LHCb

The L0(μ) processor

LHCb Technical Note

Issue:	1
Revision:	2
Reference:	LHCb 99-008
Created:	15 March 1999
Last modified:	27 April 1999

Prepared By: E. Aslanides, B. Dinkespiler, R. Le Gac, M. Menouni, R. Potheau and A. Tsaregorodtsev

CPPM, IN2P3–CNRS et Université d'Aix–Marseille II

Abstract

In this note we review the Marseille implementation for the L0(μ) processor. We describe the data flow, hardware implementation, synchronization issue as well as our first ideas on debugging and monitoring procedure. We also present the performance of the proposed architecture with an estimate of its cost.

Document Status Sheet

1. Document Title: The L0(μ) processor			
2. Document Reference Number: LHCb 99-008			
3. Issue	**4. Revision**	**5. Date**	**6. Reason for change**
1	1.0	02 April 1999	First Release
	2.0	27 April 1999	Incoporate comments of the trigger group

Table of Contents

Glossary

BC	Bunch Crossing
BC Identifier	The 7 least significant bit of the bunch crossing number
DMP	Detailed Muon Processor
DPRAM	Dual Port Random Access Memory
FEBI	Front End Board Interface
FIFO	First In First Out
FIP	Fast Identification Processor
FPGA	Field Programmable Gate Array
L0(μ)	The implementation of the level zero muon trigger by the Marseille group
Sector	A sector contains 16 logical pads for station M_j where j=3,4,5 and 32 logical pads for stations M_1 and M_2
Section	A section contains 16 sectors
Synchronization Flags	2 bits word containing: a) the LSB of the BC identifier; b) a flag equal to 1 when the BC identifier is equal to a predefined value n.
TTC	Timing, Trigger and Control distribution system by optical fibre

1 Introduction

The level zero muon trigger, L0(μ), identifies a muon track using information of the muon system and measures its transverse momentum. In this status report, we review the architecture proposed in reference [1] following the L0 requirement list.

The Muon system consists of five stations. The first station M_1 is located at 12.15 m from the interaction point while station M_5 is at 18.8 m. The widths of the stations vary between 7.36 m and 11.39 m, the heights between 6.08 m and 9.41 m. A detailed description of the Muon system can be found elsewhere in reference [2].

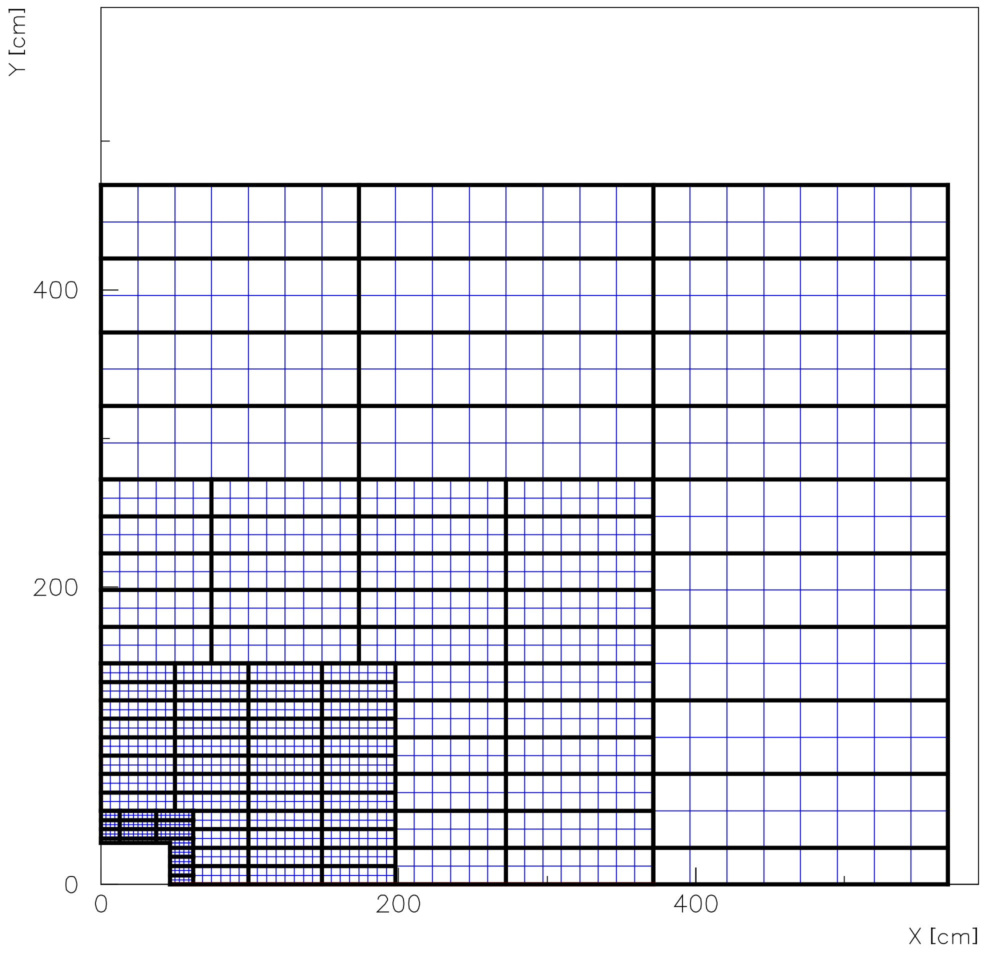


Figure 1 The sector geometry for a quarter of the muon station M_5. The thin lines show the edge of the logical pads while the thick lines delimit the sectors. In addition the 4 *Regions* grouping the logical pads of the same size can be identified.

The basic detector units of a station are *physical* pads. They are ORed to form *logical pads*. The pad layout is shown in Figure 1 for a quarter of station M_5. Station M_j where j=3, 4 and 5 contains 6452 logical pads while stations M_1 (M_2) contain the double since the logical pad size in the x direction is twice smaller. The total number of logical pads amounts to 45164.

A station can be subdivided in the following way:

1. *Regions.*
   A station encloses 4 *Regions*. Each region groups logical pads of the same size. The logical pad size is scaled by a factor 2 in both x and y direction when moving to the upper neighbouring one. The region layout is shown on Figure 1 for a quarter of the station $M_{5.}$
2. *Sectors.*
   A sector encapsulates 16 logical pads of the same size for station M_j where j=3,4,5 and 32 logical pads for stations M_1 (M_2) since the granularity is twice smaller along the x axis. A sector has a rectangular shape: 8 (16) pads along the x axis and 2 pads along the y coordinate. The total number of sectors is 448 for a station. The sector layout is also shown in Figure 1 for a quarter of the station M_5.
3. *Sections.*
   A section contains 16 sectors corresponding to 256 logical pads for station M_j where j=3, 4, 5 and 512 pads for stations M_1 (M_2). The total number of sections is 28 for a station. The organization of the sections in terms of sectors is not yet well established.

There is one to one correspondence between *Regions (Sectors, Sections)* belonging to different stations since the geometry of the logical pad layout is projective.

In this note, we describe the configuration where the Muon system is split into 2 independent parts along the y axis. Each half is connected to a separate L0(μ) processor. This configuration has been chosen to improve the robustness against the physics background.

In the L0 trigger, the data flow has to be aligned with the bunch crossing (BC) number. Such synchronization has to be preserved throughout the L0 processing. To ensure a perfect synchronization in the L0(μ) processor, we defined:

1. *BC Identifier*: the 7 [1] least significant bits of the bunch crossing number.
2. *Synchronization Flags*: reduced information encoded on 2 bits. This word contains the least significant bit of the bunch crossing number and a flag equal to 1 when the *BC Identifier* is equal to a predefined value *n*. The former will allow to detect a synchronization error every 25 ns. The second synchronization flag gives the value of the *BC identifier* every 128 bunch crossings when the BC identifier is encoded on 7 bits.

In the next section of this document, the overall scheme of the data flow is described together with the corresponding hardware implementation. In section 3, some technical solutions are presented. In section 4, the performance of the architecture is given in terms of inefficiencies and latency. Section 5 is devoted to the synchronization issue while, in section 6 our first ideas on debugging and monitoring are presented. Finally, in section 7, the cost is estimated and the prospects are described in section 8.

2 Scheme of the data flow and of the hardware implementation

Our main guidelines looking for a solution to the L0(μ) processor have been to design an architecture:

1. Minimizing the trigger efficiency losses with respect to the “theoretical” algorithm.
2. Minimizing the number of links between the FE electronics and the L0 hardware and minimizing the data exchange between the L0 boards.
3. Simple, compact, flexible and easy to debug, monitor and repair.
4. Evolutive as far as the physics requirements and the technological progress are concerned.

To fulfil these requirements, the proposed architecture is organized around three main sub-systems:

1. *Fast Identification Processor* (FIP)
   The FIP processor localizes the muon track by using a coarse

(1) The number of bits to encode the *BC Identifier* is defined by the depth of the L0 buffer. In this note 7 bits are used since the depth of the L0 buffer is equal to 128 in the Technical Proposal [2]. In the future, we will use 8 bits since the depth of the L0 buffer will increase up to ~140.

information, namely, the sectors. At the end of this stage, a FIP candidate is defined by sector addresses.

2. *Interrogation of the FE electronics*
   The sector addresses for each FIP candidate are sent to the FE electronics which returns logical pad hit maps encapsulated in the selected sectors.
3. *Detailed Muon Processing* (DMP)
   The DMP processor performs fine track finding in the five stations of the muon system, using the pad hit maps returned by the FE electronics. It looks for a muon track within predefined fields of interest, computes the transverse momentum for each candidate and builds up a decision for each bunch crossing.

In addition to the FIP and DMP processors, the *Front End Board Interface* (FEBI) boards were introduced as part of the FE electronics. They are connected to the 45164 logical pads and should contain L0, L1 buffers, the corresponding derandomizer [3] and an interface to exchange data with the L0(μ) trigger.

2.1 The FIP algorithm

The inputs to the FIP algorithm are the sector hit maps for the four stations: M_2, M_3, M_4 and M_5. The starting point of the FIP algorithm is given by a sector hit in station 3. This sector is defined as a *central sector.*

We look for the corresponding central sectors in stations M_2, M_4 and M_5. There is one to one correspondence between sectors belonging to different stations since the geometry of the sectors is projective.

Four cases are possible:

1. The central sectors S_i are hit in stations M_2, M_3, M_4 and M_5.
   A FIP candidate is found
2. The central sector is not hit in station M_j ($j \neq 3$) but hit in the others. In that case we scan the *neighbouring sectors* in station M_j. Their number depends on the station and on the position of the central sector within a region. It is equal to 8 when the central sector is inside a *Region*, containing pads of the same size. However, it can increase up to 12 when the central sector is located on a *Region* boundary. For station M_2, it varies between 2 and 4. They correspond to the neighbouring sector

along the *x* coordinate. Top and bottom sectors are not taken into account since they are always outside the field of interest applied in the DMP algorithm. A FIP candidate is found if one and only one of the neighbouring sectors is hit.

3. Central sectors are not hit in station M_j ($j \neq 3$) and M_k ($k \neq 3$). For each of them, we apply the previous recipe.
4. The central sector is hit in station M_3 but not in the other stations. In that case, the FIP candidate is rejected.

At the end of this stage, the FIP candidate is defined by 4 sector addresses corresponding to the sectors crossed by the muon in stations M_2, M_3, M_4 and M_5.

Sector hit maps of station M_1 are not taken into account by the FIP algorithm since the high occupancy of station M_1 does not improve the muon localization. However, the binary pad information is mandatory to measure accurately the transverse momentum of the muon track. To obtain the pad information at the DMP stage, we find by extrapolation the sector in station M_1 which might be crossed by the muon. Up to 4 sector addresses can be possibly identified depending on the boundary conditions. They correspond to the central sector in station M_1 with its neighbouring sectors along the *x* coordinate. The top and bottom neighbouring sectors are not taken into account since they are always outside the field of interest used in the DMP algorithm.

At the end of the FIP algorithm, a FIP candidate is defined by up to 8 sector addresses: one for each station M_j where j=2,...,5 and up to 4 for station M_1. They are used to interrogate the FE electronics.

If the number of FIP candidates exceeds the maximum value authorized (~12), the bunch crossing is not accepted by the L0(μ) decision.

2.2 The DMP algorithm

The DMP receives the *pad hit maps* of the sectors selected by the FIP processor. The starting point of the algorithm is given by a pad hit in the station M_3.

From the pad position in station M_3, the position of the track is extrapolated to stations M_2, M_4 and M_5 assuming a straight line coming from the interaction point. There is one to one correspondence between the pad hit in station M_3 and the extrapolated ones since the layout of the logical pads is projective.

In stations M_4 and M_5, we look for at least a hit within a field of interest centred on the extrapolated pad. In station M_2, we look for the closest hit to the extrapolated pad within the field of interest. The size of these fields depends on the station. They are summarized in Table 1.

Table 1 Size of the field of interest. The centre of the field is given by the position of the pad extrapolated. The units are pads.

Station	**Size of the field of interest**	
	x coordinate	*y* coordinate
M_1	±8.0	±0.5
M_2	±2.5	±0.5
M_3	–	–
M_4	±1.5	±1.5
M_5	±2.5	±1.5

When the muon identification is achieved, the procedure to measure the transverse momentum is launched. Firstly, the position of the track is extrapolated to station M_1, assuming a straight line based on the pad positions in stations M_2 and M_3. Then, the nearest pad hit is searched within the field of interest. Secondly, the transverse momentum of the found track is computed using the pads position in stations M_1 and M_2.

2.3 Implementation and data flow (Road Map)

In this section, the hardware implementation and the corresponding data flow are described. They are driven by the FIP and DMP algorithms and by the following considerations:

1. The station geometry is based on the pad layout described in [2].
2. The depth of L0 buffer is 128 [2].
3. The L0(μ) decision is generated at fixed time for each bunch crossing.

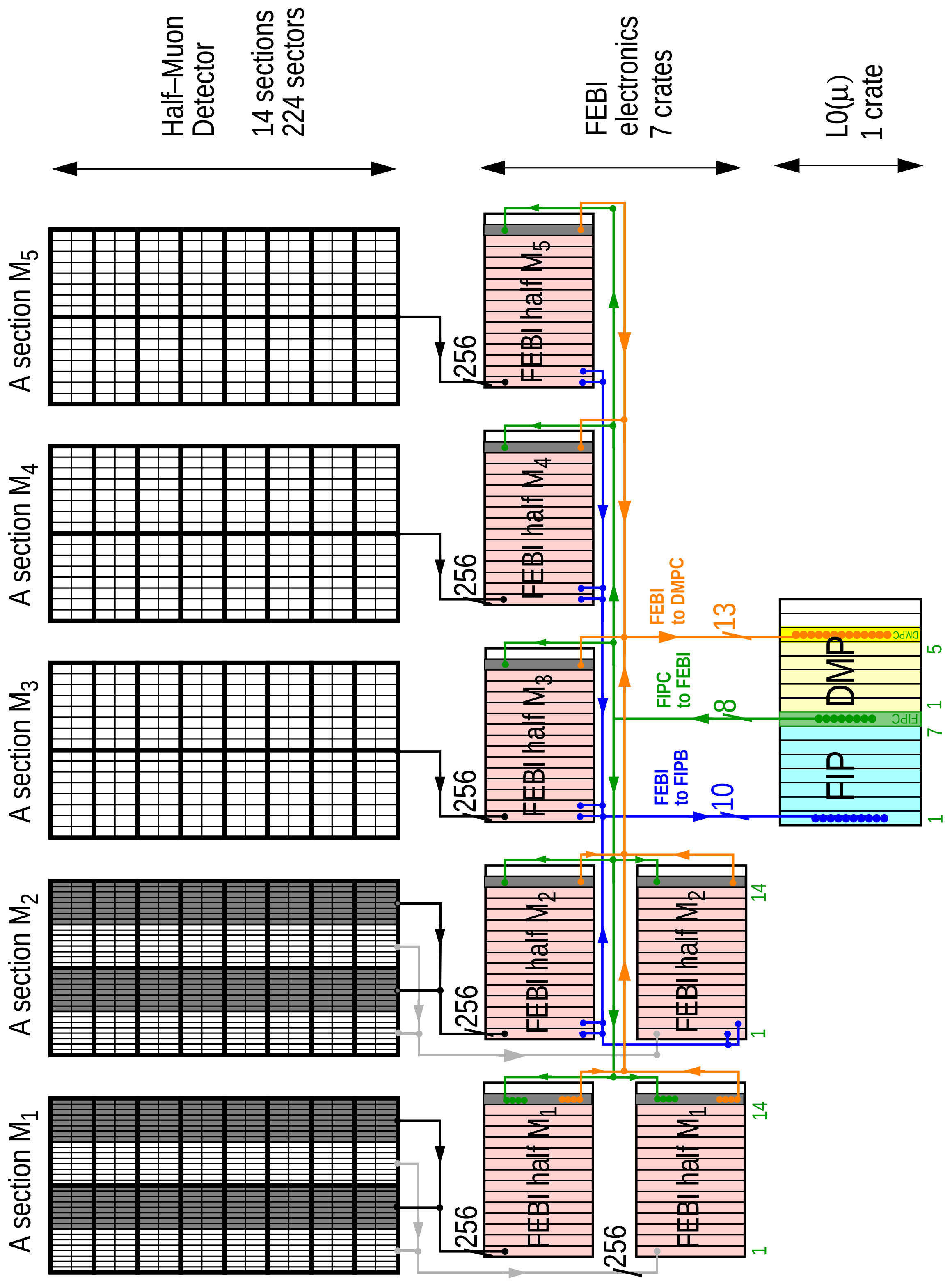


Figure 2 Overview of the L0(μ) trigger, for half of the muon detector. The following data flow are shown between: a) a section and FEBI boards; b) FEBI's and a FIP board; c) between the FIP and FEBI controllers; d) between the FEBI and DMP controllers.

4. The maximum number of FIP candidates, the DMP time out and the size of the fields of interest are not frozen in the hardware. However for stations M_1 the size of the field of interest does not depend on the y coordinate. It is fixed to ±0.5 pad.
5. The x and y position of a pad is encoded on 10 bits each. The transverse momentum of a muon is encoded on 8 bits.
6. The electronics is installed in VME64xP crates [4].

An overview of the implementation for half of the muon detector is shown in Figure 2. It will be discussed step by step in the next sub-sections.

2.3.1 From the detector to the FEBI's

In the current scenario of the Muon system layout, the logical pads are produced in the off-detector electronics located on the left and on the right side of the station by ORing the physical cells of the muon chambers. At this stage the *BC identifier* should be attached to the pad. The 45164 pads together with either the *BC identifier* or simplified *synchronization flags* are sent behind the shielding wall to the FEBI's electronics, via ~1 Gb / s optical links.

The logical pad data transfer has to be arranged in such a way that the input of a FEBI board corresponds to pads of 16 *sectors* belonging to the same *section*. We assume that such data grouping can be implemented.

At the input of a FEBI board 256 pads arrive together with their BC identifier, for each bunch crossing. If the BC identifier is not attached to the pads in the off-detector electronics, it can be either regenerated from the synchronization flags or associated to the pad information at this stage. The pads are stored in the L0 buffer and distributed to the L0(μ) interface with their BC identifier. We assume that both can be located in the same board with the L1 buffer and the derandomizers.

In this frame work, a *section* is read by one FEBI board for station M_j where j=3,4,5. Two FEBI boards are required for stations M_1 (M_2). Half of the pads belonging to a sector go to the first board while the rest is connected to the second one. They are located in different crates. This scheme simplifies the L0(μ) interface since the 2 parts of the sector are at the same address within their crates.

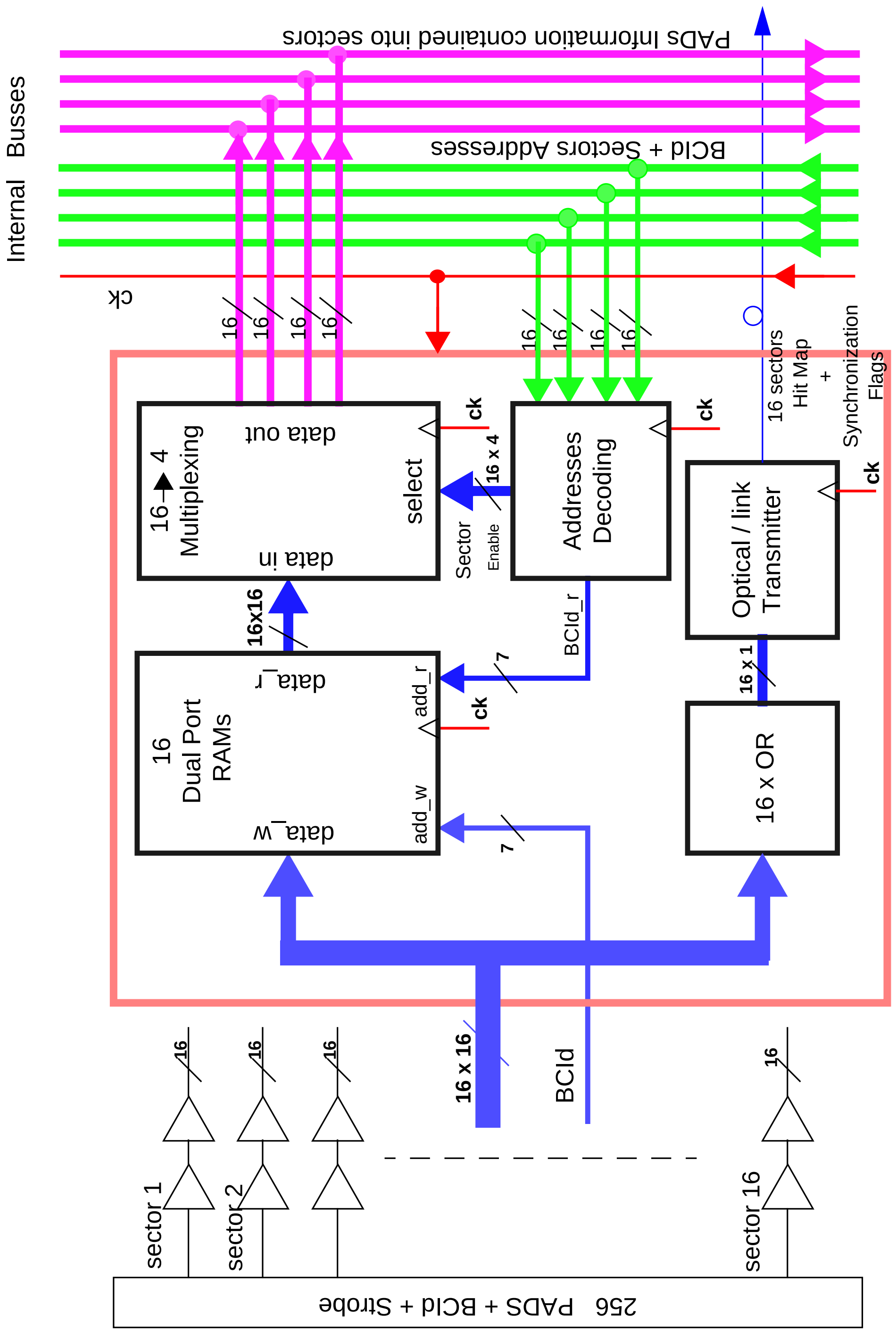


Figure 3 Overview of the FEBI interface to the L0(μ) hardware. The 16 ORs forming the sector hit map are only implemented for stations M_j where j=2,...,5. One sector is addressed every 25 ns for stations M_j where j=2,...,5, while 4 sectors are read simultaneously for station M_1.

In the FEBI electronics the L0(μ) interface is composed of two parts. The first one is implemented in each FEBI board while the second one is a standalone board named FEBI Controller. This controller talks to the interfaces implemented in each FEBI board. The two parts exchange data using a custom bus implemented in the VME64 back plane. For one half of the Muon system, there is 1 FEBI controller for each station M_j where j=3,4,5 and 2 for station M_1 or M_2.

2.3.2 From the FEBI's to the FIP

The FEBI interface to the L0(μ) processor is shown in Figure 3.

For each bunch crossing, 256 pads arrive with their BC identifier. In one hand, the binary pad information is stored in a dual port DPRAM. The address is given by the BC identifier. In the other hand, the 16 pads of a given sector are ORed to determine whether the sector is hit or not, for station M_j ($j \neq 1$). Then, the hit map for 16 sectors is sent to a FIP board with the *synchronization flags*, for each bunch crossing.

2.3.3 From the FIP board to the FIP controller

The FIP processor analyzes the data coming from one half of the Muon system. It is composed of 7 FIP boards and 1 FIP controller. The FIP controller and the FIP boards exchange data using a custom bus. The FIP boards exchange data between themselves through custom connections. The custom bus and connections are embedded in the VME64 back plane. The pattern of the interconnection depends on the *section* layout. It will not be frozen in the hardware since we use programmable devices. In addition, a sufficient number of interconnection will be provided to accommodate any kind of reasonable geometry.

A scheme of the FIP board is shown in Figure 4. It analyzes 32 sectors of station M_3 and the corresponding sectors in stations M_2, M_4 and M_5 in parallel, for each bunch crossing.

At the input of a FIP board, there are ten 20 bits words containing 16 sectors hit maps for stations M_2, M_3, M_4, M_5 and synchronization flags. They correspond to the same bunch crossing and to two *sections* of a muon station. They enter in FIFOs which are required to synchronize data belonging to different *sections*

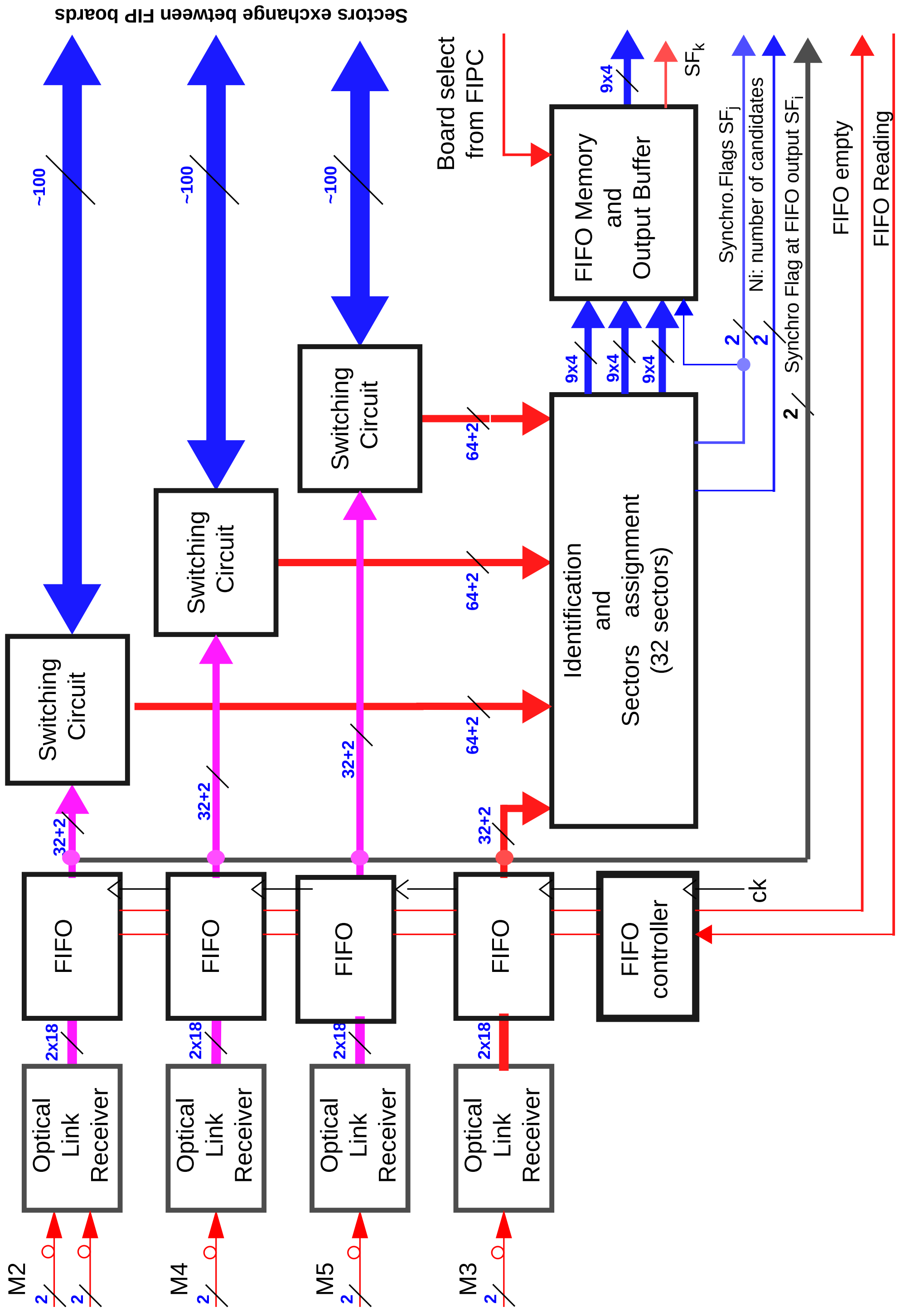


Figure 4 Overview of the FIP board

and to different stations. FIFOs are read with a master clock running at 40 MHz. Sectors shared by several FIP boards are distributed via the custom back plane.

Sectors belonging to the two sections and their neighbouring ones coming from the back plane are fed to the *Identification Processor*. The *Identification Processor* runs the first step of the FIP algorithm. For each bunch crossing, the 4 sector addresses defining a FIP candidate are memorized for all the candidates and stored in a FIFO. In the current design the maximum number of the FIP candidates per FIP board and per bunch crossing is limited to 3.

For each bunch crossing and for each FIP board, the numbers of FIP candidates found are read by the FIP controller via the custom bus and stored together with the BC identifier. Thus, the FIP controller knows for each bunch crossing the number of FIP candidates and the FIP boards in which they are located. Up to this stage the architecture is fully parallel and pipelined. To guarantee synchronization up to the end, a 16 bits word is sent to the DMP processor, for each bunch crossing. It contains the number of FIP candidates (4b), the corresponding BC identifier (7b) and error flags (5b).

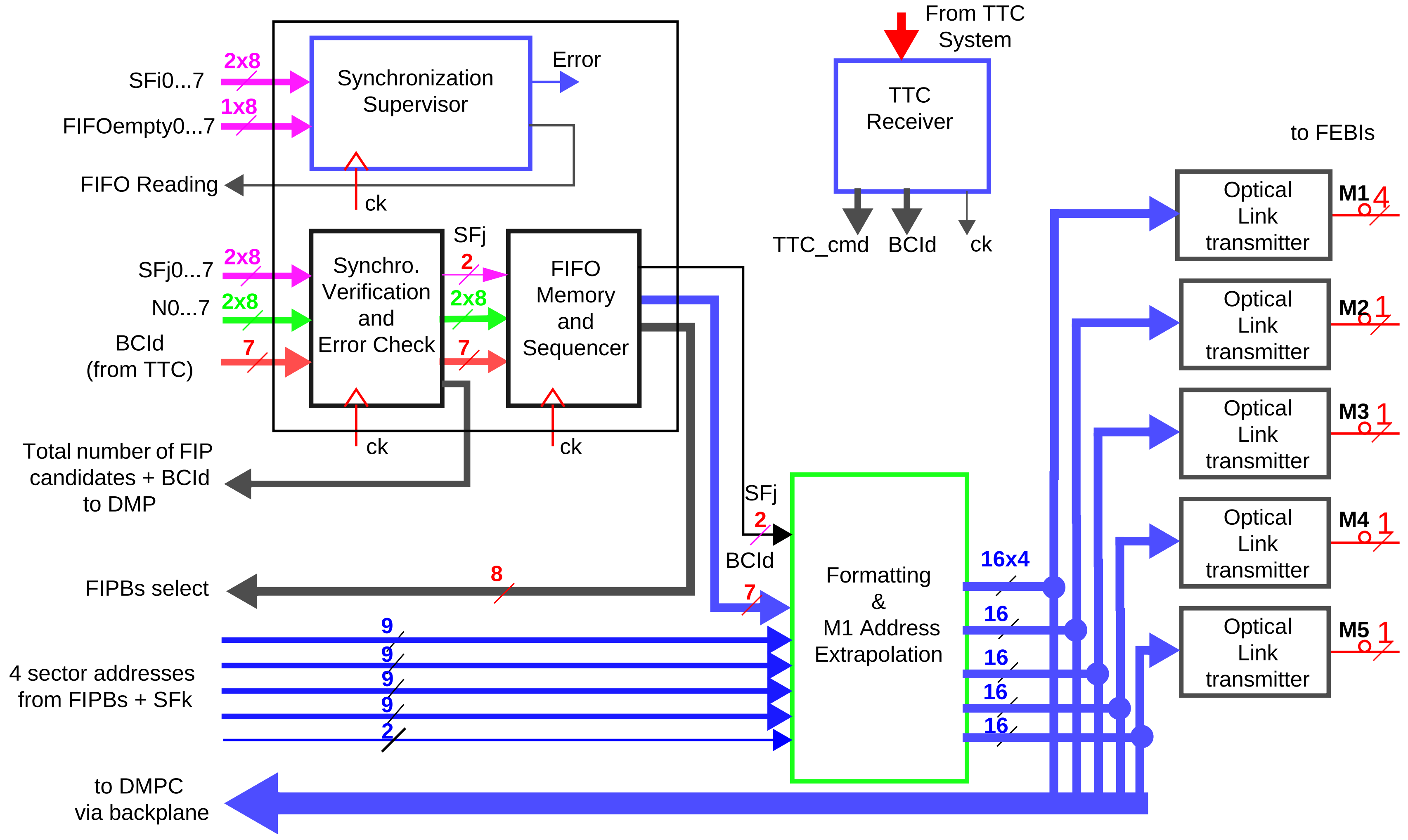
The FIP controller, shown in Figure 5, reads in parallel the 4 sector addresses defining a FIP candidate and its synchronization flags. The FIP candidates information is transmitted one after the other preserving the bunch crossing order. The transfer frequency is set to 40 MHz. At this stage the data flow is not synchronized with the bunch crossing any more.

For each candidate, the FIP controller regenerates the BC identifier from the synchronization flags, performs the sector extrapolation to station M_1. Then, eight 16 bits words are transmitted to the FEBI controllers and copied to the DMP controller. They contain the sector addresses (9b) and the corresponding BC identifier (7b). They are sent for all five stations at the same time. For stations M_1 (M_2) the data are duplicated to the 2 FEBI controllers.

If the total number of FIP candidates exceeds the maximum number of FIP candidates authorized or if the number of FIP candidates is equal to zero, the bunch crossing is not accepted by the L0(μ) decision. In this case, nothing is sent to the FEBI controllers.

2.3.4 From the FIP controller to the FEBI Controller

A FEBI controller receives, every 25 ns, a 16 bits word with a sector address (9b) and its BC identifier (7b), for stations M_j where j=2,...,5. At the same time, it receives four 16 bits words for station M_1.


SFi0...7
2x8
FIFOempty0...7
1x8
FIFO Reading
Synchronization Supervisor
Error
ck
From TTC System
TTC Receiver
TTC_cmd
BCId
ck
to FEBIs
SFj0...7
2x8
N0...7
2x8
BCId (from TTC)
7
Synchro. Verification and Error Check
SFj
2
2x8
7
FIFO Memory and Sequencer
ck
ck
Total number of FIP candidates + BCId to DMP
FIPBs select
8
SFj
2
BCId
7
Formatting & M1 Address Extrapolation
16x4
16
16
16
16
4 sector addresses from FIPBs + SFk
9
9
9
9
2
to DMPC via backplane
Optical Link transmitter
M1
4
Optical Link transmitter
M2
1
Optical Link transmitter
M3
1
Optical Link transmitter
M4
1
Optical Link transmitter
M5
1

We now describe the data flow for station M_1 which is the most demanding one, since we treat 4 sectors in parallel. The FEBI controller writes the 4 sector addresses and the corresponding BC identifier on the custom bus connecting the FEBI controller to the FEBI boards. As shown in Figure 2, FEBI boards decode the sector addresses. If they correspond to those handled by the board, the on-board DPRAM is read for each sector. The memory address is given by the BC identifier. In any case, it is different from the current one since the depth of the DPRAM covers 3.2 µs. Up to 4 sectors can be read at the same time by a FEBI board. The 16 *pads hit maps* for the selected sectors are written on the custom bus in parallel. We use predetermined connections to avoid collisions when 2 boards write on the bus simultaneously (Pad hit map of sector S_k where k=1,...,4 is put on the group of lines k of the bus).

The FEBI controller receives in parallel the pad content of the 4 sectors. It attaches the corresponding BC identifier. Every 25 ns, the four 32 bits words are sent in parallel to the DMP controller. They contain the BC identifier (7b), the sector address (9b) and the pad hit map of the sector (16b).

2.3.5 From the FEBI's to the DMP decision

The DMP processor analyzes data coming from one half of the muon system. It is composed of 6 DMP boards and 1 DMP controller. DMP controller and DMP board exchange data using a custom bus. For half of the muon system the FIP and the DMP processors are installed in the same VME64 crate. They exchange information via a dedicated bus. The custom buses are embedded in the VME64 back plane.

A scheme of the DMP controller is shown in Figure 6.

For each bunch crossing, the DMP processor receives:

1. A 16 bits word containing the number of FIP candidates (4b), the BC identifier (7b) and error flags (5b). This information is synchronized with the bunch crossing. The back plane of the crate is used to transport this word between the FIP controller and the DMP controller.
2. For each FIP candidate, a copy of the eight 16 bits words is sent by the FIP controller to the FEBI controllers. For each station, it contains the sector address (9b) and the BC identifier (7b). The eight words arrive in parallel, every 25 ns. This information is not synchronized with the bunch crossing, since the FIP candidates are treated one after the other.

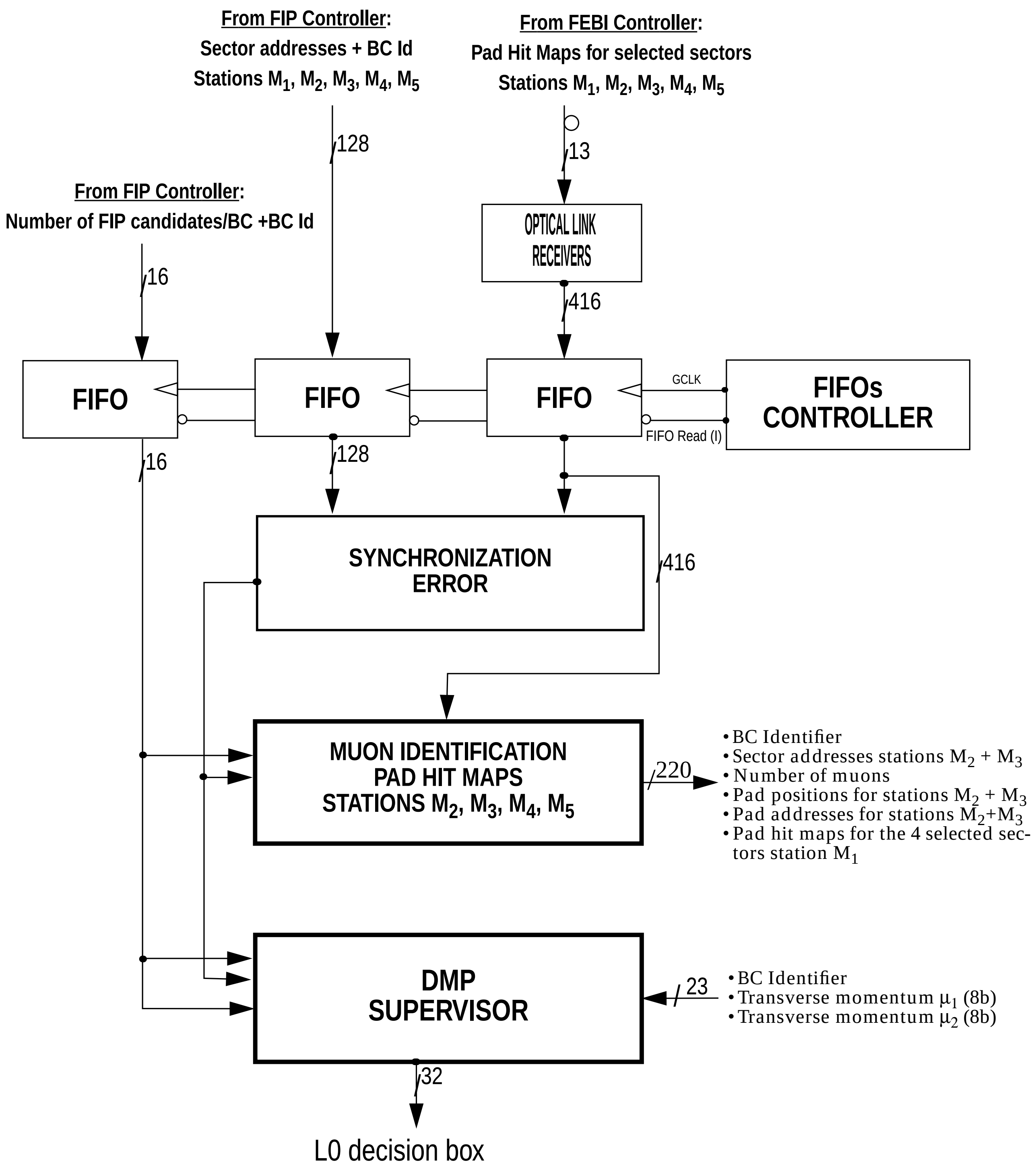

From FIP Controller:
Sector addresses + BC Id
Stations M1, M2, M3, M4, M5
From FEBI Controller:
Pad Hit Maps for selected sectors
Stations M1, M2, M3, M4, M5
128
13
From FIP Controller:
Number of FIP candidates/BC +BC Id
OPTICAL LINK RECEIVERS
16
416
FIFO
FIFO
FIFO
GCLK
FIFOs CONTROLLER
FIFO Read (l)
16
128
SYNCHRONIZATION ERROR
416
MUON IDENTIFICATION PAD HIT MAPS STATIONS M2, M3, M4, M5
220
•BC Identifier
•Sector addresses stations M2 + M3
•Number of muons
•Pad positions for stations M2 + M3
•Pad addresses for stations M2+M3
•Pad hit maps for the 4 selected sectors station M1
DMP SUPERVISOR
23
•BC Identifier
•Transverse momentum μ1 (8b)
•Transverse momentum μ2 (8b)
32
L0 decision box

Figure 6 Overview of the DMP controller

The back plane of the crate is used to transport these words between the FIP controller and the DMP controller.

3. Later on, for each FIP candidate, thirteen 32 bits words containing the pad information encapsulated in a sector (16b), the sector address (9b) and the BC identifier (7b). These words arrive in parallel for the five stations. This information is not synchronized with the bunch crossing, since the FIP candidates are treated one after the other at a frequency of 40 MHz.

The input data coming from the FIP controller and from the FEBI controllers are stored in FIFOs. They are required to synchronize data belonging to different stations and to accommodate delay differences between data coming from the FIP controller and the FEBI controllers. FIFOs are driven by a master clock running at 40 MHz.

At the output of the FIFOs, the DMP controller checks that requests sent by the FIP processor to the FEBI electronics are in agreement with the answers returned by the FEBI controllers. If an error is detected, an appropriate action is taken.

Every 25 ns, for each FIP candidate, the muon identification is performed using the pad information encapsulated in the selected sectors of stations M_2, M_3, M_4 and M_5. It is performed in parallel for all the 16 pads contained in the M_3 sector. At the end of this stage, we know the number of muons crossing the M_3 sector. Currently, it is limited to two. For each muon we also know the pad positions and addresses in stations M_2 and M_3.

The DMP supervisor activates the first DMP board which can accept a muon candidate. The next FIP candidate arrives 25 ns later at the input of the DMP controller. The corresponding muon candidate is distributed to the next DMP board available. The DMP boards are activated in a circular way.

For each muon candidate, a 220 bits word is transferred into the DMP board register via the back plane bus. It contains: the BC identifier (7b), the sector address for station M_2 and M_3 (18b), the number of muons (2b), the pad position for station M_3 and M_2 for the 2 possible candidates (40b), the relative address of the pad in its sector for stations M_2 and M_3 for the 2 possible candidates (24b), the pad hit maps encapsulated in the selected sectors of station M_1 (128b).

The architecture of the DMP board is pipelined. It analyzes two muon candidates in parallel. Four actions are executed successively:

1. *Pad extrapolation to station M_1.*
   The position of the muon track in station M_1 is computed assuming a straight line using the pad position in station M_3 and M_2.
2. *Conversion of the pad position into a pad address.*
   The pad position extrapolated to station M_1 is converted into a sub-sector address ranging from 1 to 4. The relative address of the pad in its sector is also determined.
3. *Extraction of the nearest pad hit.*
   The latter information and the pad hit maps for station M_1 for the 4 selected sectors are used to find the nearest hit to the extrapolated one. It is performed in parallel for the 128 pads contained in the 4 sectors. Then the address of the pad found is converted into a pad position. The candidate is rejected if there is no hit within the field of interest.
4. *P_T computation.*
   From the pad position in station M_1 and M_2, a look-up table return the transverse momentum of the tracks. It is encoded as an 8 bits word.

At the end of the DMP board processing, a 23 bits word is sent to the DMP supervisor located in the DMP controller board. The word contains: the transverse momentum of the 2 possible muons and their BC identifier.

The main tasks of the DMP supervisor are to re-synchronize the muons information with their bunch crossing, to make a decision for each bunch-crossing and to send it to the L0 decision box. The DMP supervisor knows the number of FIP candidates for each bunch-crossing, the BC identifier, the correspondence between a FIP candidate and a muon track as well as the transverse momentum of each track. It sorts the P_T information to find the 2 tracks with the largest P_T values and applies the P_T cuts.

The DMP supervisor makes a decision for a given bunch crossing, as soon as possible. The decision can be taken: a) very quickly if there are no FIP candidates; b) when none of the FIP candidates are identified as a muon; c) when the last muon is processed. The answer is written to a circular buffer at the address given by the BC identifier. To avoid conflict access appearing when two providers write in the circular buffer at the same time, each provider is connected to its own circular buffer.

To guarantee a fixed processing time for each bunch crossing, circular buffers are read after a fixed time after the collision. It takes into account: the time to process a candidate, delays to accommodate the arrival time of the last FIP candidate to the DMP processor, and a margin to handle statistical fluctuation in the number of FIP candidates. If the decision for a bunch crossing is not ready by the moment when the circular buffers are read, the bunch crossing is not accepted. An adequate error message is sent to the L0 decision box. In addition, the DMP supervisor stops and frees DMP boards processing the muon candidates of the corresponding bunch crossing.

Finally, a 32 bits word is sent to the L0 decision box for each bunch crossing. It contains: the BC identifier (7b), two values of the transverse momentum (16b), a decision flag (1b) and some error flags (8b).

3 Technical solutions envisaged

To determine the dimensions of the system, the processing time and to evaluate its feasibility, we simulated the behaviour of the basic components of the L0(μ) processor using the VERILOG language. For example, we studied: the FEBI interface, the identification processor in the FIP board, the dialogue between the FIP board and the FIP controller, the DMP muon identification,… In addition, they were fitted in available FPGA chips (ALTERA FLEX 10K logic family).

In the following sub-sections we briefly summarize the results of these studies.

3.1 FEBI's

The number of FEBI boards is determined by the VME64xP standard. An overview of a possible FEBI board design following this standard is shown in Figure 7.

The 9U VME board is connected to the back plane containing a VME bus and 479 user definable pins [4]. To maximize the space in the VME board in order to place the L0, L1 buffers, the corresponding derandomizers and the L0(μ) interface, all the input connectors and their receivers are installed in the *Transition Module*.

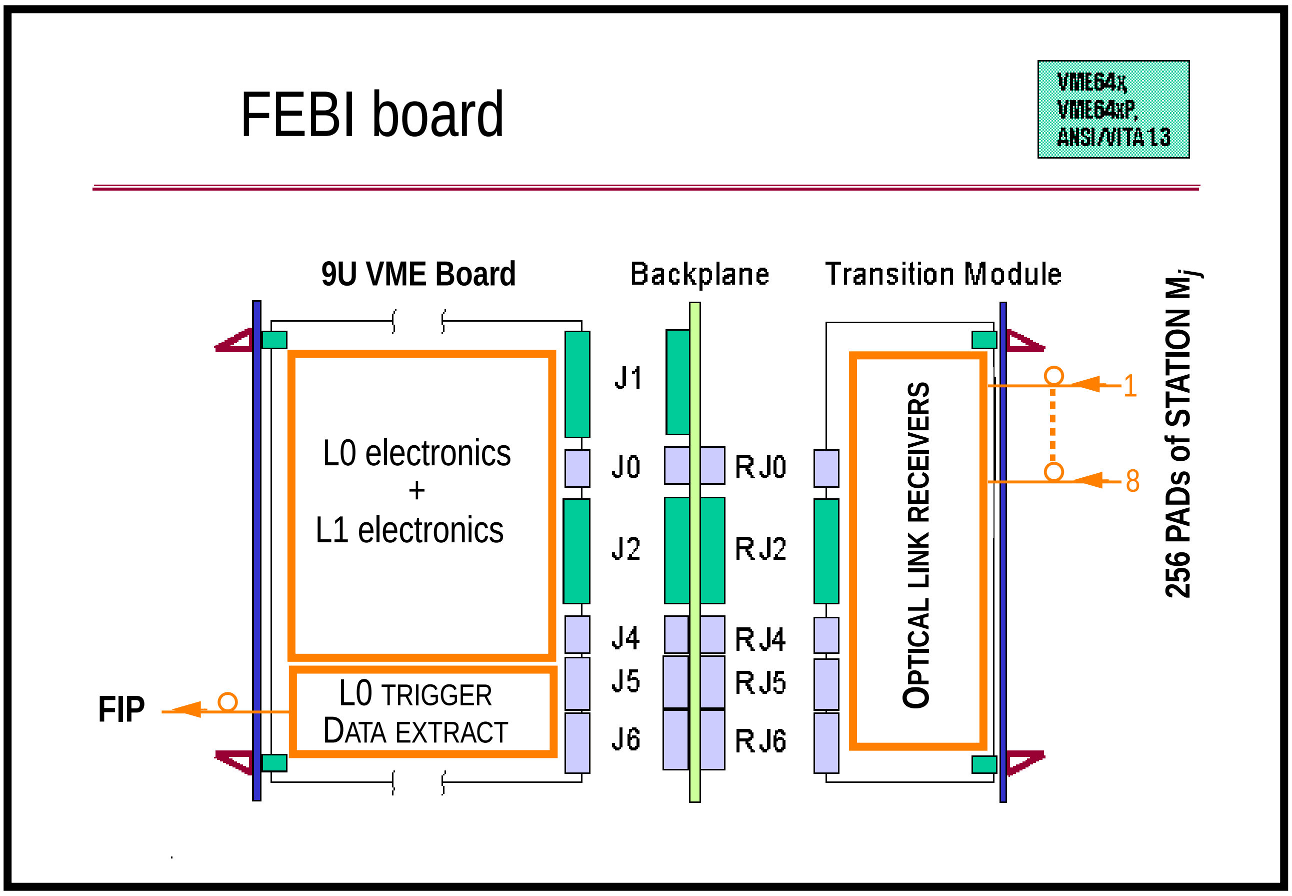


Figure 7 Overview of the FEBI board within the VME64xP standard

In this configuration, only 256 pads can be read by one board. The main limitation is due to the number of user pins available since they are used to transfer the binary pad information from the transition module to the VME board. In addition, 128 user pins are required for the custom bus for the dialogue between the FEBI boards and the FEBI controller.

The VME bus is used for the setting and for the monitoring of the FEBI electronics.

The simulation of the L0(μ) interface electronics showed:

1. Provided that DPRAM is built in FPGA chips, every 25 ns we can simultaneously: a) write 256 binary pad information; b) read the pad contents of 4 sectors.
2. 4 ALTERAs *EPF10K130* (~530×10^3 gates, 16 memory blocks, 470 user I/Os) are required to build the L0(μ) interface in a FEBI board for

station M_1. They will cover about 20% of the VME64 9U board including the link driver.

3. The processing time to read out 4 sectors at once is 2 clock cycles.

In the proposed implementation, the FEBI electronics is placed very close to the FIP and to the DMP processors. The distance between the FEBI boards and the FIP boards is less than 5 m. To transmit data from one FEBI board to one FIP board, we intend to use serial links. Therefore, we have to accommodate ten ~1 Gb/s links on a single FIP board and fourteen 1.6 Gb/s links on the DMP controller. Two solutions can be envisaged:

1. Copper link LVDS [5]. 21 or 28 bits at 40 MHz are serialized and transmitted on 3 or 4 twisted pair cables at 280 MHz. The length of the cable is strongly limited to 5 meters.
2. Optical link GLINK [6], carrying: a) 16 or 20 bits at 40 MHz b) 32 bits at 40 MHz, using double frame mode. For the former a low power consumption chip might be available in 1999.

The time overhead to serialize the data is of the order of 50 ns. An additional delay of ~50 ns is required to deserialize the data.

For one half of a muon station M_j where j=3,4,5 about 14 FEBI boards and 1 FEBI controller are required. The exact number of the FEBI boards depends on the layout of *sections* which is not well defined yet. They can be installed in one VME64 crate. Half of station M_1 (M_2) requires 2 crates, each of them containing 1 FEBI controller. Thus, 14 VME64 crates are required to house the FEBI electronics of the Muon system.

In total, the number of links between the FEBI electronics and the L0(μ) hardware is 212 with a bandwidth of 1 Gb/s each (20 data bits + 4 control bits at 40 MHz).

3.2 FIP processor

The simulation of the FIP processor electronics showed that:

1. The *Identification Processor* and the *output FIFO* can be built with 3 ALTERAs *EPF10K50* ($\sim 50\times 10^3$ gates, 310 user I/Os). Its processing time is of the order of 5 clock cycles.

2. The part in the FIP controller reading each FIP candidate require 1 ALTERA *EPF10KATC100* (~10×10^3 gates, 134 user I/Os). One FIP candidate can be transferred every 25 ns.
3. The FIP processor can be implemented in a VME64 crate. In that frame work, the user pins of the back plane will be used in the following way: 28 to synchronize input data, 44 for the dialogue between the FIP controller and the FIP boards, ~300 will be assigned to sectors exchange between FIP boards. The latter is strongly related to the mapping between sections and logical sectors. The total number of 372 pins is well below 479, giving us a reasonable safety factor.

For one half of the muon system, the FIP processor is composed of about 7 FIP boards and 1 FIP controller. The exact number of the FIP boards depends on the layout of *sections* which is not well defined yet. The FIP processor can be housed in one half of a VME64 crate.

3.3 DMP processor

The studies of the DMP processor electronics showed that:

1. The identification of the muon using the pads encapsulated in the selected sectors of stations M_2, M_3, M_4 and M_5 can be performed in 2 clock cycles using FPGA technology.
2. The extrapolation of the muon track position to station M_1 can be done by using a DSP or a dedicated FPGA. With a DSP SHARC/TIGER running at 250 MHz, the processing time is ~100 ns (available end 1999).
3. Finding the nearest pad hit in station M_1 in parallel for the 4 selected sectors, requires FPGA technology and 5 clock cycles.
4. The P_T computation requires a look-up table of 128 MB. The $|x|$ coordinate of the pad is encoded on 9 bits for station M_1 and for M_2 while the $|y|$ coordinate is encoded on 8 bits for station M_1. The P_T is encoded on 8 bits. Such look-up table can be built using 128 MB DRAM. The refresh procedure can be performed after each extraction and in any case with a frequency compatible with the component requirements. Such device is a basic component of any PC computer now.

5. The DMP processor can be implemented in a VME64 crate. ~300 user pins on a back plane allow data exchange between the DMP controller and the DMP boards.

For one half of the muon system, the DMP processor is composed of 6 DMP boards and 1 DMP controller. They can be installed in the remaining space available in the FIP crate.

In the configuration, where the muon detector is split in 2 parts, each half being connected to 1 L0(μ) processor, the L0(μ) hardware can be installed in 2 VME64 crates.

The VME bus is used for setting up and monitoring the FIP and DMP processors.

4 Performance of the L0(μ) processor

In this section, we describe the performance of the L0(μ) processor. We summarise the hardware limitations of the proposed architecture and we evaluate inefficiencies for good events. In the second part, we review the latency of the L0(μ) processor.

4.1 Hardware inefficiencies on b$\rightarrow\mu$X events

The major hardware limitations of the proposed implementation are the maximum number of FIP candidates per processor and the time out in the DMP. The minor ones are the maximum number of FIP candidates per FIP board and the maximum number of muon tracks per M_3 sector.

In this study we neglected minor limitations since they are strongly related to the organization of the section in terms of sectors which is currently not well defined.

The acceptance and hardware inefficiencies on B$\rightarrow\mu$X events were estimated in the following way:

1. The FIP and DMP algorithms are run on minimum bias and on B$\rightarrow\mu$X data sets without hardware limitations. The minimum bias retention

plot as a function of the B→μX acceptance is calculated for bunch crossings with only one interaction. Then, the B→μX acceptance, $\varepsilon^{\mathrm{T}}_{2\%}(\mathrm{B} \rightarrow \mu\mathrm{X})$, is determined for the minimum bias retention level of 2%.

2. From the distribution of the number of FIP candidates for B→μX events with one interaction, we compute the probability, $P^{Max}_{FIP}(\mathrm{B} \rightarrow \mu\mathrm{X})$, to reject good events when the cut on the maximum number of FIP candidates is applied.
3. The effect of the DMP time out is estimated by using a simplified Monte-Carlo which can generate the number of FIP and the corresponding number of muon tracks as a function of the luminosity for each bunch crossing. It allows to determine the probability, $P^{Out}_{DMP}(\mathrm{MB})$, to loose a minimum bias event for bunch crossing with one interaction.
4. The hardware inefficiency is given for bunch crossings with one interaction by:

$$\varepsilon^{\mathrm{H}}(\mathrm{B} \rightarrow \mu\mathrm{X}) = P^{Max}_{FIP}(\mathrm{B} \rightarrow \mu\mathrm{X}) + P^{Out}_{DMP}(\mathrm{MB}) . \qquad (1)$$

5. The final acceptance on B→μX events when the minimum bias retention is fixed to 2% is defined for bunch crossing with 1 interaction by:

$$\varepsilon^{\mathrm{F}}_{2\%}(\mathrm{B} \rightarrow \mu\mathrm{X}) = \varepsilon^{\mathrm{T}}_{2\%}(\mathrm{B} \rightarrow \mu\mathrm{X}) \times \left(1 - \varepsilon^{\mathrm{H}}(\mathrm{B} \rightarrow \mu\mathrm{X})\right) \qquad (2)$$

Each data set contains 9×10^3 events. These are standard events produced with SICB version v112 for B→μX events and v113 for minimum bias data. The physics background due to showering in the muon shield and neutron capture is taken into account. It is described by a model based on GCALOR studies [7]. It produces the pad hit distribution in Muon station as a function of the distance to the beam. This is a crude model built with a simplified detector description. In addition, the detector geometry, the beam pipe and the muon shielding was slightly different from the current one (v112–v115).

In that study, the analysis is repeated by scaling the physics background. We also take into account a chamber noise at 50 $\mathrm{Hz/cm^2/plane}$ but we ignore the muons generated by the LHC machine [8].

The results are given for a configuration where stations are split into 2 independent parts. Each half is connected to a separate L0(μ) processor.

The maximum number of FIP candidates per processor and the DMP decision time delay are considered as free parameters. They have been optimized to minimize the hardware inefficiencies. The results are summarized in Table 2 and the corresponding hardware inefficiencies in Table 3.

Table 2 The maximum number of FIP candidates per processor and the DMP decision time delay as a function of the luminosity and as a function of the scale factor applied to the physics background. The values quote in the Table minimize the hardware inefficiencies (Eq. 1). They were obtained for a configuration where stations are split into 2 independent parts. Each half is connected to a separate L0(μ) processor.

Scale factor (physics background)	2×10^{32} **cm^{-2}s^{-1}**		5×10^{32} **cm^{-2}s^{-1}**	
	FIP	DMP	FIP	DMP
x1	6	200 ns	6	200 ns
x2	6	200 ns	6	500 ns
x3	8	500 ns	4	500 ns

Table 3 Hardware inefficiencies (Eq. 1) on B→μX events for a configuration where the stations are split in 2 independent parts. Each half is connected to a separate L0(μ) processor. The optimum value of the maximum number of FIP per processor and the DMP decision time delay are given in Table 2.

Scale factor (physics background)	2×10^{32} **cm^{-2}s^{-1}**	5×10^{32} **cm^{-2}s^{-1}**
x1	–	–
x2	–	~1%
x3	~1.5%	~24%

In standard conditions, the effect of the hardware limitations is negligible. They start to play a role at the maximum luminosity when the current estimation of the physics background is scaled by a factor 3. In that case, the limitation on the number of FIP candidates to 4, acts as an effective cut to reject background.

In reference [3], the B→μX acceptance, $\varepsilon^{R}_{2\%}(B \to \mu X)$, corresponding to the minimum bias retention of 2% was obtained by running the theoretical algorithm alone (the latter is identical to the DMP algorithm). Taking these results as a reference, we quote in Table 4 the overall inefficiencies as a function of the scale factor applied to the physics background. The overall inefficiencies is calculated using the following equation:

$$I_{2\%}(B \to \mu X) = \frac{\varepsilon^{F}_{2\%}(B \to \mu X) - \varepsilon^{R}_{2\%}(B \to \mu X)}{\varepsilon^{R}_{2\%}(B \to \mu X)} \tag{3}$$

where $\varepsilon^{R}_{2\%}(B \to \mu X)$ is equal to 53.6% (1). Inefficiencies are due to: a) intrinsic inefficiency of the algorithm when the background increases; b) the splitting of the stations in 2 halves; c) the hardware inefficiencies. The dominating factor is the intrinsic quality of the algorithms.

Table 4 Overall inefficiencies (Eq. 3) on B→μX acceptance for bunch crossing with one interactions when the minimum bias retention is fixed to 2%.

Scale factor (physics background)	**FIP + DMP algorithms**		**Theoretical algorithm**
	2×10^{32} **cm^{-2}s^{-1}**	5×10^{32} **cm^{-2}s^{-1}**	
x1	~1%	~1%	~2%
x2	~14%	~15%	~18%
x3	~33%	~49%	~48%

The performance of the proposed implementation is slightly better than the theoretical algorithm alone. It is due an additional cut implemented in the FIP algorithm: a FIP candidate is killed if more than one neighbouring sector is hit when the central sector is not hit in station M_j (see section 2.1).

(1) The reference sample contains B→μX events where the generated muon coming from the B decays crosses at least the muon station M_1 and M_2.

4.2 L0(μ) latency

From the electronics simulation of the basic components of the L0(μ) processor and the previous studies, we can evaluate the L0(μ) processing time. It is summarized in Table 5 assuming:

1. Cable length between FEBI and FIP boards 5 m.
2. Serial links between: a) FEBI boards and FIP boards; b) FEBI boards and DMP controller. The time overhead to serialize the data is 50 ns. In addition, 50 ns are required to deserialize the data.

Table 5 Processing time of the L0(μ) processor

	Step	**Time (ns)**
From FEBI to FIP board		125
	FIP board	125
	FIP Controller	75
FIP Processor	**Total**	**200**
	FIP controller to FEBI controller	125
	FEBI processing	50
	FEBI controller to DMP controller	125
FEBI interrogation	**Total**	**300**
	Muon identification	100
	Pad extrapolation (M_1)	125
	Nearest hit finder (M_1)	100
	P_T computation	50
	BC decision	125
	Decision time delay[a]	200–500
DMP Processor	**Total**	**700–1000**
$\mathbf{t}_\mu$ [9]		**1200–1500**

a. To accommodate the arrival time of the last FIP candidate to the DMP processor and to handle statistical fluctuation in the number of FIP candidates per bunch crossing.

Following the computation performed in reference [9], the L0 latency for the muon system is equal to 1750+t_{μ}+t_0 ns, where

1. The fixed value takes into account the elapse time between the *pp* collision and the availability of the data to the L0(μ) processor (675+150 ns) as well as the processing time for the L0 decision box (275 ns) and the time to broadcast the decision to all the FE boards (650 ns).
2. t_{μ} is the L0(μ) processing time estimated in Table 5.
3. t_0 is a provisional time to elaborate more complex decision in the L0 decision box (225 ns).

Adding all these numbers, we obtain a total L0(μ) latency ranging from 3175 up to 3475 ns without contingency.

5 Synchronization

In such system data have to be synchronized with the BC number which is common to all sub-detectors.

Our guidelines to design the synchronization of the system are the following:

1. Process all the bunch crossings.
2. Maintain bunch crossing order at all stages.
3. Produce the L0(μ) decision word at a fixed latency.
4. Minimize tuning operations to ensure correspondence between a BC identifier and the data.
5. Provide a fast detection of synchronization errors and a local error recovery whenever it is possible.
6. Introduce redundancy for debugging and monitoring
7. Provide a special procedure for safe (re)synchronization in case of major problems.

We assume:

1. The 256 binary pad information is aligned in time at the input of the L0(μ) interface in the FEBI board. Their BC identifier is known. It is attached to the data.
2. The arrival times of the pad information are different between FEBI boards. However, we expect a small difference, well below 25 ns, to minimize the L0(μ) latency.
3. After a general reset of the electronics, the first pad information arriving in the FEBI boards belongs to the same bunch crossing.
4. Data can be validated (control bits, strobe,...).
5. TTC flag control signals are available at each stage.

In our framework, the BC identifier is associated once with the data in the FEBI electronics. The BC identifier is known in the L0(μ) interface. Later on, sufficient information is carried along with the data in order to perform a quick detection of the synchronization errors and to guarantee the integrity of the data through the L0(μ) processing. It is performed in the following ways:

1. In the transfer of the sector hit maps between the FEBI's and the FIP boards, 2 *synchronization flags* are carried with each sector hit map: the Least Significant Bit of the BC identifier and a flag telling us that the BC identifier is equal to a predefined value *n*. Synchronization flags follow the data along the FIP board processing.
2. In the FIP controller, the BC identifier is regenerated from the synchronization flags. Then the BC identifier is attached to each FIP candidate: the BC identifier and the sector addresses are sent to the FEBI controllers; BC identifier, sector addresses and pad hit maps are returned to the DMP controller. In addition, the FIP controller sends the number of FIP candidates and the corresponding BC identifier to the DMP controller, to synchronize the FIP and the DMP processor, for each bunch crossing.
3. At the end of the DMP processing, the BC identifier is used to collect muon candidates belonging to a given bunch crossing and to build the decision. The BC identifier is also a part of the word sent to the L0 decision unit.

Sector hit maps are not aligned in time at the input of the FIP processor. To synchronize them and to ensure that they belong to the same bunch crossing, we introduce the following tools:

1. 70 flags *"FIFO empty"* attached to each input FIFO of the FIP boards. They are in the *"empty"* state when there are no data in the FIFO or after a general reset / clear of the electronics.
2. A unique flag *"FIFO reading"* seen by all the FIP boards. In the *disable* state data are kept in the input FIFO while in the *enable* state data are extracted using the master clock of the processor. After a general reset, this flag is in the *disable* state.
3. The input of the FIFO is setup in such a way that only *valid* data are written into it.

The synchronization procedure works in the following way with these tools when assumption 3 is satisfied. After a general reset of the electronics, the first *valid* data appearing at the entrance of the FIP processor are latched in the FIFOs. Thus, all the data coming from different sections and stations belonging to the same bunch crossing are stored in the first position of the FIFOs at different time. To synchronize all the FIP data, we wait for the flags *"FIFO empty"* to switch to the state *"not empty"* for all the input FIFOs of the FIP processor. When this condition is satisfied, the flag *"FIFO reading"* flip into the "*enable*" state. The reading of the FIFOs starts. It is performed simultaneously for all FIFOs using the master clock available everywhere in the FIP processor. Thus, the data are aligned in time at the output of the FIFO, for the FIP processor. The same technique is implemented at the input of the DMP processor. After a general reset, the *FIFO are empty* and the *"FIFO reading"* is disabled. The latter flag can also flip if an error is detected or if the condition *"FIFO not empty"* is suddenly unsatisfied.

If after a reset the pad information arriving in the FEBI boards belongs to different bunch crossings, a safe procedure is required to synchronize them. It can be performed in the following way. After a reset, FEBI boards send their first *valid* data when the BC identifier is equal to a predefined value *k*. Thus, the data in the first position in the FIFOs of the FIP boards belong to the same bunch crossing *k*. In that case, the BC identifier is known, but up to *k* bunch crossings are lost.

A synchronization failure can be detected at the output of the FIFOs, every 25 ns, since the synchronization flags are part of the data. The synchronization flags have to be identical for all the data at the output of FIFOs of the whole FIP processor. If the *Synchronization Supervisor* detects an error in a FIP board, the error message is carried up to the L0 decision box. To recover the error, an other message is sent to the FEBI controllers. It tells the FEBI boards to stop sending sector hit maps. Then, the FIFOs in the FIP boards are cleared. FEBI boards will restart sending data when the BC identifier is equal to a predetermined value. Up to 128 bunch crossings are lost during the recovery procedure.

In the FIP controller the BC identifier has to be matched with the synchronization flags. This operation requires the TTC control signals which allows to build a BC number for each bunch crossing. Synchronization flags allow tuning of the internal delay of the TTC system in order to obtain a perfect agreement between synchronization flags and the BC identifier. This procedure is only performed once. In running conditions, the comparison between the synchronization flags and the BC identifier allows to detect errors.

6 Debugging and monitoring

Our main guidelines to design a setup to debug and to monitor the L0(μ) processor will be the following:

1. Check the functionality of each major components at any time.
2. Detect a malfunction as soon as possible and provide enough information to localize and to understand the problem.
3. Perform a local error recovery whenever it is possible.

We intend to introduce at the input of each major component (e.g. FIP board) some FIFOs/RAMs. They are filled as part of the standard data flow in the data taking mode and accessible by an external computer to inject test patterns in the debugging mode. Intermediate results of a major component will be stored in output RAMs. Input and output RAMs can be read by an external computer via the VME bus. By comparing the intermediate results with a simulation of the component, monitoring and debugging can be performed on a subset of the events.

7 Cost estimate

The cost of the proposed architecture is detailed in Table 6 and Table 7. The prices of the FPGA's are based on a new family: ALTERA *APEX 20K*. The cost does not take into account components belonging to the FE electronics: a) optical links between the off-detector and the FEBI boards; b) 9U PCB for FEBI boards; c) VME64 crate housing the FEBI boards.

Table 6 Cost of the additional electronics implemented in the FE electronics (including spare boards).

FEBI	**Number of unit**	**Cost per unit (kCHF)**	**Total (kCHF)**
L0(μ) interface (FEBI board)	196+24	1.44	317
FEBI controller	14+3	4.66	79
Custom bus	14	3.08	43
Total			**439**

Table 7 Cost of the L0(μ) processor (including spare boards).

L0(μ)	**Number of unit**	**Cost per unit (kCHF)**	**Total (kCHF)**
FIP board	14+3	7.86	134
FIP Controller	2+2	10.09	40
DMP board	12+3	6.86	102
DMP controller	2+2	10.60	42
Custom bus	2+1	7.96	24
VME64 crate	2+1	20.0	60
Optical links (1 Gb/s)	212	0.375[a]	80
Monitoring			100
Total			**582**

a. Our estimate is based on the 1999 price for a running setup carrying 20 data bits at 40 MHz. The quoted price is a detailed projection taking into account the large number of optical links implemented in the muon system.

The total cost of the proposed architecture amount to 1 MCHF taking into account spares.

8 Conclusions and prospects

The proposed solution for the L0(μ) processor is very compact since it can be implemented in just 2 VME64xP crates. It also minimizes the number of links between the FE electronics and the L0 hardware. Still, the performance of the proposed architecture is perfectly satisfactory.

With version v112–113 of the simulation of the experiment, we can conclude:

1. The hardware limitations due to the maximum number of FIP candidates per bunch crossing and the DMP time out introduce negligible inefficiency on $B \rightarrow \mu X$ events.
2. The proposed architecture is robust against the physics background. Hardware inefficiencies are negligible when the physics background is scaled by a factor 2. The processor works at the maximum luminosity with degraded performance when the physics background is scaled by a factor 3.

Our implementation can be built by using current technologies like FPGA’s, optical links and DSP’s. It can also be housed in a VME64xP crate which will be the standard for the LHC experiments.

To proceed towards the technical design, the following goals must be achieved.

1. Improve the simulation of the processor by:
   - Moving to a more realistic simulation of the muon system and of its physics background in order to estimate properly the performance of the L0(μ) processor. In such studies, we have to take into account: the section layout, the maximum number of candidates per FIP board, the reduced precision due to a limited data encoding.
   - Performing full electronics simulation of the L0(μ) boards to validate the synchronization procedure and to ensure the flexibility of the system.

2. Follow the FE electronics design to guarantee that L0, L1 buffers, their derandomizers and the L0(μ) interface can be implemented in one 9U VME board.
3. Validate the technology for basic components.
   The proposed architecture relies on our capability to: a) connect a large number of high speed links to a single 9U board; b) design a custom back plane running at 40 MHz within the VME64xP standard. The feasibility of these basic elements have to be demonstrated. We intend to build a small prototype to:
   - Study a solution to embed our custom high density buses in the VME64xP back plane and test it.
   - Connect 10/14 optical links running at 1.6 Gb/s to one 9U board to study effects of high frequency cross-talks and the error rates. In addition test the mechanism to validate data.
   - Manipulate FPGA with a high number of I/O pins and develop a method to debug them.
4. Run a prototype of a FIP board to merge all the involved technologies and to learn about high speed links and high density FPGA's in a real environment.

Finally, our architecture will have to evolve in order to cope with the optimization of the muon station layout. The latter might be a mixture of pads and strips where the projectivity is abandoned.

References

[1] E. Aslanides et al., An alternative architecture of the L0(μ) processor, LHCb Note, LHCb 97–24

[2] LHCb Collaboration, Technical Proposal, CERN / LHCC 98–4

[3] F. Harris et al., LHCb data flow requirements, LHCb Note, LHCb 98–27

[4] VME64 extensions for physics and other applications (VME64xP), VITA 23–199x, Draft 1.3, 10 December 1997

[5] National Semiconductor, DS90CR285

[6] Hewlett Packard HDMP1022 and HDMP1024

[7] N. Saguidova et al., GCALOR studies of background in the LHCb muon chambers, LHCb Note, LHCb 98–59

[8] I. Azhgirey and V. Talanov, LHC generated muons background on LHCb formulation of the problem, LHCb Note, LHCb 97–13

[9] I. Videau, Comments on the latency of the level 0 trigger. LHCb Note, LHCb 99-xxx