Hardware/Software Co-design of an ATCA-based Computation Platform for Data Acquisition and Triggering

Qiang Wang*†, Axel Jantsch‡, Dapeng Jin*, Andreas Kopp†, Wolfgang Kuehn†, Johannes Lang†,Soeren Lange†,
Lu Li*, Ming Liu†‡, Zhen'an Liu*, Zhonghai Lu‡, David Muenchow†, Johannes Roskoss†, Hao Xu*
* Experimental Physics Center, Institute of High Energy Physics, Beijing, 100049, China
†II.Physikalisches Institut, Justus-Liebig-Universitaet, Giessen, Germany
‡Dept. of Electronics, Computer and Software Systems,Royal Institute of Technology, Sweden

***Abstract*—An ATCA-based computation platform for data acquisition and trigger(TDAQ) applications has been developed for multiple future projects such as PANDA, HADES, and BESIII. Each Compute Node (CN) appears as one of the fourteen Field Replaceable Units (FRU) in an ATCA shelf, which in total features a high performance of 1890 Gbps inter-FPGA on-board channels, 1456 Gbps inter-board backplane connections, 728 Gbps full-duplex optical links, 70 Gbps Ethernet, 140 GBytes DDR2 SDRAM, and all computing resources of 70 Xilinx Virtex-4 FX60 FPGAs. Corresponding to the system architecture, a hardware/software co-design approach is proposed to ease and accelerate the development for different experiments. In the uniform system design, application-specific computation is to be implemented as customized hardware co-processors, while the embedded PowerPC processor takes charge of flexible slow controls and transmission protocol processing.**

***Index Terms*—ATCA, Hardware/Software Co-design, Data Acquisition and Triggering.**

I. Introduction

NOWADAYS nuclear/hadron/partical physics experiments make efforts in two directions. One is to achieve higher energy, like the experiments at LHC, proposed ILC. The other is to achieve higher operation luminosity, like BESIII at BEPCII, SuperBelle at Supper KEKB and PANDA at FAIR. At the same time, detectors, front-end electronics, data acquisition and trigger system need to be able to accept higher event rates which usually means one or two orders of magnitude increase when compared with the previous experiments[1][2].

For the BESIII Trigger and Data Acquisition system design[3], an FPGA based hardware trigger selects events of interest and suppresses background to an acceptable level for the DAQ system. This system features low latency, thus requiring only a small amount of events to be buffered in the front-end electronics boards. The whole readout system works in a pipeline mode with a global system clock, representing a deadtime-less system. For future projects, much larger bandwidth both for the DAQ system and trigger system is required. Furthermore, much more sophisticated trigger algorithms with longer latency will be employed. At the same time, conventional bus based architectures such as VME limit the data throughput and should be replaced by approaches featuring high speed point to point links.

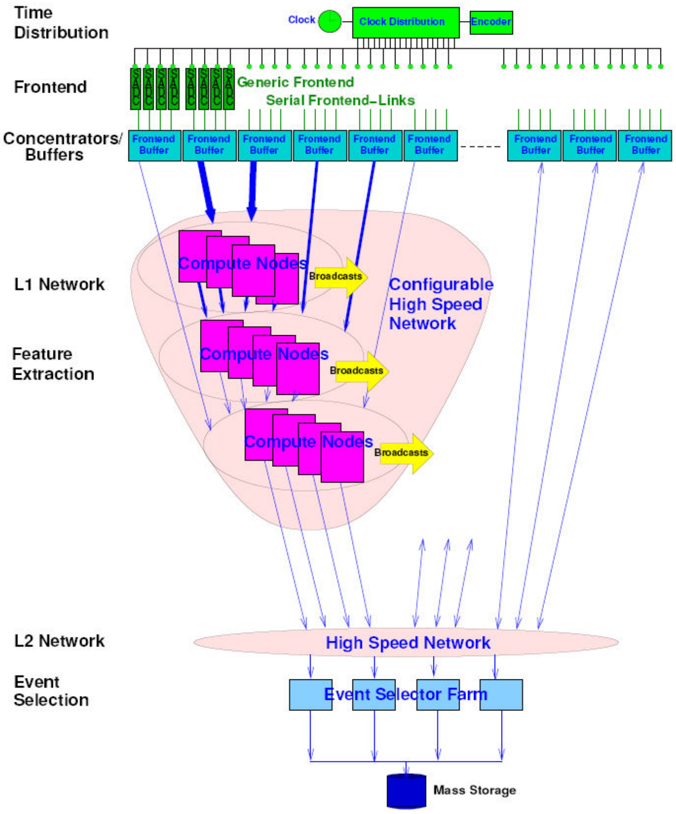


Fig. 1. The PANDA Trigger and DAQ system architecture

The PANDA experiment[4] plans to operate at interaction rates of more than 10MHz and raw data rates of up to 200GB/s. A new self-triggered data push architecture is proposed[5] as shown in Fig.1. A global precision timestamp will be used to tag the data of the individual sub-events belonging to the same event. All data from the readout electronics will steam through the TDAQ system and be accepted or rejected on the fly. Thus, high bandwidth data channels and high process capability need to be achieved. At the same time, since the PANDA experiment has a broad research program, flexibility should be a main feature of the system. Rich feature extraction algorithms for different detectors need to be implemented on a common platform and correlation

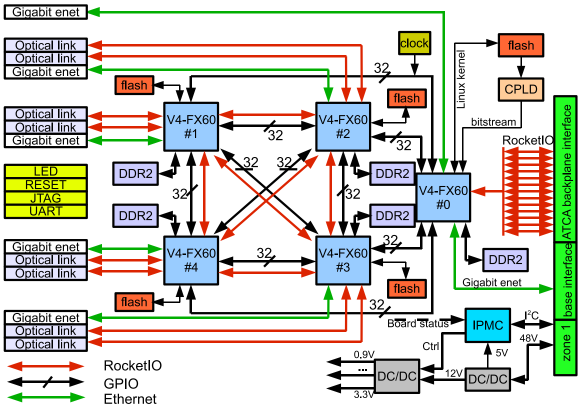


Fig. 2. Compute Node Hardware Design Diagram

of information from various detectors need be considered to provide better trigger efficiency.

II. Compute Node Hardware Design

An FPGA based architectures is still the most cost-effective solution for our requirements when a flexible, parallel and pipelined processing structure needs to be constructed to provide high processing ability as compared to systems based on commercial processor or ASIC products. High computing density with multi-FPGAs on one board imply special requirements on power supply and thermal design of the system. The correlation of different detector data requires the existence of high speed data channels interconnecting several FPGAs on one board or even interconnecting serveral boards. The new ATCA standard[6] is chosen which features good power supply, thermal design and also high speed serial point to point(P2P) link on the backplane to meet our requirements.

Fig.2 shows a block diagram of the CN board design. On one CN, five large capacity Xilinx V4FX60 FPGAs are placed. The upper four FPGAs are used to process incoming data with dedicate algorithms and the fifth FPGA is mainly used for switching data with other CN boards in the same ATCA shelf via the full mesh backplane. Each process FPGA has two RocketIO based optical links running at 2 Gbps to transmit data between FEE and CN. A Gigabit Ethernet port for each process FPGA is designed which provides an option to transmit processed data directly from the CN to online PC farms. The Gigabit Ethernet design utillizes the FPGA embedded MAC hardcore and adds an Ethernet PHY chip externally. A 2 GByte DDR2 memory is connected to each FPGA for data buffering, lookup table storage and for the on-chip embedded system. A 64 MByte parallel FLASH for each FPGA is the only permanent storage component on the board which is used to store the FPGA configuration bitstreams and the Linux kernel for the embedded system. A 32 bits parallel link and a RocketIO based serial link are available as a cross link between each two FPGAs on the same board. An Intelligent Platform Management Controller(IPMC), designed as a daughter board, is added to the CN to provide rich environment monitoring capabilities and some on-board control functions. The design of the IPMC is discussed in more detail in the contribution of Johannes Lang to this conference.

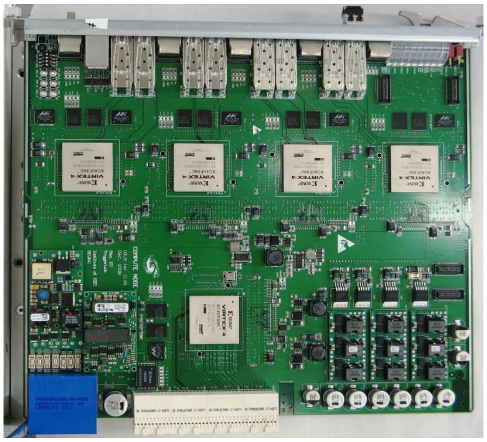


Fig. 3. Compute Node Version2 Hardware Top view

Fig.3 shows a top view of the CN version 2.0 PCB implemented within the 8U ATCA standard. The board has been demonstrated during the conference.

III. HW/SW Co-design

A. On Chip System Architecture

Considering the on-chip system architecture design, there are two different types of data flow, (i) the data received via optical link is processed directly in custom processing units and the results are stored in DDR2 memory for further processing or transmitted to the PC farm through Gigabit Ethernet; (ii) the data in DDR2 memory is processed by custom units and the results are stored back to the memory. The selection of these two types depends on the processing complexity of the trigger algorithm.

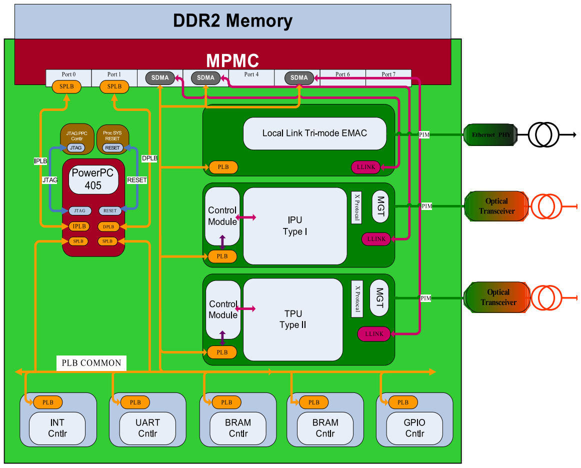


Fig. 4. MPMC based System on Chip Architecture, including a PowerPC405, Multi-Port Memory Controller, and custom processing modules

In both cases, high bandwidth data channels between custom processing units and the DDR2 memory should be provided. At the same time, the PPC405 and other components also need to access memory. Fig.4 show a System on Chip architecture employing a Multi-Port Memory Controller(MPMC).

An MPMC provides eight ports for memory access with an integrated arbiter where each port can be set to different interface standards. In Fig.4, most of the data flow originates from memory and custom processes which uses point to point LocalLink connections. LocalLink is a high-performance synchronous protocol designed for packet-oriented data transfer, that can run at 100MHz with 32 bits data width. At the same time, the Soft DMA engines integrated in each MPMC port can effectively move data between memory and custom modules. PLB is used to connect the PPC405 and other slow components to simplify the system interconnection. The modular design concept enhances system reusability. When designing for different applications, common components as well as the system structure can be reused. With this system architecture, high performance can be achieved and the system design process also turns out to be greatly simplified.

B. PPC405 based embedded system design

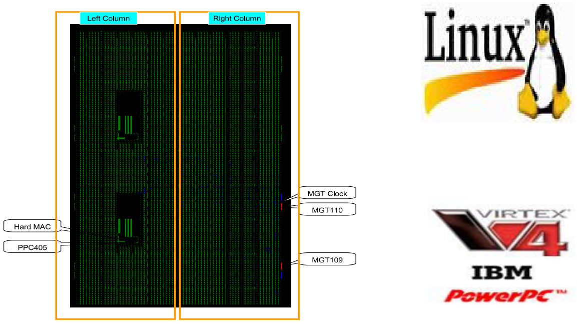


Fig. 5. Embedded system on Virex-4 FPGA

As shown in Fig.5, one Vitex-4 FX60 FPGA has two embedded PowerPC405 RISC Cores, which run up to 450 MHz. With the Xilinx provided Embedded Development Kit, most of the common interface controllers like UART, JTAG, GPIO, FLASH, BlockRAM, Gigabit Ethernet MAC can be integrated using dedicated IP cores. An open source Linux OS is ported to this platform with necessary device drivers distributed by Xilinx Git[7]. A cross compiler tool chain[8] is used to remove unnecessary components from the Linux kernel and recompile it for our application. Further existing software tools such as webserver, telnet, Flash writer can be directly ported to our platform while custom software needs to be developed. With Linux OS and cross compiler, high level languages like C/C++ or Script language can be used for custom program development. Another consideration for porting Linux OS to the platform is that we want to use Linux TCP/IP stack to process incoming UDP/TCP packets.

C. Co-design Arrangement

Fig.6 show the system Co-design arrangement and the data flow. The detector data frames are received via optical link and buffered in DDR2 memory; Rich feature extraction algorithm will be applied to extract interesting physics information such as EMC shower, Tracks and PID information; After that, sub-detector information will be correlated to suppress background events. The final stage is Event Building, which assembles selected sub-events into events. Within the FPGA fabric resources, pipelined structures can be organized between different processing stages and parallel processing algorithms at the event level can easily be designed. At all processing stages, status monitoring is necessary. At some stages, such as feature extraction, correlation and event building, some run parameters need to be set. For Gigabit Ethernet transfer, the HW/SW co-design concept is discussed in more detail:with the embedded hardcore Tri-Mode MAC in FPGA, most of the tasks like data moving from/to DDR2 memory is done by the Soft DMA controller. Some of the IP stack processing steps such as packet checksum calculations can also be offloaded to the Tri-Mode MAC. Other features of the Tri-Mode MAC such as Jumbo frame support and Tx/Rx interrupt collection improve the net transfer performance.

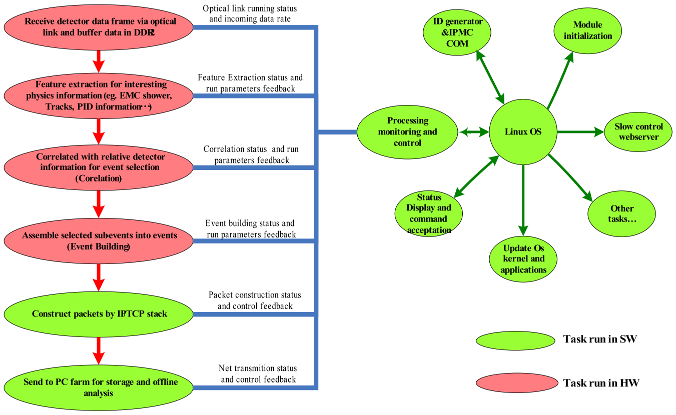

Receive detector data frame via optical link and buffer data in DDR2
Feature extraction for interesting physics information (eg. EMC shower, Tracks, PID information··)
Correlated with relative detector information for event selection (Corelation)
Assemble selected sub-events into events (Event Building)
Construct packets by IP/TCP stack
Send to PC farm for storage and offline analysis
Optical link running status and incoming data rate
Feature Extraction status and run parameters feedback
Correlation status and run parameters feedback
Event building status and run parameters feedback
Packet construction status and control feedback
Net transmition status and control feedback
Processing monitoring and control
Linux OS
ID generator &IPMC COM
Module initialization
Slow control webserver
Status Display and command acceptation
Update Os kernel and applications
Other tasks...
Task run in SW
Task run in HW

Fig. 6. Co-design Arrangement

Software tasks including module ID generation and communication with the IPMC, module initialization, status display and command processing will run on the Linux OS. Most of the custom software will be developed using high level programming languages.

D. Bootstrap Procedure

To boot up the system, all the five FPGAs need to be configured with dedicated bitstreams and the Linux OS should be booted up correctly. Fig.7 shows the system boot up steps. A dedicated on-board CPLD controls the FPGA configuration process and the FLASH memory is used to store the configuration files. The FLASH connected to FPGA0 and the CPLD stores both the bitstreams of all five FPGAs and the corresponding Linux kernels. Other FLASH chips connected to FPGA1-FPGA4 store only the Linux kernels for each FPGA.

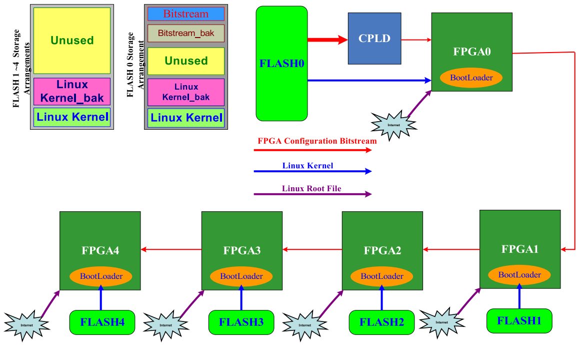


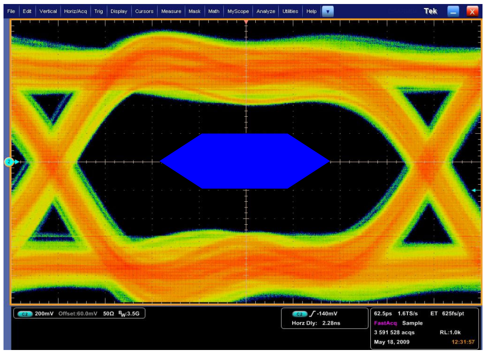
Fig. 7. Boot up system block diagram

All five FPGAs are connected in a daisy chain and configured in slave serial mode. When the board is powered up or receives a reboot signal, the CPLD moves data from FLASH to each FPGA. The bitstreams for each FPGA are concatenated with some special control codes to build a single configuration file[9]. These special control codes are used to stop configuration memory writing of one FPGA and shift the data to the next connected FPGA. In this way all five FPGAs are configured with dedicated bitstreams at one time. After that, a small boot loader program, which is loaded to BlockRAM during FPGA configuration, runs and moves the Linux kernel from FLASH to DDR2 memory. The Linux kernel starts to run when the bootloader finishes data movement and jumps to a special address. An NFS root file system is provided on a host PC which is mounted as root file system when Gigabit Ethernet becomes accessible. After that, the whole system is ready to run. For each FPGA, a backup bitstream and a backup Linux kernel are stored in the FLASH memory and can be used to recover from a corrupted bitstreams or Linux kernel. The selection of the backup system or the normal system can be done via the IPMC.

IV. System Performance Evaluation

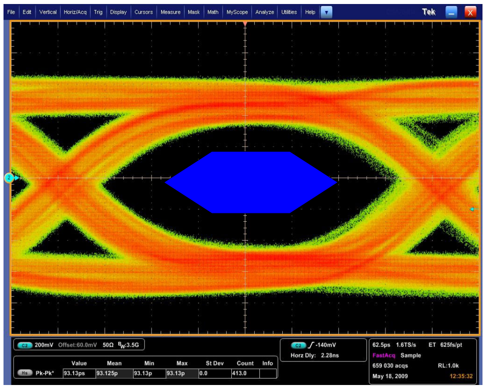
To evaluate the system performance, several tests have been done on the platform. In our system, the RocketIO based high speed serial links are applied with the optical links, cross links of on board FPGAs and backplane point to point connections. Serial links of these three types were tested and we observed no error in 144 TBytes data transferring(equaling to EBR<7*10^{-16})over the optical link. For backplane connection, the signal integrity problem becomes more critical where the 2 GHz signal is transmitted over the copper backplane for a path length of up to 40 cm from slot 1 to slot 14. A pseudo random data sequence generated by a 16 bits linear feedback shift register is used in the test[10]. As shown in Fig.8 and Fig.9, signal quality is still fine when transmitting data between slot 1 and slot 14.

Gigabit Ethernet performance is another important aspect which is measured with netperf[11]. Netperf is a benchmark that can measure various aspects of networking performance. On the host side, a netserver program needs to be started first to listen to a given port. This is similar to our application when transmitting a block of selected events from CN to PC farm. To maximize the net performance, some modifications were



```
C2 200mV  Offset:60.0mV  50Ω  BW:3.5G
C2 ∫ -140mV
Horz Dly:  2.28ns
62.5ps  1.6TS/s    ET  625fs/pt
FastAcq  Sample
3 591 528 acqs          RL:1.0k
May 18, 2009           12:31:57
```

Fig. 8. Eye diagram measured at Slot1 Tx side



```
C2 200mV  Offset:60.0mV  50Ω  BW:3.5G
           Value    Mean     Min     Max     St Dev  Count  Info
H5 Pk-Pk*  93.13ps  93.125p  93.13p  93.13p  0.0     413.0
C2 ∫ -140mV
Horz Dly:  2.28ns
62.5ps  1.6TS/s    ET  625fs/pt
FastAcq  Sample
659 030 acqs            RL:1.0k
May 18, 2009           12:35:32
```

Fig. 9. Eye diagram measured at Slot14 Rx side

made on both the CN side and PC side. On the Compute Node side, the Rx/Tx FIFO depth is increased, the Scatter Gatter DMA engine is enabled and the checksum offload function of the TriMode MAC is also used. Some modifications on the host PC side were made to achieve improved performance[12]. 212.37 Mbps throughput from PowerPC to PC can be achieved and a better result of 241.42 Mbps throughput is obtained when transmitting data from PC to CN. In comparison, a VxWorks 5.5 OS was also ported to the same platform and the net performance of these two OS are almost the same. During the test, we also found that the CPU utilization reached 100% which seems to be the main limitation for further performance improvement. Given that the PowerPC405 is not a powerful processor, the result is acceptable. However the performance is sufficient assuming that the trigger algorithms on the CN can reduce the incoming data rate to one thirtieth of original data rate[4](roughly from 1.6Gbps*2 to 107Mbps).

V. Conclusion and Future work

We have presented an FPGA-based hardware/software Co-design approach for the new TDAQ system which can not only achieve high performance but also greatly facilitate the system design process. The CN represents a general purpose hardware platform which is suitable for many future applications in data acquisition and triggering. A flexible I/O architecture based on optical links, RocketIO and Gigabit Ethernet provides high bandwidth connectivity to both front-end electronics and PC farms. Up to 10 GByte DDR2 RAM storage provides ample space for data buffering and other memory intensive applications. Five high end FPGA provides the required computing resources. Multiple CNs can be combined in a single ATCA shelf and multiple shelves can be combined to create a large system. Thus the system can be easily scaled to fit all kinds of requirements. As for algorithm development, please see the contribution of Ming Liu to this conference.

For this platform, still some more studies are needed. In the future, further development may cover serial link protocols for backplane and optical link transmission, drivers for custom components, as well as studying how to effectively share memory with multiple processing cores.

ACKNOWLEDGMENT

This work was supported in part by China national 973 project under contract No.2008CB817702, BMBF under contract Nos.06GI179 and 06GI180, FZ-Juelich under contract No. COSY-099 41821475, Internationales Buero des BMF and DAAD-CAS Joint Scholarship.

REFERENCES

[1] BESIII Collaboration, *BESIII Preliminary Design Report*, 2004.

[2] K. Abe et al., http://superb.kek.jp/documents/loi/img/LoI_detector.pdf, http://superb.kek.jp/documents/loi/img/LoI_accelerator.pdf

[3] Z.A. Liu, W.X. Gong, Y.N. Guo, D.P. Jin, L. Li, Y.P. Lu, Q. Qiao, K. Wang, S.J. Wei, H. Xu, Y.Y. Zhang, D.X. Zhao, *Trigger System of BESIII*, 15th IEEE NPSS Real Time Conference, 2007.

[4] PANDA Letter of Intend, *PANDA Technical Design Report*, http://www.gsi.de/panda

[5] W. Kuehn, C. Gilardi, D. Kirschner, J. Lang, S. Lange, M. Liu, T. Perez (JLU, Giessen), L. Schmitt (GSI, Darmstadt), D.P. Jin, L. Li, Z.A. Liu, Y.P. Lu, Q. Wang, S.J. Wei, H. Xu, D.X. Zhao (IHEP Beijing, Beijing), K. Korcyl, J.T. Otwinowski, P. Salabura (Jagiellonian University, Krakow), I. Konorov, A. Mann (TU Mnchen, Garching), *FPGA-Based Compute Nodes for the PANDA Experiment at FAIR*, 15th IEEE NPSS Real Time Conference, 2007.

[6] *PICMG3.0*,http://www.picmg.org

[7] Xilinx Inc. *http://git.xilinx.com*

[8] ELDK. *http://www.denx.de/wiki/DULG/ELDK*

[9] Xilinx Inc. *http://www.xilinx.com*, UG071, April 8, 2008, P54

[10] H. Xu, Z.A. Liu,Y.P.Lu, L. Li, D.X. Zhao, Y.N. Guo, *FPGA based high speed data transmission with optical fiber in trigger system of BESIII*, 07's Nuclear Science Symposium Conference, 2007.

[11] *http://www.netperf.org*

[12] Xilinx Inc. *Benchmarking the Performance of the Virtex-4 10/100/1000 TEMAC System*, October 3, 2007.