A HT3 Platform for Rapid Prototyping and High Performance Reconfigurable Computing

Frank Lemke, Sven Kapferer, Alexander Giese, Holger Fröning, Ulrich Brüning
Computer Architecture Group
University of Heidelberg
Mannheim, Germany

{frank.lemke,sven.kapferer,alexander.giese,holger.froening,ulrich.bruening}
@ziti.uni-heidelberg.de

***Abstract* — FPGAs as reconfigurable devices play an important role in both rapid prototyping and high performance reconfigurable computing. Usually, FPGA vendors help the users with pre-designed cores, for instance for various communication protocols. However, this is only true for widely used protocols. In the use case described here, the target application may benefit from a tight integration of the FPGA in a computing system. Typical commodity protocols like PCI Express may not fulfill these demands. HyperTransport (HT), on the other hand, allows connecting directly and without intermediate bridges or protocol conversion to a processor interface. As a result, communication costs between the FPGA unit and both processor and main memory are minimal. In this paper we present an HT3 interface for Stratix IV based FPGAs, which allows for minimal latencies and high bandwidths between processor and device and main memory and device. Designs targeting a HT connection can now be prototyped in real world systems. Furthermore, this design can be leveraged for acceleration tasks, with the minimal communication costs allowing fine-grain work deployment and the use of cost-efficient main memory instead of size-limited and costly on-device memory.**

Hyper Transport, FPGA, High Performance Reconfigurable Computing

I. INTRODUCTION

In the area of accelerated computing the vast amount of research and development focuses on using GPUs [1] [2] [3]. Compared to this, FPGAs are very sparely used. The main reasons for this are certainly the cost advantage of GPUs (with a mass market behind), and the easier way of programming. FPGAs are for most users difficult to program, and due to their small volume they have approximately one order of magnitude higher costs.

However, GPUs are very limited in their usage. Only if the application to be ported to the accelerator has characteristics similar to graphical processing, it can be successfully accelerated [4]. Additionally, a recent report by Intel [5] shows that the speedup between CPUs and GPUs is only about 2.5 in average. Also, the limited amount of graphics memory is preventing a broad use, because the stream processors of a GPU can only operate on this memory.

FPGAs, on the other hand, are much more flexible due to their completely reconfigurable architecture. In particular for applications which are not suitable for GPUs they play an important role [6] [7] [8]. It is also possible to attach a large amount of memory to the FPGA, making it suitable for data-intensive applications.

GPUs with their stream based processing do not rely on a close coupling between accelerator and host system, thus they cannot offer applications the possibility of fine grain accesses to and from the host system. However, many applications rely on such a tight integration. Again, this demand can be fulfilled by FPGAs, in particular if a system interface like HT is used and not a peripheral interface like PCIe. If the interface to the host system is lean enough, the costs for accessing main memory are not higher than accessing memory attached to the FPGA. Then, it is possible for the FPGA to operate directly on main memory, making arbitrary amounts of memory possible.

Last, as more and more performance computing systems are facing the power wall, the GFLOPs achieved per Watt are of paramount importance. FPGAs are certainly one of the best architectures for high GFLOPs/Watt. By equipping installations with FPGA based accelerators, the power consumption can be significantly reduced while maintaining the computing performance.

As hardware platform enabling above described features a Stratix IV HTX3 Board was used. Based on an existing first version prototype it was enhanced by placing additional components onto the board and some refinements resulting in the version presented here providing all required basic functionalities. For using it as fully capable HT3 device in a system the HT3 core [9] had to be ported onto the Altera FPGA.

Also to ensure the usability and reliability of communication between the device and the processors in HT3 systems HW simulations had to be performed. Additionally a physical interface (PHY) had to be created to deliver an interface for the HT3 core to be compatible with the provided hardware environment. This work will enable

the Stratix IV HTX3 Board being used as a unique single FPGA HT3 solution which supports all the required features for HT3 and therefore representing an efficient platform for Rapid Prototyping and High Performance Reconfigurable Computing.

The next section presents the HyperTransport protocol as base technology for low latency communication. The architecture of the Stratix IV HTX3 Board serving as rapid prototype platform is specified in section 3 followed by the description of the HT3 implementation enabling high performance reconfigurable computing on top of it in section 4. The fifth section presents measurement results. Finally a conclusion and an outlook are given in section 6.

II. HYPERTRANSPORT

HT is a unique possibility to easily connect a device directly to a processor. As it is the only public specification [10] available to do so, it is the perfect vehicle for a low latency communication as there are no unnecessary protocol conversions or bridges involved. With the HTX3 connector which is defined by the HyperTransport-Consortium (HTC) [11] and the availability of Opteron mainboards a system can be easily set up [12].

HT allows a broad variation of link widths and frequencies from a 2 bit link at 200 MHz DDR (HT200) up to a 32 bit link at 3.2 GHz DDR (HT3200). Current Opteron architectures support link widths and frequencies from 8 bit at 200 MHz DDR up to 16 bit at 3.2 GHz DDR. This results in a theoretically maximum unidirectional bandwidth of 12.8 GB/s. The signal lines carry the control-, data- and info-packets and are called CAD. Depending on the link width those signals are grouped into independent byte lanes. Every byte lane is accompanied by a single signal lane of additional control information called CTL and a clock signal. As HT is doubleword (32 bit) aligned every doubleword of CAD comes along with 4 bit of CTL which contains additional information about what kind of data is transported.

Three types of the specification exists HT1, HT2 and HT3. HT1 and HT2 only differ in the maximum link frequency. The functionality of the first two versions is described in [13]. HT3, which is realized in state of the art Opteron processors, begins at a link speed of HT1200 and requires features to be implemented such as link training, link deskew, a retry protocol and stomping which were in earlier versions optional or not defined.

To realize link training, each bit lane has to support a mechanism to align its logic with the help of a special training pattern sequence. After link initialization the single bit lanes are deskewed to ensure proper data alignment. Therefore the receiving fifos must be able to handle an amount of 8 bit-times of misalignment from one lane to another. Compared to HT1 and HT2, the higher frequencies of HT3 result in an increased possibility of bit errors on the physical level. A retry mechanism is introduced to handle those errors. The error detection is enhanced due to changing the periodic CRC from HT1/HT2 every 512 bit times to a per packet CRC. Thereby latency and the needed buffer space for retransmission are reduced and a better performance can be achieved. Each packet which CRC is checked correctly increments an acknowledgement counter. If a NOP packet is sent it contains the counter value of the last correctly checked packet. The retry buffers on the receiver side of the NOP packet can then be released. If an error occurred during a transmission the retry handshake is initiated and the data from the last correctly received packet is retransmitted. Stomping is an additional feature to reduce latency. It is used to speculatively forward a packet without CRC being checked. If later the CRC shows an error the CRC is inverted to show the final endpoint that the packet has to be invalidated. A block diagram of the HT3 implementation is shown in figure 1.

HT3 leverages the possibility of higher link speeds by introducing fault detection and recovery mechanism to the HyperTransport protocol. But it requires changes on the physical layer as well which will be described in section 4.

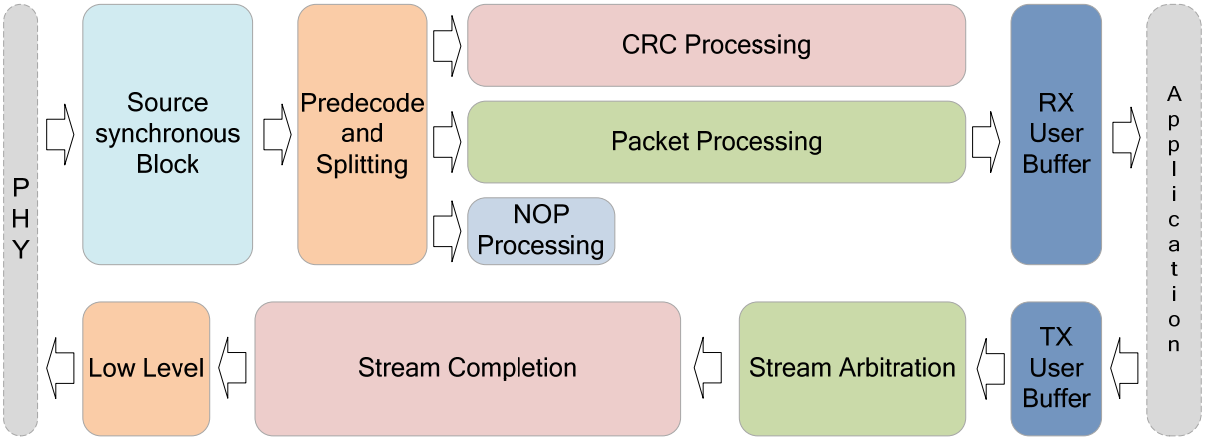

Figure 1. HT3 Blockdiagram

III. BOARD ARCHITECTURE

The board design is based on a PCI normal sized card using a HTX3 connector slot for a low latency HT link connection to the system. The main board component is a Stratix IV GX device family FPGA [14]. The selected FPGA uses a F1517 footprint enabling EP4SGX 180, 230, 290, 360 and 530 variants. The used device provides an adequate number of LVDS and I/O pins to enable numerous prototyping features and 36 transceivers giving the capability to use HT3 and two CX4 links with up to 6.375 Gbps per lane for network connections. The CX4 links do support implementations for Infiniband DDR. There are also standard interfaces and components available to use the board in different environments.

For prototyping purposes using extension cards or user defined connectors extension adapters have been placed onto the board. The primary used adapter is a SEAF connector from Samtec with 500 pins supporting single-ended signaling up to 9.5 GHz and differential pair signaling up to 10.5 GHz. Thus speed restriction is primarily defined by the FPGA. The pins used within the connector are shielded considering the suggestions of Samtec. This resulted in 114 single-ended and 55 differential pair connects together with the FPGA.

Figure 2. Stratix IV HTX3 Board

Further three QTH series Samtec connectors with 120 pins each organized in two banks with integrated metal plane used as ground are assembled. These connectors provide at least 9GHz single-ended and 8 GHz differential pair capability. The connections to the FPGA are designed to provide up to 108 differential pairs plus sideband signals.

The board was enhanced and upgraded in several design steps. Figure 2 shows the latest revision. All components are tested. It can be used as a prototyping platform or directly for high performance reconfigurable computing needs.

IV. HT3 IMPLEMENTATION

Before porting the HT3 core onto the Altera device an implementation of a PHY had to be realized. Therefore the high frequency traces had to be simulated ensuring that all parameters were within the specification.

A. Simulation

During simulation of the HT link all HT tracks between the Opteron processor and the Stratix IV GX were analyzed. All simulations were performed using IBIS and HSpice models. For the FPGA high speed serial transceiver an HSpice model and for the Opteron processor IBIS models were available. For the HTX connector, which is identical to the PCIe connector, the Samtec Spice model has been used. The required S-Parameter files among others for the vias are generated by the Cadence Allegro PCB design suite. HT3 starts with a minimum of HT1200 with a frequency of 1200MHz and a data rate of 2.4Gbps. This was also the simulation target for the first simulations. Figure 3 shows a representation of the simulated tracks at HT1200 for a HTX3 CADOUT signal. There are three different measure points available, the signal after the Stratix IV GX package, on the receiver pin, and after equalization through the Stratix IV GX Clock Data Recovery (CDR) unit. Depending on the measure point the eye height is in the range from 531 mV to 998mV and the eye width is around 374ps. According to the HT physical specification [11] the eye height must be over 140mV and the minimal eye width must be 0.55 unit intervals (UI), the UI for 2.4Gbps is 416ps. All simulated tracks at HT1200 were clearly within the specification.

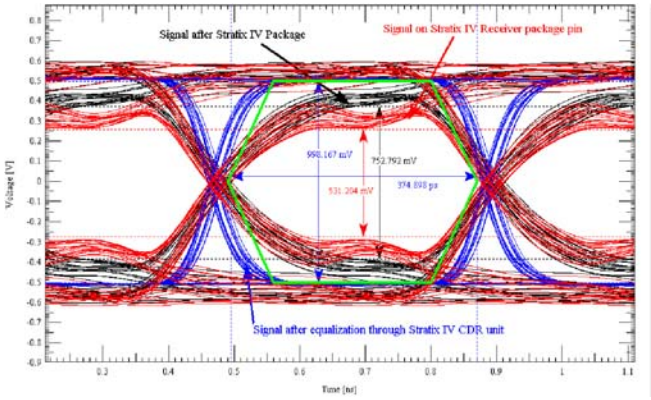

Figure 3. HTX Track Simulated at HT1200

Also simulations using the maximum frequency of the high speed Stratix IV GX transceivers at 6.4Gbps were performed. The HT specification for this frequency requires a minimum eye width of 0.65 UI, which results in 100ps and a minimum eye height of 170mv. One of the most critical extracted tracks is depicted in figure 4. Its eye width is 107ps and the height is 224mV. All simulations show, that the hardware is capable of HTX3 usage.

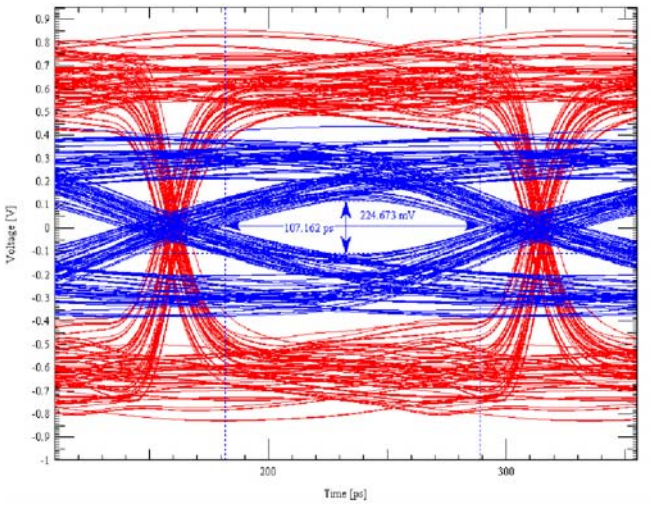

Figure 4. Eye Diagram at 6.4Gbps Link Speed.

B. *HT3 PHY*

A PHY for HyperTransport 3 must also support HT1 operation because the HyperTransport protocol is backwards compatible. However, this means that the PHY must support two inherently different operation modes. HT1 is working in a source synchronous mode and transmits a link clock in addition to the data lanes which is used to sample the incoming data. Since HT3 operation starts at a link frequency of 1.2 GHz and can go up to 3.2 GHz a different technique must be used. Because the skew requirements between clock and data would be in the range of picoseconds if the same source synchronous mode was used for HT3 frequencies the clock is now recovered at the receiver side by using CDR. In order to ensure enough transitions for a reliable clock reconstruction scrambling is mandatory for HT3 operation. The HyperTransport protocol specification defines several line rates for HT3 operation in the range of 2.4 Gbps to 6.4 Gbps. Because these line rates exceed the maximum supported data rate of LVDS transmitter / receivers by far, high speed serializers must be used to work in HT3 mode. In order to implement proprietary protocols the Stratix IV GX transceivers support an operating range from 600 to 3750 Mbps in single width mode using an 1:16 serialization factor and from 1000 to 6500 Mbps in double width mode with an 1:32 ratio for the -2 speed grade we used for our board. In Stratix IV devices transceivers are grouped in blocks consisting of 6 transceivers as shown in figure 5. Four of those channels support both physical coding sublayer (PCS) and physical medium attachment (PMA), the other two channels are clock multiplier unit (CMU) channels that can be configured either as a normal data channel without PCS support or as a clocking block that provides both the serial and the parallel clock to the other channels.

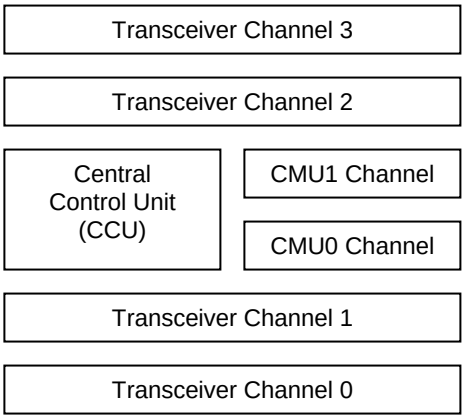

Figure 5. Stratix IV Transceiver Block Architecture [14]

For each sublink, 9 lanes (8 CAD + 1 CTL) are connected to the fully featured serializers and both the incoming and outgoing link clock are connected to separate CMU channels. In order to provide a deterministic latency across all channels the transceivers are configured in PMA direct mode. All required PCS functionality is provided inside the FPGA, the existing transmitter and receiver PCS blocks in the hardware are completely bypassed. Since all HyperTransport clocks are derived from the 200 MHz HT reference clock this clock is also connected to a global clock pin of the Stratix IV FPGA after it was jitter cleaned to improve the transceiver performance.

Although the HyperTransport 3 specification specifies an AC coupled operation mode as well as a DC coupled mode, AC coupled operation is not supported by AMD's Opteron processor and therefore irrelevant for all practical purposes. Since the HyperTransport specification and the Stratix IV datasheets define different common modes electrical compatibility between the Opteron and Altera's transceivers had to be verified by HSPICE simulations as described in the previous paragraph and also confirmed by Altera engineers.

All HyperTransport systems start in Gen1 mode running with a 200 MHz link clock (HT200). The resulting data rate of 400 Mbps is below the minimum supported rate of the Stratix IV transceivers. In order to overcome this limitation, the PHY runs five times faster than actually required by the data rate and uses a 5 time oversampling mode for incoming data. In the same way, for the outgoing direction each bit is just replicated 5 times to emulate a link running at HT200.

Since HT1 employs neither scrambling nor 8b/10b encoding clock recovery from the data stream cannot be used, therefore the transceivers are configured in lock-to-reference (LTR) mode and use the HT reference clock to create the sampling points. The link clock on the receiving side is not used to sample the incoming data. In order to create the transmit clock that must be shifted by 90 degrees in relation to the data stream as defined in the HT specification the clock data pattern is padded accordingly

so that the clock is driven one half of a bit-time after a data transition. As described above all PCS handling is done in the FPGA fabric. This means that the PHY only handles the basic serialization and deserialization, data word boundaries are not detected at all by the PHY. All alignment is done later inside the HT3 core.

In order to switch to HT3 mode, starting at 1.2 GHz, several things must happen inside the PHY. The oversampling path that was used for HT200 must be bypassed; the data is now processed directly as the link data rate is now natively supported by the transceiver. The transceivers also switch from LTR to CDR and the clock recovery circuitry must lock to the data stream. This means that each lane has its own recovered clock that is used to sample the data. Although each of these lanes will run at the same frequency there will be a phase difference between the different lanes. Elastic buffers are used in order to transfer all the lanes in a single clock domain to process the data stream in parallel. Unlike in HT1 mode this can also lead to inter-lane skew which will also be removed in the HT3 core. The link clock in HT3 is not used at all and requires no special handling since there is no relation between clock and data and each lane has its own embedded clock.

The PHY also supports the LDTSTOP signal defined by HyperTransport specification that is used to disconnect the links. During this time no data is transmitted over the link and the link is idle. Because the CDR circuitry does not recover reliably from this condition after the link restarts the PHY switches back to LTR mode during LDTSTOP and goes back to CDR after the link resumes normal operation and scrambled data patterns are transmitted again.

The PHY does not include any error detection mechanisms. All signal integrity issues are caught by the HT3 core using the reliability features defined in the HT3 protocol.

V. MEASUREMENTS

The measured latency of our HT3 core together with the HT3 PHY at a HyperTransport link frequency of 1600 MHz running in 8 bit mode in a Tyan 2912-E motherboard with two Opteron processors running at 2800 MHz was 655ns round trip for a single PIO access to the device. This is much higher than the latency measured for our HT1 core [15] in an older system with a slower processor. There are several reasons for this large difference. The higher complexity of the HT3 protocol forced us to implement more pipeline stages to decode the incoming packets. The most prominent factor, however, is the usage of serializer technology inside the FPGA instead of normal LVDS IO cells and the crossing of several clock domains inside the PHY.

The first bandwidth measurements showed rather disappointing results that were not even in the range of half the available bandwidth offered by the link. This was caused by credit starvation [9] because the default BIOS configuration of the link did not allocate enough credits for the posted VC inside the processor. After redistributing the credits to achieve a better link utilization bandwidth measurements using data packets with the maximum allowed payload of 64 bytes showed a write performance of about 2000 MB/s for a DMA Write operation and an average bandwidth of about 1600 MB/s for a DMA Read operation. These numbers, albeit being a huge improvement, show that full utilization of a HT link can only be reached by a device with a fast internal clock speed that can release credits almost instantaneously as soon as new packet is received. The performance that can be reached by an FPGA suffers mainly from the credit starvation that occurs during operation that is caused by the latency added by the serializers and the many pipeline stages in the core.

The consumption of resources within the FPGA shows that there is enough space left to add user logic for prototyping and high performance computing. The synthesis results for the Stratix IV GX 230 device depict resource usage of combinational ALUTs 42,534 / 182,400 (23 %), memory ALUTs 49 / 91,200 (< 1 %), dedicated logic registers 40,009 / 182,400 (22 %), a logic utilization of 34 %, and a total block memory bits 739,154 / 14,625,792 (5 %).

VI. CONCLUSION AND OUTLOOK

Both the HT3 PHY in conjunction with the HT3 core and the developed Altera FPGA based card work reliably in our Tyan test system. Sporadic bit errors that were encountered during operation were easily caught and recovered by the reliability features defined in the HT3 protocol and had no impact on the functionality.

Both HT1200 and HT1600 implementations are stable and work as expected. Unfortunately, the core speed directly scales in relation to the HT link speed as there is no flow control between the PHY and the HT3 core. Thus, reaching higher HT link speeds is currently limited by the HT3 protocol that leads to a complex hardware architecture for the HT3 core and makes internal core frequencies larger than 200 MHz rather difficult to achieve.

The HT3 platform for rapid prototyping and high performance reconfigurable computing was a successful development. It represents the first single FPGA HT3 implementation in comparison to the 3 FPGA solution developed in [16]. Due to the provided low latency high bandwidth connection directly to the processor this platform delivers an ideal environment for developments and research in the areas of coprocessors or FPGA accelerators. Also its numerous extension connectors enable the usage of extender cards such as a card with a Content-Addressable Memory (CAM) and a reasonable amount of RAM to realize a network search engine (NSE).

ACKNOWLEDGMENT

We would like to express our thanks to Elmar Greulich for the work he brought into the first revision of the Stratix IV HTX3 Board. We also thank Altera for the PHY development and AMD for their excellent support.

REFERENCES

[1] Owens, J. D., Luebke, D., Govindaraju, N., Harris, M., Krüger, J., Lefohn, A. E., and Purcell, T. 2007. A Survey of General-Purpose Computation on Graphics Hardware. Computer Graphics Forum, volume 26, number 1, 2007, 80-113.

[2] Owens, J. D., Houston, M., Luebke, D., Green, S., Stone, J. E. and Phillips, J. C. 2008. GPU Computing. In Proceedings of the IEEE, 96, 5 (May 2008), 879–899.

[3] Alerstam, E., Svensson, T., and Andersson-Engels, S. 2008. Parallel computing with graphics processing units for high-speed Monte Carlo simulation of photon migration. In *Journal of Biomedical Optics, vol. 13, issue 6*, Nov. 2008.

[4] Khailany, B., Dally, W. J., Kapasi, U. J., Mattson, P., Namkoong, J., Owens, J. D., Towles, B., Chang, A., and Rixner, S. 2001. Imagine: Media Processing with Streams. *IEEE Micro 21, 2* (Mar. 2001), 35-46.

[5] Lee, V. W., Kim, C., Chhugani, J., Deisher, M., Kim, D., Nguyen, A. D., Satish, N., Smelyanskiy, M., Chennupaty, S., Hammarlund, P., Singhal, R. and Dubey, P. 2010. Debunking the 100X GPU vs. CPU Myth: an Evaluation of Throughput Computing on CPU and GPU. *SIGARCH Comput. Archit. News* 38, 3 (June 2010), 451-460.

[6] Das, S., Agrawal, D., and Abbadi, A. E. 2008. CAM conscious integrated answering of frequent elements and top-k queries over data streams. In *Proceedings of the 4th International Workshop on Data Management on New Hardware*, Vancouver, Canada, June 2008.

[7] Bandi, N., Metwally, A., Agrawal, D., and El Abbadi, A. 2007. Fast data stream algorithms using associative memories. In *Proceedings of the 2007 ACM International Conference on Management of Data (SIGMOD '07)*, Beijing, China, June 2007.

[8] Fröning, H., Nüssle, M., Litz, H., and Brüning, U. 2010. A Case for FPGA based Accelerated Communication. In *Proceedings of 9th International Conference on Networks (ICN 2010)*, Menuires, France, April 2010.

[9] B. Kalisch, A. Giese, H. Litz and U. Bruening, "Hypertransport 3 Core: A Next Generation Host Interface with Extremely High Bandwidth", First International Workshop on Hyper Transport Research and Applications (WHTRA), Mannheim, Germany, Feb. 2009.

[10] HyperTransport™ Consortium: HyperTransport™ I/O Link Specification, http://www.hypertransport.org, June 2010

[11] HyperTransport™ Consortium: HTX3™ Specification for HyperTransport™ 3.0 Daughtercards and ATX/EATX Motherboards, http://www.hypertransport.org, Jun. 2008.

[12] Hypertransport Consortium: The Future of High-Performance Computing: Direct Low Latency CPU-to-Subsystem Interconnect, http://www.hypertransport.org, 2008.

[13] D. Anderson, and J. Trodden, HyperTransport System Architecture, Addison-Wesley, 2003.

[14] Altera: Stratix IV Device Handbook, SIV5V1-4.1, http://www.altera.com, July 2010.

[15] David Slogsnat, Alexander Giese, Mondrian Nüssle, Ulrich Brüning, "An Open-Source HyperTransport Core", ACM Transactions on Reconfigurable Technology and Systems (TRETS), Vol. 1, Issue 3, p. 1-21, Sept 2008.

[16] Heiner Litz, Holger Fröning, Maximilian Thürmer, Ulrich Brüning, "An FPGA based Verification Platform for HyperTransport 3.x", 19th International Conference on Field Programmable Logic and Applications (FPL2009), Prag, Czech Republic, Aug. 31-Sept. 2, 2009.