EPJ Web of Conferences **214**, 05023 (2019) https://doi.org/10.1051/epjconf/201921405023
CHEP 2018

A modular software framework for test-beam data analysis. The TbGaudi package.

Bartłomiej Rachwał[1]* and *Tomasz* Szumlak[1]on behalf of the LHCb collaboration

[1]AGH-UST University of Science and Technology, Faculty of Physics and Applied Computer Science, Krakow, Poland

Abstract. For purposes of test beam off-line data analysis a modular software framework has been developed and described in theses proceedings. The `TbGaudi` package allows to handle the test beam data for a set of runs, as well as the set of different devices under test in one-go, and finally obtain an integrated workflow to present the results.
All code is written in C++ programming language. A class based design makes it flexible to add any new features of the device under investigation following a plug-in scheme. Currently, the toolkit handles different types of analysis such as charge collection efficiency, track resolution and eta correction, implemented for non-uniform irradiated sensors.
The framework is applied for the LHCb upgrade studies for the VELO and UT groups.

1 Introduction

In high energy physics experiments, silicon detectors are often subjected to a harsh radiation environment, specially at hadronic colliders. Understanding the impact of radiation damage on the detector performance is an indispensable prerequisite for a successful operation throughout the lifetime of the planned experiment.

A dedicated irradiation programme followed by detailed studies with a particle beam are essential for the proper evaluation of the detector prototypes and predict their performance after accumulating the design fluence.

In order to perform precise measurements with the LHCb VELO detector prototypes a dedicated high resolution pixel beam telescope was developed based on 8 Timepix3 detector planes [1]. This telescope has been taking data at CERN in the PS and SPS facilities since 2014. The Timepix3 can readout on data driven mode with very precise timestamps which makes triggering unnecessary. At the centre of the telescope a Device Under Test (DUT) can be installed on a motion stage allowing angular rotations about the y axis, and x and y translations, where the z coordinate is the beam direction. The telescope provides precise measurements of particle trajectories with a pointing resolution of $\sim 2\mu m$ and a time resolution of $\sim 1ns$ allowing in-depth analysis of the DUT performance.

The data produced by the telescope can easily incorporate the signals from the DUT and in particular for Timepix3 devices the analysis is straight forward. The LHCb software embedded in the so-called `KEPLER` project[2] performs the decoding of raw-data and produces

---

[*] e-mail: brachwal@agh.edu.pl

track objects inside the official LHCb's Gaudi framework [3]. In this way it is simple to produce histograms and ROOT NTuples with track and cluster data information.

Subsequently, the off-line analysis compares the performance after irradiation with several fluences of protons or neutrons from 2 to 8×10^{15} 1 MeV n_{eq}/cm^2 for devices of different vendors taking into account: (a) different silicon substrates (n-on-p or n-on-n); (b) distances from last pixel to the edge; (c) guard rings designs. In these studies the charge collection efficiencies (CCE), track resolution (TR), Eta correction are of particular interest as function of fluence.

For purposes described above, a modular software framework `TbGaudi` has been developed. It allows to handle the test beam data for a set of runs, as well as the set of different DUTs in one-go, and obtain an integrated workflow to present the results.

All code is written in C++, which is a general-purpose objective programming language. A class based design makes it flexible to add any new features of the device under investigation following a plug-in scheme. The interface is implemented for a series of algorithms in a user-friendly way. Currently, the toolkit handles different types of analysis such as CCE, TR and Eta correction, implemented for non-uniform irradiated sensors.

This proceeding describes the TbGaudi analysis framework as a proposal for a prototype of a general design framework for test beam campaigns, that could be used to obtain a quick ROOT based application for complex test beam data analyses being performed in different facilities and different irradiation sources or profiles.

2 General principles

At the outset, one needs to realize that the `TbGaudi` framework is not a kind of top-level configuration package, hence a basic c++ skill is essential for the analysis developers.

The framework is being developed under the gcc 6.3.1 compiler with c++14 standard features used. The package is primarily based on ROOT6 libraries [5].

The main goal of this analysis framework is to perform the off-line analysis of the test beam data which are needed to be initially produced with external software. The baseline are NTuples containing track and cluster data information.

2.1 Working at the lxplus s/w environment

It is very comfortable to utilize the CERN LXPLUS [4] resources to handle the analysis (e.g. it's easy to shared test-beam data to be analysed using LXPLUS EOS area). The dedicated automatic configuration has been prepared (experiment independent) in order to load necessary libraries and packages. Once the repository is cloned to the User's area, the environment setup can be performed simply by running: `source Tb/setup-slc6-cvmfs.sh`

3 The TbGaudi philosophy and packages structure

The main idea behind the `TbGaudi` is very simple - deliver to the test-beam analyst common tools, implemented as an abstract interfaces. These are related mostly to data handling including any feature binned to the DUT and test-beam run condition which are treated as parameters-of-interest. In other words all algorithms and methods to be used by analysers are intended to be in one place. There are common plotting style, fitting procedures etc. delivered, however they can be easily adjusted if necessary.

One of the most important feature is the suitable DUT area binning, which should be related to the different DUT irradiation profiles (especially when the DUT was irradiated non-uniformly). There are other features implemented in order to make the user's coding as short and simple as possible. However, there are analysis steps which are strictly related to the particular analysis, e.g. the data fitters or fluence to DUT binning mapping which are supposed to be defined by the User.

3.1 General

The `TbGaudi` is a middle-size project, where it was assumed to keep a flat structure with any new analysis related package. All packages compilation process is being handled with the `CMake` tool, hence there is a top level `CMakeLists.txt` combining all of them. Once the user is aiming to create a new test-beam analysis package, inheriting from the `TbGaudi` functionality the corresponding `CMakeLists.txt` should be created, where the `TbGaudi` shared library will be linked with. Finally the new package name should be added to the top level `CMakeLists.txt`.

3.2 The TbGaudi specific objects

Within the package there are only a few specific objects which the user needs to make himself familiar with, in order to understand the general analysis workflow. In most cases there are simple c++ classes inheriting from other classes, however some of them are specific, i.e. of singleton type, which means that the instantiation of a such class is restricted only to one object.

3.2.1 The TbGaudi singleton

The TbGaudi singleton is designed to control the most general features of the framework, such as the application run verbosity level, errors handling methods, or massage printing services.

3.2.2 The TbStyle singleton

Since the plotting service of the `TbGaudi` framework is based on the `ROOT TCanvas` functionalities the different styles can be defined. Hence, the TbStyle singleton is defined, where predefined styles can be simple grabbed by the user.

3.2.3 The TbJob

The most comprehensive object is the `TbJob`. It inherits from a series of base classes (i.e. `RunUtils`,`DutUtils`,`DutRootUtils`) in order to collect broad type of information defining the particular test-beam run. In addition through this type of object user can have access to test-beam run details.

3.2.4 The TbAnalysis

The `TbAnalysis` is designed as the base class (being virtual) to any new analysis package (e.g. the `TbVeloPixAnalysis`, see below). Within this class also general analysis requirements and run condition management is defined trough the static variables (e.g. `TbAnalysis::ClusterSize()`, `TbAnalysis::OutputLocation()`).

3.2.5 The TbResults singleton

The `TbResults` singleton is the one of the most important and intended to be expansively used by the user. It includes generic container for any type of results (e.g. `ROOT::TObject`, `std::vector`, `std::map`) which can be added in the user's data processing loop, and later to be used for filling the compilation of the plots. The plotting procedures are also predefined within this class.

3.3 The TbGaudi specific tools

Apart from the general `TbGaudi` objects, there are few most important tools, which help to ultimately define the User's analysis. These are the `TbDataHandler` and the `TbDutBinning` briefly described below.

3.3.1 The TbDataHandler

The baseline for this data analysis package is the `ROOT` framework [5], however it is not necessary to have an input data in corresponding `NTuple` format. In such a case, the `TbDataHandler` is a tool to handle different test-beam data formats. Anyhow the current tools are implemented to read from the `NTuple` files, which are assumed to be created one per each test-beam run. Series of data files can be automatically read-in, using the user's defined data base input file (specific format is proposed). From the the level of such data base file the user can define analysis parameters (such as the DUT binning scheme, cluster size, etc.) to be applied for each run individually. If such data does not exists in the input data base file, the global configuration is used.

3.3.2 The TbDutBinning

As it was already mentioned, the DUT binning scheme is the crucial point, since it is related to the possibly non-uniform DUT irradiation profiles. The `TbDutBinning` delivers predefined different types of such binning (Fig.1). Of course, the adopted DUT binning scheme to the particular device should be followed by the corresponding fluence mapping, which have to be performed by the User in a given implementation of his analysis package.

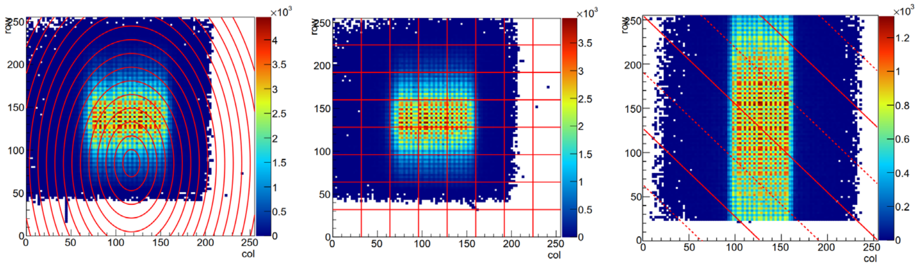

Figure 1: Different types of DUT binning schemes available in `TbGaudi` framework. The binning boundaries are placed on top of the DUT hit map.

4 Application development

Any new data analysis of the test-beam related campaign can be defined as a new application package, based on the whole functionality of the `TbGaudi`, as described in 3.1. The main point is to define a new class with inheritance from the `TbAnalysis`. Afterwards, the analysis object instantiation should be included within the main function assumed to be defined within the user's analysis package. The specific analysis methods are then to be simply executed from the level of the user's main function (e.g. Listing 1), where the example `CCE()` is defined at the level of user's analysis package, Listing 2.

Listing 1: The user's analysis object instantiation

```
auto MyAnalysis = std::make_unique<TbUserAnalysis >();
MyAnalysis->CCE();
```

Listing 2: The user's analysis class definition

```
class TbUserAnalysis : public TbAnalysis {
public:
        TbUserAnalysis();
        ~TbUserAnalysis();
        void CCE();
};
```

4.1 The TbVeloPixAnalysis

The `TbVeloPixAnalysis` is the `TbGaudi` package for the off-line analysis of the LHCb VeloPix test-beam data initially processed under `KEPLER` [2].

Radiation-induced effects (radiation damage) are investigated with fluence dependence. In the case of the LHCb VeloPix the two types of irradiation profiles have been performed. Uniform irradiation is useful to check all sensor characteristics, and non uniform irradiation allowing for ultimate check of *IV* performance.

The two types of irradiation profiles have been incorporated from the `TbGaudi`, with specific information for fluence to DUT binning mapping implemented in the `TbVeloPixAnalysis` package.

There is the IRRAD facility[1] that allows for non uniform irradiation which possess an approximately Gaussian beam. Utilizing such in our study the different DUT areas were irradiated with different fluence. From the point of view of the test-beam analysis, in practice there is not enough statistics and one need to have to make bigger bins. In that case, it is justified to handle this issue with the `TbGaudi` implemented binning schemes (compare Figure 1. and Figure2.), where the different radiation schemes are followed by adaptive binning – benefit for statistic and physics analysis.

As an example analysis the CCE vs fluence study is being performed by fitting the charge distribution for a given DUT binning scheme by implementing the Landau convoluted with Gaussian fitting model. That can tell us how much charge we can get for a given level of fluence. Example plots for this type of analysis are shown on Figure 3. Its worth to mention that it would be really hard to compile series of different DUT analyses without a framework, in addition to the possibility to re-run the whole analysis – if needed.

---

[1] The IRRAD proton facility is located on the T8 beam-line at the CERN PS East Hall.

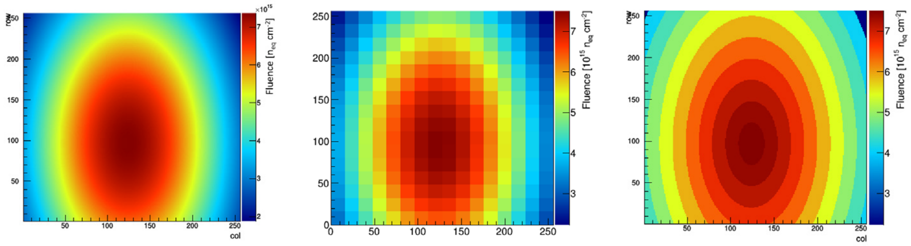

Figure 2: The example of the IRRAD fluence profile and corresponding DUT binning mapping.

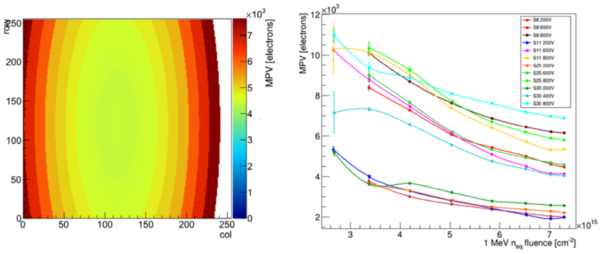

Figure 3: The preliminary LHCb VeloPix test-beam analysis results. The example of the MPV map for the DUT area (left) and the MPV vs fluence profiles for the series of sensors under investigation (right).

5 Summary and prospects

The described `TbGaudi` framework is an easy to adapt framework by any group that is working on new sensor technology. It's available as the gitlab repository [6]. We believe it is a valuable complement to be used by different groups performing detector R&D programs.

We acknowledge support from the national agencies: MNiSW and NCN (Poland) UMO-2015/17/B/ST2/02904 and UMO-2016/21/B/ST2/01083

References

[1] Barrand, G. and others `The Timepix3 Telescope for LHCb Upgrade RD 1 measurements`, PoS ICHEP2016 (2017) 1186
[2] The KEPLER package, Tpx3TestbeamSoftware
[3] Saunders, Daniel Martin (for the LHCb collaboration) `GAUDI - The software architecture and framework for building LHCb data processing applications`, Proceedings CHEP 2000
[4] The CERN LXPLUS service, lxplus-service
[5] The ROOT toolkit, ROOT data analysis framework
[6] The TbGaudi repository, https://gitlab.cern.ch/TbAnalysis/Tb