



PUBLISHED BY INSTITUTE OF PHYSICS PUBLISHING AND SISSA

RECEIVED: October 26, 2007

ACCEPTED: December 14, 2007

PUBLISHED: January 28, 2008

## TECHNICAL REPORT

# The ATLAS ROBIN

R. Cranfield,<sup>g</sup> G. Crone,<sup>g</sup> D. Francis,<sup>a</sup> B. Gorini,<sup>a</sup> B. Green,<sup>b</sup> M. Joos,<sup>a</sup> G. Kieft,<sup>d</sup> A. Kugel,<sup>c\*</sup> A. Misiejuk,<sup>b</sup> M. Müller,<sup>c†</sup> V. Perera,<sup>e</sup> J. Petersen,<sup>a</sup> J. Strong,<sup>b‡</sup> P. Teixeira-Dias,<sup>b</sup> L. Tremblet,<sup>a</sup> J. Vermeulen,<sup>d</sup> F. Wickens,<sup>e</sup> M. Yu<sup>c</sup> and G. Unel<sup>a†</sup>

<sup>a</sup> CERN, Geneva, Switzerland

<sup>b</sup> Royal Holloway University of London, London, U.K.

<sup>c</sup> University of Mannheim, Mannheim, Germany

<sup>d</sup> FOM - Institute SAF and University of Amsterdam/Nikhef, Amsterdam, Netherlands

<sup>e</sup> Rutherford Appleton Laboratory, Didcot, U.K.

<sup>f</sup> University of California, Irvine, U.S.A.

<sup>g</sup> University College London, London, U.K.

<sup>h</sup> Bundeskriminalamt, Wiesbaden, Germany

E-mail: [kugel@ti.uni-mannheim.de](mailto:kugel@ti.uni-mannheim.de)

**ABSTRACT:** The ATLAS readout subsystem is the main interface between ~1600 detector front-end readout links and the higher-level trigger farms. To handle the high event rate (up to 100 kHz) and bandwidth (up to 160 MB/s per link) the readout PCs are equipped with four ROBIN (readout buffer input) cards. Each ROBIN attaches to three optical links, provides local event buffering for approximately 300 ms and communicates with the higher-level trigger system for data and delete requests. According to the ATLAS baseline architecture this communication runs via the PCI bus of the host PC. In addition, each ROBIN provides a private Gigabit Ethernet port which can be used for the same purpose. Operational monitoring is performed via PCI. This paper presents a summary of the ROBIN hardware and software together with measurements results obtained from various test setups.

**KEYWORDS:** Digital electronic circuits; Data acquisition concepts; Data acquisition circuits.

\* Corresponding author.

† Now at Bundeskriminalamt, Wiesbaden, Germany.

‡ Deceased.

---

## Contents

|                                                |           |
|------------------------------------------------|-----------|
| <b>1. Introduction</b>                         | <b>1</b>  |
| <b>2. Readout system</b>                       | <b>2</b>  |
| 2.1 Environment                                | 2         |
| 2.2 Requirements                               | 3         |
| <b>3. Development and production procedure</b> | <b>4</b>  |
| <b>4. Hardware design</b>                      | <b>5</b>  |
| <b>5. Functional description</b>               | <b>6</b>  |
| <b>6. ROS software</b>                         | <b>9</b>  |
| <b>7. Performance results</b>                  | <b>10</b> |
| <b>8. Conclusion</b>                           | <b>12</b> |

---

## 1. Introduction

The ATLAS [1] experiment is one of the four large experiments aimed at studying high-energy particle interactions at the Large Hadron Collider (LHC). The design of the ATLAS Trigger and Data-Acquisition (TDAQ) system has been documented in [2], the base-line data-flow of the system in [3]. The ATLAS readout subsystem (ROS) is presented in [4], which incorporates the ROBIN as a key element to handle the high-rate, high-bandwidth input stream from the detector and to buffer events during the decision latency of the High-Level Trigger (HLT) system.

A ROS PC is equipped with four<sup>1</sup> ROBIN cards and each ROBIN attaches to three optical links and communicates with the HLT system for data and delete requests. According to the ATLAS baseline architecture this communication runs via the PCI bus of the host PC. In addition, each ROBIN has a private Gigabit Ethernet port which provides an alternative path for this communication.

A total of 700 ROBIN cards have been produced and tested. Installation and commissioning started in 2005 and finished in 2007. The majority of the ROBINS have been in use continuously since installation. A low level of hardware problems has been observed on a few boards, but most have been resolved by firmware resets or replacement of faulty components. A final production run for another 70 cards to satisfy the needs of additional sub-detectors and test-beds has been completed in August 2007.

---

<sup>1</sup> Four ROBIN cards is the typical case. Some ROS-PCs serving high-bandwidth detectors are equipped with 3 ROBINS only, while others serving low-bandwidth detectors have five ROBINS.

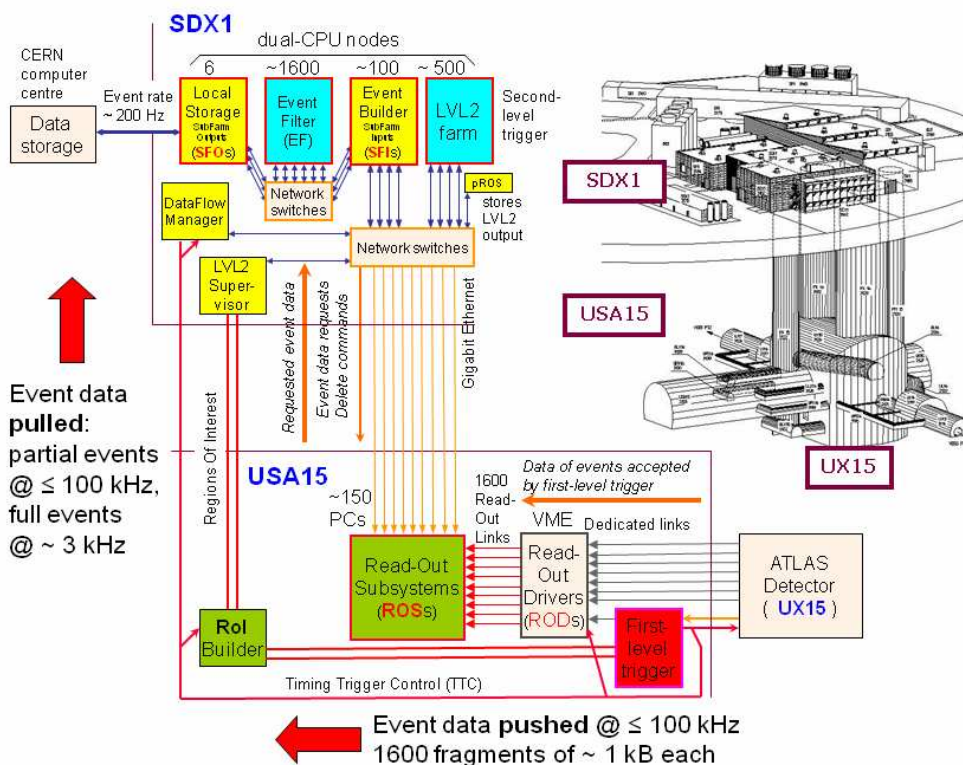


Figure 1. ATLAS TDAQ.

In this paper an overview of the ROBIN hardware and software and of relevant measurement results is presented. For further information see [2] and [4] for system aspects, [5], [6] and [7] for design documentation, [8] for the user manual and [9] and [10] for performance measurement results.

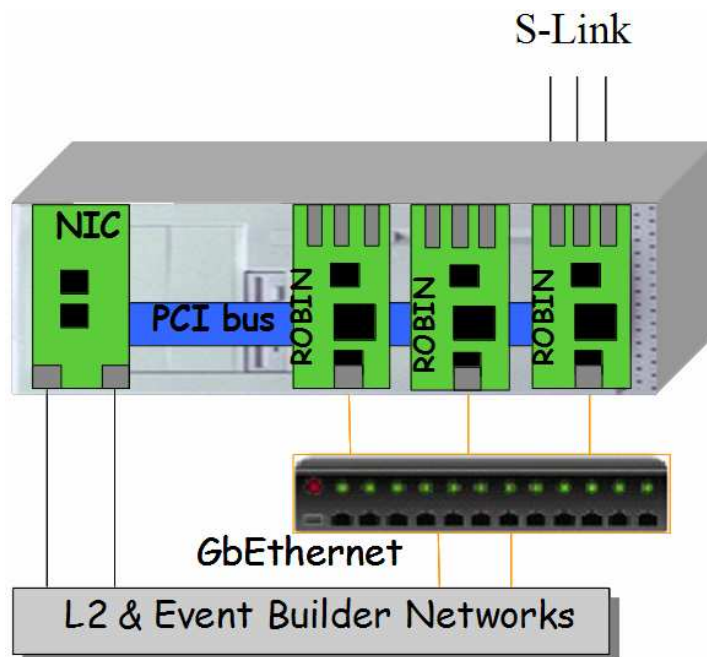
A glossary of terms and abbreviations used in this paper is given in [11].

## 2. Readout system

### 2.1 Environment

The ATLAS ROS is one of the major components of the ATLAS TDAQ system (figure 1). Event data are generated in the detector and sent at the level-1 trigger rate ( $75/100$  kHz<sup>2</sup>) via 1600 Read-Out-Links (ROL) to the ROS. The ROS is composed of roughly 160 PCs, each one equipped with typically four ROBIN cards, which decouple the ROS PCs from the high-rate, high-bandwidth input load coming from the ROLs. For each event one of the level-2 processors in the LVL2 farm decides whether the event should be retained using event data requested from a subset of the detector, the so-called region-of-interest (RoI). After the typical level-2 latency of a few tens of ms the event is either marked for acceptance or rejection. Reject decisions are collected by the dataflow-manager (DFM) subsystem and broadcast to the ROS in groups of typically 100 events. For an accepted event the event-builder/switch-farm-interface (EB/SFI)

<sup>2</sup> 100 kHz is the maximum rate (and requires upgrading of parts of the TDAQ system), 75 kHz is the nominal rate.



**Figure 2.** Schematic overview of a ROS PC with ROBINs connecting to the Level-2 and Event Builder networks via PCI bus as well as via their dedicated interfaces.

subsystems builds a full event record from the data of the entire detector – all ROBINs including the ones of the RoI regions will be requested in this case – and subsequently marks the copy of the event remaining in the ROS for deletion via the DFM. The requirements on the ROS are documented in [12].

All communication between ROS, LVL2, DFM and EB run via a Gigabit Ethernet (GbE) network. In the baseline ROS the ROBINs communicate only with the ROS-PC, via PCI. In the switch-based ROS scenario ROBINs are additionally connected to the network via their private GbE port and may for example respond to data requests and delete requests over PCI and GbE concurrently. It depends on the overall system configuration how the two interfaces are used.

However, all communication not directly related to event data, for example configuration and operational monitoring, is always via the PCI interface. The additional GbE path provides for extra scalability of the ROS, when the PCI bus reaches its limits. Figure 2 shows the general arrangement of a ROS PC with ROBINs.

## 2.2 Requirements

According to [4] the rates of RoI requests received by the ROS PCs have been estimated with a "paper model", where "paper" refers to "back-of-the-envelope" calculations. In practice, the required calculations are done with a C++ program. The calculations are based on the assumption that the RoI rate does not depend on the  $\eta$  and  $\Phi$  of the centre of the RoI, but only on the area in  $\eta$ - $\Phi$  space associated with the RoI. The RoI rates for each possible RoI location and type (electromagnetic shower, jet, single hadron, muon) are obtained with a straightforward calculation. Inputs for it are: the level-1 accept rate, exclusive fractional rates for the various level-1 trigger menu items, the number and type of RoIs associated with each trigger item and the  $\eta$ - $\Phi$  area associated with the RoI location and type. The rates of requests received by each ROS PC and the request rates for each RoI are then obtained using the mapping of the detector onto the RoIs, the acceptance factors of the various level-2 trigger steps, and the RoI rates for

the RoI locations associated with the  $\eta$ - $\Phi$  areas from which data are requested (RoI type and detector dependent). The model predicts that for the design luminosity trigger menu, for each level-1 accept on average buffered event fragments will be requested from 16.2 ROLs connected to 8.4 ROS PCs, i.e. from on average 2 ROLs per PC for these ROS PCs. This illustrates that RoI-driven processing is a key property of the ATLAS level-2 system. With 1 - 1.5 kByte per fragment and 100 kHz level-1 accept rate a bandwidth of  $\sim 2$  GByte/s is needed, instead of  $\sim 150$  GByte/s for full read-out. Furthermore the maximum rate of requests per ROL by the level-2 trigger is about 5 - 8 kHz, instead of 100 kHz that would be required for full read-out. The ROS PC takes care of distributing requests to the ROBINS (for each ROL individually) and of partial event building. The maximum request rate and output event fragment rate per ROS PC, for level-2 triggering only, are about 20 - 25 kHz for a few ROS PCs; for the remaining PCs they are less than 15 kHz. The maximum data volume to be transferred per ROS PC is predicted to be less than 35 MByte/s. Note that, if needed, there is some room for minimizing request rates and data volumes per ROS PC by interchanging ROL connections.

Event Building is anticipated to take place at a maximum rate of  $\sim 3$ - $3.5$  kHz with an associated total bandwidth requirement of 3-5 GByte/s. In view of possible non-standard triggers the requirement for the request rate per ROL (by Level-2 and Event Builder) that the ROBIN should be able to handle has been set to 21 kHz, i.e. much higher than the requirements resulting from the paper model.

### 3. Development and production procedure

The final ATLAS ROBIN is the culmination of activities [13], [14] over an extended period at different institutes. The development process for hardware<sup>3</sup> and software<sup>4</sup> followed the standard CERN rules with prototyping and a number of reviews. The design team was formed in early 2002 and developed a prototype design, which was reviewed and accepted in October 2002 [15]. After the successful production and test of the prototypes the design of the final ROBIN was developed and reviewed in May 2004 [16]. Initial samples of the final ROBIN became available in Feb 2005 and confirmed the performance expectations. Subsequently the ROBIN passed the production readiness review (PRR) in March 2005 [17].

The production of the 700 cards was divided in two halves, produced at sites in the UK and Germany. In the UK the main production was held until a pre-series<sup>5</sup> batch had been produced, to gain experience with the UK production site and to provide ROBINS for a 10% "Pre-Series" TDAQ system. The German part was ordered immediately after the PRR, as all previous prototypes and samples had been produced there. The German production was completed in Oct 2005, the UK production in May 2006. In Sept 2006 the majority of ROBINS was installed in the 160 ROS PCs procured so far. A final production run of 70 cards has been performed in August 2007 at the German production site and the boards have been tested in early September 2007.

---

<sup>3</sup> Hardware relates here to the physical device and the VHDL code for the FPGA. The production and assembly of the PCBs was done by external companies.

<sup>4</sup> Software relates to four distinct areas: boot-loader and application running on the ROBIN, device driver and applications running on the host PC.

<sup>5</sup> A pre-series ROS system was built in order to do tests on a 10%-system. The pre-series ROBIN batch was 50 cards.

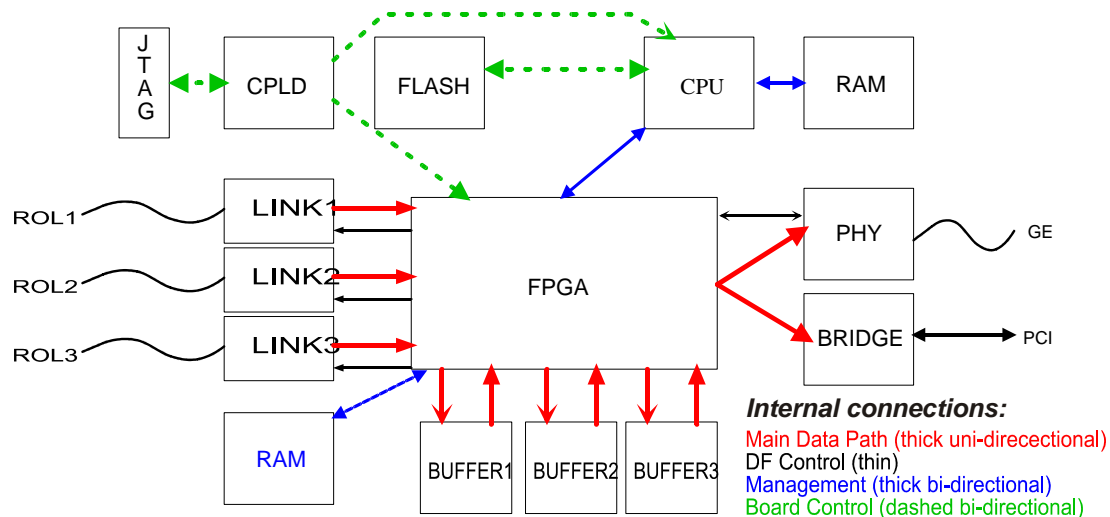


Figure 3. ROBIN Block Diagram.



Figure 4. The ROBIN card. The connectors at the left (from top to bottom) are for Gigabit Ethernet (electrical connection) and for three HOLA S-links (optical). The FPGA is at the center of the board (part labelled with "Xilinx"), the chip below it is the PCI bus interface, immediately right of the PCI interface the PowerPC 440 microcontroller can be seen. The connector at the right is for the 100 Mbit Ethernet connection of the PowerPC. The left connector of the pair of connectors at the lower right corner of the board provides an RS-232 connection to the microcontroller. Just above the FPGA is a connector for a mezzanine board.

#### 4. Hardware design

The hardware of the ROBIN device, shown in figure 4, is based on two main elements: a Xilinx Virtex-II 2000 Field-Programmable-Gate-Array (FPGA) [18] in a 896-pin package and a PowerPC PPC440GP microcontroller [19] operated at a clock frequency of 466 MHz. I/O-interfaces and buffer memories are attached to the FPGA, see figure 3. The FPGA handles all real-time, high-rate and high-bandwidth functions; the processor performs book-keeping and less time-critical functions. Each channel owns a dedicated input port, a buffer memory and

independent resources inside the FPGA, to avoid bottle-necks. Multiplexing of the channels towards the output interface is done by the DMA-engine<sup>6</sup> in the FPGA.

The FPGA is attached to the external bus of the CPU and all communication runs via first-in/first-out (FIFO) and dual-ported (DPR) memory elements, embedded in the FPGA. The connectivity to the ATLAS detector is provided by three read-out-links (ROLs), following the HOLA S-Link specifications [20], [21]. They are implemented with 2 GBits/s optical transceivers, discrete serialisers / de-serialisers (TLK2501 [22]) and an S-Link protocol engine embedded into the ROBIN FPGA as a VHDL core.

The PCI interface uses a discrete PCI-to-local bus bridge device PLX PCI9656 [23] attached to the FPGA. The PCI-side operates at 64 bit / 66 MHz while the local side is only 32 bit wide and limits the maximum throughput to 264 MB/s. The electrical Gigabit Ethernet interface uses a discrete Marvell Alaska Ultra+88E1011S transceiver (PHY) [24] and a media-access-controller (MAC) embedded in the FPGA. The connection to the HLT farm (level-2 and EB/SFI) can run over both PCI and GbE interfaces concurrently.

Data coming from the ROLs is stored in SDRAM buffers (64 MB per channel). Additional memory is available for the FPGA (512 kB static SRAM as network packet buffer) and for the processor (128MB DRAM as main memory, including buffer management data structures).

Low-level control (e.g. reset, JTAG) of the board is accomplished by a Complex Programmable Logic Device (CPLD) [25] device. An 8 MB FLASH memory stores the boot-code for the processor, the FPGA bit stream and some environment variables. In addition it provides a one-time-programmable sector for serial number and manufacturing information.

Factory programming is done via JTAG by first loading the CPLD device. The FLASH memory can be loaded subsequently either from a JTAG-debugger attached to the processor or via PCI, using a special configuration file for the FPGA. The FPGA can be configured by one of three interfaces: a separate JTAG port<sup>7</sup> on the CPLD JTAG connector, a JTAG port emulated by the PCI bridge or a JTAG port emulated by the CPU.

An 80 pin connector is available to add mezzanine boards providing extra functionality. The connector has 50 single-ended signals, one input clock and one 66 MHz output clock.

The built-in Ethernet port of the processor is routed to an additional RJ45 connector at the rear end of the PCB. This interface can be used by appropriate OS-software in stand-alone operation.

## 5. Functional description

The FPGA receives events fragments from the ROLs and stores them in the associated buffer (figure 5). A paged memory layout is used to store fragments, with a programmable page size. Possible values of the page size are between 1 kB and 128 kB, the typical value is 2 kB. Free pages are provided by the CPU via a Free-Page-FIFO (FPF), which takes up to 1k entries. For every incoming fragment a new page entry is taken from the FIFO, which gives the starting address for that fragment in the buffer memory. If the size of the fragment is larger than one page, subsequent entries are taken from the FPF. The fragment size at this stage is only limited by the number of available free pages. When a page is full or the fragment itself is complete, the page information (starting address, length and status) is written to the Used-Page-FIFO (UPF), from where the CPU picks it up. The UPF can buffer 256 page records with 4 words per entry: page info, level-1 event identifier (L1ID), status and run-number. The FPF and UPF for every

<sup>6</sup> PCI and GbE have separate, independent DMA engines.

<sup>7</sup> This JTAG port is used only during factory testing to verify the functionality of the FPGA



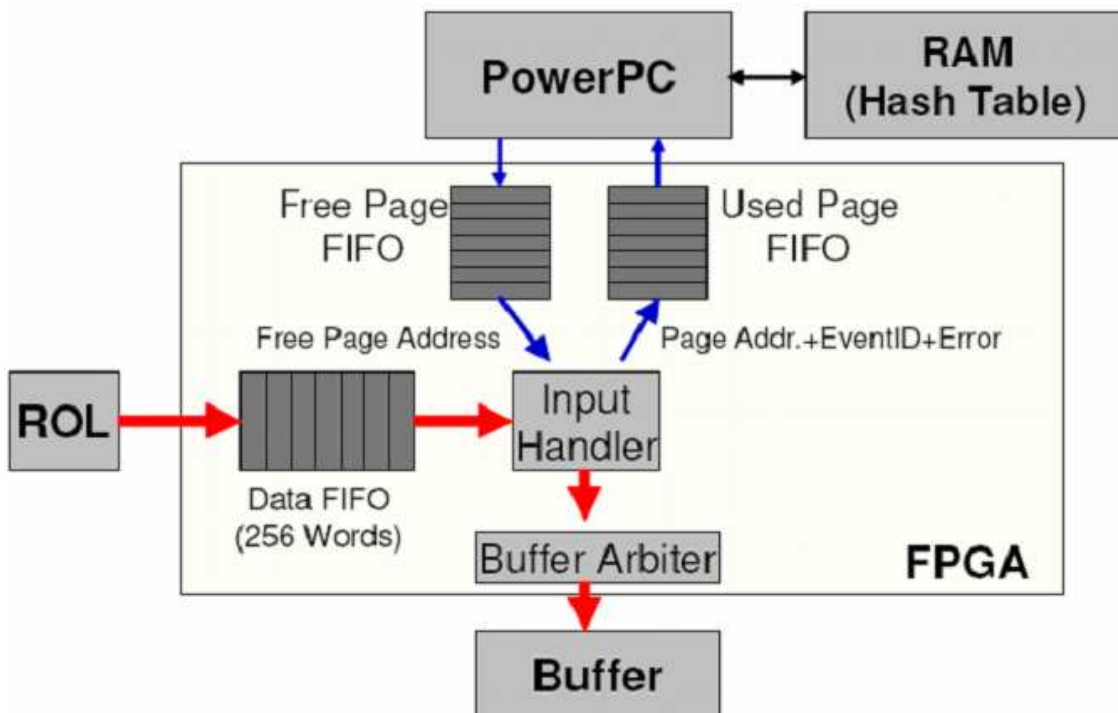


Figure 5. Handling of incoming fragments.

ROL are separate and independent. The UPF status includes error information related to fragment transmission quality and format.

The CPU picks up the information from the UPF and performs the book-keeping. A fragment is identified by its 32 bit extended L1ID,<sup>8</sup> which is in general monotonically<sup>9</sup> incrementing. To find quickly the information related to a L1ID a simple hashing algorithm is used:

1. A hash-key is created from some (typically 16) lower bits of the L1ID
2. All pages with the same key are stored in a linked list

The time to find a fragment is determined by the average search depth in the linked list. With 20 bit<sup>10</sup> hashing the entire range of L1ID for a 0.1 Hz event-counter-reset (ECR) rate is covered and the probability for multiple entries is very low, leading to an average search depth of close to 1. If a duplicate L1ID is recognized, the pages allocated for the previous fragment are freed and the new fragment is entered at a new location. The new fragment is marked as a “duplicate”, in the ROB status field. Finally the processor periodically moves free pages from the list of free pages to the FPF. Fragments with errors (e.g. invalid format) are kept, provided they appear to have at least a proper L1ID.

<sup>8</sup> The ROBIN firmware supports the use of a second 32-bit field called run-number to identify fragments, thus having a 64-bit identification mechanism. This feature is not yet used by the ROS software.

<sup>9</sup> The 24-bit primary L1ID is reset by an event counter reset at a rate of 0.1 to 10 Hz. The resets are counted and the value is mapped into the most significant byte of the L1ID field. The combined field is called extended L1ID.

<sup>10</sup> A large number of hash-bits reduces the time to find a fragment, but it badly affects the performance of garbage collection and other maintenance functions. As the number of stored fragments cannot exceed 64k it is a good compromise to use 16 bits only for the hash-key.



The ROL buffers are accessed in a virtual dual-ported fashion, using fixed time-slice arbitration (1:1) for WRITE and READ paths, in order to guarantee full input bandwidth (160MB/s according to the S-Link specification).<sup>11</sup> During normal operation access to the buffers is done in write-only mode from the ROL side or in read-only mode from the DMA side. There is an additional path available, which allows the CPU to write and read data, however this interferes with the other interfaces and is used only during self-test or in S-Link emulation mode.

Sending messages via PCI to the ROBIN is a 2 step process. First, message data is written to a DPR (see page 6) in the FPGA. After the entire message has been written, a message descriptor (single 32 bit value) is written to a FIFO, specifying offset and length of the message. The size of the DPR is 2 kWords (8 kB), the size of the FIFO is 32 words. Using a DPR for the message enables the ROBIN-CPU to use a pre-fetching bus access, which provides a better performance compared to reading all data from a FIFO.

The GbE message interface uses basically the same mechanism. The FPGA receives Ethernet packets and transfers them directly into the external static memory, which is organised as a ring-buffer. In addition, it writes the packet descriptor (size and offset in the memory) to a FIFO. A flow-control message is sent when the number of messages in the FIFO or the occupancy of the memory reaches a hard-wired threshold.

Responses are sent from the CPU to either PCI or GbE via separate DMA engines in the FPGA (PCI example: figure 6). The handling is identical in both cases and comprises:

1. Write DMA descriptor to DMA FIFO, specifying header data size, buffer data size, buffer offset and ROL channel number. For PCI, the destination address descriptor is given as well.
2. Write header data (if any) to DMA FIFO

After step 1 the DMA engine starts to evaluate the descriptor, then fetches the specified amount of header data (if any) from the FIFO and forwards it to the output interface (MAC or PCI). When all header words are processed, data from the buffer (if any) is appended. The DMA FIFO has a size of 512 words, so multiple responses can be queued for the DMA engine, without blocking the CPU. The GbE DMA engine has the ability to terminate the header section on an odd 16 bit boundary (= suppress the upper 16 bits of the last header word), which is required to run IP over Ethernet.<sup>12</sup>

The CPU performs operational monitoring on a variety of items, e.g. number of fragments, number of messages, fragment size histogram, buffer occupancy histogram, etc. Extensive error checking is done as well, both by the CPU and the FPGA.

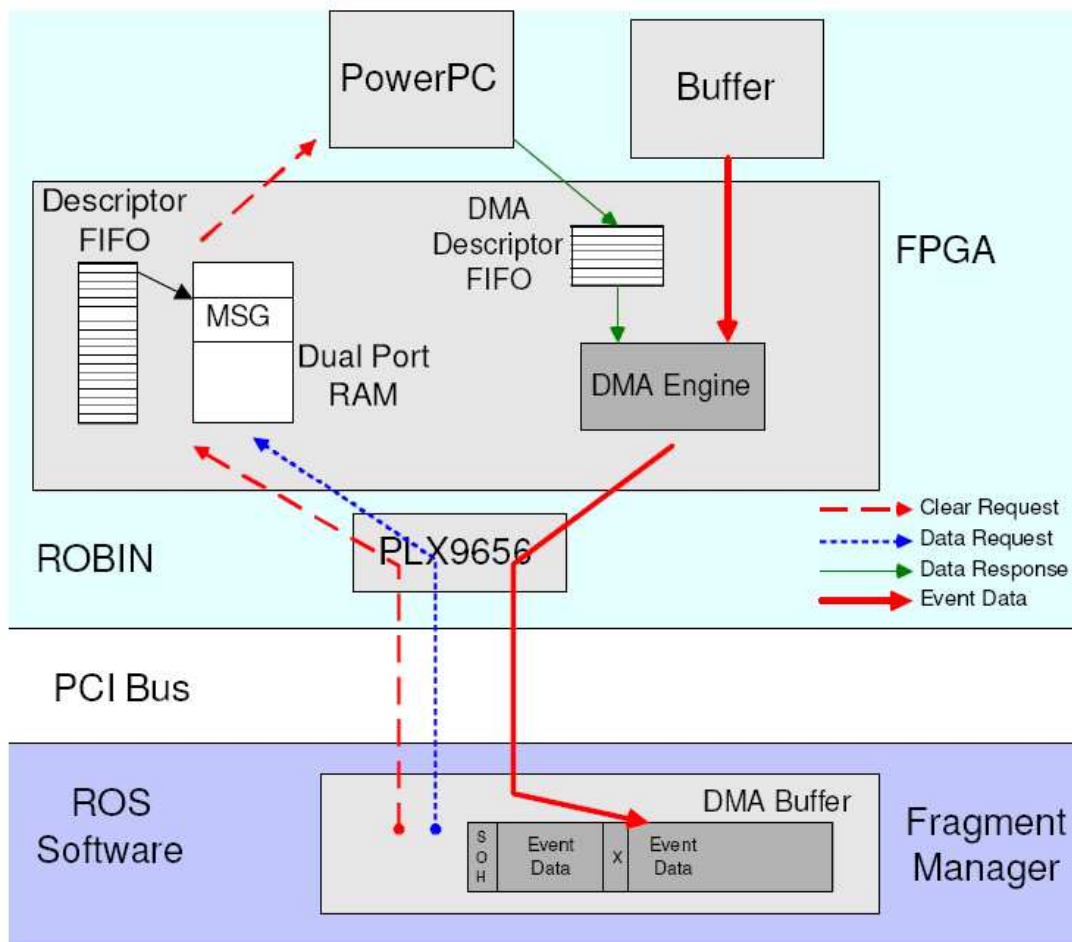
The FPGA compares the actual length of an event fragment against the length information in the header field. An eventual mismatch is flagged to the CPU. In addition, the FPGA calculates a CRC on the entire fragment and appends the CRC value to the last memory page of the fragment. This CRC can be used to verify data integrity when the event is further processed upstream.

The CPU checks for format errors indicated by the FPGA, minimum fragment size and incorrect sequence of the event-id. Fragments are only discarded at this stage, when they contain both a sequence error and one of the other errors.

---

<sup>11</sup> The optical links operate at 2 Gb/s (= 200 MB/s), which is more than the nominal 160 MB/s of the S-Link specification. The available memory bandwidth is in the order of 450MB/s per channel, thus significantly higher than the worst-case of 200 MB/s IN plus 200 MB/s OUT.

<sup>12</sup> The Ethernet header is not 32 bit aligned, but has either 14 or 18 bytes, depending on the existence of the VLAN field.



**Figure 6.** PCI fragment request.

The CPU can perform a “garbage-collection” operation, which removes all left-over fragments outside a validity range. The range is identified by the host PC, when it requests the garbage-collection.

Statistics and monitoring information is available to the host PC via the message interface.

After power-up the processor loads the ROM monitor from the FLASH memory. At the end of the initialisation sequence it checks the FLASH environment variables, which should contain a command to load the ROBIN application from the FLASH. The ROBIN application then configures the FPGA with the bit-stream from FLASH and performs self-test and final initialisation. This procedure is sufficiently quick to bring the ROBIN into full operation while the host PC is still busy loading the operating system.

## 6. ROS software

The software running on the ROS PCs is based on the IOManager (IOM) framework, also used for the ROD Crate DAQ (RCD) [26]. It consists of a multi-threaded C++ program using Linux POSIX threads. The configuration and control of the ROS software is similar to that of the ROD Crate DAQ. Here we concentrate on the threads involved in the data flow for which the thread structure is sketched in figure 7.

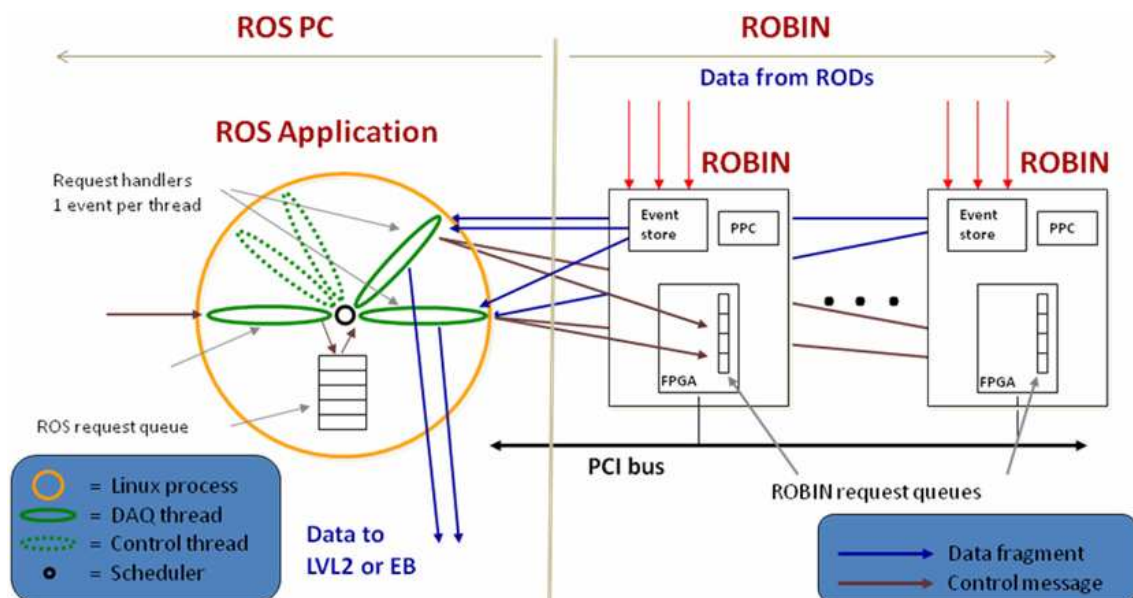


Figure 7. ROBIN - ROS PC interaction.

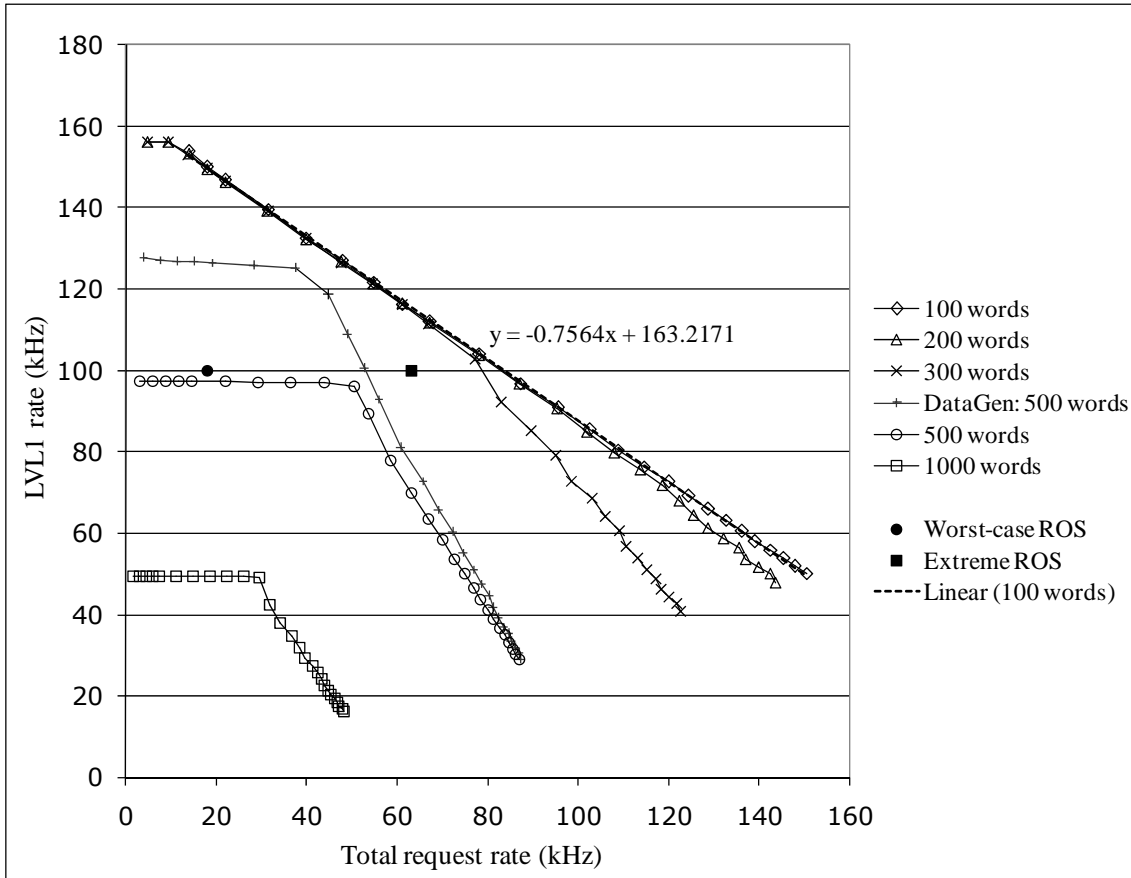
The receipt of a message from one of the level-2 trigger processors or Event Building nodes activates the trigger thread (it may also be activated by an internal “event” in case of emulation of the reception of messages). The trigger thread creates a request object, specifying the action to be executed according to the type and contents of the message, and posts it to a queue. The action can consist of requesting the ROBINS to delete event fragments or of the retrieval of one or more event data fragments from the ROBINS to be sent back to the requesting node. The request objects are retrieved from the queue by request handler threads. Each handler handles one request at a time, but different handlers can work in parallel in order to achieve better CPU utilization. The number of threads is configurable. In case of data requests, the request handler thread builds a larger fragment from the event fragments received from the ROBINS (as described in the previous section, the ROBINS transfer the requested data from their buffer memories to the memory of the PC) and output it to the destination specified in the request object.

## 7. Performance results

There are different views on the performance of the ROBIN. The very basic parameters are the processing times of the two time-critical tasks, fragment handling and message processing. The latter is quite different for PCI-based and network-based messages. Bandwidth limitations come into play when the ROBIN operates at high request rates. For a typical baseline ROS the system performance is largely dominated by the PC itself.

Stand-alone performance measurements have been obtained for two scenarios, PCI-based and network based. The initial network-based setup for the PRR has been done using a custom protocol on top of raw Ethernet packets. Later-on the custom protocol was replaced by standard UDP.

The typical stand-alone PCI setup is derived from figure 2 by driving the S-Link inputs from a separate PC, which generates event fragments of programmable size at the maximum



**Figure 8.** Single ROBIN PCI performance.

speed<sup>13</sup> of the physical link (2 Gb/s). A test program in the host PC requests fragments from the ROBINS at a variable ratio. Figure 8 displays the performance of a ROBIN with 3 active channels for different fragment sizes and request rates.<sup>14</sup> For fragment sizes of 1600 bytes and larger the L1-rate is limited by the S-Link bandwidth<sup>15</sup> for low request ratios and by the PCI bandwidth for higher request ratios. For smaller fragment sizes (up to about 1kB) there is a linear region with a constant slope, defined by the performance characteristics of the ROBIN.

For 100 kHz L1-rate and 1 kB fragments the maximum total request rate is around 80 kHz, which equates to approximately 27 kHz per channel. The ROBIN performance in this area is characterised by 2 basic parameters: the time  $tf$  required to process the incoming fragments (including the time to delete them) and the time  $tr$  to respond to data requests. In the recent test setup the model parameters<sup>16</sup> were measured as  $tr \sim 4.6 \mu\text{s}$  and  $tf \sim 6.1 \mu\text{s}$ . The data point marked as “Worst-case ROS” corresponds to an RoI-size of 2.7 ROLs at design luminosity. The

<sup>13</sup> Hence the rate can go above the nominal 100 kHz, for fragments up to 2 kB.

<sup>14</sup> Note: the X-axis shows the combined rate of request for all 3 channels. Every channel has one third of this rate.

<sup>15</sup> These measurements were done using a remote data source and an optical S-Link with a bandwidth limit of 200MB/s. One item was measured using the internal data generator with a bandwidth limit of 264 MB/s.

<sup>16</sup> Parameter  $tf$  is calculated for 3 ROLs, while  $tr$  is calculated per ROL.

data request rate<sup>17</sup> per ROBIN channel in this case is around 6 kHz for a typical fragment size of 1 kB. The data point marked as „Extreme ROS” corresponds to exotic trigger scenarios like an inner-detector scan at 18 kHz [27]. The performance of a typical baseline ROS system with 4 ROBINS is somewhat above the worst-case point. However for an extreme scenario, the number of ROBINS per PC or the number of ROLs per ROBIN has to be reduced.

The typical stand-alone network setup is again derived from figure 2, by adding the same data source on the S-Links plus 2 PCs<sup>18</sup> attached to the network switch. All data and delete requests are then sent via the switch to the ROBIN network interface. Using the custom network protocol the parameters were found<sup>[9]</sup> to be  $tr \sim 9.3 \mu\text{s}$  and  $tf = 5.24 \mu\text{s}$ , which leads to approximately 17 kHz request rate per channel at 100 kHz L1-rate. The increased request processing time is explained by the additional overhead required to decode and encode the network messages.

## 8. Conclusion

The ROBIN has been successfully developed, built and installed into the ATLAS TDAQ system and is heavily used in the current commissioning of the ATLAS detector. For the baseline PCI-based readout architecture the performance requirements are easily met. Concerning the network-based readout a realistic test-setup has not yet been established. However, the available results indicate that the ROBIN performance is at least close to the worst-case requirements.

## References

- [1] The ATLAS collaboration, *ATLAS Detector and physics performance technical design report*, CERN, Technical report CERN-LHCC 99-13, May 1999, [http://www.cern.ch/Atlas/GROUPS/PHYSICS/TDR/physics\\_tdr/printout/Volume\\_I.pdf](http://www.cern.ch/Atlas/GROUPS/PHYSICS/TDR/physics_tdr/printout/Volume_I.pdf).
- [2] The ATLAS collaboration, *ATLAS High-Level Trigger, Data-Acquisition and Controls Technical Design Report*, ATLAS TDR 016, Technical report CERN-LHCC 2003-022, June 2003, <http://atlas-proj-hltdaqdcs-tdr.web.cern.ch/atlas-proj-hltdaqdcs-tdr/tdr-v1-r4/PDF/TDR-2up.pdf>.
- [3] H.P. Beck et al., *The Base-Line DataFlow System of the ATLAS Trigger and DAQ*, *IEEE Trans. Nucl. Sci.* **51** (2004) 470.
- [4] J. Vermeulen et al., *ATLAS DataFlow: the Read-Out Subsystem, Results from Trigger and Data-Acquisition System Testbed Studies and from Modelling*, *IEEE Trans. Nucl. Sci.* **53** (2006) 912.
- [5] A. Kugel et al., *ROBIN Design Document*, Technical Report, CERN, Mar. 2004, <https://edms.cern.ch/file/473396/1.0/rdd.pdf>.
- [6] A. Kugel et al., *The final design of the ATLAS Trigger/DAQ Readout-Buffer-Input (ROBIN) device*, ATL-DAQ-CONF-2005-009, 10<sup>th</sup> Workshop on Electronics for LHC and Future Experiments, Boston, MA, USA, 13 - 17 Sep 2004, pp.258-263.
- [7] EDMS online documentation at <https://edms.cern.ch/project/ATL-0000007976/0>.

<sup>17</sup> The request rate is the sum of RoI-rate per ROL and the EB rate, e.g. at 100 kHz L1-rate: 100 kHz \* 2.7% = 2.7 kHz RoI rate, 100 kHz \* 3% = 3 kHz EB rate.  $\Sigma = 5.7 \text{ kHz}$

<sup>18</sup> Two PCs are required because a single PC cannot necessarily reach the ROBIN performance limitation. A simple load distribution can be done by using one PC for odd and even events.

- [8] A. Kugel et al., *ATLAS ROBIN User Manual*, CERN, Technical report, <https://edms.cern.ch/file/719553/1/robinUserManual.pdf>.
- [9] M. Müller, *ROS Architectures and Requirements*, Presentation at *ROBIN FDR*, CERN, May 2004, <http://indico.cern.ch/getFile.py/access?contribId=s1t0&resId=1&materialId=0&confId=a041681>.
- [10] L. Tremblet, *ROS performance presentation*, Presentation at *ATLAS TDAQ workshop*, Oct 2005, <http://indico.cern.ch/materialDisplay.py?subContId=1&contribId=s5t6&sessionId=s5&materialId=0&confId=a051610>.
- [11] *The common ATLAS T/DAQ glossary*:  
<http://atlas.web.cern.ch/Atlas/GROUPS/DAQTRIG/glossary.html>.
- [12] R. Cranfield et al., *ROS Requirements Document*, CERN, Technical report, May 2004, [https://edms.cern.ch/file/356336/1.0.1/ROS\\_URD\\_1.0.doc](https://edms.cern.ch/file/356336/1.0.1/ROS_URD_1.0.doc).
- [13] D. Francis et al., *The Read-Out Buffer in DAQ/EF Prototype -1*, ATL-DAQ-2000-053, CERN, Aug. 2000, <http://doc.cern.ch/archive/electronic/cern/others/atlnot/Note/daq/daq-2000-053.pdf>.
- [14] M. Müller, *Evaluation of an FPGA and PCI Bus based Readout Buffer for the Atlas Experiment*, Ph.D. thesis, Universität Mannheim, Aug. 2004, [http://madoc.bib.uni-mannheim.de/madoc/volltexte/2005/1070/pdf/Dissertation\\_Matthias\\_Mueller.pdf](http://madoc.bib.uni-mannheim.de/madoc/volltexte/2005/1070/pdf/Dissertation_Matthias_Mueller.pdf).
- [15] P. Farthouat, *Final design review of the ROBIN prototype*, CERN, Technical report, Oct 2002, <https://edms.cern.ch/file/359014/1/robin-pdr.pdf>.
- [16] P. Farthouat, *Final design review of the ATLAS read-out input buffer*, CERN, Technical report, May 2004, <https://edms.cern.ch/file/478897/2/Robin-fdr-may2004.pdf>.
- [17] P. Farthouat, *PRR of the ROBIN*, CERN, Technical report, Mar 2005, <https://edms.cern.ch/file/571432/1/Robin-prr-mar2005.pdf>.
- [18] Xilinx Virtex-II documentation, [http://www.xilinx.com/products/silicon\\_solutions/fpgas/virtex/virtex\\_ii\\_platform\\_fpgas/resources/index.htm](http://www.xilinx.com/products/silicon_solutions/fpgas/virtex/virtex_ii_platform_fpgas/resources/index.htm).
- [19] AMCC PowerPC documentation, [https://www.amcc.com/MyAMCC/jsp/public/productDetail/product\\_detail.jsp?productID=PPC440GP](https://www.amcc.com/MyAMCC/jsp/public/productDetail/product_detail.jsp?productID=PPC440GP).
- [20] O. Boyle et al., *The S-LINK Interface Specification*, CERN, Mar 1997, <http://hsi.web.cern.ch/HSI/s-link/spec/spec/s-link.pdf>.
- [21] HOLA S-Link documentation, <https://edms.cern.ch/item/CERN-0000008086/0>.
- [22] Texas Instrument on-line documentation, <http://focus.ti.com/docs/prod/folders/print/tlk2501.html>.
- [23] PLX documentation, [http://plxtech.com/pdf/product\\_briefs/9656.pdf](http://plxtech.com/pdf/product_briefs/9656.pdf).
- [24] Marvell Gigabit Transceiver documentation, <http://www.marvell.com/products/transceivers/singleport/index.jsp>.
- [25] Xilinx CoolRunner-II CPLD documentation, [http://www.xilinx.com/products/silicon\\_solutions/cplds/coolrunner\\_series/coolrunner\\_ii\\_cplds/index.htm](http://www.xilinx.com/products/silicon_solutions/cplds/coolrunner_series/coolrunner_ii_cplds/index.htm).



- [26] S. Gameiro et al., *The ROD crate DAQ software framework of the ATLAS data acquisition system*, *IEEE Trans. Nucl. Sci.* **53** (2006) 907.
- [27] A. Kugel et al., *ATLAS TDAQ/DCS ROS ROBIN Production Readiness Review: Performance Measurements (PRRPM)*, Technical Report, CERN, Jan 2005, [https://edms.cern.ch/file/555103/1/prr\\_performance\\_050221.pdf](https://edms.cern.ch/file/555103/1/prr_performance_050221.pdf).

2008 JINST 3 T01002