24.5 A Twin-8T SRAM Computation-In-Memory Macro for Multiple-Bit CNN-Based Machine Learning

Xin Si[1,2], Jia-Jing Chen[1], Yung-Ning Tu[1], Wei-Hsing Huang[1], Jing-Hong Wang[1], Yen-Cheng Chiu[1], Wei-Chen Wei[1], Ssu-Yen Wu[1], Xiaoyu Sun[3], Rui Liu[3], Shimeng Yu[4], Ren-Shuo Liu[1], Chih-Cheng Hsieh[1], Kea-Tiong Tang[1], Qiang Li[2], Meng-Fan Chang[1]

[1]National Tsing Hua University, Hsinchu, Taiwan
[2]University of Electronic Science and Technology of China, Chengdu, China
[3]Arizona State University, Tempe, AZ
[4]Georgia Institute of Technology, Atlanta, GA

Computation-in-memory (CIM) is a promising avenue to improve the energy efficiency of multiply-and-accumulate (MAC) operations in AI chips. Multi-bit CNNs are required for high-inference accuracy in many applications [1-5]. There are challenges and tradeoffs for SRAM-based CIM: (1) tradeoffs between signal margin, cell stability and area overhead; (2) the high-weighted bit process variation dominates the end-result error rate; (3) trade-off between input bandwidth, speed and area. Previous SRAM CIM macros were limited to binary MAC operations for fully connected networks [1], or they used CIM for multiplication [2] or weight-combination operations [3] with additional large-area near-memory computing (NMC) logic for summation or MAC operations.

This work presents a configurable SRAM CIM macro with 1-4b inputs, 1-5b weights, and a MAC-value (MACV) of up to 7b outputs. The CIM uses (1) a compact-rule compatible twin-8T (T8T) cell for weighted CIM MAC operations to reduce the area overhead and vulnerability to process variation; (2) an even-odd dual channel (EODC) array to double the input bandwidth; (3) a two's complement mapping and processing unit (C2PU) to enable MAC operations using positive and negative weights within a cell array, so as to reduce the area overhead and computation time. Our fabricated 55nm 64 × 60b T8T macro demonstrated the first configurable multiple-bit SRAM-CIM. This work also achieves the fastest CNN multi-bit MAC operations (3.2ns for 1b input, 2b weight and 3b MACV, and 5ns for a 2b input, 5b weight and a 5b MACV).

Figure 24.5.2 shows our configurable multi-bit T8T SRAM CIM macro, comprising of T8T-SRAM cells. Also shown is an even/odd dual-channel (EODC) array structure, a two's complementary processing unit (C2PU), an output combiner (OC), and a configurable reference column (CRC). The j-bit signed weights (W_i[j-1:0]) of each $n{\times}n$-CNN filter are stored in the T8T array using $2n^2$ rows and j columns, in two's complement form. Each T8T cell comprises of two read-decoupled 8T (RD8T) cells: a most-significant 8T (M8T) and least-significant 8T (L8T). The transistor width of N2/N1 in M8T is 2× that in L8T to provide a weighted cell current ($I_{\text{MC-M8T}} = 2{\cdot}I_{\text{MC-L8T}}$). The read decoupled transistor (RDT) structure enables, (1) full(large)-voltage swing on the RBL to increase the signal margin without encountering the read-disturb or write-disturb issues common for a 6T-SRAM CIM [1, 3]; (2) a small area overhead by using the foundry's compact-rule 8T-SRAM cells rather than the dedicated large-area logic-rule SRAM cells. Each input (IN) is applied to a read-WL (RWL) that supports binary- (2 levels: 0V and V_{WLL3}) or 2b- (4 levels: 0V, V_{WLL1}, V_{WLL2}, & V_{WLL3}) input precision. A dual-rail power distribution network allows for V_{WLL3} to exceed the standard supply voltage (V_{DD}). The WL voltage is used to generate a weighted cell current (I_{MC}): $^1/_3{\cdot}I_{\text{MC}}$, $^2/_3{\cdot}I_{\text{MC}}$, and I_{MC}. V_{WLL1}, V_{WLL2} and V_{WLL3} are determined by 1k Monte-Carlo simulations using different MAC patterns. For $n \times n$ MAC operations n^2 inputs (IN) are applied to n^2 RWLs in parallel, for each channel. Each T8T cell performs 2b-input 2b-weight multiplication with a discharge cell current ($I_{\text{MC-T8T}} = I_{\text{MC-M8T}} + I_{\text{MC-L8T}}$) based on the input-weight-product (IWP, which is IN[1:0] × W[1] + IN[1:0] × W[0]). Summing all I_{MC} on a RBL against a BL pull-up transistor results in a RBL voltage (V_{RBL}) that is the summation of n^2 2b-input 2b-weight products ($\Sigma\text{IN}_i[1{:}0] \times \text{W}_i[1{:}0]$). By combining several contiguous RBL lines, multi-bit MAC operations ($\Sigma\text{IN}_i[j\text{-}1{:}0] \times \text{W}_i[k\text{-}1{:}0]$) can be implemented via (C2PU) and OC.

The developed EODC scheme (Fig. 24.5.3) extends the bandwidth of CNN operations and reduces energy consumption. The even-row T8T cells, in a column, are connected to RBL-even (RBLE), while the odd-row cells are connected to RBL-odd (RBLO). This structure reduces in half the number of cells (and parasitic load) on an RBL in a typical cell array. By using the original RBL trace for each RD8T cell, the RBLO/RBLE in EODC does not consume additional area. The EODC scheme supports two operating modes: a single-channel (SC) and a two-channel-accumulation (2CA) mode. In SC mode, one of the even/odd channels (n^2 RWLE/RWLOs) is activated and its RBLs are connected to C2PUs. Thanks to the reduced BL-load in EODC, the settling time of V_{RBL} and power consumption on RBLs is lower than that without EODC. In 2CA mode, n^2-odd RWLs (RWLO) and n^2-even RWLs (RWLE) are activated simultaneously for parallel MAC operations on RBLEs and RBLOs. RBLE and RBLO on the same T8T column are dotted. The MACVs of these two channels are then accumulated using the same C2PU to double the bandwidth over that in SC mode. The even and odd channels both support configurable (1/2/4-bit) input precision. When the input precision is binary or 2b, T8T CIM requires only one cycle to compute MACV. When the input precision is 4b (IN[3:0]), each input is applied to an assigned channel using two cycles: IN[3:2] and IN[1:0]. The OC then combines the MACV results from the 1st cycle and 2nd to output the final multi-bit MACV.

Figure 24.5.4 presents the proposed two's-complement weight mapping and C2PU. In prior work [3], only multi-bit weights are implemented within an array, and all MAC operations are realized outside of the memory array (as NMC). In this work, the signed weights are stored in the T8T array in two's-complement form, which allows for the support of multiple RWL activations and to sum both positive and negative multiplied results together within the memory array. Owing to the proposed two's complement weight mapping scheme, the NMC area could be reduced by 158% compared to [3] for 2b-input 4b-weight MACs. For a given IN[1:0] codeword on an odd channel, the resulting RBLs are as follows: RBLS=$\Sigma(\text{IN}_i[1{:}0] \times \text{W}_i[4])$, RBLO[1]=$\Sigma(\text{IN}_i[1{:}0] \times \text{W}_i[3{:}2])$, RBLO[0]=$\Sigma(\text{IN}_i[1{:}0] \times \text{W}_i[1{:}0])$. When using C2PU and configurable reference columns (CRC), KS in Fig. 24.5.4 is switched from V_{DD} to MACS to lower the voltage of SUM (V_{SUM}) by $^1/_2$. In the meantime, K1 and K0 are switched from MAC[1:0] to V_{DD} to increase V_{SUM} by $^1/_8$, and $^1/_{32}$. KS, K1, and K0 have the similar structure to a 2:1 multiplexer, except with differently weighted capacitor loading. The total summation (TS) with a 5b-MACV is derived as follows: MACS × (-16) + MAC[1] × (+4) + MAC[0] × (+1) = $\Sigma(\text{IN}_i[1{:}0] \times \text{W}_i[4{:}0])$.

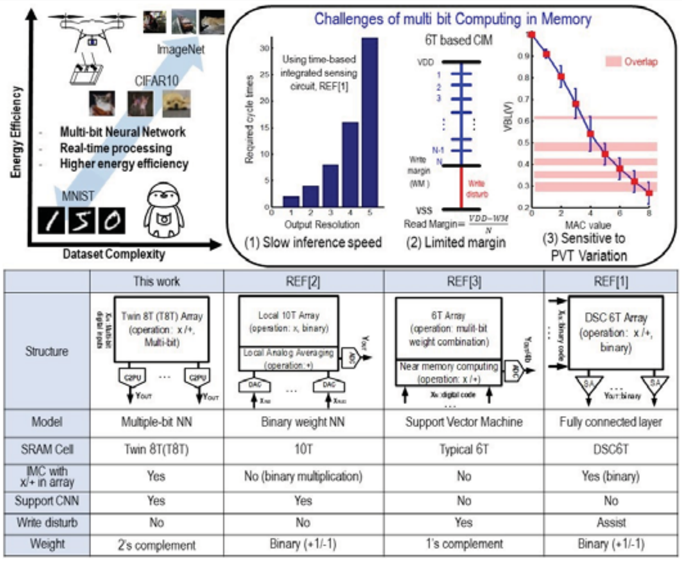
Figure 24.5.5 shows the simulated performance of the proposed schemes. A larger voltage swing (with no write-disturb) and a larger transistor width at the read-port allows the T8T cell to achieve a 1.4-1.6× increase in sensing margin, compared to 6T cells with the same MACV precision. Thanks to its compact area and double throughput, the EODC (SC mode) scheme achieves a throughput/area FoM that is over 1.98× larger than that of fully-serial and fully-parallel input/weight structures. When the EODC and 2CA schemes are combined they achieve over 2.35× better FoM. When using two's-complement weight mapping, C2PU and OC achieve computation times that are 2.6× (3b MACV) and 4.85× (5b MACV) faster than [2]. This is also 2.04× (3b MACV) and 2.08× (5b MACV) faster than [3]. [1] and [3] do not use a signed MACV.

Figure 24.5.6 presents the measured results from a test chip fabricated using a 55nm 64 × 60b T8T-SRAM CIM macro. For CNN operations with 3 × 3 kernels the shmoo plot confirms a $t_{\text{AC-MAC}}$ of 5ns for a 2b-input 5b-weight 5b-MACV. Using 4b-inputs, 5b-weights and a 7b-MACV the array achieves a 90.42% accuracy in the inference of 10k CIFAR-10 images. This work achieves over a 22× improvement in the energy-efficiency × throughput/capacity FoM (TOPS/W × GOPS/KB) compared to prior SRAM-CIM work [1-3]. Figure 24.5.7 presents the die photo and a chip summary.

Acknowledgements:
The authors would like to thank MOST-Taiwan, CIC, NTHU-TSMC JDP, NTHU-MTK-JDP, M31 for financial and manufacturing support.

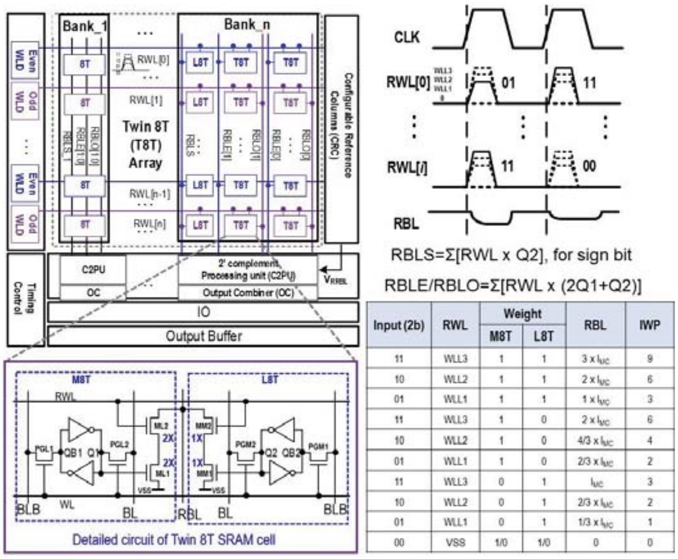
References:
[1] W.-S. Khwa, et al., "A 65nm 4Kb Algorithm-Dependent Computing-in-Memory SRAM Unit-Macro with 2.3ns and 55.8TOPS/W Fully Parallel Product-Sum Operation for Binary DNN Edge Processors," *ISSCC*, pp. 496-497, 2018.
[2] A. Biswas, et al., "Conv-RAM: An Energy-Efficient SRAM with Embedded Convolution Computation for Low-Power CNN-Based Machine Learning Applications," *ISSCC*, pp. 488-489, 2018.
[3] S. K. Gonugondia, et al. "A 42 pJ/Decision 3.12TOPS/W Robust In-Memory Machine Learning Classifier with On-Chip Training," *ISSCC*, pp. 490-491, 2018.
[4] K. Ueyoshi, et al. "QUEST: A 7.49 TOPS Multi-Purpose Log-Quantized DNN Inference Engine Stacked on 96 MB 3D SRAM Using Inductive-Coupling Technology in 40nm CMOS," *ISSCC*, pp. 216-217, 2018.
[5] J. Zhang, et al., "In-Memory Computation of a Machine-Learning Classifier in a Standard 6T SRAM Array", *JSSC*, pp. 915-924, 2017.

978-1-5386-8531-0/19/$31.00 ©2019 IEEE



	This work	REF[2]	REF[3]	REF[1]
Structure	Twin 8T (T8T) Array (operation: x /+, Multi-bit) C2PU ... C2PU YOUT	Local 10T Array (operation: x, binary) Local Analog Averaging (operation: +) DAC ... DAC ADC	6T Array (operation: mulit-bit weight combination) Near memory computing (operation: x /+) ADC	DSC 6T Array (operation: x /+, binary) SA ... SA
Model	Multiple-bit NN	Binary weight NN	Support Vector Machine	Fully connected layer
SRAM Cell	Twin 8T(T8T)	10T	Typical 6T	DSC6T
IMC with x/+ in array	Yes	No (binary multiplication)	No	Yes (binary)
Support CNN	Yes	Yes	No	No
Write disturb	No	No	Yes	Assist
Weight	2's complement	Binary (+1/-1)	1's complement	Binary (+1/-1)

Figure 24.5.1: Overview of previous SRAM-CIM and this work.

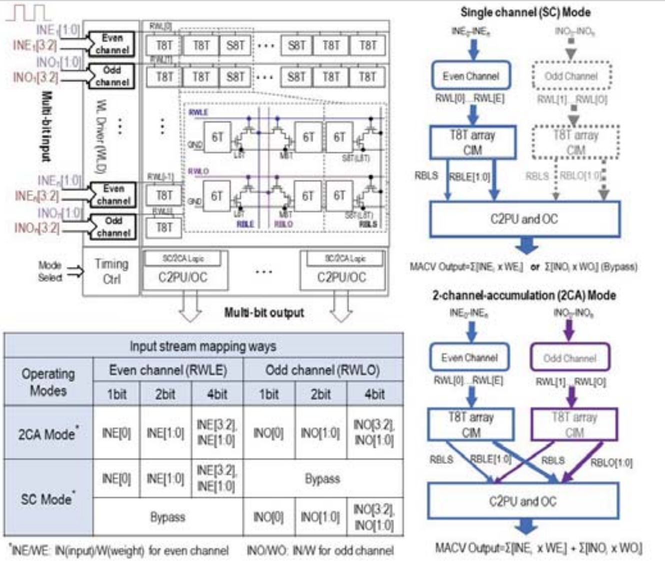


RBLS=Σ[RWL x Q2], for sign bit

RBLE/RBLO=Σ[RWL x (2Q1+Q2)]

Input (2b)	RWL	Weight		RBL	IWP
		M8T	L8T		
11	WLL3	1	1	3 x I_{UC}	9
10	WLL2	1	1	2 x I_{UC}	6
01	WLL1	1	1	1 x I_{UC}	3
11	WLL3	1	0	2 x I_{UC}	6
10	WLL2	1	0	4/3 x I_{UC}	4
01	WLL1	1	0	2/3 x I_{UC}	2
11	WLL3	0	1	I_{UC}	3
10	WLL2	0	1	2/3 x I_{UC}	2
01	WLL1	0	1	1/3 x I_{UC}	1
00	VSS	1/0	1/0	0	0

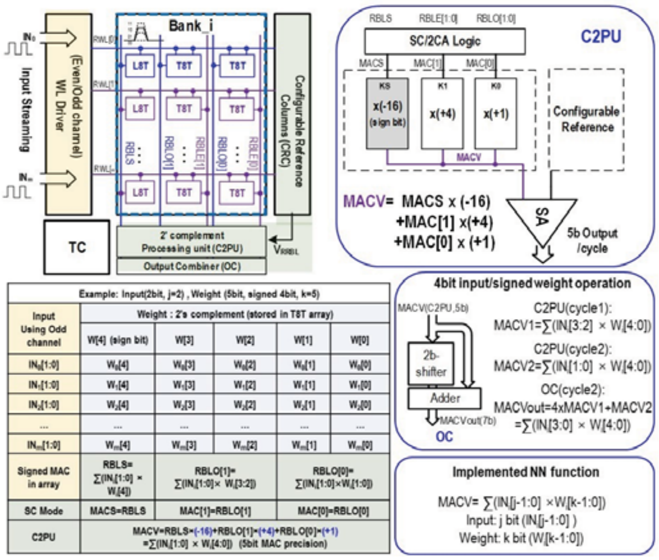
Figure 24.5.2: Macro level structure and twin 8T (T8T) SRAM cell.



Operating Modes	Even channel (RWLE)			Odd channel (RWLO)		
	1bit	2bit	4bit	1bit	2bit	4bit
2CA Mode*	INE[0]	INE[1:0]	INE[3:2], INE[1:0]	INO[0]	INO[1:0]	INO[3:2], INO[1:0]
SC Mode*	INE[0]	INE[1:0]	INE[3:2], INE[1:0]	Bypass		
	Bypass			INO[0]	INO[1:0]	INO[3:2], INO[1:0]

*INE/WE: IN(input)/W(weight) for even channel INO/WO: IN/W for odd channel

Figure 24.5.3: Even odd dual channel (EODC) scheme for multi-bit input stream.



MACV= MACS x (-16) +MAC[1] x(+4) +MAC[0] x (+1)

Example: Input(2bit, j=2), Weight (5bit, signed 4bit, k=5)

Input Using Odd channel	Weight : 2's complement (stored in T8T array)				
	W[4] (sign bit)	W[3]	W[2]	W[1]	W[0]
IN_0[1:0]	W_0[4]	W_0[3]	W_0[2]	W_0[1]	W_0[0]
IN_1[1:0]	W_1[4]	W_1[3]	W_1[2]	W_1[1]	W_1[0]
IN_2[1:0]	W_2[4]	W_2[3]	W_2[2]	W_2[1]	W_2[0]
...	...	...	...	...	...
IN_m[1:0]	W_m[4]	W_m[3]	W_m[2]	W_m[1]	W_m[0]
Signed MAC in array	RBLS= Σ(IN_i[1:0] × W_i[4])	RBLO[1]= Σ(IN_i[1:0]× W_i[3:2])		RBLO[0]= Σ(IN_i[1:0]×W_i[1:0])	
SC Mode	MACS=RBLS	MAC[1]=RBLO[1]		MAC[0]=RBLO[0]	
C2PU	MACV=RBLS×(-16)+RBLO[1]×(+4)+RBLO[0]×(+1) =Σ(IN_i[1:0] × W_i[4:0]) (5bit MAC precision)				

4bit input/signed weight operation

C2PU(cycle1): MACV1=Σ(IN_i[3:2] × W_i[4:0])

C2PU(cycle2): MACV2=Σ(IN_i[1:0] × W_i[4:0])

OC(cycle2): MACVout=4xMACV1+MACV2 =Σ(IN_i[3:0] × W_i[4:0])

Implemented NN function

MACV= Σ(IN_i[j-1:0] ×W_i[k-1:0])

Input: j bit (IN_i[j-1:0])

Weight: k bit (W_i[k-1:0])

Figure 24.5.4: Two's complement weight mapping and signed MACV.

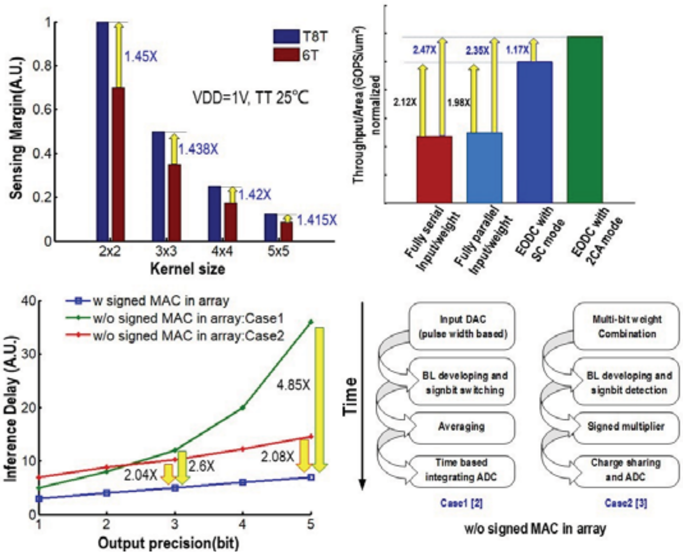


Figure 24.5.5: Simulated performance comparison.

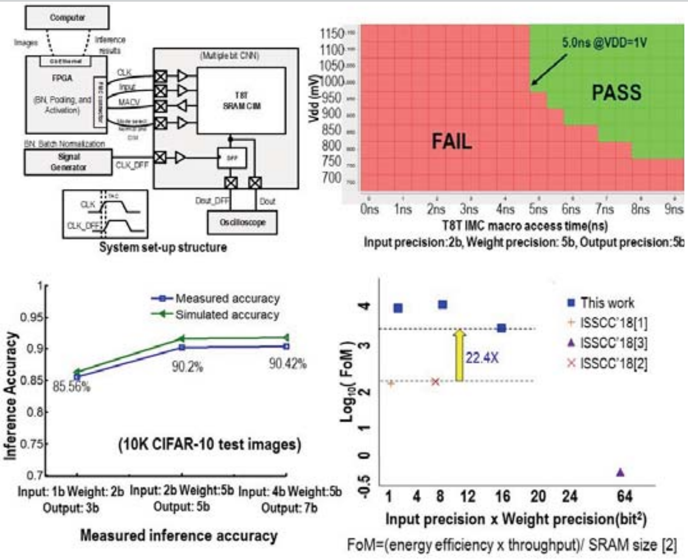
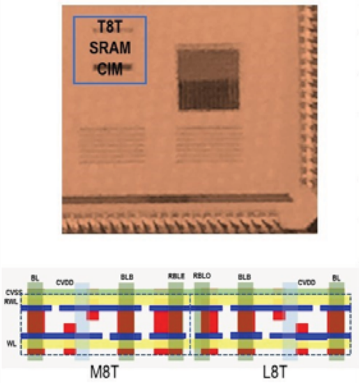


Figure 24.5.6: Measured results.



CHIP SUMMARY			
Technology	55nm CMOS		
Unit-macro Size	64X60b		

Bit cell size
(In 65nm before shrinking to 55nm)
	M8T(0.5um x 1.73um)
	L8T(0.5um x 1.59um)		

Input precision(bit)	1	2	4
Weight precision(bit)	2	5	5
Output precision(bit)	3	5	7
Inference time(ns)	3.2	5.0	10.2
Energy(pJ)	2.5	4.8	9.8
Energy efficiency (TOPS/W)	72.1	37.5	18.37
Measured accuracy MNIST	99.02%	99.18%	99.52%
Measured accuracy CIFAR10	85.56%	90.2%	90.42%

Figure 24.5.7: Die photo and summary table.