

**AN ALTERNATIVE ARCHITECTURE
OF THE L0(μ) PROCESSOR**

E. Aslanides, B. Dinkespiler, R. Le Gac, M. Menouni and R. Potheau

Centre de Physique des Particules de Marseille
CNRS/IN2P3 et Université de la Méditerranée
Faculté des Science de Luminy, Case 907, 13288 Marseille Cedex 09, France

Abstract

An alternative architecture of the L0(μ) processor and its implementation are presented. The architecture of the processor is based on a strong zero-suppression in order to minimize the data flow coming from the muon detector. It can be achieved by the fast identification of the muon tracks in all muon chambers using adequately dimensioned pad sectors and by transferring the individual pad information only for the regions close to the muon tracks. The proposed solution is simple, flexible and compact. Based on present technology the processor could execute the complete L0(μ) algorithm and make its decision available within less than 3 μ s.

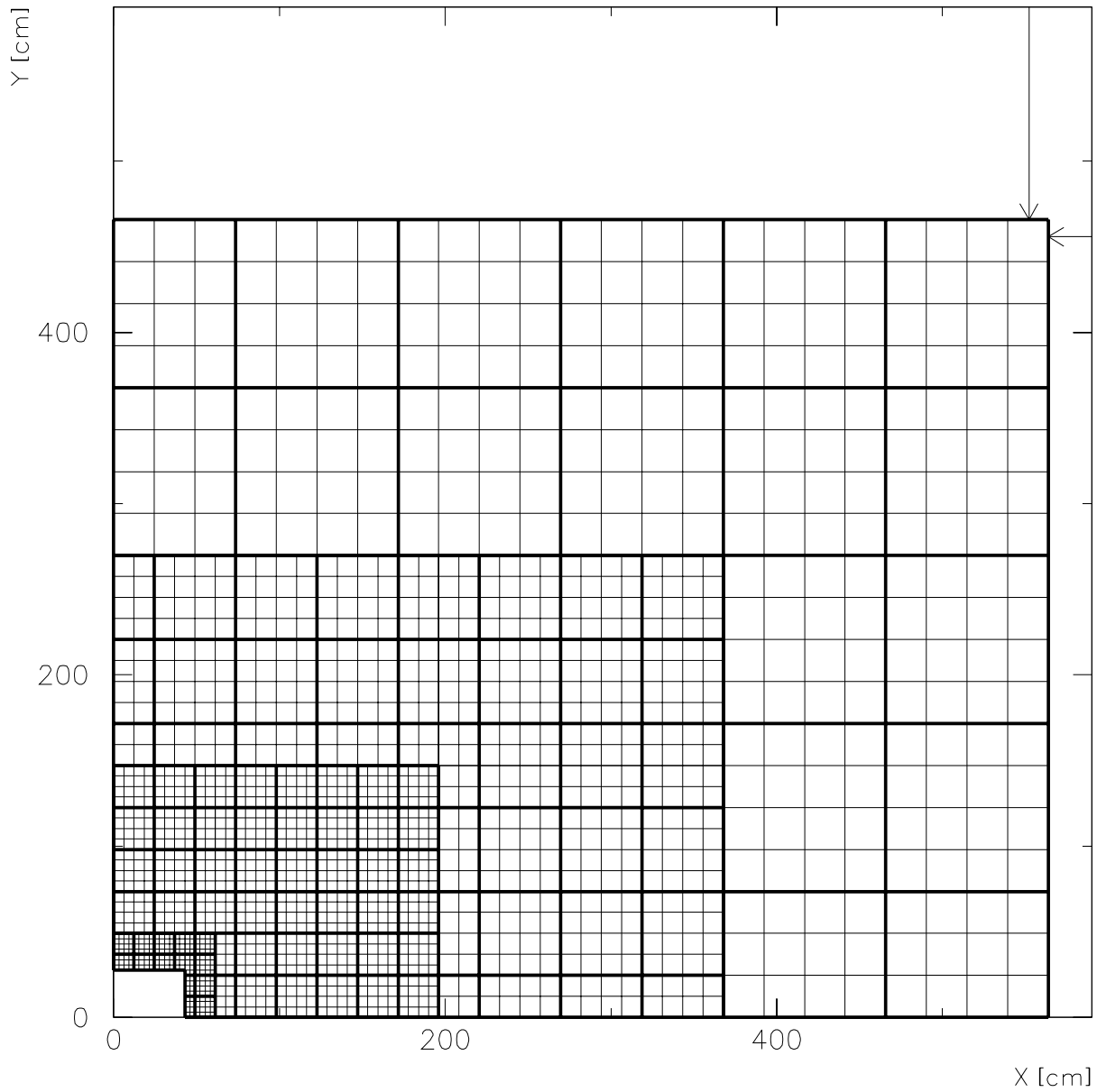


Figure 1: The sector geometry for a quarter of the muon chamber 5. The thin lines show the edge of the logical pads while the thick lines delimit the sectors.

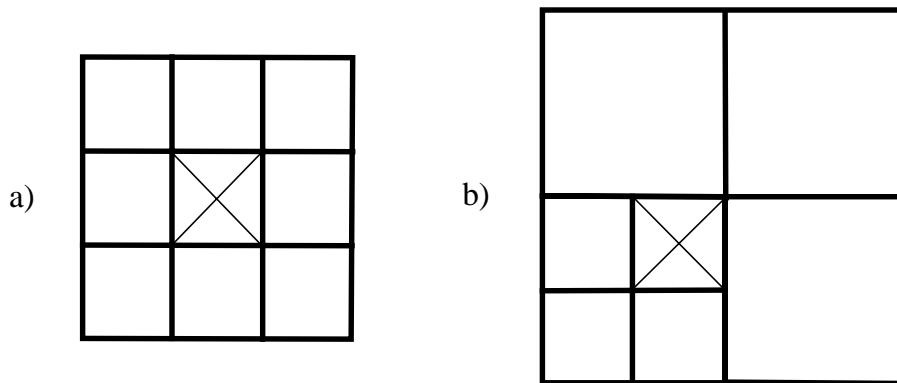


Figure 2: Two examples of cluster shape when the central sector is a) far away from a chamber region boundary b) close to a chamber region boundaries.

1. Introduction

The $L0(\mu)$ processor is one of the first stages of the LHC-B trigger system which should reduce the event rate from 40 MHz down to 1 MHz. To achieve such a reduction, the processor has to identify muons and to reconstruct their tracks in the muon detector. It measures the transverse momentum of muon candidates and the slope of their tracks at the entrance/exit planes of the muon filter. It keeps an event if the transverse momentum of at least one of the reconstructed muon tracks is above 1.1 GeV/c. The $L0(\mu)$ processor receives $\approx 45k$ binary informations, at a rate of 40 MHz. Its decision has to be taken within less than 3.2 μ s for each beam crossing.

The description of the muon filter device [1] and the algorithm proposed for the $L0(\mu)$ processor [2] can be found elsewhere. Indeed, this note is devoted to the architecture of the $L0(\mu)$ processor and its hardware implementation. The solution is based on current technologies and might be an alternative to the 3D-Flow design [3]. Our main guidelines looking for such a solution have been to design an architecture: a) minimizing the data flow between the muon detector and the processor b) simple, compact and flexible c) which can be debugged, monitored and repaired easily d) being evolutive as far as the physics requirements and the technological progress are concerned.

We studied the feasibility of this hardware processor by splitting it in blocks of functions and by evaluating the data flow between them.

The zero-suppression is based on the sectorization of the muon chambers and on the use of a fast track identification algorithm. Both are described in section 2. In section 3, we present a scenario for the data transfer between the front-end electronics of the muon chambers and the $L0(\mu)$ processor. These two ingredients are sufficient to design the $L0(\mu)$ architecture which is described in section 4. Finally in section 5, the conclusions and the perspectives of this work are summarized.

2. Pad sectors and the fast muon identification algorithm

The full binary pad information arriving at the input of the front-end electronics [4] corresponds to a rate of $(45000 \text{ pad}) \times (40 \text{ MHz}) \approx 225 \text{ Gbyte/s}$. They have to be transferred to the $L0(\mu)$ processor. However, the present simulation shows that average occupancy of muons is ≤ 0.3 per event. A sizeable reduction of the data flow, ≈ 20 , can be obtained by identifying quickly the muon candidates, by finding in all muon chambers adequately dimensioned regions crossed by the muon tracks and by using the detailed pad information only for the regions close to the muon track.

To perform the fast muon identification, we divide the muon chamber into sectors including 16 (4×4) pads. A possible map of the sectors is shown in Fig. 1, for a quarter of the chamber 5. For the other chambers the map is identical but the sector sizes are scaled down profiting from the projectivity of the muon detector geometry. In this design most of the sectors contain 16 pads. However, some of them are smaller since the dimensions of the chamber regions of given granularity are not always a multiple of 4 pads. This could be optimized to insure a constant trigger efficiency. A muon chamber contains 448 sectors.

The fast muon identification algorithm uses the sector hit maps of chambers μ_2 , μ_3 , μ_4 and μ_5 . It requests a logical coincidence between one sector in chamber μ_3 and clusters of sectors in chambers μ_2 , μ_4 and μ_5 . The sector at the centre of the cluster corresponds to the sector in chamber μ_3 used as the seed of the algorithm. (There is a one to one correspondence between sectors belonging to different chambers since the geometry of the muon detector is projective). The cluster also contains the sectors surrounding the central

one: ± 1 in both X and Y direction. However, the cluster shape depends on the chamber region boundaries as shown in Fig. 2. The cluster shape is designed to contain at least all pads appearing in the search window, which is used in the fine muon tracking algorithm. A cluster is hit if at least one of the encapsulated sectors is hit. The muon identification algorithm requires that for each sector hit in chamber μ_3 , all the corresponding clusters in chambers μ_2 , μ_4 and μ_5 are hit.

To evaluate the performances of this algorithm, we run the LHC-B simulation version v110 on minimum bias and $b \rightarrow \mu X$ data sets. A thousand of events are read in each data sample. The geometry data base was modified to conform the pad geometry chosen for the Technical Proposal.

The muon finding efficiency is measured on true muons coming from b decays with a transverse momentum above 1 GeV/c. It was found to be $(99.0 \pm 0.5)\%$.

We also determine the distribution of the number of muon, $\mu_{\text{CH2-CH5}}$, found in chambers μ_2 through μ_5 per event, for minimum bias data. The associated probability can be described by the function: $p(n) = \exp(-bn)/(1 - \exp(-b))$ where n is an integer running from zero to infinity. The parameter b was fitted on the simulated distribution. The average number of $\mu_{\text{CH2-CH5}}$ candidates as a function of the luminosity was derived taking into account the probability to have more than one pp interaction per beam crossing. The resulting distribution is shown on Fig. 3 together with the maximum number, n_{max} , of muon candidates expected. It is defined as the number satisfying the probability condition:

$$\sum_{k=0}^{n_{\text{max}}} p(k) \geq 0.99.$$

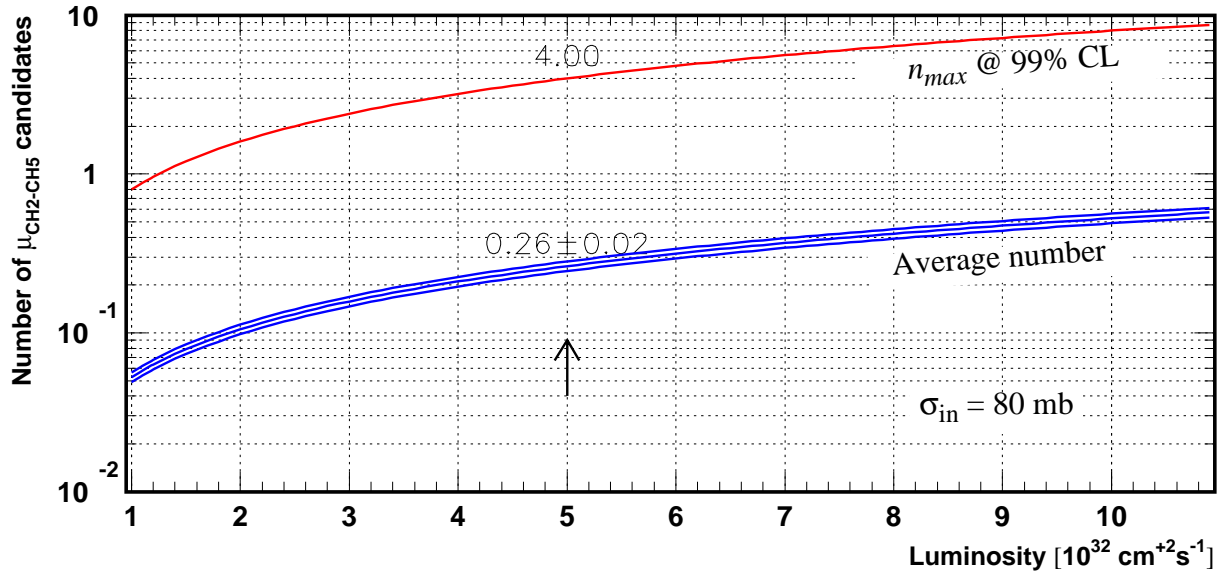


Figure 3: The average number of $\mu_{\text{CH2-CH5}}$ candidates as function of the luminosity

For a maximum luminosity of $5 \times 10^{32} \text{ cm}^2 \text{ s}^{-1}$, the average number of $\mu_{\text{CH2-CH5}}$ candidates per beam crossing is ≈ 0.3 and the maximum number is ≈ 4 . The same result was obtained by using only chambers μ_3 , μ_4 and μ_5 . Charged particles generated by the capture

of slow neutrons (not included in the present simulation) dominate the density of charged particles at large radius [5]. Such a background as well as electronics noise might be rejected using a fourfold instead of threefold “coincidence” for the fast muon identification.

Moreover, the estimation of the number of muon candidates has to take into account an additional rate due to muons produced by the LHC machine [6]. Presently, this number is not well known. However, it has been estimated at the level of 9 muons/cm²/s, from ATLAS and CMS studies. This flux adds ≈ 0.5 to the muon rate per beam crossing, integrated over the whole surface of chamber μ_5 . It decreases when the muon track crosses at least three chambers. This additional reduction can crudely be estimated from our minimum bias data set. We can expect a reduction factor by at least a factor two and probably much more.

In conclusion, the muon detection efficiency is very good and the average number of $\mu_{\text{CH2-CH5}}$ candidates per beam crossing is below one at the maximum luminosity. Therefore we assume an average of one $\mu_{\text{CH2-CH5}}$ candidate per beam crossing in our design.

3. Scenario for the data transfer between the muon detector and L0(μ).

The general scheme for the front-end electronics is given in ref. [4]. Following the amplifier, shaper and discriminator boards, they contain FIFOs and buffers to store events during the L0 and L1 processing times. These boards associate a beam crossing identifier to each data sample and also incorporate an interface to transfer the data to the L0 processor.

For the muon detector an additional function appears in the electronics before the front-end boards, since the muon chambers are made of several layers. The corresponding physical pads of each layer have to be ORed to create a unique logical pad hit map. The latter is sent to the front-end boards and to the L0(μ) processor.

A *front-end board interface* which sends data to the L0(μ) processor is shown in Fig. 4. The binary information associated to 512 pads and the corresponding beam crossing identifier arrive every 25 ns. The 512 pads are divided in 32 sectors of 16 pads. For each sector, the binary pad information is stored in a double access RAM, according to the beam crossing identifier. At the same time, the pads of a given sector are ORed to determine if the sector is hit or not. The 32 sector hit map is build. Thus at each beam crossing the sector hit map is stored in a register and the corresponding binary pad information are stored in RAMs. In addition, one can find the binary pad information of the 127 previous beam crossings.

The output of the *front-end board interface* are send on two optical links with a bandwidth of 1.2 Gbit/s. On the first link, one finds the 32 sector hit map send for each beam crossing. On the other stream, one gets the detailed binary pad information encapsulated into one sector. To obtain it, a request is sent on the selected bus containing a beam crossing identifier (7b) and a sector identifier (9b)¹. The interrogation box selects the corresponding RAM, extracts the requested information and return it to the user (32b).

The sector stream will be used to run for each beam crossing, the fast identification algorithm, and to determine where the muon track is located in the muon chambers. The second stream will be used later to only transfer the pad information in the vicinity of the muon track.

1. It is defined by two words. The first one corresponds to the front-end board number which requires 4 bits. The second word is the RAM number which can be encoded with 5 bits.

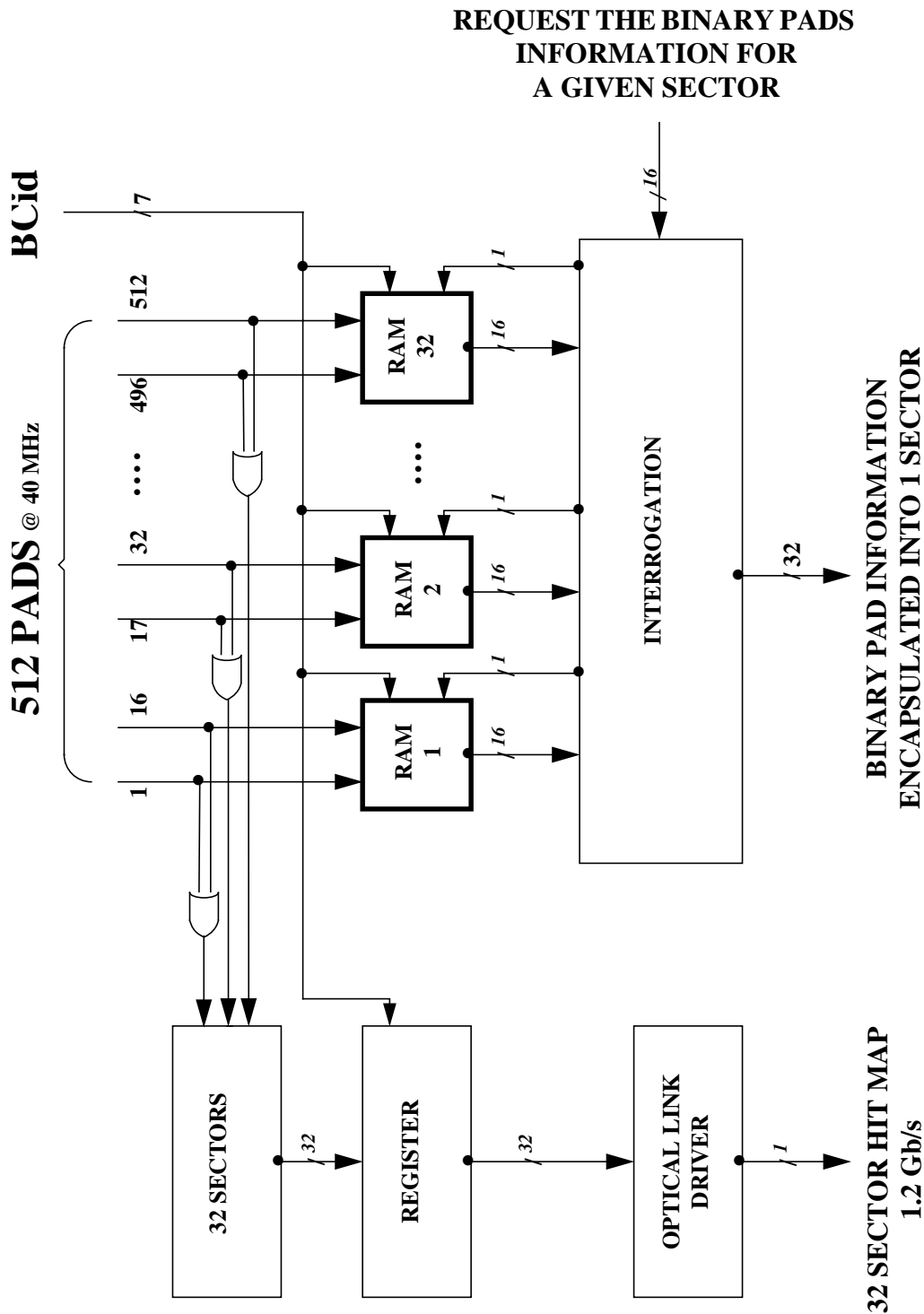


Figure 4: The front-end board interface which sends data to the L0(μ) processor.

The muon chambers μ_3 , μ_4 and μ_5 are read by 16 *front-end board interfaces* each. For the chambers μ_1 and μ_2 this number is 32 since the pad sizes in the X coordinate are smaller by a factor of two.

This data exchange (zero-suppression) scheme has many advantages. It does not depend on the chamber occupancy. It reduces the data transfer between the front-end boards and the muon processor by a factor ≈ 20 . The L0(e) and the L0(μ) processors can interrogate at the same time the front-end board of the chamber μ_1 , if the RAMs are duplicated.

4. The L0(μ) architecture and its implementation

The global L0(μ) processor architecture based on the fast muon identification and the data transfer scenario, is shown in Fig. 5. Beyond the front-end boards described in the previous section, it contains two main components. The first one is the fast *muon identification* processor which runs the fast muon identification algorithm and determine the region of interest close to the muon tracks. The second one is the *muon processing* component which runs the detailed L0(μ) algorithm for fine tracking, and transverse momentum determination.

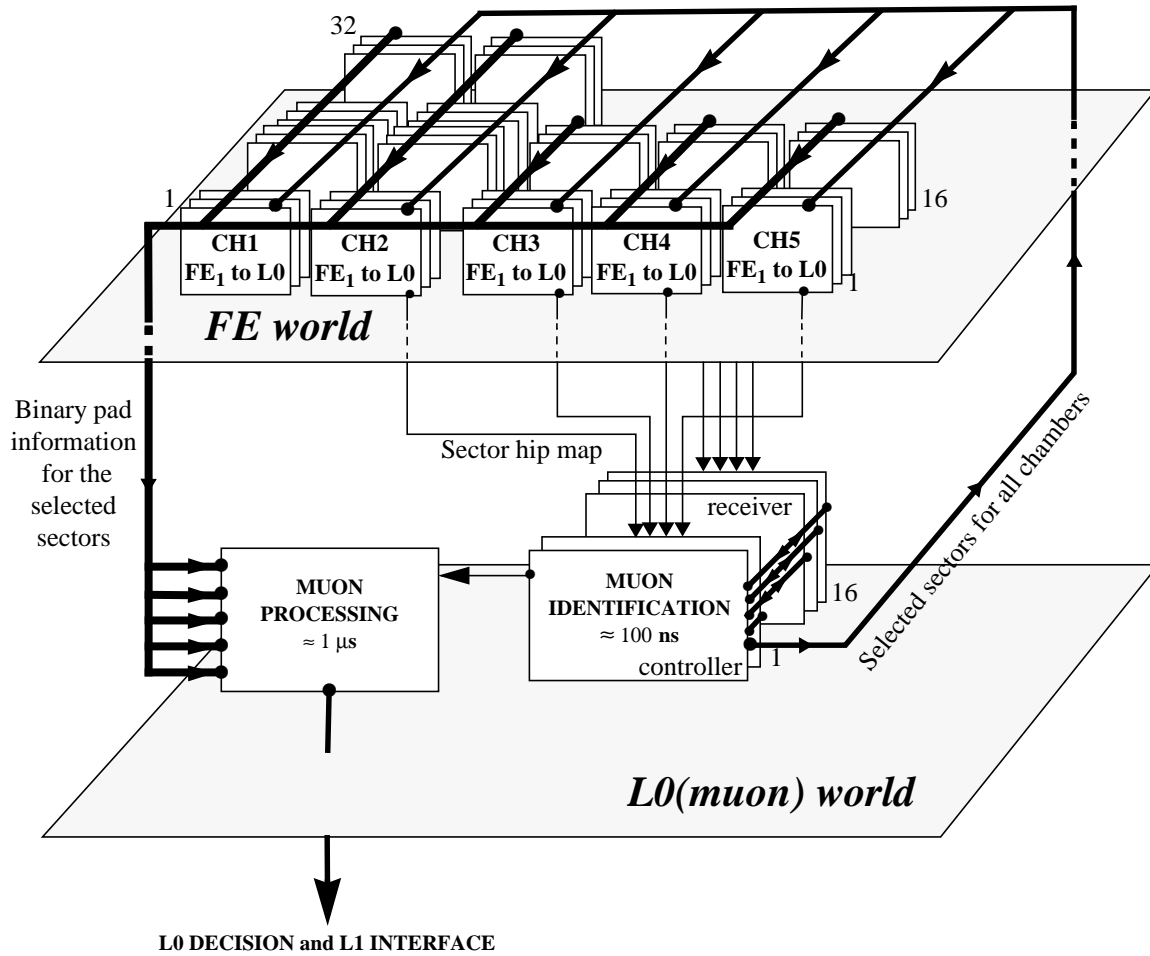


Figure 5: The architecture of the L0(μ) processor.

The architecture of the *muon identification* processor is parallel (16 receivers \times 32 processor devices) and fully pipelined for each beam crossing. It performs a processing step in 25 ns, each step being protected by a register. It is fast and finds muon candidates within 100 ns.

The architecture of the *muon processing* component is different since it works on a variable number of muon candidates per event. It is designed to analyse in the average one muon candidate per beam crossing. FIFO is at the input and output of this stage allow to cope with the statistical distribution of the number of muon candidates per beam crossing. It uses powerful processing devices in order to analyse a muon candidate within 1 μ s. The

architecture of this board is also parallel in order to receive a new muon candidate every 25 ns.

The global L0(μ) processor works in the following way. The *front-end interfaces* send the sector hit maps for chamber μ_2 , μ_3 , μ_4 and μ_5 to the *muon identification* processor. These data transfer is performed for each beam crossing. When the *muon identification* processor finds at least one muon candidate, it returns to the *front-end interfaces* the selected sectors for all muon chambers and the corresponding beam crossing identifier. For these muon candidates the *front-end interface* send back to the *muon processing* section the binary pad information of the selected sectors and the corresponding beam crossing identifier. Fine tracking, transverse momentum and slope measurements are then performed. Muon tracks above transverse momentum threshold are flagged. The muon track list and its parameters are sent to the L0 decision box.

In this architecture, the *muon identification* processor and the *muon processing* board are synchronized on the beam crossing clock. Their internal clock speed is 40 MHz.

The present design is optimum when the average number of $\mu_{\text{CH2-CH5}}$ candidates per beam crossing is equal to one. We can, however, easily deal with the maximum of four $\mu_{\text{CH2-CH5}}$ candidates per beam crossing, by adequately subdividing the muon detector and by increasing the number of processing devices.

4.1 The fast muon identification processor

The fast *muon identification* processor is made of 16 receiver boards and 1 common controller board. The implementation of this processor is related to the hardware implementation of the front-end boards.

To determine the dimension of the system, we used the hardware organization shown in Fig. 6. In that scheme, a muon chamber is read by 16 front-end boards (32 for chambers μ_1 and μ_2). Each front-end board is connected to a maximum of 32 sectors or equivalently to 512 pads.

The architecture of the receiver board is parallel and shown in Fig. 7. It contains 32 devices, S_i , where the fast muon identification is performed. Each device is associated to one sector of the chamber μ_3 .

At the input of the receiver board arrive the sector hit maps coming from chambers μ_2 , μ_3 , μ_4 and μ_5 . The sector hit maps correspond to one of the 16 regions in chamber μ_3 , denoted by 1,2,... in Fig. 6, and the corresponding projections in chambers μ_2 , μ_4 and μ_5 .

At the inputs of the devices S_i , are connected one sector of chamber μ_3 , and the corresponding sectors of chambers μ_2 , μ_4 and μ_5 (the central one and its neighbours) according to the chamber region boundaries. From time to time, the information of the

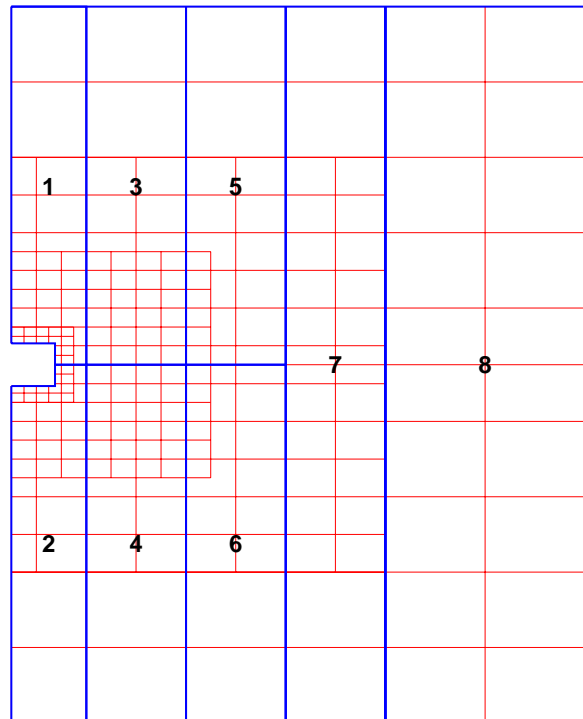


Figure 6: The hardware implementation of the front-end board for half-muon chamber. The thick lines delimit the sectors read by one board. The thine lines show the sectors.

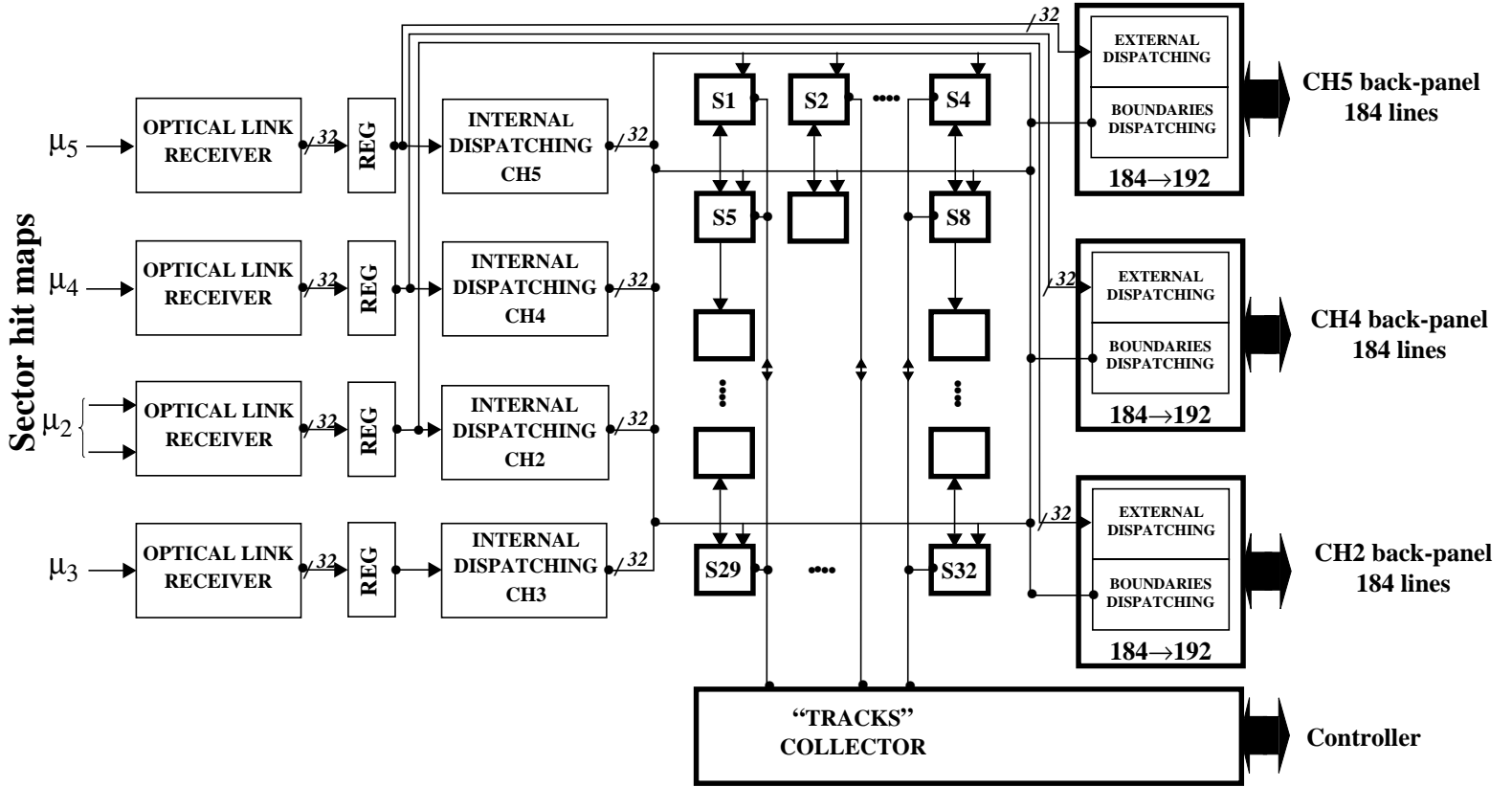


Figure 7: The general scheme of the receiver boards for a half muon detector

neighbouring sectors can be presented on a different receiver board. The informations of sectors shared by several receiver boards are distributed via back-panel lines linking all receiver boards together. In the current implementation we need 184 lines per chamber for a one to one correspondence.

To distribute sector information either on the inputs of the receiver board or on the back-panel lines, we use programmable dispatching. The most demanding one is the device interfacing the back-panel lines. It uses ≈ 400 I/O which can already be accommodated by present technology (FPGA).

The receiver board works in the following way. When all sectors information related to one beam crossing are there, they are transferred into registers. Thus the fast muon identification is performed in parallel for the 448 sectors. In each device S_i , clusters of sectors are built and the logical coincidence between them and the sector in chamber μ_3 is established. It will take 25 ns to get the answers since they are only few AND and OR operations. When a muon candidate is found it is stored into a local FIFO attached to each device S_i .

In the remaining part of the receiver board, the architecture is designed to handle an average of one muon candidate per beam crossing. Each processing step takes 25 ns and is isolated through FIFOs to cope with the statistical distribution.

At the bottom of the receiver board, the “track collector” box knows the pattern of the FIFOs with a muon candidate, for the current beam crossing. It copies the information at the inputs of the device S_i into its own FIFO. This transfer happens 1 over 16 beam crossings in average.

The controller board is common to all receiver boards. It knows the addresses of the receiver boards with muon candidates, for the current beam crossing. For each muon candidate, the controller determines one sector per chamber defining the region of interest close to the muon track.

The selected sectors and the corresponding beam crossing identifier are sent to the *front-end interfaces*. In parallel, the beam crossing identifier and the number of muon candidates are sent to the *muon processing* section.

The presented architecture for the fast *muon identification* processor handles naturally the chamber region boundaries. This property is given by the device S_i which is programmable and by the sector distribution which is flexible.

This architecture is flexible and can deal with most of the front-end/sectors mapping. This is possible due to programmable identification and dispatching devices and the size of buses distributing the sector information between the devices S_i .

The muon identification and the definition of the region of interest close to the muon tracks can be determined in few clock steps. The processing time to perform all these operations is around 100 ns. This processor seems feasible with present technology like FPGA's.

4.2 The muon processing board

The *muon processing* board receives an average of one muon candidate every 25 ns. It receives the binary pad information of the selected sectors and the associated beam crossing identifier. They are fed into one of the 40 devices running in that board.

The general scheme of this board is shown on Fig. 8. Here, also, the architecture is parallel. Each processing device deals with one muon candidate. The fine tracking algorithm is performed at the pad level. The transverse momentum is computed. The track is flagged if its transverse momentum is above the trigger threshold. A powerful processing device is used. It might be a DSP running at 200 MHz or a custom device (ASIC, FPGA,....). It will execute the complete algorithm within 1 μ s.

The *muon processing* board works in the following way. When the data of the first muon candidate are there, the first processing device is launched. It copies the input data

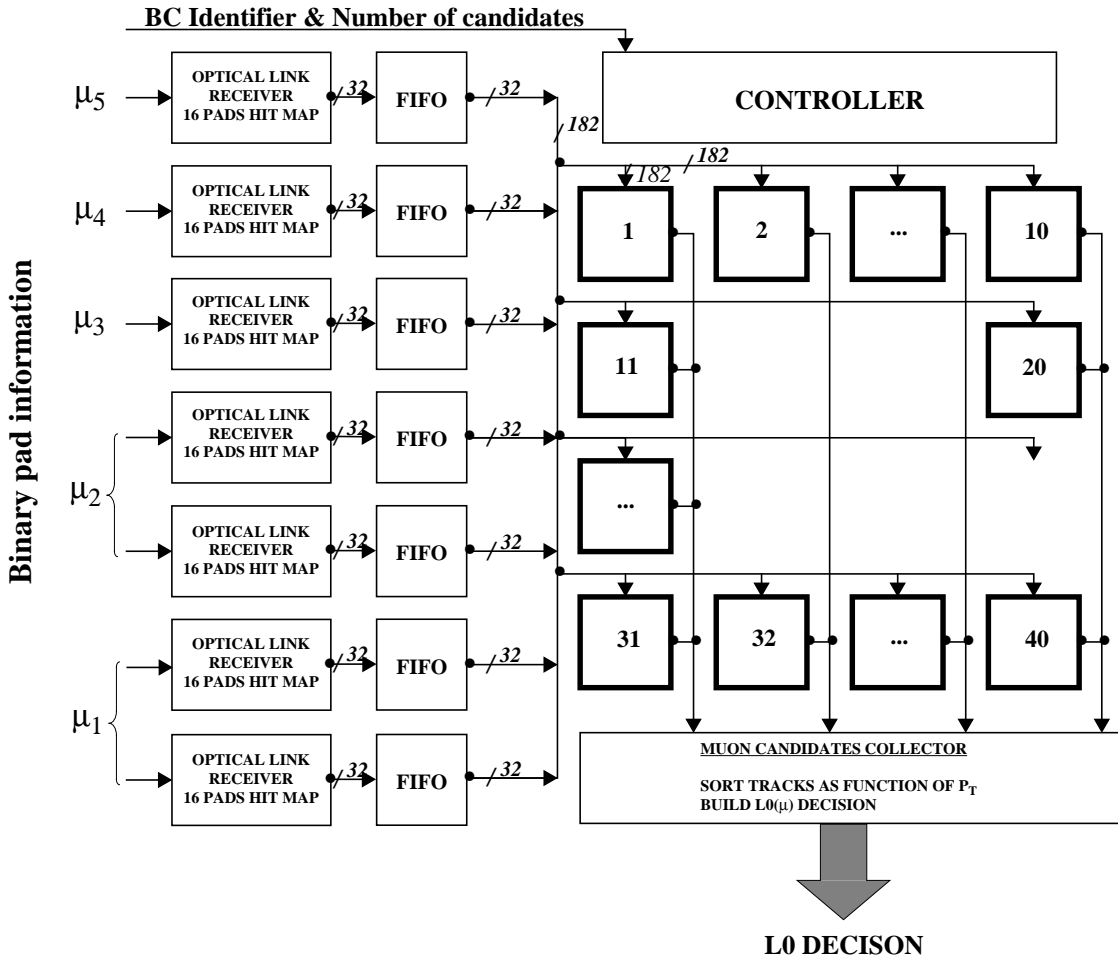


Figure 8: The general scheme of the *muon processing board*

and runs the $L0(\mu)$ trigger algorithm. 25 ns later, the data of the second track are presented. They are taken by the second device. 1.025 μ s later, the data of the 41th track arrives. They are treated by the first processing device which is free at that time.

The “Muon collector” knows the number of muon candidates belonging to each beam crossing and the devices in which they are analysed. At a fixed time, it looks the results at the output of the processing device. It copies the track parameters into its own buffer if they satisfy the trigger criteria. It takes the track with the highest transverse momentum first. Finally, it sends the track information to the L0 decision box.

4.3 $L0(\mu)$ processing time and L0 latency

In this section we review the time spend when an event travels through the $L0(\mu)$ processing chain and we compare it to the L0 latency. This estimation is very crude and strongly dependent on the physical location of the $L0(\mu)$ processor.

In this exercise, we studied two scenarii:

- a) the processor is near the muon detector. It is located in one of the racks which are on the side of the UX85 cavern. This position is marked by the symbol **1** in Fig. 9. The maximum distance between the top corner of the

muon detector and this location is ≈ 45 m. In that option, we have no access to the processor during data taking period.

- b) the processor is in one of the control rooms behind the shielding of the cavern. This position is marked by the symbol ② in Fig. 9. In that case, the maximum distance between the top corner of the muon detector and the processor is ≈ 60 m. In that option, we have always access to the processor.

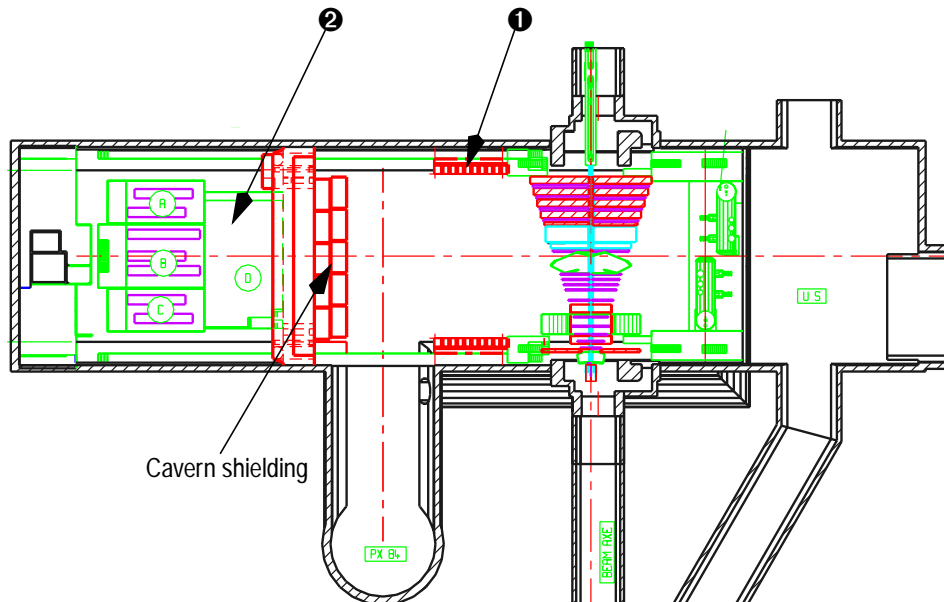


Figure 9: The top view of the UX85 cavern with two possible locations for the $L0(\mu)$ processor.

We also assume that the TTC system is not too far away from the $L0$ decision box. We took 10 meters. In addition we fixed arbitrarily the TTC processing time to 25 ns, the $L0$ decision time to 100 ns and the maximum number of muon candidates to 4. The resulting processing time are summarized in Table 1.

In these two scenarii about half of the time is spend in signal travelling through cables. The total $L0(\mu)$ processing time is smaller than the $L0$ latency fixed to $3.2 \mu\text{s}$. The option where the processor is close to the muon detector gives a safety margin at the level of 1.3 while for the other option it is around 1.2.

	Scenario a)	Scenario b)
Transfer of the sector hit map	225	300
Muon identification processing time		100
Front-end interrogation	225+225	300+300
Final muon processing time	1000 up to 1100	
L0 Decision		100
Broadcast of the L0 Decision	420	500
Total	$\approx 2.3\text{--}2.4 \mu\text{s}$	$\approx 2.6\text{--}2.7 \mu\text{s}$

Table 1: Estimation of the L0(μ) processing time. In scenario a), the processor is close to the muon detector. In scenario b), it is far away and located in the control room D.

5. Conclusions and perspectives

The L0(μ) architecture reported in this note is based on our capability: a) to store binary pad information in FIFOs and at the same time in double access RAMs b) to create at the level of the front-end board sector hit maps c) to identify quickly muon candidates by using the sector information of chamber μ_2 , μ_3 , μ_4 and μ_5 .

The current implementation is simple and compact with a small number of connections. The total number of the L0(μ) processor boards is around 18 and the total number of optical links is ≈ 89 . It does not depend on the chamber occupancy.

This version of the L0(μ) processor is flexible and programmable. By construction it can handle any shape of chamber region boundaries. It might be feasible with current technologies such as FPGA, DSP and optical links. It can deal with up to four $\mu_{\text{CH2-CH5}}$ candidates per beam crossing giving us a safety factor ranging between 4 to 10. Finally the L0(μ) processing time is smaller than the L0 latency by $\approx 20\%$.

To validate this conceptual design, we have to simulate the chaining of the fast muon identification and the standard L0(μ) algorithm, to insure that no physics is lost. We also have to estimate the limitations which might come from physical background and electronics noise.

On the electronic side, we have to simulate the behaviour of this architecture to find hidden vice and to investigate the modifications imposed by the debug, calibration and monitoring functions as well as by the automatic checks.

Acknowledgements

We would like to thank A. Tsaregorodtsev who introduced us to the muon detector and the related software as well as S. Conetti and U. Straumann for their help in understanding the present trigger architecture. This work was supported by the French CNRS/Institut National de Physique Nucléaire et de Physique des Particules and Université de la Méditerranée.

References

- [1] M. Borkovsky et al., LHC-B muon trigger, LHC-B Note, in preparation
- [2] G. Corti and B. Cox, LHC-B muon trigger, LHC-B 97-01
M. Borkovsky et al., Study of the LHC-B muon trigger, LHC-B 97-007
- [3] G. Corti et al., An implementation of the LHC-B level 0 muon trigger using 3D-Flow, LHC-B note, in preparation
- [4] LHC-B Collaboration, Trigger and data acquisition system for the LHC-B experiment, LHC-B 97-008
- [5] See for example transparencies presented by B. Cox in the plenary meeting of the LHC-B Week, 1-5 December 1997
- [6] I. Azhgirey et al., LHC generated muons background on LHC-B formulation of the problem, LHC-B 97-013