

Title:	A 42pJ/Decision 3.12TOPS/W Robust In-Memory Machine Learning Classifier with On-Chip Training
Archived version	Accepted manuscript: the content is similar to the published paper, but without the final typesetting by the publisher
Published version DOI	10.1109/ISSCC.2018.8310398
Published paper URL	http://ieeexplore.ieee.org/abstract/document/8310398/
Authors (contact)	Sujan K. Gonugondla (gonugon2@illinois.edu) Mingu Kang (mkang17@illinois.edu) Naresh R. Shanbhag (shanbhag@illinois.edu)
Affiliation	University of Illinois at Urbana Champaign

Article begins on next page

Sujan Gonugondla, Mingu Kang, Naresh Shanbhag

University of Illinois at Urbana-Champaign

Embedded sensory systems (Fig. 31.2.1) continuously acquire and process data for inference and decision-making purposes under stringent energy constraints. These always-ON systems need to track changing data statistics and environmental conditions, such as temperature, with minimal energy consumption. Digital inference architectures [1,2] are not well-suited for such energy-constrained sensory systems due to their high energy consumption, which is dominated (>75%) by the energy cost of memory read accesses and digital computations. In-memory architectures [3,4] significantly reduce the energy cost by embedding pitch-matched analog computations in the periphery of the SRAM bitcell array (BCA). However, their analog nature combined with stringent area constraints makes these architectures susceptible to process, voltage, and temperature (PVT) variation. Previously, off-chip training [4] has been shown to be effective in compensating for PVT variations of in-memory architectures. However, PVT variations are die-specific and data statistics in always-ON sensory systems can change over time. Thus, on-chip training is critical to address both sources of variation and to enable the design of energy efficient always-ON sensory systems based on in-memory architectures. The stochastic gradient descent (SGD) algorithm is widely used to train machine learning algorithms such as support vector machines (SVMs), deep neural networks (DNNs) and others. This paper demonstrates the use of on-chip SGD-based training to compensate for PVT and data statistics variation to design a robust in-memory SVM classifier

Figure 31.2.2 shows the system architecture with an analog in-memory (IMCORE) block, a digital trainer, a control block (CTRL) for timing and mode selection, and normal SRAM R/W interface. The system can operate in 3 modes: conventional SRAM mode, in-memory inference mode, and training mode. IMCORE comprises a conventional 512×256 6T SRAM BCA and in-memory computation circuitry: 1) pulse width modulated (PWM) word-line (WL) drivers to realize functional read (FR), 2) bit-line processors (BLPs) implementing signed multiplication, 3) cross BLP (CBLP) implementing summation, and 4) A/D converter and a comparator bank to generate final decisions. While the IMCORE implements the feedforward computations of SVM algorithm, the trainer implements batch mode SGD algorithm (update equations in Fig. 31.2.2) to train the SVM weights \mathbf{W} stored in the BCA. The input vectors \mathbf{X} are streamed into the *input buffers* in the trainer. Gradient estimate Δ is accumulated for each input based on the label y_n and outputs $\delta_{1,n}$ and $\delta_{-1,n}$ of IMCORE. At the end of each batch, the accumulated gradient estimate Δ is used to update the weights in BCA via the normal R/W interface. While 16b weights are used in the trainer during the weight update, feedforward/inference only use 8b weights. The learning rate γ and regularization factor α can be reconfigured in the powers of 2.

During the feedforward computations, \mathbf{W} is read in analog domain on the bit-lines (BLs) and the input vectors \mathbf{X} transferred to the BLP via a 256b bus. The mixed-signal capacitive multiplier in the BLP realizes multiplication via sequential charge sharing similar to the one introduced in [3]. Based on their sign, the multiplier outputs are charge shared either on the positive or on the negative CBLP rails across the BLs. The voltage difference of the negative and positive rails is proportional to the dot product $\mathbf{W}^T \mathbf{X}$. The rail values are either sampled and converted to a digital value by an ADC pair or a decision is obtained directly via a comparator bank. Three comparators are used where one generates the decision \hat{y} while the other two comparators implement a SVM margin detector which triggers a gradient estimate update.

Functional read (Fig. 31.2.3) uses 4-parallel pulse-width and amplitude-modulated (PWAM) WL enable signals resulting in the BL discharge ΔV_{BL} (or ΔV_{BLB}) proportional to W_i s stored in column-major format (Fig 31.2.3), in one precharge cycle. The BL discharges (ΔV_{BL}) of 4b words read in the adjacent BLs are combined in 1:16 ratio to realize an 8b read out. This enables 8b 128 dimensional vector processing per access. The weights are represented in 2's complement. A comparator detects the sign of W_i which is then used to select its magnitude, both of which are passed on to the signed multipliers. Spatial variations impacting ΔV_{BL} is measured across 30 randomly chosen 4-row groups. When the maximum ΔV_{BL} ($\Delta V_{BL,max}$), corresponding to $W_i = 15$, is set to 320mV, the maximum variation in ΔV_{BL} ($(\sigma/\mu)_{max}$) across all 16 values, is found to be 16% vs. 7% at $\Delta V_{BL,max} = 560$ mV. This increase in the impact of variation leads to an increase in the misclassification rate: from 4% to 18%.

The MIT CBCL face detection data is used to test the IC. The dataset has 4000 training images and 858 test images. During training, input vectors are randomly sampled with replacement from the training set. At the end of each batch, the classifier is tested on test set to obtain the misclassification (error) rate. Figure 31.2.4 shows the benefits of on-chip learning in overcoming process and data variations, and the need for learning chip-specific weights. Beginning with random initial weights and $\Delta V_{BL,max} = 560$ mV, the learning curves converge to within 1% to floating point accuracy in 400 batch updates for learning rates $\gamma \geq 2^{-4}$. The misclassification rate increases dramatically to 18% when $\Delta V_{BL,max}$ is reduced to 320mV at batch number 400 due to increased impact of process variations during FR. Continued on-chip learning reduces this misclassification rate down to 8% for $\gamma \geq 2^{-4}$. Similar results are observed when illumination changes abruptly at batch number 400 indicating robustness to variations in data statistics. The table in Fig. 31.2.4 shows the misclassification rate measured across 5 chips when the weights are trained on one chip and used in others.

The use of chip-specific weights (diagonal) results in an average misclassification rate of 8.4% vs. 43% when it is not, indicating the need for on-chip learning.

Figure 31.2.5 shows the trade-off between the misclassification rate, IMCORE energy, and $\Delta V_{BL,max}$. On-chip training enables the IC to achieve an misclassification rate to $< 8\%$ at a 38% lower $\Delta V_{BL,max} = 320\text{mV}$ and a lower IMCORE supply $V_{dd,IMCORE} = 0.675\text{V}$, compared to the use of weights obtained at $\Delta V_{BL,max} = 560\text{mV}$ and $V_{dd,IMCORE} = 0.925\text{V}$. Thus, the IMCORE energy is reduced by 2.4 \times without any loss in accuracy. The energy cost of the training is dominated by the normal SRAM writes of updated weights done once per batch. This cost reduces with batch size N reaching 26% of the total energy cost, for a batch size of 128. At this batch size, 60% of the total energy is due to CTRL. This CTRL energy overhead will reduce with increase in SRAM size.

Figure 31.2.6 shows that the prototype IC achieves a system IMCORE energy efficiency of 42pJ/decision at a throughput of 32M decisions/s which corresponds to a computational energy efficiency of 3.12TOPS/W (1 OP = 8b \times 8b MAC). This work achieves the lowest reported precision-scaled MAC energy as well as the lowest reported MAC energy when SRAM memory access costs are included. Energy consumption of digital architectures [1,2] to realize the 128 dimensional SVM algorithm of this work is estimated from their MAC energy, which shows a savings of $>7\times$ thereby demonstrating the suitability of this work for energy-constrained sensory applications.

The die micrograph of the 65nm CMOS IC and performance summary is shown in Fig. 31.2.7.

Acknowledgements:

This work was supported in part by Systems On Nanoscale Information fabriCs (SONIC), one of the six SRC STARnet Centers, sponsored by MARCO and DARPA. The authors would like to acknowledge constructive discussions with Professors Pavan Hanumolu, Naveen Verma, Boris Murmann, and David Blaauw.

References:

- [1] Y.H. Chen, et al., "Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," IEEE ISSCC, pp. 262-263, Feb 2016.
- [2] P.N. Whatmough, et al., "A 28nm SoC with a 1.2GHz 568nJ/prediction sparse deep-neural-network engine with >0.1 timing error rate tolerance for IoT applications," IEEE ISSCC, pp. 242-243, Feb 2017.
- [3] M. Kang, et al. "A 481pJ/decision 3.4 M decision/s multifunctional deep in-memory inference processor using standard 6T SRAM array," arXiv preprint arXiv : 1610.07501, 2016.
- [4] J. Zhang, et al., "In-memory computation of a machine learning classifier in a standard 6T SRAM array," IEEE JSSC, vol. 52, no. 4, pp. 915-924, April 2017.
- [5] E.H. Lee, et al., "A 2.5GHz 7.7TOPS/W switched-capacitor matrix multiplier with co-designed local memory in 40nm," IEEE ISSCC, pp. 418-419, Feb 2016.
- [6] S. Joshi, et al., "2pJ/MAC 14b 8×8 linear transform mixed-signal spatial filter in 65nm CMOS with 84dB interference suppression," IEEE ISSCC, pp. 364-365, Feb 2017.

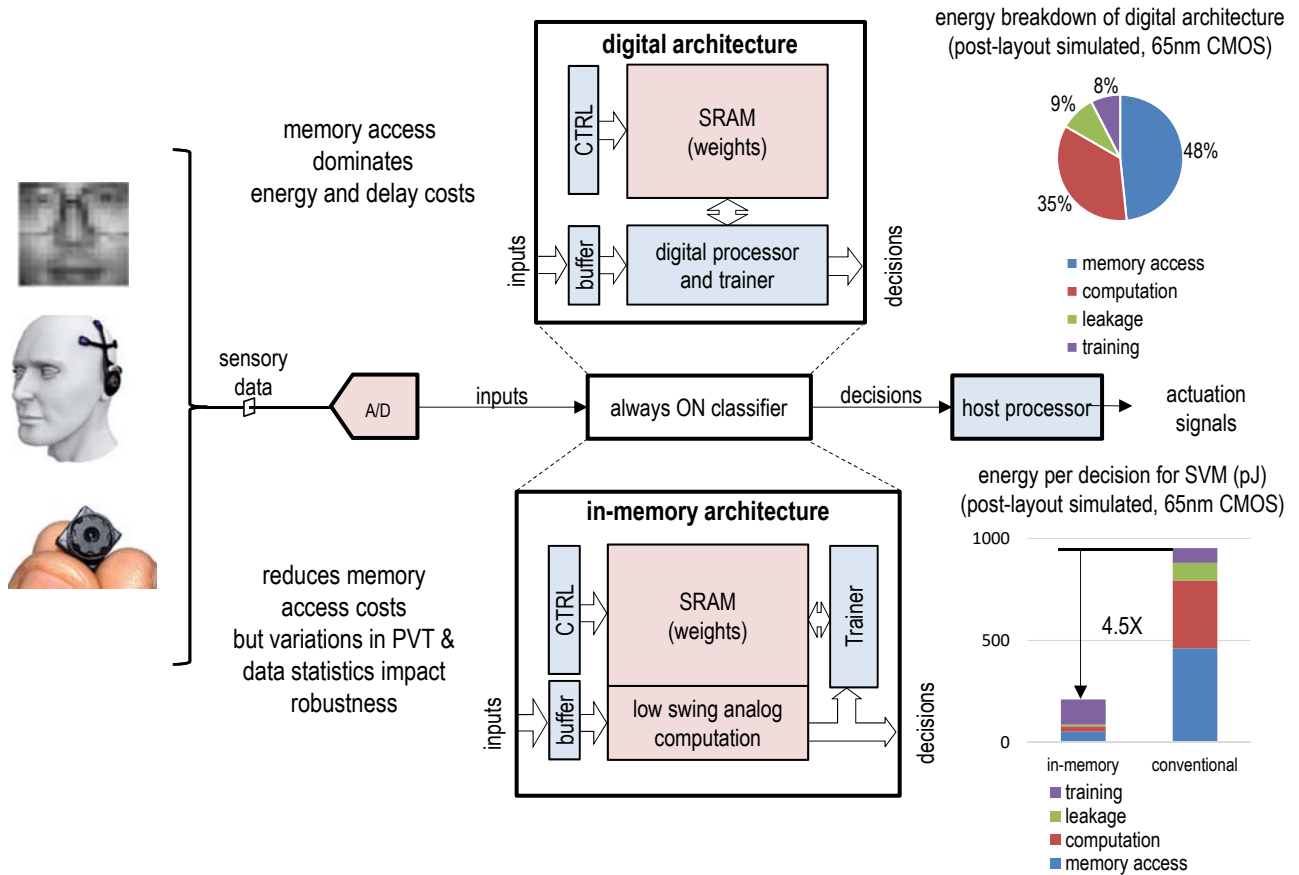
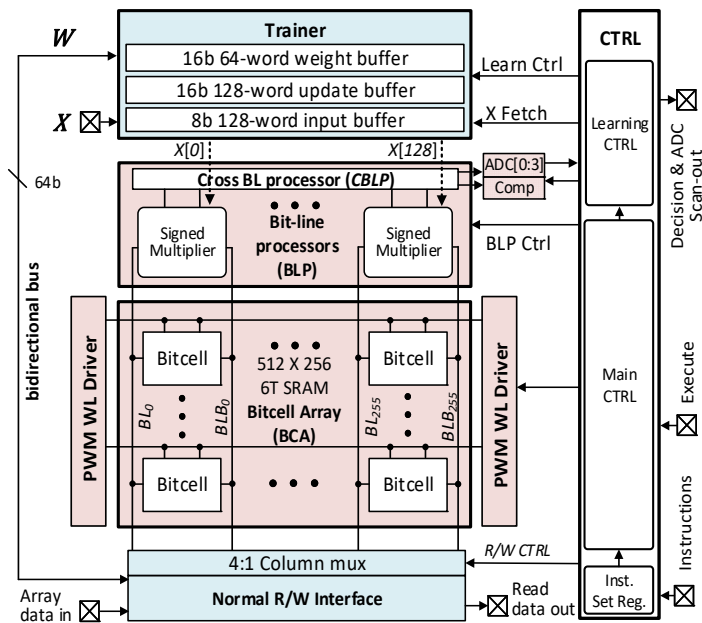


Figure 31.2.1: An SGD-based on-chip learning system for robust energy efficient always-ON classifiers.



support vector machine (SVM) computations

IMCORE (analog)

$$z_n = W_m^T X_n + b$$

functional read & BLP

$$\left. \begin{aligned} \delta_{1,n} &= \text{sign}(z_n - 1) \\ \delta_{-1,n} &= \text{sign}(z_n + 1) \end{aligned} \right\}$$

Comparator-based SVM margin detector

$$\hat{y}_n = \text{sign}(z_n)$$

Comparator

Trainer & normal R/W (digital)

$$\Delta = \sum_1^N \begin{cases} y_n X_n & y_n \delta_{1,n} \leq 0 \\ 0 & \text{or } y_n \delta_{-1,n} \leq 0 \\ 0 & \text{otherwise} \end{cases}$$

gradient estimate

$$W_{m+1} = (1 - \alpha)W_m + \frac{\gamma \Delta}{N}$$

weight update

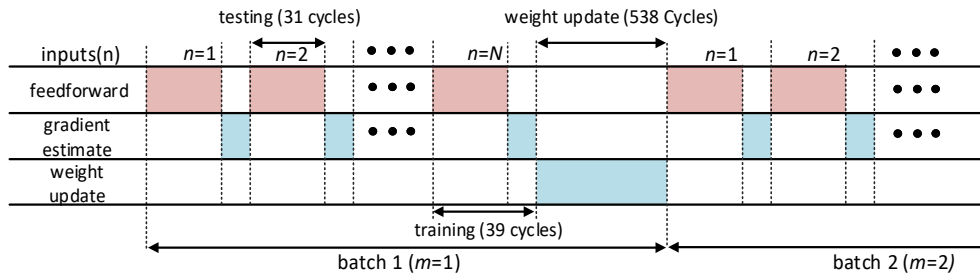
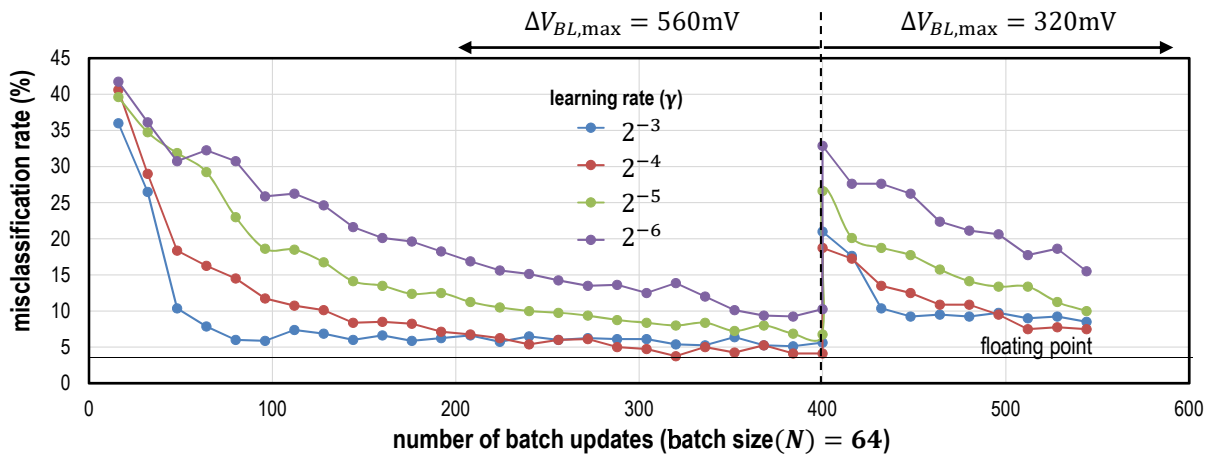


Figure 31.2.2: Proposed SGD-based in-memory classifier architecture.



misclassification rate (%)
($\Delta V_{BL,max} = 320mV$)

		tested on				
		Chip1	Chip2	Chip3	Chip4	Chip5
trained on	Chip1	8.25	38.3	48.3	51.5	48.8
	Chip2	45.8	9	48	49.8	34.5
	Chip3	47	51.3	8.5	29.8	49.3
	Chip4	51.5	51	17.5	8.25	51.3
	Chip5	38.3	18	48.5	48.5	8

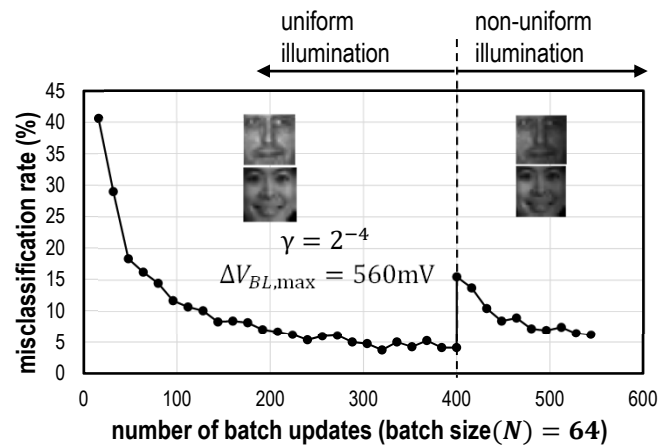


Figure 31.2.4: Measured robustness to spatial variations and non-stationary data.

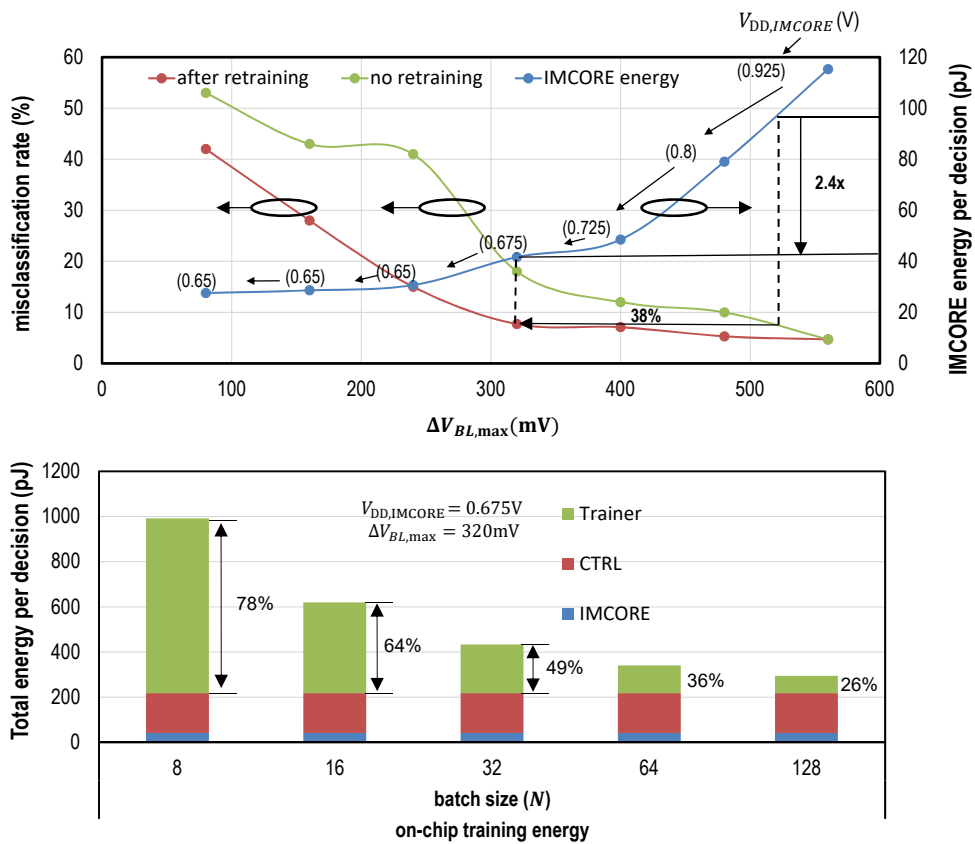
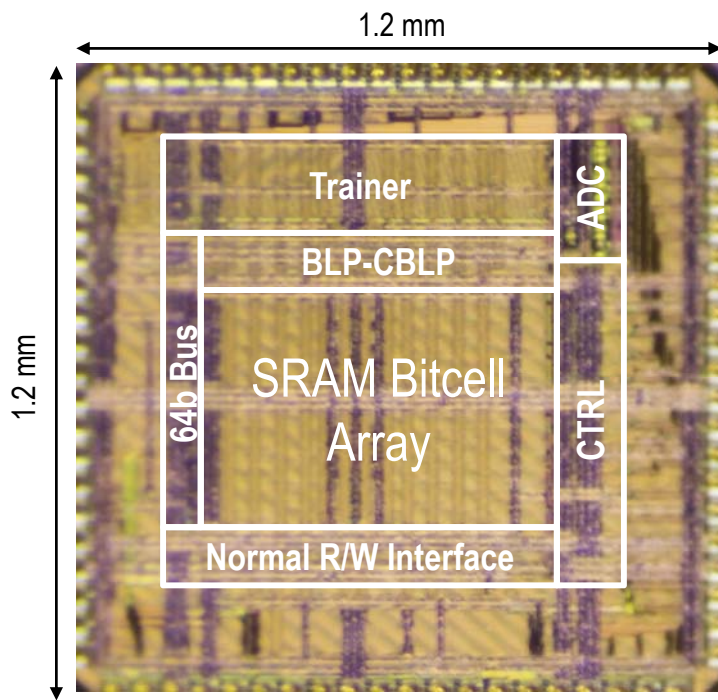


Figure 31.2.5: Measured energy via supply voltage and BL swing scaling, and energy cost of training.

	[1]	[2]	[5]	[6]	[3]	[4]	this work
Technology	65nm	28nm HPC	40nm	65nm	65nm	180nm	65nm
Algorithm	CNN	FC-DNN	matrix mult.	filtering	SVM	AdaBoost	SVM
Data set	ImageNet	MNIST			MIT-CBCL	MNIST	MIT-CBCL
Architecture	digital	digital	analog	analog	in-memory	in-memory	in-memory
On-chip learning	No	No	No	No	No	No	Yes
Total SRAM size (kb)	1449.2	9248	—	—	128	103.6	128
Energy/Decision	7.94mJ ^d	0.56uJ	—	—	0.4nJ	0.6nJ	0.042nJ
Decisions/s	35	28.8k ^d	—	—	9.2M	7.9M	32M
# of MACs/Decision	2663M	334k	—	—	512	—	128
Max. accuracy (%)	—	98	—	—	96	91	96
MAC level metrics							
MAC precision ^a ($B_x \times B_w$)	16 ^s ×16 ^s	8 ^s ×8 ^s	3 ^s ×6 ^s	8×14 ^s	8×8	5×1	8×8 ^s
Efficiency (TOPS/W)	0.336 ^d	0.56 ^d	3.84 ^b	0.5 ^b	1.25	—	3.125
MAC energy (E_{MAC}) (pJ)	2.98 ^d	1.79 ^d	0.26 ^b	2 ^b	0.8	—	0.32
precision-scaled MAC energy ^c (fJ)	11.6	28	14.4 ^b	17.857 ^b	12.5	—	4.9
Estimated performance to realize SVM algorithm with vector dimension of 128							
Energy/Decision (nJ)	0.381	0.229	0.033 ^b	0.256 ^b	0.102	—	0.042
Decisions/s	250M	75M	19.5M	350k	36.8M	—	32M
# MACs per cycle	168	8	1	64	256	10,368	128
^a s indicates signed; B_x : input precision; B_w : weight precision				^c normalized to account for operand precision ($E_{MAC}/(B_x \times B_w)$)			
^b does not include SRAM memory access				^d estimated from reported data			

Figure 31.2.6: Comparison table.



Technology	65nm CMOS	
Die size	1.2 mm × 1.2mm	
Memory capacity	16KB (512 × 256)	
Nominal Supply	1.0 V	
CTRL operating frequency	1 GHz	
Energy per decision (nJ)	Test	0.21
	Training	0.34
Average throughput (decision/s)	Test	32.3 M
	Training	21 M

Figure 31.2.7: Die micrograph and chip summary.