

Performance Measurements of a microEnable RobIn

Abstract

This document describes the use of the FPGA coprocessor microEnable as Read Out Buffer (ROB) hardware for the trigger system of the Atlas detector. It shows with performance measurements the possibilities of FPGA hardware to buffer and manipulate data received from the detector very fast. It points out that FPGA hardware can be used successfully in the Atlas trigger environment.

1. Introduction

FPGA technology provides the possibility to transfer and manipulate data in a very fast manner. To make this technology available for the Read Out Buffer (ROB) of the Atlas detector the FPGA coprocessor card microEnable was used to build up a ROB Input Device (RobIn). To show the functionality of the microEnable card as RobIn, performance measurements were made. Chapter 2 of this document describes the setup of the performance measurements. Chapter 3 presents the measurements and Chapter 4 presents the implemented pre-processing algorithm for the TRT which can reduce and prepare the data sent to the LVL2 feature extraction processor (FEX).

2. Setup

To show the functionality and to measure the performance of the microEnable based RobIn a source and a data destination for event fragment data is required.

As data source, a microEnable card with a SLink interface was used. It emulates a ROD and transfers the data via an electrical SLink to the RobIn microEnable. The FPGA designs as well as the SLink was clocked with 20 MHz so the raw bandwidth of this connection was 80MB/s. The sender microEnable and the RobIn microEnable cards are plugged into a standard WindowsNT PC with a 266MHz PentiumII and 64 MB SDRAM. This PC acts as RobController and requests the buffered data by event ID. The requested data is transferred with a DMA on demand via PCI bus into the memory of the PC. The PC acts as data destination. This flow of data is shown in Figure 1.

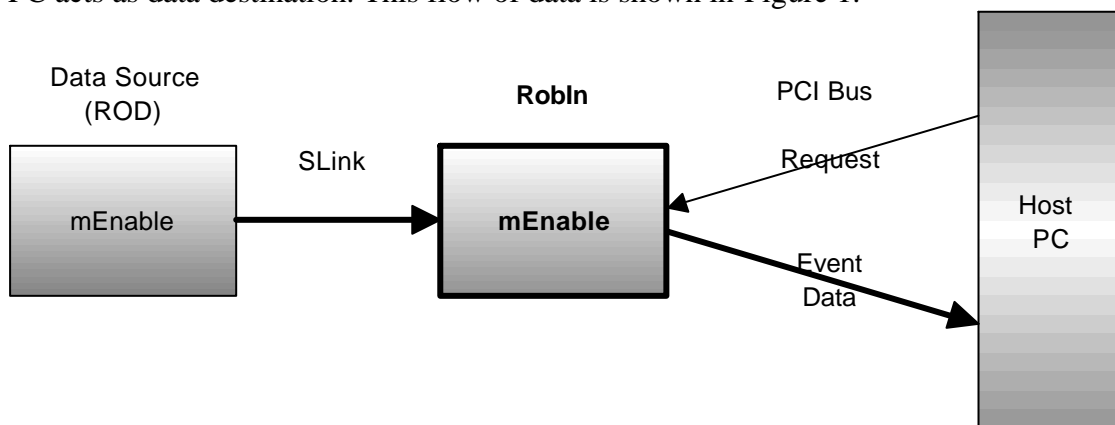


Figure 1

Setup description for microEnable based RobIn performance measurements

3. Measurement Results

With the setup, described in the previous chapter, the data-rates for the PCI output and the output fragment rates over PCI bus was measured. The host software requests the data via PCI write into a FPGA register. The FPGA design gets the requested data from the memory and transfers it via a DMA access to host memory. To get knowledge of the end of the DMA transfer, the host CPU has two possibilities. Either an interrupt is initiated from the microEnable PCI bridge or the CPU has to poll the microEnable card. Both variants were measured.

a) Output Data-Rates

The data-rates for PCI transfer for the fragment sizes 256, 512, 1k, 2k and 4k bytes are shown in figure 2 with and without interrupt after DMA access.

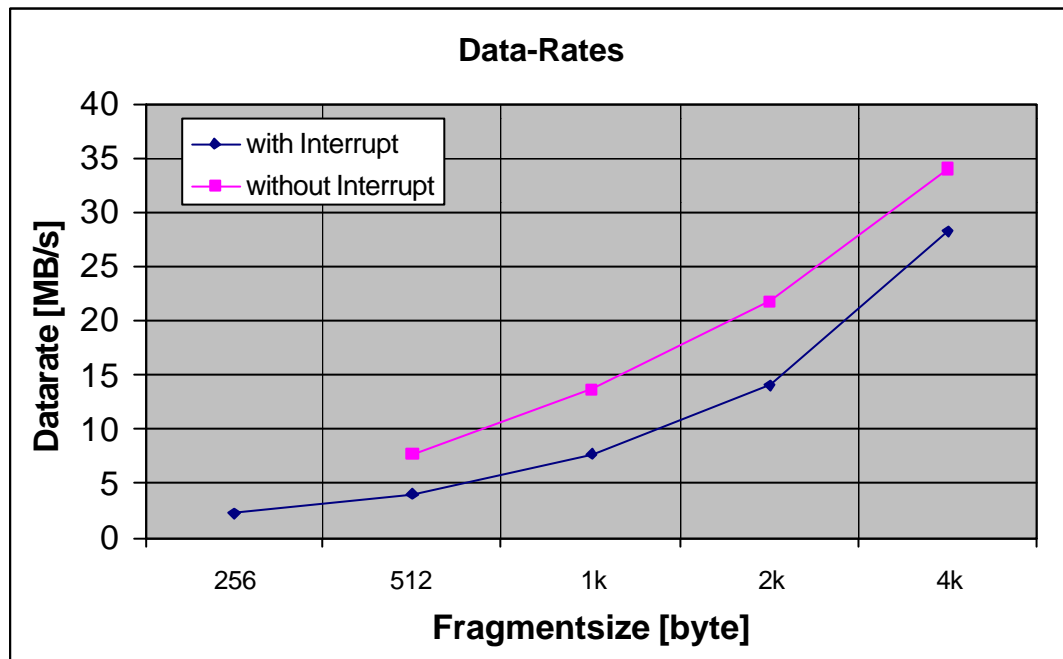
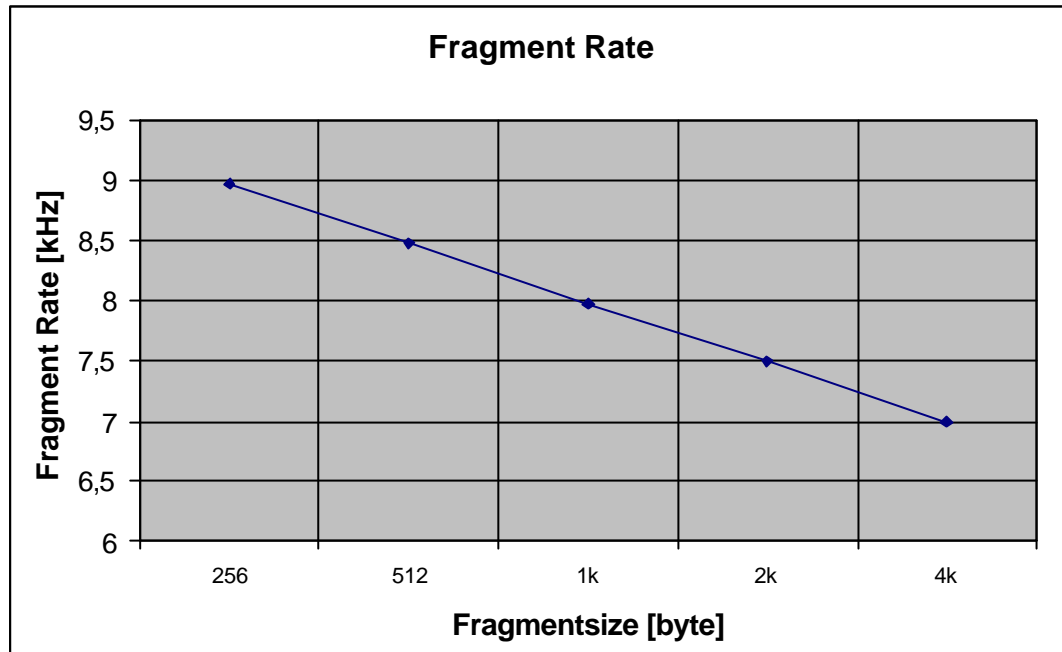


Figure 2

It can be seen that the data-rates are getting higher with increasing fragment size. This is because of the overhead of $105 \mu\text{s}$ for initialising every single DMA transfer. With smaller fragments more data fragments have to be transferred and with every fragment a overhead has to be added to the transfer time. This also explains the effect, that transfers ended with an interrupt have smaller data-rates. The interrupt leads to a context switch in WindowsNT and this context switch increases the transfer overhead to $146 \mu\text{s}$.

b) Output Fragment-Rates

Figure 4 shows the fragment rates for different fragment sizes. These can be calculated from the data-rates which have been shown above. This is the maximal rate to serve event fragment data to the LVL2 trigger and the Event filter.



Figur 3

Except for the inner detector in the low luminosity case, these fragment rates suffice the required fragment rates, computed in the paper modell of Jos Vermeulen (for the detailed rates, refer to the paper modell).

4. Pre-processing for the LVL2 TRT-trigger

To reduce the data volume to the LVL2 TRT trigger and to reduce the decision time of this trigger, a pre-processing algorithm was implemented in the microEnable FPGA. The idea of this algorithm is to transmit a list of hit straws, instead of the number of hits per every straw. This reduces the datavolume to the LVL2 trigger. The reduction is depending on the occupancy of the detector. With 30% straws hit the algorithm can reduce the data to 77% of the input. Transmitting a list of hit straws is also a more useable format for the Hough transform based LVL2 TRT trigger. So pre-processing leads to a reduction of the decision time because the LVL2 FEX algorithm has not to manipulate the incoming data. The chosen pre-processing format is shown in Figure 3. For a more detailed description refer to ATLAS DAQ Note 66 (1997).

2 bit t_0	3 bit first valid drift time	1 bit threshold	10 bit straw number
-------------	------------------------------	-----------------	---------------------

Figure 3

Preprocessing data format for the TRT

To measure the pre-processing time, realistic TRT event fragments have been send to the microEnable RobIn. The time to pre-process one event fragment was measured to 130 μ s

on average. This leads to a event fragment rate of 4 kHz with pre-processing. So with pre-processing the fragment rate is critical for several high occupied detectors (em calorimeter in high luminosity case and inner detector in low luminosity case).

5. Conclusions

The measurements show that a FPGA based RobIn can be done with satisfying data-rates. To increase the data-rates the Atlantis system was developed. It provides faster hardware as well as the possibility to group more than one input link to one RobIn board. This increases the fragment sizes transferred via the PCI bus and decreases the influence of the DMA overhead. Also a new DMA transfer mechanism, done by software, leads to a lower DMA overhead which increases the data-rates.

From very recent measurements (Nov 1999) at CERN the following preliminary results are available (1 physical RobIn, Pentium II, 400MHz):

Fragment size	Datarate
128 Byte	23 MB/s
1024 Byte	60 MB/s
4096 Byte	75 MB/s

These result show that the performance at 1kByte fragments could be improved by a factor of approx. 7.5. With smaller fragment sizes this factor increases, e.g. to 11.5 at 128 Byte fragments. At very large fragments the data rates are similar.

More results will be presented at the Beatenberg meeting in Dec. 1999