

A Demonstrator for the ATLAS Level-1 Muon to Central Trigger Processor Interface (MUCTPI)

A. Corre, N. Ellis, P. Farthouat, Y. Hasegawa, G. Schuler, C. Schwick, R. Spiwoks
 CERN, 1211 Geneva 23, Switzerland

Abstract

The Level-1 Muon Trigger Interface (MUCTPI) to the Central Trigger Processor (CTP) receives trigger information from the detector-specific logic of the muon trigger. This information contains up to two muon-track candidates per sector. The MUCTPI combines the information of all sectors and calculates total multiplicity values for each of six programmable p_T thresholds. It avoids double counting of single muons by taking into account the fact that some of the trigger sectors overlap. The MUCTPI sends the multiplicity values to the CTP which takes the final Level-1 decision. For every Level-1 Accept (L1A) the MUCTPI sends region-of-interest (RoI) information to the Level-2 trigger and event data to the data acquisition system. A demonstrator of the MUCTPI has been built which has the performance of the final system but has limited flexibility for calculating the overlap. The functionality and the performance of the demonstrator are presented.

I. MUON TRIGGER SYSTEM

The ATLAS Level-1 trigger [1] is based on multiplicity information from clusters found in the calorimeters and from tracks found in dedicated muon trigger detectors. The muon trigger detector uses resistive plate chambers (RPCs) in the barrel region and thin-gap chambers (TGCs) in the end-cap and forward region. Coincidences of hits in different layers are used to identify muon-track candidates. The width of the roads used to define the coincidences determines the trigger transverse-momentum threshold of the candidates. Candidates are counted for six different programmable p_T thresholds. An overview of the ATLAS muon trigger is shown in figure 1.

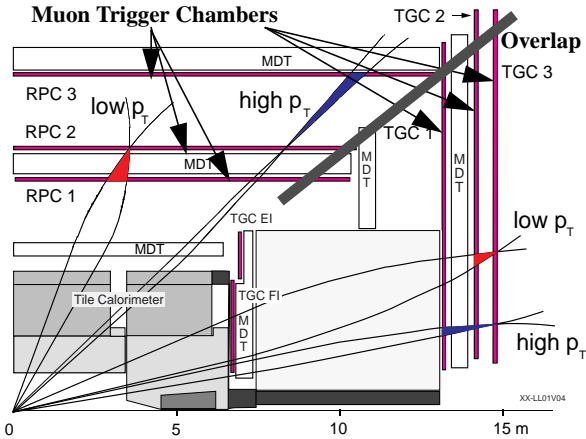


Figure 1: ATLAS Level-1 Muon Trigger

The trigger detectors are segmented into sectors. Each sector can identify up to two muon-track candidates in a total of six programmable p_T thresholds. Detector-specific sector logic sends details of the two highest- p_T candidates in each sector to the MUCTPI [2] which evaluates detector-wide multiplicities. Special attention is paid to muons that traverse overlapping trigger chambers [3]. In particular, within the barrel, and between the barrel and the end-cap regions, muon-track candidates could be counted twice. This would lead to an unacceptable rate of fake di-muon triggers. Overlap within sectors and between endcap and forward sectors is handled by the detector-specific logic. There is no overlap between different octants in ϕ . The overlapping of sectors for one octant in ϕ is shown in figure 2.

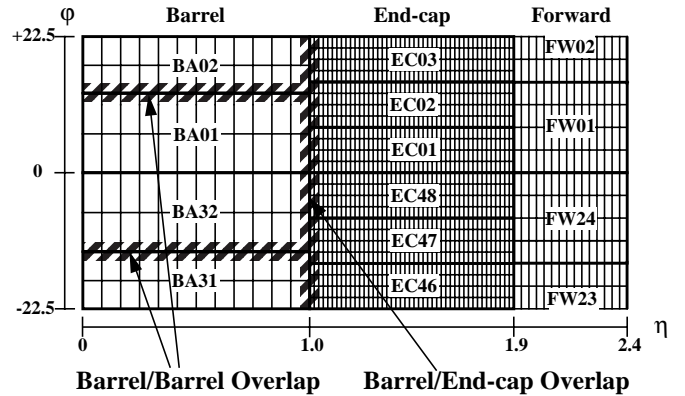


Figure 2: Overlap of Muon Trigger Sectors in one Octant

The MUCTPI sends the total multiplicity values for the six programmable p_T thresholds to the CTP which takes the Level-1 decision. An overview of the muon trigger system is shown in figure 3.

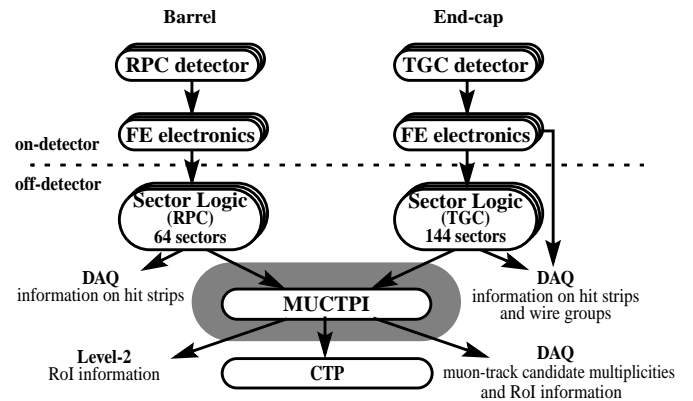


Figure 3: Muon Trigger System Overview

II. MUON TO CENTRAL TRIGGER PROCESSOR INTERFACE (MUCTPI)

The MUCTPI receives, for each bunch crossing (BC) of the LHC, trigger information from the muon trigger detectors and calculates multiplicities of muon-track candidates taking into account the overlap of detector sectors. It sends the total multiplicity information for each of six p_T thresholds to the CTP. On reception of an L1A it sends RoI information to the Level-2 trigger and event data to the data acquisition system.

The MUCTPI must be capable of receiving data at a rate of 33 GByte/s, corresponding to a total of 208 sectors sending 32-bit words at 40 MHz. The latency of the multiplicity summation must not exceed 200 ns (8 BCs). The multiplicity value for each threshold is limited to 3 bits, i.e. a maximum value of 7. The MUCTPI must be capable of accepting L1As up to a maximum rate of 100 kHz and providing information to the Level-2 trigger and the data acquisition without any loss. Sufficient on-line monitoring must be provided to verify the correct functioning of the MUCTPI.

The MUCTPI is built around several modules. 16 MIOCT modules each receive the data for one octant in ϕ and one half in η of the muon trigger detector. The MIBAK backplane sums the multiplicities of all MIOCTs, provides data transfer and broadcasts fast synchronization signals for all modules. The MICTP module sends the multiplicities to the CTP and receives the fast signals from the CTP. The MIROD module collects information from the MICTP and the MIOCTs and sends data to the Level-2 trigger and the data acquisition system. An overview of the MUCTPI is shown in figure 4.

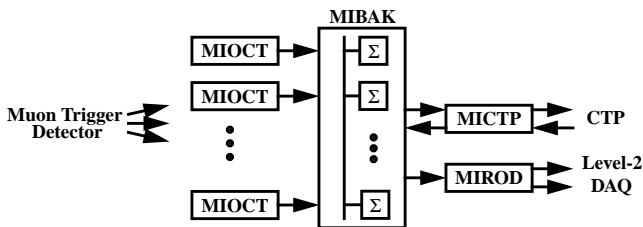


Figure 4: Overview of MUCTPI

A. MIOCT Module

The MIOCT module receives data words from each of 14¹ sectors at 40 MHz. The 32-bit data words contain details of up to two muon-track candidates. The data are synchronized to the rising or falling edge of the BC clock and can be aligned among the different sectors using programmable delays. The MIOCT module sums the multiplicities for six p_T thresholds taking into account the overlap of sectors. The results are sent to the MIBAK backplane. The MIOCT module also stores the sector data in a pipeline. When an L1A is received the corresponding data are formatted into an event fragment and written into the read-out FIFO. Up to two BCs before and after the triggering BC can be included. Empty sector data can be suppressed. The event fragment can also be written into a

1. The two outer forward sectors of each octant provide data to two MIOCT modules.

monitoring FIFO which can be read out via VME.

B. MIBAK Backplane

The MIBAK backplane consists of three parts. The first part is an active backplane for the summation of the multiplicities from all 16 MIOCT modules. The second part performs the data transfer from the MICTP and the 16 MIOCT modules to the MIROD module. The third part broadcasts fast signals between the MICTP and all other modules. Those fast signals include the BC clock, the bunch counter reset (BCR), the L1A, the event counter reset (ECR), and the test and monitoring signals (see section IV.B.). The third part also contains a wired-OR BUSY line from all modules.

C. MICTP Module

The MICTP module receives the total multiplicities for the six p_T thresholds and sends them to the CTP. It also writes the multiplicities into a pipeline for read-out by the MIROD module on reception of an L1A. The MICTP module further receives the fast signals from the CTP and makes them available on the MIBAK backplane. It also receives the wired-OR BUSY signal from all modules. The modules use this signal to indicate that buffers for the data collection associated to L1As are filling up. The MICTP module sends the BUSY signal to the CTP where it can be used to throttle the L1A generation.

D. MIROD Module

When all MICTP and MIOCT modules have data associated to an L1A available they use a wired-AND READY signal to indicate to the MIROD module that it can start the data collection. The MIROD module collects the event fragments over the MIBAK backplane using a token protocol. General event information and all muon-track candidates are extracted. Thresholds can be applied to the p_T values of either of the two candidates of each sector and sector numbers can be mapped to geometrical identifiers. The extracted event and candidate information is pushed into three different data processing branches. In the first branch the candidates are sorted in descending order in p_T , limited in number, formatted and sent as RoI information to the Level-2 trigger. The second branch takes all the candidates, formats them and sends them to the read-out buffers of the data acquisition system. The third branch selects events and writes them into a monitoring FIFO; the selection can be done for all events, for every n-th event, for an event with a given BC identification or a given event number, for an event with the monitoring flag set (see section IV.B.), or a combination of these criteria. The monitoring FIFO can be read out through the VME bus.

III. DEMONSTRATOR PROTOTYPE IMPLEMENTATION

A demonstrator prototype of the MUCTPI has been implemented. Two MIOCT modules and one MIROD module have been built as 9U \times 400 mm VME modules. Several emulator cards for the MICTP and the MIOCT modules have been built as 9U \times 60 mm cards. The MIBAK backplane has been built

and mounted in the position of the J3 backplane in a 9U VME crate.

A. MIOCT Module

The MIOCT modules have been built using programmable logic devices, mainly from Altera Corp [4]. For a full description of the MIOCT module see reference [5]. The reception of data from the detector-specific logic is carried out using LVDS receivers. The relative phase of bit(0) of each sector data word can be measured using the CERN/EP/MIC TDC32 [6] which is controlled by an embedded JTAG controller accessible through the VME backplane. Test memories implemented in embedded RAM of the FPGAs responsible for receiving the data can be used to provide test data instead of the external data. The synchronization and timing alignment of data, the calculation of overlap, and the summation of multiplicities are carried out with several FPGAs and CPLDs, as well as SRAMs used as look-up tables. The programming of the CPLDs is achieved through a JTAG and Altera Bitblaster port, while the FPGAs are configured using a Flash memory. A CPLD-based VME controller gives access to the on-board resources using VME A32 D16/D32 data transfers. Figure 5 shows a photograph of the MIOCT module.



Figure 5: The MIOCT Module

Additional MIOCT emulator cards have been developed in order to provide correct electrical loading of the MIBAK backplane. These cards only receive their power from the VME backplane, but are fully connected to the MIBAK backplane. The emulator cards are passive: the multiplicities are always zero; the token of the read-out protocol is immediately passed on.

B. MIBAK Backplane

The multiplicity summation on the MIBAK backplane has been implemented using Altera CPLDs. The data transfer from MICTP and MIOCTs to MIROD is implemented using Bus LVDS (BLVDS). Signal integrity studies have been carried out for this part in order to validate the design before construction².

2. Cadence SigXplorer has been used with the help from CERN/IT/CAE/AE group.

The fast signals are transferred between the MICTP module and all other modules using a low-voltage positive ECL. Hard-metric connectors of 2-mm type with 5 columns and shielding are used to connect all modules to the MIBAK backplane.

C. MIROD Module

The MIROD module has been implemented using programmable logic devices from Altera Corp. [4]. For a full description of the MIROD module see reference [7]. The MIROD module uses FPGAs for the implementation of the token passing protocol, the event extraction and distribution over the three subsequent data processing branches. Play-back memories implemented in SRAM for the MIBAK data and in embedded RAM for the MIBAK control can be used to effectively emulate the functioning of the MIBAK backplane. S-Link [8] is used for the transfer of data to the Level-2 trigger and to the data acquisition. The event formatter for the Level-2 trigger and the data acquisition uses the ATLAS read-out driver (ROD) event format [9]. The design file is written using VHDL and can be used by other sub-detectors [10]. The MIROD module can be used in MIBAK analyser mode in which information on the MIBAK signals is stored in the candidate FIFO. The S-Link controller contains an optional analyser FIFO which allows the storage of data and control signals in real-time. These FIFOs can be read out through VME. The content can be analysed by a script which provides waveforms. The monitoring FIFO allows access to selected events. An FPGA-based VME controller implements VME A32 D16/D32 access. The MIROD module can generate a VME interrupt when the monitoring FIFO reaches a programmable watermark. A Flash memory contains the configuration files of the FPGAs.

D. MICTP Module

An emulator card for the MICTP has been developed which allows injection of the fast signals onto the MIBAK backplane. It also extracts the multiplicity values and the BUSY signal. The emulator card is otherwise passive. The fully functional MICTP module is currently under design.

IV. TESTS AND RESULTS

A. Test Programs

Test software has been developed for the RIO II 8061 VME processor from Creative Electronics Systems SA [11]. The processor runs LynxOS 3.0.1. The software is written in C++ and uses the GNU gcc compiler v2.7. It contains a class library for the VME master mapping which allows single-word read and write access in A32 D16 and D32 mode. Registers of the modules can be accessed using a register description file which maps register names to their corresponding VME addresses. For the MIOCT modules, the software can, in particular, generate and down-load test data into the test memories, configure the module and read out the monitoring FIFO. For the MIROD module, the software allows generation and down-loading of MIBAK data into the play-back memories, configuration of the module, and read-out of the monitoring FIFO and the analyser FIFOs for the MIBAK and the S-Link analysers.

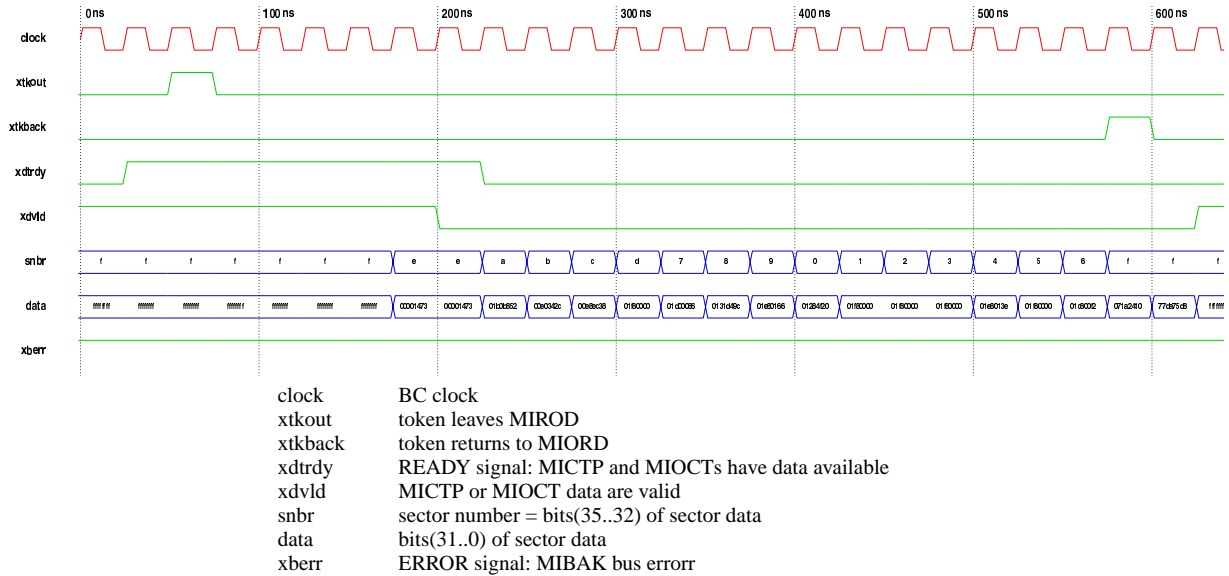


Figure 6: The Token Protocol for Read-out Observed with the MIBAK Analyser of the MIORD Module

B. Results

The MIOCT modules have been tested thoroughly. The dispersion of the clock and of the other fast signals on the MIOCT module have been measured to be within ± 1 ns. The phase measurement of bit(0) of the sector data works correctly. Using the same input signal on each sector, differences of the phase of ± 1 ns have been measured. The synchronization of the sector data with the internal clock, and the relative timing alignment of the sectors work correctly for latency differences up to 400 ns (10 BCs). Data in the test memories together with a test signal from the MICTP were used to verify the correct functioning of the overlap handling logic which is found to be in accordance to reference [3]. The multiplicities are summed correctly with the maximum value for each p_T threshold limited to 7. The read-out of the monitoring FIFO has been tested. The data can be correctly aligned with the internal BC identification counter and the multiplicities. The mechanism for windows around the triggering BC and the zero suppression work correctly. The transfer of data to the MIORD using the token passing protocol works successfully.

The MIBAK backplane has been tested extensively. The fast signal propagation has been tested and the timing to all modules has been verified. The differences in timing are within ± 1 ns. The data transfer lines have been checked with a time-domain reflectometer and results were found to be in accordance with the simulations carried out for the signal integrity. The multiplicity summation has been tested with one MIOCT module and the emulator cards providing multiplicity values of zero. The summation works correctly and the total multiplicity values arrive at the MICTP emulator card with a relative alignment of about 6 ns. The final MICTP module will latch the multiplicities before it sends them, fully aligned, to the CTP. The total latency of the MUCTPI from sector data at the input of the MIOCTs to multiplicities at the output to the CTP can be estimated to stay clearly within the specified 200 ns.

The MIORD module has also been tested thoroughly. The token protocol for data collection from the MIBAK backplane works correctly. This has been tested with the play-back memories and real data transfers using one MIOCT module. Figure 6 shows an example of such a transfer using the MIBAK analyser of the MIORD module. In the example, the MIORD sends out the token after the MIOCT module has activated the READY signal. The MIOCT module sends its data before it returns the token to the MIORD. The event extraction of the MIORD module works correctly. Thresholds can be applied to the p_T values of either of the two candidates of each sector. Event header and the muon-track candidates are provided in the corresponding FIFOs. Extracted events can be sent correctly to all three data processing branches. The monitoring works for all five criteria or combinations thereof. Candidates for the Level-2 trigger are correctly sorted according to the p_T threshold. The total number of candidates sent to the Level-2 trigger can be limited. The event formatting for the Level-2 trigger and the data acquisition works correctly. The functioning of the S-Link read-out towards the Level-2 trigger and towards the data acquisition have been tested successfully using the ROD formatter analyser as well as an S-Link infinite data drain (SLIDAD) [8]. The ROD formatter analyser uses the same technique as was used to obtain the waveform in figure 6. The behaviour of the S-Link in case of the link being busy from time to time has also been tested successfully on a statistical basis.

The MUCTPI demonstrator prototype contains further a mechanism for synchronized monitoring of event fragments in MICTP and the MIOCT modules. A signal injected in the MICTP (emulator) is put into the data as a flag at the level of the MICTP and MIOCT modules. When the data from the MICTP and the MIOCTs are collected by the MIORD they can be selected for monitoring based on this monitoring flag. Since the monitoring signal arrives at all modules at the same time,

the monitored event fragments are synchronized in time with respect to each other.

The MICTP (emulator) receives a BUSY signal which is the wired OR of the individual BUSY signals of all modules. The working of this signal has been verified for one MIOCT module with all the emulator cards signalling a non-BUSY condition.

Errors in the timing alignment of the data in the MIOCT module with respect to the BC identification are detected. In case the BC identification of the data and the internal BC counter do not match, a multiplicity of zero is sent to the MICTP backplane. The following errors in the data transfer between MICTP, MIOCT and MIROD can be detected: if the MIOCT module receives the token and the data are not ready or the data are not correctly framed between a header and a trailer, the ERROR signal of the transfer bus on the MIBAK backplane is activated. The MIROD module monitors the ERROR signal and also checks the correctness of the token protocol itself. In case of an error the event will be flagged as erroneous but will be processed normally in order not to disturb the data flow.

V. CONCLUSION

A demonstrator prototype for the MUCTPI has been implemented and tested. It works according to the specification and has already almost the full functionality required for the final system. It could be used as is in the experiment at the expense of relatively little flexibility in the treatment of the overlap. For the final system the programmability of the overlap treatment will be enhanced.

The MICTP module will be built and full system tests carrying data from the input of the MIOCT modules to sending of the total multiplicity values to the CTP will be carried out. In the near future, the MUCTPI will be integrated with (prototypes of) the detector sector logic, the CTP, the Level-2 trigger (RoI builder module), and a read-out buffer of the data acquisition system. It will also be integrated with the run control system, the configuration database and the monitoring system of the data acquisition.

VI. REFERENCES

- [1] ATLAS Collaboration, First-level Trigger Technical Design Report, CERN/LHC/98-14, June 1998, in particular, chapter 15.
- [2] C. Schwick et al., The ATLAS Muon Trigger Interface (MUCTPI), Proc. Fourth Workshop on Electronics for LHC Experiments 1998, Rome, CERN/LHCC/98-36, p. 275 - 279.
- [3] P. Farthouat, Interfaces and Overlaps in the Level-1 Muon Trigger System, ATLAS EDMS note ATL-DA-EN-0001.
- [4] MIOCT and MIROD modules use mainly Altera Max7000S and Flex10K devices; see <http://www.altera.com>.
- [5] G. Schuler, R. Spiwojs, The Octant Module of the Muon-CTP-Interface (MIOCT), ATLAS EDMS note ATL-DA-

ER-0004.

- [6] <http://pcvlsi5.cern.ch:80/MicDig/tdc32.htm>.
- [7] C. Schwick, The MIROD (Muon-CTP-Interface Read-out Driver) User's Guide, <http://home.cern.ch/schwick/muctpi/mirod/manual/ug.html> or [ug.pdf](#).
- [8] <http://hsi.web.cern.ch/HSI/s-link/>.
- [9] C. Bee et al., The Event Format in the ATLAS DAQ/EF Prototype -1, ATLAS-DAQ-98-129 (unpublished).
- [10] <http://home.cern.ch/schwick/vhdl/AtlasRODFormatter/index.html>.
- [11] <http://www.ces.com>.