ATLANTIS - a modular, hybrid FPGA/CPU processor for the ATLAS Readout Systems

A. Kugel, Ch. Hinkelbein, R. Männer, M. Müller, H. Singpiel
University of Mannheim, B6, 26, 68131 Mannheim, Germany
{kugel, hinkelb, maenner, mmueller, singpiel}@ti.uni-mannheim.de

L. Levinson
Weizmann Institute of Science, Rehovot, Isreal
Lorne.Levinson@weizmann.ac.il

Abstract

ATLANTIS realizes a hybrid architecture comprising an industry standard PC platform plus different FPGA based modules for high-performance I/O (AIB) and computing (ACB). It is a flexible and modular system which can be the platform for several applications. CompactPCI provides the basic communication mechanism, enhanced by a private bus. The system can be tailored to a wide range of applications by selecting an appropriate combination of modules. Acceleration of computing intensive Atlas level-2-trigger tasks has been demonstrated with an ACB based system. The Atlas RoD and RoB systems will profit from the flexible and highly efficient AIB I/O architecture. Various high-speed interface modules (e.g., S-Link / M-Link) are supported, allowing up to 28 links per CompactPCI crate.

I Introduction

The goal of ATLANTIS is to provide a general purpose CPU plus FPGA co-processor system on which several applications can be run.

High performance I/O applications, as appearing in the Atlas detector trigger/DAQ, need beside a fast interface technology a huge computing capacity. In many applications a communication protocol must be implemented or data must be stored or processed with very low latency and high performance. FPGA technology is a very good candidate to solve high performance I/O applications. Communication protocols can be implemented and run at very high data rates. The re-configurability gives flexibility for future changes in the protocol. This use of FPGAs is already done in the SLink technology [1]. But FPGAs can also be used to manipulate or process data very quickly as shown in the LVL2 trigger [2]. Combining both, an FPGA can handle the interface protocol as well as process data "on the fly".

The ATLANTIS System described in this paper gives a flexible solution using FPGA technology for I/O applications. Two applications in the area of the Atlas project are presented to describe the use of ATLANTIS.

II The ATLANTIS System

The ATLANTIS System is the third generation FPGA processor build at the University of Mannheim. Its basic concept is to use FPGA technology beside conventional CPU hardware with both parts communicating via a CompactPCI bus system [3]. Thus FPGA technology becomes available in a normal PC environment easily. Figure 1

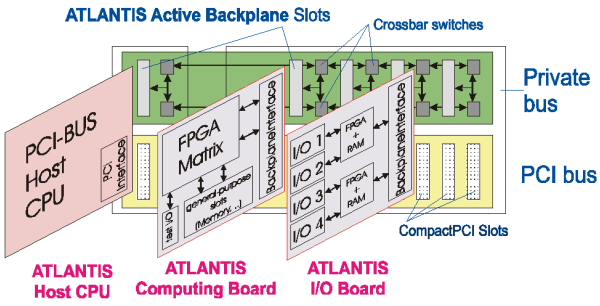

Figure 1: The ATLANTIS System

shows the structure of ATLANTIS. The Host CPU part is an embedded CPU module, based on a standard Intel processor. Currently one Pentium 200MMX and one PentiumII 400 are in use. Both modules are used inside a CompactPCI crate. The CompactPCI bus provides the communication between the CPU and the FPGA hardware. Up to seven boards per bus are possible.

Two types of FPGA boards are part of the architecture. One board, the ATLANTIS Computing Board (ACB), is used mainly for computing applications, e.g. the Atlas LVL2 low luminosity TRT trigger [2]. The second board, the ATLANTIS I/O board, was designed for high performance I/O applications with variable interfaces and is in the PCB routing process. Its technical design is described in the next chapter.

The complete system can be used either with WindowsNT or Linux. A special software library was written in C++ to access the FPGA hardware, by using a commercial low level PCI driver.

Programming the FPGA designs can be done with VHDL based tools or with CHDL, a C++ like class library for FPGA programming developed at the University of Mannheim [4].

III The ATLANTIS I/O Board

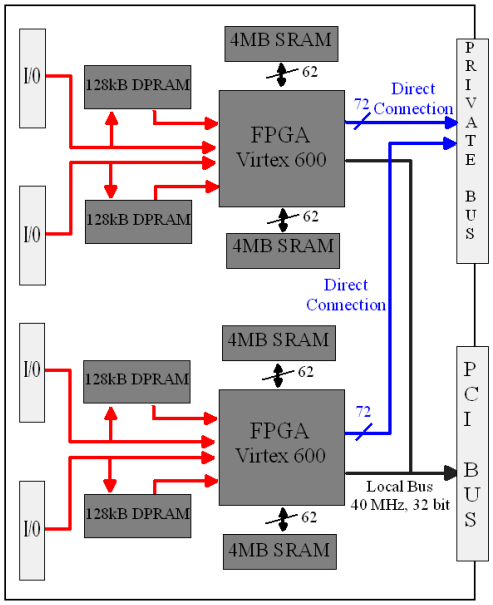

Figure 2: The ATLANTIS I/O Board (AIB)

In order to be compatible with the existing ATLANTIS Computing Board the I/O board was designed using the same basic structure and PCI interface hardware. Figure 2 gives an overview of the board. The main components, PCI bridge, FPGAs, memory, I/O channels, private bus system and the clock and control system, are described below.

The connection to the 32 bit 33 MHz CompactPCI bus is done by a PLX9080 PCI bridge chip. Two Xilinx XCV600 Virtex FPGAs, each with 512 I/O pins, are available to perform interface protocols, buffer management and data processing. To store data, each FPGA can use two 4 MB on board banks of synchronous SRAM memory with an access time of 12 ns. A 36 bit data bus provides a maximum memory bandwidth of 375 MB/s. So the board has very fast access to a total of 16 MB of SRAM memory.

The connection to the "outside world" is done by four flexible I/O channels with a capacity of 36 bit at 80 MHz (360 MB/s). Two channels are dedicated to each FPGA. To insure that no data of the input channels is lost when the FPGA data processing design is busy, each channel is additionally equipped with 128 kB of true Dual Port Ram (DPRAM). With a maximal clock frequency of 80 MHz and a data bus width of 36–bits on each port this DPRAM can be read and written simultaneously with up to 360 MB/s. The I/O channel data paths are designed to use either DPRAM to buffer the data, or to transfer data directly to the FPGA.

To provide maximum flexibility, the I/O interface specific hardware is located on a separate daughterboard connected to the AIB by mezzanine IEEE1386 connectors or 64 pin Mini SUB-D connectors. Currently two different modules exist. The first one is a standard SLink interface board for electrical or optical SLink connections. Because of the dimensions [1] of these daughterboards at most two can be used on one AIB simultaneously.

To reach up to four links per AIB board a new variant of the SLink was developed at the Universities of Mannheim and Krakov: the MLink.

MLink is compatible to the electrical SLink adapter. It can be linked with a special cable converting SCSI-2 to the higher density SCSI-4 connector of MLink.

In order to fit a small (54×149mm) mezzanine daughterboard, the SLink protocol engine was transferred to the AIB FPGAs. So the board consists only of the LVDS transceiver electronics and the SCSI-4 connector. The maximal transfer rate of MLink is 160 MB/s with a 16–bit data bus and 80 MHz clock frequency.

In addition to the PCI interface a private bus system is available for a fast data transfer between ATLANTIS boards. This private bus system consists of 144 freely programmable lines directly connected to the FPGAs—72 for each FPGA. So with a clock frequency of 80 MHz data rates up to 1.44 GB/s are possible. Two kinds of backplanes can be used, the ATLANTIS Test Backplane (ATB) or the ATLANTIS Active Backplane (AAB). The ATB is a one–to–one connection between neighboring boards without any additional electronics. The AAB has programmable switch interfaces to set each point to point line connection individually.

Finally an programmable clock system based on several ICD2053B generates clocks up to 80 MHz and distributes them to the FPGAs and the I/O ports.

All these features of the AIB are well usable for several I/O applications in the Atlas environment. The following sections will introduce two applications each with a high I/O bandwidth and a high computing effort.

IV ReadOut Driver (ROD) for the ATLAS Muon Endcap Trigger Chambers

The ROD for ATLAS Muon Endcap Trigger Chambers must perform the following functions in re-

sponse to each LVL1 trigger Accept:

- build an event from fragments arriving asynchronously from the 14 input links that comprise one octant
- verify the integrity of the event format
- translate the encoding of hits into a list of hits
- map HW channel numbers to chamber z, R and ϕ bins
- collect statistics on data rates and occupancies
- re-do the trigger algorithm to verify the functioning of the trigger hardware
- sort the hits into an order optimized for LVL2 track finding
- send the hits in Atlas standard Read Out Buffer (ROB) format to the ROB

For the Muon Endcap trigger, the data volume is not high – the 14 inputs do not fill the bandwidth of the 100 MB/s output link to the ROB – but the message rate is high. The LVL1 Accept rate can be up to 75 kHz, and there are 14 event fragments for each LVL1 Accept. Coping with this message rate is very difficult for a conventional processor but not for an FPGA. Several requirements such as communication with the experiment and central trigger, managing the monitoring, and supervising the whole process is best done in a conventional processor. The problem is ideal for an FPGA co-processor.

The 14 input links – the so-called Front End links – and the output link to the ROB will be conform to the S-link standard. The prototype Muon Endcap trigger ROD consists of a 600 MHz Pentium processor running Linux and several AIBs. Each AIB has two S-link daughter boards. The Atlantis system will be used to determine the optimal division of the ROD tasks between software and FPGA. Atlas standard ROD crates are VME crates rather than cPCI crates. Experience with the Atlantis prototype ROD will enable the design of an FPGA IO processor for use with a VME-based Pentium CPU.

The ROD tasks can be expressed as a series of pipelined data processing stages. Because the input data is zero-suppressed, the ROD is obliged to handle data streams consisting of headers followed by a variable length list of items. The architecture of a generic processing stage in this pipeline is shown in Figure 3. The simple pipelines for processing fixed length strings or arrays of repeated elements, e.g. pixelated images, must be augmented to cope with more structured data. This gives rise to a more complex architecture. The separation into item and header/trailer paths allows easier implementation of the control logic. This representation of the ROD processes is independent of whether the processing stage is implemented in hardware or software.

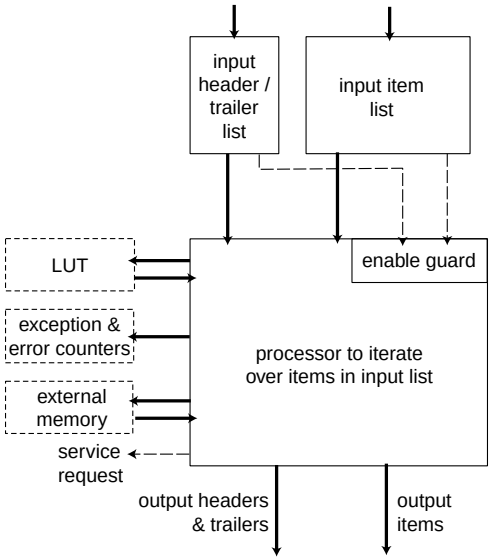

Figure 3: The generic stage for a pipeline that processes variable length data records. The data flow is from top to bottom. Processing begins when a header is available and items are processed until the trailer is received. All items must be sent only after their header and before their trailer. The header usually contains data common to all items; the trailer usually contains the item count and error flags.

Items may be data or pointers to data in external memory. External memory may also be used for accumulated data such as histograms. External memory and Look-up tables (LUTs) may be implemented as block ram in the FPGA or external SRAM.

An architecture for building events from fragments is implemented by sending headers for data stored in external memory on the *item* path. The output item is a control structure for a gather DMA.

V Atlas Readout Buffer (ROB) Complex Application

The Atlas Readout Buffer (ROB) Complex tasks are similar to that of the ROD application. After the Atlas detector event data has passed the LVL1 trigger decision it has to be stored. This is done in the Readout Buffer (ROB). To process the data, the LVL2 trigger requests the data from the ROB. If accepting the event the ROB keeps the data until it is fetched by the Event Filter. Otherwise the ROB deletes all data of the event.

The simplest approach to a ROB is one SLink input, one processing unit with memory, and one output to the network per detector ROD link. But for most Atlas sub-detectors the required bandwidth from one ROB to the LVL2 trigger and Event Filter is many times lower than the bandwidth of the

network link. Thus grouping of many input links to one network interface card (NIC) leads to a better utilization of the network link. The whole system consisting of several input links (ROBIns), a control engine and one NIC is called a ROB Complex.

The presented ATLANTIS System fulfills all requirements for a ROB Complex. AIBs can be used as flexible and powerful input hardware (ROBIn's) and, in contrast to other systems, four input ports are available per board. Messages and data inside the crate can be either passed over the CompactPCI or the private bus. The embedded CPU controls the complex and provides also the network interface. The ROB Complex can benefit in several aspects from the AIB features:

PCI utilization

Concentrating several links on one ROBIn board enlarges the data packages to be transferred over the PCI bus. Larger packages make a better utilization of the PCI bus. This is documented in the DMA performance measurements with the ATLANTIS System shown in Figure 4. With 1 kB

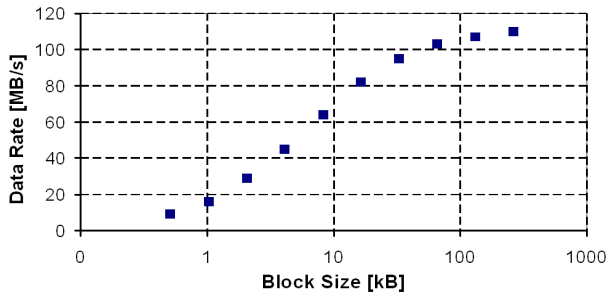

Figure 4: Datarates for DMA read from the ATLANTIS Board

packages a data rate of only 16 MB/s is possible. Increasing the package size to 2 kB or 4 kB improves the data rate to 29 MB/s and 45 MB/s.

But the concentration of links will not be utilized completely. Since the LVL2 trigger is guided by Regions of Interest (RoIs), it is generally not necessary to request the data of all links when the trigger processes an event. To estimate the average number of links, really requested on one AIB for different RoIs, a modified version of the RobsPerRoi program is used [5][6]. It loops over all possible detector RoIs and counts the number of ROBIns containing the hit data. The results are summarized in Table 1 for the ROB detector mapping used by the RobsPerRoI program[5]. All links on one AIB are neighbored in ϕ direction. So for a lot of subdetectors two or more links are requested for one RoI and merging the data of these links on one AIB leads to the desired increment of data size on the PCI bus.

In low luminosity detector mode, when B-Physics events are investigated, the LVL2 trigger algorithm

Table 1: Average number of input links requested per RoI on one AIB

	RoI		
	Muon	EM	Jet
Pixel	1.27	1.29	3.66
SCT	1.41	1.55	2.17
TRT	2.44	2.23	2.75
Hadr.Calo.	1.82	2.04	2.49
EM Calo.	1.36	1.56	1.90
Muon Trigger	1		
Muon MDT	1		

needs the full data of the TRT and Silicon Trackers to make a decision. Then the maximum advantage of concentrating links on one board is reached because the data of all links on the AIBs are required. This happens also in the case when the event is accepted by LVL2 and the Event Filter requests the complete event data.

Reduction of messages inside the ROB Complex

More links on one board also reduces the number of request messages on the PCI bus. To request 10 event fragments on five boards only five instead of 10 event fragment request messages are required.

Having fewer messages on the PCI bus provides more capacity for data transfer. In addition, using the private bus for distributing requests or other messages is a way to save PCI resources.

More computing power

To guarantee that no data coming from the I/O links is lost, and requested data is forwarded with low latency, a powerful algorithm on fast hardware is required. The AIB provides enough capacity and computing power inside its two FPGAs.

Two tasks must be done by the FPGAs: an input task and an output task supporting preprocessing "on the fly".

The Input Task stores data from every I/O port in the SRAM memory with data rates up to 137 MB/s. True DPRAM buffering in front of each channel reduces the need to pause the input link and helps to prevent data losses. Independent of the input, the output task has to process requests of event data. It has to find the required event fragments in the memory of each input channel and merge them before the transfer over PCI starts. Preprocessing as a part of the output process prepares data to be processed in the LVL2 trigger only. The algorithm differs for the various subdetectors [7], e.g. zero suppression, coordinate transformation or data format

conversion has to be done. One preprocessing algorithm for the TRT has already been shown in[8].

Reduction of total hardware and cost

The ATLANTIS based ROB Complex can carry up to 28 input ports per system crate. This is a chance to reduce the number of crates and the amount of hardware, especially network hardware, necessary for the Atlas trigger. It can also help to keep the costs of the total system moderate.

Table 2 shows the expected numbers of input links per crate for the different subdetectors for high luminosity operation, Table 3 for low luminosity.

Table 2: Maximum number of input links per crate dedicated to one 100 MBit or one 1 GBit output NIC at high luminosity. A LVL1 rate of 75 kHz is assumed.

	100 MBit	1 GBit
Pixel	7.6	76.3
SCT	3.9	38.9
TRT	6.1	61.5
Hadr.Calo.	3.5	34.6
EM Calo.	3.5	34.6
Muon Trigger	15.4	154
Muon MDT	7.6	76.3

Table 3: Maximum number of input links per crate dedicated to one 100 MBit or one 1 GBit output NIC at high luminosity. A LVL1 rate of 75 kHz is assumed.

	100 MBit	1 GBit
Pixel	9.3	92.6
SCT	3.9	39.2
TRT	1.5	14.9
Hadr.Calo.	3.3	32.7
EM Calo.	3.4	34
Muon Trigger	15.1	151.1
Muon MDT	7.5	74.8

Most subdetectors will require more than one 100 Mbit NICC or a 1 Gbit NIC to take advantage of the large number of input links possible in an ATLANTIS ROB Complex.

VI Conclusions and Outlook

ATLANTIS equipped with ATLANTIS I/O Boards provides a flexible and powerful platform for I/O tasks managing data rates up to 4×375MB/s. Two FPGAs are able to store the data "on the fly" and do preprocessing tasks like zero suppression, coordinate transformation, and data monitoring. This enables ATLANTIS to be used in the ATLAS Readout System.

Two ATLANTIS Systems equipped with ACBs have been available for one year and have proven their computing power in [2]. Two AIBs are expected in autumn 2000, the implementation of a ROB Complex and the ROD prototype will be available in Summer 2001.

References

[1] E-SLink Users Manual, http://www.ifj.edu.pl/~iwanski/ e-slink/uman.html

[2] Ch. Hinkelbein et al, LVL2 Full TRT Scan Feature Extraction Algorithm for B Physics Performed on the Hybrid FPGA/CPU Processor System ATLANTIS: Measurement Results, ATL-DAQ-2000-012, CERN, March 2000

[3] O. Brosch et al, ATLANTIS - A Hybrid FPGA/RISC Based Re-configurable System, RAW2000, Cancun, May 2000

[4] K. Kornmesser et al, "Simulating FPGA-Coprocessors Using the FPGA Development System CHDL", Proc. PACT Workshop on Reconf. Comp., Paris (1998) pp. 78-82

[5] J. C. Vermeulen, Computer modelling of the ATLAS LVL2 trigger system, ATL-COM-DAQ-2000-035, CERN, Mar 2000

[6] Jos Vermeulen, The Simdaq software package, ftp://ftp.nikhef.nl/pub/experiments/ atlas/tdaq/simdaq4/

[7] F. J. Wickens et al, Detector and Read-Out Specifications, and Buffer-RoI Relations, for Level-2 Studies, ATL-COM-DAQ-99-017, CERN, Oct 1999

[8] R. Rißmann, Implementierung von Vorverarbeitungsalgorithmen für der ATLAS Level 2 Trigger auf dem FPGA-Prozessor microEnable, Diploma thesis, Heidelberg, Feb 1999