
Review document: Low Level Trigger (LLT)

S. Akar¹, J.-P. Cachemiche¹, H. Chanal², J. Cogan¹, C. Drancourt³, R. Lefevre²,
R. Le Gac¹, N. Neufeld⁴, P. Robbe⁴, M. Vesterinen⁵

¹CPPM, Marseille, France

²LPC, Clermont Ferrand, France

³LAPP, Annecy, France

⁴CERN, Geneva, Switzerland

⁵LAL, Orsay, France

⁶Physikalisches Institut, Ruprecht-Karls-Universität Heidelberg, Heidelberg, Germany

Abstract

This document presents the possible implementations of a Low Level Trigger (LLT) for the upgrade of the LHCb detector. Two scenarios are under study: a hardware LLT similar to the currently existing L0 trigger and a software trigger using CPU of the event builder computers. The performances, timeline, cost and manpower of the LLT are also described.

Contents

1	Introduction	1
2	Hardware implementation	1
2.1	LLT Calo	2
2.2	LLT Muon	4
3	Software implementation	5
3.1	LLT Calo	6
3.2	LLT Muon	7
4	Performances	8
4.1	LLT Calo	9
4.2	LLT Muon	11
5	Project organization	12
6	Conclusions	12
	References	13

1 Introduction

The upgrade trigger system includes the possibility to have a fast first level of trigger, that would reduce the rate at the input of the software trigger (HLT) farm, between 1 and 40 MHz. This Low Level Trigger (LLT) is a re-implementation of the current hardware first level of trigger of LHCb (L0 trigger) [1] and would be used during the first months of the running of the upgraded experiment [2]. As the reduced luminosity in this period will very likely not require the CPU capacity of the entire projected HLT farm, it is of interest to stage the acquisition of this farm. During this period, the LLT can efficiently be used to reduce the rate accordingly.

The LLT uses information from the calorimeter (ECAL and HCAL) and the muon detectors to search for high transverse energy (E_T) or high transverse momentum (p_T) particles and to compute global event hit multiplicities which are proportional to the entire detector occupancy. High momentum particles (electron and photons for the ECAL, hadrons for the HCAL and muons for the Muon detectors) are the sign of the presence of heavy flavour hadrons. Therefore selecting events with at least a particle with high momentum will enrich the data sample in D and B events, and reject uninteresting minimum bias events. It is also crucial to reject as early as possible events with extremely large detector occupancy, since these take a large amount of processing time for little impact for the final analyses.

The LLT contains two independent sub-systems: LLT Calo and LLT Muon dealing with ECAL and HCAL, and Muon detector data, respectively. Two scenarios are studied for the implementation of the LLT: one purely in hardware, very similar to the current L0 system, and one in software. These implementations are detailed in this document.

2 Hardware implementation

Conceptually, the LLT hardware implementation is very similar to the current L0. The quantities from the detectors are processed in Front-End boards (FEB) located close to the LHCb detector, in the cavern of the experiment. The information from the FEB is sent on long distance optical links to electronics boards (LLT boards) in which the computation of the trigger quantities takes place. These boards are located on the surface of the LHCb experimental area. The results of the LLT Calo and Muon algorithms are then sent to the S-ODIN board, responsible to control the readout of the entire detector [3,4]. The S-ODIN board regulates the rate of the readout system, sending throttle signals to the LHCb readout boards, and can then use the LLT information to decide for each event if it has to be sent to the event building network or not, for further processing by the HLT. The main difference with the current running scheme of the LHCb experiment is that now, no buffering occurs in the FEB, only in the readout boards [5].

The LLT boards are the same boards as the readout boards [6,7]. Two solutions are explored to realize the readout boards, a PCI-express board (PCIe40) plugged directly in the computers of the event building farm or a board in AMC format (AMC40) hosted on ATCA carrier boards [8]. These two solutions are completely identical for the LLT

implementation because they both provide the same functionalities needed for the LLT. These functionalities are:

- a large FPGA where a custom firmware is loaded to perform the LLT computations,
- 36 optical GBT input links,
- 36 optical output links.

For simplicity, only the PCIe40 alternative will be presented in this document.

The optical links of the PCIe40 boards are used to receive information from the detector's FEB (input links), to exchange data between trigger PCIe40 boards (input and output links) and to send the final results to the S-ODIN boards. The S-ODIN board is also a PCIe40 board which has 28 optical input links reserved for the LLT. The S-ODIN firmware implements a decision block (decision unit or LLT-DU) to interpret the informations of the two LLT sub-systems and to compute the final LLT decision.

2.1 LLT Calo

From the energy deposits in the ECAL and HCAL calorimeter cells, the LLT Calo computes 6 quantities used for the final decision:

- the E_T and position (address) of the cluster formed with 2x2 ECAL cells, with the highest E_T over the entire ECAL, called *electron candidate* (but which can also be a photon),
- the E_T and position (address) of the cluster formed with 2x2 HCAL cells, with the highest E_T over the entire HCAL, called *hadron candidate*,
- the sum of the E_T of all ECAL cells,
- the sum of the E_T of all HCAL cells,
- the number of ECAL cells with E_T above a programmable threshold, or *ECAL multiplicity*,
- the number of HCAL cells with E_T above a programmable threshold, or *HCAL multiplicity*.

These quantities are computed in steps, with a flow which is identical for the ECAL and HCAL, and represented in Fig. 1.

1. The energy measurements from the ECAL and HCAL photomultiplier tubes are digitized in the FEB. At this point, the measurements are sent to two different paths: the DAQ path at the end of which the information is written on tape for offline analysis, and the trigger path, at the end of which the informations are sent to the LLT-DU.

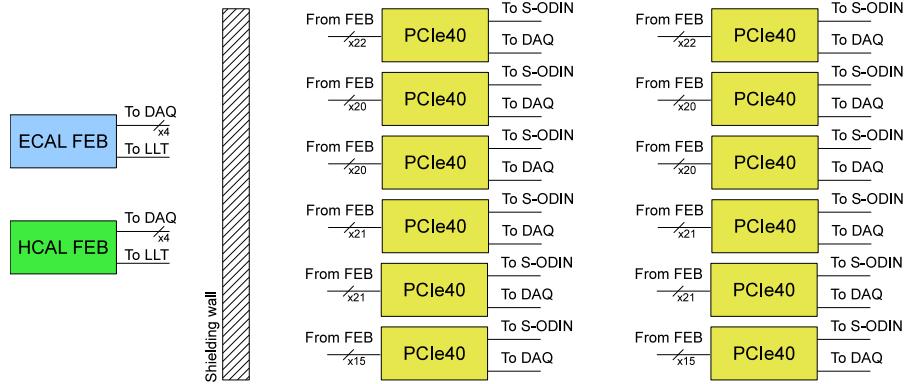


Figure 1: Schematics of the LLT Calorimeter hardware implementation.

2. The digitized energy measurements of each cell are transformed into 8 bits of roughly calibrated ADC values for the LLT computations. This step is realized in the Front-End FPGAs of the FEB which are in common with the DAQ path. One FEB handles 32 cells.
3. One FPGA (TrigFPGA) on each of the FEB receives the 32 ADC values from the previous steps and computes the E_T of the 32 associated clusters, counts the number of cells with E_T above a programmable threshold for the multiplicity measurement, and sums the E_T of the 32 cells for the total energy measurement. It then selects the cluster with the highest E_T to send it to the next step. In order to compute the E_T of all possible 32 clusters, the TrigFPGA needs to access the E_T of cells of neighbouring boards. This information exchange is done with Ethernet cables and through dedicated connections on the backplane of the Front-End crate. This infrastructure already exists for the current L0 and will be re-used for the LLT without modification. The FEB is a new board designed for the upgrade, and the TrigFPGA is also new but reusing a large fraction of the firmware of the TrigFPGA of the current board. The new functionality to implement for the upgrade is the computation of the multiplicities. The development of the TrigFPGA firmware is part of the Calorimeter Upgrade project [9].
4. The LLT quantities of each FEB, namely 8 bits of E_T and 5 bits of address of the highest energy cluster, 6 bits of multiplicity, 13 bits of total energy, 8 bits of board identifier and 8 bits of bunch crossing identifier, are sent to the PCIe40 boards on long distance optical fibers. The FEB has 5 output optical links, 4 are used for the DAQ path and one is dedicated to the LLT. The link uses the GBT protocol with 80 bit frames.
5. 12 PCIe40 boards receive the LLT informations from all FEB. They select the highest E_T of their inputs, and sum the input multiplicities and total energies. The results are sent via optical fibers to the LLT-DU on the S-ODIN board which computes

from them the 6 final quantities of the LLT Calo subsystem. In addition, the PCIe40 boards send to the event building computers all the values that they receive and that they compute. These values will then be stored in the complete event, to be able to monitor the LLT behaviour offline.

In summary, the LLT Calo hardware system consists in one TrigFPGA per FEB (188 for the ECAL and 50 for the HCAL), 238 long distance optical fibers and 12 PCIe40 boards. The total LLT processing time (from the input of the TrigFPGA to the output of the PCIe40) is not yet known, but is faster than the current L0 which is more complex [10]. It should then be at maximum equal to $2\ \mu\text{s}$, which is the current L0 processing time.

2.2 LLT Muon

The LLT Muon selects high transverse momentum muons. Tracks are searched looking for hits defining straight lines in the muon stations and pointing towards the interaction point. The transverse momentum of the candidates are then estimated assuming a particle originating from the interaction point and a single kick from the magnet. The candidate with highest transverse momentum is selected and sent to the LLT-DU to make a decision.

The muon detector upgrade is described in the LHCb PID upgrade TDR [9]. The main points of interest from the LLT point of view are:

- the design of the new off-detector electronics (new ODE) using the GBT-based communication protocol, resulting in a large decrease of the number of links in input of the LLT Muon.
- the removal of the station M1, reducing the amount of logical channels used in the trigger.
- the removal of the Intermediate Boards in region 4 of the station M5, increasing of the granularity in this area.

As a baseline solution, the LLT is implemented in PCIe40 boards. These boards receive the binary hit informations from the new ODE. At this stage, the data are synchronous and non zero-suppressed. In this context and given the large capacity of the FPGA used on the PCIe40 (50 times more logical elements than the one used in the current L0 Muon), the LLT Muon can be implemented in a compact system using 16 PCIe40 boards, each board treating the data from one quarter's region. Only 4 PCIe40 boards are then necessary to cover one full quadrant. Like in the L0 Muon, a patch panel is used to group the links from the same region for the four muon stations. At the input of the PCIe40 board, the number of links from the new ODE varies from 12 to 14 depending on the underlying region. To avoid edge effects between adjacent regions that would lead to efficiency loss, data need to be exchanged between PCIe40 boards. The amount of data transferred between two PCIe40 is well fitted in a single bidirectional link. The candidates found in each PCIe40 of one quadrant are concentrated in one of the PCIe40 boards to be selected further and sent to the LLT-DU. The total number of additional links needed for

the data and candidate transfer between boards reaches a maximum of 6 (in and out) and can be accommodated by the PCIe40 remaining links. Finally, the compressed input data are sent to the event-building farm using the PCIe bus.

The candidate finding algorithm currently implemented in the L0 Muon is used. The main steps of the algorithm are :

1. data transfer from and to the neighbouring boards,
2. the building of the logical pads from the crossing of the horizontal and vertical strips,
3. seeding in station M3,
4. searching for hits inside fields of interest around the seed extrapolation in M2, M4 and M5,
5. getting the candidate transverse momentum using a look up table addressed with the hits position in station M2 and M3,
6. transfer of the candidate to a common PCIe40,
7. selection of the two candidates with highest transverse momentum and sending to the LLT-DU.

Extrapolating from the current implementation, this algorithm would take about half of the available logical elements in the FPGA. The other half is used to perform the data compression for the readout of the muon hits.

In summary, the LLT Muon is implemented in 16 PCIe40 boards. The system is very similar to the current L0 Muon but much more compact and therefore much simpler.

3 Software implementation

The event building farm will be capable from the beginning of the running of the upgraded detector, to build events at the full 40 MHz rate [8]. This farm will consist in about 500 CPU. It is estimated that between 1 and 2 ms of 2014 CPU time per event will be available for something else than the event building tasks. It is then conceivable to use this time to run LLT algorithms. In this case, the LLT final decision is not sent to S-ODIN, but is attached to each full event, and from this information, it is then decided to send the event to the HLT or to drop it completely. The LLT-DU block in the S-ODIN firmware is then not anymore necessary.

There are two main reasons why the event-building of the full event of every bunch-crossing can be performed at the same cost as was foreseen in the FTDR [2] for a 10 MHz system (corresponding to one third of the full system).

1. The advent of a new generation of PC-servers which have an I/O capability of 800 Gbit/s (full-duplex) using PCIe Generation 3.

2. The advent of data-centre network technology, which has drastically lower prices per switch-port than the telecom-class routers used in the current LHCb DAQ.

Both are needed together because the data-centre networking such as InfiniBand or data-center Ethernet requires significant buffering and intelligence in the end-nodes. It has been studied if this can be provided by stand-alone FPGA-based modules. This has been found to be expensive and to increase the complexity of these modules by a lot¹. Since modern servers can easily handle 100 Gbit/s, which is the target bandwidth for a readout-card in the LHCb upgrade, it has been decided to make the modules into PCIe-plugin cards in the PCs. These PCs do then the event-building using a suitable 100 Gbit/s network technology (to be decided close to acquisition of the servers)². It could be shown based on current quotes that such a system, with 500 event-building PCs is affordable in the financial envelope foreseen for the event-building in the FTDR. The total capacity of this event-builder is 50 Tbit/s. The fact that the expected aggregated traffic is about 24 Tbit/s and that the farm and connections would allow to go to twice as much leave a safety of about 2 against unforeseen behaviour and/or higher luminosity. It should be kept in mind that if the aggregated data rate from the detector increases significantly beyond this, the limitation will come from the aggregated capacity of the input GBT links and not from the event-builder.

It has been tested that in addition to event-building at 100 Gbit/s, the machine can be loaded up to 50% with trigger-processing. The limitation does not come from the CPU but from the memory bandwidth available in the machine. The memory bandwidth of PCs will increase over time to keep up with the increasing number of cores, however the memory bandwidth for event-building will stay constant since we will not add more than a single 100 Gbit/s FPGA card into one event-building PC. We are therefore confident that at least 80% of the PC will be available for processing other than event-building, such as a software LLT.

3.1 LLT Calo

The most straightforward implementation of the LLT Calo in software is to use the hardware architecture described in Sec. 2.1 and to perform only the step corresponding to the LLT-DU in software. The LLT data from all ECAL and HCAL FEB are available in the full event because the PCIe40 for the LLT Calo send to the event building farm their inputs. The LLT Calo software task consists in extracting these data from the full event, loop over them to select the highest E_T electron and hadron candidates, and to sum the per-FEB multiplicities and total energies to obtain the global ECAL and HCAL multiplicities and total energies. This algorithm takes 10 μ s of CPU time per event (including the raw event decoding time), estimated on a dual-socket Intel X5650

¹For instance an InfiniBand IP-core will take about 30 to 40% of the available logic resources of the planned Stratix V FPGA

²Since the bandwidth per port is so high, there is little interest (and cost-gain) in running at lower speed and thus just by installing the system one gets the full event-building for 40 MHz bunch-crossings.

("Westmere") machine, and removes the need of a dedicated firmware that would do the same computations in the LLT Calo PCIe40 boards, and of the connections between these PCIe40 and S-ODIN.

Since no dedicated firmware is needed for the LLT Calo PCIe40 boards in this software approach, a further step ahead is to send the LLT data from the FEB through the DAQ path. The LLT data will then be received by the standard calorimeter readout boards (DAQ Calo PCIe40) and treated as normal data. The LLT information will then be also available in the full event and the software algorithm described above will perform the final LLT computations. This option removes the need of the dedicated LLT long distance fibers between the FEB and the PCIe40 (238 fibers) and of the dedicated PCIe40 boards (12 PCIe40).

The extreme option is to compute in software everything which is done in the hardware implementation, starting directly from the calorimeter full information. This would mean perform the rough cell-by-cell calibration done in the FEB and the 2x2 clustering done in the TrigFPGA of the FEB. The software algorithm corresponding to that scenario takes 2.9 ms of CPU time per event, in a version which is not optimized. In this scenario, the TrigFPGA in each FEB will not be needed anymore (238 FPGA).

The detailed study and optimization of the LLT Calo implementation in software is still ongoing, but the second option presented in this section seems already attractive: for the modest extra cost of one TrigFPGA per FEB, the LLT information is pre-processed in a way that saves enough CPU time to run the LLT calorimeter in software. Note that the algorithms presented in this section are coded in a way independent of the instantaneous luminosity of the LHC: each cell is processed independently of its energy, including cells with no energy deposit. This means that the timings reported here are independent of the luminosity.

3.2 LLT Muon

A full software scenario has been studied for the software implementation of the LLT Muon. In this approach, the algorithm starts by retrieving the digitized information from each of the four muon stations (M2 to M5). Each muon station is divided in several sectors, which are themselves divided in overlapping horizontal and vertical logical channels. A hit in the muon stations is identified by at least one logical channel. These are then used to construct logical pads defined as the intersection of two hit logical channels. Using a realistic detector geometry, the cartesian coordinates of the logical pads are then calculated and stored. In this algorithm, a muon track is defined as four aligned hits in the muon stations; iteratively starting from the muon station M5 down to M2. At each iteration, the extrapolated point in station M_i is obtained using a straight line from the considered hit in station $M_{(i+1)}$ to the interaction point. For each muon candidate, the transverse momentum is further calculated from the coordinates of the hits in M2 and M3 of the considered track. The p_T calculation is done in the thin lens approximation for the dipole magnet and without further approximation on small angles. For each event, the muon candidates, with their corresponding p_T , are finally written in order to be later accessed

by the HLT farm.

The software algorithm corresponding to that scenario takes on average ~ 0.7 ms of CPU time per event, estimated on a dual-socket Intel X5650 ("Westmere") machine. Table 1 gives the break down of the timing for each of the different steps described above.

The detailed study and optimization of the LLT Muon implementation in software is still ongoing. The full software LLT muon option presented in this section seems attractive, given that no extra cost from hardware implementation is required, and that the current timing, combined with the second option for the LLT Calo presented in Sec. 3.1, is already below the CPU time available for the LLT in the event building.

4 Performances

This section presents the performances of the LLT for decay channels representative of the LHCb physics program of the upgrade [11]. The trigger efficiency for these channels and the minimum bias retention rates are estimated from full Monte Carlo simulation generated for the conditions expected for the upgrade: a proton beam energy of 7 TeV, a luminosity of $2 \cdot 10^{33} \text{ cm}^{-2}\text{s}^{-1}$ and a bunch spacing of 25 ns. This corresponds to 30 MHz of visible collisions in LHCb.

Table 1: Timing break-down for each of the LLT Muon algorithm step.

Algorithm sequence	Time (% of total)	Time (ms)
Load hits from Raw Data Banks	17.3	0.120
Build logical pads from logical channels	17.9	0.124
Calculate cartesian coordinates	36.6	0.253
Search for muon candidates	5.8	0.040
Calculate the p_T	0.3	0.002
Writing down muon candidates information	22.1	0.153
Total	100.0	0.692

4.1 LLT Calo

The performances are evaluated from a complete emulation of the LLT algorithms, and are identical for the hardware and software implementations. They are computed for the decay modes $B^0 \rightarrow K^+ \pi^-$, $B^0 \rightarrow D^+(K\pi\pi)D^-(K\pi\pi)$, $B_s^0 \rightarrow \phi(KK)\phi(KK)$, $D^0 \rightarrow K_s^0 \pi^+ \pi^-$ and $D^0 \rightarrow K^+ K^-$, for the *hadron* LLT line only. The minimum bias retention rate is also computed looking only at the *hadron* line decision.

Figure 2 (left) shows the efficiency that the reconstructed signal decay alone passes the *hadron* threshold, as a function of the value of this threshold. This is called *TOS efficiency* (TOS = Trigger On Signal). The right plot shows the same values represented as a function of the minimum bias retention rate achieved with the LLT hadron line only.

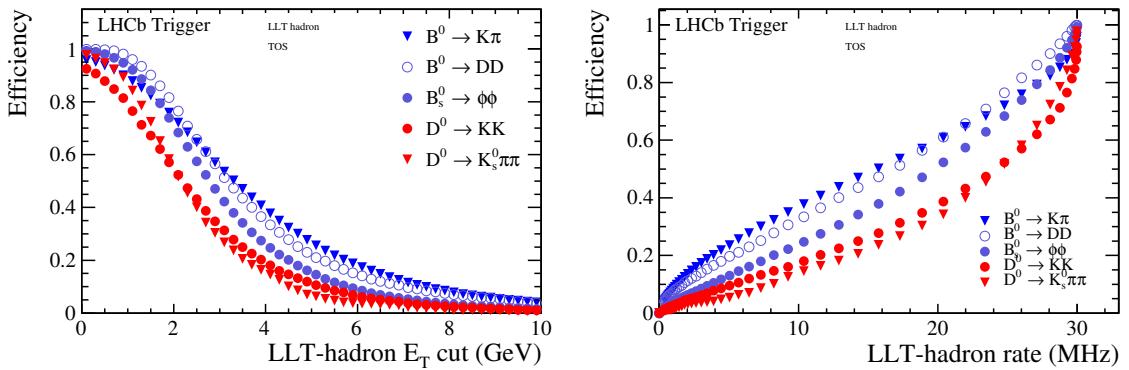


Figure 2: TOS efficiencies as a function of the hadron E_T cut (left) and of the minimum bias retention rate of the LLT hadron (right).

Figure 3 (right) shows the efficiency that an event containing the signal decay passes the *hadron* threshold, as a function of the value of this threshold, because of the signal decay products or because of the rest of the event. This is called *TIS or TOS efficiency* (TIS = Trigger Independent of Signal). The right plot shows the same values as a function of the minimum bias retention rate.

To estimate the performance during the period of initial running, at the reduced luminosity of $10^{33} \text{ cm}^{-2} \text{s}^{-1}$, the same efficiencies and retention rates are also estimated. They are shown on Figs. 4 and 5.

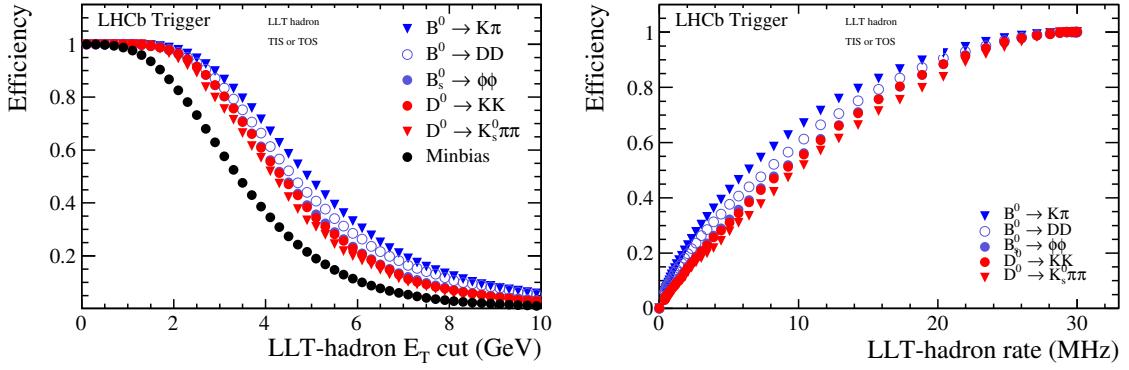


Figure 3: TIS and TOS efficiencies as a function of the hadron E_T cut (left) and of the minimum bias retention rate of the LLT hadron (right).

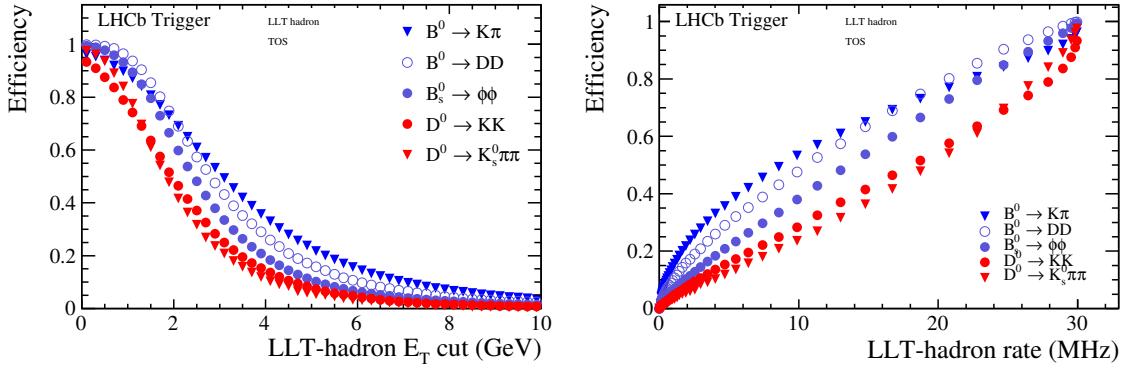


Figure 4: TOS efficiencies as a function of the hadron E_T cut (left) and of the minimum bias retention rate of the LLT hadron (right), for $L = 10^{33} \text{ cm}^{-2} \text{s}^{-1}$.

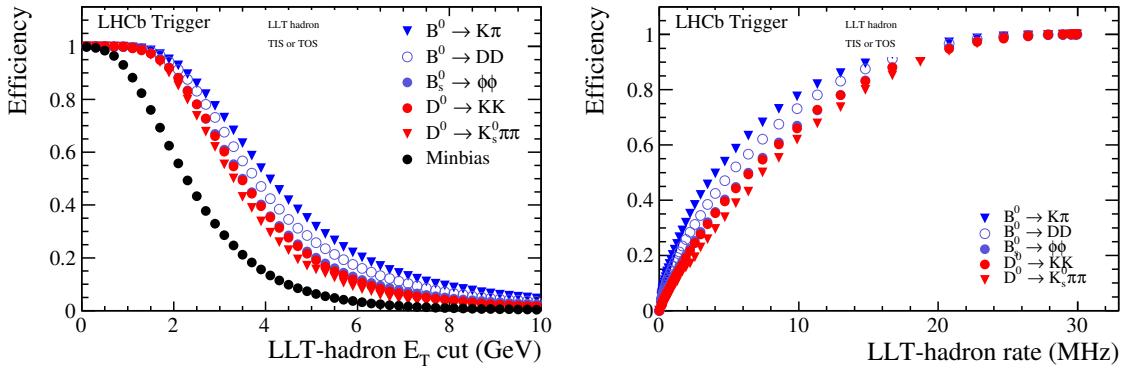


Figure 5: TIS and TOS efficiencies as a function of the hadron E_T cut (left) and of the minimum bias retention rate of the LLT hadron (right), for $L = 10^{33} \text{ cm}^{-2} \text{s}^{-1}$.

4.2 LLT Muon

The performances are evaluated for the two algorithms presented in the previous sections : the one corresponding to the current L0 with a seeding in M3 and the one developed for the LLT software implementation study with a seeding in M5. Any of these algorithms could in principle be used regardless of the technical choice of the actual trigger implementation.

The TOS efficiency for the $B^0 \rightarrow K^* \mu\mu$ decay, defined as the fraction of events for which at least one of the signal muon has a p_T above a threshold, is presented as a function of this threshold on Figure 6 (left) together with the retention fraction for minimum bias data defined as the fraction of minbias event giving a candidate above the threshold. The right plot shows the same values represented as a function of the minimum bias retention rate achieved with the LLT muon line only. The same quantities are shown on Figure 7 for a luminosity of $10^{33} \text{ cm}^{-2}\text{s}^{-1}$.

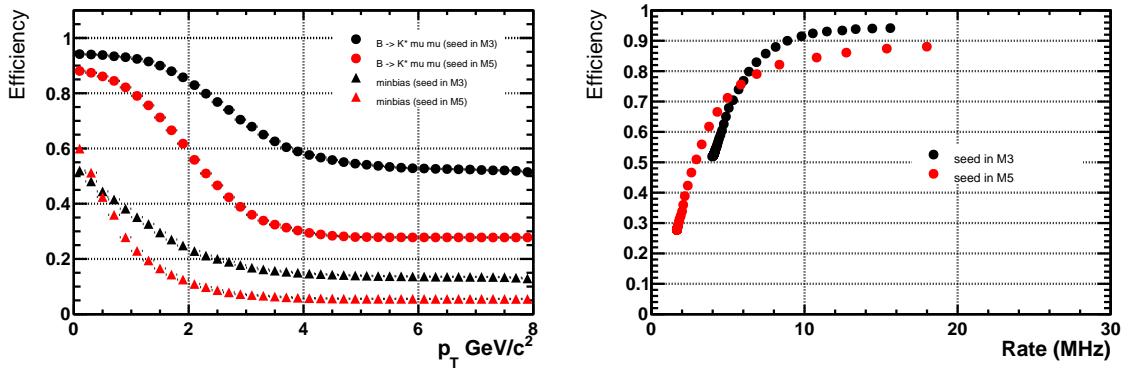


Figure 6: TOS efficiencies as a function of the muon p_T cut (left) and of the minimum bias retention rate of the LLT Muon (right) for a luminosity of $2 \cdot 10^{33} \text{ cm}^{-2}\text{s}^{-1}$.

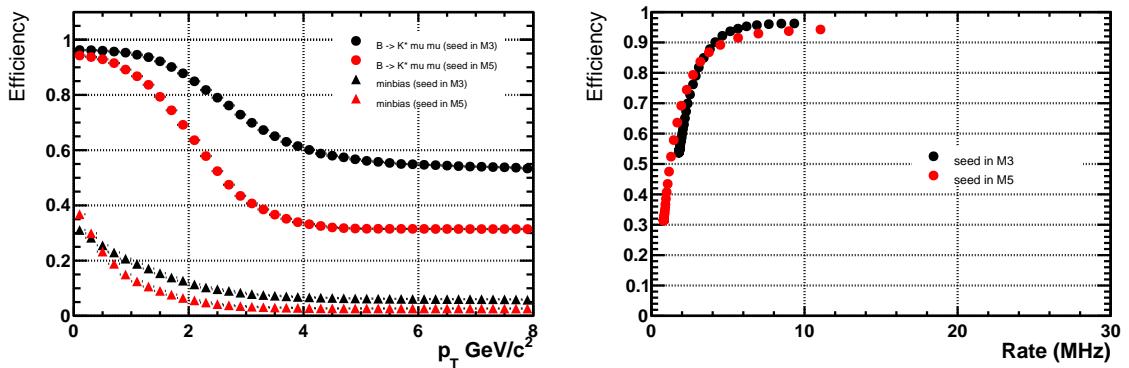


Figure 7: TOS efficiencies as a function of the muon p_T cut (left) and of the minimum bias retention rate of the LLT Muon (right) for a luminosity of $10^{33} \text{ cm}^{-2}\text{s}^{-1}$.

Table 2: LLT responsibilities and manpower.

Task	Institute	2014	2015	2016	2017	2018	2019	2020
Global coordination								
and <i>LLT Calo</i> software	LAL Orsay	0.5	0.5	0.5	0.5	1	1	1
<i>LLT Calo</i> firmware	LAPP Annecy			0.6	0.6			
<i>LLT Muon</i> coordination								
and software	CPPM Marseille	1	1	1	1	1	1	1
<i>LLT Muon</i> firmware	CPPM Marseille			0.5	2	2		
<i>LLT-DU</i> firmware	LPC Clermont				2			

Table 3: LLT costs.

Item	Cost (CHF)
<i>LLT Calo</i>	203
<i>LLT Muon</i>	117
<i>LLT-DU</i>	22

5 Project organization

The feasibility of the hardware implementation is established. The feasibility studies of the preferred implementation, the software one, will be continued. If the hardware implementation is required, the design of the firmware of the PCIe40 boards will start and take place in 2016 and 2017. From 2018, the installation and commissioning of the system will take place. Table 2 shows the various responsibilities for the project, and Table 3 the different costs of the LLT project.

6 Conclusions

A hardware first level of trigger can be realized for the upgrade of the LHCb detector. The implementation in the common readout boards (PCIe40) of the experiment is ensured and the performances established. The LLT system provides the necessary components to be able to reduce the rate at the input of the HLT farm, if needed at the startup of the upgrade phase.

An interesting possibility to realize partly or completely the task of LLT is emerging, using available CPU time in the event building computers. The study and optimization of this possibility will be pursued in the coming year, in order to move the project in that

direction.

References

- [1] R. Aaij *et al.*, *The LHCb trigger and its performance*, JINST **8** (2013) P04022, arXiv:1211.3055.
- [2] LHCb collaboration, I. Bediaga *et al.*, *Framework TDR for the LHCb upgrade: technical design report*, Tech. Rep. CERN-LHCC-2012-007, LHCb-TDR-12, 2012.
- [3] F. Alessio and R. Jacobsson, *System-level specifications of the timing and fast control system for the LHCb upgrade*, LHCb-PUB-2012-001.
- [4] F. Alessio, *Timing and fast control aspects in a PCIe-based readout architecture for the LHCb upgrade*, tech. rep., 2014.
- [5] K. Wyllie *et al.*, *Electronics architecture of the LHCb upgrade*, LHCb-PUB-2011-011.
- [6] J.-P. Cachemiche *et al.*, *TELL40 implementation - feasibility study of implementation of the LHCb readout board in several form factors*, Tech. Rep.
- [7] J.-P. Cachemiche *et al.*, *Readout board specifications for the LHCb upgrade*, Tech. Rep. EDMS 1357162, 2012.
- [8] N. Neufeld *et al.*, *The LHCb readout for the run 3*, Tech. Rep. EDMS 1357418, 2014.
- [9] LHCb collaboration, I. Bediaga *et al.*, *LHCb PID upgrade technical design report*, Tech. Rep. CERN-LHCC-2013-022, LHCb-TDR-14, 2013.
- [10] A. Martín Sánchez, P. Robbe, and M.-H. Schune, *Performance of the LHCb L0 Calorimeter Trigger*, LHCb-PUB-2011-026.
- [11] R. Aaij *et al.*, *Updated sensitivity projections for the LHCb upgrade*, LHCb-PUB-2013-015.