Native Digital Processing for Optical Networking

Joe Touch
Information Sciences Institute
University of Southern California
Marina del Rey, CA, USA
touch@isi.edu

Alan E. Willner
Ming Hsieh Dept. of EE/Systems
University of Southern California
Los Angeles, CA, USA
willner@usc.edu

***Abstract*—The Optical Turing Machine (OTM) is an approach to digital optical processing that supports computation in the same format used for high-speed transmission. This paper identifies the key capabilities required to support native digital optical processing for typical in-network functions including forwarding, security, and filtering. Current analog and binary digital approaches – including optical transistors – are considered and shown insufficient for optical networks. The requirements for a single encoding are presented, as are the capabilities required for network computation.**

Keywords*— Optical processing, optical computation, digital optics, network processing, packet processing, network security.

I. Introduction

Network communication increasingly relies on the optics to support high data rates over long distances. At the same time, network functions are becoming more prevalent, *e.g.*, Software-Defined Networking (SDN), network security, and data filtering [5]. The transmission benefits of optics can be an impediment to implementing computational functions using current approaches.

The Optical Turing Machine (OTM) is a new effort intended to enable in-network digital processing of long-distance optical transmission [20]. Unlike other approaches to optical computation, OTM focuses on the unique need for digital optical computation to implement in-network functions. OTM attempts to unify communication and computation using a single, common optical encoding.

This paper presents the case for this unification, discusses the need for in-network computation, and explains some key capabilities required for the OTM approach in comparison to other past and recent approaches to optical computation.

II. Background

The performance of computation and communication has each increased by several orders of magnitude in the past few decades. Both benefit from increased state-change frequency (processor clock rate, transmission modulation rate) and independent concurrency (multiple cores, different wavelengths). Computation also achieves significant speedup using parallel binary encodings, whereas communication leverages increased density of serial encoding (*e.g.*, Quadrature Amplitude Modulation, *i.e.*, QAM), as shown in Fig. 1.

This work is partly supported by National Science Foundation under Contract (1344221). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

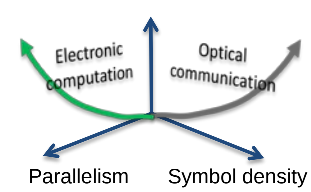

Fig. 1. Electronics *vs*. optical speed-up.

It is currently common to compute using electronics and communicate using optics. Electronic switching can be fast (2 ps), integrated easily (4 G transistors per device), and support bit- and device-level parallelism (128-bit processors; 500 graphics cores). However, high-speed electronic signals do not propagate well; signals of 10 Gb/s over more than 10m are necessarily optical.

Optics is more efficient for long-distance, high-speed communication. Unamplified signals can propagate tens of km, each symbol representing several bits, encoding 40Gb/s on each of dozens of wavelengths. However, it is difficult to integrate more than a handful of components into a single circuit and optical switches often operate in the ms-μs range.

The electronic and optical approaches conflict when we consider functions that occur on data in-transit. A hybrid approach using electronics for computation and optics for communication requires optical-electrical-optical (OEO) conversion, which is complex, expensive, and can waste energy. A unified approach would avoid these issues, but only if a single encoding were used for both communication and computation. High-speed, long-distance communication is necessarily optical [7], so the encoding must be optical too.

III. The Need for In-Network Computation

Networks can transfer data opaquely, but increasingly they support in-network computation involving data meta-information (headers) or the communicated data itself. The most common examples are packet processing, network security, and data filtering.

A. Packet Processing

Networks use either circuits or packets; circuits are more efficient when traffic patterns are known in advance, but packets (whether fixed or variable length) are more efficient for unpredictable patterns, and are thus preferred, where possible. Packet processing can include switching, indexing addresses, adjusting hop counts, managing checksums, and de-conflicting overlap - all necessarily occurring inside the network where traffic from different sources combines to share resources (Fig. 2) [6][9][14][24]. This processing can include encapsulation (and decapsulation) or direct translation, supporting virtual networks, SDN, and network address translation (NATs) [5][16].

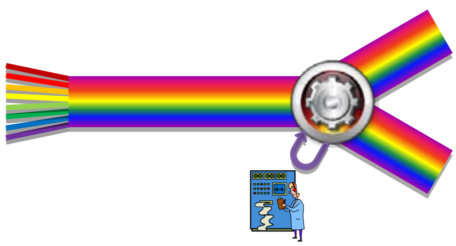

Fig. 2. Packet processing modifies data in-transit to change its direction.

B. Network Security

Data transiting a network is vulnerable to tampering and copying, and thus is often processed to include integrity protection or encryption [5]. Such security is often applied between the communicating endpoints, but additional layers of security can ensure protection between enterprises or when transiting an untrusted area (Fig. 3). This additional protection can only be implemented inside the network, at the boundaries between trusted network components. Network security also protects the network itself from both control-plane attacks and data-plane denial-of-service (DOS) attacks.

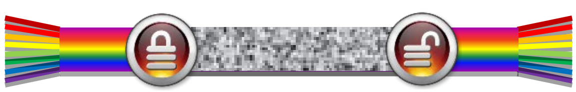

Fig. 3. Network security modifies data in transit to protect it.

C. Data Filtering

The "big data" trend involves collecting and exploring large amounts of information, some collected in advance and others collected on-the-fly. It can be impractical to store these data sets for off-line processing; instead, an on-line approach can digest large streams, either completely or as a preprocessing reduction step. In-network processing enables on-line filtering or coalescing of these streams (Fig. 4). Note that filtering need not be perfect; a low occurrence of false positives can be tolerated if false negatives can be avoided.

Fig. 4. Data filtering extracts a subset of data in transit.

IV. Past and Related Approaches

There are several approaches to optical computation, many intended to exceed the capabilities of electronic computation [3]. In contrast, OTM focuses on optical computation to implement in-network functions, specifically using formats capable of long-distance high-speed communication.

Most current approaches to optical computation use binary encodings processed using transistor-like switching [1][13][17]. These approaches are expected to support high-speed through parallelism, but optical communication is typically serial and multibit. Switched binary processing for in-network functions would require costly conversion between these encodings that might be as costly as current optical-electronic-optical (OEO) conversion.

Additionally, feature sizes suggest that optical processing will be limited to domains where signals are already photonic. Electronics switching relies on the properties of individual electrons, with a limiting diameter near 10 fm (1E-14m); optics operates on wavelengths in the range of 1 μm (1E-6m) [28]. Electronics can potentially be up to eight orders of magnitude (100 billion times) smaller - a difference that is compounded when considering 2-dimensional integration.

A. Analog vs. Digital

Optics natively supports analog processing, *e.g.*, 2-dimensional Fourier transform processing using lenses or phase-encoded information in holograms [22]. Fourier transforms are useful for correlation and convolution, and holograms are useful for storage in 3-dimensional bulk media.

However, analog functions cannot be repeatedly composed, wherein the output of one function becomes the input to the next. Errors in analog systems propagate and can be amplified, limiting the overall complexity of analog processing. This is why both communication and computation are digital; by limiting the signal to a fixed set of values, some errors can be eliminated even when composing a sequence of functions.

B. Binary vs. Multibit

Binary encodings use two values, such as on-off keying (OOK) or binary phase-shift keying (PSK). Binary digital optical computation has been under active investigation for over 60 years, often seeking an "optical transistor" (see Sub-section C below) [2][10].

Some approaches to digital optical processing have used 2-dimensional switching using a spatial light modulator (SLM) [3][8][11]. SLM devices typically require free-space transmission, three-dimensional system architectures, and operate at very slow rates due to the slow switching speed of modulators (1 ms - 1 μs). SLMs also cannot support more general computations needed for in-network functions, nor do they natively support serial network data [18]. A more exotic form of digital optics involves quantum encoding, but the processing expectations for quantum networks have only recently been considered [26].

High-speed optical communication uses symbols that represent multiple bits, *e.g.*, 16QAM using eight phases and three power levels to represent four bits of information. Computation that supports network processing of high-speed optical signals thus needs to support multibit encodings [4][7].

C. Switched vs. Transformational

Most electronic computation uses switching, wherein an input signal is used to control whether the output is connected to one of two or more other signal sources. These sources are typically the direct-current anode or cathode, so the output either sources or sinks electrons *via* the power supply. The input signal stops at the switch; the output is controlled by, but not a direct derivation of the input (Fig. 5, left).

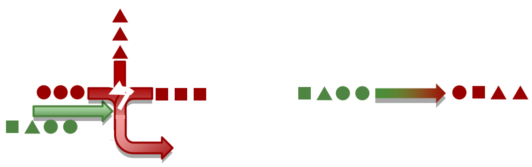

Fig. 5. Electronics uses switching (left); optics transforms its input (right).

Optics also supports switching, but often far more slowly than electronics (15ps for 22nm CMOS [19]), typically because optical switching often relies on a mechanical mirrors (1ms), bulk thermal (1µs), or electro-optic properties (1ns). High-speed optical processing more typically relies on transforms in which the input waveform is converted to an output waveform (Fig. 5, right), such as with four-wave mixing - which are limited by frequency bandwidth rather than symbol rates.

D. Parallel vs. Serial

Most electronic computation relies on a significant parallelism - 64b is common, with larger word sizes used in graphics and network processors. This parallelism is used to overcome the limited symbol rate of individual components - *e.g.*, typically 3-6 Gb/s. Some functions that are the most complex when parallel become nearly trivial when serialized; addition has *O*(N*log*N) elements and *O*(*log*N) delays for N-bit summands (Fig. 6, left), whereas the same operation in serial optics requires one element and its output is delayed by one symbol (Fig. 6, right).

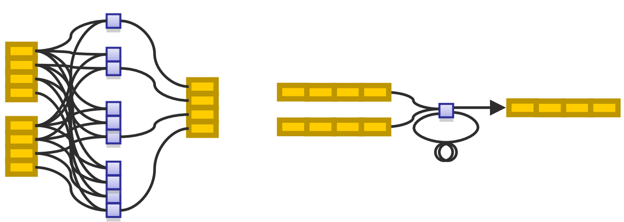

Fig. 6. Comparison of adder complexity: parallel (left) and serial (right).

Many of the functions important for in-network processing support efficient serial implementations, including:

- Statistics (sums, averages, standard deviations, *etc.*)
- Pattern matching (correlation)
- Decrement and drop if zero (hop count processing)
- Error processing (checksums, CRCs)
- Crypto processing (authentication, encryption)

V. The Optical Turing Machine Approach

OTM supports digital computation for in-network functions using multibit encodings *via* serial, transformational optical processing. It uses a single encoding format that is compatible with both transmission and computation to avoid costly translation. The following describes the approach currently being used to develop OTM based on our recent observations.

A. Impact of Encoding on Computation

Information being transmitted or processed is digitally represented in an encoding that consists of a set of discrete symbols and a value mapping. Symbols include optical phase, (*e.g.*, PSK, Fig. 7, left) and {phase,amplitude} pairs (*e.g.*, QAM, Fig. 7, right). Value maps assign specific information to each symbol (*e.g.*, numbers in Fig. 7).

Different encodings can enable or impede computation [11]. Consider the "+1" Hamiltonian path (a path through a sequence of physical states) whose values increase by 1 (shown as the dashed paths in Fig. 7). PSK's path is continuous, uniform, and unambiguous; the transitions are in the same relative direction, each transition is the same amount of relative phase shift, and the path never crosses itself. Rectilinear QAM's path has two kinds of discontinuities - short jumps (3-4, 7-8, 11-12), and one large jump (15-0). None of the transitions are similar as either absolute or relative amplitude:phase transforms. Finally, the QAM path crosses itself in five different places.

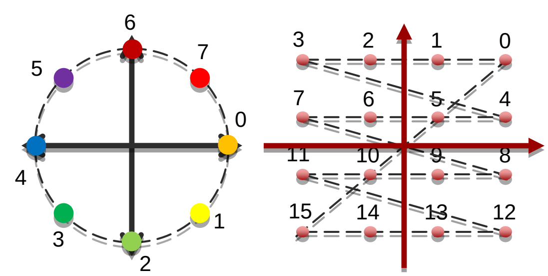

Fig. 7. Phase (left) and QAM (right) encodings; the +1 Hamiltonian is shown as a dashed path.

Computation requires translation of one set of symbols into another - a function. Translation can be performed by switching or transformation, and, as discussed, optics prefers transformation. Transformation processing is simpler with continuous, uniform Hamiltonians. Such processing strictly requires unambiguous Hamiltonians; otherwise, on the way between transitions, the transformation device would enter a state with multiple outcomes, and the result of the computation would be ambiguous.

A variety of encodings have been considered for OTM, and compared as to their ability to support computation and communication. We seek the following properties:

- ***Symbols that can represent multiple bits***: to efficiently support high-speed, long-distance communication.
- ***Symbols that can be processed using transformational optics,*** *e.g.*, four-wave mixing, to support high-speed serial processing.

- ***Encodings with Hamiltonian paths that are***:
  - ***Continuous***: avoiding changes in direction.
  - ***Uniform***: represented by some combination of absolute and relative transformations.
  - ***Unambiguous***: transformations guaranteed to have a single, known outcome.

The need for multibit symbols excludes binary encodings such as OOK, binary PSK, and polarization. The need for transformational processing favors one-dimensional encodings, *e.g.*, PSK or multi-amplitude, rather than those that vary multiple properties, *e.g.*, QAM and code division multiple access (CDMA). This is further emphasized by the Hamiltonian path requirements. The N-value PSK (N-PSK) encoding satisfies these requirements, but we continue to explore non-traditional encodings that might also suffice.

B. Leverage Properties of Optics

A goal of OTM is to leverage the unique properties of optics. This has been the focus of repeated investigations [10][12]. Our approach is distinct in focusing on first principles the theory of computation and discrete mathematics and considering the unification of computation and communication.

OTM focuses on the use of multibit symbols to enable efficient high-speed, long-distance communication. As a result, binary logical operations are considered insufficient, even when transformational logic is considered [10]; their use would require costly conversion between efficient transmission encoding and binary computation encoding.

OTM addresses the requirements for computation, distilled from the abstract Turing machine mechanism combined with the need for recurrence relations and conditional operation from computational theory. Other approaches to optical computing discuss the capabilities of optics as functions independent of their necessity for computation [12].

C. Key Components

We have identified four key capabilities required for digital optical network processing, based on the need to compose digital processing operations.

Re-digitization - Also known as restoration, re-digitization is critical for error detection and correction. It enables long-distance communication by recovering digital values before they become ambiguous and supports creating complex functions by the composition of simpler ones [25]. This is an active area of current optical processing research [23][29].

Chainable operations - Computational processing requires functions that support group operations. In discrete mathematics, a *group* is a set of symbols and two binary operators (*i.e.*, 2-input functions). The symbol set is closed under both operators, each operator has an identity element ($a \bullet I = a$), and every element of the set an inverse under each operator ($a \bullet a^{-1} = I$). The operators are commutative ($a \bullet b = b \bullet a$), and one operator is distributive over the other ($a \bullet (b \circ c) = (a \bullet b) \circ (b \bullet c)$). Chainable operations support mathematical operations such as accumulation and hop count decrement and are needed for group-based manipulations such as checksums, encryption, and authentication. Recent work is developing support for these operations on N-PSK encodings, implementing modulus arithmetic [27].

Latch and deflection switch - A latch is the simplest form of state persistence. When coupled with a deflection switch, it enables control of data flow through a serial processing system. Note that we expect the need for latches for control but not in the data path. Deflection switches differ from computational switching elements; the former are similar to pass transistors that control data flow, where the latter are the more conventional use of transistors to gate output values determined from the power supply. The combination of a latch and deflection switch is important for conditional computation.

Ephemeral storage - All computation requires state that persists, even if temporarily. Without persistent state, only simple functions can be generated; recurrence relations (recursive functions) would be prohibited. Recursion is a key requirement for general-purpose computation. There are various forms of such ephemeral storage, some involving fiber loop. There is a large body of work in using ephemeral storage, beginning with acoustic waves in mercury delay lines in early electronic computers and in recent reservoir computing [21].

We also assume a complete reconsideration of computation architecture based on these capabilities combined with the serial nature of communication. Current architectures focus on increased parallelism and synchronous, phased operation due to the use of switched, highly-integrated transistor systems. Digital processing of optical communication streams is better matched to asynchronous serial processing.

D. An Example - Hop Count Processing

The simplest network function is hop count processing. Packets contain a hop count (a.k.a. “time to live”) field to ensure that forwarding loops and misdirected traffic doesn’t overload the network [15]. Each packet is given an initial count (*e.g.*, 255), which is decremented at each hop. When the count reaches zero, the packet is silently discarded. This avoids the need for a separate mechanism to flush out stale packets.

Hop count processing is currently implemented using either a general-purpose CPU or complex, dedicated bit-parallel arithmetic hardware. Subtraction is typically implemented as addition of “-1”, a complex operation involving “carries” that cascade across a parallel bit field.

A simpler approach is possible when processing the data serially, using only three active components [15][24]. Subtraction of an unsigned bit field involves inverting the ‘0’ values (starting at the low-order bit) until the first ‘1’ value is encountered. That value is inverted, and all subsequent inputs are copied (Fig. 8, left). The entire system can be implemented using only three active elements - an inverter, a 2x2 switch, and a set-reset (S/R) flip-flop (FF). The switch is set to select the inverted input until the first ‘1’ bit arrives. That bit sets the flip-flop, whose output is delayed by one symbol to change the switch just before the next symbol. After the change, the input is copied to the output without inversion.

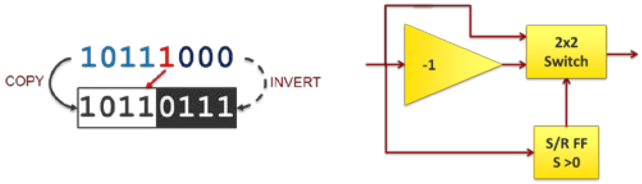

Fig. 8. Serial hop count decrement (left) and its implementation (right).

Our team previously implemented this mechanism for a binary OOK encoding, using an electronic S/R FF. The mechanism has been extended to support multibit encoding, replacing inversion by a "-1" transformation, and all of the optical components already exist for N-PSK encodings. For example, "-1" transform for 8PSK encoding is a -π/4 phase shift which can be accomplished using a fixed difference in signal path, *e.g.*, of 187.5nm for a 1500nm wavelength signal. A complete candidate design would need to further extend a OOK micro-ring flip-flop (such as [30]) to support N-PSK encoding and to identify an optical switch that can be controlled using an independent OOK input (*i.e.*, the flip-flop output). These devices are under active investigation.

VI. Conclusions

This paper presented an approach to optical computation focused on supporting network processing in the native transmission format. This perspective was motivated by the need for network functions for packet processing, network security, and data filtering. Past approaches were evaluated in this context indicating a new goal for digital optical computation based on serial multibit transformational processing. The combined requirements of efficient transmission and computation were analyzed to yield N-PSK as the preferred common encoding, and four key capabilities were identified: re-digitization, chainable operations, latch and deflection switching, and ephemeral storage. The example of hop count processing was used to demonstrate the relevance of these observations. We are currently developing demonstrations of these capabilities based on this analysis.

References

[1] E. Abraham, C. Seaton, and S. Smith, "The Optical Computer," *Scientific American*, Feb. 1983, pp. 85-93.

[2] P. Ambs, "Optical Computing: A 60-Year Adventure," *Adv. In Optical Tech.*, V2010, pp. 1-15.

[3] H. Caulfield and S. Dolev., "Why future supercomputing requires optics," *Nature Photonics*, V4, May 2010, pp. 261-263.

[4] T. Chattopadhyay and J. Roy, "All-optical quaternary computing and information processing: a promising path," *Journal of Optics*, Mar. 2013.

[5] N. Chowdhury and R. Boutaba. "A survey of network virtualization," *Computer Networks* 54.5 (2010): 862-876.

[6] P. Green, "An All-Optical Computer Network: Lessons Learned," *IEEE Network*, Mar. 1992, pp. 56-60.

[7] S. Gringeri, E. Basch, and T. Xia, "Technical Considerations for Supporting Data Rates Beyond 100 Gb/s," *IEEE Communications*, Feb. 2012, pp. 521-530.

[8] P. Guilfoyle and R. Stone, "Digital Optical Computer II," *SPIE Optical Enhancements to Computing Technology*, V1563, 1991, pp. 214-222.

[9] H. Harai and M. Murata, "High-speed buffer management for 40 Gb/s-based photonic packet switches," *IEEE/ACM Trans. on Networking*, V14 N1, Feb. 2006, p.191-204.

[10] J. Hardy and J. Shamir, "Optics inspired logic architecture," *Optics Express*, V15 N1, Jan. 2007, pp. 150-165.

[11] A. Huang, "Parallel Algorithms for Optical Digital Computers," Proc. SPIE Int'l. Optical Computing Conf., Apr. 1983, p. 13-17.

[12] D. Jackson, "Photonic processors: a systems approach," *Applied Optics*, V33 N23, Aug. 1994, pp. 5451-5466.

[13] K. Jain and G. Pratt, G., "Optical transistor," *Applied Physics Letters*, V28, 1976, pp. 719-721.

[14] M. Jeon, Z. Pan, J. Cao, Y. Bansal, J. Taylor, Z. Wang, V. Akella, K. Okamoto, S. Kamei, J. Pan, and S. Yoo, "Demonstration of all-optical packet switching routers with optical label swapping and 2R regeneration for scalable optical label switching network application," *IEEE/OSA Journal of Lightwave Technology*, V21 N 11, 2003, pp. 2723.

[15] J. McGeehan, S. Kumar, D. Gurkan, S. Motaghian Nezam, J. Bannister, J. Touch, and A. Willner "All-Optical Decrementing of a Packet's Time-To-Live (TTL) Field and Subsequent Dropping of a Zero-TTL Packet," *IEEE/OSA Journal of Lightwave Technology*, Special Issue on Optical Networks, V21 N11, Dec. 2003, pp. 2746-2752.

[16] N. McKeown, "Software Defined Networking," Keynote presentation, IEEE Infocom 2009.

[17] D. Miller, "Are optical transistors the logical next step?" *Nature Photonics*, V4, Jan. 2010, pp. 3-4.

[18] D. Miller, "Correspondence - The role of optics in computing," *Nature Photonics*, V4, July 2010, pp. 406.

[19] K. Mistry, "Tri-Gate Transistors: Enabling Moore's Law at 22nm and Beyond," Presentation at Semicon West 2014, http://www.semiconwest.org/sites/semiconwest.org/files/docs/Kaizad%20Mistry_Intel.pdf

[20] Optical Turing Machine project web pages, http://www.isi.edu/otm

[21] Y. Paquot, F. Duport, A. Smerieri, J. Dambre, B. Schrauwen, M. Haelterman, and S. Masar, "Optoelectronic Reservoir Computing," *Scientific Reports*, V2 N287, Feb. 2012, pp. 1-6.

[22] A. Sawchuk and T. Strand, "Digital Optical Computing," *Proc. IEEE*, V72 N7, July 1984, pp. 758-779.

[23] R. Slavík, F. Parmigiani, J. Kakande, C. Lundström, M. Sjödin, P. Andrekson, R. Weerasuriya, S. Sygletos, A. Ellis, L. Grüner-Nielsen, D. Jakobsen, S. Herstrøm, R. Phelan, J. O'Gorman, A. Bogris, D. Syvridis, S. Dasgupta, P. Petropoulos and D. Richardson "All-optical phase and amplitude regenerator for next-generation telecommunications systems," *Nature Photonics*, V4 N10, 2010, pp. 690-695.

[24] J. Touch, J. Bannister, S. Suryaputra, and A. Willner, "A design for an Internet router with a digital optical data plane," Proc. Photonics West, 2014.

[25] R. Tucker, "Correspondence - The role of optics in computing," *Nature Photonics*, V4, July 2010, pp. 405.

[26] R. Van Meter, *Quantum Networking*, Wiley, 2014.

[27] J. Wang, J., J.-Y. Yang, X. Wu, O. Yilmaz, S. Nuccio, and A. Willner, "40-Gbaud/s (120-Gbit/s) Octal and 10-Gbaud/s (40-Gbit/s) Hexadecimal Simultaneous Addition and Subtraction Using 8PSK/16PSK and Highly Nonlinear Fiber," Proc. OFC, Mar. 2011.

[28] Q. Xu, D. Fattal, and R. Beausoleil, "Silicon microring resonators with 1.5-μm radius," *Optics Express*, Vol. 16, Issue 6, pp. 4309-4315 (2008).

[29] M. Ziyadi, A. Mohajerin-Ariaei, J.-Y. Yang, Y. Akasaka, M. Chitgarha, S. Khaleghi, A. Almaiman, A. Abouzaid, J. Touch, and M. Sekiya, "Experimental Demonstration of Optical Regeneration of DP-BPSK/QPSK Using Polarization-Diversity PSA," Proc. CLEO, 2014.

[30] L. Liu, *et al.*, "An ultra-small, low-power, all-optical flip-flop memory on a silicon chip," *Nature Photonics*, V4, Mar. 2010, pp. 182-187.