Photonic Switched Optically Connected Memory: An Approach to Address Memory Challenges in Deep Learning

Ziyi Zhu, *Student Member, IEEE*, Giuseppe Di Guglielmo, *Member, IEEE*, Qixiang Cheng, *Member, IEEE*, Madeleine Glick, *Senior Member, IEEE, Senior Member, OSA*, Jihye Kwon, *Student Member, IEEE*, Hang Guan, Luca P. Carloni, *Fellow, IEEE*, and Keren Bergman, *Fellow, IEEE, Fellow, OSA*

(Invited paper)

***Abstract*—Deep learning has been revolutionizing many aspects of our society, powering various fields including computer vision, natural language processing, and activity recognition. However, the scaling trends for both datasets and model size are constraining system performance. Variability of memory requirements can lead to poor resource utilization. Reconfigurable photonic interconnects provide scalable solutions and enable efficient use of disaggregated memory resources. We propose a photonic switched optically connected memory system architecture that tackles the memory challenges while showing the functionality of optical switching for deep learning models. Our proposed system architecture utilizes a "lite" (de)serialization scheme for memory transfers via optical links to avoid network overheads and supports the dynamic allocation of remote memories to local processing systems. In order to test the feasibility of our proposal, we built an experimental testbed with a processing system and two remote memory nodes using silicon photonic switch fabrics and evaluated the system performance. The optical switching time is measured to be 119 μs and an overall 2.78 ms latency is achieved for the end-to-end reconfiguration. The collective results and existing high-bandwidth optical I/Os show the potential of integrating the photonic switched optically connected memory to state-of-the-art processing systems.**

***Index Terms*— Deep learning, memory architecture, optical switches, silicon photonics**

I. Introduction

Deep learning is a branch of machine learning that has drastically improved the state-of-the-art in many applications that enhance our daily lives and impact various aspects of our society. The computational models used in deep learning, called deep neural networks (DNNs), have been successfully applied to various fields including image classification [1], language processing [2], and activity recognition [3]. The DNNs consist of many processing layers whose computation is mainly defined by weights and biases. These weights and biases, called parameters of the DNNs, are learned during the training and used for the inference. Accelerators, such as graphics processing units (GPUs) and field programmable gate arrays (FPGAs), are used for accelerating these training and inference processes [4], [5]. Large convolutional neural network (CNN) architectures, such as VGG16 [1], ResNet152 [6], and NASNetLarge [7], contain millions of parameters and can require tens of gigabytes (GBs) of memory during the training phase for image classification applications [8]. More complicated deep learning architectures for image captioning [9] and video analysis [10] with recurrent neural networks (RNNs), can exacerbate the situation by requiring larger model and large-scale dataset size [11]. For inference, large embedding tables [12] in deep learning recommendation models can also easily exceed tens of GBs. Recent studies [13], [14], however, indicate that the deep learning datasets and models are continuously scaling, which will inevitably exceed the memory capacity in today's systems and limit the performance of deep learning applications.

While the maximum memory requirement keeps growing the real-time memory usage is application dependent and often requires on-demand solutions. First, different deep learning applications show varying memory requirements based on their architectures (for example CNNs, RNNs, CNNs+RNNs, and etc.). Second, the memory capacity requirement for various batch size [15] and optimization strategies [16] can change within a large range, but the method requiring a larger memory size does not always guarantee a better system performance [17]. Lastly, the size of embeddings that are used in recommendation applications is dependent on the entry size and

This work was supported in part by the Advanced Research Projects Agency Energy (ARPA-E) under the Enlightened Project under Grant DE-AR0000843, in part by the U.S. Department of Energy (DOE) SBIR/STTR Program under Photonic-Storage Subsystem Input/Output (P-SSIO) Interface Project under Grant FPHOTO S7146-01, and in part by the program "Energy-Efficient Computing: from Devices to Architectures (E2CDA)" (A#: 1640108), a joint initiative between the National Science Foundation and the Semiconductor Research Corporation.

Ziyi Zhu, Qixiang Cheng, Madeleine Glick, and Keren Bergman are with Department of Electrical Engineering, Columbia University, New York, NY 10027, USA. (e-mail: zz2374@columbia.edu; qc2228@columbia.edu; msg144@columbia.edu; bergman@ee.columbia.edu).

Giuseppe Di Guglielmo, Jihye Kown, and Luca P. Carloni are with Department of Computer Science, Columbia University, New York, NY 10027, USA. (e-mail: giuseppe.diguglielmo@columbia.edu; jihyekwon@cs.columbia.edu; luca@cs.columbia.edu).

Hang Guan is with Elenion Technologies LLC, 171 Madison Ave. STE 1100, New York, NY 10016, USA. (e-mail: hang.guan@elenion.com).

number of models [18]. Having fixed and preconfigured amount of memory in the local system for the maximum memory capacity requirement is inefficient and will become more so. A scalable and dynamic solution is required to address the memory challenges for future deep learning applications.

Several approaches to tackle the memory capacity issue for large DNNs have been explored. Virtualizing the memory usage of DNNs such that both host and device memory can be utilized by a careful study on the data dependency and network topology of the DNNs is proposed in [8]. Parallelizing deep learning models across multiple GPUs can be another approach: data parallelism and model parallelism algorithms presented in [19] show how to distribute large networks among GPUs to relieve the memory capacity limitation. To reduce the communication overhead and achieve better resource utilization, in [20] a memory-centric architecture is demonstrated in simulation and proposed for future high-performance computing systems. Memory modules are aggregated locally and connected with device nodes using NVLink. Ref.[21] proposed using non-volatile memory (NVM) for storing embeddings in deep learning models with caching data in volatile memory to relieve the constraints. The first three approaches tackling the memory capacity issue with preconfigured and fixed memory resources do not provide a scalable solution to the on-demand memory requirement while the last approach can still be limited by the NVM bandwidth.

Photonic interconnects can enable disaggregated high-bandwidth networks reconfiguring compute and memory resources to meet application requirements in a more efficient and scalable network [22] than those using fixed resource configurations. Memory resources can be pooled and connected to other resources using reconfigurable optical switch fabrics [23]. The system can then be adaptively configured, according to dynamic resource requirements of deep learning applications, to achieve high resource utilization and deliver required system performance. Optically connected memory technique has been demonstrated using custom network interface card [24] with the inevitable overheads in memory-to-network conversions [20]. An optically connected system with emulated processors and a custom memory controller has been reported in [25], [26] without an end-to-end program-level demonstration.

In this work, we investigate the feasibility of integrating photonic switched optically connected memory into processing systems to address memory challenges in deep learning. The proposed system architecture enables on-demand allocation of additional memory to processing systems with a constant reconfiguration time that is independent of the required memory size. A “lite” (de)serialization scheme, compatible with standard memory interface protocol, is proposed to eliminate the communication overheads and is applied to demonstrate memory transfers between the processing system and remote memory nodes at program-level. We built a testbed with a processing system node and two remote memory nodes to evaluate the system performance with memory read/write operations. This testbed experimentally demonstrates an end-to-end reconfiguration latency of 2.78 ms and showed a step towards deploying photonic interconnects and optically connected memory for deep learning. Compared to the latency introduced by using storage devices for the DNNs, the proposed system achieves a significant speedup with remote memories.

The remainder of the paper is organized as follows: Section II describes the system architecture and implementation details; Section III presents the testbed we built to evaluate the system performance. Section IV shows the experimental results. In Section V, we discuss optical switch requirements, limitations of our testbed and technologies to further improve the system performance. Lastly, the paper concludes in Section VI.

II. System Architecture

Figure 1A left depicts the traditional system architecture. Each processing system is composed of CPU, memory, storage, accelerator, and network resources. In order to achieve better accuracy, larger datasets and more complex larger models are being used [13]. Adding more fixed memory modules to the processing system or to the accelerator for large DNNs is not an indefinitely scalable solution that will meet the scaling requirements. Furthermore, incorporating new more advanced hardware with fixed resources cannot guarantee an efficient

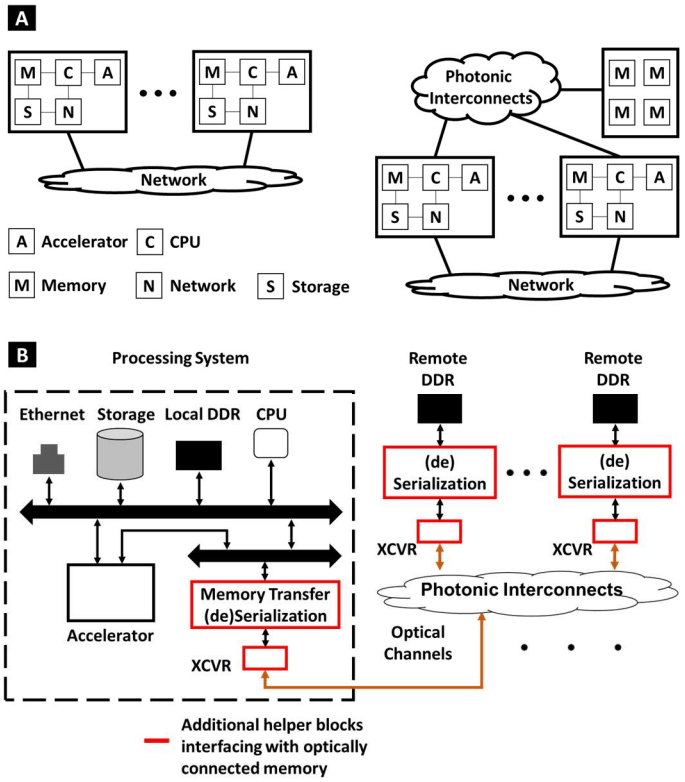

Fig. 1. (A) On the left, the traditional system architecture with each processing system composed of preconfigured and fixed CPU, memory, storage, accelerator and network resources. In our proposed system architecture, on the right, each processing system using optical I/Os is also connected to a remote memory pool through photonic interconnects. (B) Detailed implementation of photonic switched system architecture with optically connected memory. The processing system includes additional (de)serialization and transceiver (XCVR) helper blocks for (de)serializing memory mapped transactions being transmitted through optical links. On the right, remote double date rate synchronous dynamic random-access memory (DDR) nodes, are also equipped with the (de)serialization and XCVR helper blocks, and the photonic interconnects physically connect remote memory nodes to the processing system.

utilization of compute and memory resources, as the memory capacity requirement for DNN models can vary significantly with applications [1], [10], [15], [18], [27]. We note, therefore, the traditional system architecture suffers from for deep learning applications is facing scaling and resource utilization challenges. In our proposed system architecture, as shown in Fig. 1A right, the reconfigurable photonic interconnects enable decoupling of additional memory modules from the processing systems and therefore enable flexible allocation of the additional memory capacity to systems or accelerators as required or on-demand. This system architecture breaks through the memory capacity limitation, improves the resource utilization, and is compatible with existing processing systems using designated (de)serialization and memory mapping schemes. Figure 1B shows more details of our proposed photonic switched optically connected memory system architecture. The processing system on the left is initially equipped with CPU, memory, accelerator, network, and storage resources. Based upon the memory capacity requirement of the deep learning applications, additional remote memory resources can be connected to the processing system using photonic interconnects through high-speed serial optical links. Helper blocks directly (de)serialize memory requests avoiding potential overheads introduced by network protocols and the NVMs.

Disaggregated memory blocks can be assigned to the processing system using reconfigurable photonic interconnects for two cases. In the first case a processing system occupies the required memory blocks until it finishes the usage of the additional memory capacity. In this case, additional remote memory blocks can be assigned solely to that processing system. The second case occurs when multiple processing systems share remote memory nodes. This case depends on the fast switching capability of the photonic interconnects. Remote memory nodes can thus be dynamically selected while applications are running. The optical switching also enables the processing system to access remote memory nodes with limited optical transceiver ports. Examples of these two cases can be found in the following subsection B. In addition, the proposed system architecture can be integrated to current systems with minor modifications to current operating systems.

In this work, we use Xilinx multiprocessor system-on-chip (MPSoC) devices to demonstrate the feasibility of integrating photonic switched optically connected memory into the processing system. Detailed system implementations: (A) a "lite" (de)serialization of memory transfers; (B) mapping remote DDR into the system address space; (C) Silicon Photonic (SiP) switch and control; and (D) accelerator design are presented in the subsections below.

A. *(de)Serialization of memory transfers*

The MPSoC system uses the AMBA AXI protocol [28] to perform memory read/write operations. To access a locally memory mapped slave device, master devices such as CPU and accelerators can simply launch requests through transaction channels, such as read address, read data, write address, write data, and write response, in order to finish the memory transactions. To access an optically connected remote memory slave, however, the AXI memory mapped channel signals have to be combined and serialized before being transmitted to the remote side through high-speed serial links. We leveraged existing IP blocks designed by Xilinx to achieve the "lite" (de)serialization of the remote memory transfers. Without using any network layer protocol, our scheme directly serializes the AXI channel signals and transfers the high-speed serial signals to the remote nodes through optical links. On the receiver side, the high-speed serial signals are deserialized back to the parallel AXI channel signals.

We primarily used two IP blocks, AXI chip2chip [29] and Aurora 64B/66B [30] IP cores in this system design. The AXI chip2chip core converts the AXI memory mapped channel signals into AXI streaming signals or vice versa and interfaces to the Aurora 64B/66B core. The latter core utilizes a link-layer protocol, including transceiver initialization, multi-lane handling, and link negotiation for the high-speed serial communication between our optically connected nodes. The AXI chip2chip core can be connected to the AXI interconnects that can be consequently accessed by CPU and accelerators. To achieve an error-free operation, specific transceiver control settings are necessary to be properly configured. These settings depend on the link characteristics. Further details are shown in Section IV.

B. *Map to Local System Address Space*

The master CPU and accelerators can only see and

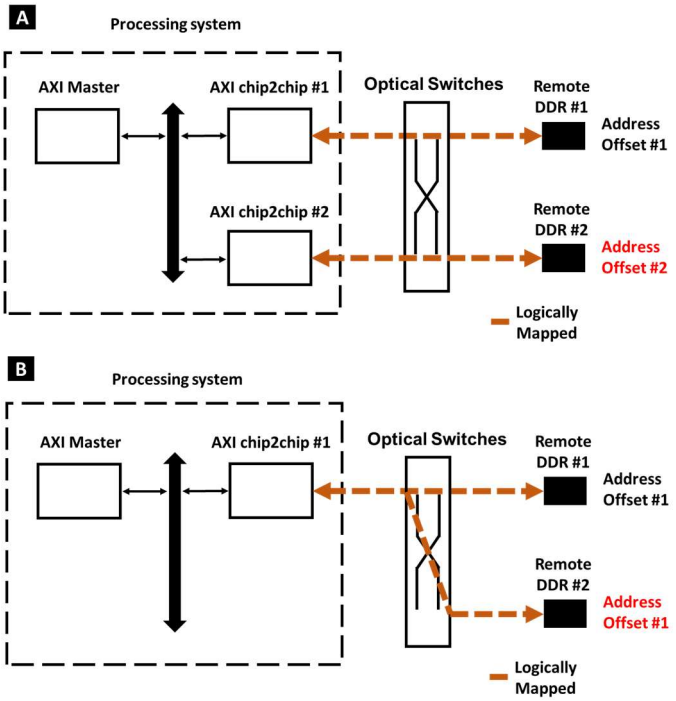

Fig. 2. (A) An example of case 1, two remote memory resources mapped to two AXI chip2chip cores in the local processing system for the unswitched case after the resources are assigned. Each chip2chip core is assigned with a unique memory address offset (B) An example of case 2, the switching case. Both remote DDR #1 and remote DDR #2 are mapped to the AXI chip2chip #1 in the processing system. They share the same memory address offset.

communicate with the AXI chip2chip IP blocks in the processing system. In fact, the AXI chip2chip core exposes the remote DDR slave to the local system space. Memory address offsets of the AXI chip2chip and the remote DDR are set to be the same. In this way, CPU and accelerators can seamlessly access the remote DDR as a "local" device. For the case where the processing system occupies multiple memory blocks without optical switching during the application, the remote memory blocks are assigned with different memory address offsets (as they are connected to different chip2chip cores). An example of this case is shown in Fig. 2A. Remote DDR #1 node is projected by chip2chip #1 and remote DDR #2 is projected by chip2chip #2. Two remote DDR nodes have different address offset values because they are mapped to separate chip2chip cores. However, for the switching case, the memory address offset of all the remote DDRs is set to be the same. This is due to the fact that CPU and accelerators are accessing the remote DDRs through the same AXI chip2chip core. An example of two remote DDR nodes projected by a single chip2chip core is shown in Fig. 2B. The mapping configuration for both cases is one of the modifications to the operating systems.

C. SiP Switch and Control

Lithography-based photonic integration technologies hold great promise for large-scale optical integrated switch fabrics by reducing the device footprint and also the overhead in terms of assembly and calibration [31]. Planar integrated optical switches have been developed on several material platforms, such as indium phosphide, lithium niobate, silica, and silicon [32]–[36].

Silicon photonics, fabricated in high volume CMOS compatible foundries, is promising for low-cost, power-efficient interconnects. The primary switching cells that are being explored are Mach-Zehnder interferometers (MZIs) [36], MEMS-actuated couplers [37], and microring resonators (MRRs) [38]. Whilst the former two have demonstrated higher-scale integration [36], [37], the resonant devices have shown great potential for ultra-compact and energy-efficient applications [35], [39]. In addition, the wavelength-selective feature of MRRs can be utilized to route data spectrally and spatially [40], which significantly simplifies the device design and fabrication. In this work, we use silicon thermo-optic MRR based 1×8 switch fabrics as spectral-and-spatial de-multiplexers for data routing. We use the MRR to select/drop a specific wavelength to connect communicating nodes. We note that our proposed architecture is agnostic to the choice of switching device, although the individual properties of the switch cell choice will have an effect on system performance.

We choose to have an independent switch controller for future system scalability. Controlling high-radix SiP switches generally requires a large number of analog control pins due to the large number of switching elements that forms the switching matrix. A scalable solution is to have a separate switch controller with the required number of analog pins. The processing system will only be required to send configuration requests to the switch controller and the switch controller applies required analog control signals to the switching

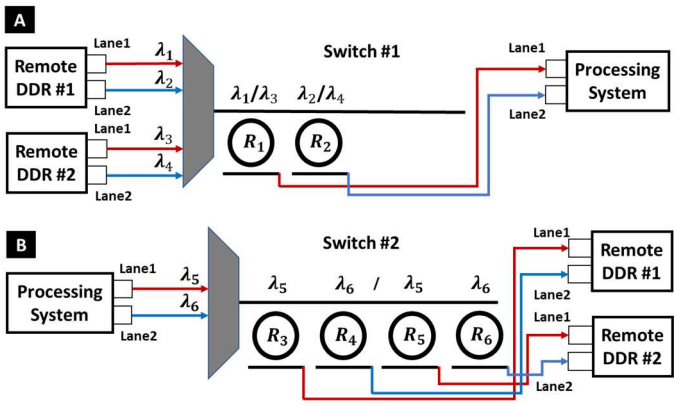

Fig. 3. SiP switches' configurations for the dynamic access to remote DDRs. (A) Remote memory resources to the processing system direction. (B) The processing system to remote memory resources direction.

elements in the SiP switches. We apply this methodology to our proposed system architecture and use group peripheral I/O (GPIO) pins as the interface to the switch controller. These control pins contain 1 bit for triggering and a power of 2 bits for the configurations. Based on the physical configuration required by users or deep learning applications, the processing system will first stabilize the configuration bits and toggle the trigger bit from logic high to logic low to initiate the reconfiguration process. For the switch controller, the procedure is as following: (1) The control logic in the switch controller samples the triggering signal and the configuration bits; (2) if triggered, it reads registers that contain pre-stored digital voltage values associated with each switching element for required configurations and (3) applies the parallel digital voltage values to digital-to-analog convertors (DACs) that bias the switching elements of the SiP switches.

D. Accelerator Design

We designed a "vanilla" accelerator on the FPGA of the ZCU106 board to further evaluate the feasibility of our photonic switched optically connected memory system architecture. The accelerator uses the standard AXI memory interface and it has the access to remote memory nodes through the AXI chip2chip core. The accelerator functions as a data mover that can "copy" and "paste" data from local DDR to remote DDR or vice versa. Although it does not heavily process the fetched data from either local or remote memories, the functionality of accessing remote memory through a standard memory interface is achieved. The ARM CPU in the processing system initially comes with AXI interface and it does not require additional implementations.

III. Testbed

We built an experimental testbed to evaluate the optical links and switching characteristics, and to demonstrate the feasibility of integrating SiP switches and remote DDRs into the processing system for DNNs. It includes one processing system node dynamically connecting two remote DDR memory blocks.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JLT.2020.2975976, Journal of Lightwave Technology

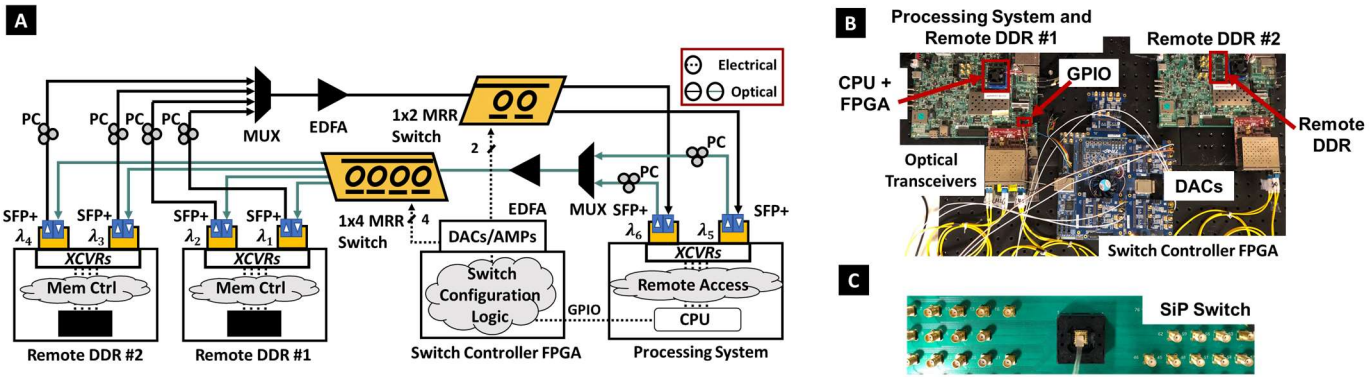

Fig. 4. (A) Experimental setup demonstrating a case of photonic switched optically connected memory system with dynamic allocation of remote DDR resources to the processing system. (B) Key hardware components. One Xilinx ZCU106 board containing the processing system and the remote DDR #1 nodes, another ZCU106 board containing only the remote DDR #2 node, and the TR4 switch controller FPGA board. (C) A packaged SiP MRR based switch with electrical SMA interface.

Two SiP switches connect the processing system to the remote DDR nodes. In this specific implementation of our architecture only a 1×2 switch and a 1×4 switch are required, although we used 1×8 SiP switches for the experiment. As we are only accessing the first MRRs, the experimental results are not impacted. Based on our system configurations, two MRRs in one of the 1×8 SiP switches are used for the direction from remote DDR nodes to the processing system and four MRRs in the other 1×8 SiP switch are used for the processing system to remote DDRs direction. We label them as 1×2 and 1×4 switches in the rest of the paper. In addition, each optical link contains two bundled lanes.

Figure 3A shows the direction from remote DDR nodes to the processing system. If the processing system requires the connection to remote DDR node #1then MRR #1 and MRR #2 in the 1×2 switch are tuned to select and forward λ_1 and λ_2 to the processing node. For the connection to the remote DDR #2, λ_3 and λ_4 are selected. Figure 3B shows the other direction for data transactions. If the system is configured as remote DDR #1 node being connected to the processing system node, the first two MRRs connected to the remote DDR #1 node in this 1×4 switch will drop λ_5 and λ_6. When the remote DDR #2 node is acquired by the processing system, MRR #3 and MRR #4 in the 1×4 switch are detuned from λ_5 and λ_6 to allow the light to pass through while MRR #5 and MRR #6 are tuned to drop and forward the light to the corresponding receiver ports of the remote DDR #2 node.

Figure 4A shows the experimental setup. Two Xilinx ZCU106 and a Terasic TR4 evaluation boards are used to evaluate the system. One of the ZCU106 boards contains both the processing system and remote DDR #1 nodes. The physical connection is only through the optical link that can be steered by the SiP switches. The other ZCU106 only comprises the remote DDR #2 logics. Each remote DDR node contains a 2 GB 64-bit wide DDR4 memory system. Six transceivers in total are used to support multi-lane optical communications. Each link contains two lanes and each lane operates at 10 Gb/s data rate. The maximum throughput for the serial link between the processing system and a remote DDR node can reach up to 20 Gb/s. Four C-band SPF+ transceivers, with wavelengths at 1545.32 nm (λ_1), 1546.92 nm (λ_2), 1553.33 nm (λ_3) and 1554.94 nm (λ_4), are used for the two remote DDR nodes to transmit data to the processing system, and two wavelengths at 1554.94 nm (λ_5) and 1556.56 nm (λ_6) are used for the opposite direction. Optical signals are combined by the multiplexers (MUX) and then enter the SiP MRR based switch chips. The polarization controllers (PC) change the polarization of the light of each lane to maximize the optical power being coupled in to and out of the SiP chips. An erbium doped fiber amplifier (EDFA) is necessary to compensate the loss due to the grating couplers of the SiP switch chips. The processing system sends configuration requests to the switch controller FPGA, on the Terasic TR4 board, which configures each MRR by tuning the resonance of each MRR with bias voltage through DACs and electrical amplifiers (AMPs). The electrical amplifiers are used to provide sufficient voltage levels to the MRRs. The configuration and trigger signals are transmitted through GPIO pins from the processing system ZCU106 board to the TR4 board. Figure 4B illustrates the key hardware components that enable the evaluation of the system. CPU, FPGA, remote DDRs, GPIO, optical transceivers, switch controller and DACs are used for the evaluation of optical link and switching characteristics. A packaged SiP chip on a printed circuit board with SMA interface is shown in Fig. 4C.

IV. Experiments and Results

The proposed photonic switched optically connected memory system architecture is evaluated by the link characteristics and the system performance measurements of the physical layer switching time, the end-to-end reconfiguration latency, the loading time of the parameters of a VGG16 DNN from hard drive to local main memory, the execution time to classify an image on CPU, and the time for storing data to/loading data from the remote DDR nodes. We use the VGG16 DNN model which is a well-known and widely used image classification model [1] as an example of a large model used in deep learning.

A. Optical Spectra

We first demonstrate that the SiP MRR based switches are capable of supporting the multi-lane optical communications required for the two different physical memory access topologies. Figure 5A shows the optical spectra at the drop port

of each MRR configured for prioritizing the physical connection between the processing system to the remote DDR #1 node. MRR #1 and MRR #2 are tuned to drop the optical wavelengths at 1545.32 nm (λ_1) and 1546.92 nm (λ_2) for the lane #1 and lane #2 from the remote DDR #1 node. The received optical power of the data signal at the corresponding receiver ports is -15.60 dBm and -18.32 dBm respectively. MRR #3 and MRR #4 are configured to select and forward the wavelengths at 1554.94 nm (λ_5) and 1556.56 nm (λ_6) from the processing system to the remote DDR #1 node. The received optical power for lane #1 and lane #2 are -17.48 dBm and -16.39 dBm respectively. For the plots of MRR #1 to MRR #4, the highest peak is the optical data signal and other peaks are crosstalk from adjacent optical channels. For the plots of MRR #5 and MRR #6, the peaks show leakage power from previous MRRs, MRR #3 and MRR #4.

Figure 5B shows the optical spectra at the drop output of each MRR for the second case where the remote DDR #2 node is connected to the processing system. The optical power received by the processing system at 1553.33 nm (λ_3) and 1554.94 nm (λ_4) is -14.96 dBm and -18.25 dBm respectively. MRR #3 and MRR #4 are detuned to allow the light to pass through these MRRs and the light can be dropped by MRR #5 and MRR #6. The received optical power at the receivers of DDR #2 node are -20.3 dBm and -18.3 dBm respectively. We ensured the received optical signal power is above the receiver sensitivity of -23 dBm.

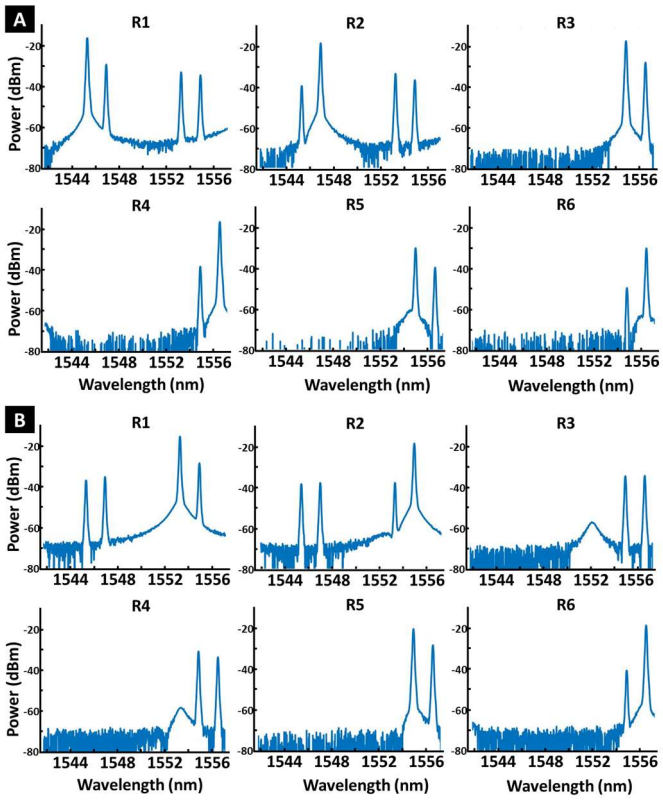

Fig. 5. Optical spectra at the drop port of each MRR for two different configurations. (A) Two SiP switches configured as the processing system connecting to the remote DDR #1 node. (B) Two SiP switches configured as the processing system connecting to the remote DDR #2 node. (In this figure, the MRR numbers are consistent with the MRR numbers shown in Fig. 3.)

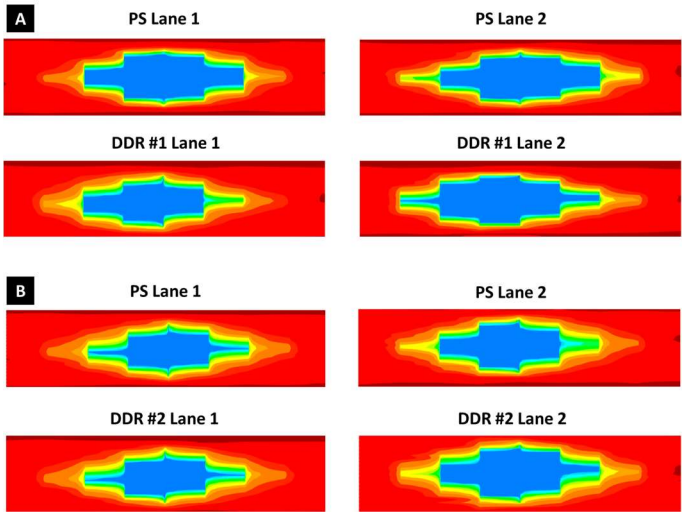

Fig. 6. Screen shots of open eye diagrams of connected receiver ports at 10 Gb/s PRBS-31. (A) Two SiP switches configured as the processing system (PS) connecting to the remote DDR #1 node. (B) Two SiP switches configured as the processing system connecting to the remote DDR #2 node.

B. Eye Diagrams

Data transmission at 10 Gb/s non-return-to-zero (NRZ) on-off keying (OOK) using 2^{31}-1 pseudo-random bit sequence (PRBS-31) was performed to extract transceiver settings for the Aurora 64B/66B IP core. With transmitter driver swing at an amplitude of 647 mV_{PPD}, pre-cursor TX pre-emphasis of 0.68 dB and post-cursor TX pre-emphasis of 1.16 dB, error-free operations over the optical links are achieved. All the connected paths for the two different configurations show clear eye-openings as shown in Fig. 6.

C. Switching Time

We performed measurements of two switching cases between two configurations: (1) the remote DDR #2 node connected to the processing system, and (2) the remote DDR #1 node connected to the processing system. The first switching case is changing from configuration #1 to configuration #2 and the second switching operation happens 330 μs after the first switching operation, which is changing from configuration #2 to configuration #1. In Fig. 7, we show the transient responses of the received optical power, normalized individually for each MRR.

As shown in Fig. 7A, the first switching case starts at the time that approximately equals to 50 μs. We notice that MRR #4 experiences faster rise time than MRR #3 and becomes stabilized within a shorter time. The local maxima and the local minima of the orange curve are due to the fact that MRR #4 passes through 1554.94 nm (λ_5) during the first switching process. MRR #5 and MRR #6 are initially tuned at 1554.94 nm (λ_5) and 1556.56 nm (λ_6), and the control bias voltages are not changed during the process, thus the transient response of MRR

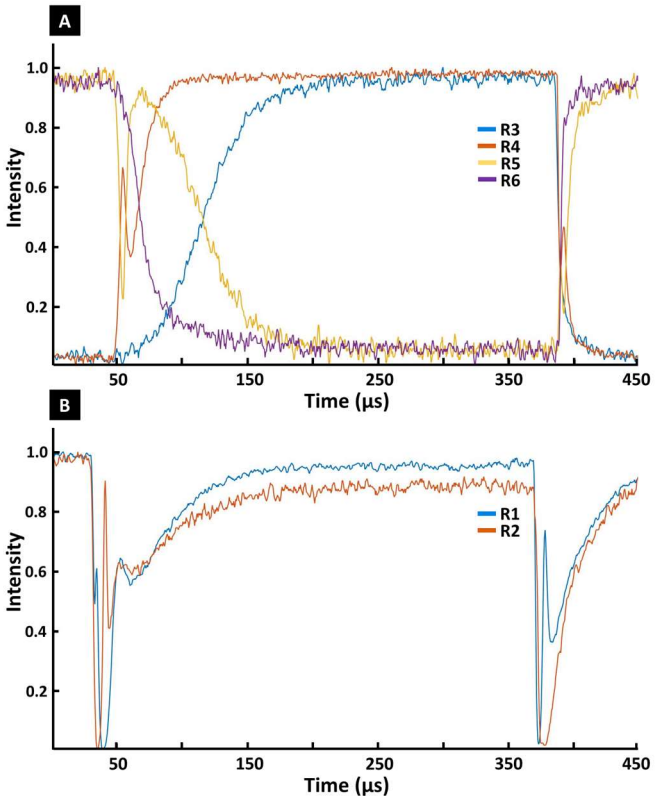

Fig. 7. Transient responses at the receiver ports of all MRRs for two switching cases separated by a time duration of 330 μs. (A) Transient responses of MRR #3, MRR #4, MRR #5 and MRR#6 in the 1×4 switch. (B) Transient responses of MRR #1 and MRR #2 in the 1×2 switch.

#5 is reciprocal to the superposition of the transient responses of MRR #3 and MRR #4. The transient response of MRR #6 is reciprocal to MRR #4 only. For the configuration #2 to configuration #1 case, MRR #3 and MRR #4 are detuned to allow the optical signals to pass through, and the transient responses can be observed at the time approximately equal to 380 μs. The limiting factor of the switching operation is the slowest transient response of all the responses. We can see from Fig. 7A that the rise time of MRR #3 for the switching from configuration #1 to configuration #2 is the slowest transient response and the latency is approximately 119 μs. We have, however, shown that the thermo-optic switching time can be as low as 1.2 μs with optimized driving circuitry [41].

Figure 7B illustrates the transient responses at the receiver ports for MRR#1 and MRR #2 in the two switching scenarios. Since both MRR #1 and MRR #2 are dropping optical signals for two configuration cases, the transient response of each individual MRR is expected to fall first and then rise back during the switching process. As we find from Fig. 7B, the slowest transient response is approximately 107.5 μs at the receiver port for MRR #1 in the first switching case.

D. End-to-end reconfiguration time

The system end-to-end reconfiguration latency consists of (1) the time for AXI chip2chip and Aurora 64/66B cores to reset, (2) optical switching time, and (3) link re-negotiation time. To reconfigure the physical connections between the processing system and remote DDR nodes, the AXI chip2chip and Aurora 64/66B cores in the processing system are required to be put into reset state. This reset action will also be propagated to the remote DDR end to restart the link-renegotiation process. The reset process and the link-renegotiation process are described in [29], [30]. One requirement for this process is that the asserted reset state needs to last at least 128 user clock cycles and we chose to set the cores to be in reset state for 2 ms. The optical switching time shown in the previous section is approximately 119 μs and we chose to wait 330 μs to ensure the optical link is stabilized. The reset was then released and the link-renegotiation process started. This renegotiation time was measured to be 0.45 ms. In total, the end-to-end reconfiguration time was 2.78 ms.

E. Application and Execution Time

We built a Linux kernel image based upon the system implementation using Xilinx PetaLinux tool and booted the operating system with Ubuntu 18.04 filesystem on the Xilinx ZCU106 board. The kernel image is stored in the SD card boot partition while the filesystem is stored in the hard drive root partition. The hard drive is connected to the processing system through SATA interface.

We evaluated the system performance by measuring the latencies of loading data from storage to local memory, storing data from local memory to storage, loading data from remote memory to local memory, storing from local memory to remote memory, and classifying an image on the ARM Cortex CPU. A VGG16 model was pretrained using TensorFlow in Python and its parameters, such as weights and biases for each layer in the network, are also saved in the hard drive. A feedforward implementation of the neural network including the convolutional and fully-connected layers is coded in the C programming language, thus the processing system is capable of running a C program to load the parameters and classify an image using the pretrained VGG16 model on the ARM CPU. The VGG16 model contains 13 convolutional layers and 3 fully-connected layers with 138,357,544 parameters and we use 32-bit floating point data type for each parameter. Thus, the total size of the VGG16 is approximately 528MB. The loading time from the hard drive to the local main memory is 5.70 s for the entire VGG16. The execution time to classify an image is 63.34 s on the ARM CPU.

To measure the latencies of using remote DDR for storing/loading parameters, i.e. weights and biases, we use our designed accelerator in a standalone design (without the operating system). The time for storing 528 MB data, the same size as the VGG16, from the local contiguous memory allocation (CMA) region to the remote DDR takes 1.40 s for the accelerator and loading the data from remote DDR to the local CMA region takes 1.34 s.

The accelerator's equivalent throughput for loading from the remote memory to local memory is 3.31 Gb/s, and 3.16 Gb/s

for storing. The limited throughput is due to the fact that the designed accelerator operates at 250 MHz with 32-bit AXI data channel width, which can theoretically achieve up to 7.8 Gb/s without overhead. In addition, the accelerator performs "copy" and "paste" operations which lead to an overhead factor of approximately 0.5 over the entire system. By increasing the clock frequency and data channel width of the accelerator, higher throughput can be achieved.

During training, memory space is required to store each layer's output and its corresponding gradients. This space required is the same size as the layer's output for backpropagation when stochastic gradient descent (SGD) [16] optimization strategy and ReLu [42] activation function are used. To evaluate the feasibility of our proposed architecture for increasing the memory capacity for training, we performed a forward propagation of the VGG16 with a batch size of 1, 2, 4, 8, and 16 images in the software. Based upon the memory requirement for training, we stored/loaded the intermediate layer results and randomly initialized gradients to/from both remote memory and the hard drive for the purpose. The intermediate results include the output of each convolutional layer, max pooling layer and fully-connected layer. There are 15,087,080 elements of intermediate layer results per image to be stored for backpropagation. Considering the gradients, there are in total 30,174,160 elements per image that are being stored/loaded during the process. The time for storing to the hard drive is 1.14 s, 2.49 s, 4.98 s, 10.53 s, and 22.58 s, respectively. For loading from the hard drive, the latencies are 1.26 s, 2.53 s, 5.24 s, 10.61 s, and 20.57 s, respectively. As expected, the latencies for storing/loading using remote DDR are less than using the hard drive in the testbed. The storing latencies using the remote memory are 0.31 s, 0.61 s, 1.23 s, 2.45 s, and 4.92 s, while the loading latencies are 0.30 s, 0.60 s, 1.21 s, 2.41 s, and 4.85 s, respectively. Figure 8 compares the latencies of using hard drive and remote DDR memory and shows that the required memory space for layer output and layer gradients grows with the batch size. We note that larger batch size will require more memory and the memory requirement is also related to the use of other optimizers [16], but the functionality of our architecture and the remote memory remains the same. Table I lists the results for the system performance measurements.

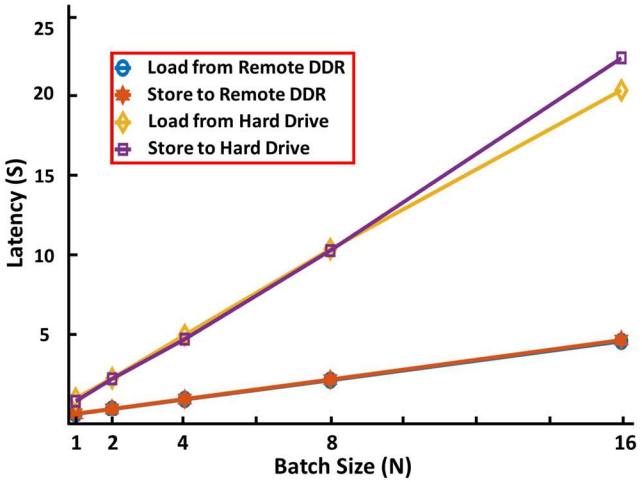

Fig. 8. Loading/storing latencies using hard drive and remote DDR memory of different batch sizes.

TABLE I
SYSTEM PERFORMANCE MEASUREMENTS

Operations	Latency
Optical switching	119 μs
End-to-end reconfiguration	2.78 ms
Load VGG16 (528 MB) from hard drive to local DDR memory	5.70 s
Load 528 MB data from remote DDR to local DDR CMA region (accelerator)	1.34 s
Store 528 MB data from local DDR CMA region to remote DDR (accelerator)	1.40 s
Load intermediate results and gradients from hard drive to local DDR memory (batch size of 16)	20.57 s
Store intermediate results and gradients from local DDR memory to hard drive (batch size of 16)	22.58 s
Load intermediate results and gradients from remote DDR to local DDR CMA region (batch size of 16)	4.85 s
Store intermediate results and gradients from local DDR CMA region to remote DDR (batch size of 16)	4.92 s
Classify an image using VGG16 on ARM CPU	63.34 s

Figure 9 compares the three scenarios for the test case of inference: processing system loading from storage, processing system with remote DDR and optical interconnect, accelerator with remote DDR and optical interconnect. The total execution time consists of both compute time and the time for data access. For the latter, we can achieve a speedup of 4.3 when loading the data from remote DDR compared to loading from the storage device to the local DDR. The end-to-end reconfiguration latency we observed is much shorter than the loading time therefore we use 1.34 s as the total time for the processing system to load the data from the remote memory. In the case of the accelerator, the end-to-end reconfiguration time is the only one considered as the accelerator can directly access the remote memory without loading. We note that the optical reconfiguration time is a constant overhead independent of the data size. With increased data size the impact of the overhead is amortized.

V. DISCUSSION

Optical switching technology enables reconfigurable disaggregation allowing the processing system to dynamically access additional memory resources. In order to successfully integrate the photonic switched optically connected memory into the system, several requirements for the optical switches need to be taken into consideration including: optical power budget, reconfiguration time, power consumption and scalability.

The optical power budget available is based on the receiver sensitivity and the optical power launched by the transmitter. The insertion loss of the optical switches should be well below

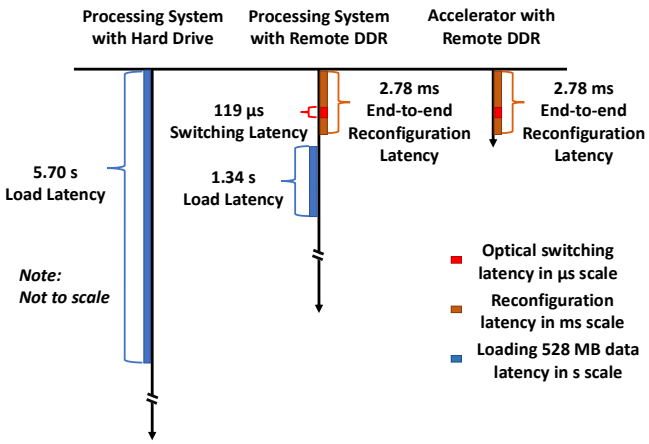

Fig. 9. Timelines comparing system latencies in different scenarios for switching case #2.

this if the system has no optical amplification. If the insertion loss of the switch and additional losses in the link go beyond this optical power budget, optical amplification is required, which is generally not desirable due to energy and cost considerations, although recent work with semiconductor amplifiers has shown promise [43]. The extinction ratio (ER) of the optical switch also depends on the optical transceiver. For a transmitter ER of 3.5 dB, we measured more than 10 dB power suppression ratio of the optical signal power to the optical leakage power which can guarantee error-free operation.

End-to-end reconfiguration latency is an important network parameter. This parameter includes optical switching time, transceiver reset and link negotiation. In order to not introduce excessive overhead, the optical switch reconfiguration should not occupy more than ten percent of the entire reconfiguration latency. In this case we have shown that the optical switch reconfiguration time is not detrimental to the system performance. To decrease the overhead of the optical switching and link reconfiguration latency, advanced high-speed devices could be employed. Electro-optic switches and burst-mode transceivers can be deployed in the system. Electro-optic silicon photonic switches provide nanosecond-scale reconfiguration time [44] and sub-nanosecond clock and data recovery has been demonstrated in an optically switched link via clock phase catching [45]. The achievable end-to-end reconfiguration latency can thus be reduced to the nanosecond scale.

The power consumption of the optical switch should be a small fraction of the power consumption of the entire system. The state-of-the-art GPU [46] can consume up to 280 W while reported silicon photonic switches [47], are in the range of Watts and are therefore relatively power efficient when integrated into the system to support dynamic memory resource allocation. For example a 32×32 MZI-based switch consuming a power of 1.9W [36]. The switch fabric used in this experiment consumes approximately 10 mW per MRR.

Although we demonstrated a 1×2 switching scenario in the testbed, larger N×M optical switches in application dependent topologies would support the system requirements, depending on the number of compute/accelerator nodes (N) and the remote memory nodes (M) within the subsystem. Ref. [20] indicates a use case of 8 compute and 8 memory nodes. A full analysis of the relationship between the radix/topology of the optical switch and the overall system performance/cost can be performed using the same methodology as shown in our previous work [48], for specific applications and switch architectures.

Our experimental testbed was designed to experimentally demonstrating the proof-of-concept functionalities of our proposed system architecture. Although we used legacy SATA based storage devices in our testbed, commercially available storage drives can support up to 2,375 MB/s throughput (Amazon Web Service [49]). In order for our proposed architecture to demonstrate comparable speedup using commercial high-end storage devices, one would build an optical system with comparable high-end bandwidth optical I/Os and optimized transceiver circuitry. Multiwavelength terabit optical links are under development with state-of-the-art silicon transceivers capable of modulating [50] and detecting [51] at over 100 GHz bandwidth.

In summary, our proposed photonic switched system architecture demonstrates the concept of using dynamic allocation of memory to tackle the scaling challenge of deep learning. Our test cases demonstrate the capability of increasing memory capacity at the program-level using an architecture based on MRR optical switches, FPGA processing systems, and optically connected DDR memories. The designed “lite” (de)serialization and memory mapping scheme show a path towards lowering the system latency, a critical metric for disaggregated systems. The independent switch controller is scalable and is able to be applied in the systems requiring large number of switching elements as long as they are controlled by biasing voltages. The proposed system architecture shows a significant step toward deploying photonic interconnects and optically connected memory for deep learning applications. More generally, with specific optimizations the approach would also be applied to other workloads that face the same memory challenges.

VI. Conclusion

We demonstrate a proof of concept system architecture, showing the functionality of photonic switched optically connected memory for large DNNs in deep learning. It features dynamic allocation of additional memory to the processing system and a constant reconfiguration latency. The experimental testbed demonstrates real memory transactions between the processing system and remote memory nodes. We measured a 119 µs latency for optical switching and an overall 2.78 ms latency for the end-to-end reconfiguration. Our results and silicon-based high-bandwidth I/O capabilities show the feasibility of using photonic switched optically connected memory to solve the memory challenges in future deep learning applications.

ACKNOWLEDGMENT

The authors would like to acknowledge Elenion Technologies for support of some of the devices used in this work.

REFERENCES

[1] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv:1409.1556 [cs]*, Apr. 2015.

[2] R. Socher *et al.*, "Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Seattle, Washington, USA, 2013, pp. 1631–1642.

[3] F. J. Ordóñez and D. Roggen, "Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition," *Sensors*, vol. 16, no. 1, p. 115, Jan. 2016, doi: 10.3390/s16010115.

[4] R. Raina, A. Madhavan, and A. Y. Ng, "Large-scale Deep Unsupervised Learning Using Graphics Processors," in *Proceedings of the 26th Annual International Conference on Machine Learning*, New York, NY, USA, 2009, pp. 873–880, doi: 10.1145/1553374.1553486.

[5] S. Chakradhar, M. Sankaradas, V. Jakkula, and S. Cadambi, "A Dynamically Configurable Coprocessor for Convolutional Neural Networks," in *Proceedings of the 37th Annual International Symposium on Computer Architecture*, New York, NY, USA, 2010, pp. 247–257, doi: 10.1145/1815961.1815993.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[7] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning Transferable Architectures for Scalable Image Recognition," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8697–8710, doi: 10.1109/CVPR.2018.00907.

[8] M. Rhu, N. Gimelshein, J. Clemons, A. Zulfiqar, and S. W. Keckler, "vDNN: Virtualized deep neural networks for scalable, memory-efficient neural network design," in *2016 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2016, pp. 1–13, doi: 10.1109/MICRO.2016.7783721.

[9] MD. Z. Hossain, F. Sohel, M. F. Shiratuddin, and H. Laga, "A Comprehensive Survey of Deep Learning for Image Captioning," *ACM Comput. Surv.*, vol. 51, no. 6, pp. 118:1–118:36, Feb. 2019, doi: 10.1145/3295748.

[10] J. Donahue *et al.*, "Long-Term Recurrent Convolutional Networks for Visual Recognition and Description," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 677–691, Apr. 2017, doi: 10.1109/TPAMI.2016.2599174.

[11] S. Abu-El-Haija *et al.*, "YouTube-8M: A Large-Scale Video Classification Benchmark," *arXiv:1609.08675 [cs]*, Sep. 2016.

[12] J. Park *et al.*, "Deep Learning Inference in Facebook Data Centers: Characterization, Performance Optimizations and Hardware Implications," *arXiv:1811.09886 [cs, stat]*, Nov. 2018.

[13] J. Hestness *et al.*, "Deep Learning Scaling is Predictable, Empirically," *arXiv:1712.00409 [cs, stat]*, Dec. 2017.

[14] J. Hestness, N. Ardalani, and G. Diamos, "Beyond human-level accuracy: computational challenges in deep learning," in *Proceedings of the 24th Symposium on Principles and Practice of Parallel Programming*, Washington, District of Columbia, 2019, pp. 1–14, doi: 10.1145/3293883.3295710.

[15] A. Canziani, A. Paszke, and E. Culurciello, "An Analysis of Deep Neural Network Models for Practical Applications," *arXiv:1605.07678 [cs]*, Apr. 2017.

[16] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv:1609.04747 [cs]*, Jun. 2017.

[17] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, "On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima," *arXiv:1609.04836 [cs, math]*, Feb. 2017.

[18] U. Gupta *et al.*, "The Architectural Implications of Facebook's DNN-based Personalized Recommendation," *arXiv:1906.03109 [cs]*, Jun. 2019.

[19] A. Krizhevsky, "One weird trick for parallelizing convolutional neural networks," *arXiv:1404.5997 [cs]*, Apr. 2014.

[20] Y. Kwon and M. Rhu, "Beyond the Memory Wall: A Case for Memory-Centric HPC System for Deep Learning," in *2018 51st Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2018, pp. 148–161, doi: 10.1109/MICRO.2018.00021.

[21] A. Eisenman *et al.*, "Bandana: Using Non-volatile Memory for Storing Deep Learning Models," *arXiv:1811.05922 [cs, stat]*, Nov. 2018.

[22] G. Zervas, H. Yuan, A. Saljoghei, Q. Chen, and V. Mishra, "Optically Disaggregated Data Centers With Minimal Remote Memory Latency: Technologies, Architectures, and Resource Allocation [Invited]," *J. Opt. Commun. Netw.*, vol. 10, no. 2, p. A270, Feb. 2018, doi: 10.1364/JOCN.10.00A270.

[23] K. Bergman, J. Shalf, G. Michelogiannakis, S. Rumley, L. Dennison, and M. Ghobadi, "PINE: An Energy Efficient Flexibly Interconnected Photonic Data Center Architecture for Extreme Scalability," in *2018 IEEE Optical Interconnects Conference (OI)*, 2018, pp. 25–26, doi: 10.1109/OIC.2018.8422036.

[24] Y. Yan *et al.*, "All-Optical Programmable Disaggregated Data Centre Network Realized by FPGA-Based Switch and Interface Card," *Journal of Lightwave Technology*, vol. 34, no. 8, pp. 1925–1932, Apr. 2016, doi: 10.1109/JLT.2016.2518492.

[25] T. Shiraishi *et al.*, "A reconfigurable and redundant optically-connected memory system using a silicon photonic switch," in *OFC 2014*, 2014, pp. 1–3, doi: 10.1364/OFC.2014.Th2A.10.

[26] D. Brunina, C. Lai, A. Garg, and K. Bergman, "Building Data Centers With Optically Connected Memory," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 3, no. 8, pp. A40–A48, Aug. 2011, doi: 10.1364/JOCN.3.000A40.

[27] A. G. Howard *et al.*, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *arXiv:1704.04861 [cs]*, Apr. 2017.

[28] "AMBA AXI4 Interface Protocol." [Online]. Available: https://www.xilinx.com/products/intellectual-property/axi.html. [Accessed: 24-Nov-2019].

[29] "AXI Chip2Chip." [Online]. Available: https://www.xilinx.com/products/intellectual-property/axi-chip2chip.html. [Accessed: 24-Nov-2019].

[30] "Aurora 64B/66B." [Online]. Available: https://www.xilinx.com/products/intellectual-property/aurora64b66b.html. [Accessed: 24-Nov-2019].

[31] Q. Cheng, M. Bahadori, M. Glick, S. Rumley, and K. Bergman, "Recent advances in optical technologies for data centers: a review," *Optica, OPTICA*, vol. 5, no. 11, pp. 1354–1370, Nov. 2018, doi: 10.1364/OPTICA.5.001354.

[32] Q. Cheng, A. Wonfor, J. L. Wei, R. V. Penty, and I. H. White, "Demonstration of the feasibility of large-port-count optical switching using a hybrid Mach-Zehnder interferometer-semiconductor optical amplifier switch module in a recirculating loop," *Opt. Lett., OL*, vol. 39, no. 18, pp. 5244–5247, Sep. 2014, doi: 10.1364/OL.39.005244.

[33] P. J. Duthie and M. J. Wale, "16*16 single chip optical switch array in lithium niobate," *Electronics Letters*, vol. 27, no. 14, pp. 1265–1266, Jul. 1991, doi: 10.1049/el:19910793.

[34] S. Sohma, T. Watanabe, N. Ooba, M. Itoh, T. Shibata, and H. Takahashi, "Silica-based PLC Type 32 x 32 Optical Matrix Switch," in *2006 European Conference on Optical Communications*, 2006, pp. 1–2, doi: 10.1109/ECOC.2006.4801113.

[35] Q. Cheng *et al.*, "Ultralow-crosstalk, strictly non-blocking microring-based optical switch," *Photon. Res., PRJ*, vol. 7, no. 2, pp. 155–161, Feb. 2019, doi: 10.1364/PRJ.7.000155.

[36] K. Suzuki *et al.*, "Low-Insertion-Loss and Power-Efficient 32 × 32 Silicon Photonics Switch With Extremely High-Δ Silica PLC Connector," *J. Lightwave Technol., JLT*, vol. 37, no. 1, pp. 116–122, Jan. 2019.

[37] T. J. Seok, T. J. Seok, K. Kwon, J. Henriksson, J. Luo, and M. C. Wu, "240×240 Wafer-Scale Silicon Photonic Switches," presented at the Optical Fiber Communication Conference, 2019, p. Th1E.5, doi: 10.1364/OFC.2019.Th1E.5.

[38] Q. Cheng, M. Bahadori, Y. Hung, Y. Huang, N. Abrams, and K. Bergman, "Scalable Microring-Based Silicon Clos Switch Fabric With Switch-and-Select Stages," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 25, no. 5, pp. 1–11, Sep. 2019, doi: 10.1109/JSTQE.2019.2911421.

[39] E. Timurdogan, C. M. Sorace-Agaskar, J. Sun, E. Shah Hosseini, A. Biberman, and M. R. Watts, "An ultralow power athermal silicon modulator," *Nat Commun*, vol. 5, no. 1, p. 4008, Sep. 2014, doi: 10.1038/ncomms5008.

[40] Q. Cheng, M. Bahadori, M. Glick, and K. Bergman, "Scalable Space-and-Wavelength Selective Switch Architecture Using Microring Resonators,"

in *Conference on Lasers and Electro-Optics (2019), paper STh1N.4*, 2019, p. STh1N.4, doi: 10.1364/CLEO_SI.2019.STh1N.4.

[41] Y. Huang *et al.*, "Multi-Stage 8×8 Silicon Photonic Switch based on Dual-Microring Switching Elements," *Journal of Lightwave Technology*, pp. 1–1, 2019, doi: 10.1109/JLT.2019.2945941.

[42] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation Functions: Comparison of trends in Practice and Research for Deep Learning," *arXiv:1811.03378 [cs]*, Nov. 2018.

[43] R. Konoike *et al.*, "SOA-Integrated Silicon Photonics Switch and Its Lossless Multistage Transmission of High-Capacity WDM Signals," *Journal of Lightwave Technology*, vol. 37, no. 1, pp. 123–130, Jan. 2019, doi: 10.1109/JLT.2018.2868084.

[44] L. Qiao, W. Tang, and T. Chu, "32 × 32 silicon electro-optic switch with built-in monitors and balanced-status units," *Scientific Reports*, vol. 7, p. 42306, Feb. 2017, doi: 10.1038/srep42306.

[45] K. Clark *et al.*, "Sub-Nanosecond Clock and Data Recovery in an Optically-Switched Data Centre Network," in *2018 European Conference on Optical Communication (ECOC)*, 2018, pp. 1–3, doi: 10.1109/ECOC.2018.8535333.

[46] "NVIDIA TITAN RTX is Here," *NVIDIA*. [Online]. Available: https://www.nvidia.com/en-us/deep-learning-ai/products/titan-rtx/. [Accessed: 17-Jan-2020].

[47] B. G. Lee and N. Dupuis, "Silicon Photonic Switch Fabrics: Technology and Architecture," *Journal of Lightwave Technology*, vol. 37, no. 1, pp. 6–20, Jan. 2019, doi: 10.1109/JLT.2018.2876828.

[48] Q. Cheng, S. Rumley, M. Bahadori, and K. Bergman, "Photonic switching in high performance datacenters [Invited]," *Opt. Express*, vol. 26, no. 12, p. 16022, Jun. 2018, doi: 10.1364/OE.26.016022.

[49] "Amazon EBS Features - Amazon Web Services," *Amazon Web Services, Inc.* [Online]. Available: https://aws.amazon.com/ebs/features/. [Accessed: 17-Jan-2020].

[50] L. Alloatti *et al.*, "100 GHz silicon–organic hybrid modulator," *Light: Science & Applications*, vol. 3, no. 5, pp. e173–e173, May 2014, doi: 10.1038/lsa.2014.54.

[51] Y. Salamin *et al.*, "100 GHz Plasmonic Photodetector," *ACS Photonics*, vol. 5, no. 8, pp. 3291–3297, Aug. 2018, doi: 10.1021/acsphotonics.8b00525.

Ziyi Zhu received the B.Eng. degree in electrical engineering with honors in 2015 from Sichuan University, Chengdu, China, and the M.S. degree in electrical engineering with honors in 2017 from Columbia University, New York, NY, USA, where he is currently working toward the Ph.D. degree with interests in developing and testing optical switches and FPGA-controlled optical interconnects.

Giuseppe Di Guglielmo (S'06–M'09) received the Laurea degree (summa cum laude) and the Ph.D. degree in computer science from the Università di Verona, Verona, Italy, in 2005 and 2009, respectively. He is currently an Associate Research Scientist with the Department of Computer Science, Columbia University, New York, NY, USA. He has authored over 50 publications. He collaborated in several U.S., Japanese, and Italian projects. His current research interests include system-level design and validation of system-on-chip platforms.

Qixiang Cheng (M'17) received his B.S from Huazhong University of Sci. & Tech., China in 2010 and Ph.D. from the University of Cambridge, UK in 2014, in the field of III/V integrated circuits. In Feburary 2015, he joined Shannon Lab., Huawei researching future optical computing systems.

He is now a Research Scientist at the Lightwave Research Lab, Columbia University, New York, USA. His current research interests focus on system-wide photonic integrated circuits for optical communication and optical computing applications, including a range of optical functional circuits such as packet-, circuit-, and wavelength-level optical switch fabrics, massively parallel transceivers, optical neural networks, and optical network-on-chip. These exploit a number of photonic integration platforms, including InP- and Si-based monolithic circuits, as well as Si-InP, Si-SiN, and photonic-electronic hybrid integration schemes.

Jihye Kwon received the B.S. degree in mathematical sciences and the M.S. degree in computer science and engineering from Seoul National University, Seoul, Korea, in 2012 and 2014, respectively. She is currently pursuing the Ph.D. degree in computer science at Columbia University, New York, NY, USA.

Her research interest includes system-level design methodology enhanced via learning and interaction, high-level synthesis for designing hardware accelerators, real-time scheduling theory, and solving optimization problems. She has interned at IBM T. J. Watson Research Center during the summer in 2015, 2016, and 2017. During her PhD studies, she has received Presidential Fellowship from Columbia University.

Madeleine Glick received her Ph.D. in physics at Columbia University for research on electro-optic effects of GaAs/AlGaAs quantum wells. After receiving her degree, she joined the Department of Physics, Ecole Polytechnique Federale de Lausanne (EPFL) Lausanne, Switzerland, where she continued her research in electro-optic effects in GaAs and InP-based materials. From 1992 to 1996, she was a Research Associate with CERN, Geneva, Switzerland, as part of the Lightwave Links for Analogue Signal Transfer Project for the Large Hadron Collider. From 2002-2011, Madeleine was Principal Engineer at Intel (Intel Research Cambridge UK, Intel Research Pittsburgh) leading research on optical interconnects for computer systems. Her research interests are in applying photonic devices and interconnects to computing systems. Madeleine is a Fellow of the Institute of Physics, and a Senior Member of IEEE and OSA.

Hang Guan received the M.S. and Ph.D. degrees from Columbia University, New York, NY, USA in 2017 and 2018, respectively, all in Electrical Engineering. He is currently with Elenion Technologies, LLC. His research interests include

high-speed silicon photonic transceiver design/testing, and silicon photonic devices design/testing.

Luca P. Carloni (S'95-M'04-SM'09-F'17) received the Laurea degree (summa cum laude) in electrical engineering from the Universita' di Bologna, Bologna, Italy, in 1995 and the M.S. and Ph.D. degrees in electrical engineering and computer sciences from the University of California at Berkeley, Berkeley, CA, USA, in 1997 and 2004, respectively. He is a Professor of computer science with Columbia University, New York, NY, USA. He has authored over 130 publications and holds two patents. His current research interests include system-on-chip platforms, system-level design, distributed embedded systems, and high-performance computer systems. Dr. Carloni was a recipient of the Demetri Angelakos Memorial Achievement Award in 2002, the Faculty Early Career Development (CAREER) Award from the National Science Foundation in 2006, the ONR Young Investigator Award in 2010, and the IEEE CEDA Early Career Award in 2012. He was selected as an Alfred P. Sloan Research Fellow in 2008. His paper on the latency-insensitive design methodology was selected for the Best of ICCAD in 1999, a collection of the best papers published in the first 20 years of the IEEE International Conference on Computer-Aided Design. In 2013, he served as the General Chair of Embedded Systems Week, the premier event covering all aspects of embedded systems and software. He is a Senior Member of the Association for Computing Machinery.

Keren Bergman (S'87–M'93–SM'07–F'09) received the B.S. degree from Bucknell University, Lewisburg, PA, in 1988, and the M.S. and Ph.D. degrees from the Massachusetts Institute of Technology, Cambridge, in 1991 and 1994, respectively, all in electrical engineering. Dr. Bergman is currently a Charles Batchelor Professor at Columbia University, New York, NY, where she also directs the Lightwave Research Laboratory. She leads multiple research programs on optical interconnection networks for advanced computing systems, data centers, optical packet switched routers, and chip multiprocessor nanophotonic networks-on-chip. Dr. Bergman is a Fellow of the IEEE and OSA.