Flexible Bit-Truncation Memory for Low-Power Quality-Adaptive Video and Deep Learning Storage

William Oswald, Md. Sajjad Hossain, Kyle Mooney, Mario Renteria-Pinon, Md. Bipul Hossain,
Mohamed Shaban, Jinhui Wang, and Na Gong
Department of Electrical and Computer Engineering, University of South Alabama, Mobile, USA 36688.

2024 IEEE 15th International Green and Sustainable Computing Conference (IGSC) | 979-8-3315-0786-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/IGSC64514.2024.00025

Abstract—**Bit-truncation has demonstrated great potential to enable run-time quality-power adaptive data storage and thus enhance the power efficiency of data-intensive applications such as videos and deep learning. However, existing bit-truncation memories are custom designed for a specific working condition of the target application. In this paper, we present a novel bit-truncation memory with full truncation flexibility, and it can truncate any number of bits for optimal tradeoff between quality requirements of applications and power savings. Our experiments show that the proposed memory can support three different video applications (including luminance-aware, content-aware, and region-of-interest-aware) with enhanced power efficiency (up to 50.03% power savings) as compared to state-of-the art solutions. Also, the proposed memory achieves up to 66.56% and 63.29% power savings for baseline and lightweight deep learning models respectively, with a low implementation cost (2.42%).**

Index Terms—**memory, truncation, quality-adaptive, low-power, videos, deep learning.**

I. INTRODUCTION

Today, the design challenge for an efficient computing system is growing due to the diminishing benefits from semiconductor technology scaling and the increased demands of data-intensive applications such as video processing and deep neural networks (DNN) [1]. Fortunately, those applications usually have non-deterministic specifications and have multiple acceptable quality levels in different conditions, which provides exciting opportunities for quality-adaptive low-power design [2]. Quality-adaptive low-power design aims to dynamically trade off quality and power consumption to optimize the power efficiency, while meeting the quality requirement of applications.

To enable quality-adaptive computing systems, bit truncation is one of the most widely-applied techniques. Compared to other techniques used for quality-adaptation, e.g., dynamic voltage scaling, bit truncation has two major advantages: (i) it can enable more energy savings [3] and (ii) it can achieve better output quality [4]. For example, studies [4], [5], [6] applied bit truncation to luminance-aware video memory design and revealed that more least-significant-bits (LSBs) of pixel data can be truncated if the video device is operating under high lighting conditions. Specifically, 3-bit and 4-bit LSBs can be truncated in overcast and sunlight, respectively. Also, considering the impact of video content on viewer's experience, Edstrom et al. [7] truncated the number of LSBs between zero-bit to 4-bit according to the micro-block characteristics of videos. Very recently, Haidous et al. [8] presented a Region-of-Interest (ROI)-aware video memory, which can truncate 3 LSBs for non-ROI regions of videos to further optimize video output quality while reducing the power consumption. As for the above techniques, in each video application, i.e., luminance-aware [5], content-aware [7], and ROI-aware [8], one custom bit truncation memory was developed to enable adaptation. In other words, existing bit truncation memory cannot provide flexibility to support different video conditions. As such, state-of-the art bit truncation memory designs cannot be used for other applications such as deep neural network (DNN).

In this paper, we propose a novel bit truncation memory, coined as TrunMEM, which enables flexible number of bits truncation with optimal truncation values to support different video systems as well as DNN inference. Compared to existing work, TrunMEM is unique in the following ways: First, TrunMEM can enable run-time power-quality adaptation to meet different requirements of the target application. For example, with full flexibility, it can be used to support all three video systems (i.e., luminance-aware [5], content-aware [7], and ROI-aware video storage [8]). Second, by integrating pruning in the model training process and truncation in the inference process, TrunMEM achieves precision-scalable DNN to meet the performance and efficiency requirement of different AI tasks. Third, TrunMEM automatically sets the truncated bits to the optimal values for specific video and DNN use-cases, thereby optimizing the quality to realize maximum power savings. To the best of the authors' knowledge, the proposed TrunMEM has made the first attempt to design a truly flexible bit-truncation memory to enable quality adaptation and power efficiency optimization for different applications.

II. ADAPTIVE BIT TRUNCATION

A. Bit Truncation

In order to enable low-power data storage, bit truncation needs to adapt the number of truncated bits as well as setting optimal truncation values. In terms of video data which are represented by 8-bit integer pixel values, it has been concluded from previous work [7] that setting the truncated LSBs to its mean value, i.e., 10...0 in binary will minimize the expected mean square error (MSE). Here, to apply bit truncation to DNN weight storage, we also study the truncated values for floating point numbers. Specifically, IEEE 754 single precision floating point representation was used in our analysis, which has been used widely in deep learning systems [9]. A 32-bit number consists of three fields including a 1-bit sign, an 8-bit

979-8-3315-0786-2/24/$31.00 ©2024 IEEE
DOI 10.1109/IGSC64514.2024.00025

exponent, and the least 23-bit mantissa [10]. In our analysis, we adopted one of the most popular DNN - AlexNet [11] with CIFAR-10 dataset [12]. Fig. 1 shows the performance of the network with two different truncation strategies. From Fig. 1, the following two observations can be made: (i) At least 16 LSBs mantissa can be truncated without accuracy loss; with more LSBs truncated, it will cause accuracy degradation with additional power savings. Thus, it has potential to enable precision-scalable DNN with different precision-power trade-offs and (ii) filling truncated bits with mean value (i.e., 10...0 in binary) retains a higher classification accuracy as compared to all zeroing the data, thereby tolerating additional truncation (Fig. 1). In other words, if a truncated bit is the most-significant bit (MSB) of all truncated bits, its truncated value should be *1*; otherwise, the truncated value should be *0*. Also, it is worthy to emphasize that, the number of truncated bits may be different as the neural network changes. A detailed analysis on different neural networks will be presented in Section 3. Also, a flexible bit-truncation memory design to adapt the number of truncated bits and to set the optimal truncation values is critical to optimize the power efficiency.

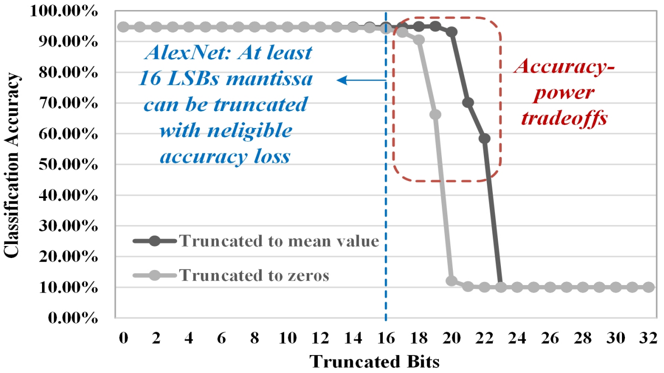


Fig. 1: Classification accuracy of AlexNet with different truncation bits and values.

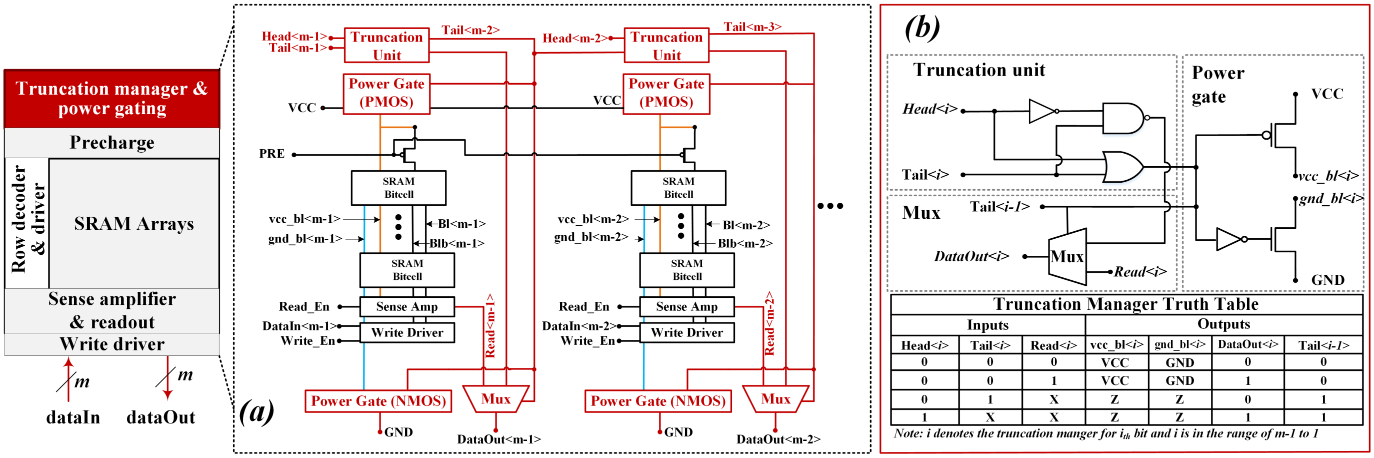
B. Proposed TrunMEM

Fig. 2 illustrates the architecture of the proposed TrunMEM. As shown in Fig. 2 (a), the SRAM array consists of N words by m bits, with each bitcell designed as a 6T SRAM. In addition to the conventional memory components such as pre-charge unit, write driver, and sense amplifier, each column of bitcells have a *truncation manager* circuitry to control truncation adaptation. Also, to reduce the power consumption of truncated bits, the proposed truncation manager uses CMOS power gating circuitry to connect to virtual power rails, which can either provide a connection to the true power rails, or truncate the signal by putting the virtual rails in high impedance, as shown in Fig. 2 (b). Accordingly, each column of SRAM bitcells have their own dedicated pair of virtual rails, such that a single power gate can disconnect an entire column of SRAM bitcells at once, thus minimizing power consumption. Peripheral circuitry operating on a specific column of bits will also be power-gated along with the SRAM cells for additional power savings. However, any SRAM bitcell disconnected from the power and ground rails quickly loses the value stored inside, and cannot be read from. As such, the truncation manager circuit also populates read values when bit-lines get disconnected, which can either be a *1* or *0*.

Specifically, as shown in Fig. 2 (b), each truncation manager consists of a truncation unit, power gating transistors, and an output multiplexer (Mux). The truncation manager has three operating states: normal operation without truncation, MSB of truncated bits (MSB truncated), and non-MSB of truncated bits (lesser truncated). The signals to control these states are the two inputs: $Head$ and $Tail$. $Head$ is used to detect whether it is MSB truncated or not and $Tail$ indicates whether it is lesser truncated or not. The operation process is detailed as follows. If neither signal for a specific bit (e.g., *ith* bit) is active, i.e., $Head < i >= Tail < i >= 0$, the *ith* bit will be in the normal state without truncation and the $DataOut < i >$ will be the normal memory readout value (i.e., $Read < i >$. During the normal state, the virtual rails ($vcc\_bl < i >$ and $gnd\_bl < i >$) remain connected to the supply voltage (VCC) and ground (GND), respectively. When $Head$ of the *ith* bit is *1*, i.e., $Head < i >= 1$, which indicates MSB truncated, $DataOut < i >$ will be *1* and $vcc\_bl < i >$ and $gnd\_bl < i >$ will be placed into high impedance for power savings. Similarly, at the lesser truncated state, when $Tail < i >= 1$, virtual power rails enter high impedance and a $dataOut = 0$ value will be generated.

As also shown Fig. 2, in either truncation states, the output of the *ith* bit, i.e., $Tail < i >$ will be applied to next bit as input, i.e., $Tail < i - 1 >$, and therefore the truncation managers of different bits work in series. As a result, $Head$ is the only external control signal required to be managed from external circuitry. This series connection enables control such that if $Head < i >= 1$, the truncation manager signals to all lesser significant truncation managers to truncate, via the $Tail$ signals. Accordingly, the truncated memory returns an optimal truncation value with the most significant truncated bit as *1* and other bits as zeros (i.e., $10...0_2$). The $Tail < m - 1 >$ is directly connected to GND, and the $Tail < 0 >$ signal is left floating in this design, as they serve no purpose.

It can also be observed from Fig. 2 (b) that the proposed truncation manager circuit is implemented with several simple logic gates, which may not induce a large area overhead. However, the sizing of power gating transistors may be point of concern, depending on the number of transistors they need to power. As a result, the number of truncation managers is a trade-off between flexibility and implementation cost. Full bit-adaptation is achieved in the design of Fig. 2, which adds m truncation managers to the memory to support any value of truncation from 0 to $m - 1$ bits. Such a flexible bit truncation provides the best flexibility to support different applications. However, to reduce the area overhead, if the target applications are known, the number of the truncation manager circuit may be reduced accordingly.



Truncation Manager Truth Table

Inputs			Outputs			
Head<i>	Tail<i>	Read<i>	vcc_bl<i>	gnd_bl<i>	DataOut<i>	Tail<i-1>
0	0	0	VCC	GND	0	0
0	0	1	VCC	GND	1	0
0	1	X	Z	Z	0	1
1	X	X	Z	Z	1	1

Note: i denotes the truncation manger for i_{th} bit and i is in the range of m-1 to 1

Fig. 2: Proposed TrunMEM: (a) Memory structure and (b) Truncation manager circuitry and truth table.

III. Experimental results

A. Experimental Methodology

Hardware-Level Implementation and Verification: To evaluate the effectiveness of the proposed memory, an SRAM with 1024 words by 32 bits was implemented using a 130 nm CMOS technology [13], [14], [15], [16]. Based on the designed memory, a comprehensive suite of simulations were performed. Specifically, functionality and performance parameters including timing diagram, power consumption, and layout area overhead were evaluated and discussed.

Application-Level Evaluation: In addition to hardware-level implementation and verification, application-level quality was also evaluated for both videos and deep learning applications. In terms of video storage, we used the proposed memory to support three different video applications: luminance-aware [6], content-aware [7], and ROI-aware [8] video storage. For each video application, the video quality and power savings were evaluated. Specifically, we used two widely-used metrics including Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) for video quality assessment.

To fully evaluate the effectiveness of the proposed memory for deep learning, two different models including one basic VGG-16 model [17], and one filter-pruned lightweight VGG-16 model are evaluated with TrunMEM. Specifically, we fine-tuned the pre-trained VGG-16 model on CIFAR-10 dataset [12], where 50,000 and 10,000 samples are used for training and testing the model, respectively. The training of the model has been executed using stochastic gradient decent as optimizer and categorical cross entropy as loss function. With batch size of 32, initial learning rate of 0.001, and maximum 150 epochs were used to train the model. These settings have been identified using the grid search algorithm where the learning rate, the batch size, and the number of epochs have been varied from 0.000001 to 0.01, 4 to 256, and 10 to 400 respectively. For both models, the test accuracy with different truncated bits for weights is evaluated and discussed.

B. Hardware-Level Simulation Results

Timing Diagram: Fig. 3 shows the timing diagram of TrunMEM. $Wlen$, $PreB$, $ReadEn$ and $Write\_en$ are used to activate the word-line for reading and writing access, pre-charge the bit-lines for reading, and enable reading and writing, respectively. All the *dataout* signals are the outputs of the SRAM from sense amplifiers, with $dataout31$ and $dataout0$ being the most and least significant bit, respectively. Specifically, to test the functionality of TrunMEM, firstly, the word-line is activated and the 32-bit data ”01010...01” is written in the first clock cycle. Then, the data is read out in the following four cycles with various levels of bit truncation. The pattern for all truncated bits includes the most significant bit becoming “1” and all other bits “0”. As shown in Fig. 3, the optimal truncated values are generated for 2-bit, 3-bit, and 16-bit truncations (2 LSBs, 3 LSBs, and 16 LSBs of *dataout*) are “10”, “100”, and “1000_0000_0000_0000”, respectively.

Power Efficiency: The power consumption with different truncated bits is shown in Fig. 4. As expected, power savings and number of truncated bits demonstrate a strong linear relationship. For example, 25%, 50%, and 75% power savings can be enabled with 8, 16, and 24 truncated bits, respectively. On average, a power savings of 3.11% will be enabled with each additional bit truncated.

Implementation cost: As discussed in Section 2.2, the silicon area overhead of TrunMEM is mainly caused by the added truncation managers. The layout design of one truncation manager is shown in Fig. 5. To minimize the area overhead, Power Gate transistors are implemented using a particular finger-based design approach. Each power transistor has a width of 16 µm and it is implemented with eight fingers. Scaling the transistor width to the total number of fingers used in order to maximize switching speed. As shown in Fig. 5, a truncation manager occupies an area of 373.37 μm^2. To achieve full bit truncation with best flexibility, 32 truncation managers are stacked in rows on top of a SRAM with 32 bits in each word, which occupies an area of 11,947.73 μm^2. The total area of a 32x1024 TrunMEM memory is measured

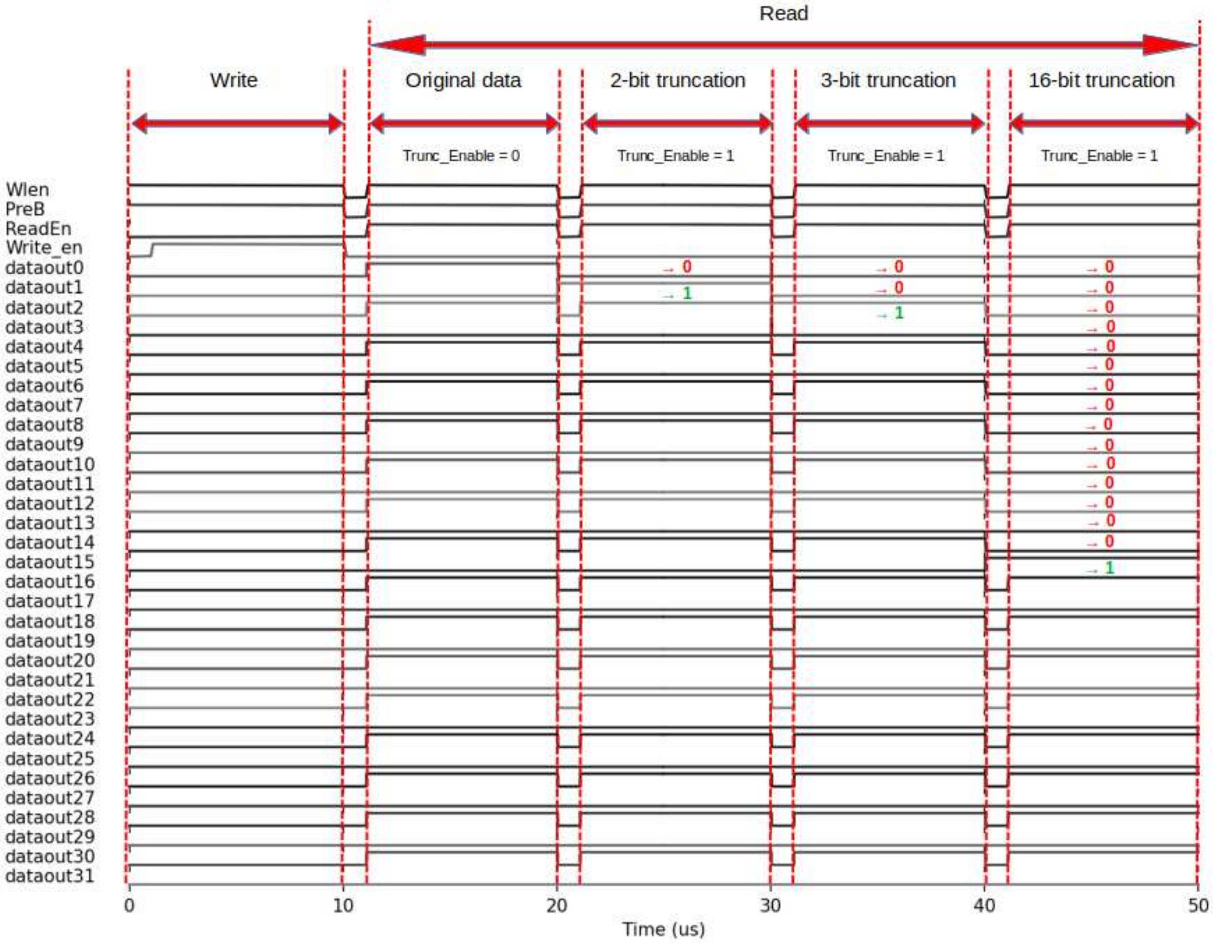


Fig. 3: Timing diagram of TrunMEM, with writing the binary value ’01010...01’ to a word followed by four reading operations (starting with normal reading, 2-bit truncation, 3-bit truncation, and 16-bit truncation).

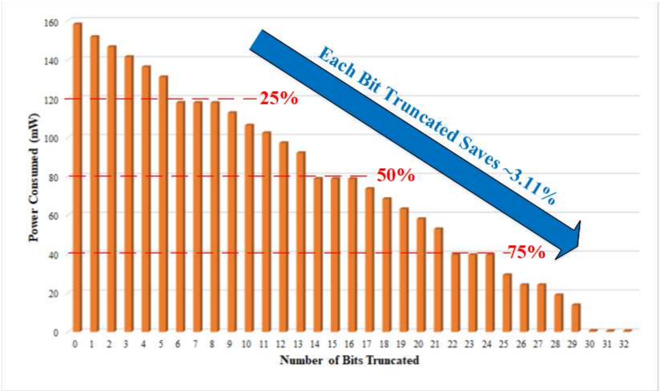


Fig. 4: Power consumption of TrunMEM at each level of bit truncation on a 32x32 memory array.

as 493511.64 μm^2, making the addition of the truncation managers consume only 2.42% silicon area as compared to the traditional memory. Since each bit-line uses one truncation manager, the area overhead ratio of TrunMEM can be further reduced as the number of words in a memory increases.

Video Storage: We evaluated the video quality with TrunMEM following the same process as prior work [6], [7], [8]. Fig. 6 shows the visual output quality with PSNR and SSIM values as well as enabled power savings using TrunMEM. It can be seen that the proposed TrunMEM can support all three video applications with additional power savings. Specifically, TrunMEM achieves 5.45% and 9.82% power savings as compared to content-aware [7] and ROI-aware [8] video memory designs, respectively. As compared to luminance-aware video memory [6], TrunMEM enables 13.79% power savings in overcast and 17.43% power savings in sunlight, respectively. Thus, the proposed TrunMEM can support all three different video applications with enhanced power efficiency.

DNN: Another DNN model - VGG-16 has been used to evaluate the effectiveness of TrunMEM. The original VGG-16 model is considered a baseline model. In addition, we investigated how the filter pruning of the model during the training process interacts with the bit truncation of TrunMEM in the inference process. We used a proposed modification of the Squeeze and Excitation channel attention module [19], connected the module between each two consecutive convolutions layers of the baseline model, and finally retrained

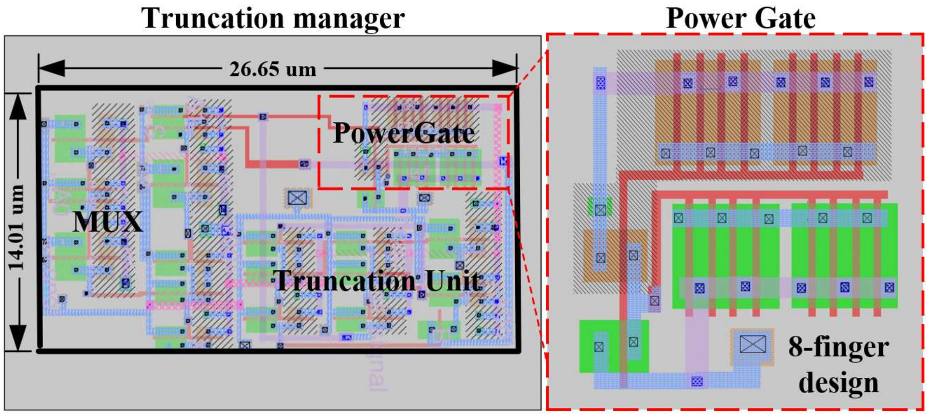
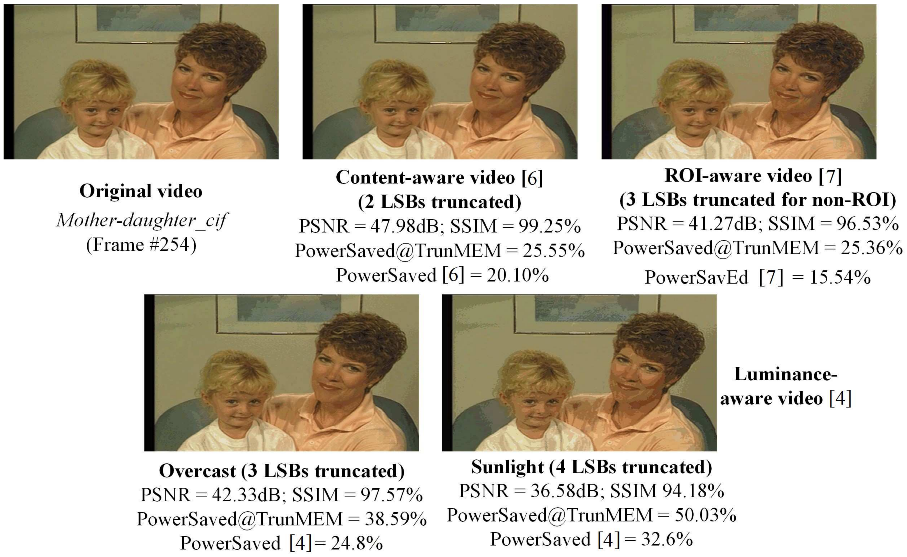


Fig. 5: Finger-based layout design in truncation managers.



Original video
Mother-daughter_cif
(Frame #254)

Content-aware video [6]
(2 LSBs truncated)
PSNR = 47.98dB; SSIM = 99.25%
PowerSaved@TrunMEM = 25.55%
PowerSaved [6] = 20.10%

ROI-aware video [7]
(3 LSBs truncated for non-ROI)
PSNR = 41.27dB; SSIM = 96.53%
PowerSaved@TrunMEM = 25.36%
PowerSavEd [7] = 15.54%

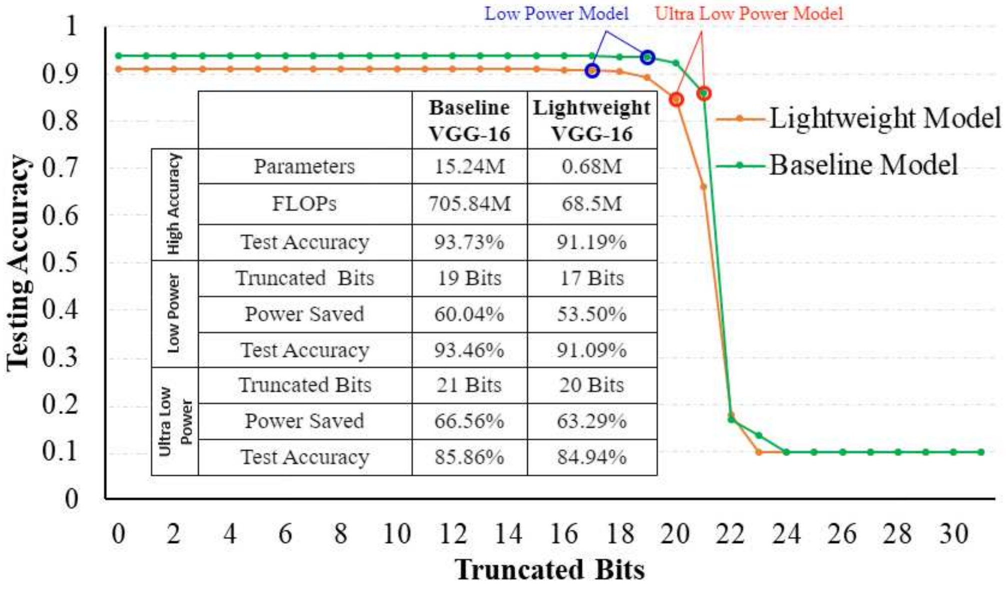
Luminance-aware video [4]

Overcast (3 LSBs truncated)
PSNR = 42.33dB; SSIM = 97.57%
PowerSaved@TrunMEM = 38.59%
PowerSaved [4]= 24.8%

Sunlight (4 LSBs truncated)
PSNR = 36.58dB; SSIM 94.18%
PowerSaved@TrunMEM = 50.03%
PowerSaved [4] = 32.6%

Fig. 6: Visual output quality with TrunMEM. Source videos from [18].

the model for 35 epochs. In addition, we calculated the significance of the feature maps of each layer of the model via measuring the scales vector generated at the output of the channel attention module [19]. We then pruned different percentages of the least significant filters within each layer of the model. After pruning the VGG-16 model, the model was retrained for 150 epochs by using an initial learning rate of 0.001. The learning rate was further reduced to half of its value after each 35-epoch to improve the convergence and the accuracy of the model. The testing accuracy of the model was then reported when different pruning percentages were considered. As shown in Fig. 7, using the channel attention method, the generated lightweight model achieved an almost 95.5% and 90.2% reduction in the model parameters and Floating Point Operations (FLOPs), respectively at a slightly lower testing accuracy as compared to the baseline model. Based on the baseline and the lightweight models, a bit truncation was applied to the weight storage for power savings. As shown in Fig. 7, TrunMEM enabled three quality-power trade-off levels for both models: (i) High Accuracy: without bit truncation, 93.73% and 91.19% test accuracy can be achieved by the baseline and lightweight models, respectively; (ii) Low Power: with 19-bit truncation and 17-bit truncation, the baseline and the lightweight models can achieve 60.04% and 53.50% power savings with negligible accuracy loss (less than 0.3%), respectively; and (iii) Ultra-Low Power: with a more aggressive bit truncation, as high as 66.56% and 63.29% power savings can be achieved by the two models and their test accuracy will further drop to 85.86% and 84.94%, respectively, which may be suitable for those relatively simple classification tasks on resource-constrained edge devices [20]. In particular, by applying TrunMEM to the lightweight model, we can see that combining pruning in the training process and truncation in the inference process has a great potential to enable power-quality adaptation and optimization. Furthermore, it can be concluded that TrunMEM is able to enable precision-scalable DNNs to meet the requirements of a variety of AI tasks.



		Baseline VGG-16	Lightweight VGG-16
High Accuracy	Parameters	15.24M	0.68M
	FLOPs	705.84M	68.5M
	Test Accuracy	93.73%	91.19%
Low Power	Truncated Bits	19 Bits	17 Bits
	Power Saved	60.04%	53.50%
	Test Accuracy	93.46%	91.09%
Ultra Low Power	Truncated Bits	21 Bits	20 Bits
	Power Saved	66.56%	63.29%
	Test Accuracy	85.86%	84.94%

Fig. 7: Performance of Deep Learning with TrunMEM.

IV. Conclusion

This paper has presented a novel quality-adaptive bit truncation memory design - TrunMEM to support different video and deep learning applications. The proposed memory enables up to 50.03% and 66.56% power savings in videos and DNN inference, respectively. With its full flexibility, the developed TrunMEM can also be applied to other data-intensive applications. It also demonstrates strong promise for general applications into application specific hardware platforms such as GPUs.

V. Acknowledgement

The work was supported by the National Science Foundation under grants OIA-2218046, CNS-2211215, ECCS-2420994, CCF-2247343, EES-2427766, and ITE-2346905.

References

[1] K. Roy and A. Raghunathan, “Approximate computing: An energy-efficient computing technique for error resilient applications,” in *2015 IEEE Computer society annual symposium on VLSI*. IEEE, 2015, pp. 473–475.

[2] M. Alioto, V. De, and A. Marongiu, “Energy-quality scalable integrated circuits and systems: Continuing energy scaling in the twilight of moore’s law,” *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 8, no. 4, pp. 653–678, 2018.

[3] F. Frustaci, D. Blaauw, D. Sylvester, and M. Alioto, “Approximate srams with dynamic energy-quality management,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no. 6, pp. 2128–2141, 2016.

[4] D. Chen, J. Edstrom, Y. Gong, P. Gao, L. Yang, M. E. McCourt, J. Wang, and N. Gong, “Viewer-aware intelligent efficient mobile video embedded memory,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 4, pp. 684–696, 2018.

[5] D. Chen, X. Wang, J. Wang, and N. Gong, “Vcas: Viewing context aware power-efficient mobile video embedded memory,” in *2015 28th IEEE International System-on-Chip Conference (SOCC)*. IEEE, 2015, pp. 333–338.

[6] J. Edstrom, D. Chen, J. Wang, H. Gu, E. A. Vazquez, M. E. McCourt, and N. Gong, “Luminance-adaptive smart video storage system,” in *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2016, pp. 734–737.

[7] J. Edstrom, Y. Gong, A. A. Haidous, B. Humphrey, M. E. Mccourt, Y. Xu, J. Wang, and N. Gong, “Content-adaptive memory for viewer-aware energy-quality scalable mobile video systems,” *IEEE Access*, vol. 7, pp. 47 479–47 493, 2019.

[8] A. Haidous, W. Oswald, H. Das, and N. Gong, “Content-adaptable roi-aware video storage for power-quality scalable mobile streaming,” *IEEE Access*, vol. 10, pp. 26 830–26 848, 2022.

[9] J. Edstrom, Y. Gong, D. Chen, J. Wang, and N. Gong, “Data-driven intelligent efficient synaptic storage for deep learning,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 64, no. 12, pp. 1412–1416, 2017.

[10] “Ieee standard for floating-point arithmetic,” *IEEE Std 754-2019 (Revision of IEEE 754-2008)*, pp. 1–84, 2019.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in neural information processing systems*, vol. 25, 2012.

[12] A. Krizhevsky, V. Nair, and G. Hinton, “Cifar-10 (canadian institute for advanced research),” accessed on 01.10.2024. [Online]. Available: http://www.cs.toronto.edu/ kriz/cifar.html

[13] “Skywater sky130 pdk.” [Online]. Available: https://skywater-pdk.readthedocs.io/en/main/

[14] “Openlane documentation,” accessed: 2023-9-6. [Online]. Available: https://openlane.readthedocs.io/en/latest/index.html

[15] “Magic vlsi layout tool,” accessed: 2023-10-12. [Online]. Available: http://opencircuitdesign.com/magic/

[16] “Xschem: Schematic capture and netlisting eda tool,” https://xschem.sourceforge.io/stefan/index.html, accessed: 2023-10-26.

[17] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

[18] “Xiph.org video test media [derf’s collection],” https://media.xiph.org/video/derf/, accessed: 2023-10-26.

[19] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[20] P. Panda, A. Sengupta, and K. Roy, “Conditional deep learning for energy-efficient and enhanced pattern recognition,” in *2016 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2016, pp. 475–480.