HolyLight: A Nanophotonic Accelerator for Deep Learning in Data Centers

Weichen Liu[1], Wenyang Liu[2], Yichen Ye[3], Qian Lou[4], Yiyuan Xie[3], Lei Jiang[4]
[1]School of Computer Science and Engineering, Nanyang Technological University, Singapore
[2]College of Computer Science, Chongqing University, Chongqing, China
[3]School of Electronics and Information Engineering, Southwest University, Chongqing, China
[4]School of Informatics, Computing and Engineering, Indiana University Bloomington, USA

***Abstract*—Convolutional Neural Networks (CNNs) are widely adopted in object recognition, speech processing and machine translation, due to their extremely high inference accuracy. However, it is challenging to compute massive computationally expensive convolutions of deep CNNs on traditional CPUs and GPUs. Emerging Nanophotonic technology has been employed for on-chip data communication, because of its CMOS compatibility, high bandwidth and low power consumption. In this paper, we propose a nanophotonic accelerator, HolyLight, to boost the CNN inference throughput in datacenters. Instead of an all-photonic design, HolyLight performs convolutions by photonic integrated circuits, and process the other operations in CNNs by CMOS circuits for high inference accuracy. We first build HolyLight-M by microdisk-based matrix-vector multipliers. We find analog-to-digital converters (ADCs) seriously limit its inference throughput per Watt. We further use microdisk-based adders and shifters to architect HolyLight-A without ADCs. Compared to the state-of-the-art ReRAM-based accelerator, HolyLight-A improves the CNN inference throughput per Watt by $13\times$ with trivial accuracy degradation.**

***Index Terms*—Nanophotonic Computing, Convolutional Neural Network, Accelerator**

I. Introduction

Deep CNNs emerge as one of the most effective solutions to a wide range of problems, e.g., object recognition [10], speech processing and machine translation. Deep CNN models trained by huge datasets are extremely relevant and critical to emerging cloud services such as speech recognition (e.g., Apple Siri and Google assistant) and face identification (e.g., Apple iPhoto and Google Picasa). However, a CNN inference involves a huge volume of computationally intensive convolutions. For example, even for a CNN created in 2012, AlexNet [10], an inference requires 724M floating point (FP) multiply-accumulate (MAC) operations. The essential computing effort of CNN inferences prevents traditional CPUs and GPUs from achieving high processing throughput per Watt [4], [9], [19] in data centers.

Once a CNN model is trained by multiple powerful GPUs, trillions of inferences will be performed on the model. Therefore, both academia and industry create ASIC-based accelerators, e.g., DaDianNao [4] and Google TPU [9], to power-efficiently accelerate CNN inferences. The ASIC-based accelerators use CMOS matrix-vector multiplication units to boost the convolution throughput for CNNs. A recent processing-in-memory (PIM) accelerator, ISAAC [19], further improves the CNN inference throughput per Watt by emerging ReRAM-based dot-product engines. However, even with ReRAM-based PIMs, it is still challenging for data centers to analyze the exponentially growing big data by CNNs, because of the looming data center power crisis [17]. According to the US National Security Agency, the Internet is processing 1,826 Petabytes of data each day and the world big data amount will reach 35 trillion gigabytes by 2020 [3]. On the contrary, the power budget of data centers where the big data are analyzed by deep learning is projected to only slightly increase. As a result, it is urgent for data centers to have more power-efficient hardware accelerators to improve the throughput of CNN inferences on big data.

Recent works [12], [20], [22] adopt fast and low power nanophotonic technology to accelerate neural network (NN) inferences and trainings. However, these fully-optical accelerators achieve only low inference accuracy even when processing tiny NN models on small datasets, since they treat matrix-vector multiplications as analog photonic signal processing without converting intermediate results to digital values for error corrections. Inevitable noises accumulate during fully-optical NN inferences and substantially degrade the inference accuracy. The latest fully-optical accelerator [12] obtains only 93.39% inference accuracy when recognizing handwritten digits on small MNIST dataset, while the state-of-the-art CNN accuracy can easily reach 99.77% [10]. Such accuracy degradation is not acceptable for emerging cloud deep learning services. In this paper, we propose a nanophotonic accelerator, HolyLight, for accurate deep CNN inferences in data centers. Our contribution is summarized as follows.

- We build a nanophotonic CNN accelerator, HolyLight-M, by on-chip photonic matrix-vector multipliers. Through analog-to-digital converters (ADCs), we convert analog photonic intermediate results to digital values to enhance the CNN inference accuracy. We then identify the performance and power bottleneck of HolyLight-M lies in the ADCs.
- To mitigate the ADC bottleneck, by on-chip photonic adders and shifters, we architect another nanophotonic accelerator, HolyLight-A, to boost the inference performance of power-of-2 quantized CNNs (P2Q-CNNs),

Corresponding author: Weichen Liu. Email: liu@ntu.edu.sg.

978-3-9819263-2-3/DATE19/©2019 EDAA

which are simplified CNN models introducing less than 1% accuracy degradation. Through fully-digital photonic computing components, HolyLight-A guarantees highly accurate P2Q-CNN inferences without ADCs.

- We implement, evaluate and compare HolyLight-M and HolyLight-A against the state-of-the-art CPU, GPU, FPGA, ASIC and ReRAM-based CNN accelerators. Our experimental results show HolyLight-A improves the CNN inference performance per Watt by 13× over the ReRAM-based PIM accelerator.

II. Background and Prior Art

A. Convolutional neutral network

The state-of-the-art CNNs [7], [10] are supervised learning algorithms invoking a feedforward function during *inferences* and a backpropagation process for *trainings*. In data centers, emerging cloud services have to be responsive to inference requests from mobile and IoT devices, so CNN inference speed matters. In contrast, we can train CNNs off-line. In this paper, we focus on accelerating CNN inferences in data centers. A typical CNN consists of multiple types of layers including convolutional, activation, pooling and fully-connected layers [8]. The convolutional layer receives IN input channels, each of which has INC columns and INR rows. Each input channel is convolved by a shifting window with a $K \times K$ weight filter to generate an element in one $OUTC \times OUTR$ output channel. The stride of the shifting window is SW ($< K$). Totally, OU output channels are generated by the convolutional layer. The pseudo code of a convolutional layer is shown in Figure 1. The convolutional layers in AlexNet cost $> 90\%$ of the inference time [29].

```
//normal convolution
for(row=0; row<OUTR; row++)
 for(col=0; col<OUTC; col++)
  for(outn=0; outn<OU; outn++){
   for(inn=0; inn<IN; inn++)
    for(i=0; i<K; i++)
     for(j=0; j<K;j++)
      output[outn][row][col]+=
            weight[outn][inn][i][j]*
            input[inn][SW*row+i][SW*col+j];
  }
```

```
// power-of-2 convolution
output[outn][row][col]+=
binaryshift(input[inn][SW*row+i][SW*col+j],
Quantize(log2(weight[outn][inn][i][j])));
```

Fig. 1. The pseudo code of a (power-of-2 quantized) convolutional layer.

B. Power-of-2 quantized CNNs

To reduce the overhead of convolution computing, recent research efforts on CNN algorithms [13], [31] quantize each FP weight into a power-of-2 representation, so that massive expensive matrix-vector multiplications can be replaced by cheap binary shifts and accumulations. In a power-of-2 quantized CNN (P2Q-CNN), $weight^T * input$ is approximately equivalent to $\sum_{i=1}^{n} input_i \times 2^{weightQ_i}$, so $\sum_{i=1}^{n} binaryshift(input_i, weightQ_i)$, where $weightQ_i = Quantize(log_2(weight_i))$, $Quantize(x)$ quantizes x to the closest integer and $binaryshift(a, b)$ shifts a by b bits in fixed-point arithmetic. The pseudo code of a P2Q-CNN layer can be viewed in Figure 1. Although a P2Q-CNN significantly reduces the convolution computing overhead, its inference accuracy degrades by only $< 1\%$ [13], [31].

C. Silicon microdisk resonator

With Moore's law approaching its end, silicon-on-insulator (SOI) photonic devices have gained substantial attention, due to its CMOS-compatibility, ultra-low latency and power dissipation. Among all photonic devices, silicon microdisk technology [24], [27] emerges as one of the most promising solutions for future photonic computing, owing to the small footprint (typically $25\mu m^2$) and ultra-low power consumption (i.e., $1fJ/bit$) [30]. As shown in Figure 2, we will utilize two microdisk structures in Holylight. Figure 2(a) shows an all-pass resonators where a microdisk is coupled into a waveguide. For an off-state resonator, optical signals are totally resonated into the microdisk and no power is transmitted to the Through port. When the resonator is turned on by applying a forward-bias voltage, the optical signals will be modulated and propagated to the Through port, as shown in Figure 3(a). The microdisk-based resonator can be further used as a multiplier: the output power is equal to the product of the power of the input signal and the modulated transmissivity. Figure 2(b) shows an add-drop microdisk resonator which is used for a crossing switch. When the crossing switch is configured to be switched off, the optical signal from the Input port will be delivered to the Through port. Otherwise, the optical signal is resonated into the microdisk and delivered to the Drop port. The transmission spectrums of the switch at the Through and Drop ports are illustrated in Figure 3(b).

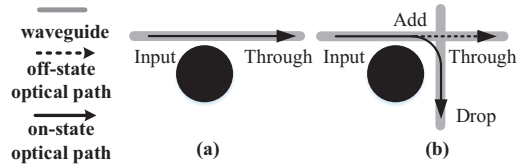


Fig. 2. (a) All-pass and (b) add-drop microdisk resonators.

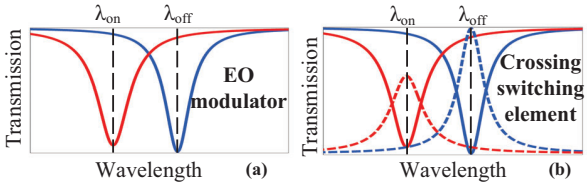


Fig. 3. Transmission spectrum of (a) an all-pass resonator and (b) an add-drop resonator at different ports (solid line: through port; dashed line: drop port).

D. Prior art

To accelerate NN inferences with low power consumption, recent works create fully-optical accelerators by microring weight banks [22], Mach-Zehnder interferometers [20] and multilayer diffractive optical elements [12]. Compared to these optical devices, microdisks enjoys smaller chip area and lower power consumption [27]. Moreover, fully-optical NN accelerators cannot achieve the state-of-the-art inference accuracy even when processing tiny NN models on small datesets, i.e., they obtain only $93\% \sim 97\%$ accuracy on small MNIST dataset, while recent CNNs easily reach $> 99.77\%$ [10] accuracy on the same dataset. This is because their pure analog computing style inevitably introduces noises accumulated during fully-optical NN inferences. The large accuracy degradation prevents fully-optical NN accelerators

from practical deployments in data centers. Moreover, no exisiting photonic CNN accelerator can inference with the state-of-the-art ImageNet dataset.

III. HolyLight

Photonic matrix-vector multipliers (MVMs), adders and shifters are the fundamental computing components for CNN and P2Q-CNN inferences. We first use optical microdisks to build, model and simulate these photonic computing components. And then, we create HolyLight to enhance the CNN inference throughput by microdisk-based MVMs, adders and shifters. We present *HolyLight-M* using photonic MVMs for convolution acceleration. But we later identify the ADCs are the critical bottleneck of HolyLight-M seriously limiting its inference computing efficiency. To unleash the computing power of photonic microdisks, we propose *HolyLight-A* that adopts only photonic adders and shifters to boost the P2Q-CNN inference throughput per Watt with negligible accuracy loss.

A. Computing unit construction and modeling

1) Photonic matrix-vector multiplier: A microdisk-based photonic MVM relies on an array of microdisks, photodetectors (PDs) and a wavelength multiplexer to conduct parallel matrix-vector multiplications. Unlike a recent microring-based counterpart [25], we adopt microdisks in an MVM to reduce its area overhead and power consumption. The microdisk-based MVM architecture is shown in Figure 4. To calculate the multiplication between matrix A and vector B, elements in matrix A are represented by the transmissivity of an $N \times N$ microdisk resonator array, while elements in vector B are denoted by the MVM input power values produced by N laser diodes (LDs) with N wavelengths $(\lambda_1, \cdots, \lambda_N)$ and modulated by N microdisk resonators. Vector B is multiplexed, split and transmitted to each row of the matrix equally. A microdisk resonator with a specific resonance wavelength in a row of the microdisk matrix only reacts on the input signal in the same wavelength, and the output power will be the product between the input power and the modulated transmissivity. The N multiplications between the row vector and the vector B are indicated by the output signals of the N microdisks in N different wavelengths in the row. The corresponding element in the result vector of $A \times B$ is the accumulation of the N output power values and this is performed by the PD who collects the power values of the N signals in different wavelengths. The same operations occur in all rows, so the photonic MVM enables all computing finished in high parallelism. We simulated a 1.28GHz 4×4 photonic MVM in Figure 6(a). Electrical pulses are applied to produce four elements in the first row of the MVM as (b_1, b_2, b_3, b_4). And then we issued input lights modulated by $A_1(a_{11}, a_{12}, a_{13}, a_{14})$ to the MVM. The vector multiplication results $A_1 \cdot B$ are exhibited in the last row of Figure 6(a).

2) Photonic adder: We adopt a 16-bit ripple-carry adder consisting of 16 microdisk-based 1-bit EO (electro-optic) full adders [27], [28]. Compared to the electrical counterpart, a microdisk-based photonic adder greatly enhances its operating

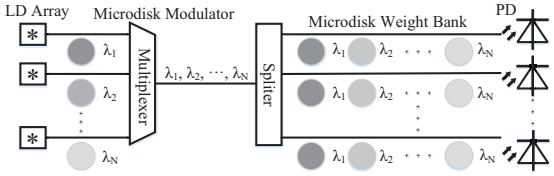


Fig. 4. An on-chip photonic MVM.

frequency. The carry and sum calculation in an n-bit adder can be summarized as $C_n = (A_n \oplus B_n) \cdot C_{n-1} + A_n \cdot B_n = P_n \cdot C_{n-1} + G_n$ and $S_n = C_{n-1} \oplus (A_n \oplus B_n) = C_{n-1} \oplus P_n$, where P_n is the propagate bit and G_n is the generate bit. We use CMOS logic gates to compute P_n and G_n but photonic microdisks to calculate the carry and sum, since the critical path of a 16-bit ripple-carry adder is formed by the carry calculations on all the 16 1-bit full adders. A microdisk EO full adder is shown in Figure 5(a). Two optical signals are injected to a full adder, one representing C_{n-1} and the other having the wavelength λ. Both input signals are separated into two halves by splitters. The electrically obtained G_n and P_n are used to modulate the three microdisks. By tuning light phase and intensity, one optical combiner implements an XOR gate to produce the sum, while the other is used as an OR gate to generate the carry. The simulation waveform of a 50GHz 1-bit microdisk-based full adder is highlighted in Figure 6(b). The performance of a 16-bit ripple-carry adder is substantially improved by using microdisks along the critical path. The 16-bit microdisk-based adder can be reliably operated at 12.8GHz [27], [28].

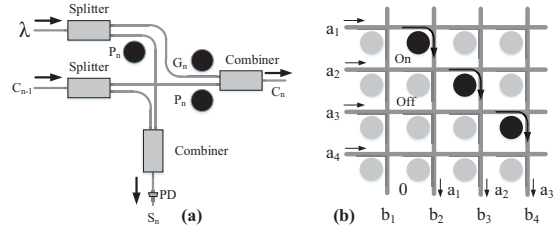


Fig. 5. Microdisk-based (a) EO full adder and (b) photonic shifter.

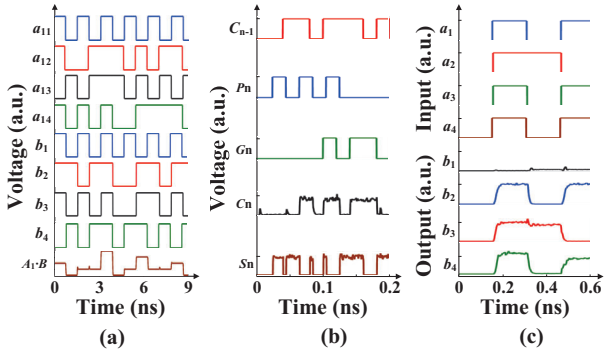
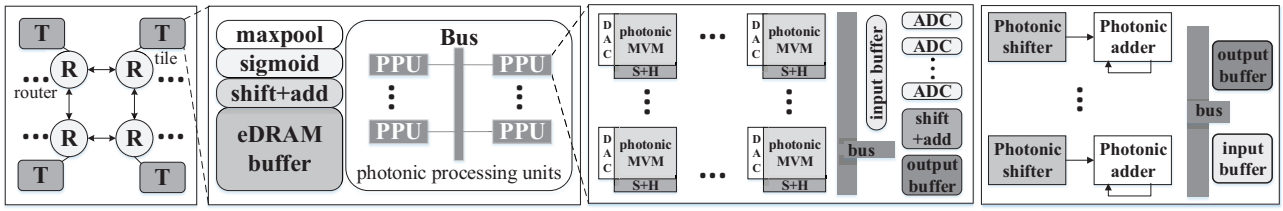


Fig. 6. The simulation waveforms for (a) a 1.28GHz photonic MVM; (b) a 1-bit 50GHz photonic full adder; and (c) a 4-bit 12.8GHz photonic shifter.

3) Photonic shifter: We build a microdisk-based binary shifter. Figure 5(b) shows a 1-bit logical right-shift operation. By configuring the microdisk crossing switches in the array, the photonic shifter performs n-bit shift operations. Binary input signals $(a_n \ldots a_1)$ are injected to the microdisk array. By configuring the on/off states of the microdisk crossing switches as described in Figure 2(b), the shift result $(b_n \ldots b_1)$ can be obtained. Note that there is no light switched to the column



(a) chip node (b) tile (c) HolyLight-M PPU (d) HolyLight-A PPU

Fig. 7. The architecture of HolyLight-M and HolyLight-A.

of b_1, representing the complemented zero. We simulated a 12.8GHz 4-bit microdisk-based shifter in Figure 6(c), where $a_4 \ldots a_1$ are 4-bit input signals and the microdisks in the shifter are configured into the 1-bit left-shift mode. The 4-bit output $b_4 \ldots b_1$ is successfully shown in Figure 6(c).

B. HolyLight-M

Overview. The chip node architecture of HolyLight is shown in Figure 7(a). A HolyLight chip consists of multiple tiles connected by a network-on-chip (NoC). Each tile described in Figure 7(b) communicates with other tiles by its router. Each tile relies on electrical shift-and-add (S&A) units and eDRAM buffers to aggregate intermediate results generated by photonic MVMs. A tile also possesses sigmoid and max-pooling units for CNN activations and pooling operations. As Figure 7(c) shows, convolutions during CNN inferences are computed by photonic processing units (PPUs) connected through a shared bus. Each PPU consists of multiple photonic MVMs, sample-and-hold (S&H) units, ADCs, input and output buffers, and S&A units.

Computing flow. After a CNN model is sufficiently trained, weight matrices of various layers of the CNN are programmed into photonic MVMs by configuring the transmissivity of the microdisks in each PPU. During CNN inferences, inputs are fetched from eDRAMs and routed to available tiles of HolyLight-M. To collaboratively compute convolutions, the HolyLight-M controller steers photonic MVMs producing intermediate results in the form of voltages by photodetectors. The intermediate result voltages are sampled and held by S&H units, and converted to digital values by ADCs. Digital results are aggregated in output buffers by S&A operations, and sent to activation units to generate layer outputs that will be buffered again in eDRAMs for the next layer processing. The computation continues until the final result is produced.

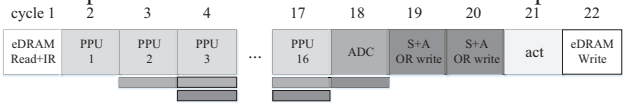


Fig. 8. The HolyLight-M pipeline.

HolyLight-M PPU and its pipeline. A smaller microdisk array in a photonic MVM reduces its computing throughput, while a larger microdisk array degrades the MVM accuracy. Based on our photonic MVM simulations following the MVM design space exploration in [25], we conservatively adopted a 128×128 microdisk array in our photonic MVM to balance the tradeoff. Although a microring-based MVM [25] suggests each weight bank microring can store > 4 bits, we found large MVM accuracy degradations appear in our microdisk-based MVM when using the same configuration. Therefore, we conservatively use the transmissivity of each microdisk to represent only 2 bits of a 16-bit weight to maintain no MVM accuracy degradation. We convert each digital input bit to a voltage by 1-bit DAC. Then, each voltage is applied on a laser diode in the MVM to start a convolution. As Figure 8 describes, in the HolyLight-M PPU pipeline, one cycle for input fetching from eDRAMs, 16 cycles for a convolution, one cycle for ADC processing, two cycles for S&A operations, one cycle for activation and one cycle for output write-back. Totally, one convolution costs 22 pipeline cycles.

Bottleneck analysis. Photonic MVMs accelerate convolutions by analog optical computing, so HolyLight-M requires DACs and ADCs to handle conversions between digital values and optical signals. Such conversions effectively avoid the accumulation and propagation of noises between analog optical operations. Although a microdisk can run at 50Gbit per second (bps) [28], we have to set its throughput to 1.28Gbps and the HolyLight-M pipeline frequency to 1.28GHz, since higher microdisk throughput asks for more power-hungry ADCs. We have to provide a 1.28GSamples per second (GSps) 8-bit ADC [16] to a column of 128 microdisks in an MVM. ADCs consume 85.7% of the HolyLight-M total power consumption. So we identify the ADC as the performance and power bottleneck of HolyLight-M.

C. HolyLight-A

Overview and computing flow. To overcome the ADC bottleneck, we architect another nanophotonic accelerator, HolyLight-A, to boost the inference throughput per Watt of P2Q-CNNs that have only shifts and additions with only negligible accuracy degradation. The overall HolyLight-A architecture is the same as HolyLight-M but has a different PPU design shown in Figure 7(d). After the P2Q-CNN training, power-of-2 quantized (P2Q) weights are distributed to allocated tiles by a NoC and stored in their eDRAM buffers. During inferences, these P2Q weights are used to control how many bits the inputs should be shifted. The inputs are sent to the photonic shifters for power-of-2 multiplications (shifts), and accumulated by photonic adders to produce final outputs. Due to the P2Q-CNN inference algorithm, HolyLight-A does not need S&A units used in HolyLight-M. Moreover, HolyLight-A eliminates all ADCs, because of its digital optical computing style.

HolyLight-A PPU and its pipeline. In a PPU, HolyLight-A has multiple photonic shifters and adders connected by a shared bus. The weights, inputs and accumulated intermediate results are stored in input buffers while final outputs are recorded in output buffers. Since P2Q weights are trained into 16 bits, HolyLight-A can share a similar pipeline to

HolyLight-M, but performs different operations from the second cycle to the 17th cycle. During these 16 cycles, HolyLight-A shifts inputs according to each weight bit, and accumulates shifted inputs to final outputs. Since a photodetector recognizes a 1-bit output signal in sufficiently high accuracy, the ADC arrays of HolyLight-M are removed in HolyLight-A. HolyLight-A can skip the 18th, 19th and 20th cycles in the HolyLight-M pipeline and directly performs activations. Totally, the HolyLight-A pipeline has only 19 cycles. By eliminating the ADC bottleneck, the operating frequency of HolyLight-A can match the frequency of photonic adders and shifters, i.e., 12.8GHz.

Buffer. HolyLight-A runs $9\times$ faster over HolyLight-M. So we also need to boost the access bandwidth of HolyLight-A buffers to match the computing throughput of HolyLight-A. To this end, we increase the number of sense amplifiers and write drivers in eDRAM buffers and SRAM input/output buffers by $9\times$. Although these buffer peripheral circuits still run at 1.28GHz, the maximal buffer access frequency can reach 12.8GHz by interleaving accesses among all peripherals.

TABLE I
HOLYLIGHT POWER AND AREA

Name	Component	Spec	Power (mW)	Area (mm^2)
128×128 matrix-vector multiplier (MVM)	E-O disk(1.28Gbps)	$\times 16384$	20.97	0.4096
	splitter&mux	$\times 128$	0	0.642
	PD(1.28Gbps)	$\times 128$	8.192	0.0018
	disk resonator	$\times 128$	0.164	0.0032
sub-total			29.33	1.0569
1.28GHz PPU-M	ADC	$\times 1024$	2048	1.2288
	DAC	$\times 1024$	8	0.0004
	S+H	$\times 1024$	0.01	0.00004
	MVM	$\times 8$	234.64	8.4552
	(S+A)-M	$\times 8$	0.2	0.00024
	out&in-put buffer	2.25KB	1.47	0.00287
sub-total			2292.32	9.688
1-bit full adder (FA)	G_n&P_n gates	$\times 1$	0.00001	0.000002
	E-O disk(50Gbps)	$\times 3$	0.15	0.000075
	splitter&combiner	$\times 2$	0	0.000002
	PD(50Gbps)	$\times 1$	2.5	0.000014
sub-total			2.65	0.000091
16-bit adder	FA	$\times 16$, 12.8GHz	42.4	0.001458
12.8GHz PPU-A	16-bit adder	$\times 64$, 12.8GHz	2713.6	0.093
	16-bit shifter	$\times 64$, 12.8GHz	13.11	0.4096
	out&in-put buffer	2.25KB	2.21	0.0043
sub-total			2728.92	0.5069
tile	eDRAM-M (A)	256KB	41.4(62.4)	0.166(0.268)
	bus	384-wire	7	0.009
	router	32-flit, 8-port	42	0.151
	sigmoid	$\times 2$	0.52	0.0006
	maxpool	$\times 1$	0.4	0.00024
	(S+A)-M	$\times 1$	0.05	0.00006
sub-total-M(A)			91.4(112.3)	0.327(0.429)
I/O interface	optical interface	4-link, BW: 8GB/s	140.18	0.0244
total-M		28-tile	66.9 W	280.42
total-A		24-tile	68.3 W	22.46

D. Design Overhead

The power and area overhead of the two configurations of HolyLight is shown in Table I. To compare against the ReRAM-based CNN accelerator ISAAC [19], we adopted its CMOS ADC, DAC, router, bus, sample and hold (S&H), shift and add (S&A), pooling, activation logic designs in HolyLight. All CMOS logic units are modeled and estimated through Cadence Virtuoso with $32nm$ PTM technology. The eDRAM, input and output buffers are modeled by CACTI. HolyLight uses electrical router inside each tile and a photonic I/O interface [26] to communicate CPUs. Compared to a photonic router, we observed an electrical router minimize smaller power, due to each tile's small bandwidth need. To simulate photonic microdisk-based computing components, we used Lumerical FDTD [14] and INTERCONNECT [15] simulation infrastructure. To build HolyLight, we modeled and adopted optical splitters & combiners from [23], optical multiplexers from [2], photodetectors from [1] and microdisks from [28].

IV. EXPERIMENT METHODOLOGY

Workload. We studied six CNNs including LeNet-5 [11], CNP [6], SCNN [21], MCDNN [5], AlexNet [10] and ResNet-18 [7]. LeNet-5, CNP, SCNN and MCDNN were trained with MNIST to identify simple handwritten digits, while AlexNet and ResNet-18 were trained with ImageNet to recognize complex objects. We trained all networks by Torch7. More network details can be viewed in Table II, where besides the network topologies, we also show the inference accuracy of each network. Compared to full precision CNNs ($Orig$), P2Q-CNN [13], [31] reduces the test accuracy (top-5) of AlexNet and ResNet-18 by $< 1\%$.

TABLE II
CNN BENCHMARKS (C: CONVOLUTIONAL; P: POOLING; F: FULLY-CONNECTED; $Orig$: ORIGINAL; $P2Q$: POWER-OF-2 QUANTIZED CNN).

Name	DataBase	Topology	Acc (%)	
			Orig	*P2Q*
LeNet-5	MNIST	3C,2P,1F	99.1	98.9
CNP	MNIST	3C,2P,1F	97.0	96.9
SCNN	MNIST	2C,2F	99.0	99.0
MCNN	MNIST	3C,3P,3F	96.8	96.3
AlexNet	ImageNet	5C,3P,2F	80.2	79.4
ResNet-18	ImageNet	18C,2P,1F	89.2	88.6

TABLE III
SIMULATED SCHEME COMPARISON.

Name	Description	Power (W)
CPU	Intel Xeon E5	130
GPU	Nvidia Tesla P100	250
FPGA	Xilinx Virtex7 VX485T	40
DaDianNao	simple ASIC	20.1
TPU	ASIC, 4-chip	384
ISAAC	ReRAM CNN	65.8
HolyLight-M	Photonic CNN	66.9
HolyLight-A	Photonic CNN	68.3

Schemes. We compared HolyLight-M&A against 6 counterparts shown in Table III. We selected an Intel Xeon CPU, an Nvidia Tesla GPU, a Xilinx Virtex7 FPGA [29], a DaDianNao ASIC chip [4], a Google TPU [9], and a ReRAM-based CNN accelerator ISAAC [19]. Compared to DaDianNao, a Google TPU consists of four chips, each of which is more powerful and costs larger power consumption. ISAAC relies on ReRAM-based dot-product engines to accelerate matrix-vector multiplications. HolyLight-M uses a similar pipeline to ISAAC, but replaces ReRAM arrays by optical microdisk matrices. In contrast, HolyLight-A depends on photonic adders and shifts to perform P2Q-CNN inferences without ADCs.

Accelerator modeling. We used a heavily modified deep learning accelerator simulator FODLAM [18] to study the performance, power and energy consumption of all accelerators. Based on a user-defined accelerator configuration and a neutral network description, FODLAM can generate the performance, power and energy details of the accelerator running that network. The FODLAM model has been correlated

and validated by physical accelerator chips such as Google TPU. We integrated micro-architectural pipeline details of our accelerators into FODLAM.

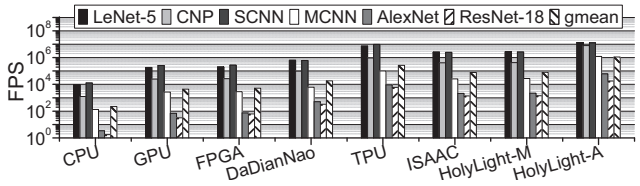


Fig. 9. The CNN inference performance of various hardware platforms.

V. Evaluation

Inference performance. The CNN inference performance comparison of various accelerators is shown in Figure 9. ASIC designs, e.g., DaDianNao and TPU, achieve higher frame per second (FPS) when inferencing with both MNIST and ImageNet, where a frame is an image, because of their highly efficient application specific design style. Compared to the single-node DaDianNao, the 4-chip Google TPU improves the CNN inference performance by $13\times$. Although ISAAC uses emerging ReRAM-based dot-product engines to accelerate matrix-vector multiplications, it has less hardware resources than TPU and hence obtains smaller FPS. HolyLight-M slightly increases the inference performance by 5% over ISAAC, since similar to ISAAC, it still depends on power hungry ADCs. On the contrary, HolyLight-A improves the CNN inference FPS by $13.5\times$ over ISAAC and by $3.1\times$ over TPU, because it performs CNN inferences by only photonic additions and shifts without ADCs.

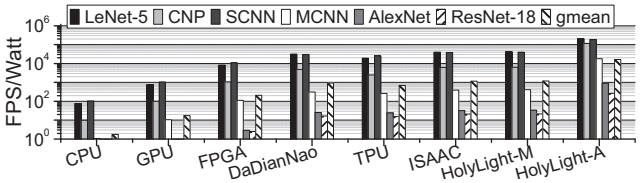


Fig. 10. The flops/watt results of different designs.

Performance per Watt. The CNN inference performance per Watt comparison of various accelerators is exhibited in Figure 10. The performance per Watt of CPU, GPU, FGPA and ASIC shares a similar trend to their performance. But ReRAM-based ISAAC obtains much better FPS/Watt than traditional CMOS DaDianNao and TPU, due to its ReRAM analog computing dot-product engines. HolyLight-M improves the FPS/Watt by only 3.2% over ISAAC, because they share the same performance and power bottleneck, i.e., the ADCs. In contrast, HolyLight-A removing the ADC bottleneck boosts the CNN inference FPS per Watt by $13\times$ over ISAAC.

VI. Conclusion

In this paper, we present HolyLight to accelerate CNN inference in data centers. We first propose HolyLight-M on top of microdisk-based matrix-vector multipliers. We observe its CNN inference throughput per Watt is seriously limited by the ADCs. Without ADCs, we further architect HolyLight-A for power-of-2 quantized CNNs by only using microdisk-based adders and shifters. Compared to ReRAM-based accelerator, HolyLight-A improves the CNN inference throughput per Watt by $13\times$ with only $< 1\%$ accuracy degradation.

Acknowledgment

This work is partially supported by NAP M4082282 and SUG M4082087 from Nanyang Technological University, Singapore and NSFC 61772094, China.

References

[1] S. Assefa, *et al.*, "CMOS-integrated high-speed MSM germanium waveguide photodetector," *Optics Express*, 2010.
[2] S. Chen, *et al.*, "Compact dense wavelength-division (de) multiplexer utilizing a bidirectional arrayed-waveguide grating integrated with a Mach–Zehnder interferometer," *Journal of Lightwave Technology*, 2015.
[3] X. Chen and X. Lin, "Big Data Deep Learning: Challenges and Perspectives," *IEEE Access*, 2014.
[4] Y. Chen, *et al.*, "DaDianNao: A Machine-Learning Supercomputer," in *MICRO*, 2014.
[5] D. Ciregan, *et al.*, "Multi-column deep neural networks for image classification," in *CVPR*, 2012.
[6] C. Farabet, *et al.*, "CNP: An FPGA-based processor for Convolutional Networks," in *FPL*, 2009.
[7] K. He, *et al.*, "Deep Residual Learning for Image Recognition," in *CVPR*, 2016.
[8] L. Jiang, *et al.*, "XNOR-POP: A processing-in-memory architecture for binary Convolutional Neural Networks in Wide-IO2 DRAMs," in *ISLPED*, 2017.
[9] N. P. Jouppi, *et al.*, "In-datacenter performance analysis of a tensor processing unit," in *ISCA*, 2017.
[10] A. Krizhevsky, *et al.*, "ImageNet Classification with Deep Convolutional Neural Networks," in *NIPS*, 2012.
[11] Y. Lecun, *et al.*, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, 86(11), Nov 1998.
[12] X. Lin, *et al.*, "All-optical machine learning using diffractive deep neural networks," *Science*, 2018.
[13] Z. Lin, *et al.*, "Neural networks with few multiplications," in *ICLR*, 2016.
[14] Lumerical, "FDTD Solutions," `http://www.lumerical.com/tcad-products/fdtd/`.
[15] Lumerical, "INTERCONNECT," `http://www.lumerical.com/tcad-products/interconnect/`.
[16] B. Murmann, "An ADC Performance, Power and Area Survey from 1997 to 2017," http://web.stanford.edu/ murmann/adcsurvey.html.
[17] R. Raghavendra, *et al.*, "No "Power" Struggles: Coordinated Multi-level Power Management for the Data Center," in *ASPLOS*, 2008.
[18] A. Sampson and M. Buckler, "FODLAM: a first-order deep learning accelerator model," `https://github.com/cucapra/fodlam`.
[19] A. Shafiee, *et al.*, "ISAAC: A Convolutional Neural Network Accelerator with In-Situ Analog Arithmetic in Crossbars," in *ISCA*, 2016.
[20] Y. Shen, *et al.*, "Deep learning with coherent nanophotonic circuits," *Nature Photonics*, 2017.
[21] P. Y. Simard, *et al.*, "Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis," in *ICDAR*, 2003.
[22] A. N. Tait, *et al.*, "Neuromorphic photonic networks using silicon photonic weight banks," *Scientific Reports*, 2017.
[23] J. Wang, *et al.*, "Sub-μm 2 power splitters by using silicon hybrid plasmonic waveguides," *Optics express*, 2011.
[24] L. Yang, *et al.*, "On-chip CMOS-compatible optical signal processor," *Optics express*, 20(12), 2012.
[25] L. Yang, *et al.*, "On-chip optical matrix-vector multiplier for parallel computation," in *SPIE*, 2013.
[26] Y. Ye, *et al.*, "3D optical networks-on-chip for multiprocessor systems-on-chip," in *3DIC*, 2009.
[27] Z. Ying, *et al.*, "Electro-Optic Ripple-Carry Adder in Integrated Silicon Photonics for Optical Computing," *IEEE Journal of Selected Topics in Quantum Electronics*, 2018.
[28] Z. Ying, *et al.*, "Silicon microdisk-based full adders for optical computing," *Optics letters*, 43(5), 2018.
[29] C. Zhang, *et al.*, "Optimizing FPGA-based Accelerator Design for Deep Convolutional Neural Networks," in *FPGA*, 2015.
[30] Z. Zhao, *et al.*, "Optical computing on silicon-on-insulator-based photonic integrated circuits," in *ASICON*, 2017.
[31] A. Zhou, *et al.*, "Incremental network quantization: Towards lossless cnns with low-precision weights," *ICLR*, 2017.