ATLAS Internal Note
DAQ-NO-085
26 March 1998

ATLAS Level-2 Trigger Demonstrator-A
Activity Report
Part 1: Overview and Summary

26 March 1998

A. Kugel, K. Kornmesser, R. Lay, R. Männer, K-H. Noffz, S. Rühl, M. Sessler, H. Simmler, H. Singpiel
University of Mannheim, Germany

V. Dörsing, W. Erhard, P. Kammel, A. Reinsch
University of Jena, Germany

L. Levinson
Weizmann Institute of Science, Rehovot, Israel

R. Bock
CERN, Geneva, Switzerland

W. Iwanski, K. Korcyl, J.Olszowska
Institute of Nuclear Physics, Cracow, Poland

D. Calvet, J.R. Hubbard, P. Le Dû, I. Mandzavidze, M. Smizanska
Centre d'Etudes Nucleaires de Saclay, Gif-sur-Yvette, France

Abstract

The demo-A activity report describes the work done on the ATLAS Level-2 trigger during the recent phase of the demonstrator program by the groups associated with architecture-A: Cracow, Jena, Mannheim and Weizmann. Important contributions by CERN (Rudy Bock et. al.) and from the Saclay architecture-C group are included.

The activity report is presented in a set of three separate documents:

- Part 1: Overview and summary
- Part 2: Measurements and results
- Part 3: Paper model

This first part gives an overview of the trigger system and the target architecture-A. The evolution of this FPGA based trigger approach towards a hybrid system is illustrated. The key element of architecture-A - the FPGA-processor - is presented with it’s most important features. A description of all components used in the demonstrator-A program together with a result summary is given. Next the focus of the activities concerning FPGA based processing during the pilot project is explained. Finally some conclusions from the demonstrator-A project are drawn.

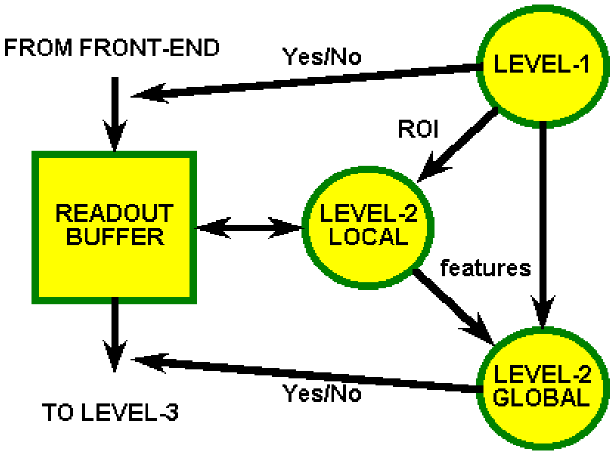
The two other documents provide more detailed information on the corresponding specific items.

Contents

1. Introduction

The ATLAS Trigger System

The ATLAS Level-2 Trigger works with the fully granular data provided by the various subdetectors of the ATLAS experiment. Local feature extraction (FEX) per subdetector and a global processing step are combined to reach the final Level-2 decision.
In high-luminosity operations the Region-of-Interest (RoI) principle is used to reduce the data rates. The event rate of 100kHz can presently not be handled by affordable standard computer systems. For the low-luminosity trigger, the critical item is the processing of the complete data volume of the Transition-Radiation-Tracker (TRT). This leads to an enormous demand in terms of computing power and bandwidth.



Some key figures of the L2-Trigger are:

- 100 kHz input event rate (L1 accept)
- 1 kHz output event rate (L2 accept)
- > 100GB/s data rate into buffers
- 2 -3 GB/s data rate from buffers to L2 processors
- ≈ 1500 readout buffers
- < 10 ms latency

Several architectural options for the Level-2 Trigger have been defined and explored during the past few years. Basic options include parallel vs. sequential selection, separate control network vs. common network, „push“ vs. „pull“ (which in fact are two flavors of push). All options using parallel selection methods also use some processing and communication (network/switch) elements explicitly assigned for local processing and others assigned for global processing. The following „pure“ architectures A,B and C [Opt] plus a small number of combinations („Hybrids“ like C’ = A + C) have found their way into the demonstrator program.

- Architecture - A:

  Local feature extraction is performed by a small number of high speed, low latency FPGA processors fed by a high performance RoI collection network (RoIC), again based on FPGA technology. The features of all subdetectors are simultaneously extracted with a repetition rate of 10us. The global decision task is performed by a small farm of standard computers.
- Architecture - B:

  Local feature extraction is performed by a farm of dedicated local processors, communicating with the ROBs via a switch/network. Feature extraction of any given subdetector typically takes more than 10us on a local processor thus requiring resource allocation capabilities in the supervisor for both local and global processors.. The global task is performed in a farm of standard computers.
- Architecture - C:

  All features of a given event are processed by the same global processor, allowing a sequential selection scheme. A large high capacity network is needed to connect all ROBs with all processors. Unlike architecture-A and -B the global processors request the ROBs to send off their data.

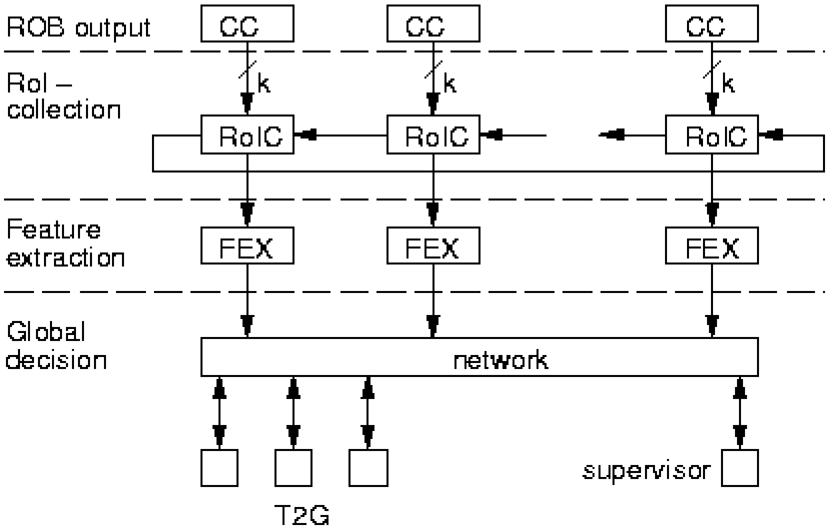
1.1 Principles of Architecture-A

Architecture-A follows the guidelines of local-global processing defined at the beginning of the ATLAS Level-2 trigger R&D work [TP]. The local processing part is based on the use of fast FPGA processors performing the feature extraction task. The FPGA processors are operating at the full 100kHz event rate with low latency and high bandwidth. Nevertheless the FPGA based processing is embedded into a farm based processing for the global LVL-2 decision and DAQ/EF allowing even hybrid systems (farm and FPGA processors working concurrently). The early (1996) paper models calculated a rather compact LVL-2 implementation of approx. 15 FPGA processors and about 50 microprocessors for the global farm. In these numbers a full scan TRT at 10 kHz and a full granularity Et calculation was included.

A major focus in the implementation was the scaleability of the system. The proposed implementation introduces a modular concept from components communicating exclusively with their nearest neighbors. This allows the system to be easily adopted to the different subdetectors and even the adaptation to high RoI flows (number of RoIs per event). The use of the fast FPGAs makes the model implementable even in today's technology and strongly limits the number of processors required.

1.1.1 The architecture model

An Architecture-A [Amod] type Level-2 trigger contains the following the functional layers:



- The ROB Interface couples architecture-A to the Readout buffers (ROB). For each ROB crate - containing a number of ROBs - it comprises a preprocessor and an interface to the RoI collection, the next hierarchical layer. The unified preprocessor/interface module is called Crate Concentrator (CC).
- The RoI Collection (RoIC) mounts full RoIs from the RoI fragments coming from the ROBs. Thus the RoIC needs a certain amount of buffering. A single RoIC usually controls several ROB-crates. The RoI data may be split between more than one RoIC unit. For that reason neighbored RoIC have a communication channel between each other.
- The feature extraction (FEX) performs the computation of algorithms in full assembled RoIs.
- The global farm consisting of a set of commercial microprocessors takes the global LVL-2 decision based on the features extracted by the previous layer.
- The supervisor interfaces to the Level-1 system and provides resource allocation and control.

In a pure architecture-A system the number of FEX processors equals roughly the average number of RoIs per event divided by the number of RoIs than can be processed simultaneously on a single processing element - plus some margin for safety and overhead. An allocation of FEX processors by the supervisor is not necessary as they keep in step with the 100kHz event rate.

The RoI collection layer takes advantage of the fact that the ROBs can be mapped into crates reflecting the geometrical layout of a subdetectors. Thus any RoI can be assembled from data fed into not more than two RoIC crates in a one-dimensional mapping or 4 crates in a two-dimensional mapping.

1.1.2 FPGA-processors: the key element of architecture-A

In standard processors - or computers - an algorithm is implemented by having a single CPU executing a sequence of single programming threads, each one at a time. On the other hand an FPGA-processor is the implementation of an algorithm directly in hardware which leads to an enormous increase in speed. This type of implementation is a well known approach for tasks near level-1 where dedicated application specific hardware is widely used. However the hardware of the FPGA-processor itself - all the chips on the board - remains the same for all different applications and the *effective hardware* is determined by the configuration loaded into the FPGA chips. This is why we use the wording FPGA-

processor to indicate that it is really a programmable object. And in fact an FPGA-processor is a device which combines the speed of hardware with the flexibility of software.
Currently most of the programming is done using a more or less hardware oriented programming language like VHDL but efforts are being made worldwide to develop more abstract design and test methods. Although FPGA-processors are, by the time being, not as easy to program as standard computers are, the superior level of performance justifies their use. This is especially true for many trigger related algorithms which employ high implicit parallelism with deep pipelines, unaligned bit manipulations, heavy use of look-up tables and very high I/O rates. FPGA processors have their strength where traditional microprocessors provide poor performance. Speed-up rates in the range of 100 - 1000 compared to RISC workstations have been demonstrated in the past [daq27].

At the university of Mannheim two different FPGA-processors have been developed, a large one - Enable++ - and a small one - microEnable - which have been successfully used in the ATLAS LVL-2 demonstrator program.
For the demo-A RoI collector another FPGA based device was developed at the University of Jena and implemented as a mezzanine extension of Enable++.

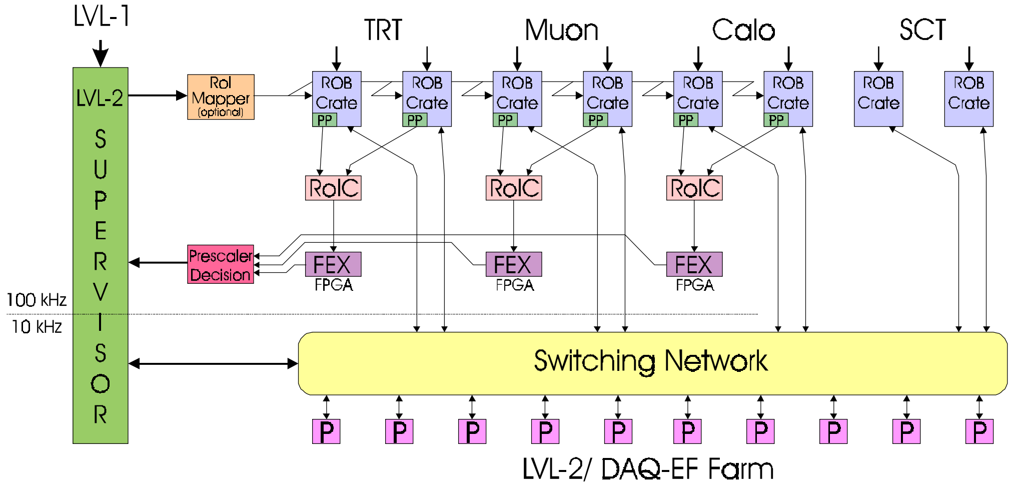
1.2 Hybrid System

Besides the pure A-type local feature extraction another concept was developed during the demonstrator program and introduced at the Argonne T/DAQ meeting in June 1997: the hybrid level-2 trigger featuring an FPGA based L2-prescaler, an FPGA based processing of the full TRT scan and a global processor farm.

At the time being there is no evidence that all high frequency and high bandwidth aspects of the L2-trigger can be handled using affordable networking technologies. The L2-prescaler takes the task of processing the incoming events at the full 100kHz rate using a low latency FPGA-processors system. Unlike the original architecture-A goal the aimed trigger reduction is now only 10. This simplifies the algorithms, allows the reduction in the number of subdetectors to be used and assures that no tracks are lost. As the prescaler is a real step in the overall sequential selection process the load on the network and processor farm is therefore reduced by roughly 90%.
In case of the full TRT scan problem an approach with encouraging results is to use the FPGA-processors to perform an initial trackfinding in the entire detector volume which then is refined by the global farm using these track to define RoIs.
Prescaling and the processing of the FullScan TRT - a co-processing task - can be done with the FPGA-RoIC/FEX system using the same type of hardware components.

1.2.1 Prescaling

The aim of the L2-prescaler is to reduce the event rate to be handled by the network and processor farm to an acceptable number, that is approximately 10kHz. To achieve this goal a number of components have to be added to the standard „single farm“ setup. The ROBs for some detectors are equipped with an additional high speed output feeding the separate data collection network - the RoIC - and there are FPGA-based FEX processors feeding a prescaler selection unit. Finally, the supervisor has to handle the reject and accept messages from the prescaler. One could say the prescaler is a second level-trigger operating concurrently with the processor farm. This might seem to be a large overhead, however the prescaler itself is a very compact system of a small number of crates and leads to significant savings in the number of farm processors. The prescaler is fully transparent and the two layers can operate totally independent using the supervisor as the single joint between the layers.



RoI guided setup

The data from the ROBs are sent to the RoI collection (RoIC) subsystem via fast links. Usually there will also be a preprocessing unit within a ROB crate. The grouping of the ROBs into crates considers the geographical mapping thus simplifying the tasks of fragment assembling. Typically an RoIC system will only need the data from it’s directly connected ROB crates plus the data from a single preceding RoIC. However, access to any other ROBs is possible.

The features of a reduced number of subdetectors - TRT, Muons and Calorimetry - are extracted by the local FEX processors in parallel and in step with the 100kHz event rate. The final number and types of used subdetectors has to be matched with performance and cost considerations and the 3 ones mentioned are most likely to be a maximum. The implemented algorithms utilize low cuts and must stay very close to 100% efficiency.

The local features are combined in the prescaler selection unit which is the first decision step in the level-2 sequential selection process. Due to the high message rates this task will also run on an FPGA-processor.

The level-2 supervisor for the hybrid system is slightly more complex than a „single-farm“ supervisor. The functions needed for the prescaler are basically the same as needed for an architecture-B type supervisor, however the resource management is much simpler. With every FEX processor running at the full event rate of 100kHz no processor allocation has to be done and monitoring and error handling is all that’s left. The Prescaler-Reject messages can be used to instantly free the ROB memories and the Prescaler-Accept messages will be forwarded to an allocated global processor, probably together with a summary of previous rejects from the prescaler. With this scheme 90% of the events can be rejected without loading the network and processor farm. A fast link from the supervisor to the ROBs must be present to keep the latency small.

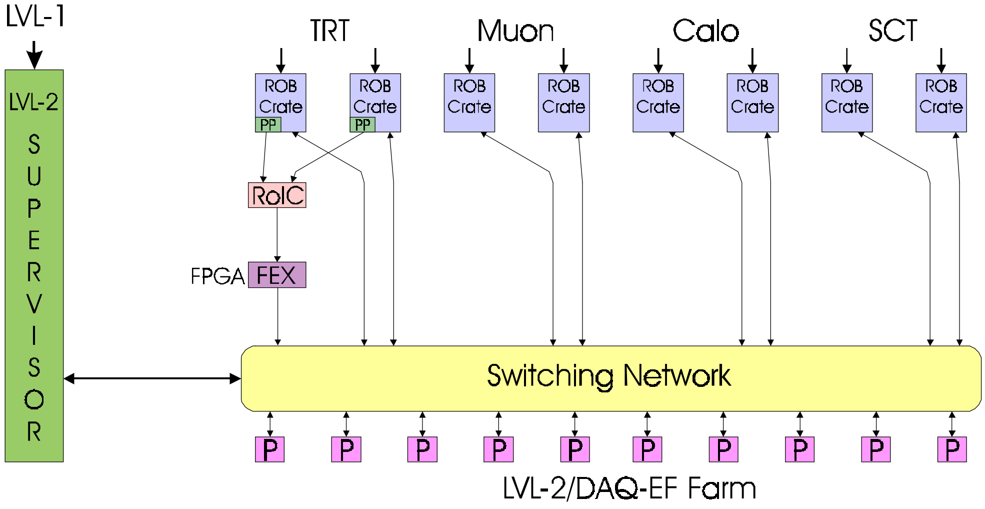
A more detailed description can be found in [daq78] and [papmo2].

1.2.2 Co-Processing

For the B-physics studies at low or medium luminosity runs a full scan of the entire volume of the TRT is necessary to define interesting regions with a rate of approx. 8kHz. The basic reference off-line

algorithm - without I/O processing - can run at about 4Hz on a 100MHz HP workstation. However a substantial part of this algorithm - the initial trackfinding - has been recoded using a LUT based approach. This version of the trackfinding algorithm has been successfully ported to the demonstrator-A Enable++ system running at approx. 7kHz including I/O processing.
Besides the high demands in terms of computing power the full detector scan also needs a large amount of data to be transferred via the network from the ROBs to the processors.

In a hybrid system the initial trackfinding as the most computing intensive task is performed by the FPGA-processor system. It operates effectively as a co-processor supporting the global farm processor designated for the given event.



B-physics setup

The data from the TRT ROBs are transferred to the RoIC system after a request from a global processor via dedicated high speed links. The RoIC is similar to the one used for the prescaler but simpler. The FEX task is to find the tracks and to calculate their Pt, this task can be simplified by dividing the detector into a number of pseudo-RoIs. The results are then sent via a network interface to the initiating global processor which completes the selection step.

Resource estimations
The recent paper model of the hybrid prescaler [daqxx] estimates a small number - around 10 - of FPGA-processors to be sufficient for both aspects prescaling and Full-Scan TRT coprocessing. The number of events to be processed by the global farm is reduced by 90% but the remaining events will on average pass more sequential selection steps compared to the average of all events. Therefore the effective load reduction will be less than 90%. However even an effective reduction of only 50% will lead to significant savings both in price and system complexity.

1.3 Preprocessing

Preprocessing in this context means a processing step executed at the ROB level in order to prepare data for the subsequent feature extraction. The most common preprocessing tasks are zero-suppression, compression and format conversion. A convenient place to perform preprocessing would be either in the ROB itself or, better, at the output of a group of ROBs, like a crate or an RSI. During the demonstrator program the preprocessing for the TRT was investigated, an algorithm was developed and two implementations[1] benchmarked. Preprocessing changes the TRT data format from pixel images with variable pixel length to coordinate lists with constant size. The datavolume is significantly reduced up to an occupancy of 30%. Also, the new format is much easier to be handled by standard CPUs leading to shorter FEX processing times. Implementations were done on a 300MHz DEC Alpha station and a 40MHz microEnable FPGA processor. Preprocessing times for a single ROB in the barrel region are 60 to 165 μs with the DEC Alpha and 12 to 27 μs with the microEnable, for occupancies between 3 and 100%. This results show both the usefulness of a preprocessing step to reduce the datavolume and the superiority of low-cost FPGA coprocessors over standard CPUs for such a task.

1.4 Programming issues

Programming of FPGA-processors like Enable++ or MicroEnable can be done using different approaches. The most common one is to use a hardware description either as a description language like VHDL[2], CHDL[3] or as a schematic. Such structural hardware description usually leads to maximum efficiency in speed and resource useage. The drawback is that the programmer must have an in-depth knowledge of the processor's internals. A description on a higher level can be done using VHDL in it's behavioral mode where design elements like processes, loops, if/case statements are available and the basic building blocks of the FPGA devices are hidden. Unfortunately present synthesis tools (= VHDL compilers) are not fully compatible, are less effective than a structural description and provide no or poor integration into external tools for simulation and debugging.

At the next level, the abstract description, much academic work has been done - including the Mannheim ppC language [ppC] - but no commercial compilers with acceptable performance are available. However recent design tools like SDL or StateMate which initially addressed the standard processor environment are finding their way into the FPGA domain. Together with the dramatic increase in complexity and speed of the latest and announced FPGA devices this might well be the most promising path towards simple and effective programming of FPGA-processors.

The main effort at Mannheim in this area is to develop the CHDL language into an easy-to-use tool for hardware-oriented description, addressing mainly the Mannheim FPGA-processors, while carefully watching the evolution of the commercial higher level design entry systems.

2 Demonstrator Program Architecture-A

2.1 Vertical Slice

In a full level-2 trigger system a chain or sequence of certain functional elements must be present. This sequence comprises:

- ROBs, representing the ATLAS subdetectors
- Local (per subdetector) data collection networks
- Feature extractors or local processing elements
- Global data collection networks
- Global processing elements
- Supervisor

[1] See DAQ Note 66: TRT Preprocessing

[2] VHDL: Very High Speed Integrated Circuit (VHSIC) Hardware Description Language. Initially developed to describe designs for full-custom ASICs or Gate-Arrays.

[3] CHDL: Intermediate level Hardware Description Language developed at the Mannheim University.

In the final system each of these elements will be present several - up to more than 1000 - times. The efforts necessary to build and evaluate such a large system were not available during the demonstrator program. In order to arrive still at reasonable results the *vertical slice* concept of the demonstrator program was adopted. It minimizes the number of replications per element for the real tests, verifies the paper and computer simulations with the measured results and uses the simulations to extrapolate from small to large scale.

The vertical slice used in demo-A was reduced to the bare minimum:

- A PC, emulating a number of ROBs of a single subdetector (TRT)
- The local data collection network was built by the RoIC subsystem, fed by a single 100MB/s Slink
- One Enable++ FEX processor
- A PC (the same as above), emulating the global network, global processors and the supervisor

This comparably small demonstrator is sufficient as the architecture-A system lacks some of the problems present in architectures using standard processors. For example, there is no complex processor allocation to be done by the supervisor as the FPGA-processors are operating in step with the 100kHz event frequency.

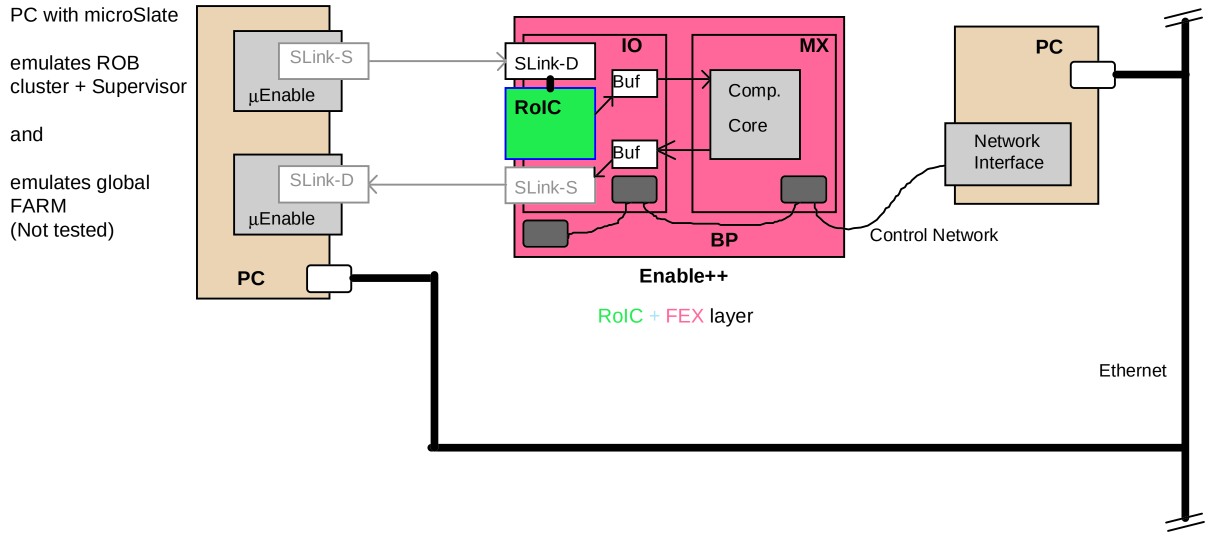


Figure 1: Demonstrator-A setup

2.2 Components

2.2.1 RoI collector: RoIC

Principles of the RoI collector

The collection of all RoI fragments belonging to a given event is an essential task for all feature extraction algorithms and is done in the architecture versions B and C via the local (per subdetector) or global network connecting all ROBs with all processors. In the architecture-A trigger system the RoI collection is done by special subsystem called RoI Collector[Kam98]. Hence in the demonstrator-A program there are dedicated RoI collection components - the RoIC - developed by the University of Jena and built as submodules of the Enable++ FPGA[4] processor system. The RoIC receives RoI fragments

[4] Enable++: The current (1996/1997) model of FPGA processor used in demonstrator-A for feature extraction. Also the possible basis for fast FEX and Coprocessing tasks in a „hybrid system“. Devloped by the University of Mannheim.

from a number of ROBs, assembles the fragments to full RoIs and transmits these RoIs to the associated FEX processor. The input and output connections are high-speed point-to-point links (Slinks with Fiber-Channel or electrical differential modules in the demonstrator program).

In architecture-A the RoIC system takes advantage from the fact, that the ROBs can be grouped into crates using their geographical localization within the detector. An optimized layout can guarantee that all possible RoIs can be assembled from ROBs in neighboring crates.

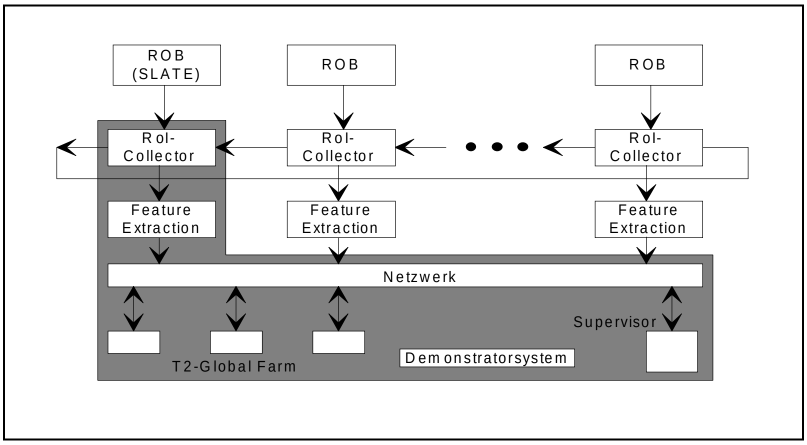


Figure 2: Demonstrator-A (grayed) part of Architecture-A

Implementation details and measurements

Considering the performance of the available components (links, memory speed etc.) a demonstrator system with 4 Slink inputs per RoIC was chosen. Three of these input being fed by the concentrated output of three ROB crates and one of them by the preceding RoIC system thus closing the loop. The RoIC output is directly connected to two channels of an Enable++ I/O board feeding the FEX processor. Due to resource limitations the 4 input channels to the RoIC had to be reduced to one. This has been compensated by a more complex structure of the used ROB datasets. Components only present through emulation are ROBs, Network, T2 Global Farm and Supervisor.

The demonstrator RoIC[DEK97] includes two modules - memory and control - each one covering two slots of a seven slot Enable++ I/O board. Up to seven memory modules can be managed by a single control module. Configuration and monitoring is accomplished by the host computer of the Enable++ system. Data input is provided by a single Slink line, first driven by a SLATE[5], later by a microEnable[6] board. Arbitrary distributions of RoIs and ROB contributions can be emulated simply by the corresponding definition of the dataset.

During the demonstrator program a number of parameters of this RoIC system has been measured[daq84]. In the following some of the results and important properties of the RoIC are presented.

	Measured	*Comments*
Available Memory/RoIC	2 MB	Avg. RoI volume is 10 .. 60 kB (Daq62)
Size of RoI fragments	0 .. 4kByte [7]	Average is below 1k (Daq62)
Max number of concurrent events in RoIC	64	
Max number of RoIs/event	8	
Max number of fragments/RoI	24	Up to 8. Up to 32 for jets (Daq62)
Input bandwidth	104MB/s	Max. 200MB/s @ 50MHz
Output bandwidth	100MB/s	Max. 400 MB/s @ 50MHz
Max event latency	3 .. 12 µs	

Overall the RoIC system showed a very good performance. The assembling of the RoIs is done with a latency smaller than the time needed for the bare transmission to the FEX thus leading to a deterministic

[5] SLATE: CERN developed VME based datasource for various high speed interconnects. Used with Slink interface.

[6] MicroEnable: PCI based FPGA processor, developed at the University of Mannheim and now commercially available, supports Slink interface in source and destination mode.

[7] Performance is reduced if a large percentage of fragments is below 64 Byte in size due to message overhead.

behavior. The minimal fragment size of 64 Byte where performance degrading starts is much smaller than can be hoped to be achieved by large networks. The event parameters (ROBs/RoI etc) are for most subdetectors better than the requirements defined by DAQ Note 62.

The present RoIC fulfills the demonstrator requirements and is a good starting point for the approaching Pilot Project. However, due to the similarity in their tasks, it seems useful to explore alternatively an integrated ROB/RoIC system.

Notes on operation in FullScan mode

In the Full-Scan mode for the TRT a more simple algorithm is sufficient for the RoI collection. The number of events to be handled concurrently is smaller and all ROBs participating in a given event are fixed and predefined by the segment (pseudo-RoI) identification to be processed. Thus a system capable of processing the high frequency RoI collection will easily be „downgraded“ for the Full-Scan operation mode.

2.2.2 Enable++ feature extractor

The Enable++ FPGA processor

The Enable++ FPGA processor [HKL95] is a 2^{nd} generation FPGA system consisting of a number of functional modules which can be set up to build an extremely powerful processing system. A 9HU VME-size board with 24 Xilinx FPGAs - the matrix board - is provided for computation. Another 9HU board with fewer FPGAs - called I/O board - can be equipped with up to 7 mezzanine I/O modules, each one with a max. datarate of 200MB/s. Finally an FPGA based configurable backplane allows to connect up to 5 Enable++ boards of either flavor in an arbitrary manner to facilitate high computing or I/O power - or both. System configuration and monitoring is done at host workstation connected to Enable++ via a transputer OS-link network.

The most important component of is course the matrix board. Four main features are responsible for it’s unique flexibility and processing power:

- Unlike previous FPGA based systems that were developed for a specific application the interconnect scheme, or topology, of the matrix board can be adapted to a broad range of different algorithms by means of reconfigurable interconnect switches.
- A total of 12MB static RAM is distributed among the 16 core FPGA chips in 3 banks of 128k * 36 Bits per FPGA. This allows both high parallelism even on the sub-chip level and very broad datapaths, up to 864 databits per clock cycle.
- The buffered interface between matrix board and backplane allows a very high data throughput over up to 8 channels at 200MB/s each.
- FPGA-processors implement an algorithm in hardware and operate without an operating system. Therefore there is ZERO overhead for taskswitching or I/O communication which usually uses a significant amount of computing performance on standard computers.

FPGA Feature extraction

The TRT feature extraction was the main issue in the demonstrator-A trigger algorithms work. The offline code (XRECON) was successfully modified at CERN [daq78] to allow look-up-table (LUT) based processing. The LUT algorithms consists of a number of steps and the most computing intensive one - the trackfinding - was implemented on both RISC computers and Enable++. The LUT stores predefined patterns with different Phi and Pt with and the principle is to find the best matching track candidates by histogramming and maximum finding. The performance depends mostly on the clock frequency and the number of bits (= patterns) accesible at each clock cycle. The algorithm itself is exactly the same for the low and high luminosity trigger but the contents of the LUT and the number of iterations is different for the two.

During the demonstrator-A program measurements of the feature extraction performance of Enable++ have been done, both for the RoI guided TRT (full FEX) and the Full-Scan TRT (partial FEX). A summary of the results is presented in the following[8].

Item	*Measured*	*Comments*
Parameters		
Input data bandwidth (Slink)	72MB/s	Max Slink datarate ~ 92MB/s @ < 2kB per packet
Number of event datasets	250	No pile-up
Number of valid tracks per event	1	
Number of found tracks	7.2	Initial trackfinding only. Further reduction to 1.x
Average event size	450 entries	4 bytes/entry on Slink, 2 bytes internally
RoI scan results		
Number of RoI search patterns	240	30 Phi, 8 Pt
RoI occupancy	30%	
Event rate	38 kHz	Resource usage ~ 30% of single Enable++ board
Event latency	5.3 μs	plus data transmission time
Possible parallelism at 100% board usage	3	Leads to > 110kHz event rate per board
Full Scan results		
Total number of full scan patterns	~ 80000	
Number of full Scan Pseudo RoIs	32	
Full Scan search patterns per Pseudo-RoI	2400	30 Phi, 80 Pt (down to 0.5GeV)
Number of parallel patterns on Enable++	448	Max is 864, only reduced capacity in test available
Occupancy	1%	
Number of passes for 2400 patterns	3	6 passes needed in test with 448 patterns/pass
Event rate	6 kHz	

Table 1: Results from Demo-A TRT FEX measurements.

Processor Type	*Execution Time*	*Comment*
100 MHz HP Workstation, no I/O processing	2000 μs	
400 MHz DEC Alpha Workstation, no I/O processing	400 μs	
50 MHz Enable++ FPGA Processor, with I/O processing	3.5 μs	Optimized design, no I/O bandwidth limitation
20 MHz Enable++ FPGA Processor, with I/O processing	26 μs	Unoptimized test design, with I/O bandwidth limitation

Table 2: A comparison of recent results for the high luminosity TRT trigger.

2.2.3 Slink

S-LINK is a CERN specification for an easy-to-use FIFO-like data-link which can be used to connect front-end to read-out at any stage in a dataflow environment [BMB96]. In demo-A Slink was used to supply data to the RoIC or FEX system and to receive the FEX results at the global emulator. Two different physical implementation were used with good results:

[8] Details of implementation and results are available from DAQ Note 78 and DAQ Note 84 - Demo-A results.

- 1GBit/S optical Fiber-Channel interface
- 80 MByte/s electrical differential interface

In both cases the Slink source or destination module was driven by a motherboard design implemented on either the Enable++ IO-board or microEnable respectively. The data throughput was close to the theoretical limit (max. 92 MB/s vs. 110 MB/s on Fiber-Channel) and limited by the Slink protocol implementation (control word processing delays). Although the datarate is not stable with variing packet sizes the Slinks in demo-A provide a high-frequency (= small packets) performance which is by far superior to any standard network (ATM etc.) interface.

All physical link modules were kindly given to demo-A by the CERN HSI group.

2.2.4 MicroEnable

The MicroEnable FPGA-processor is a PCI plug-in card with a single FPGA of the Xilinx XC4000 series, 0.5 to 2 MBytes fast SRAM, and the ability to plug on CMC or S-LINK daughter cards. MicroEnable can be equipped with any Xilinx XC4000EX/XL device between XC4013 and XC4085 providing a high degree of scaleability. The optimized device driver available for LINUX and WindowsNT 4.0 makes up to 125 MBytes/s DMA performance available to user applications---95 % of the theoretical PCI bandwidth. MicroEnable is currently programmable by VHDL and CHDL , a C++ based programming language. A graphical user interface combines CHDL compiler, simulator, emulator (using the FPGA readback capability), a C interface, and the control of the Xilinx place and route software. MicroEnable is available as a commercial product from Silicon-Software [SilSo], a spin-off company of the Mannheim University.

In demo-A MicroEnable was used for two tasks:

- TRT Preprocessing
  and
- Slink source and destination driver

The Slink driver implementation was used to supply data to the FPGA-based FEX and RoIC systems and to receive the FEX results. Although this was a „quick and dirty“ implementation the Slink could be driven at approx. 90% of it’s maximum performance level.

Unlike Enable++ MicroEnable is tightly coupled with it’s host computer both in terms of resource access and bandwidth. Together with the fast reconfiguration time of a few ms this allows optimal partitioning of preprocessing algorithms between MicroEnable and the host CPU. A sample algorithm for the preprocessing of the TRT was written at CERN [daq66].

In principle a task can be considered suitable for FPGA-based preprocessing if it has at least one of the following properties:

- High rates of short messages to be assembled
- Bit and data field manipulations
- LUTs instead of complex calculations
- No floating point arithmetic
- A series of pipelined operations executing concurrently on data as they flow through
- Parallel instances possible, since applying operations to items in a data stream, e.g. can do 4 words in parallel

Another PCI-based FPGA-processor is under development at the Weizmann institute which utilizes the novel FPGA family Xilinx XC6200. These devices provide simplified access to internal resources and very fast reconfiguration time allowing time-shared hardware multitasking.

Apart from the traditional programming approach - using VHDL etc. - the MicroEnable type of FPGA-processors will advantageously be used together with a library of hardware applets (HAPs). Applets are scaleable functional modules - probably commercial - similar to software macros which are interconnected to build an application. By this method an algorithm can be implemented without detailed knowledge of the underlying hardware properties.

2.2.5 Virtual components

To complete the vertical slice of demo-A a number of functional elements had to be emulated by virtual components. Corresponding to the list in paragraph *2.1 Vertical Slice* the virtual components in demo-A are:

- ROBs, representing the ATLAS subdetectors

  Seen by the target component - RoIC or FEX processor - a ROB is simply the source of data belonging to a certain region of a specific subdetector. Therefore an arbitrary number of virtual ROBs can be emulated an appropriate layout of the dataset containing the event fragments transmitted to the target. Jitter and delays due to uneven ROB occupancies are realized by programmable gaps in the data stream. If the accumulated datarate of the virtual ROBs exceeds the capacity of the single Slink used in demo-A the clock frequency of the target can be reduced to stay at a constant ratio of datarate vs. processing power.
- Global network and processing elements

  The global part of demo-A was realized simply by transmitting the results from the Enable++ system via an Slink to the emulating host. No algorithm was implemented and no performance measurements were carried out on the global part. This approach was justified by two facts:
  1. The main focus in demo-A was the investigation of a high-speed, low-latency local processing system which makes it’s prime difference to demo-B and demo-C.
  2. The global part of demo-A is very similar to but less complex than the corresponding part of demo-B and doublicate work was to be avoided.
- Supervisor

  The supervisor in an architecture-A like system doesn’t have to care about allocation of local processors as each one processes an event within the 10µs level-1 accept period. The intrinsic latency is in the order of a few 100µs including data transfers. Additional delays caused by uneven ROB occupancies are monitored by the RoIC system itself. Thus the demo-A virtual supervisor is fully contained in the format and ordering of the test event datasets.

2.3 Towards a hybrid system: integration test demo-A with demo-C

During the last week in November 1997 a short but successful integration test of the two different demonstrator projects A and C took place at Saclay. For this first step the following goals were defined:

- Attach the Enable++ system to the Saclay ATM network via a special node (PC with Win NT)
- Adapt the ***DATASOURCE software module*** to this special node for coprocessing operation (Full Scan TRT). The new term for this node will be COP
- Emulate the RoIC system on the COP feeding Enable++ via Slink

- Receive results from Enable++ via Slink on COP
- Run the complete Saclay framework including COP and Enable++

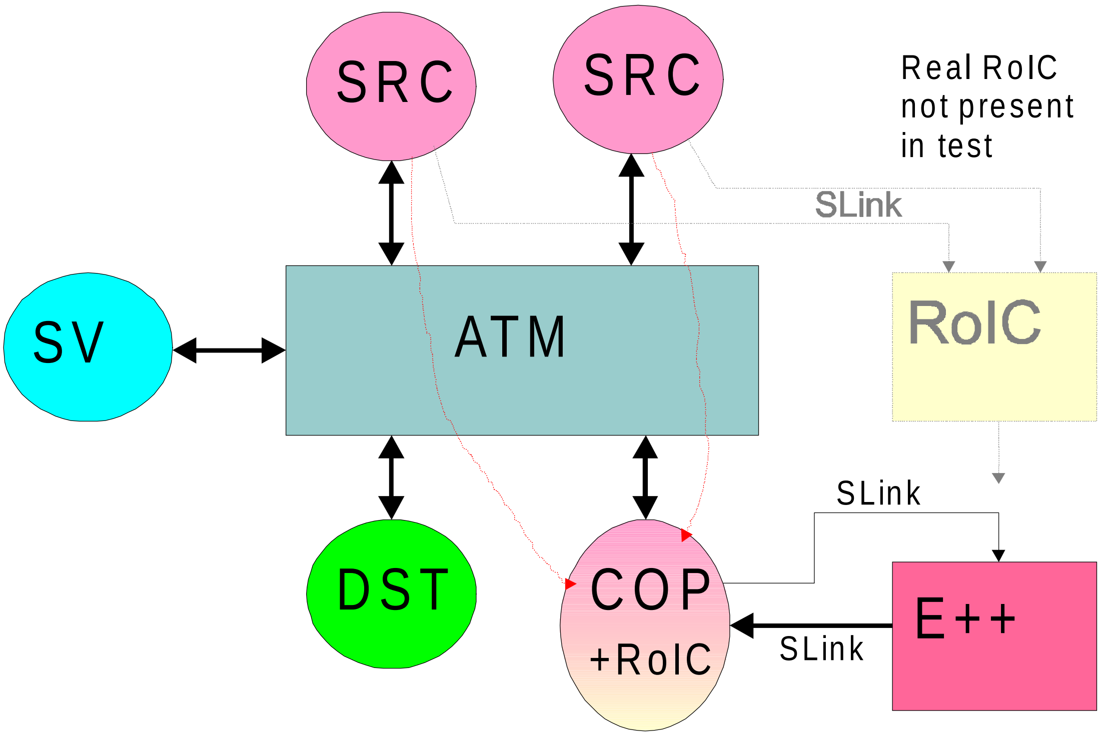
2.3.1 Implementation

To attach the Enable++ system to the network one PC was equipped with two microEnable boards, one driving an Slink line, one receiving an Slink line, and an ATM interface. The correct transmission and processing of datasets was verified with a standalone utility program.

The datasource software by Saclay was installed on this machine and modified to implement the following functions:

- Upon request from the GLOBAL via ATM send one data packet to Enable++ via Slink, thereby emulating the RoIC
- The RoIC emulation comes in two flavors: based on Slink and based on ATM. However for the attached Enable++ processor there is no difference
- Upon reception of a result from Enable++ via Slink send an acknowledge to the GLOBAL via ATM

Alternatively to the ATM interface a simple file I/O control interface was provided.



2.3.2 Results

The Slink transmitting and receiving routines had never been used in a real-time environment before and some modifications had to be applied to the device driver in order to get the system running stable enough to perform measurements. However the results obtained are very encouraging, even more if one considers the short period of time available to install and adapt both hardware and software. Conclusions from this first test are:

- A successful integration of the two demonstrators at the protocol level was performed
- The software framework provided by Saclay could be ported to the Mannheim machine with reasonable effort
- A simple API with a the functions **Open, Close, Request, Response** is sufficient to communicate with the microEnable FPGA processors serving the Slinks
- The complete demo-C system, including supervisor emulation and monitoring, was able to run stable up to 1kHz message rate with an increased latency from 600µs to 800µs caused by the COP. At 10kHz message rate quite many packets were lost.

Also a number of critical items has been identified:

- All OS specific functions must be encapsulated in a separate module with a defined API
- Time consuming system functions - like malloc - must not be used in the I/O functions
- Specific topics to be implemented and/or optimized in the Slink driver:
  - Interrupt handling
  - Queuing
  - Buffering
  - Blocking and non-blocking read
  - Don't use: DWORD, IOCTL, etc.
- Saclay's rule for byte-ordering should be followed:
  The source of information must convert data into the network byteorder which is defined in ***one single place*** *in the software framework*

3. Pilot Project work

The goal of the demonstrator-A groups was and is to explore the power of FPGA-processors and to make the advantages of this promising technology available for the ATLAS level-2 trigger. During the Pilot Project phase work is planned in the following areas:

- Input to modelling and simulation

  Due to manpower limitations demonstrator-A has provided less input to the modelling and emulation activities than the other demonstrators had. A set of parameters based on recent measurements and considering the impact of latest FPGA technology will now be produced for a number of FEX algorithms and different configurations (Full Fex, Prescaling, Coprocessing).
- Algorithms studies

  Enable++ oriented

  In the scope of the Enable++ based studies are the implementation of available feature extraction algorithms, especially for:
  - TRT Fex and Full-Scan
  - Calo Fex
  - Muon Fex
  - SCT Fex and Full-Scan

A very important aspect of this work is the partitioning of algorithms in parts best suited for implementation on FPGA-processors and on standard processors respectively. Also the use of Enable++ capability to change it's processor topology to match different algorithms shall be explored.
Another issue is to evaluate different networking schemes for the interaction between ROBs, FPGA-processors and global farm.
An Enable++ based testbed for evaluation of prescaling and coprocessing will be installed at Mannheim.

MicroEnable oriented
MicroEnable could be the prototype of a commercial multi-purpose trigger accelerator. To advance towards this goal the following areas will be investigated:

- Definition of a general FPGA-coprocessor API
- Integration of
  - ∗ RobIn
  - ∗ Preprocessing
  - ∗ Slink driver

  This work will be done in close interaction with the corresponding activity of the HPCN and RobIn groups at CERN and UCL.
- Distributed and partitioned FEX
- Further development of the XC6200 based PCI-coprocessor version

- Integration with the global farm at Saclay

  The integration work will continue as soon as there is sufficient progress on the individual items:
  - Improved Slink driver (Mannheim)
  - Updated COP software (Saclay)
  - Data format definition (Mannheim, Saclay, CERN)
  - Algorithms for coprocessing (Mannheim, Saclay)

  The testbed integration will probably be done using ATM as a continuation of the preceeding work. Anyway it is agreed that the two different RoIC options **Separate network via Slink** vs. **Shared network via ATM** should both be kept.
- Develop guidelines for the next generation FPGA-system. Evaluate *Compact PCI* as a basis for a tightly host coupled, high performance coprocessor.

4. Summary

The demonstrator-A project has produced very encouraging results concerning the use of FPGA-processors for critical tasks in ATLAS L2/DAQ like

- Feature extraction and RoI collection @100kHz
- Initial trackfinding for the full-scan TRT
- TRT Preprocessing
- High-speed communication with Slink

A number of FPGA-based components have been developed by different groups at different places and successfully integrated into the demonstrator-A system:

- Low cost 80MB/s Slink cards at Cracow, Poland
- RoI collection system at Jena, Germany
- FPGA-processors Enable++ and MicroEnable at Mannheim, Germany
- FPGA-coprocessor prototype with fast reconfigurable FPGAs at Weizmann, Israel

Additionally the demonstrators of architecture A and C have been successfully combined at Saclay to a small-scale hybrid system.

The results obtained from the demo-A measurements show that the high-frequency and high-bandwidth requirements of ATLAS level-2 can be fulfilled even with the - comparatively old - technology used in demo-A. The concept of the level-2 prescaler allows to move a significant portion of the event rate from the global processor and network system to a small cluster of Enable++ FPGA-processors. For preprocessing tasks even the small MicroEnable device provides a speedup factor of 10 compared to high-end RISC computers. State-of-the-art FPGA devices providing a total increase in speed and complexity by a factor of 10 or more will give additional performance at a much lower price. Recent systems - like MicroEnable - will use the potential of progressing technology thus opening the path to commercial multi-purpose coprocessors which take over the load where standard processors do - and will - not provide an economical alternative.

5. References

[TP] ATLAS Technical Proposal, CERN/LHCC/94-43, 1997
[HKL95] H. Högl, A. Kugel, J. Ludvig, R. Männer, K.-H. Noffz, R. Zoz. Enable++: A Second Generation FPGA Processor. In IEEE Symposium on FPGAs for Custom Computing Machines. Pages 45-53. 1995.
[BMB96] O. Boyle, R. McLaren, E. van der Bij. The S-LINK Interface Specification. Technical report, ECP Division, CERN, 1996. available at http://www.cern.ch/HSI/s-link
[Opt] ATLAS Level-2 Trigger Groups „Options for the ATLAS Level-2 Trigger", in Proc. IEEE Conf. On Computing in High-Energy Physics, Berlin, 1997
[papmo2] Level-2 Prescaling: A Hybrid Model of Architecure A. http://www-mp.informatik.uni-mannheim.de/groups/mass_par_1/projects/model_A2.ps
[daq27] DAQ note 27 - Algorithms in second-level triggers for ATLAS and benchmark results, 1994
[daq62] DAQ note 62 - Detector and Readout specifications, and buffer RoI relations, for the Level-2 trigger demonstrator, 1997
[daq66] DAQ note 66 - TRT Preprocessing - Algorithms, Implementations and Benchmarks, 1997
[daq78] DAQ note 78 - A Hybrid Approach for the ATLAS Level-2 Trigger, 1997
[daq84] DAQ note 84 - Demonstrator Results Architecture "A", March 98
[daqxx] DAQ note xx - FPGA Prescaler paper model, to be released in March 98
[ppC] R.Zoz: Eine Hochsprachen-Programmierumgebung für FPGA-Prozessoren. PhD Thesis, University of Mannheim, Germany, 1997
[SilSo] Silicon-Software, manufacturer of the commercial MicroEnable FPGA-processor. http://www.silicon-software.com
[DEK97] V. Dörsing, W. Erhard, P. Kammel, A. Reinsch, H.U. Zuehlke, RoI Collector implementation for the Level-2 Trigger of ATLAS, 10. IEEE Real Time Conference, Beaune, 1997
[Kam97] P. Kammel, Proposal for a COMPACT ROB based on components of the Region of Interest Collector (RoIC), ATLAS Trigger/DAQ Workshop, Marseille, http://isun04.inf.uni-jena.de/kps/roic.html, 1997
[Kam98] P. Kammel, Rekonfigurierbare Einheit zur Datenreduktion fuer den Level-2 Trigger von ATLAS, Friedrich-Schiller-Universität Jena, March 98