\,\mathrm{GeV}/c$. The efficiency for tracks with $p > 3\,\mathrm{GeV}/c$ is still greater than 90%, an acceptable value for LHCb reconstruction.

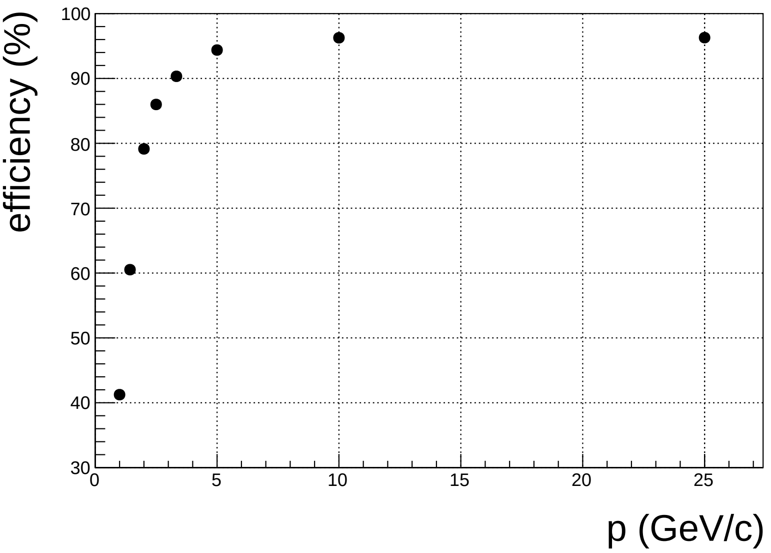

Figure 8.11: Efficiency in function of tracks momentum. 50 tracks with same momentum per event.

An additional test was done using events where tracks are generated by a *pp* collision. I produced a MC sample of *pp* collisions where the hadronization is forced to produce at least one $D^*$ meson that decays into $D^{*+} \to D^0[K^0_S(\pi^+\pi^-)\pi^+\pi^-]\pi^+$. I chose this channel because contains a $K^0_S$ and is used for estimation of tracking performance of LHCb Upgrade. The efficiency was calculated considering all the tracks inside the phase-space area covered by the "Artificial Retina" system and using the same method used for the test described above in this section. Figure 8.12 shows the efficiency as a function of the track momentum. This figure shows a trend similar to Figure 8.11, but the maximum value at high momentum is lower. This behavior could be ascribed to a residual not uniformity of the tracks in the phase space. Figure 8.13 shows the Engines excitation level for one event, but we can also noticed how tracks accumulates around the diagonal band already seen in Figure 8.4.

Finally I compared the "Artificial Retina" with the software algorithm under developing for the LHCb Upgrade 1a.
This software algorithm is called "Hybrid Seeding" [75] and it iterates several steps adding every time more information. In the first step the "hybrid seeding" finds tracks in the *xz* plane using only the hits from *x*-layers. Starting from a pair of hits, one from the first and one from the third station, the algorithm searches for another hit in the second station. Hits from other axial layers are added to every 2-hit combination if compatible with momentum hypothesis. An intermediate clone removal step is applied to the *xz* track projections.
The second step consists of a Hough-like transformation on the stereo hits used to identify potential line candidates as *yz* projections associated to the *xz* projection of the track. The hits used by the track candidates are flagged and they become

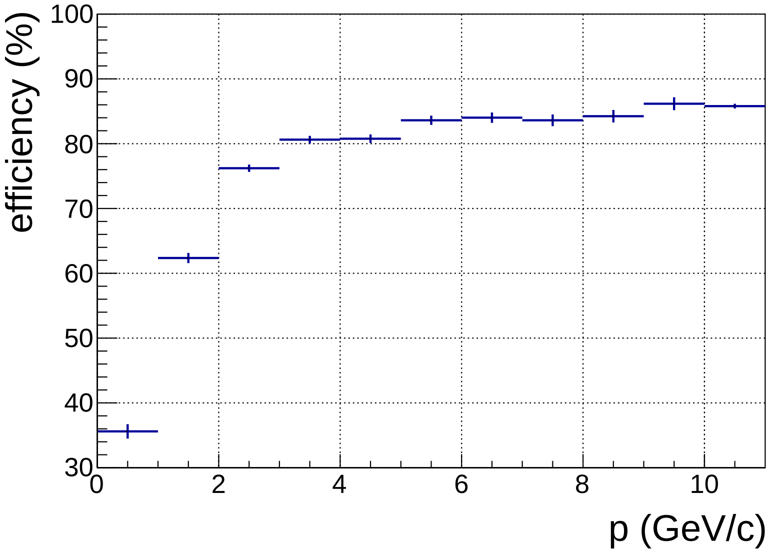

Figure 8.12: Efficiency in function of track momentum in $D^{*+} \to D^0[K^0_S(\pi^+\pi^-)\pi^+\pi^-]\pi^+$ decay with underlying event.

unavailable for the next track searches. These two steps are repeated assuming $p > 5\,\mathrm{GeV}/c$, $p > 2\,\mathrm{GeV}/c$, and $p > 1.5\,\mathrm{GeV}/c$. Once all the momentum hypothesis have been processed, a global clone removal is also applied.
Considering the information used by my implementation of "Artificial Retina", I compared my results with the "Hybrid Seeding" efficiency and ghost rate after the first step with $p > 5\,\mathrm{GeV}/c$, because my system is not using any information from stereo layers.

For the comparison with hybrid seeding I tried to obtain the same efficiency with "Artificial Retina", removing tracks with less than 6 hits and requiring a $\chi^2$ smaller than 16. Table 8.1 shows the efficiency and ghost rate for both tracking implementations. The "Artificial Retina" ghost rate is higher than the "Hybrid Seeding", but not unbearable, at least at this early stage of the processing. The "Hybrid Seeding" demonstrates that including the information from stereo layers, the rate decreases significantly, and this is expected also for the "Artificial Retina" system. Anyway we have to remember that the primary goal of this approach is not a "perfect" tracking reconstruction, but having a feasible system that runs at an event rate close to 40 MHz. To run the current version of "Hybrid Seeding" at this rate, we would need 2 additional HLT farms, because the time needed for "Hybrid Seeding" is around 25-30 ms per event [75], about twice the total processing time per event (13 ms) [59].

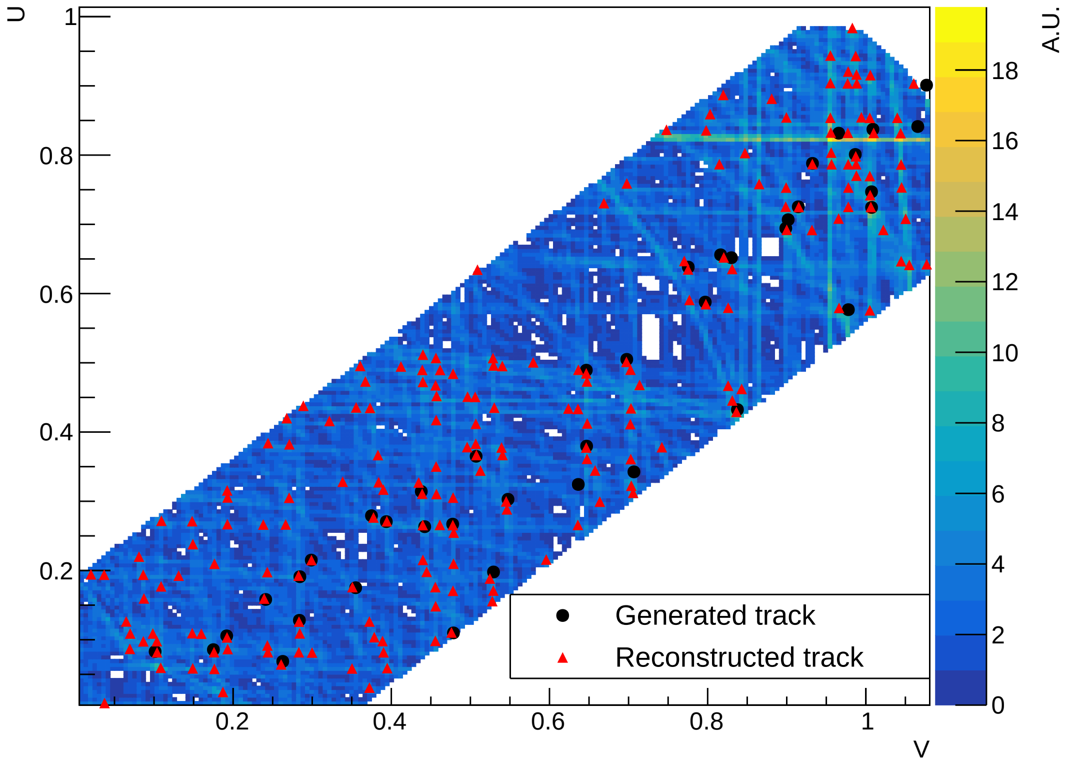

Figure 8.13: Engine excitation level distribution with generated and reconstructed tracks for a $D^{*+} \to D^0[K^0_S(\pi^+\pi^-)\pi^+\pi^-]\pi^+$ decay with underlying event.

| | $\varepsilon$ | Ghost (using $x$-layers only) |
|---|---|---|
| "Hybrid Seeding" | 66.6% | $\sim$ 50% |
| "Artificial Retina" | 68.2% | 76% |

Table 8.1: Efficiency and ghost rate achieved by the "Hybrid Seeding" and the "Artificial Retina".

## 8.4 Final remarks

I demonstrated that applying the "Artificial Retina" approach to the SciFi Tracker, while integrating the electronic boards inside the EB nodes, is feasible. I studied the tracking performances in different cases, using also data from the LHCb official simulation. While the tracking results compared with the software algorithm "Hybrid Seeding" show a higher ghost rate, the "Artificial Retina" approach remains more competitive when the reconstruction has to be implemented in real time at LHCb Upgrade event rate. The results mentioned above are a significant step towards the implementation of an "Artificial Retina" system as Downstream Tracker.

# Chapter 9

# Over all conclusion

Current and future experiments on $b$- and $c$-physics have the potential to significantly improve our knowledge on CKM. However, because of a small signal-to-background ratio for typical interesting processes, and the limited bandwidth available for storing data, the adoption of powerful and very selective trigger systems is needed, particularly at hadron colliders. The most important discriminant for decay of $b$- and $c$-hadrons is their relatively long lifetime, that requires excellent tracking systems to discriminate interesting events from the huge background.

The LHCb experiment will adopt a full software trigger running on a large PC farm, to reconstruct all tracks produced in every LHC collision, occurring at a rate of 40 MHz. On account of the significant CPU time required, it is not planned to perform the reconstruction of particles generated outside of the vertex detector ("downstream tracks"). While this covers most of the decays of $b$- and $c$-particles, not having access to this information limits efficiency for decay modes containing neutral hadrons and long lived particles ($K^0_\mathrm{S}$ and $\Lambda$). This includes many interesting decays like $D^0 \to K^0_\mathrm{S} K^0_\mathrm{S}$ or $\Lambda^0_b \to 3\Lambda$.

In this thesis I have performed a feasibility study of real-time reconstruction for downstream tracks at the earliest trigger level (the Event Builder), using a FPGA-based system organized according to the innovative "Artificial Retina" architecture. This system would allow to extend the reconstruction to the downstream part of the tracker and to handle even higher beam intensities.

In order to demonstrate that a system based on "Artificial Retina" can sustain the event rate at Level-1 of the LHC, I have produced a new implementation of the Retina system on current FPGAs (Stratix-V). I have also re-designed several parts of the previous existing firmware, both the switching network and the cell processors, to optimize performance and speed, using low-level hardware description languages (VHDL). This also included re-design the system interface for using on a completely different board, a special custom-order board aimed at the development and test of new fast ASICs projects.

Testing the system with realistic simulated events in a "general-purpose" 6-layer tracking detector, I debugged and measured the throughput of the new system as a function of the occupancy of the tracker. In this way, I managed to produce in the lab a hardware prototype capable of processing events at a limit event rate of 66.67 MHz (greater than the LHC event rate of $\sim 30$ MHz). This had never been attained before without the help of some form of time-multiplexing to reduce the rate. All currently existing designs require tens of µs, but in order to work as a transparent device incorporated within the Event Builder, it is necessary that the latency of the device be limited to very few µs. After performing an optimization of the internal pipeline of the device, I managed to achieve a latency smaller than $< 0.5$ µs, well below than the value of few µs required for the device to be incorporated within the Event Builder.

As next step, I performed a higher-level study of the efficiency, ghost rate, and event rate of this system when applied to a generic bare-bone detector. In these condition, compared with a software algorithm, the Retina showed to have similar efficiency, but event rate higher by 1-2 orders of magnitude when events have 10-20 tracks. To keep the ghost rate under control, I studied possible optimizations of the system, introducing requirements on the track $\chi^2$ computed with a linearized fit. Modern FPGA have a large number of digital signal processors (DSP) capable of floating point operation suitable for the task, and my work evidenced the necessity of adding a DSP stage to the final system.

Finally, I proceeded to study an application to the real configuration of the LHCb Scintillating Fiber detector and compare it with the performance of the traditional CPU-based reconstruction software. Given the complexity of the system, I started reconstructing tracks projected onto a 2D plane. I performed a preliminary study based on a home-made event generator, that did not include multiple scattering and the fringe magnetic field. Then I moved to a more complete study based on the actual official simulation of the LHCb detector, interfaced through a custom piece of software to my own code. Both have been performed with realistic track occupancy, as expected in the new beam conditions in the upcoming physics run of the LHC, using an amount of hardware contained within the limits of the LHCb Event Builder.

For the preliminary study I obtained an efficiency close to 95% and a ghost rate of about 50%. This ghost rate value may seem high, but it is mostly due to have used only a portion of the information available from tracking layers (only from axial ones). Regarding the application to events with decays of interesting benchmark modes ($D^{*+} \to D^0[K^0_S(\pi^+\pi^-)\pi^+\pi^-]\pi^+$), efficiency was comparable with the values obtained by a LHCb software algorithm developed for the Upgrade, while ghost rate was higher. Anyway, given the slowness of the software approach, I found that the "Artificial Retina" approach remains much more competitive than current software solutions when the reconstruction has to be implemented in real time at LHCb Upgrade event rate.

In conclusion, my work demonstrates that a special-purpose processor based on the "Artificial Retina" approach can be built at a reasonable cost using FPGA devices. This is a significant step towards real-time tracking at HL-LHC, a methodology that will also open the possibility to trigger purely on long-lived neutrals, increasing significantly the acceptance for some channels and expanding our Physics reach. Encouraged by these results longer and more extensive studies are needed including all layers of the full 3D detector.

# Bibliography

[1] LHCb collaboration, *Expression of Interest for a Phase-II LHCb Upgrade: Opportunities in flavour physics, and beyond, in the HL-LHC era*, CERN-LHCC-2017-003.

[2] L. Ristori, *An artificial retina for fast track finding*, Nucl. Instrum. Meth. **A453** (2000) 425.

[3] J. H. Christenson, J. W. Cronin *et al.*, *Evidence for the 2$\pi$ decay of the $K_2^0$ meson*, Phys. Rev. Lett. **13** (1964), no. 4 138.

[4] A. D. Sakharov, *Violation of CP invariance, C asymmetry, and baryon asymmetry of the Universe*, Pisma Zh. Exp. Theor. Fiz. **5** (1967) 32, English translation in JETP Lett. 5, 24 (1967), reprinted in Sov. Phys. Usp. 34, 392 (1991).

[5] NA48 collaboration, *A new measurement of direct CP violation in two pion decays of the neutral kaon*, Phys. Rev. Lett. **B465** (1999), no. 335.

[6] KTeV collaboration, *Observation of direct CP violation in $K_{S,L} \rightarrow \pi\pi$ decays*, Phys. Rev. Lett. **83** (1999), no. 22.

[7] L. Wolfenstein, *Violation of CP invariance and the possibility of very weak interaction*, Phys. Rev. Lett. **13** (1964), no. 18 562.

[8] BABAR collaboration, *Observation of CP violation in the $B^0$ meson system*, Phys. Rev. Lett. **87** (2001), no. 9:091801.

[9] BELLE collaboration, *Observation of large CP violation in the neutral B meson system*, Phys. Rev. Lett. **87** (2001), no. 9:091802.

[10] N. Cabibbo, *Unitary symmetry and leptonic decays*, Phys. Rev. Lett. **10** (1963), no. 12 531.

[11] M. Kobayashi and T. Maskawa, *CP-Violation in the Renormalizable Theory of Weak Interaction*, Prog. Theor. Phys. **49** (1973), no. 2 652.

[12] C. Jarlskog, *Commutator of the Quark Mass Matrices in the Standard Electroweak Model and a Measure of Maximal CP Nonconservation*, Phys. Rev. Lett. **55** (1985), no. 10 1039.

[13] C. Jarlskog, *A basis independent formulation of the connection between quark mass matrices, CP violation and experiment*, Z. Phys. C **29** (1985) 491.

[14] I. Dunietz, O. W. Greenberg, and D.-D. Wu, *A priori definition of maximal CP nonconservation*, Phys. Rev. Lett. **55** (1985), no. 27 2935.

[15] CKMfitter Group, J. Charles *et al.*, *CP violation and the CKM matrix: assessing the impact of the asymmetric B factories. Updated results and plots available at: http://ckmfitter.in2p3.fr*, The European Physical Journal C - Particles and Fields **41** (2005) 1.

[16] L. Wolfenstein, *Parameterization of the Kobayashi-Maskawa Matrix*, Phys. Rev. Lett. **51** (1983), no. 21 1945.

[17] M. Battaglia, A. J. Buras *et al.*, *The CKM Matrix and the Unitarity Triangle*, arXiv **0304132** (2003).

[18] J. E. Augustin, A. M. Boyarski *et al.*, *Discovery of a Narrow Resonance in $e^+e^-$ Annihilation*, Phys. Rev. Lett. **33** (1974), no. 23 1406.

[19] J. J. Aubert, U. Becker *et al.*, *Experimental Observation of a Heavy Particle $J$*, Phys. Rev. Lett. **33** (1974), no. 23 1404.

[20] S. W. Herb *et al.*, *Observation of a dimuon resonance at 9.5 GeV in 400-GeV proton-nucleus collisions*, Physical Review Letter **39** (1977), no. 5 252.

[21] N. Ellis and A. Kernan, *Heavy quark production at the CERN $p\bar{p}$ collider*, Phys. Rept. **195** (1990) 23.

[22] CDF collaboration, *Measurement of the $B^0\bar{B}^0$ flavor oscillations frequency and study of same side flavor tagging of B mesons in $p\bar{p}$ collisions*, Physical Review D **59:032001** (1999).

[23] CDF collaboration, *Measurement of sin2$\beta$ from $B \to J/\psi K^0_S$ with the CDF detector*, Physical Review D **61:072005** (2000).

[24] Particle Data Group, C. Patrignani *et al.*, *Review of particle physics*, Chin. Phys. **C40** (2016) 100001, and 2017 update.

[25] BaBar collaboration, D. Boutigny *et al.*, *BaBar technical design report*, in *BaBar Technical Design Report EPAC Meeting Stanford, California, March 17-18, 1995*, 1995.

[26] A. Abashian *et al.*, *The Belle Detector*, Nucl. Instrum. Meth. **A479** (2002) 117.

[27] C.-h. Cheng, *Measurements of the CKM Angle beta/phi(1) at B factories*, eConf **C070512** (2007) 010, `arXiv:0707.1192`.

[28] Belle-II collaboration, T. Abe *et al.*, *Belle II Technical Design Report*, arXiv:1011.0352.

[29] A. Achilli *et al.*, *Total and inelastic cross sections at LHC at* $\sqrt{s} = 7$ TeV *and beyond*, Phys. Rev. D **84** (2011) 094009.

[30] LHCb collaboration, Yu. Guz, *Studies of open charm and charmonium production at LHCb*, Nucl. Phys. Proc. Suppl. **207-208** (2010) 355.

[31] CDF collaboration, *Measurement of the* $J/\phi$ *and b-Hadron Production Cross Sections in* $p\bar{p}$ *Collisions at* $\sqrt{s}$ *= 1960 GeV*, Phys. Rev. D **71** (2005), no. 032001.

[32] LHCb collaboration, *Measurement of* $\sigma(pp \to b\bar{b}X)$ *at* $\sqrt{s}$ *= 7 TeV in the forward region*, Phys. Lett. B **694** (2010) 209.

[33] O. S. Brning *et al.*, *LHC Design Report*, CERN Yellow Reports: Monographs, CERN, Geneva, 2004.

[34] LHCb collaboration, A. A. Alves Jr. *et al.*, *The LHCb detector at the LHC*, JINST **3** (2008) S08005.

[35] LHCb collaboration, R. Aaij *et al.*, *LHCb detector performance*, Int. J. Mod. Phys. **A30** (2015) 1530022, arXiv:1412.6352.

[36] A. S. Dighe, I. Dunietz, H. J. Lipkin, and J. L. Rosner, *Angular distributions and lifetime differences in* $B_S \to J/\psi\phi$ *decays*, Physics Letters B **369** (1996), no. 2 144 .

[37] S. Stieberger and T. R. Taylor, *Non-Abelian BornInfeld action and TypeIheterotic duality (I): Heterotic F6 terms at two loops*, Nuclear Physics B **647** (2002), no. 1 49 .

[38] C.-S. Huang, L. Wei, Q.-S. Yan, and S.-H. Zhu, $B_s \to l^+l^-$, Phys. Rev. D **63** (2001) 114021.

[39] R. Aaij *et al.*, *Performance of the LHCb Vertex Locator*, JINST **9** (2014) P09007, arXiv:1405.7808.

[40] R. Arink *et al.*, *Performance of the LHCb Outer Tracker*, JINST **9** (2014) P01002, arXiv:1311.3893.

[41] M. Adinolfi *et al.*, *Performance of the LHCb RICH detector at the LHC*, Eur. Phys. J. **C73** (2013) 2431, arXiv:1211.6759.

[42] A. A. Alves Jr. *et al.*, *Performance of the LHCb muon system*, JINST **8** (2013) P02022, arXiv:1211.1346.

[43] LHCb collaboration, *First Evidence for the Decay* $B_s^0 \to \mu^+\mu^-$, Phys. Rev. Lett. **110** (2013) 021801.

[44] LHCb collaboration, *Observation of $J/\psi p$ Resonances Consistent with Pentaquark States in $\Lambda_b^0 \to J/\psi K^- p$ Decays*, Phys. Rev. Lett. **115** (2015) 072001.

[45] LHCb collaboration, *Tagged time-dependent angular analysis of $B_s^0 \to J/\psi\phi$ decays at LHCb*, LHCb-CONF-2012-002.

[46] LHCb collaboration, R. Aaij *et al.*, *Measurement of the CP-violating phase $\phi_s$ in $\overline{B}_s^0 \to J/\psi\pi^+\pi^-$ decays*, Phys. Lett. **B713** (2012) 378, `arXiv:1204.5675`.

[47] Heavy Flavor Averaging Group, Y. Amhis *et al.*, *Averages of b-hadron, c-hadron, and $\tau$-lepton properties as of summer 2016*, `arXiv:1612.07233`, updated results and plots available at `http://www.slac.stanford.edu/xorg/hflav/`.

[48] LHCb collaboration, *Differential branching fraction and angular analysis of the $B^0 \to K^{*0}\mu^+\mu^-$ decay*, LHCb-CONF-2012-008.

[49] LHCb collaboration, R. Aaij *et al.*, *Measurement of the isospin asymmetry in $B \to K^*\mu^+\mu^-$ decays*, JHEP **07** (2012) 133, `arXiv:1205.3422`.

[50] LHCb collaboration, *First observation of $B^+ \to \pi^+\mu^+\mu^-$*, LHCb-CONF-2012-006.

[51] LHCb collaboration, R. Aaij *et al.*, *Strong constraints on the rare decays $B_s^0 \to \mu^+\mu^-$ and $B^0 \to \mu^+\mu^-$*, Phys. Rev. Lett. **108** (2012) 231801, `arXiv:1203.4493`.

[52] M. Bona *et al.*, *The 2004 UTfit collaboration report on the status of the unitarity triangle in the standard model*, Journal of High Energy Physics **2005** (2005), no. 07 028.

[53] LHCb collaboration, R. Aaij *et al.*, *Evidence for CP violation in time-integrated $D^0 \to h^-h^+$ decay rates*, Phys. Rev. Lett. **108** (2012) 111602, `arXiv:1112.0938`.

[54] LHCb collaboration, *Framework TDR for the LHCb Upgrade: Technical Design Report*, CERN-LHCC-2012-007. LHCb-TDR-012.

[55] R. Aaij *et al.*, *The LHCb trigger and its performance in 2011*, JINST **8** (2013) P04022, `arXiv:1211.3055`.

[56] M. Tobin, *Performance of the LHCb Tracking Detectors*, Tech. Rep. CERN-LHCb-PROC-2013-015, 2013.

[57] LHCb collaboration, *LHCb VELO Upgrade Technical Design Report*, CERN-LHCC-2013-021. LHCb-TDR-013.

[58] LHCb collaboration, *LHCb Tracker Upgrade Technical Design Report*, CERN-LHCC-2014-001. LHCb-TDR-015.

[59] LHCb collaboration, *LHCb Trigger and Online Technical Design Report*, CERN-LHCC-2014-016. LHCb-TDR-016.

[60] Y. Amhis *et al.*, *The Seeding tracking algorithm for a scintillating detector at LHCb*, Tech. Rep. LHCb-PUB-2014-002. CERN-LHCb-PUB-2014-002, CERN, Geneva, Mar, 2014.

[61] M. Vesterinen, A. Davis, and G. Krocker, *Downstream tracking for the LHCb upgrade*, Tech. Rep. LHCb-PUB-2014-007. CERN-LHCb-PUB-2014-007, CERN, Geneva, Jan, 2014.

[62] LHCb collaboration, R. Aaij *et al.*, *Study of $B^0_{(s)} \to K^0_S h^+ h'^-$ decays with first observation of $B^0_s \to K^0_S K^\pm \pi^\mp$ and $B^0_s \to K^0_S \pi^+ \pi^-$*, JHEP **10** (2013) 143. 18 p, Comments: 18 pages, 4 figures, submitted to JHEP.

[63] A. Abba *et al.*, *A specialized track processor for the LHCb upgrade*, Tech. Rep. LHCb-PUB-2014-026. CERN-LHCb-PUB-2014-026, CERN, Geneva, Mar, 2014.

[64] P. Hough, *Machine analysis of Bubble Chamber Pictures*, Proc. Int. Conf. High Energy Accelerators and Instrumentation **C590914** (1959).

[65] P. Hough, *Method and mean for recognizing complex patterns*, US Patent **3069654** (1962).

[66] D. Ninci, *Real-time track reconstruction with FPGA at LHC*, Master's thesis, University of Pisa, Pisa, Italy, Dec, 2014.

[67] Intel Corporation, *Arria 10 product table*, , `https://www.altera.com/products/fpga/arria-series/arria-10/overview.html`.

[68] Intel Corporation, *Stratix 10 product table*, , `https://www.altera.com/products/fpga/stratix-series/stratix-10/overview.html`.

[69] Xilinx Inc., *Virtex UltraScale+ Product Table*, , `https://www.xilinx.com/products/silicon-devices/fpga/virtex-ultrascale-plus.html`.

[70] Xilinx Inc., *Kintex UltraScale+ Product Table*, , `https://www.xilinx.com/products/silicon-devices/fpga/kintex-ultrascale-plus.html`.

[71] Intel Corporation, *Stratix V Device Handbook*, , `https://www.altera.com/documentation/sam1403479391092.html`.

[72] R. Cenci *et al.*, *First Results of an "Artificial Retina" Processor Prototype*, EPJ Web Conf. **127** (2016) 00005.

[73] B. Angelucci, E. Pedreschi, M. Sozzi, and F. Spinella, *TEL62: an integrated trigger and data acquisition board*, Journal of Instrumentation **7** (2012), no. 02 C02046.

[74] CDF, S. Belforte *et al.*, *SVT TDR (Silicon Vertex Tracker Technical Design Report)*, .

[75] R. Quagliani, Y. S. Amhis, P. Billoir, and F. Polci, *The Hybrid Seeding algorithm for a scintillating fibre tracker at LHCb upgrade: description and performance*, Tech. Rep. LHCb-PUB-2017-018. CERN-LHCb-PUB-2017-018, CERN, Geneva, May, 2017.

# Ringraziamenti

Anche questo lavoro è giunto al termine e mi ritrovo a scrivere il capitolo più letto di ogni tesi. In questo anno diverse persone mi hanno guidato, aiutato e hanno condiviso con me questa avventura, vorrei fare a tutte loro i miei più sentiti ringraziamenti.

Ringrazio il Prof. Giovanni Punzi per avermi offerto la possibilità di svolgere questo lavoro e di avermi sempre seguito durante questo periodo, ciò potrebbe sembrare ovvio, ma la quasi totalità dei laureandi che ho conosciuto non hanno avuto relatori altrettanto presenti. Lo voglio ringraziare anche per avermi permesso di partecipare a diverse conferenze da Bari fino al CERN passando per l'isola d'Elba (e quando mi ricapita una cosa del genere?).
Non posso certo non ringraziare il Dott. Riccardo Cenci, che mi ha seguito e guidato dall'inizio alla fine di questo lavoro o, in altre parole, da quando mi ha introdotto ai linguaggi di descrizione dell'hardware fino a quando ha corretto le mie contortissime frasi in inglese, per le quali gli rendo omaggio cominciando questo periodo con una doppia negazione.
Infine ringrazio tutti i componenti del gruppo di ricerca, in particolar modo gli altri due laureandi che hanno iniziato il loro lavoro più o meno insieme a me, Tommaso e Giulia, per aver condiviso questa avventura.

Un ringraziamento speciale va ai miei genitori, non penso che sia possibile riportare a parole ciò che mi hanno donato, vorrei però scusarmi con loro per averli fatti preoccupare tirando per le lunghe questo ciclo di studi.
Un grazie anche a tutti gli amici con cui sono cresciuto questi anni, in particolare Arianna che mi ha sorretto ed incitato come nessun altro, ora può prendersi la soddisfazione di essere prima di Alessio (mi aspetto un fragoroso "Almeno questo!").
Avendolo tirato in causa non posso che ringraziare Alessio per l'aiuto che mi ha offerto nello svolgere questo lavoro visto sia l'argomento della sua tesi magistrale (con palesi "similitudini") che quello del dottorato, risparmiandomi di dover reinventare la ruota; grazie anche per avermi reso meno testardo, anche se ha voluto dire ricevere una marea di craniate.

Non so se non dovrei ringraziare il mio portatile per essersi rotto poco prima della discussione o ringraziarlo di aver resistito fino a subito dopo la consegna della tesi nonostante mi stesse lanciando palesi segnali di allarme da tempo, comunque sia ringrazio mio fratello Giacomo per essersi privato del suo portatile per fornirmi un sostituto.
Parlando di croci e delizie informatiche di questa tesi parlerei di Git il quale avrà tante belle qualità ma anche la diabolica capacità di mostrarti in maniera inoppugnabile

che un'intera giornata di lavoro passata a cercare un errore nel codice si riduce nella modifica di una singola riga, trasformando rapidamente momenti di tripudio in sconforto. Se pensate che sia io ad averci messo troppo tempo a trovare gli errori, posso dire a mia discolpa che il debugging del codice VHDL è un tantino più complicato rispetto a quello di un software scritto in un qualsiasi linguaggio di programmazione e che se gli errori sono nel codice fornito dalla ditta produttrice dell'hardware allora ti senti anche po' preso in giro (evitando francesismi).

Tirando le somme posso affermare di essere felice di quest'anno così ricco di emozioni e chiudo con un classico

Addio, e grazie per tutto il pesce!