An FPGA-Based Emulation Platform For Optical-Enabled System

Wembo Shen, Xinxin Zhang, Qigang Wang, Shunyu Zhu, Xiangbin Wu, Prabhat Gupta, Dong Liu

An FPGA-Based Emulation Platform For Optical-Enabled System

Wenbo Shen, Xinxin Zhang, Qigang Wang, Shunyu Zhu, Xiangbin Wu, Prabhat Gupta, Dong Liu
Intel Corporation
Contact: {wenbo.shen, xinxin.zhang}@intel.com

***Abstract*—This paper introduces an FPGA-based emulation platform for optical-enabled computer system researches. As electrical links are reaching their physical limits, optical technologies begin increasingly to play a role in computer systems by enabling high speed IO, optical memory extension and system bus links. With its excellent latency, bandwidth, power and scalability performance, optical links will move steadily closer to the processor in the future. New design challenges and trade-offs must be considered at the system level to make a balanced and efficient system from both power and cost perspectives. With an IO-centric research methodology in mind, we try to explore the broad design space of optical-enabled computer systems by implementing an FPGA-based emulation platform as the research vehicle. On this platform, high speed IO device is connected to an FPGA-based IO controller through optical fiber and the IO controller talks to the FPGA-based synthesizable processor core via PCIe link. With the in-field-configuration characteristic of FPGA, it gives us the freedom to modify both the CPU part and the IO part of the emulated system. The first system implementation is up and running with the an example application at this stage.**

I. Introduction

Optical links have been playing a role at the metropolitan and wide area distance scales and is becoming a good alternative in LAN and cluster networks [1], [2]. Thanks to its excellent latency, bandwidth, power and scalability characteristics, system architects expect that optical links will approach the processing element even closer by enabling high speed IO, optical memory extension [3] and system bus links [4], [5], with IO system as the first focus [6]. Amdahl's balanced system law indicates that for a balanced system, a 1MB IO throughput needs a 1MIPS processing power. Accordingly, with the high data throughput [7] and low latency brought in by optical IO subsystem, we may need to revisit the design methodologies and architectures for the CPU, the optical IO subsystem and the interconnection between them together with the whole SW stack to make a balanced and efficient system. We don't see any processor with native optical IO links in mass production yet, so we think optical technologies tightly coupled with the computer system are still in exploration stage and there should be a broad design space to look into.

Both industrial and academia researchers are working around optical technologies and come up with new ideas and innovative technologies [8]. Each new idea and technology needs to be evaluated exhaustively before it's adopted seriously by industry. Simulation and emulation are two most commonly used approaches when architects evaluate new technologies. However, with the current trends of multi-core, system-on-chip (SoC), diversified IO devices with ever increasing complexities, both approaches are being challenged.

Simulation is flexible and cost effective, but it usually suffers from slow simulation speed and insufficient simulation model accuracy. FPGA emulation is seen as a solution to this problem. The flexibility, speed, high integration level, and capacity of modern FPGA qualify itself as an ideal choice for system level emulation [9]. Emulation can run orders of magnitude faster than simulation and is able to accurately model the full system. We adopt this approach to address our research purposes, with a goal to make the emulation platform flexible enough to explore the design space at both the processor side and the IO side.

In this paper, we will share our view of the system with optical IO links first, then we will present the abstract system model and our research methodology, which is followed by an FPGA-based research platform implementation. We will describe our first experiment done on this platform and conclude the paper by briefing the next research topics.

The key contributions of this paper are

- *IO-centric research methodology.* We think when optical links are introduced in the computer system with 100Gbps or even higher throughput, the fundamental design methodology and system architecture should be revisited. IO subsystem will be as important as the processing element and more innovations should be done at the IO side.
- *An FPGA-based emulation platform implementation.* To make the research platform as flexible as possible, we use FPGA as one of the building blocks of the platform. We implemented both the processor core and the IO controller on FPGAs to give us the flexibility to change them when needed.

II. Our view on optical-enabled system and research mothodology

In the next 5-10 years, people will have many reasons for higher bandwidth. Lots of attractive applications that significantly improve user experiences are depending on huge volume of data capturing, transfer, storage, and reconstruction. People will have more and more electrical devices, such as High Definition (HD) video camcorders, HD monitors, Mobile Internet Device (MID) s, laptops and other handheld devices, and they want to be able to share data between these devices

smoothly, quickly and easily. All these user requirements call for higher bandwidth. But existing electrical cable technology in mainstream computing devices is approaching the practical limit for higher bandwidth and longer distance, due to the signal degradation caused by electro-magnetic interference (EMI) and signal integrity issues. Higher bandwidth can be achieved by sending the signals down with more wires, but apparently this approach increases cost, power and difficulty of PCB layout, which explains why serial links such as SATA, SAS, USB are becoming the mainstream. However optical communications do not create EMI by using photonics rather than electrons, thus allowing higher bandwidth and longer distances. Besides, optical technology also allows for small form factors and longer, thinner cables.

It's important to design flexible and effecient protocol to leverage the raw bandwidth enabled by optical fiber. Intel has announced its high speed optical cable technology, Light Peak (LPK) [10], which delivers high bandwidth starting at 10Gbps and has the ability to multiplex multiple protocols simultaneously over a single optical cable.

Combining the high bandwidth of optical fiber with Intel's practice to mulplex multiple protocol over a single fiber, optical technology may change the landscape of IO system design in the future. It's possible that most of the legacy IO protocols can be tunneled by optical-capable protocols, so some of the legacy IO interfaces can be converged to one single optical interface, significantly simplifying the form factor design of computers. This change in IO system will definitely affect the design of systems. The ultimate goal of system architects is to make a balanced and efficient system, on both power and cost grounds. It makes no sense to have a high throughput IO system with insufficient processing power or overloaded interconnections between IO system and the processor.

Mobile and handheld devices are two fast growing market segments which attract interests from processor vendors including Intel and they are our targets for application selection. For mobile and handheld devices, user interface and IO are two important factors besides computing power that affect end users' purchase decision. Taking power into account, it's possible that more carefully tuned IO workload offloading engines will be integrated into the IO controller, saving the power to move the data from IO a long way to the system memory.

Because we think that the future computing devices, especially low-end and consumer electronics will be more IO-centric, our research philosophy is an IO-oriented one. We try to identify the future IO devices and their usage models first, then we investigate the requirements of the interconnection from the IO's standpoint, after that we move up to the processor level, trying to figure out the needs of processing power and thus the best architecture for it.

With this philosophy in mind, an optical-enabled system model can be illustrated in Fig. 1. There are four main components in this figure, the IO devices, the IO controller which connects to the IO devices through optical fiber, the processing unit and the interconnection between the IO controller and the processing unit, whatever it can be implemented as. We are looking at the system from IO to processor as shown by the arrow.

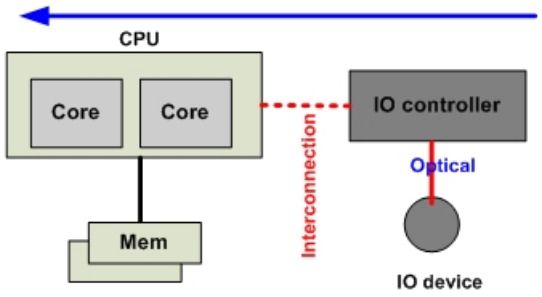

Fig. 1: Abstract model of the optical-enabled system

III. The Emulation Platform Implementation

We tried to map the abstract model dictated in Fig. 1 to a real implementation. FPGA is a popular prototyping platform and it's a natural choice for us to implement Intel Architecture (IA) cores and the IO controller since we want the full freedom to change the core and the IO controller when doing experiments.

As for the IO device, after evaluating a few options, we decided to take SATA HDD as the starting point because HDD is an important component in modern computer systems which has remarkable influence on the whole system performance if the storage traffic is heavy.

We spent quite a long time to evaluate the possible interconnection implementations between the processor and the IO system. Based on the facilities available in the current system, we have a few alternatives, such as Intel Front Side Bus (FSB), Intel QuickPath Interconnection (QPI), PCIe. In an ideal scenario, the IO system should be as close to the processor as possible, and it should meet the bandwidth and latency requirements. FSB or QPI may be a better solution than PCIe because of their advantages of latency and bandwidth as CPU bus between CPU and chipset, but in the real world, we have to balance the requirements with the development effort. FSB and QPI links are much challenging to develop, and PCIe is an evolving open IO standard with good bandwidth capacity and moderate development effort. We think PCIe is an appropriate candidate to begin with.

After all component selections are done, we get the full emulation platform implementation as illustrated in Fig. 2 (we keep a copy of Fig. 1 in this figure to make the mapping clear). Synthesizable IA cores are hosted by an FPGA module, which is plugged into the CPU socket, working as a real CPU to boot unmodified Linux distributions (Ubuntu, CentOS, OpenSuse) as well as WindowsXP and run applications. The SATA HDD connects to the IO controller via optical fiber on which the Intel Light Peak technology runs. The IO controller is implemented with FPGA also and is connected with the IA core through PCIe.

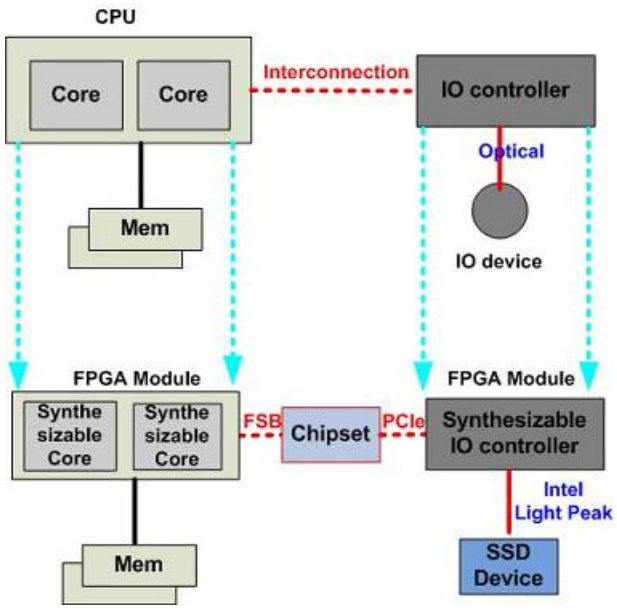

Fig. 2: Map the abstract model to the real implementation

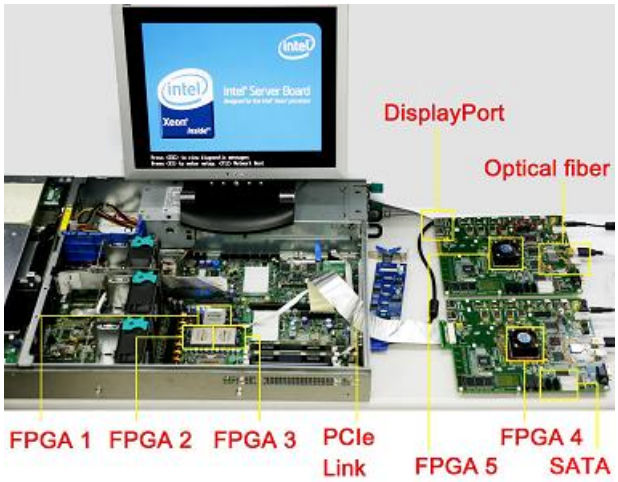

Fig. 3: Emulation platform implementation

Fig. 3 shows the picture of the real system when the system is booting BIOS. On the in-socket module which is hosted by a Xeon-based server, there are three Altera PFGA devices, marked as FPGA1, FPGA2 and FPGA3, with FPGA1 to implement the FSB protocol so that the IA core can talk with the chipset. The other two FPGAs, FPGA2 and FPGA3, are used to implement IA cores and we put two Pentium cores into them. The capacity of the FPGA2 and FPGA3 is big enough to accommodate other IP blocks. To make the Pentium core run on the Xeon based server, we modified the system BIOS and implemented some FSB features which are not present on a Pentium silicon.

The first processor core used in our platform is the sec ond generation Pentium. It is a 32bit in-order 5 stage dual pipeline processor supporting the IA32 instruction set in cluding floating point instructions using an pipelined floating point module. It is equipped with two separate level 1 caches for code and data and implements the MESI protocol for use in multi-processor environments. It also includes local Advanced Programmable Interrupt Controller (APIC) and a processor bus supporting dual processors. Although Pentium is not a modern processor design, the current trend of multi-core draws much attention back to such simple cores [11]. Atom is the next processor to be used in this platform. It is a two-way Hyper-Threading in-order pipeline processor supporting the complete Core 2 Duo compatible instruction set architectures including Streaming SIMD Extension (SSE) 3 and Intel64. It is built for low power devices and targets mobile and low-cost PC market segments.

IO board No.1 is connected to the motherboard through PCIe interface and an optical fiber links IO board No.1 and IO board No.2. There is an FPGA on each of the IO board. Since the protocol running on the optical fiber is Intel Light Peak, FPGA4 terminates PCIe, gets the payload data and transforms it into a data format which can be transmitted by Light Peak. FPGA5 reconstructs the payload data and delivers it to the IO device. We integrated many legacy IO ports on IO boards, including SATA, USB, Ethernet, DisplayPort (DP) [12] and so on.

This platform supports all synthesizable IA compatible processors and any optical IO device theoretically. As for the PCIe interconnection, we may use reserved fields in the protocol packets for any enhancements we want. The flexibility of FPGA allows one to make quick change to the system and evaluate the effect immediately, which offers a much shorter turnaround cycle time than software simulation. And, data collected with a real emulation platform is more convincing to architects.

To the best of our knowledge, there is no such an emulation platform for optical-enabled system as flexible to this level.

IV. The first experiment with the platform

Fig. 4 shows the main components of our first experiment. Some of them run on the host and some of them run on FPGA boards as illustrated by the figure. To make the system up and running is our initial goal, without any performance considerations yet. Essentially, we implemented SATA traffic over Light Peak protocol on optical fiber. At the software side, we implement a standard SATA driver which provides APIs to applications. Take HDD write as example, when application writes to SATA SSD, these APIs are responsible for preparing SATA Frame Information Structure (FIS) packets in memory , and call host controller driver to send these FIS packets to downstream device. On board 1, a Light Peak host controller is integrated in FPGA 4 and communicates with processor over PCIe link. The host controller gets the FIS packets from memory, encapsulate them into Light Peak packets, and sends them to board 2 over optical fiber. On board 2, a Light-Peak-to-SATA adapter is implemented in FPGA 5. The adapter resolves the Light Peak packets from board 1 and sends these FIS

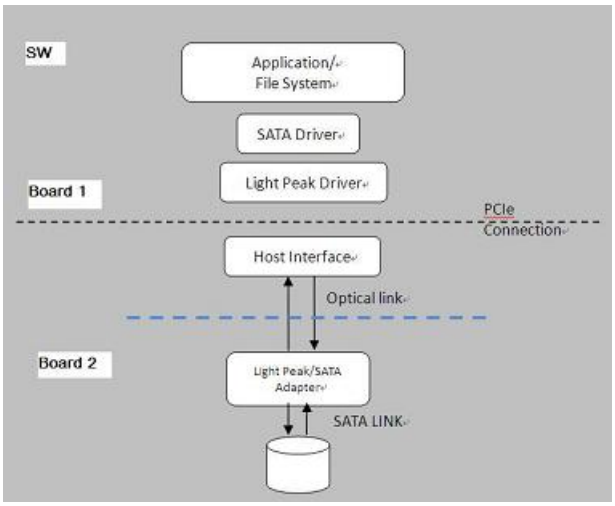

Fig. 4: Components of the SATA HDD experiment

packets to SATA SSD.

V. Conclusion and Future Work

We presented a view of the future computer system enabled by optical technologies and we built this FPGA-based emulation platform as the first implementation of the research vehicle. We are trying simple data copy on this platform now. The use of this platform greatly accelerates prototype implementation and testing. As the next step we plan to tunnel more legacy protocols, such as Ethernet, DisplayPort over the optical links, and study how to improve the system QoS, power management, processor utilization and system latency. We can find a more efficient SW/HW partitioning and a host interface with higher convergence level for the system architecture.

We will replace the PCIe interconnection with more aggressive ones such as Intel QPI or even direct optical link to the CPU package enabled by silicon photonics. Without the bottleneck of bandwidth introduced by PCIe link, we can analyze the impact to system architecture when optical IO bandwidth scales from 10Gbps to 100Gbps.

Acknowledgment

We would like to thank Bhushan Chitlur, Nelson Ijih, Arthur Sheiman, Rolf Kassa, Michael Konow, Zhiyuan Zhang, Lu Cao, Qianying Zhu, and XtremeData for their help and contribution to this project.

References

[1] Fibre Channel Basics. White paper of Apple Computer Inc.2006.

[2] 10 Gigabit Ethernet Technology Overview. White paper of Intel Corporation.2003.

[3] Y. Katayama and A. Okazaki. Optical Interconnect Opportunities for Future Server Memory Systems. In Proceedings of the 2007 IEEE 13th International Symposium on High Performance Computer Architecture,2007.

[4] M. Petracca, B. G. Lee, K. Bergman and L. P. Carloni. Design Exploration of Optical Interconnection Networks for Chip Multiprocessors. In Proceedings of 16th IEEE Symposium on High Performance Interconnects,2008.

[5] N. Kirman, M. Kirman, R. K. Dokania, J. F. Martinez, A. B. Apsel, M. A. Watkins and D. H. Albonesi. Leveraging Optical Technology in Future Bus-based Chip Multiprocessors. In Proceedings of the 39th Annual IEEE/ACM International Symposium on Microarchitecture,2006.

[6] A. F. Benner, M. Ignatowski, J. A. Kash, D. M. Kuchta, M. B. Ritter. Exploitation of optical interconnects in future server architectures. In Proceedings of the 2007 IEEE 13th international symposium on high performance computer architecture, 2007.

[7] http://www.ibm.com/press/us/en/pressrelease/29595.wss.

[8] B. E. Lemoff, M. E. Ali, G. Panotopoulos, G. M. Flower, B.Madhavan, A. F. J. Levi, and D. W. Dolfi. MAUI: Enabling Fiber-to-the-Processor with Parallel Multiwavelength Optical Interconnects. IEEE J. Lightwave Technol. 22, 2043C2054, 2004.

[9] P. H. Wang, J. D. Collins, C. T. Weaver, B. Kuttanna, S.Salamian, G. N. Chinya, E. Schuchman, O. Schiling, T. Doil,S. Steibl and H. Wang. Intel Atom processor core made FPGA-synthesizable. In FPGA, 2009.

[10] M.Paniccia, V.Krutul, R.Jones, O.Cohen, J.Bowers, A.Fang, H.Park. A Hybrid Silicon Laser: Silicon photonics technology for future tera-scale computing. Intel white paper, 2006.

[11] L. Seiler, D. Carmean, E. Sprangle, T. Forsyth, M. Abrash, P. Dubey, S. Junkins, A. Lake, J. Sugerman, R. Cavin, R. Espasa, E. Grochowski, T. Juan and P. Hanrahan. Larrabee: a many-core x86 architecture for visual computing. In ACM Transactions on Graphics, Vol.27, No.3, Article 18, 2008.

[12] http://www.displayport.org.