An All-Weights-on-Chip DNN Accelerator in 22nm ULL Featuring 24×1 Mb eRRAM

*Zhehong Wang[1], *Ziyun Li[1,2], Li Xu[1], Qing Dong[3], Chin-I Su[4], Wen-Ting Chu[4], George Tsou[4], Yu-Der Chih[4], Tsung-Yung Jonathan Chang[4], Dennis Sylvester[1], Hun Seok Kim[1], David Blaauw[1]
[1]University of Michigan, Ann Arbor, MI, [2]Facebook, Seattle, WA, [3]TSMC, San Jose, CA, [4]TSMC, Hsinchu, Taiwan

***Abstract*—We present a DNN accelerator in 22nm ULL CMOS featuring 24×1 Mb embedded RRAM. The accelerator, composed of 4 PEs and 512 MACs, achieves 0.96 TOPS/W at 120 MHz with 0.8 V VDD. Each PE contains 6 RRAM macros, equipped with a dynamic clamping offset-canceling sense amplifier that offers sub-μA current input offset.**

Keywords— DNN Accelerator, RRAM, Offset Canceling Sense Amp

Hardware accelerators for Machine Learning (ML) using Deep Neural Networks (DNNs) have gained significant interest in recent years. Various approaches to improve power efficiency while maintaining neural network inference accuracy have been proposed, including maximizing weight reusage [1], dynamic precision scaling [2], sparsity awareness [3], analog/in-memory computation [4], and others. These prior DNN implementations, however, are not readily scalable to high-accuracy ML due to limited compute precision. Furthermore, although non-volatile memories (NVM) such as RRAM, have been extensively exploited for analog/in-memory computation [4], there has been little work that leverages RRAM's high density and low standby power for dedicated weight storage in large-scale digital ML (versus a general purpose non-volatile microcontroller [5]). We present the first digital DNN accelerator featuring eRRAM for dedicated weight storage to eliminate off-chip weight accesses, thereby reducing the overall system operating power. The design employs a 4-Core architecture in 22nm ULL CMOS technology with 24×1 Mb embedded RRAM. Using on-the-fly weight decompression, we achieve on average ~1.5 b/weight resulting in a total capacity of 16 M weights on-chip. To reliably read and write the RRAM, we propose a dynamic clamping offset-canceling sense amplifier (DCOCSA) achieving sub-μA input-sensing offset and a Write-Verify scheme for reliable programing. Combined with a mesh-connected processing element (PE) architecture and 8 Mb shared SRAM, the proposed DNN accelerator operates at 120 MHz at 0.8 V VDD, achieving 0.96 TOPS/W.

Fig. 1(a) depicts the overall architecture of the proposed CNN accelerator with embedded RRAM. The proposed design consists of 4 mesh-connected PEs, where each PE includes 1) an array of 128 MACs, 2) local 4 KB weight, 3) 32 KB input SRAM buffer, 4) 6 Mb RRAM for parameter storage, and 5) a 32 KB instruction cache. To enable flexible CNN/ResNet network mapping on the proposed architecture, each PE allows neighboring PEs to access its local SRAM. An 8Mb global 1R/1W SRAM that supports multi-cast (coalesced) reading is shared across all PEs. Overall, the proposed flexible architecture with full parallelism achieves 123 GOPS.

Fig. 1(b) details the operation of a single PE unit. During neural network operation, compressed weights are first read from the RRAM and then decompressed into 4kB interleaved weight buffers for high bandwidth frequent accesses. Inspired by [6], convolutions are performed by shifting input activations (IAs), which executes 1-D convolution detection in a row of MAC units in k (kernel size) cycles. This operation is then repeated on the second row of IAs to perform the 2-D convolution. The partial outputs are accumulated locally in each MAC. Each PE uses 4 clusters of 32 mesh-connected MACs for parallel processing. A 32 KB instruction SRAM with 256 b VLIW ISA controls PE operation and synchronization over various layer functions.

To enable all weights on chip, we leverage state-of-the-art weight compression [7] that combines weight pruning, non-uniform quantization, run-length coding, and Huffman coding. On average, the design with 8b weight precision achieves ~2.7 bit per weight for convolutional layers and 0.25 bit per weight for fully connected layers. The weights are stored in the RRAM as variable length packets with multiple 96b words. Each packet contains a layer specification followed by Huffman-encoded weights and run-length coded indices (Fig. 2) so that invalid packets due to static RRAM errors can be bypassed/overwritten by the subsequent correct packets.

Fig. 3(a) shows the block diagram of the custom-designed 1Mb RRAM bank, which uses a butterfly architecture with 4 256×1024 RRAM arrays with 32 b word length, employing a common SL cell arrangement [8]. Thus, the column-wise peripherals include an equalizer to reduce the mutual influence between neighboring columns.

To address the high variation common in RRAM conductance, a 2-stage offset-cancelling current-mode SA is proposed in Fig. 4. The first stage uses two cross-coupled current sampling branches similar to the scheme in [9], which doubles the input current difference and effectively halves the offset. In addition, the first stage incorporates dynamic clamping, instead of typical static clamping, to bring down the bit line settling time and increase the sensing speed. Unlike conventional clamping amplifiers, which are large and power hungry, a carefully designed self-biased inverter provides the feedback loop. Settling time is reduced by 50% (simulation) compared to a static clamping SA. The second stage provides further amplification and offset-reduction with a single-cap auto-zero regenerative amplifier [10]. Fig. 4 shows the operation of the proposed SA. In the first step, the input and output of the inverter are shorted to self-bias the clamp transistors, with bias voltage sampled on the C_i's. Meanwhile, the regenerative amplifier of the second stage is also shorted to sample the offset and cancel it out in the following steps. This step overlaps with address decoding to avoid timing penalty. Then the shorted inverter in phase one is disconnected to function as a negative feedback amplifier and the WL is turned on to allow the two diode-connected PMOS headers to sample the current, I_{ref} and I_{cell}, on their respective branches. After the current settles, the two headers are switched to the other branch and function as a current source, which generates a doubled current difference $2(I_{cell} - I_{ref})$ as input to the second stage. Finally, the second stage is fired and latches the output. Voltage waveforms of the internal nodes are shown in Fig. 3(b). A sub-μA current offset is achieved at 21 μA common mode input under 1.2 V VDD from Monte Carlo simulation; Fig. 5(a) gives the distribution across temperature.

Using a fine-grained iterative Write-Verify control approach, each bit in a word is separately controlled based on the read result, ruling out correlation between fast and slow cells, which alleviates locality-dependent variation. Meanwhile, with Write-Verify, each bit automatically adapts to the corresponding SA offset, further reducing the variation requirements. Fig. 5(b) illustrates the block diagram of the Write-Verify control. Following a write request, each RRAM cell of the target address is read out first to compare with the input data (DG), initiated by the global control. In the case that the read-out value (d) of a cell is the same as the corresponding bit in DG, the write process of that cell concludes for better endurance. On the other hand, the cell is programed to the desired value by the iterative Write-Verify process. Fig. 6 shows a measured resistance distribution of RRAM with ~10k samples.

Implemented in 22nm ULL CMOS technology, the test chip achieves 120 MHz core clock frequency at 0.8 V VDD and consumes 42.4mW when evaluating a CNN layer of size 4×3×3×16. Fig. 7(a) shows the tested Power/Freq vs. VDD plot. The RRAM clock is hard coded to be half of the core clock. A power breakdown of the four RRAM power domains is shown in Fig. 7(b), with 1V for the SA and control, 1.4V for the WL, 1.25V for the column mux, and 1.1V for the inverter amplifier. The proposed accelerator consumes 127.9 mW in total including weight decompression and transfer from RRAM to SRAM resulting in a power efficiency 0.96 TOPS/W. Fig. 8 compares the work to recent NN accelerators. The proposed design achieves the highest number of on-chip-stored weights and is also the only design employing non-volatile memory as dedicated weight storage, reducing standby power for-edge devices. Fig. 9 shows the die photo.

ACKNOWLEDGMENT

This work was supported by university joint development program and university shuttle program of TSMC, and ADA JUMP center.

REFERENCES

[1] Y. Chen, et al., ISSCC, pp. 262-264, 2016
[2] J. Lee, et al., ISSCC, pp. 218-220, 2018.
[3] P. Whatmough, et al, ISSCC pp. 242-4, 2017
[4] C. Xue, et al., ISSCC, pp. 388-390, 2019.
[5] T. Wu, et al., ISSCC, pp.226-228, 2019.
[6] Z. Li, et al., ISSCC, pp. 134-136, 2019.
[7] S. Han, et al., ICLR, 2016.
[8] C-C. Chou, et al., ISSCC, pp. 478-480, 2018.

*The first two authors are equally contributed.

978-1-7281-9942-9/20/$31.00 ©2020 IEEE

Authorized licensed use limited to: University of Michigan Library. Downloaded on August 19,2020 at 17:53:34 UTC from IEEE Xplore. Restrictions apply.

[9] P. Jain, et al., ISSCC, pp. 212-214, 2019.
[10] Q. Dong, et al., ISSCC, pp. 480-482, 2018.
[11] K. Ueyoshi, et al., ISSCC, pp.216-218,2018
[12] J. Zhang, et al., VLSI, pp. 306-307, 2019.
[13] Z. Yuan, et al., VLSI, pp. 33-34, 2018.

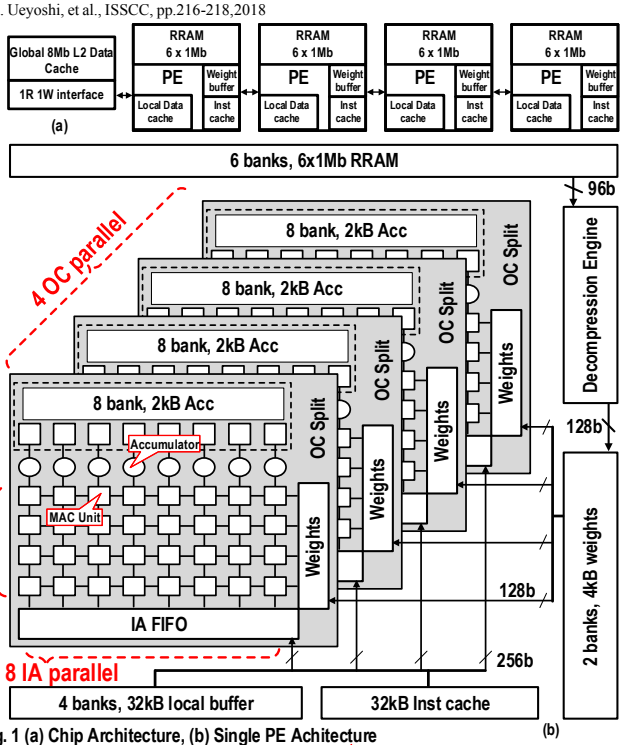

Fig. 1 (a) Chip Architecture, (b) Single PE Achitecture

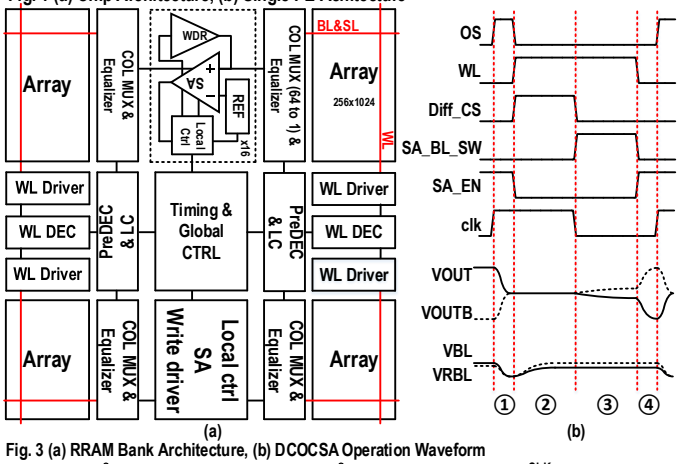

Fig. 3 (a) RRAM Bank Architecture, (b) DCOCSA Operation Waveform

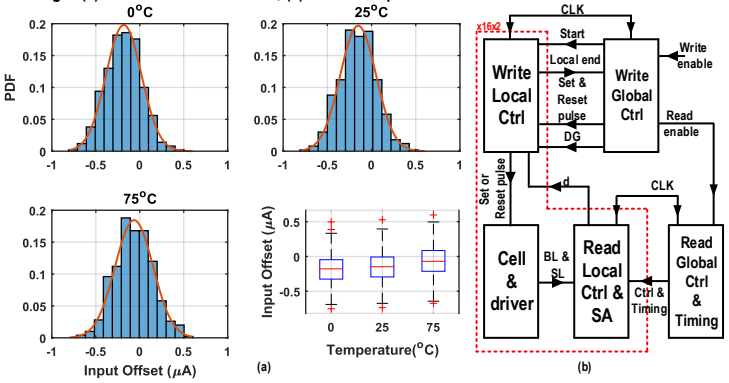

Fig. 5 (a) DCOCSA Input Current Offset MC Simulation Distribution, (b) Write-Verify Block Diagram

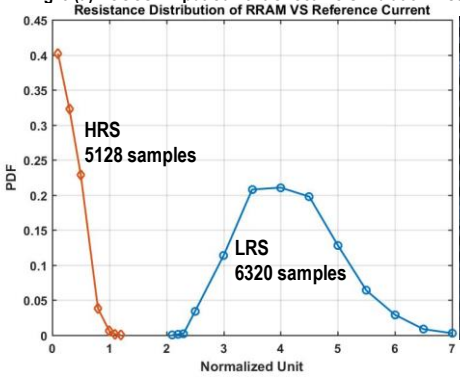

Fig. 6 Tested Resistance Distribution VS Reference Current

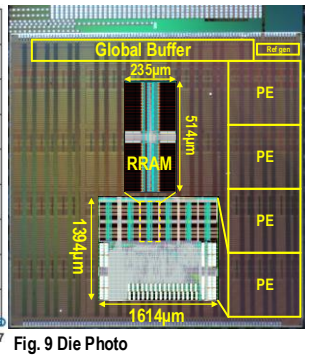

Fig. 9 Die Photo

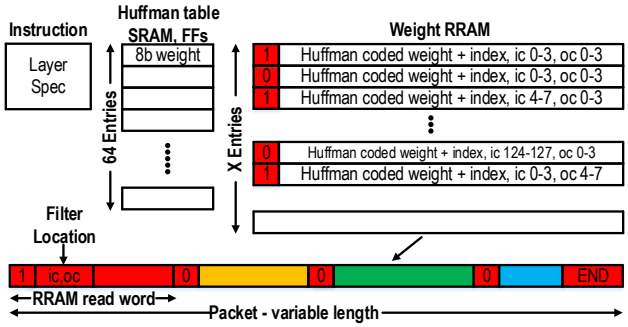

Fig. 2 Weight Packet Composition

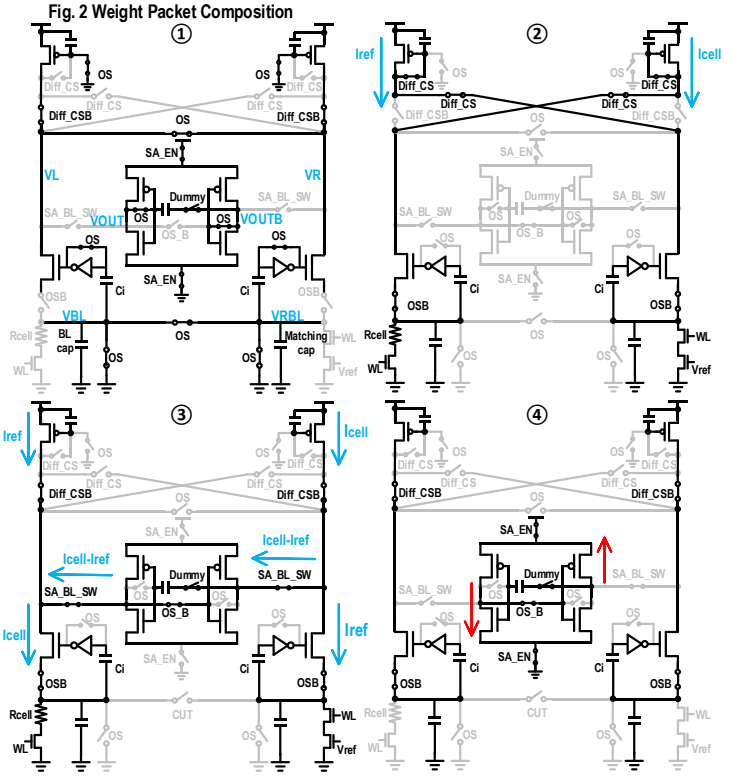

Fig. 4 Illustration of Dynamic Clamping Offset Canceling Sense Amplifier Operation

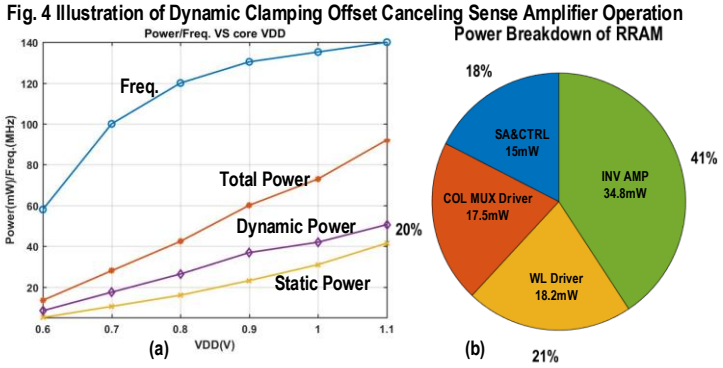

Fig. 7 (a) Tested Core Freq/Power VS Core VDD, (b) Tested Power Breakdown of RRAM

	This Work	QUEST[11]	SNAP[12]	STICKER[13]	UNPU[2]
Technology	**ULL 22nm**	40nm	16nm	65nm	65nm
On-chip RAM(B)	**3M RRAM 1.3M SRAM**	7.68M 96M 3D SRAM	280.6K	170K	256K
Max On-chip Weight	**16M@8b Non-Volatile**	15.36M@4b Volatile	140.3K@16b Volatile	170K@8b Volatile	256K@8b Volatile
Off-chip Memory	**No**	Yes	Yes	Yes	Yes
MACs	**4x128 (8x8b)**	24x512 (1x1b log)	252 (16x16b)	256 (8x8b)	4x576 (1x16b)
Voltage (V)	**1.0-1.2 RRAM 0.6-1.1 Core**	1.1	0.55-0.8	0.67-1.0	0.63-1.1
Freq. (MHz)	**60 RRAM 120 Core**	300	33-480	20-200	200
TOPS/W	***0.96@8b**	**0.59@4b	***3.61@16b	***1.038@8b	***5.57@8b
GOPS	**123@8b**	1960@4b	65.52@16b	102@8b	690@8b
Power (mW)	**127.9 @120MHz**	3300 @300MHz	364 @480MHz	284.4 @200MHz	297 @200MHz
Area (mm^2)	**10.8**	122	2.4	12	16

*Including power of loading weights from RRAM to SRAM and MAC arrays
**Including power of loading weights from 3D SRAM to on-chip SRAM and MAC arrays
*** Excluding power of loading weights from off-chip memory

Fig. 8 Comparison Table

Authorized licensed use limited to: University of Michigan Library. Downloaded on August 19,2020 at 17:53:34 UTC from IEEE Xplore. Restrictions apply.