ATCA-BASED COMPUTATION PLATFORM FOR DATA ACQUISITION AND TRIGGERING IN PARTICLE PHYSICS EXPERIMENTS

Ming Liu[1,3], *Johannes Lang*[1], *Shuo Yang*[1], *Tiago Perez*[1], *Wolfgang Kuehn*[1], *Hao Xu*[2], *Dapeng Jin*[2], *Qiang Wang*[2], *Lu Li*[2], *Zhen'An Liu*[2], *Zhonghai Lu*[3], *Axel Jantsch*[3]

1. II. Physics Institute, Justus-Liebig-University Giessen, Germany

2. Experimental Physics Center, Institute of High Energy Physics in Beijing, China

3. Dept. of Electronic, Computer and Software Systems, Royal Institute of Technology, Sweden

{ming.liu, tiago.perez, johannes.lang, shuo.yang, wolfgang.kuehn}@exp2.physik.uni-giessen.de

{xuhao, jindp, qwang, lilu, liuza}@mail.ihep.ac.cn {zhonghai, axel}@kth.se

ABSTRACT

An ATCA-based computation platform for data acquisition and trigger applications in nuclear and particle physics experiments has been developed. Each Compute Node (CN) which appears as a Field Replaceable Unit (FRU) in an ATCA shelf, features 5 Xilinx Virtex-4 FX60 FPGAs and up to 10 GBytes DDR2 memory. Connectivity is provided with 8 optical links and 5 Gigabit Ethernet ports, which are mounted on each board to receive data from detectors and forward results to outer shelves or PC farms with attached mass storage. Fast point-to-point on-board interconnections between FPGAs as well as the full-mesh shelf backplane provide flexibility and high bandwidth to partition algorithms and correlate results among them. The system represents a highly reconfigurable and scalable solution for multiple applications.

1. INTRODUCTION

In nuclear and particle physics, an "event" describes the result of a single reaction between a projectile particle and a target particle. Typically an "event" consists of "sub-events" referring to the activities of different detectors recording reaction products. Modern nuclear and particle physics experiments, for example HADES [1] and PANDA [2] at GSI, BESIII [3] at IHEP, are expected to run at a very high reaction rate (e.g. PANDA, 10-20 MHz) and able to deliver a data rate of up to hundred GBytes/s (PANDA, up to 200 GBytes/s). Among the huge amounts of events, only a rare proportion is of interest due to its particular physics contents and should be selected for in-depth physics analysis. Besides, all the events cannot be entirely stored because of the storage limitation. Therefore it is essential to realize an efficient on-line data acquisition and trigger system to process and filter events. As a result, the data rate to be stored can be reduced by several orders of magnitude. Depending on the physics focus of the experiment, sophisticated real-time feature extraction algorithms such as *Cherenkov ring recognition*, *particle track reconstruction*, *Time-Of-Flight (TOF) analysis*, *Shower recognition* [4] [5] [6] and *high level correlations* are implemented for recognizing the interesting data. Only the events which meet expected patterns and correlations receive a positive decision and will be forwarded to the mass storage for later off-line analysis. Others are discarded on the fly.

FPGA-based solutions have important advantages to realize the feature extraction algorithms. Typically pattern recognition and event selection are implemented as specific-purpose processors running in parallel and/or pipelined mode for high processing speed. The reconfigurability gives the possibility to change the algorithms on different experiment requirements. As the development of FPGA technologies, many new features related to computation and communication have been integrated in modern FPGAs. For instance the Xilinx Virtex-4 FX series features hardcore PowerPC embedded CPUs as well as Multi-Gigabit Transceivers (MGT) and Gigabit Ethernet MACs. Such features are convenient in physics experiments to (i) realize slow control tasks in software; (ii) accept huge amounts of raw data from detector circuits via high-speed links; (iii) easily connect with the commodity PC farm by widely-used Ethernet and standard network protocols.

The Advanced Telecommunications Computing Architecture (ATCA) fabric interface [7] was architected to provide the bandwidth needed for the next generation computation platform. A full-mesh shelf can support 2.1 Tbps of data transport when using 3.125 GHz signaling and 8B/10B encoding. Compared to the VMEbus which was conventionally used in data acquisition systems [8] [9] [10] [11], the ATCA standard offers advantages especially with respect to communication bandwidth and shelf management. To meet the high computation and communication require-

978-1-4244-1961-6/08/$25.00 ©2008 IEEE.

ments from modern experiments, our platform is based on the ATCA standard. In section 2, previous related work will be addressed and in section 3 we will demonstrate the hierarchical platform architecture, specifically from the top computation network down to the node. Furthermore, we focus on the hardware/software co-design of the system on FPGA in section 4. In section 5, implementation results and performance measurements are reported. Finally we conclude the paper and discuss the future work in section 6.

2. RELATED WORK

Reconfigurable computing is an modern computing paradigm which satisfies the simultaneous demands of application performance and flexibility. Although nowadays cluster-based supercomputers still dominate the fields of super computation tasks, reconfigurable computing begins showing large potential and perspective on some performance-critical areas such as real-time scientific computing. Currently many commercial and academic projects are developing hardware and software systems to employ the raw computational power of FPGAs. Most of them, however, are augmented computer clusters with FPGAs attached to the system bus as accelerators. One major drawback of these systems is the bandwidth bottleneck between the microprocessor and the FPGA accelerator. Other stand-alone platforms for example the Dini Group products [12], target mainly hardware emulation applications. It is not straightforward to upgrade the system to a supercomputer equivalent scale due to the lack of an efficient communication standard for inter-board connections. The Berkeley Emulation Engine 2 (BEE2) did provide a good platform which is powerful and scalable for large scale data processing [13]. Unfortunately the external links use only Infiniband, which results in an inflexible and expensive interface to detector front-end circuits and PC clusters in physics experiments. Moreover compared to the ATCA point-to-point direct links, its all-board-switched or tree-like topology may cause communication latency and throughput penalty when large algorithms are partitioned and span over multiple boards.

3. PLATFORM ARCHITECTURE

3.1. Computation Network Architecture

To manage the data rate of up to hundred GBytes/s, all feature extraction algorithms will be partitioned and run in parallel in a network architecture. Shown in figure 1, Compute Nodes (CN) are interconnected with each other and provide also external channels. Via the channel bonding of external connections including optical links and Gigabit Ethernet, data streams are received from detectors and processing results are forwarded to the PC farm for storage and high level analysis. The internal high-speed connections are employed to partition the algorithms and correlate results. We utilize the ATCA full-mesh backplane to provide flexible communications among all compute nodes in the chassis.

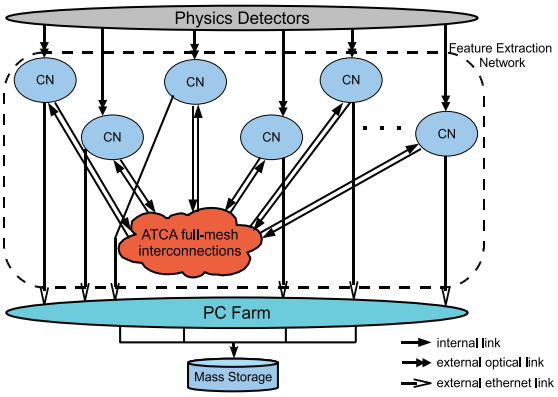

Fig. 1. Computation network for on-line feature extraction

Figure 2 is an ATCA shelf with the full-mesh backplane. The point-to-point direct interconnections offer much flexibility for various network configurations, such as the vertical pipelined processing, horizontal parallel processing, or intermedia solutions with more complicated connections vertically and horizontally. This feature provides significant freedom and convenience to partition algorithms and distribute sub-tasks in multiple compute nodes.

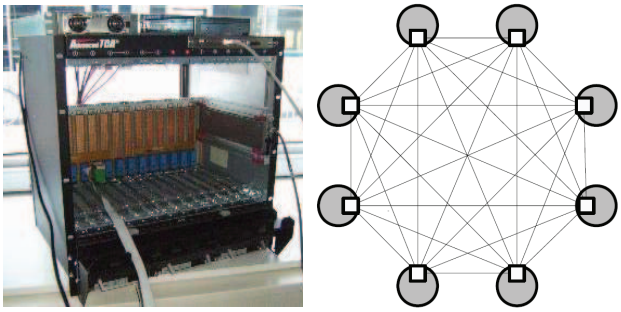

Fig. 2. ATCA shelf and full-mesh backplane

3.2. Compute Node

Figure 3 shows the schematic of a CN board for prototype design. On each board there are five Xilinx Virtex-4 FX60 FPGAs, four of which (No. 1 to 4) work as algorithm processors and the fifth (No. 0) as a switch interfacing to other CNs via the full-mesh backplane. Each processor FPGA has two optical links and one Gigabit Ethernets. All five FPGAs are equipped with 2 GBytes local DDR2 memory for data buffering and large look-up table purposes.

On-board point-to-point I/O interconnections make it easy to partition complicated algorithm implementations, which are too large to be fitted on one single FPGA chip.

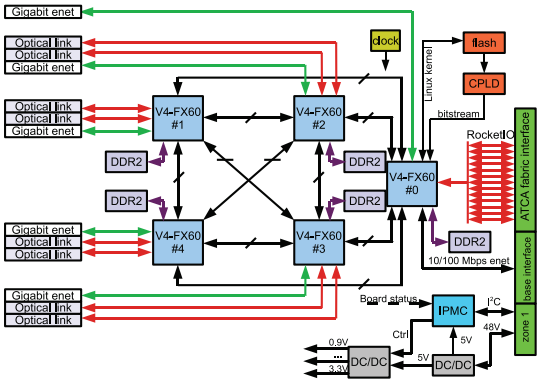

Fig. 3. Compute node design

A customized Intelligent Platform Management Controller (IPMC) fulfills the ATCA requirements on power negotiation, voltage monitoring, temperature sensoring, and FPGA configuration check, etc.. It talks to the shelf manager via two I^2C buses. The design is based on the AVR micro-controller and appears as an add-on card on the compute node.

Figure 4 shows our first prototype PCB. All main components are placed on the top side except the DDR2 SDRAM memories. Five ultra low profile Small Outline Dual In-line Memory Modules (SO-DIMM), which are compatible with the normal DDR2 memory in laptops, are arranged on the back side. Currently the board is under test and will soon be ready for the system implementation and measurements.

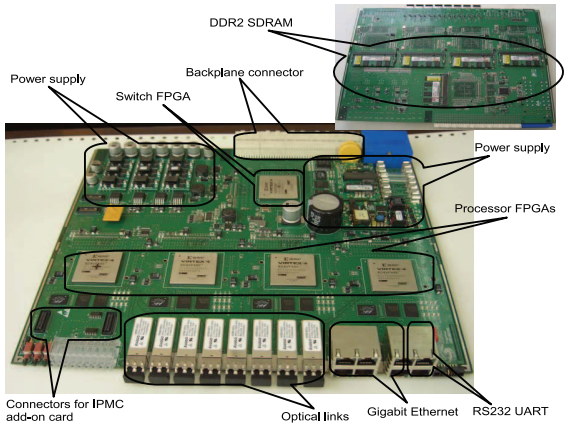

Fig. 4. Prototype PCB of compute node

4. FPGA NODE DEVELOPMENT

4.1. Hardware Design

The Xilinx Virtex-4 platform FPGA has many kinds of hardcore components, such as the PowerPC 405 embedded CPU, RocketIO Multi-Gigabit Transceivers, and Gigabit Ethernet MACs. Accompanied with softcore libraries, a bus-based system can be built and integrated on a single FPGA (see figure 5). Feature extraction processors, for instance *ring recognizer*, *shower recognizer* or *tracking processing unit*, are connected to the system Peripheral Local Bus (PLB). They utilize the hardware parallel or pipelined architecture to speedup the data processing over software. The embedded PowerPC takes charge of network protocol processing, as well as slow control tasks. Hence the CPU/FPGA solution is implemented on a single chip, with which the density of computing power as well as the flexibility of reconfiguring components and interconnections are increased significantly.

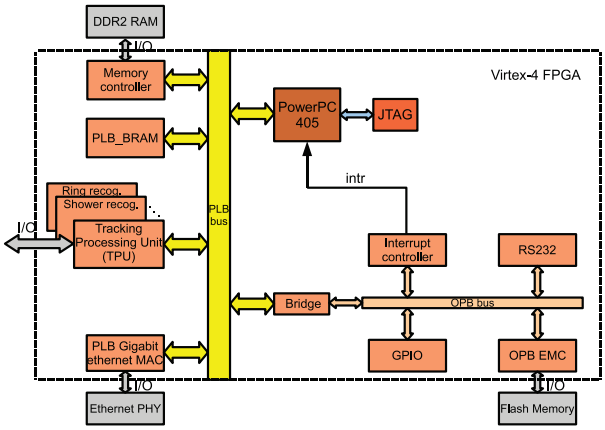

Fig. 5. Bus-based design with feature extraction processors

4.2. Software Design

An open-source Linux 2.6 kernel was ported to run on the PowerPC system. The soft Linux TCP/IP stack (including UDP transfers) drives the Ethernet communications with the PC farm. Some device drivers including the Gigabit Ethernet, RS232 and Memory Technology Devices (MTD), have already been included in the kernel's package as options. Others like the feature extraction processors are to be customized for accessing them in the Linux operating system. Ethernet data transferring and slow control tasks are implemented as application programs which might be written in C/C++ or high level scripts.

4.3. Feature Extraction Processors

For different focuses of the physics experiments, corresponding feature extraction algorithms are needed to be installed on the computation platform. Driven by our application projects of HADES and BESIII upgrade and PANDA construction, we are currently designing the specific-purpose processors for particle track reconstruction, Cherenkov ring recognition, shower recognition, event building and event selection. They can share an identical architecture by which they are connected to the system bus. Shown in figure 6, the customized algorithm processor is connected to the PLB bus via an IP interface (IPIF). The IPIF offers many optional features for the custom processor to exchange data with the system memory and interact with the CPU. The supported features mainly include: DMA transfer, interrupt, burst transfer, software accessible register, write FIFO and read FIFO, etc.. On the other side, the direct I/O access allows direct processing of data streams imported via external links. It is also possible to partition complex algorithms and distribute in multiple FPGAs which are mutually interconnected by I/O pins.

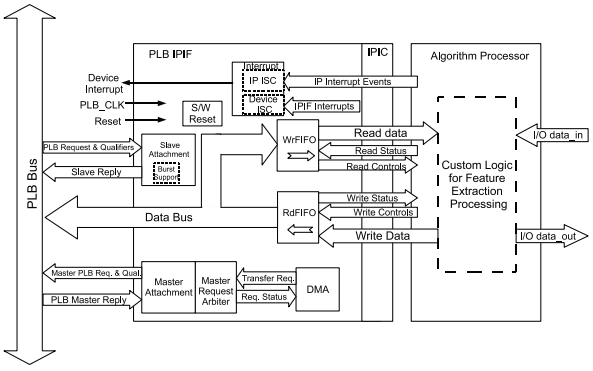

Fig. 6. Algorithm processor connected to the bus via IPIF

To study this general architecture, we designed and implemented an *event selector* to reject uninteresting events and accept good ones. Currently the selection rule is based on checking the event structure. In the future more sophisticated criteria may be added without changing the fundamental framework. To release the CPU from moving data back and forth between the memory and the IP, DMA and interrupt are enabled in the IPIF. DMA takes care of feeding event data to the Write FIFO (WrFIFO) and collecting results from the Read FIFO (RdFIFO). After each DMA transfer, the CPU is noticed by interrupt. In the *event selector* IP core, a buffer is dedicated to temporarily store an entire event. A complex FSM controls to analyze the event structure. If the event is interesting, it will be forwarded to the RdFIFO for collection. Otherwise it is rejected on the fly.

The *Tracking Processing Unit* (*TPU*) [14] is the processor which recognizes and reconstructs particle flying tracks in Mini Drift Chamber (MDC) detectors. The tracking work is most CPU-intensive and we expect high speedup when porting the software solution into the FPGA fabric. Based on the principle of Dubna track reconstruction [15] [16], we implemented a *TPU* module for particle track reconstruction in inner MDCs (MDC I - II). Like the *event selector*, DMA and interrupt are enabled in the IPIF. However neither read nor write buffering is necessary, since the *TPU* core has the capability to digest the memory data in real-time.

4.4. Remote Reconfiguration

Remote reconfigurability is desired due to the spatial constraint in experiments. In our design, XOR flash memory is the non-volatile storage where both FPGA bitstreams and Linux kernels reside. To upgrade the system, what we need to do is to overwrite the old bitstream and kernel image files with the new versions and then restart the system. The MTD driver in the Linux package provides the application interface to read and write the flash memory in Linux. With the support of the Ethernet network and the Network File System (NFS) [17], operators can remotely login the embedded Linux from a desktop PC and issue the upgrade and restart commands. To avoid fatal errors during the upgrading process, backups are arranged in the flash memory (shown in figure 7). In case of upgrading errors, Linux can be booted from the backup bitstream and kernel image to resume the upgrade process. Switching between the normal booting and the backup booting is controlled by the IPMC on the compute node.

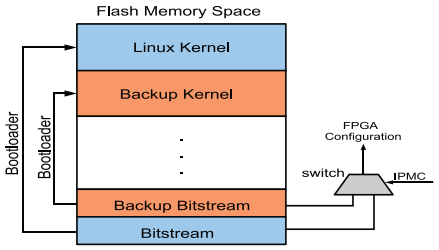

Fig. 7. Backup configuration in the flash memory

5. EXPERIMENTAL RESULTS

5.1. Implementation Results

The bus-based System-on-an-FPGA design is constructed by integrating various IP cores in Xilinx XPS 8.2 tool. The custom processors are described in VHDL. We use Modelsim 6.1e and Xilinx Bus Functional Models (BFM) [18] to simulate the bus behaviors. Xilinx ISE 8.2 is used to synthesize and implement the design. Table 1 shows the resource

Resources	FPGA node platform (no custom processor)	*TPU* + PLB-IPIF (no WrFIFO, RdFIFO in PLB-IPIF)	*event selector* + PLB-IPIF (4kB WrFIFO, RdFIFO and Event Buffer)	system with *TPU*	system with *event selector*
4-input LUTs	8531 out of 50560 (16.9%)	8075 out of 50560 (16.0%)	4674 out of 50560 (9.2%)	16606 out of 50560 (32.8%)	13205 out of 50560 (26.1%)
Slice Flip-Flops	5724 out of 50560 (11.3%)	3355 out of 50560 (6.6%)	2830 out of 50560 (5.6%)	9079 out of 50560 (18.0%)	8554 out of 50560 (16.9%)
Block RAMs	18 out of 232 (7.8%)	41 out of 232 (17.7%)	6 out of 232 (2.6%)	59 out of 232 (25.4%)	24 out of 232 (10.3%)
DSP Slices	8 out of 128 (6.3%)	0	0	8 out of 128 (6.3%)	8 out of 128 (6.3%)

Table 1. Resource utilization

consumption statistics as well as the utilized percentages of the Xilinx Virtex-4 FX60 FPGA. We observe that the system with a *TPU* or an *event selector* connected utilizes 32.8% or 26.1% LUT resource and 18% or 16.9% Flip-Flops respectively. The Block RAM and DSP slice utilizations are also small. From the shown statistics, we conclude that it is feasible to integrate the entire system in a single Virtex-4 FX60 and there is still plentiful resource left to perform the application specific computation. Considering the two embedded PowerPCs and four Gigabit Ethernet MACs on-chip, we also do not exclude the possibility to integrate two identical bus systems on one chip, if the resources could be optimally utilized.

With global timing constraints, the synthesis timing summary shows that the PLB bus system can run at above 100 MHz and the PowerPC CPU at 300 MHz. The *TPU* and the *event selector* modules are easily able to run at above 100 MHz without any optimization. To match the bus speed, we fix their clock frequency as 100 MHz.

5.2. Power Consumption Evaluation

For our computation platform which contains not only multi-gigabit serial links but also high-speed data processing, the power consumption problem is not trivial and should be evaluated seriously. We pessimistically accumulated all the power consuming components on the compute node and arrived at the maximum consumption of 170 W per board. This requirement can be met by the ATCA specification, which supplies the power up to 200 W per slot.

5.3. Computing Throughput of the Event Selector

Our development work has been executed using the Xilinx commercial board ML405, whose heart is a Virtex-4 FX20 FPGA. Except less resources, FX20 has the same features as FX60 which is expected in our products.

Figure 8 shows the experimental setup for measuring the computing throughput of the *event selector* core. We reserve two large blocks in the DDR memory, Mem0 for buffering the incoming events and Mem1 for collecting the interesting ones. DMA0 feeds events from Mem0 into WrFIFO and DMA1 gathers results from RdFIFO to Mem1. With continuous event stream supplied from Mem0, the computing

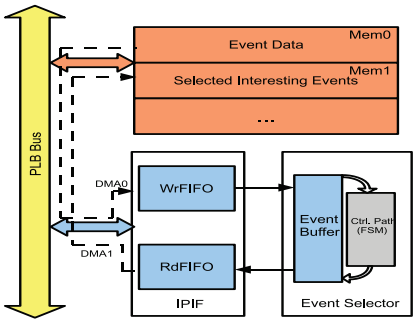

Fig. 8. Computing throughput measurements

throughput of the *event selector* is calculated by the division of data size and spent time. After the measurements and calculations at two different event selection rates (100% and 25% interesting event rates respectively), the processing capability is shown in figure 9 as functions of the DMA transfer size, which is equal to both WrFIFO and RdFIFO size for minimizing transfer overhead. In the figure, we see a lower event selection rate corresponds to a higher processing capability. This is due to the fact that a lower selection rate decreases DMA1 transfer times for collecting interesting events. With the WrFIFO and RdFIFO size of 32 KBytes, the highest computing throughput of 148.1 and 97.3 MBytes/s can be achieved at the interesting event rates of 25% and 100% respectively.

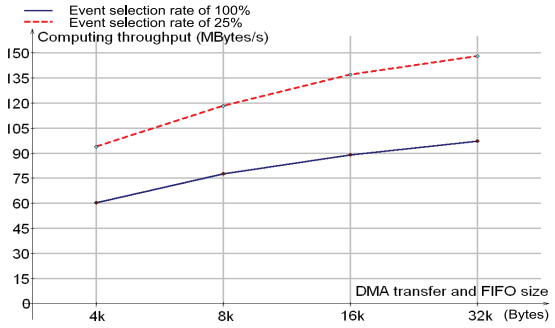

Fig. 9. Computing throughput vs. DMA transfer size

5.4. Optical Link Test

In the HADES experiment facility, our computation platform connects to the front-end Trigger and Readout Board version 2 (TRBv2) [19] by optical links and receives raw data for feature extraction processing. To conform the data rate of TRBv2, the optical link on compute nodes runs at 2 Gbps with the 8B/10B encoding. We have tested the communication between compute node and TRBv2, and no bit error occurred during our test of 150 hours.

5.5. Gigabit Ethernet Bandwidth

An end-to-end connection between the development board and a powerful desktop PC was setup to measure the bandwidth of the Gigabit Ethernet. We used the benchmark software "Netperf" [20] for bulk data transfer measurement. With all hardware/software supported features enabled which may improve performance, including Scatter/Gather DMA, checksum offloading, data realignment engines, interrupt coalescing, jumbo frame of 8982, etc., we can get the highest throughput of around 400Mbps for UDP transmitting and receiving. Details are shown in table 2, as well as for TCP transfers. In the procedure of searching for the bottleneck which restricts the practical performance lower than the physical Gigabit limitation, we found that the CPU utilization was almost hundred percent during transmitting and receiving. So we summarize that it is the CPU processing capability that dominates the Ethernet throughput in our application.

Protocol Type	Direction	Max. Throughput (Mbps)
UDP/IP	Board → PC	394.5 (TX)
UDP/IP	PC → Board	$\geq$ 394.5[1] (RX)
TCP/IP	Board → PC	297.8
TCP/IP	PC → Board	316.6

Table 2. Ethernet bandwidth measurements.

6. CONCLUSION AND FUTURE WORK

We have presented a hierarchical computation platform based on ATCA and FPGA technologies. It features flexible high bandwidth interconnections, large processing power and large DDR2 storage capability. The system is easily scalable and well suitable for various applications of data acquisition and triggering in particle physics experiments. It is also conceivable to be promising in other areas, such as medical imaging or stock market prediction for real-time processing.

Our future work includes to partition and implement the feature extraction processors. Moreover experiments are to be done on compute nodes to study the network communications.

[1] We didn't get an exact number for the maximum receiving speed. However the board could successfully receive all packets sent at the rate of 394.5 Mbps. So its receiving capability should be no less than 394.5 Mbps.

Acknowledgment

This work was supported in part by BMBF under contract Nos. 06GI179 and 06GI180.

7. REFERENCES

[1] High Acceptance Di-Electron Spectrometer (HADES) @ GSI, Darmstadt, Germany, www-hades.gsi.de.

[2] antiProton ANnihilations at DArmstadt (PANDA) @ GSI, Darmstadt, Germany, www.gsi.de/panda.

[3] BEijing Spectrometer (BES) @ Institute of High Energy Physics, Beijing, China, http://bes.ihep.ac.cn/bes3/index.html.

[4] I. Froehlich, A. Gabriel, D. Kirschner, J. Lehnert, E. Lins, M. Petri, T. Perez, J. Ritman, D. Schaefer, A. Toia, M. Traxler, and W. Kuehn, "Pattern recognition in the HADES spectrometer: an application of FPGA technology in nuclear and particle physics", *In Proc. of the 2002 IEEE International Conference on Field-Programmable Technology*, pages 443-444, Dec. 2004.

[5] Michael Traxler, "Real-time dilepton selection for the HADES spectrometer", November 2001, Ph.D thesis, II.Physikalisches Institut, Justus-Liebig-Universitaet Giessen.

[6] C. Hinkelbein, A. Kugel, R. Manner, M. Muller, M. Sessler, H. Simmler and H. Singpiel, "Pattern recognition algorithms on FPGAs and CPUs for the ATLAS LVL2 trigger", *IEEE Transactions on Nuclear Science*, Volume 48, Issue 3, Part 1, pp. 296-301, Jun. 2001.

[7] PCI Industrial Computers Manufactures Group (PICMG), "PICMG 3.0 Advanced Telecommunications Computing Architecture (ATCA) specification", Dec. 2002.

[8] R. Merl, F. Gallegos, C. Pillai, F. Shelley, B. J. Sanchez and A. Steck, "High speed EPICS data acquisition and processing on one VME board", *In Proc. of the 2003 Particle Accelerator Conference*, volume 4, pages 2518-2520, May. 2003.

[9] F. H. Worm, "Modular data acquisition system for physics", *In Proc. of the 1991 IEEE Nuclear Science Symposium and Medical Imaging Conference*, pages 823-827, Nov. 1991.

[10] Buddy Walls, Michael McClelland, and Steven Persyn, "A hybrid PCI/VME architecture for space", *In Proc. of the 2001 Aerospace Conference*, volume 5, pages 2227-2232, Mar. 2001.

[11] Y. Tsujita, J. S. Lange, and C. Fukunaga, "Construction of a compact DAQ-system using DSP-based VME modules", *In Proc. of the 11th IEEE NPSS Real Time Conference*, pages 95-98, Jun. 1999.

[12] www.dinigroup.com

[13] Chen Chang, John Wawrzynek, and Robert W. Brodersen, "BEE2: a high-end reconfigurable computing system", *IEEE Design & Test of Computers*, Volume 22, Issue 2, pp. 114-125, March-April 2005.

[14] Ming Liu, Wolfgang Kuehn, Zhonghai Lu, and Axel Jantsch, "System-on-an-FPGA Design for Real-time Particle Track Recognition and Reconstruction in Physics Experiments", *In Proc. of the 11th EUROMICRO Conference on Digital System Design*, Parma, Italy, Sep. 2008.

[15] G. N. Agakishiev and V. N. Pechenov, "Dubna tracks reconstruction user manual", Dec. 2001, HADES internal manual.

[16] Daniel Kirschner, "Level 3 trigger algorithm and hardware platform for the HADES experiment", Oct. 2007, Ph.D thesis, II.Physikalisches Institut, Justus-Liebig-Universitaet Giessen.

[17] Hal Stern, Mike Eisler, and Ricardo Labiaga, "Managing NFS and NIS (Second Edition)", O'REILLY & Associates, Inc., ISBN: 1-56592-510-6.

[18] Xilinx Inc., "BFM Simulation in Platform Studio", November 11, 2005.

[19] I. Froehlich, M. Kajetanowicz, et al., "A general purpose trigger and readout board for HADES and FAIR-experiments", *IEEE Transactions on Nuclear Science*, Volume 55, Issue 1, Part 1, pp. 59-66, Feb. 2008.

[20] www.netperf.org