Photonics for HPEC: A Low-Powered Solution for High Bandwidth Applications

E. Robinson†, G. Hendry*, V. Gleyzer†, J. Chan*, L. P. Carloni‡, N. Bliss†, R. Bond†, K. Bergman*
*Lightwave Research Lab, Department of Electrical Engineering, Columbia University, New York, NY
† MIT Lincoln Laboratories, Lexington, MA
‡ Computer Science Department, Columbia University, New York, NY

Photonics offer high bandwidth for minimal power. While critical for the future of HPC, this has an immediate impact in HPEC, where power is critical. Here, a 4-10x improvement in performance/watt can be demonstrated.

Introduction

While Moore's law continues to provide increased processing power, that increase now requires a corresponding increase in energy that is no longer insignificant. This problem will have to be addressed by the high performance computing (HPC) community in general going forward. However, its impacts are being felt immediately in the high performance embedded computing (HPEC) community, where size, weight, and power (SWAP) constraints are typical on platforms such as unmanned air vehicles (UAV), satellites, etc. Here, key applications typically involve signal and image processing (SIP). There is an increasing demand to perform SIP applications both in real time, and within a tight power budget. These applications are growing rapidly, especially in the image domain, where processing requirements grow with the square of data collection ability.

One of the key difficulties in the SIP domain is supplying the cores in a multicore system with the appropriate data fast enough to keep up with the demand from those cores. While providing more channels to RAM, or more RAM chips in general, is one option, that option is typically a power-hungry one. RAM is commonly located off-chip, oftentimes far from the actual processing core. There are good reasons for this setup in terms of chip design and fabrication, however, it also leads to relatively long electrical connections that require a large amount of power to transmit data.

Fortunately, this problem is simplified by the fact that in a typical SIP application, data is only affected by data that is "nearby". For example, a pixel in an image may blend with adjacent pixels, but not typically pixels located halfway across the image. This type of data dependence can be exploited in an implementation to take advantage of streaming-only memory access, which is known to be more efficient than random access. However, even with this, the data must still travel long distances to arrive at the appropriate processors.

One potential solution to this problem is the use of photonic interconnects. Photonics offer a low-power, high bandwidth data transfer medium. Current photonic technology places a high cost both for fabrication as well as cost to set up a photonic channel. One can expect fabrication costs to drop as photonic devices become more popular. In addition, in domains where streaming access dominates, such as SIP, the large set up costs are subsumed by the large amount of data transferred at high bandwidths once the channel has been established.

In this paper, we examine the impact of photonics to SIP applications in particular. We simulate various functional blocks commonly found in SIP applications on a variety of potential hardware configurations. The performance metric of importance, as specified previously, is measured not in GOPS, but rather GOPS/W, taking into account the power required to produce the desired result. We show a performance win for the use of photonic interconnects, particularly to memory. Going forward, this will be of increasing importance not only to the SIP and HPEC community, but also to the HPC community as a whole, as power consumption becomes of increasing importance.

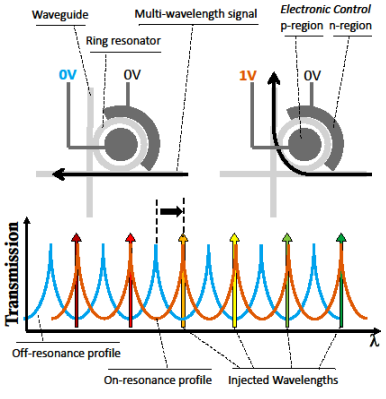

Figure 1 - Ring Resonator used as a PSE

Photonic Devices

Circuit-switching photonic networks can be achieved using active broad-band ring-resonators whose diameter is manufactured such that its resonant modes directly align with all of the wavelengths injected into the nearby waveguide. The ring resonator can be configured to be used as a photonic switching element (PSE), as shown in Figure 1. By electrically injecting carriers into the ring, the entire resonant profile is shifted, effectively creating a spatial switch between the ports of the device [1]. This process is analogous to setting the control signals of an electronic crossbar.

Our proposed circuit-switched memory access architecture requires slightly different usage of DRAM modules. Figure 2(a) shows the Photonic Circuit-Accessed Memory Module (P-CAMM) design. Individual conventional

* This work is sponsored by Defense Advanced Research Projects Agency (DARPA) under Air Force contract FA8721-05-C-0002. Opinions, interpretations, conclusions and recommendations are those of the author and are not necessarily endorsed by the United States Government.

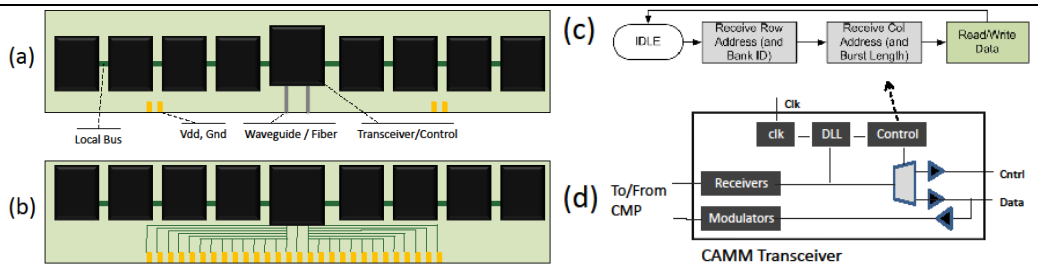

Figure 2 - Circuit-Accessed Memory Module (a) Photonic CAMM (b) Electronic CAMM (c) CAMM control (d) CAMM Transceiver.

DRAM chips are connected via a local electronic bus to a central optical controller/transceiver, shown in Figure 2(d). The controller, shown in Figure 2(c), is responsible for demultiplexing the single optical channel into the address and data bus much in the same way as Rambus RDRAM memory technology [2], using a simple control protocol. Figure 2(b) shows the anatomy of an Electronic Circuit-Accessed Memory Module (E-CAMM), similar to the P-CAMM in structure, but still requiring electronic pins as I/O.

Network Architectures

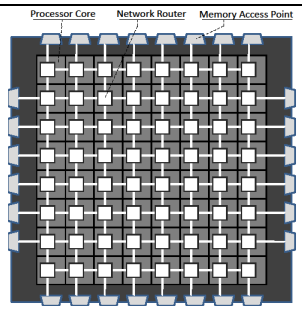

Figure 3 - 64 Node Network Topology

As part of this work, three types of network control protocols are considered, these protocols are described below.

Packet-switched. Packet-switched NoCs use router buffers to store and forward small *packets* through the network, where a packet is a small number of flits (flow control units). Typically, purely electronic store-and-forward routers use multiple physical buffers to implement virtual channels, alleviating head-of-line blocking under congestion. If a core-to-DRAM or core-to-core application-level *message* is larger than the physical buffers themselves, or larger than the flow control mechanism can reasonably sustain without deadlock, these messages must be broken into several smaller packets.

Circuit-switched. In a circuit-switched network, a control network provides a mechanism for setting up and tearing down energy-efficient high-bandwidth end-to-end circuit paths. If a network node wishes to send data to another node, a PATH-SETUP message is sent to reserve the necessary network resources to allocate the path. A PATH-BLOCKED message is returned to the node if some parts of the path is currently reserved by another circuit. A PATH-ACK message is returned if the path successfully made it to the end node. After data is transmitted along the data plane, a PATH-TEARDOWN message is sent from the sending node to release network resources for other paths.

Time division multiplexing. In a time division multiplexed network, direct paths between cores are set up similar to a circuit-switched network. However, in order to prevent blocking and maintain fairness, the time of the system is divided into communication windows where during each window, switches in the network are configured to allow communication between one or more pairs of access points. Due to the fact that the communication schedule is determined statically at compile time (unlike in many other TDM schemes), this greatly reduces communication setup costs. However, it also forces very large messages to be broken up across multiple sending windows.

Electronic packet-switched and circuit-switched networks, along with photonic circuit-switch and time division multiplexing networks are considered. They are run using a 64 node mesh topology shown in Figure 3.

Testing Framework

We evaluate the proposed network architectures using an application modeling framework, the *Lincoln Lab Mapping and Optimization Environment* (LLMOE), to collect traces from the execution of high-performance embedded signal and image processing applications.

The LLMOE system is designed to project a user program written in Matlab onto a distributed or parallel architecture and provide performance results and analysis. The LLMOE framework translates application code into a *dependency-based instruction trace*, which captures the individual operations performed as well as their interdependencies. By creating an instruction trace interface for PhoenixSim [3], a simulation environment which models both photonic and electronic network components, we were able to accurately model the execution of applications on the proposed architectures.

LLMOE consists of the following primary components:

- The *program analysis* component is responsible for converting the user program, taken as input, into a *parse graph*, a description of the high-level operations and their dependences on one another.
- The *data mapping* component is responsible for distributing the data of each variable specified in the user code across the processors in the architecture.
- The *operations analysis* component is responsible for taking the parse graph and data maps and forming the *dependency graph*, a description of the low-level operations and their dependences on one another.

PhoenixSim then reads the dependency graphs produced by LLMOE, generating computation and communication events. Combining PhoenixSim with LLMOE in this way allows us to characterize photonic networks on the physical level by generating traffic which exactly describes the communication, memory access, and computation of any application. Three applications are considered:

Projective Transform. When registering multiple images taken from various aerial surveillance platforms, it is frequently advantageous to change the perspective of these images so that they are all registered from a common angle and orientation (typically straight down with north being at the top of the image). In order to do this, a process known as projective transform is used [4].

Matrix Multiply. Matrix multiplication is a common operation in signal and image processing, where it can be used in filtering as well as to control hue, saturation and contrast in an image. It is a natural candidate for consideration on our architecture, given that multiple data points need to be accessed and then summed to form an single entry in the result.

Fast Fourier Transform. Computing the Fast Fourier Transform (FFT) of a set of data points is an essential algorithm which underlies many signal processing and scientific applications. In addition to the widespread use of the FFT, the inherent data parallelism that can be exploited in its computation makes it a good match for measuring the performance of networks-on-chip.

Simulation Results

	Projective Transform			Matrix Multiply			FFT		
Network	*Power (Watts)*	*Perf. (GOPS)*	*Impr. (GOPS/W)*	*Net. Pow. (Watts)*	*Perf. (GOPS)*	*Impr. (GOPS/W)*	*Power (Watts)*	*Perf. (GOPS)*	*Impr. (GOPS/W)*
Emesh	11.2	1.04	1x	11.1	0.78	1x	11.4	1.75	1x
EmeshCS	19.0	47.3	26.9x	15.8	31.82	29.01x	11.2	4.74	2.82x
PS-1	4.37	27.80	68.6x	4.35	26.51	87.64x	4.28	4.32	6.72x
PS-2	2.21	17.76	86.7x	2.17	13.48	89.33x	2.15	3.12	9.67x

Figure 4 - Results for a matrix of size 1024

The results of running the above three algorithms on a matrices of size 1024 are shown in **Error! Reference source not found.**. These are shown for the standard electronic packet switched mesh (Emesh), an electronic circuit switched mesh (EmeshCS), a high-provisioned photonic circuit switched solution, and a low-provisioned photonic circuit switched solution. The TDM results are ignored in this instance as they are outperformed by the circuit switched solution. In all cases, the photonic circuit switched solutions yields performance gains over electronic solutions. Photonics offers an 80x performance gain over electronic packet switched (except in FFT, where this is only 10x), and a 3x performance gain over electronic circuit switch architectures.

Network	Net. Pow. (Watts)	Perf. (GOPS)	Impr. (GOPS/W)
Emesh	11.22	1.11	1x
PS-2	15.49	7.55	4.95x
P-TDM	16.02	20.87	13.22x
P-ETDM	23.97	51.04	21.61x

Figure 5 - Results for a 256x256 projective transform

In order to see the benefits of the TDM, an intermediate problem size must be chosen. Performance results in Figure 5 show that both a low-powered and high-powered photonic TDM network design outperforms an electronic packet switched network (by a factor of 20x) and also the best performing photonic circuit switched network (by a factor of 4x).

References

[1] W. Wolf. Embedded computer architectures in the mpsoc age. In WCAE '05: Proceedings of the 2005 workshop on Computer architecture education, page 1, New York, NY, USA, 2005. ACM.

[2] Rambus. RDRAM Memory Technology. http://www.rambus.com/us/products/rdram/index.html.

[3] J. Chan, G. Hendry, A. Biberman, K. Bergman, and L. P. Carloni. Phoenixsim: A simulator for physical-layer analysis of chip-scale photonic interconnection networks. In DATE: Design, Automation, and Test in Europe., Mar. 2010.

[4] H. Kim, E. Rutledge, S. Sacco, S. Mohindra, M. Marzilli, J. Kepner, R. Haney, J. Daly, and N. Bliss. PVTOL: Providing Productivity, Performance and Portability to DoD Signal Processing Applications on Multicore Processors. HPCMP-UGC '08: Proceedings of the 2008 DoD HPCMP Users Group Conference, pages 327–333. IEEE Computer Society.