Performance Modeling Sparse MTTKRP Using Optical Static Random Access Memory on FPGA

Sasindu Wijeratne*, Akhilesh Jaiswal†, Ajey P. Jacob†, Bingyi Zhang*, Viktor Prasanna*
*Department of Electrical and Computer Engineering, University of Southern California, Los Angeles, CA, USA
†Information Sciences Institute (ISI), University of Southern California (USC), Marina Del Rey, CA, USA
Email: kangaram@usc.edu, {akjaiswal, ajey}@isi.edu, {bingyizh, prasanna}@usc.edu

***Abstract*—Electrical static random memory (E-SRAM) is the current standard for internal static memory in Field Programmable Gate Array (FPGA). Despite the dramatic improvement in E-SRAM technology over the past decade, the goal of ultra-fast, energy-efficient static random memory has yet to be achieved with E-SRAM technology. However, preliminary research into optical static random access memory (O-SRAM) has shown promising results in creating energy-efficient ultra-fast static memories.**

This paper investigates the advantage of O-SRAM over E-SRAM in access speed and energy performance while executing sparse Matricized Tensor Times Khatri-Rao Product (spMTTKRP). spMTTKRP is an essential component of tensor decomposition algorithms which is heavily used in data science applications. The evaluation results show O-SRAMs can achieve speeds of 1.1× - 2.9× while saving 2.8× - 8.1× energy compared to conventional E-SRAM technology.

***Index Terms*—Optical Static Random Access Memory, energy efficiency, spMTTKRP, Memory Systems, FPGA, Tensor Decomposition**

I. Introduction

Recent advances in collecting and analyzing large datasets have led to the information being naturally represented as higher-order tensors. Tensor Decomposition transforms input tensors to a reduced latent space which can then be leveraged to learn salient features of the underlying data distribution. Tensor Decomposition has been successfully employed in many fields, including machine learning, signal processing, and network analysis [1]–[3]. Canonical Polyadic Decomposition (CPD) [4] is the most popular method of decomposing a tensor to a low-rank tensor decomposition model. It has become the standard tool for unsupervised multiway data analysis. The Matricized Tensor Times Khatri-Rao product (MTTKRP) kernel [5] is known to be the computationally intensive kernel in CPD. Due to the sparse nature of real-world tensors, specialized hardware accelerators are becoming increasingly popular for improving the efficiency of sparse tensor computations. However, memory access time has become the bottleneck in sparse MTTKRP (spMTTKRP) operation due to irregular data access patterns.

The 6 transistor E-SRAM is currently the de-facto standard for on-chip memory storage. However, the memory access speed for E-SRAM is constrained by long electrical wires, constituting the bit-lines and wordlines in an SRAM array and associated parasitic resistances and capacitances. Optical memory systems have the capability to achieve orders of magnitude faster memory access speed using ultra-fast optical signals that do not suffer from fundamental signal transfer speed bottlenecks like their electrical counterparts. Various implementations of optical-SRAM (O-SRAM) have been explored in the past [6]–[13]. However, an O-SRAM technology amenable to existing foundry manufacturing exhibiting ultra-high speed and low-energy consumption has remained challenging. Recently, however, an O-SRAM technology built using foundry-friendly optical devices, with excellent speed and energy-efficiency has been reported in [14]. The O-SRAM reported in [14] consists of an optical bistable element formed by a feedback connection between photodiodes and microring resonators. The bistable optical element can store two levels in a differential manner (i.e. storing the bit and the complement of the bit). Due to its differential nature, similar to E-SRAM, the O-SRAM of [14] features differential read and write ensuring robust memory operations. Further, the photodiodes and the microring resonators are reverse biased ensuring small static current dissipation and hence energy-efficiency compared to previous works. Thus, the recent advances in O-SRAM solutions as in [14], has driven optical memory system closer to large-scale manufacturing while providing ultra-fast speed and excellent energy-efficiency. It is thereby important to quantify system-level benefits of such emerging optical memory solutions on representative data intensive computational kernels as in Tensor Decomposition.

Since real-world tensors are sparse, specialized hardware accelerators are attractive for improving the compute efficiency of sparse tensor computations [15]–[19]. As spMTTKRP is memory bound, improving the accelerator's internal sustained memory bandwidth and latency can significantly reduce the computation time. Our work mainly focuses on FPGA as it facilitates near memory computing with custom adaptive hardware due to its reconfigurability and large on-chip memory [20], [21]. It enables the development of a custom memory hierarchy and compute units.

In this work, we develop a performance model to analyze the acceleration and power efficiency we can achieve by replacing the internal E-SRAMs inside an FPGA with O-SRAMs. The contributions of our paper are as follows:

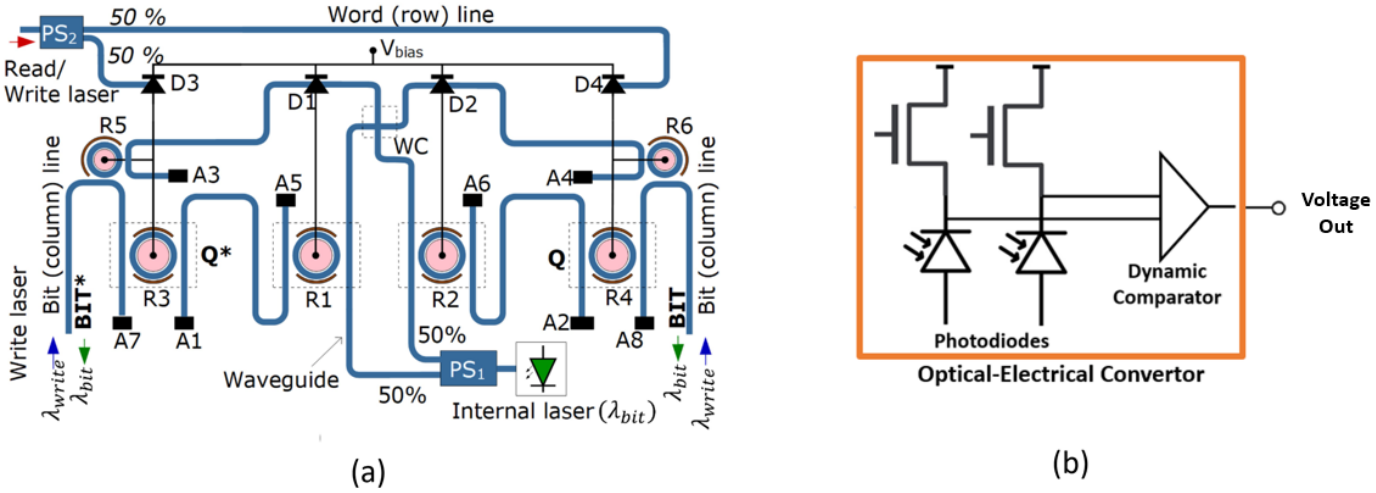

Fig. 1: (Left) A 2x2 Optical-SRAM array [14] showing cascaded optical bit-cells (Right) electro-optic sense amplifiers used to sense the optical data from memory array and convert it to electrical data

- We develop a performance model to analyze the O-SRAM memory design for a FPGA environment.
- We evaluate the impact of O-SRAM on a memory-bound algorithm by evaluating spMTTKRP using our proposed performance model.
- Our results show that O-SRAM can improve the total execution time of spMTTKRP by up to 2.9× compared with traditional E-SRAMs.
- Our system also shows 2.8× - 8.1× energy savings while performing spMTTKRP using O-SRAM compared with traditional E-SRAMs.

II. Optical Memory Technology

The O-SRAM shown in Fig. 1 is adopted from [14]. It consists of a bistable element formed by photodiodes D1 and D2 and microring resonators (MRRs) R1 and R2. The bistable element is coupled to photodiodes D3-D4 and MRRs R3-R6, which act as read/write access devices. To access a particular data the Wordline waveguide is activated by sending a light pulse through it and the data can be accessed through the waveguides BIT and BIT* in Fig. 1. Thus, the O-SRAM of Fig. 1 is functionally similar to E-SRAM in the following ways: 1) the data is stored in the bistable element as complementary optical signals, similar to E-SRAM that stores complementary electrical data in a bit-cell 2) the Wordline and BIT/BIT* waveguides are orthogonal to each other, allowing the O-SRAM bit-cell to be cascaded to create a large memory array.

Due to the optical nature of storage, the operating speed of O-SRAM can be orders of magnitude faster than its electrical counterparts. Further, as the array size increases, the speed of E-SRAM drastically reduces due to parasitic resistance and capacitances associated with metal wires. O-SRAM on the other hand can have very large area arrays without any significant degradation in speed as data is accessed through waveguides that carry optical signals as opposed to metal lines carrying electrical signals. O-SRAM are thus suitable for large-scale chips as in wafer-scale systems since they can transfer ultra-fast data across long distances using optical signaling. Wafer-scale systems are chips built on an entire 300mm wafer [22]. Wafer-scale system also helps to accommodate the high area associated with optical memories. An O-SRAM bit-cell is over three orders of magnitude larger in size compared to an E-SRAM bit-cell. This is because while the CMOS transistor has undergone unprecedented scaling over the past few decades, the sizes for state-of-the-art optical devices like photodiodes and MRRs are typically in the range of micro-meters [23], [23]. Thus, there exists a crucial non-trivial trade-off for O-SRAMs compared to E-SRAMs, wherein O-SRAMs can provide orders of magnitude faster speed and high energy efficiency while occupying significantly larger areas compared to E-SRAMs. In this work, we aim to quantify the system-level benefits of O-SRAMs keeping in view the performance-power and area trade-off. For this work, we assume that O-SRAM acts as the on-chip memory for a wafer-scale FPGA, while the processing engines are built using conventional CMOS technology. While proposals for a high-speed optical logic circuit can be found in the literature [24], [25], the scalability, noise resilience, and programmability of CMOS logic circuits are currently beyond the capability of optical logic circuits. Thus, our wafer-scale system is a heterogeneous system consisting of silicon photonics-based optical memories and CMOS-based processing engines. Note, due to the use of well-established silicon photonics devices for O-SRAM, the optical memory system can be seamlessly fabricated on the same wafer consisting of silicon CMOS transistors.

III. Performance Modeling Optical SRAM Memory

A. Modeling the operation of O-SRAM

Fig. 1 shows the high-level view of an O-SRAM block. An O-SRAM block consists of optical storage, an optical-to-electrical conversion unit, and an electrical-to-optical con-

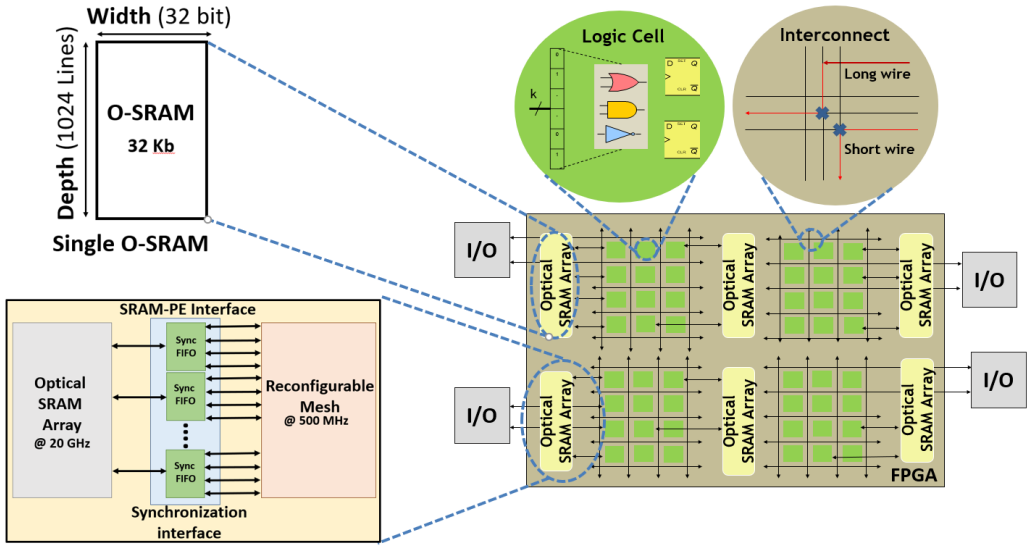

Fig. 2: Overall FPGA with Optical SRAMs integrated into the architecture

version unit. Note that the electrical-to-optical conversion unit is not explicitly shown in Fig. 1 since it can be easily implemented using an active micro-ring resonator [26]. The O-SRAM, in this case, operates at 20 GHz. Also, it can support multiple optical wavelengths (typically 5) through the use of wavelength division multiplexing, which enables concurrent access to the same O-SRAM block.

Fig. 2 illustrates the overall FPGA architecture with O-SRAM integrated. In the proposed work, O-SRAMs reside along with a configurable electrical mesh architecture. An O-SRAM uses a synchronization interface to connect with the configurable mesh due to the operation frequency difference between electrical compute components (i.e., electrical LUTs and DSPs) and optical memory components (O-SRAMs).

In our work, we consider a FPGA with electrical memory components (i.e., electrical Block RAMs [27] and Ultra RAMs [28]) replaced by the same amount of O-SRAM memory. As illustrated in Figure 2, A single O-SRAM can store 32 Kb of data. It contains 1024 data lines, where each data line has a width of 32 bits. Also, each O-SRAM consists of 200 parallel read-write ports with 32 bit-width as O-SRAMs support multiple optical wavelengths with high operating frequency.

For an O-SRAM that runs at f_{optical} frequency using λ number of wavelengths where each read-write port has a width of z bits, the number of bits (b_{process}) it can provide to the electrical compute elements which runs at $f_{\text{electrical}}$ frequency is:

$$b_{\text{process}} = \frac{\lambda \times f_{\text{optical}} \times z}{f_{\text{electrical}}} \quad (1)$$

Also, we focus on large tensor datasets where inputs initially reside in the FPGA external memory. FPGA external memory contains multiple DRAMs which use DDR4 technology.

B. Modeling the energy consumption of O-SRAM

The total energy consumption of an FPGA accelerator design (E_{FPGA}) on an O-SRAM-based FPGA is calculated by:

$$E_{\text{FPGA}} = P_{\text{compute}} \times t_{\text{runtime}} + E_{\text{DRAM-FPGA}} + (P_{\text{O-SRAM}} \times n_{\text{O-SRAM}}) \times t_{\text{runtime}} \quad (2)$$

Here, $P_{compute}$, $E_{\text{DRAM-FPGA}}$, $n_{\text{O-SRAM}}$, $P_{\text{O-SRAM}}$, and t_{runtime} refer to power consumption of compute resource of the FPGA, total energy consumption of DRAM-FPGA interface during external memory transactions, number of O-SRAMs used by the accelerator design, power consumption of a single O-SRAM, and run time of the accelerator, respectively.

The power consumption of a single O-SRAM block ($P_{\text{O-SRAM}}$) depends on the static power ($P^{\text{static}}_{\text{O-SRAM}}$) and switching power consumption ($P^{\text{switching}}_{\text{O-SRAM}}$) of the memory block. Static Power is primarily depends on the size of the SRAM ($S^{\text{total}}_{\text{O-SRAM}}$), electrical power per bit ($\hat{p}_{\text{optical storage}}$) and optical static power consumption per bit ($\hat{p}^{\text{static}}_{\text{optical}}$). $\hat{p}_{\text{optical storage}}$ and $\hat{p}^{\text{static}}_{\text{optical}}$ are a result of leakage power of optical and electrical components inside an O-SRAM. The switching power of O-SRAM ($P^{\text{switching}}_{\text{O-SRAM}}$) depicts the power consumed by O-SRAM during a read or write operation. It depends on the active number of bits of the O-SRAM in a given clock cycle ($S^{\text{active}}_{\text{O-SRAM}}$) following the power consumption of optical-electrical conversion per bit ($\hat{p}_{\text{optical-electrical conversion}}$) and per bit power consumption of optical storage units ($\hat{p}_{\text{optical storage}}$) shown in Fig. 1(a). Equation 3 summarizes the O-SRAM power calculations.

$$\begin{aligned} P_{\text{O-SRAM}} &= P^{\text{static}}_{\text{O-SRAM}} + P^{\text{switching}}_{\text{O-SRAM}} \\ P^{\text{static}}_{\text{O-SRAM}} &= S^{\text{total}}_{\text{O-SRAM}} \times (\hat{p}^{\text{static}}_{\text{optical}} + \hat{p}^{\text{static}}_{\text{electrical}}) \\ P^{\text{switching}}_{\text{O-SRAM}} &= S^{\text{active}}_{\text{O-SRAM}} \times (\hat{p}_{\text{optical-electrical conversion}} + \hat{p}_{\text{optical storage}}) \end{aligned} \quad (3)$$

Algorithm 1: SPMTTKRP OPERATION FOR MODE 0 OF A TENSOR WITH 3 MODES

1 Input: A sparse tensor $\mathcal{X} \in \mathbb{R}^{I_0 \times I_1 \times I_2}$, dense factor matrices $\mathbf{B} \in \mathbb{R}^{I_1 \times R}$, $\mathbf{C} \in \mathbb{R}^{I_2 \times R}$
2 Output: Updated dense factor matrix $\mathbf{A} \in \mathbb{R}^{I_0 \times R}$
3 **for** *each* i_0 *output factor matrix row in* $\mathbf{A}$ **do**
4 $\mathbf{A}(i_0, :) = 0$
5 **for** *each nonzero element in* $\mathcal{X}$ *at* (i_0, i_1, i_2) *with* i_0 *coordinates* **do**
6 Load($\mathcal{X}(i_0, i_1, i_2)$)
7 Load($\mathbf{B}(i_1, :)$)
8 Load($\mathbf{C}(i_2, :)$)
9 **for** $r = 1, \ldots, R$ **do**
10 $\mathbf{A}(i_0, r) += \mathcal{X}(i_0, i_1, i_2) \times \mathbf{B}(i_1, r) \times \mathbf{C}(i_2, r)$
11 Store($\mathbf{A}(i_0, :)$)
12 **return** $\mathbf{A}$

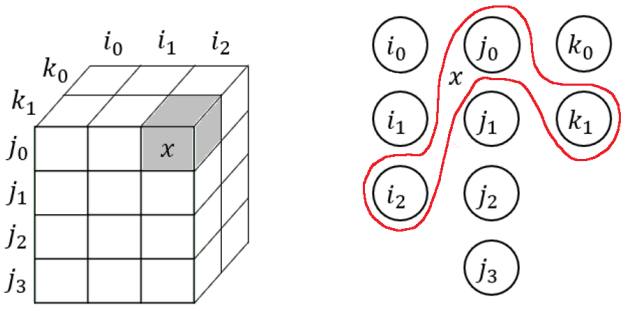

Fig. 3: A hypergraph example of a sparse tensor

IV. SPMTTKRP ACCELERATOR

A. spMTTKRP Computation

We use a hypergraph model to explain the spMTTKRP operation on a given input tensor. For illustrative purposes, we consider a 3 mode sparse tensor $\mathcal{X} \in \mathbb{R}^{I_0 \times I_1 \times I_2}$ where (i_0, i_1, i_2) denote the coordinates of tensor element x in $\mathcal{X}$. Here, I_0, I_1, and I_2 represent the size of each tensor mode. Note that the following approach can be applied to tensors with any number of modes.

For a given tensor $\mathcal{X}$, we can build a hypergraph $H = (V, E)$ with the vertex set V and the hyperedge set E as follows: vertices correspond to the tensor indices in all the modes and hyperedges represent its non-zero elements. For a 3D sparse tensor $\mathcal{X} \in \mathbb{R}^{I_0 \times I_1 \times I_2}$ with M non-zero elements, its hypergraph $H = (V, E)$ consists of $|V| = I_0 + I_1 + I_2$ vertices and $|E| = M$ hyperedges. A hyperedge $\mathcal{X}(i, j, k)$ connects the three vertices i, j, and k, which correspond to the indices of rows of the factor matrices. Fig. 3 shows an example of the hypergraph for a sparse tensor.

Our goal is to determine a mapping of $\mathcal{X}$ into memory for each mode so that the total time spent on (1) loading tensor data from external memory, (2) loading input factor matrix data from the external memory, (3) storing output factor matrix data to the external memory, and (4) element-wise computation for each non-zero element of the tensor is minimized.

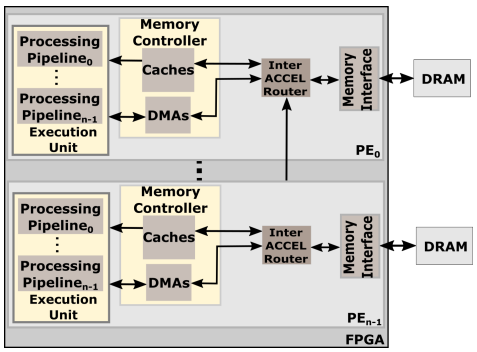

Fig. 4: Overall Architecture

In Algorithm 1, all hyperedges that share the same vertex of the output mode are accessed consecutively. The input vertices of the hyperedge are traversed to access rows of the remaining input factor matrices. It follows the element-wise multiplication and addition. Since the order of hyperedge depends on the output mode, the output factor matrix can be calculated without generating intermediate partial sums (Algorithm 1: line 10).

The total computations of the approach: For a general sparse tensor with $|T|$ non-zero elements, N modes, and factor matrices with rank R, since every hyperedge will be traversed once, and there are $N-1$ multiplication and one addition for computing spMTTKRP, the total computation per mode is $N \times |T| \times R$.

The total external memory (i.e., DRAMs) accesses: It requires $|T|$ load operations for all the hyperedges and the total factor matrix elements transferred per mode is $(N-1) \times |T| \times R$, which corresponds to accessing input factor matrices of vertices in the hypergraph model. Let I_{out} represent the length of the output mode. Then the total amount of data transferred is $|T| + (N-1) \times |T| \times R + I_{out} \times R$.

The proposed sparse spMTTKRP computation has 4 main actions: (1) load a non-zero tensor element, (2) load corresponding factor matrices, (3) perform spMTTKRP operation, and (4) store the final output.

We use a memory controller to decrease the total DRAM memory access time. It supports following access types:

1) **Cache transfers**: Memory controller contains O-SRAM based multiple caches that Support random memory accesses. Load/store individual requests in minimum latency. Access patterns with high spatial and temporal locality are transferred using cache lines.
2) **DMA stream transfers**: Memory controller contain SRAM based multiple Direct Memory Access (DMAs) that Support streaming accesses. Load/store operations on all requested data with minimum latency from memory.
3) **DMA element-wise transfers**: DMAs can also be used to access data with no spatial and temporal locality.

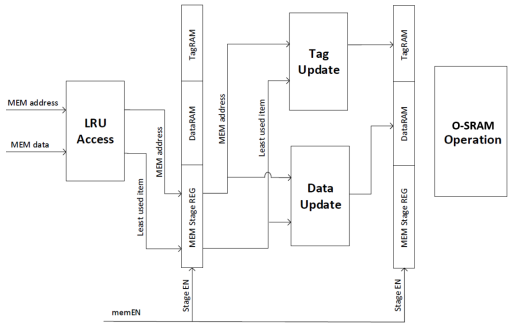

Fig. 5: Memory (MEM) pipeline of the cache

B. Proposed FPGA Accelerator Design

Fig. 4 shows the overall architecture of our FPGA accelerator design. We keep the number of Processing Elements (PEs) equal to the number of DRAMs attached to the FPGA. PE consists of a memory controller, execution unit, and external memory interface. The execution unit inside the PE consists of parallel pipelines. It is a simple pipeline structure where computation demonstrated in Algorithm 1 is executed. Here, all the partial sums of spMTTKRP are stored inside an O-SRAM-based partial sum buffer.

The cache subsystem includes multiple O-SRAM-based caches. Each cache is shared with multiple input factor matrices. Each cache focuses on satisfying a single memory request with minimum latency. The cache uses two separate pipelines namely the PE pipeline and memory pipeline to support the high data rate of O-SRAMs due to high frequency. Fig. 5 and Fig. 6 depict the memory pipeline (MEM pipeline) and PE pipeline, respectively. They share the same Tag RAM, Data RAM, and LRU which are implemented using O-SRAMs. The PE pipeline is made into four stages, starting with a tag access step. Based on the address of the PE requests, tags are pulled out from the Tag RAM, denoted as *Tag_x*, and then compared to the incoming tag in the next stage. After the Tag comparator, cache hit information is generated and sent into the third stage. In this stage, the HIT information will be used as an evaluation criterion on whether the LRU update is needed or not. For read requests of *m* (associativity) number of data, notated as *Data_x*, the data is pulled out from the Data RAM at the same time. Otherwise, for a written request with a hit, the updated data will be written into the corresponding entry of the Data RAM.

Each PE also contains Direct Memory Access units (DMAs) to access large tensors from the external memory. These DMAs have large buffers implemented using O-SRAMs.

V. Evaluation

A. Experiments Setup

As described in section III, we consider a wafer-scale FPGA platform developed using 12 nm technology with O-SRAM as its internal memory. It contains a total of 54 MB of O-SRAM memory replacing the electrical SRAMs (E-SRAM) in a typical data center FPGA (eg., Xilinx Alveo U250) [20]. It also contains 6433K LUTs, and 8474K Flip-flops, with 31K

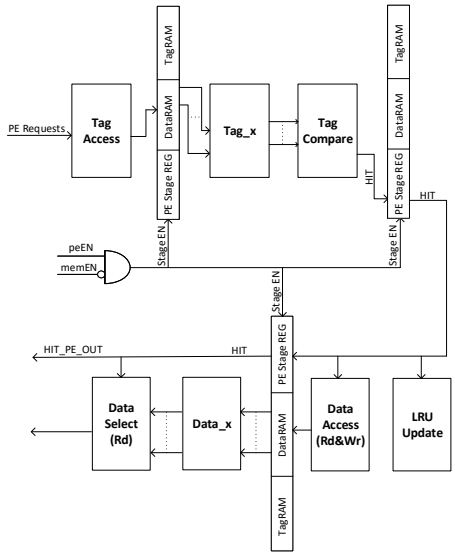

Fig. 6: Process element (PE) pipeline of the cache

DSPs. The compute units (configurable mesh) operate at 500 MHz, simulating a typical FPGA setup. Meanwhile, the O-SRAMs operate at 20 GHz as described in Section II.

The power and performance estimates for O-SRAM were obtained from electro-optic simulations using *Lumerical Interconnect* [29]. Estimates for E-SRAM were based on SRAM design in Globalfoundries 12nm node, compute and PE array primitives were synthesized to obtain power-performance-area estimates at 12nm Globalfoundries PDK. Finally, SPICE simulations were used to obtain the energy estimate for the optical-to-electrical interface.

TABLE I: Configurations of the accelerator

Module	Configuration
PE	Number of PEs: 4
Parallel Pipelines	No. of pipelines: 80
	Partial Matrix Buffer size: 1024 elements
Cache sub system	Number of caches: 3
	Associativity: 4
	Number of cachelines: 4096
	cachelines width: 64 B
DMAs	No. DMA buffers: 6
	DMA buffer size: 64 KB

1) Dataset: We use the sparse tensors, derived from real-world applications, that appear in Table II. All of these tensors are provided by The Formidable Repository of Open Sparse Tensors and Tools (FROSTT) dataset [30]. The selected dataset exhibits a variety of different sparse tensors in terms of dimensions, size, and density.

2) Implementation: Table I shows the configuration parameters and the configuration we used in our experiments. Also,

TABLE II: Characteristics of the targeted sparse tensors

Tensor	Dimensions	#NNZs	Density
NELL-1	$2.9M \times 2.1M \times 25.5M$	$143.6M$	9.1×10^{-13}
NELL-2	$12.1K \times 9.2K \times 28.8K$	$76.9M$	2.4×10^{-05}
PATENTS	$46 \times 239.2K \times 239.2K$	$3.6B$	1.4×10^{-03}
LBNL	$1.6K \times 4.2K \times 1.6K \times 4.2K \times 868.1K$	$1.7M$	4.2×10^{-14}
DELICIOUS	$532.9K \times 17.3M \times 2.5M \times 1.4K$	$140.1M$	4.3×10^{-15}
AMAZON	$4.8M \times 1.8M \times 1.8M$	$1.7B$	1.1×10^{-10}
REDDIT	$8.2M \times 177K \times 8.1M$	$4.7B$	4.0×10^{-10}

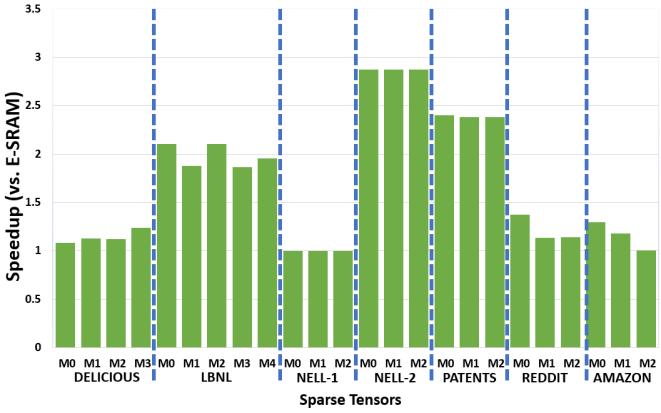

Fig. 7: Speedup achieved by replacing E-SRAM with O-SRAMs

the tensor rank (R) is set to 16 [31].

3) Baseline platform: We consider the same FPGA with E-SRAM as the internal memory (i.e., electrical Ultra RAM [28] and Block RAM [27]) for the baseline.

B. Overall Execution Time Performance

Fig. 7 shows the speedup the accelerator design achieved while using O-SRAMs. The baseline environment used for comparison is described in Section V-A3. In the horizontal axis, Mi ($i \in \{0, 1, 2, 3, 4\}$) refer to the each mode of the input tensor.

According to experiments, due to high data locality in memory accesses, NELL-2 and PATENT shows significant speedup while using O-SRAM. The support for concurrent accesses and the high clock frequency of O-SRAM contribute to this substantial speedup. Also, NELL-1 and DELICIOUS do not show significant speed up with O-SRAM as external FPGA memory access dominates the total execution time in these datasets.

TABLE III: Energy consumption of the memory devices while FPGA operating at 500 MHz

Per bit Energy Consumption (pJ/cycle)			
Static		Switching	
Electrical Technology	Optical Technology	Electrical Technology	Optical Technology
1.175×10^{-6}	4.17×10^{-6}	4.68	1.04

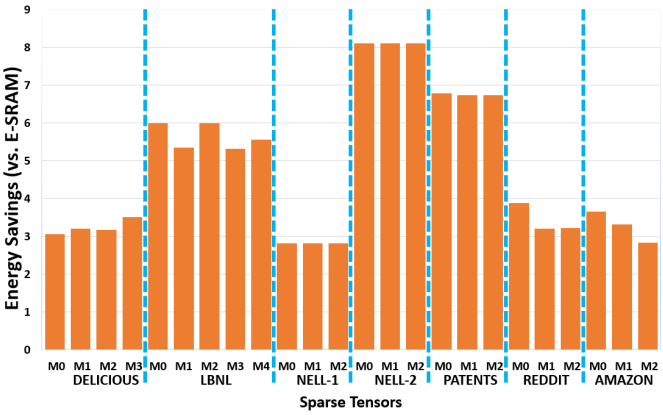

Fig. 8: Energy Savings by using O-SRAM technology

TABLE IV: The area with different SRAM technologies

	On-chip Memory	PEs	Total
E-SRAM system	$43.2\ mm^2$	$202.2\ mm^2$	$247.2\ mm^2$
O-SRAM system	$103.7 \times 10^4\ mm^2$	$202.2\ mm^2$	$103.7 \times 10^4\ mm^2$

C. Overall Energy Performance

The energy consumption of two memory technologies is shown in Table III. Compared with the electrical memory device, the optical memory device has lower switching energy driven by ultra-high operating frequency.

To compare the energy efficiency, the two FPGAs execute the same accelerator design with the datasets mentioned in Section V-A1. The energy comparison of two FPGAs (i.e., O-SRAM-based FPGA and E-SRAM based FPGA) on different datasets is shown in Fig. 8. According to the experiments results the O-SRAM FPGA is 2.8× - 8.1× more energy efficient than the E-SRAM FPGA.

D. Area Comparison

Although optical memory has shown low energy consumption with faster execution time, it occupies a significant area compared with the electrical, limiting the density of the optical memory and making large area wafer-scale systems a necessity for practical use of optical memory. Table IV shows a breakdown of the area comparison between O-SRAM-based FPGA and E-SRAM-based FPGA.

VI. Conclusion

In this paper, we perform comprehensive performance and energy consumption modeling of an optical SRAM-based FPGA environment for a sparse MTTKRP accelerator. The evaluation results show that using state-of-the-art optical memory technology, a FPGA can achieve an average of 1.68× speedup in execution time while saving an average of 5.3× more energy compared to an electrical SRAM-based system. Our future work includes reducing the area consumption of optical SRAM through multi-bit storage and optical device optimization.

REFERENCES

[1] M. Mondelli and A. Montanari, "On the connection between learning two-layer neural networks and tensor decomposition," in *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, 2019, pp. 1051–1060.

[2] Z. Cheng, B. Li, Y. Fan, and Y. Bao, "A novel rank selection scheme in tensor ring decomposition based on reinforcement learning for deep neural networks," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 3292–3296.

[3] F. Wen, H. C. So, and H. Wymeersch, "Tensor decomposition-based beamspace esprit algorithm for multidimensional harmonic retrieval," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 4572–4576.

[4] A. H. Phan, P. Tichavský, and A. Cichocki, "On fast computation of gradients for CANDECOMP/PARAFAC algorithms," *CoRR*, vol. abs/1204.1586, 2012. [Online]. Available: http://arxiv.org/abs/1204.1586

[5] I. Nisa, J. Li, A. Sukumaran-Rajam, R. Vuduc, and P. Sadayappan, "Load-balanced sparse mttkrp on gpus," in *2019 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, 2019, pp. 123–133.

[6] N. Pleros, D. Apostolopoulos, D. Petrantonakis, C. Stamatiadis, and H. Avramopoulos, "Optical static RAM cell," *IEEE Photonics Technology Letters*, vol. 21, no. 2, pp. 73–75, 2008.

[7] A. Tsakyridis, T. Alexoudi, A. Miliou, N. Pleros, and C. Vagionas, "10 Gb/s optical random access memory (RAM) cell," *Optics Letters*, vol. 44, no. 7, pp. 1821–1824, 2019.

[8] B. Dong, H. Cai, Y. Gu, Z. Yang, Y. Jin, Y. Hao, D. Kwong, and A. Liu, "Nano-optomechanical static random access memory (SRAM)," in *2015 28th IEEE International Conference on Micro Electro Mechanical Systems (MEMS)*. IEEE, 2015, pp. 49–52.

[9] B. Li, M. I. Memon, G. Mezosi, Z. Wang, M. Sorel, and S. Yu, "Optical static random access memory cell using an integrated semiconductor ring laser," in *2009 International Conference on Photonics in Switching*. IEEE, 2009, pp. 1–2.

[10] T. Alexoudi, D. Fitsios, A. Bazin, P. Monnier, R. Raj, A. Miliou, G. T. Kanellos, N. Pleros, and F. Raineri, "III–V-on-Si photonic crystal nanocavity laser technology for optical static random access memories," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 22, no. 6, pp. 295–304, 2016.

[11] S. Pitris, C. Vagionas, T. Tekin, R. Broeke, G. Kanellos, and N. Pleros, "WDM-enabled optical RAM at 5 Gb/s using a monolithic inp flip-flop chip," *IEEE Photonics Journal*, vol. 8, no. 2, pp. 1–7, 2016.

[12] Y. Liu, R. McDougall, M. Hill, G. Maxwell, S. Zhang, R. Harmon, F. Huijskens, L. Rivers, H. Dorren, and A. Poustie, "Packaged and hybrid integrated all-optical flip-flop memory," *Electronics Letters*, vol. 42, no. 24, pp. 1399–1400, 2006.

[13] A. Trita, G. Mezosi, M. Zanola, M. Sorel, P. Ghelfi, A. Bogoni, and G. Giuliani, "Monolithic all-optical set-reset flip-flop operating at 10 Gb/s," *IEEE Photonics Technology Letters*, vol. 25, no. 24, pp. 2408–2411, 2013.

[14] R. Kudalippalliyalil, S. Chandran, A. P. Jacob, and A. Jaiswal, "Towards scalable, energy-efficient and ultra-fast optical sram," 2021. [Online]. Available: https://arxiv.org/abs/2111.13682

[15] J. Li, J. Sun, and R. Vuduc, "Hicoo: Hierarchical storage of sparse tensors," in *SC18: International Conference for High Performance Computing, Networking, Storage and Analysis*, 2018, pp. 238–252.

[16] A. E. Helal, J. Laukemann, F. Checconi, J. J. Tithi, T. Ranadive, F. Petrini, and J. Choi, "Alto: Adaptive linearized storage of sparse tensors," in *Proceedings of the ACM International Conference on Supercomputing*, ser. ICS '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 404–416. [Online]. Available: https://doi.org/10.1145/3447818.3461703

[17] S. Smith, N. Ravindran, N. D. Sidiropoulos, and G. Karypis, "Splatt: Efficient and parallel sparse tensor-matrix multiplication," in *2015 IEEE International Parallel and Distributed Processing Symposium*, 2015, pp. 61–70.

[18] J. Li, B. Uçar, U. V. Çatalyürek, J. Sun, K. Barker, and R. Vuduc, "Efficient and effective sparse tensor reordering," in *Proceedings of the ACM International Conference on Supercomputing*, ser. ICS '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 227–237. [Online]. Available: https://doi.org/10.1145/3330345.3330366

[19] I. Nisa, J. Li, A. Sukumaran-Rajam, P. S. Rawat, S. Krishnamoorthy, and P. Sadayappan, "An efficient mixed-mode representation of sparse tensors," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, ser. SC '19. New York, NY, USA: Association for Computing Machinery, 2019. [Online]. Available: https://doi.org/10.1145/3295500.3356216

[20] Xilinx, "Alveo u250 data center accelerator card," https://www.xilinx.com/products/boards-and-kits/alveo/u250.html, 2019.

[21] R. Zhang, S. Wijeratne, Y. Yang, S. R. Kuppannagari, and V. K. Prasanna, "A high throughput parallel hash table on fpga using xor-based memory," in *2020 IEEE High Performance Extreme Computing Conference (HPEC)*, 2020, pp. 1–7.

[22] K. Rocki, D. Van Essendelft, I. Sharapov, R. Schreiber, M. Morrison, V. Kibardin, A. Portnoy, J. F. Dietiker, M. Syamlal, and M. James, "Fast stencil-code computation on a wafer-scale processor," 2020. [Online]. Available: https://arxiv.org/abs/2010.03660

[23] M. de Cea, D. Van Orden, J. Fini, M. Wade, and R. J. Ram, "High-speed, zero-biased silicon-germanium photodetector," *APL Photonics*, vol. 6, no. 4, p. 041302, 2021.

[24] Q. Xu and M. Lipson, "All-optical logic based on silicon micro-ring resonators," *Optics express*, vol. 15, no. 3, pp. 924–929, 2007.

[25] P. Singh, D. K. Tripathi, S. Jaiswal, and H. Dixit, "All-optical logic gates: designs, classification, and comparison," *Advances in Optical Technologies*, vol. 2014, 2014.

[26] P. Dong, S. Liao, D. Feng, H. Liang, D. Zheng, R. Shafiiha, C.-C. Kung, W. Qian, G. Li, X. Zheng *et al.*, "Low V, ultralow-energy, compact, high-speed silicon electro-optic modulator," *Optics Express*, vol. 17, no. 25, pp. 22 484–22 490, 2009.

[27] Xilinx, "Using the block ram (bram) sheet," https://docs.xilinx.com/r/en-US/ug440-xilinx-power-estimator/Using-the-Block-RAM-BRAM-Sheet, 2022.

[28] ——, "Ultrascale architecture memory resources," https://docs.xilinx.com/v/u/en-US/ug573-ultrascale-memory-resources, 2022.

[29] Lumerical, "Aphotonic integrated circuit simulator," https://www.lumerical.com/products/interconnect/, 2022.

[30] S. Smith, J. W. Choi, J. Li, R. Vuduc, J. Park, X. Liu, and G. Karypis. (2017) FROSTT: The formidable repository of open sparse tensors and tools. [Online]. Available: http://frostt.io/

[31] J. Li, J. Sun, and R. Vuduc, "Hicoo: Hierarchical storage of sparse tensors," in *SC18: International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE, 2018, pp. 238–252.