An FPGA based Verification Platform for HyperTransport 3.x

Heiner Litz Holger Froening Maximilian Thuermer Ulrich Bruening

University of Heidelberg
Computer Architecture Group
Germany
{heiner.litz, holger.froening, ulrich.bruening}@ziti.uni-heidelberg.de

ABSTRACT

In this paper we present a verification platform designed for HyperTransport 3.x (HT3) applications. It is intended to be used in computing environments in which it is directly connected over a HyperTransport link to the main CPUs. No protocol conversions or intermediate bridges are necessary, which results in a very low latency. An FPGA technology is chosen due to of its reconfigurability. The main challenge of this work is the implementation of an HT3 link using FPGA technology. We provide a design space exploration in order to find the most suitable I/O technology. This includes verification of the HyperTransport-specific signal levelling, the development of an I/O architecture supporting a link width of up to 16 lanes with source synchronous clocking and a clocking scheme to readjust transmission rates as it is demanded by the HT3 specification. The functionality of the developed verification platform is shown in a real world environment, in which the first prototypes are brought up in an HT3 system connected to an AMD processor. Early adopters of HT3 benefit from the results of this work for rapid prototyping and Hardware/Software co-verification of new HT3 designs and products. Additionally, the developed platform is highly suitable for accelerated computing and for fine grain communication applications.

1. Introduction

The large capacity of today's silicon technology allows complex *System-on-Chip (SoC)* designs. These match the demand for more and more powerful products while keeping the cost - compared to multi-chip solutions - relatively low. Due to of the huge complexity of these SoC designs, verification is mandatory. In particular, for *Hardware/Software (HW/SW)* co-verification *Field Programmable Gate Arrays (FPGAs)* have several advantages compared to *Electronic Design Automation (EDA)* tools. FPGAs allow to run prototypes in real world environments, in which both hardware modules and software layers can be verified in an early prototyping stage. However, the drawback is that FPGA-based verification platforms have to be specially designed for certain applications, in particular if the application is based on a new technology.

The verification platform presented here connects to a *HyperTransport (HT)* link. *HyperTransport 2.x (HT2)* [1][2] is an established technology for chip-to-chip connections offering a very low latency and a high bandwidth. It is used in all major AMD processors, including the Opteron [3]. The most recent development in the area of HyperTransport is the availability of the *HyperTransport 3.x (HT3)* [6] and the *HyperTransport Extension 3 (HTX3)* [7] specification. The first one allows for very high operating frequencies (3.2GBit/s per lane), the second defines a connector for add-in cards with direct HT3 connection to one of the CPUs.

With the availability of HT3 the need arises to prototype and verify new products. The HTX3 interface introduces the possibility to use add-in cards for verification of HT3 based products. FPGAs are the most appropriate choice for this task, as they combine full reconfigurability with high performance. Best to our knowledge no HT3 prototyping or verification platform is available today.

Beside the need to prototype new developments, the advantages of the HT3 interface can be leveraged. The excellent performance in terms of high bandwidth, low latency and optional cache coherency makes an HT3 unit highly suitable for communication engines or accelerated computing platforms. For networking engines typical message passing based applications leverage the high band-

width and the low latency, allowing fine grain communication schemes. With a coherent HT3 interface coherency domains can be built, comprising more than one node. These coherency domains can be leveraged by *Partitioned Global Address Space (PGAS)* applications. Accelerated computing platforms generally demand high bandwidth for data transfers and close coupling for fine grain work issue and synchronization. Depending on the application they can also benefit from a cache coherent interface: The accelerator can either implement coherent caches or expose its memory to the coherency domain.

These application areas and the success of our recent HT2 rapid prototyping platform [8][9][10] are motivation us for the work presented herein. While the HT2 interface will probably not be able to keep the pace with future requirements for even higher bandwidth, the HT3 specification contains enough headroom for tomorrow's applications.

1.1 Contributions

This paper embraces the work to develop a verification platform which is based on FPGAs and compliant to the HT3 and HTX3 specification. Our contributions of this work are as follows:

- First HT3 verification platform based on FPGAs.
- Interface technology for adjustable transmission rate.
- Verification of the FPGA's electrical interface compliance to the HT specification.
- Development of an architecture to support a full 16 bit wide link, enabling a bandwidth of up to 9.6 GByte/s.

The functionality of the developed verification platform is shown in a real world environment, with first prototypes being brought up in an HT3 system connected to an AMD processor.

The rest of the paper is organized as follows: The background of this work is presented in Section 2. Section 3 is dedicated to the HT interface and the simulation of an HT link. In section 4 the basic architecture of the verification platform is described. The evaluation is presented in Section 5, followed by a conclusion in the last section.

2. Background

HT3 is an emerging technology which is currently used by all AMD processors. HT3 units can either be located on the mainboard, in the CPU socket or on an HTX3 add-in card. In all cases HT units are directly connected to the CPU(s) without any intermediate bridges or any kind of protocol conversion. Custom HT3 units will emerge as soon as HT3 CPUs are available. The complex design of these units and the integration in the system requires in-system verification. This, as well as the possibility of rapid prototyping is the main motivation for the work presented in here.

Beside this, the high performance of HT3 in terms of extremely high bandwidth of up to 25.6 GByte/s[1] and low latency due to the direct connection makes it suitable for high performance applications. Additionally, the direct connection allows to participate in the cache coherency protocol of the CPUs [5]. Typical applications are accelerated computing [13][14][15], fine grain communication [4] and distributed shared memory [12][16][17].

Figure 1 shows the system-level block diagram of an HT3 environment. It consists of two CPUs and the HT3 verification platform, all interconnected by HT3 links. Using the CPU's integrated memory controllers the unit can access the system's main memory. This set-up allows for HW/SW co-verification, as the hardware modules are implemented on the FPGA and the software layers run on the CPUs. Nonetheless, similar set-ups with more or fewer CPUs are also possible.

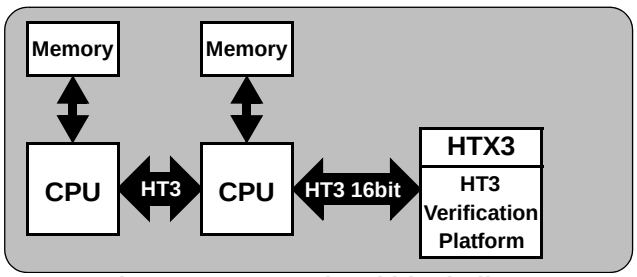

Figure 1. System-level block diagram

3. HT Interface and Simulation

Most important for a working HT3 verification platform is the successful implementation of the HyperTransport link itself. The difficulty of this task is that HyperTransport 3 runs at a minimum clock frequency of 1.2 GHz in *Double Data Rate (DDR)* mode resulting in a per lane bandwidth of 2.4 Gbps. For current FPGA technologies this represents a real challenge. The HT interface is comprised of 16 lanes that carry *Control and Data (CAD)* information subdivided into two 8 bit blocks called links. Despite its CAD lanes a link is comprised of an additional *control (CTL)* signal and a link *clock (CLK)*. Both links can be either aggregated or used independently from each other. The links can be operated in two different modes: HT 1.x/2.x is called *Gen1* which supports link frequencies of up to 1 GHz and HT 3.x is referred to as *Gen3* which increases the range to 1.2 GHz - 3.2 GHz. The Gen1 interface is source synchronous and uses the link clock which is phase shifted by 90 degrees to sample the incoming data. Due to the high clock speeds in Gen3 mode and the resulting skew requirements for the

1. Maximum theoretical bandwidth for a 16 bit wide HT 3.x link.

different lanes, source synchronous operation is not possible. Instead, a *Clock Data Recovery (CDR)* mechanism is used to sample the individual CAD lines independently. The HT protocol requires all endpoints to initialize the link at 200 MHz in Gen1 mode. After a negotiation phase both ends can then ramp up their frequencies to Gen3 speeds.

It can be seen that the electrical and protocol specific demands imposed by HT3 are manifold. Both low and high speed operation have to be supported as well as different mechanisms for sampling the incoming data. Regular IO cells do not meet the performance requirements of HT3. Since the implementation presented here is based on Xilinx Virtex5 FPGAs it is mandatory to use the *GTP* transceiver logic they contain. These blocks allow for communication in the multi gigabit range, however are not suited for the HyperTransport technology originally. A careful design space analysis is required to determine whether GTP blocks can be deployed for HT3 compliant signalling.

3.1 GTP IO Cell Analysis

Using the GTP IO cells leads to some major problems as outlined in the following. First, there is the rather difficult setup in *Hardware Description Language (HDL)* code, despite the presence of software wizards, especially if numerous link rates are to be supported. Second, the limited number and fixed location of the GTP elements within the FPGA together with the predefined HTX pinout makes PCB routing difficult and inflexible which affects overall signal integrity. Third, during initialization HT is a source synchronous transmission standard with a forwarded clock, hence not natively supported by the protocol engine of the GTP logic. Fourth, an HT link starts up in HT1 mode with 400 Mbps link speed and is switched to higher frequencies, especially those of HT3, after successful initialization only. The GTP interface, therefore, needs to support at least one frequency switch for an HT3 interface to be enabled. And fifth, electrical compatibility of the CML transceivers needs to be validated for in most cases an LVDS like signaling scheme with common modes far above those of the HT standard are used.

3.2 Electrical compatibility

The CML transceivers of Xilinx Virtex5 GTPs operate in the 0 to 1.2 Volt range. The HT3 specification for transmitters and receivers operating at 2.4 Gbps [6] can mostly be fulfilled with ease, especially regarding switching speeds, rise and fall times, input sensitivities, voltage swing and post-cursor deemphasis.

GTPs mainly operate in AC coupled mode on systems with separate ground reference planes. The respective devices' input and output common mode voltages are of no concern in this case. With the definition of the HT3 standard an AC mode was introduced. To our best knowledge, however, all of today's AMD Opteron CPUs implementing the HT3 standard do not support this mode of operation. Therefore, the first step is to simulate GTP HSPICE models in conjunction with IBIS models of the Opteron HT3 transmitters and receivers to verify the general functionality of such a DC mode arrangement.

Various simulations were carried out with varying GTP power supply. For optimum performance it is desirable to operate the GTPs in the median of the tolerable power supply range. Simulations showed successful operation throughout the entire power supply sweeping range, hence Virtex 5 GTPs can indeed be used within the desired specification margins.

3.3 Source synchronous standard

Virtex5 GTPs provide a wealth of supporting logic for various different standards including PCI Express, SONET and Gigabit Ethernet. These standards all work with independent clock data recovery on each data lane and utilize clock correction and channel bonding by means of special characters found within the standard's protocol. Due to its legacy, HyperTransport cannot entirely benefit from these extras. All HT devices, including HT3 devices, start up in the slowest HT2 compliant mode. This mode supports data rates of 400 Mbps and defines data validity only with respect to a forwarded clock. Two problems need to be solved: First, all GTP receivers must output their data to a common clock domain whilst providing data sampled by the individual CDR circuits at the same point in time. Second, all transmitters inside a *Clock Forward Group (CFG)* which are the eight control and data (CAD), one control (CTL) and one clock (CLK) lane, need to be synchronized. Additionally, the source clock generated by a transmitter must be 90° phase shifted with respect to the data lanes.

The HTX interface for PC daughtercards provides a 200 MHz reference clock which the Opteron system uses for serializing data on the HT link. This clock is suitable to serve as the GTP *Phase Locked Loop (PLL)* reference clock input. One of the GTP dual tiles can then be used to provide its recovered clock from the CDR to a *Digital Clock Manager (DCM)* which in turn generates the required user clock for the receiving domain *(RXUSRCLK)* sources for all tiles. Figure 2 shows this arrangement. In this way a single RX domain is generated. The elastic buffer structures contained in every GTP receiver make up for the phase differences in every RX channel's *Physical Medium Attachment (PMA)* structure which are CDR dependent and the *Physical Coding Sublayer (PCS)* structures which are in the RXUSRCLK dependent domain.

While this is not the standard procedure for a true source synchronous interface which normally utilizes the forwarded clock to sample all data lanes at the same time, it is much simpler than distributing the CFG clock to all dual tiles and run the CDRs in lock to reference mode. The oversampling mode of GTP transceivers usually associated with low 100 Mbps operation cannot be used since it is impossible to activate it on the receiving and deactivate it on the transmitting side (see below). Instead, the GTP receivers operate at 800 Mbps lane rate and present a 16 Bit wide interface to the FPGA fabric. From the resulting bit vector, only every second bit is passed on. This manual oversampling lets the receiver act as it was in 400 Mbit mode.

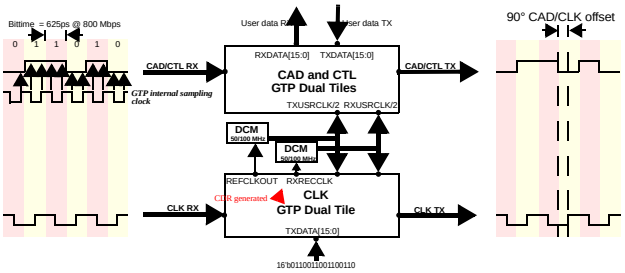

Figure 2. GTP-based Interface Block Diagram

The transmitters, too, operate at 800 Mbps with a 16 bit fabric interface on which every bit of a transmitted byte is replicated once. Xilinx provides a tx alignment procedure to minimize the output skew among the transmitters. The procedure only works with the GTP oversampling circuitry deactivated. Furthermore, all GTP transmission channels need to possess the same TXUSRCLK source thus again creating a single TX domain. A 90° phase shift on the clock lane with respect to the data lanes can be achieved by merely applying the correct static bit pattern to the GTP channel's fabric interface that transmits the clock. Using TX phase alignment also requires bypassing the tx elastic buffer which reduces overall latency for the outgoing traffic. The RX elastic buffer cannot be bypassed as this requires a GTP internal data width of 10 bits. This is incompatible with the 16 bit fabric interface. The latency for incoming and outgoing data will therefore be unequal. The RXUSRCLK source is provided by the GTP Dual Tile connecting to the HT CLKOUT trace. This is the preferred solution as the HT link clock has to be stable before data is transmitted over the link.

When running at the lowest HT3 frequency of 2.4 Gbps, the interface width of both receiving and transmitting channels need to be set to eight bit width. Due to the requirements of the GTP PLLs, the reference clock for all GTPs needs to be switched to a 300 MHz source. Phase alignment on the transmission side is no longer required as the HT3 protocol initialization takes care of lane to lane skews. Also, the training pattern phases guarantee enough time to establish a CDR lock on every lane. Furthermore, the HT3 scrambling mechanism introduces enough volatility into the data stream even in phases of an idling transmission lane. On the receiving side, except for the adoption of the lane rate and the interface width, no further changes are required. An adoption of the TXUSRCLK/RXUSRCLK DCMs is mandatory due to the changed frequency. Therefore, the *Dynamic Reconfiguration Port (DRP)* of the GTPs has to be used which allows reprogramming of specific settings at runtime.

3.4 Clock sources and interface frequency switch

After a successful HT link initialization at 400 Mbps, switching the interface to an HT3 frequency requires readjustment of the GTP dual tiles in several ways. The optimal reference clock frequency at 400 Mbps operation (HT200) is 200 MHz while at 2.4 Gbps operation a 300 MHz source clock is required for best PLL performance. As the 200 MHz in HT2 mode is provided by the HTX connector, an external clock multiplexer may be used to perform the switching operation. As an addition to this, one can rely on the internal GTP clock muxing network to derive GTP PLL clock references. A major advantage of this approach is also the ability to test the interface with entirely different clocking schemes as a Virtex5 LXT50 possesses six reference clock inputs.

Depending on the scheme used, switching the reference clock requires reprogramming the internal GTP dual tile muxing structure via the DRP port. This DRP reconfiguration step is also required to adapt important GTP internal settings for the PLL to be in optimal operation limits. Furthermore, the TX and RX channel specific clock dividers need to be changed and the interface width decreased to eight bit. As an option, post-cursor de-emphasis and RX equalization may also be enabled.

HyperTransport requires a warm reset for switching the operating frequency. In prior, the system has to access the HT unit to reprogram its internal configuration registers which then is followed up by a re-initialization of the link. This scheme allows for more than enough time to ramp up the GTP internal PLLs and the DCMs that depend on their stable clocks.

3.5 Basic Architecture

Having defined the requirements of the HyperTransport links a basic architecture can be developed. For HT, 20 high speed transceivers are required and as the board is planned to be used in networking applications, additional high speed links for their interconnects are needed. For maximum flexibility it is decided to provide two CX-4

connectors on the board which are the standard connectors for 10G Ethernet and Infiniband. The CX-4 connectors each implement four links which leads in combination with the HyperTransport interface to a total amount of 28 transceivers. Such multi gigabit transceivers are offered by all of the major FPGA vendors and incorporate serializers/deserializers, 8b/10 encoding, comma detection, CDR and eye tracking mechanisms.

Besides the number of transceivers, the maximum achievable clock frequency of the chip's internal logic is critical. For an HT3 core operating at the minimal link speed of 2.4Gbit per lane, an internal data bus of 128 bit running at a clock frequency of 300 MHz is required. Achieving timing closure for FPGA designs that run at 300 MHz is extremely challenging, especially for timing critical modules like the *Cyclic Redundancy Check (CRC)* calculation. Choosing the fastest available device is, therefore, mandatory. Currently, best performance is provided by the Virtex5 LXT devices of speedgrade 3 by Xilinx.

At the time of developing this board no FPGA of this type with 28 transceivers was available, which leads to the conclusion of crafting a multi-FPGA board. Three FPGAs are implemented, offering a combined amount of 40 multi gigabit transceivers and 210.000 look up tables (LUTs). This setup offers the best perspective of reaching timing closure for the HT3 core and provides the maximum flexibility.

4. Evaluation

The correct functionality of the board was evaluated using two different approaches. First, an IBIS model based simulation environment was set up to proof the electrical compliance of the board to the HyperTransport protocol at all supported line rates. Second a complex, real world test design was developed which uses the board inside a commodity computer to boot up a system that is able to communicate with the verification platform.

4.1 PCB Interface Simulation

Considering the fixed location of the dual tiles and the fixed HTX signal layout, routing the interface while maintaining good signal integrity requires advanced simulation techniques. The Cadence PCB SI GXL environment provides a toolset to integrate the above mentioned HSPICE and IBIS models into the PCB layout process for means of channel simulation. In order to account for the frequency attenuation and voltage dampening of the motherboard traces, a channel of 9.25 inches is assumed on the system side. Furthermore, a connector model is integrated to account for the losses of the HTX socket.

Figure 3. Assembled Printed Circuit Board

The performed trace and via extraction process along with the simulation gives assurance of post production functionality and ensures that the design stays within the HT3 specifications. Figure 4 shows the simulation result for 2.4 Gbps at the Opteron buffer input pin of a lane within the Clock Forward Group requiring the longest channel on the daughtercard design. The horizontal eye opening of 363 ps and the vertical opening of 285 mV show a considerable improvement over the measurements of the regular IOs shown in figure 3. These results also comply with the HyperTransport 3.x specification which requires a minimum eye width of 228ps and an eye height of 140mV that shows the successful layout of the PCB regarding signal integrity of the GTP traces.

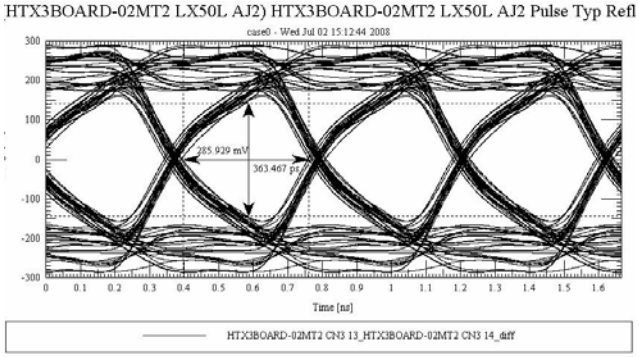

Figure 4. CTL lane post extraction simulation

4.2 In System Verification

The final proof of our verification platform is carried out in a real world system including all required hardware and software components. As a starting point for the FPGA logic, the HT-Core [10] has been used which is completely available as open source in Verilog. The core is capable to directly interface an Opteron processor via an HTX interface and has been mapped to various FPGA technologies. To support our specific verification platform several modifications had to be applied to the core. In particular, the physical layer interface had to be rewritten to support the high speed serial transceivers on the HTX interface. The

completed design was then loaded on the three FPGAs and the PCB was plugged into an actual system running Linux OS. A driver has been developed to provide accessibility of the card from application software. A small test routine has been implemented which writes to and reads from the device. An eye-diagram of a single lane has been recorded with a scope while accessing the device which is shown in Figure 5. The communication between software and the board as well as the excellent eye diagrams prove successful operation of the design.

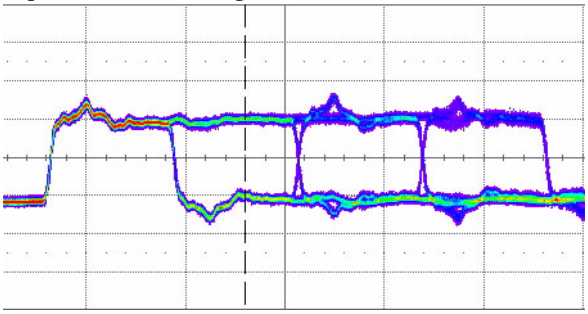

Figure 5. Eye Diagram of the in system HT link

5. Conclusion and Outlook

To our best knowledge, this paper presents the first fully functional HT3 verification platform. A design space exploration has been presented which analyzes different FPGA technologies and compliance to the HT3 protocol. To ensure quality of the design, simulations have been performed which led to a final architecture that has been implemented in the form of a prototype. The ported HT2-Core has proved correct functionality of the verification platform.

The main purpose of the presented design is the verification of HT3. Therefore, our focus will be to finalize the implementation of the core and to partition it for the three FPGAs. This configuration can then be used in a HTX3 capable mainboard as a verification platform. Apart from that, such a design represents a powerful platform for application use. The theoretical, bidirectional, peak bandwidth is 9.6 GByte/s which enables the board to be used in networking or coprocessing applications that require excessive throughput.

6. References

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

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

[3] Keltcher, C.N., McGrath, K.J., Ahmed, A., and Conway, P. 2003. The AMD Opteron processor for multiprocessor servers. IEEE Micro, 23(2):66-67.

[4] Litz H., Fröning, H., Nüssle, M., and Brüning, U. 2008. VELO: A Novel Communication Engine for Ultra-low Latency Message Transfers. In Proceedings of the 37th International Conference on Parallel Processing (ICPP-08), Portland, Oregon, USA.

[5] Conway, and P., Hughes, B. 2007. The AMD Opteron Northbridge Architecture. IEEE Micro, 27(2):10-21.

[6] Hypertransport Consortium. 2008. HyperTransport™ I/O Link Specification Revision 3.10. http://www.hypertransport.org.

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

[8] Nüssle, M., Fröning, H., Giese, A., Litz, H., Slogsnat, D., and Brüning, U. 2007. A Hypertransport based low-latency reconfigurable testbed for message-passing developments. In Proceedings of 2.Workshop Kommunikation in Clusterrechnern und Clusterverbundsystemen (KiCC'07), Chemnitz, Germany.

[9] Fröning, H., Nüssle, M., Slogsnat, D., Litz, H., and Brüning, U. 2006. The HTX-Board: A Rapid Prototyping Station. In Proceedings of the 3rd annual FPGAworld Conference, Stockholm, Sweden.

[10] Slogsnat, D., Giese, A., and U. Brüning. 2007. A versatile, low latency HyperTransport core. In Proceedings of the 15th ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, Monterey, California, USA.

[11] Lattice. 2008. SC/M Family Datasheet v0.20.

[12] Chen, W., Iancu, C., and Yelick, K. 2005. Communication Optimizations for Fine-Grained UPC Applications". In Proceedings of the 14th International Conf. on Parallel Architectures and Compilation Techniques, Washington, DC, USA.

[13] Gothandaraman, A., Peterson, G. D., Warren, G. L., Hinde, R. J., and Harrison, R. J. 2008. FPGA acceleration of a quantum Monte Carlo application. Parallel Computing 34, 4-5 (May 2008), 278-291.

[14] Kelm, J. H., Gelado, I., Murphy, M. J., Navarro, N., Lumetta, S., and Hwu, W. 2007. CIGAR: Application Partitioning for a CPU/Coprocessor Architecture. In Proceedings of the 16th international Conference on Parallel Architecture and Compilation Techniques (PACT).

[15] Zhuo, L. and Prasanna, V. K. 2005. High Performance Linear Algebra Operations on Reconfigurable Systems. In Proceedings of the 2005 ACM/IEEE Conference on Supercomputing.

[16] El-Ghazawi, T., Carlson, W., Sterling, T., and K. Yelick. 2005. UPC: Distributed Sharend Memory Programming. Wiley Series on Parallel and Distributed Computing. John Wiley and Sons, Inc., Hoboken, NJ.

[17] Sheng Li. Kuntz, S., Kogge, P., and Brockman, J. 2008. Memory model effects on application performance for a lightweight multithreaded architecture. In Proceedings of IEEE International Symposium on Parallel and Distributed Processing (IPDPS 2008).