A FeRAM based Volatile/Non-volatile Dual-mode Buffer Memory for Deep Neural Network Training

Yandong Luo, Yuan-Chun Luo and Shimeng Yu
School of Electrical and Computer Engineering, Georgia Institute of Technology
Email: yandongluo@gatech.edu, shimeng.yu@ece.gatech.edu

***Abstract*— Deep neural network (DNN) training produces a large amount of intermediate data. As off-chip DRAM access is both energy and time consuming, sufficient on-chip buffer is preferred to achieve high energy efficiency for DNN accelerator designs. However, the low integration density and high leakage current of SRAM lead to large area cost and high standby power. The frequent refresh of embedded DRAM (eDRAM) degrades the energy efficiency due to its short refresh interval (40~100µs). In this paper, a dual-mode buffer memory that can operate in both volatile eDRAM mode and non-volatile ferroelectric RAM (FeRAM) mode is proposed, which is based on the CMOS compatible $HfZrO_2$ material. The functionality of the proposed dual-mode memory design is verified using SPICE simulation with the multi-domain Preisach model. A data lifetime-aware memory mode configuration protocol is proposed to optimize the buffer access energy. The architectural benchmark for DNN training shows 33.8%, 17.1% and 109.4% higher energy efficiency than baseline designs with eDRAM, FeRAM and SRAM with the same buffer area, respectively. The chip standby power is reduced by 26.8×~47.5× and 1.5×~10.6× compared with the SRAM and eDRAM baselines. The chip area overhead of the dual-mode buffer design is 5.7%.**

Keywords—Deep learning accelerator, embedded DRAM, ferroelectric RAM, refresh optimization

I. Introduction and Motivations

Deep neural network (DNN) based machine learning algorithms are the most popular and dominant approaches for artificial intelligence. However, DNN training produces a large amount of intermediate data (e.g. activations, errors, gradients), which could be in the range of GB. In state-of-the-art DNN accelerator designs, on-chip buffer are usually based on SRAM or embedded DRAM technologies. For example, in Google's tensor processing unit (TPU), 28MB on-chip SRAM buffer is used [1]. In the DaDianNao architecture [2], the on-chip buffer is implemented with 36MB eDRAM.

However, both SRAM and eDRAM have drawbacks that can degrade the performance of DNN accelerators. For SRAM, although its read energy is low, the large cell size and high leakage power lead to high area cost and standby power. It is undesired for edge devices where the area budget is limited and standby is frequent. Therefore, for DNN accelerator designs using SRAM buffer, the on-chip buffer capacity is usually limited and off-chip DRAM access is frequent, which is energy consuming. For eDRAM, it is featured of similarly low read energy and 3~4× higher memory density than SRAM [3]. However, due to its short retention time (40µs~100µs [4]), a large portion of energy is consumed for the periodic refresh.

Therefore, a memory technology with low read/write energy, high memory density and low standby power is preferred as on-chip buffer. Ferroelectric random access memory (FeRAM) has low leakage power due to its non-volatility. It can achieve high memory density by using the deep trench or stack capacitor structure as used in eDRAM, as state-of-the-art ferroelectric material, i.e., hafnium-zirconium oxide (HZO) is compatible with the advanced silicon CMOS fabrication process [5]. However, its read and write energy are higher than SRAM or eDRAM due to the read-destructive nature and high write voltage (1.8V~2.5V [6] [7]).

To reduce the read energy, one solution is to operate the FeRAM in the volatile charge domain like eDRAM when frequent read and write access are needed. When the read and write access are not frequent, the FeRAM operates in the non-volatile polarization domain that eliminates the refresh operation. The memory that can operate in the volatile and non-volatile mode is termed as dual-mode memory in this paper. The concept of dual-mode memory based on HZO FeRAM is first proposed by NaMLab [5]. However, the effectiveness of the concept has not been verified with either experiment or simulation. The performance using such dual-mode memory for system-level architectural design has not been investigated. Besides, most compute-in-memory (CIM) based accelerator designs so far focuses on the memory technologies for weight elements rather than buffers for intermediate data.

In this paper, a HZO FeRAM-based dual-mode memory is proposed as on-chip buffer for CIM-based DNN accelerator designs. The contributions are summarized as follows.

- A dual-mode buffer design is proposed for DNN accelerators using HZO FeRAM. The bit-cell configurablity between volatile and non-volatile modes is verified by SPICE simulation using the multi-domain Preisach model.
- A data lifetime-aware memory mode configuration protocol is proposed to decide the memory operation mode for the optimal buffer access energy.
- The system-level performance is benchmarked for DNN training. The hardware overhead for the dual-mode memory is estimated considering the modifications to commercial eDRAM architecture.

II. Background and Related Works

A. DNN Training Basics

Generally, the DNN training is divided into four phases: feed forward (FF), error calculation (EC), gradient calculation (GC) and weight update (WU). During FF, the training input (e.g. images) are fed into the DNN layer by layer. The activations of each layer and the output scores are obtained. During EC, the errors are calculated from the output layer and propagate back to the input layer. During GC, the weight gradient for each layer is calculated using the errors and the activations stored. The gradients are accumulated over a batch of images and update to the weights during WU. More details on DNN training can be found in [8].

B. HZO-based FeRAM and Its Dual-mode Operation

In ferroelectric materials, the polarization (P) state can be tuned under external voltage (V), as illustrated in the P-V loop in Fig. 1 (a). The polarization state remains after the voltage is removed and therefore the data storage is non-volatile. At

present, HZO-based ferroelectric material is widely used for the ferroelectric field effect transistor (FeFET) and the FeRAM. Compared with perovskite-based ferroelectric materials, HZO shows lower write voltage, better scalability towards nanoscale thin film [5]. The primary advantage is that HZO is compatible with CMOS logic process, where hafnium oxide is used for the high-k/metal gate, and the zirconium oxide is used as the dielectrics for DRAM. The deposition of HZO material is usually by atomic layer deposition (ALD) below 400°C, which is suitable for back-end-of-line (BEOL) capacitor integration. Industrial-grade FeRAM test chip using HZO planar capacitor at BEOL has been prototyped at 130nm technology node by Sony [7]. We envision a stack capacitor structure will be employed towards scaling to 28nm in the future. Since ALD deposition shows good step coverage, HZO could be used as stack capacitor with cylinder structure [9].

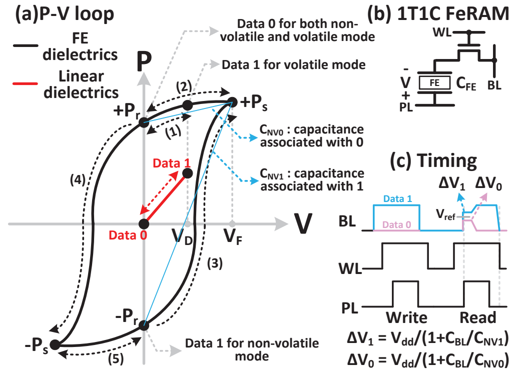

Fig. 1. (a) The P-V loop of FeRAM (b) the schematic of a 1T1C FeRAM cell (c) the timing diagram for FeRAM operation.

The FeRAM uses a 1-transistor 1-capacitor (1T1C) structure with three control signals: word line (WL), bit line (BL) and the plate line (PL), as shown in Fig. 1 (b). The data "1" and "0" are stored at the points with -Pr and +Pr in Fig. 1(a), respectively. The timing diagram for the 1T1C FeRAM operation is shown in Fig. 1(c). To write data "1", the BL voltage is raised to V_{dd} so that a negative voltage bias is applied to the FeRAM before PL voltage rises. Therefore, the polarization state is switched to $-P_s$ following the trajectory (4) or (5) in Fig. 1(a), which depends on the initial state. When the PL voltage becomes V_{dd}, the polarization state is switched back to $-P_r$ that represents data "1". To write data "0", the BL is grounded. When the PL voltage is raised to V_{dd}, the polarization state will be switched to $+P_s$ following the trajectory (2) or (3) in Fig. 1(a), which depends on the initial state. The polarization state will return to $+P_r$ that represents data "0" when the PL voltage returns to 0.

To read the data, a pulse of V_{dd} is applied to PL while BL is first pre-charged to zero voltage and then remains floating. The polarization state will be switched to $+P_s$ following trajectory (2) and (3) for data "0" and "1", respectively. However, as trajectory (3) corresponds to a larger capacitance value than (2) ($C_{NV1} > C_{NV0}$), the BL voltage increase for data "1" (ΔV_1) is higher than data "0" (ΔV_0), as shown in the two formulas in the bottom of Fig. 1(c) [10]. The voltage difference is sensed by using a reference voltage in the middle of ΔV_1 and ΔV_0, which can be generated by a reference cell with data "0" but larger cell size [10]. This sensing scheme has been verified with a 64kb FeRAM test chip [7].

FeRAM can be also used in the charge domain as volatile-memory, where the point $+P_r$ is used to store data "0", and an intermediate polarization state between $+P_r$ and $+P_s$ is used to store the data "1", as shown in the trajectory (1) in Fig. 1(a). The operation point for data "1" depends on the V_{dd} for the volatile operation mode. The timing diagram for volatile mode follows the timing for conventional eDRAM operation [11]. It is noted that the capacitance corresponds to the volatile data "1" and "0" in this dual-mode memory is nonlinear in the trajectory (1) of P-V loop, while it is a constant for eDRAM using linear dielectrics (the red trajectory in Fig. 1(a)).

C. eDRAM Refresh Scheme and Its Optptimization

The conventional eDRAM with linear dielectrics also uses the 1T1C bit cell design. Deep trench capacitor on silicon (by IBM [12]) or stack capacitor at BEOL (by Intel [11]) are usually used for eDRAM. However, since the transistor of eDRAM are fabricated with logic process, its retention time is only about 40µs~100µs due to higher leakage [4], [12]. This value is much lower than the 64ms refresh interval of commodity DRAM, where the process is optimized for low leakage. A concurrent refresh scheme is implemented for eDRAM due to the short retention time [11], [12]. In this scheme, the banks or bank groups that are not being accessed can be refreshed during a read or write operation at other banks.

Due to the short refresh interval, a large portion of the eDRAM system energy is consumed for refresh. It is reported that the refresh energy takes about 70% of the total energy for a 16MB eDRAM L3 cache with CPU benchmark traces [13]. In DNN accelerator designs, the refresh energy can be 5.5% ~ 60% of the total energy [2] [4]. To reduce the refresh energy, it is suggested to identify and refresh only the cache lines that are likely to be used in the future [13]. RANA increases the refresh interval by retraining the DNN model considering the bit error rate induced by longer refresh interval [4]. Our work focuses on reducing the refresh energy mainly from the bit-cell operation innovation and data lifetime-aware architectural protocol, which are orthogonal to the prior works.

III. Dual-mode Bit-Cell Operation

The functionality of our dual-mode memory is verified by SPICE simulation with a 28nm foundry PDK and the Verilog-A model that captures Preisach multi-domain switching dynamics for FeRAM [14]. The parameters for HZO under 1.8V write voltage is used for simulation [6], as listed in TABLE I. For the volatile mode, V_{dd}=1.0V is assumed. The FeRAM is assumed to be 1T1C and the capacitor is a stack cylinder capacitor with aspect ratio 20.

TABLE I. The FeRAM Parameters Used for Simulation [6]

Symbol	Meaning	Value	Unit
Ps	Saturation polarization	31.8	µC/cm^2
Pr	Remnant polarization	18.8	µC/cm^2
t_{FE}	Thickness of FE dielectrics	10	nm
E_c	Coercive field	0.88	MV/cm
V_{dd}	V_{dd} for non-volatile/volatile mode	1.8/1.0	V

The timing diagram for the volatile mode operation is shown in Fig. 2, which is similar to the timing for conventional eDRAM [11]. A cylinder capacitor with 80nm diameter (D) and 20fF BL capacitance load are assumed. For write, the BL is raised to V_{dd} or grounded to write data "1" and "0", respectively. During read, both the voltages at BL and BL_bar are first pre-charged to V_{dd}/2. The voltage of BL will either rise or drop when the access transistor is turned on by the WL. The sense amplifier will sense the voltage difference between BL and BL_bar and drive the BL voltage to either V_{dd} or 0, which restores the charge state at the storage node. The PL is

grounded for the volatile mode operation. From Fig. 2 (a), the read and write operation can be completed within 2ns.

The timing diagram for the non-volatile mode operation is shown in Fig. 2(b). 1.8V voltage is applied to the BL and PL to write the data. For write, the BL voltages are raised to V_{dd} or 0 for data "1" and "0", respectively. For read, the reference voltage in the middle of ΔV_1 and ΔV_2 are generated with a reference cell that stores "0" but has a larger capacitor size. The sense amplifier will sense the difference of the BL and reference voltage and then drive the BL to either V_{dd} or 0. The polarization state of the FeRAM can be automatically restored after the PL pulse. From Fig. 2(b), the read and write latency in non-volatile mode can be 20ns, which is consistent with the sub-20ns access latency from FeRAM chip measurement [7].

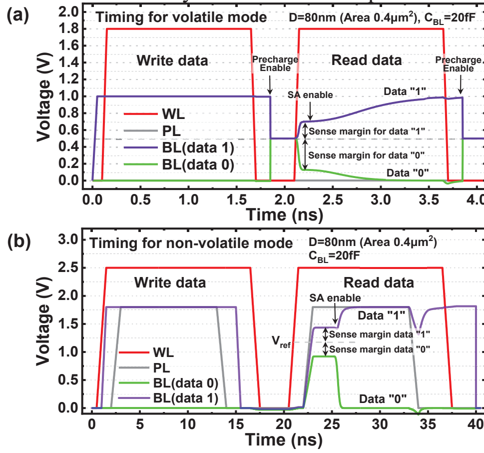

Fig. 2. The timing diagram from SPICE simulation for (a) volatile mode operation and (b) non-volatile mode operation of the same HZO based 1T1C dual-mode memory.

The read margin sweep with different capacitor diameters and BL load capacitances is shown in Fig. 3. For volatile mode, the read margin is defined as the difference between the BL voltage and the reference voltage $V_{dd}/2$. It is observed that the read margins for both data "1" and "0" decrease with larger BL load capacitance. Besides, with the same capacitor diameter and BL load capacitance, reading data "0" has larger read margin than reading data "1" because the capacitance associated with data "0" is larger than that associated with data "1", which is different from the symmetrical read margin in conventional eDRAM. To maintain > 0.2V read margin, capacitor diameter > 80nm and BL load capacitance smaller than 20fF is needed.

For the non-volatile mode, the read margin is half of the BL voltage difference when read "1" and "0", respectively. For a small cell size with D=40nm, the sense margin is decreased with larger BL load capacitance. However, for larger cell size (D=80 or 100nm), the sense margin is low with a small BL load capacitance. When reading non-volatile data "1" and "0", the BL voltage is raised toward the V_{dd}. Therefore, when the cell capacitance is large, the BL voltage is high for both "1" and "0", which reduces the sense margin. It indicates that the cell size and the array capacitance need to be co-designed to achieve good sense margin.

Based on the above results, the buffer memory sub-array size is assumed to be 256 (row)×512 (column). The cell size is about $38F^2$ with 80nm capacitor diameter. The BL wiring capacitance is about 11.18 fF in the sub-array with cell height 6.8F and the BL wiring capacitance 0.2fF/μm. The drain capacitance along the BL is about 5.89fF with 23aF drain capacitance per transistor (W=100nm). The total BL load capacitance is 18.8fF assuming 10% capacitance from parasitic (e.g. vias). From the simulation results, the sense margin for the sub-array design is 0.2V and 0.25V for the volatile and non-volatile mode, respectively, which is sufficient for a typical sense amplifier.

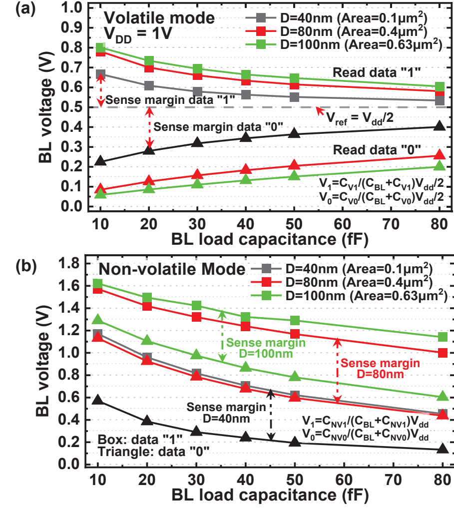

Fig. 3. The sense margin for (a) volatile mode and (b) non-volatile mode of the HZO-based 1T1C with different capacitor diameters and BL load capacitances.

IV. Data Lifetime-aware Configuration Scheme

To determine the operation mode of the memory, a data lifetime-aware configuration scheme is proposed. The data lifetime is defined as the time duration that the data is stored in the buffer before the next access. It can be determined by the layer index i, the processing time of the pipeline stage $T_{pipeline}$ and the number of pipeline stages N. The pipeline design in PipeLayer [15] is used here, where each layer of the DNN is regarded as one stage, as shown in Fig. 4 with a 5-layer example network (N=10 for both FF and EC). The red circles are the data that needs to be stored in buffer. Four types of data are considered: activation, error, gradient and parameters for the normalization layer. In this paper, the group normalization is considered so that the activation does not need to be stored for a batch.

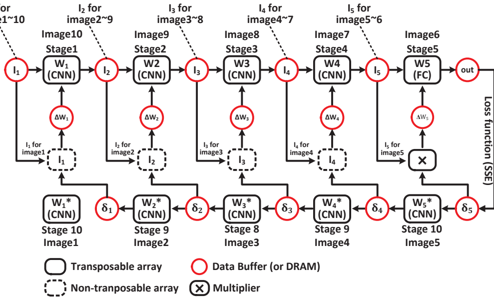

Fig. 4. An illustration of the pipeline for DNN training.

In Fig. 4, the activation for layer i is stored at stage i during the FF phase and it is fetched at stage N+1-i during the GC phase. Therefore, the lifetime for the activation of layer i is $(N+1-2i)\times T_{pipeline}$. For example, when image 2 is at stage 9 for GC for layer 2, its activation of layer 2 was stored at stage 2, which is $7T_{pipeline}$ before. For error, it is fetched at the next pipeline stage after being stored. Therefore, its lifetime is just $T_{pipline}$. The weight gradients of a layer is accumulated over a batch of images. Therefore, the lifetime of gradient is also

$T_{pipeline}$ as the gradient for one image is produced every $T_{pipeline}$ for each layer. The parameters of the normalization layer need to be maintained even after training. Therefore, it is always stored using the non-volatile mode to eliminate the requirement of refresh.

With the lifetime of each data type, the buffer access energy for different mode is determined based on the access energy per bit (E_{NV_read}, E_{NV_write}, E_{V_read}, E_{V_write}) and the number of read and write accesses needed (N_{read}, N_{write}). For activation, error and gradient, N_{write}=1 as they are written into the buffer one time. For activations and errors, with the novel weight mapping strategy [16], it is read by 3 times during FF and EC for a 3×3 CNN kernel, respectively. Besides, with the conventional mapping in GC, the activations are read by 1 time and errors are read by 9 times as the input. Therefore, N_{read}=4 for activation and N_{read}=12 for error. For gradient, N_{read}=1. For volatile mode, its access energy also includes the refresh energy during the data lifetime.

The buffer operation mode is determined for each layer for each data type. If data lifetime is shorter than the refresh interval, the buffer will be configured to volatile mode without refresh, which is at the granularity of a row of sub-arrays. Otherwise, the energy consumption for volatile and non-volatile mode are compared and the mode with less energy consumption is selected. The non-volatile mode is configured at the granularity of bank as it requires higher supply voltage V_{dd}.

V. Hardware Architecture of Dual-mode Memory

Fig. 5 shows the architecture of a dual-mode memory buffer, where the differences between conventional eDRAM design are highlighted in blue. In eDRAM design [12], the concurrent refresh scheme is implemented with a global refresh controller, which issues the refresh bank select (RBSEL) signal for the banks that are not currently being accessed. When a bank receives the RBSEL signal, the row address to be refreshed is fetched from a local refresh address counter and the corresponding row will be refreshed.

To support the dual-mode memory, a refresh status table implemented by 4kb SRAM is added to store the 1-bit buffer mode bit for each bank, which is marked as (1) in Fig. 5. If the bank is configured to non-volatile mode, the bank mode bit is set to 0 so that the RBSEL signal will always be 0 after the AND gate. When the data is to be stored from non-volatile mode to volatile mode or vice versa, it is first read from its current bank and stored into the write data buffer. Then, it is written to the memory bank with the target operation mode. A mux (marked as (2)) is added to select the input for the write data buffer, as the data can come from the external I/O or from the internal memory banks. Since the transfer process can be conducted row-by-row, additional buffer is not needed.

Within each bank, there are 4×4 sub-arrays. A PL driver is added to each array to provide the PL pulse for non-volatile mode, which is marked as (6) in Fig. 5. The enable signal of the PL driver comes from the 1-bit bank mode signal (marked as (6)), which is stored in the bank status table in each bank (marked as (3)). If the bank mode bit is “1”, the PL driver will be enabled for the non-volatile mode operation. Otherwise, it is grounded for the volatile mode. The bank status table also contains 4-bit row refresh signal (Row_ref) that indicates whether a row of sub-arrays need refresh or not for the volatile mode without refresh. The row refresh signal is AND with the RBSEL signal to generate Ref_sel signal (marked as (4-2)). If the Ref_sel is 0, 0 will be selected over the output from pre-decoder so that a row of sub-arrays that does not need refresh can be skipped (marked as (4-1)).

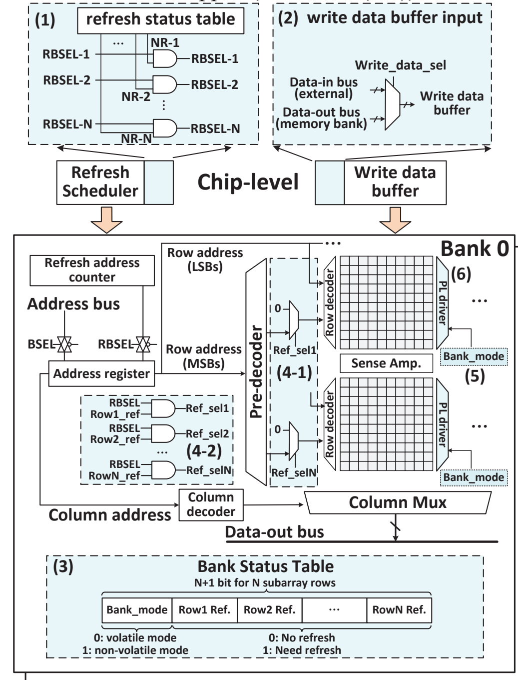

Fig. 5. An illustration of the dual-mode memory buffer design. The differences between conventional eDRAM design are highlighted in blue.

VI. DNN Accelerator Benchmark Methodologies

The performance of dual-mode buffer is estimated by modifying the DESTINY simulator [17]. 32nm node is assumed. Other buffer technologies including SRAM, eDRAM and FeRAM (non-volatile mode only) are selected as the baselines. For dual-mode buffer, the refresh interval of the volatile mode is 40μs [18], which is the same as the eDRAM. For dual-mode buffer, eDRAM and FeRAM, sufficient buffer capacity is assumed to store the intermediate data on-chip. Besides, these three designs use the same capacitor size (D=80nm). For eDRAM, high-k linear dielectrics (κ=35) is assumed and the capacitance of the storage node is about 12.4fF. For dual-mode buffer and FeRAM, the ferroelectric dielectric listed in TABLE I is used. For SRAM, two cases, one with sufficient SRAM capacity (SRAM-size) and the other with the same area as eDRAM (SRAM-area) are considered. The cell size of SRAM is $145F^2$ and the sub-array size is 128×128.

TABLE II. The Performance of a 2-Mb Bank for Different Memory Technologies (32nm Tech. Node)

	SRAM	eDRAM	FeRAM	Dual-mode buffer
Cell size	$145F^2$	$38F^2$	$38F^2$	$38F^2$
Sub-array size	128×128	256×512	256×512	256×512
Area(mm²)	0.456	0.121	0.141	0.141
Read(fJ/bit)	331	206	503	214(v-mode) 503(nv-mode)
Write(fJ/bit)	305	182	706	204(v-mode) 706(nv-mode)
Refresh interval(μs)	--	40	--	40
Refresh energy(fJ/bit)	--	74	--	81
Leakage(μW)	328.7	8.7	10.7	10.7

The performance of a 2Mb bank for different memory technologies are listed in TABLE II. Multiple banks are grouped for the on-chip buffer design. It should be noted that the energy consumption at interconnect is included in the read and write energy. Therefore, SRAM shows higher read and write energy than dual-mode buffer (volatile mode) and eDRAM due to its larger bank size.

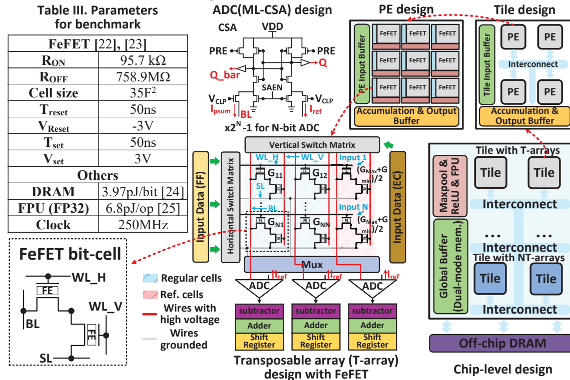
Table III. Parameters for benchmark

FeFET [22], [23]	
R_{ON}	95.7 kΩ
R_{OFF}	758.9MΩ
Cell size	$35F^2$
T_{reset}	50ns
V_{Reset}	-3V
T_{set}	50ns
V_{set}	3V
Others	
DRAM	3.97pJ/bit [24]
FPU (FP32)	6.8pJ/op [25]
Clock	250MHz

Fig. 6. The CIM-based DNN accelerator architecture with FeFET weight cells and dual-mode memory buffer as global buffer.

The system-level performance is evaluated with a CIM-based DNN accelerator using NeuroSim framework [19] at 32nm node. The accelerator architecture for benchmark is shown in Fig. 6, which follows the design in [20]. HZO-based FeFET is assumed as the weight cell for the CIM array for the process compatibility with the HZO-based 1T1C buffer memory. As demonstrated in [6], HZO can be integrated at the BEOL for a FeFET. The FeFET parameters are listed in TABLE III. There are two types of CIM arrays: transposable arrays (T-array) with twin-FeFET weight cell are used for FF and EC while non-transposable arrays (NT-array) with single FeFET weight cell are used for GC. During GC, the activations are programed into the NT-arrays and errors are used as the input. A 5-bit ADC is implemented by multi-level current-mode sense amplifier (ML-CSA) to ensure the partial sum accuracy. Each FeFET cell stores 1-bit weight to ensure memory window under variability. Therefore, an n-bit weight is split into n FeFET cells and shift-and-add is needed. In a processing element (PE), 9 CIM arrays are grouped. There are two types of PEs that contain either T-arrays or NT-arrays. On the chip level, the global buffer is implemented by the dual-mode memory. Floating point unit (FPU) is used for the FP32 operations in the normalization layer.

Three DNN models are considered: ResNet-20 (~0.27M weights) and DenseNet-40 (~1M weights) for CIFAR-10, and ResNet-18 (~11M weights) for ImageNet, where 8MB, 128MB and 128MB global buffer are budgeted, respectively. The weight, activation and error are quantized to 8-bit. The gradient precision is FP32 for accumulation over a batch. It is quantized to 8-bit for WU. The efficacy of low precision training scheme has been verified [21].

VII. RESULTS AND DISCUSSIONS

The buffer mode configuration for different intermediate data types are shown in Fig. 7. For ResNet-20 with short $T_{pipeline}$ and pipeline length, volatile mode (v-mode) is prefrred. All the errors (E) and gradients (G) are stored as volatile mode without refresh as $T_{pipeline}$ is shorter than the refresh interval. About 67% of the activations (A) are stored as the non-volatile mode (nv-mode) as the activations from shallow layers need to be stored in the buffer for a long time for GC. For DenseNet-40 with short $T_{pipeline}$ but long pipeline length, nv-mode is preferred for the activations while errors and gradients are stored as the v-mode with refresh. When $T_{pipeline}$ further increases to 926.1μs in ResNet-18, the buffer mode switches to nv-mode for gradient while it is still v-mode for error. It is because that errors are more read-intensive than gradients and that the nv-mode has higher read energy.

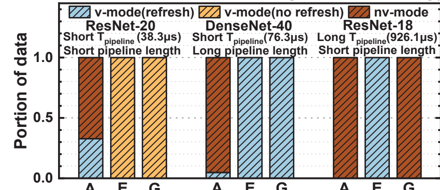

Fig. 7. The buffer mode configurations for different intermediate data types, where activation (A), errors (E) and gradients (G) are considered.

The energy efficiency for DNN training is shown in Fig. 8 (a), where the baseline with sufficient SRAM size shows the best performance only for the ResNet-18 model. For ResNet-20, the energy efficiency of dual-mode memory is slightly higher than SRAM-size and eDRAM while it is 26.6% higher than FeRAM. It is attributed to the higher read and write energy of FeRAM, as shown in the energy breakdown in Fig. 8(b). When $T_{pipeline}$ or pipeline length becomes longer, dual-mode buffer shows higher energy efficiency than eDRAM by storing the data with long lifetime (e.g. activations) into nv-mode. For example, the energy efficiency of dual-mode buffer is 53.8% and 44.7% higher than eDRAM for DenseNet-40 and ResNet-18, respectively, where a large portion of energy is consumed by eDRAM refresh. Compared with FeRAM, the energy efficiency advantage of dual-mode buffer becomes less significant for these two DNN models that nv-mode is preferred, which are 23.4% and 1.2%, respectively. The SRAM-area baseline shows the worst performance for the three DNN models because of insufficient on-chip buffer capacity ,which leads to frequent off-chip DRAM access.

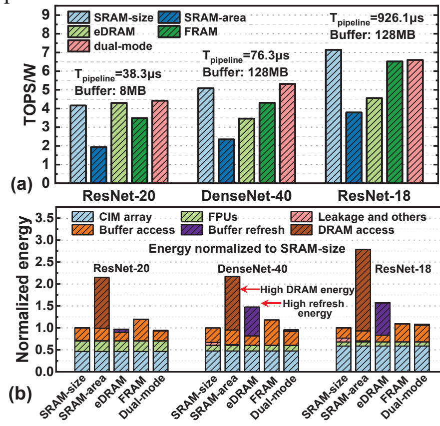

Fig. 8. (a) The energy efficiency for training. (b) The energy breakdown that is normalized to the baseline SRAM-size that has sufficient SRAM.

The above results indicate that different buffer technologies are preferred for different intermediate data types and different DNN models. By using dual-mode buffer,

the DNN accelerator shows 33.8%, 17.1% and 109.4% higher average energy efficiency than the baslines using eDRAM, FeRAM and SRAM with the same area, respectively.

The chip area is shown in Fig. 9, which is normalized to the area of eDRAM. Compared to eDRAM, the area overhead for a dual-mode memory bank is about 16.5%, as listed in TABLE II. It is mainly attributed to the larger transistor size in the periphery circuitry for high voltage operation during non-volatile mode. The average chip area overhead is about 5.7% for the three DNN models. SRAM-size shows 43% ~ 128% larger chip area than dual-mode buffer, due to the large cell size of SRAM.

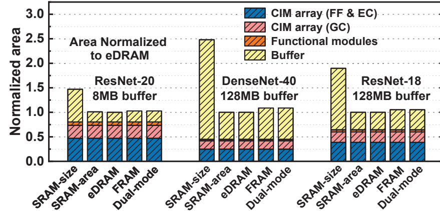

Fig. 9. The total chip area breakdown. The area is normalized by eDRAM.

During standby, the parameters for the normalization layer need to be kept for the subsequent inference. The leakage power for dual-mode buffer includes the power consumption to store the data as nv-mode, assuming 1 second idle time. For eDRAM baseline, only the refresh power for the banks that store data is considered. FeRAM does not need refresh as it is non-volatile. The standby power is plotted in Fig. 10. The accelerator designs using SRAM or eDRAM show high standby power because of the high leakage power of SRAM and the refresh power of eDRAM, respectively. Compared with SRAM-size, dual-mode buffer shows 26.8×~47.5× standby power reduction because of smaller buffer leakage power. Compared with eDRAM, dual-mode buffer shows 1.5×~10.6× standby power reduction because of the elimination of refresh power.

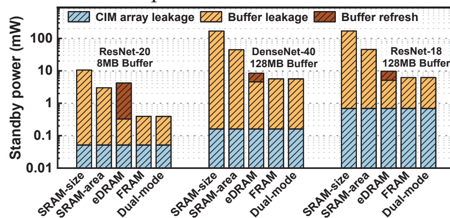

Fig. 10. The standby power breakdown.

The write endurance for the non-volatile mode is a potential concern. The non-volatile buffer that stores activations are written by 5 times for each image, considering N_{write}=1, N_{read}=4 and destructive read for 150 epochs. The total number of write operation is 3.75×10^7 with 50k CIFAR-10 training images and 7.5×10^8 with 1M ImageNet training images, which is substantially less than the reported 10^{11} write endurance of FeRAM under 2.5V operation voltage [7].

VIII. CONCLUSIONS

By using dual-mode buffer as the global buffer, the energy efficiency of CIM-based DNN accelerator can be improved by an average of 33.8%, 17.1% and 109.4%, compared with the baseline designs using eDRAM, FeRAM, SRAM with the same area, respectively. Besides, it also reduces the standby power by storing the data as non-volatile mode. The proposed dual-mode buffer memory is also applicable to other well-established accelerators including the TPU-like digital domain-specific architectures.

ACKNOWLEDGMENT

This work is supported by ASCENT, one of SRC/DARPA JUMP centers.

REFERENCES

[1] N. P. Jouppi et al., "In-datacenter performance analysis of a tensor processing unit," *ISCA*, 2017.
[2] Y. Chen et al., "DaDianNao: A machine-learning supercomputer," *MICRO*, 2014.
[3] S. S. Iyer et al., "Embedded DRAM: Technology platform for the blue gene/L chip," *IBM Journal of Research and Development*, vol. 49, no. 2.3, pp. 333-350, 2005.
[4] F. Tu et al., "RANA: Towards efficient neural acceleration with refresh-optimized embedded DRAM," *ISCA*, 2018.
[5] J. Muller et al., "Ferroelectric hafnium oxide based materials and devices: assessment of current status and future prospects," *ECS Journal of Solid State Science and Technology*, vol. 4, no. 5, pp. N30-N35, 2015.
[6] K. Ni et al., "SoC logic compatible multi-bit FeMFET weight cell for neuromorphic applications," *IEDM*, 2018.
[7] J. Okuno et al., "SoC compatible 1T1C FeRAM memory array based on ferroelectric $Hf_{0.5}Zr_{0.5}O_2$," *Symp. VLSI Technology*, 2020.
[8] L. Deng et al, "Model compression and hardware acceleration for neural networks: A comprehensive survey," *Proceedings of the IEEE*, vol. 108, no. 4, pp. 485-532, 2020.
[9] P. Polakowski et al., "Ferroelectric deep trench capacitors based on Al:HfO_2 for 3D nonvolatile memory applications," *IEEE IMW*, 2014.
[10] A. Sheikholeslami and P. G. Gulak, "A survey of circuit innovations in ferroelectric random-access memories," *Proceedings of the IEEE*, vol. 88, no. 5, pp. 667-689, 2000.
[11] F. Hamzaoglu et al., "A 1Gb 2GHz embedded DRAM in 22nm tri-gate CMOS technology," *ISSCC*, 2014.
[12] T. Kirihata et al., "An 800-MHz embedded DRAM with a concurrent refresh mode," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 6, pp. 1377-1387, 2005.
[13] A. Agrawal et al., "Refrint: Intelligent refresh to minimize power in on-chip multiprocessor cache hierarchies," *HPCA*, 2013.
[14] K. Ni et al., "A circuit compatible accurate compact model for Ferroelectric-FETs," *Symp. VLSI Technology*, 2018
[15] L. Song et al., "PipeLayer: A pipelined RRAM-based accelerator for deep learning," *HPCA*, 2017.
[16] X. Peng et al., "Optimizing weight mapping and data flow for convolutional neural networks on RRAM based processing-in-memory architecture," *ISCAS*, 2019.
[17] M. Poremba et al., "DESTINY: A tool for modeling emerging 3D NVM and eDRAM caches," *DATE*, 2015. https://code.ornl.gov/3d_cache_modeling_tool/destiny
[18] D. Fainstein et al., "Dynamic intrinsic chip ID using 32nm high-k/metal gate SOI embedded DRAM," *Symp. VLSI Circuits*, 2012
[19] X. Peng et al., "DNN+NeuroSim: An end-to-end benchmarking framework for compute-in-memory accelerators with versatile device technologies," *IEDM*, 2019. https://github.com/neurosim/DNN_NeuroSim_V1.1
[20] Y. Luo and S. Yu, "Accelerating deep neural network in-situ training with non-volatile and volatile memory based hybrid precision synapses," *IEEE Transactions on Computers*, vol. 69, no. 8, pp. 1113-1127, 2020.
[21] G. Yang et al., "SWALP: Stochastic weight averaging in low-precision training," 2019, arXiv:1904.11943.
[22] S. Dünkel et al., "A FeFET based super-low-power ultra-fast embedded NVM technology for 22nm FDSOI and beyond," *IEDM*, 2017.
[23] P. Wang et al., "Drain–erase scheme in ferroelectric field-effect transistor—Part I: device characterization," *IEEE Transactions on Electron Devices*, vol. 67, no. 3, pp. 955-961, 2020.
[24] M. O'Connor et al., "Fine-Grained DRAM: Energy-Efficient DRAM for Extreme Bandwidth Systems," MICRO, 2017.
[25] S. Galal and M. Horowitz, "Energy-efficient floating-point unit design," *IEEE Transactions on Computers*, vol. 60, no. 7, pp. 913-922, 2011.