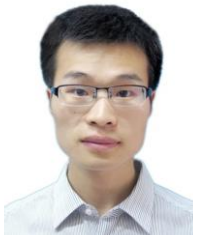
2020 IEEE International Symposium on Circuits and Systems
Virtual, October 10-21, 2020



CRAM: Collocated SRAM and DRAM with In-Memory Computing Based Denoising and Filling for Neuromorphic Vision Sensors in 65 nm CMOS

Xueyong Zhang, Vivek Mohan, Arindam Basu

Nanyang Technological University

2020 IEEE International Symposium on Circuits and Systems
Virtual, October 10-21, 2020

Outline

- Introduction
- CRAM structure
    - Memory mode
    - In-memory computing (IMC) mode
- Results and discussion
- Conclusion
- Future Work

Outline

- **Introduction**
- **CRAM structure**
    - **Memory mode**
    - **In-memory computing (IMC) mode**
- **Results and discussion**
- **Conclusion**
- **Future Work**

Introduction: Computer Vision

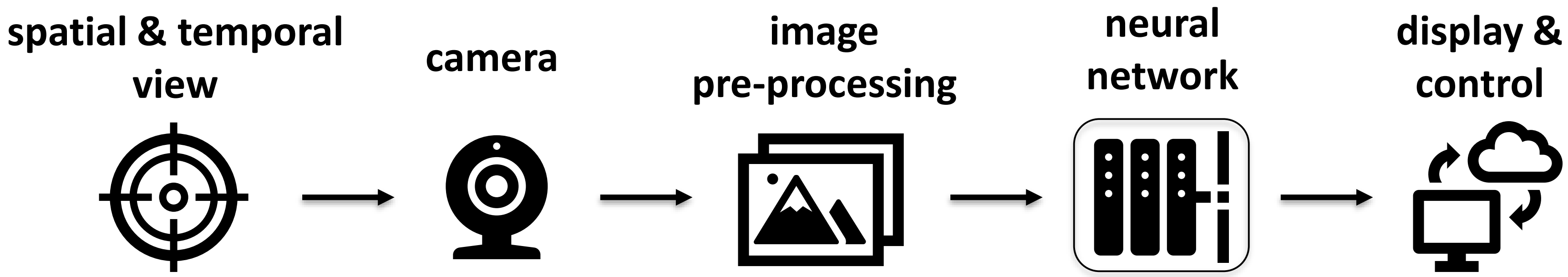
Using AI/ML in computer vision is a trend

Smart car
Autopilot

Smart UVA
Object Tracking

Smart camera
Active Surveillance

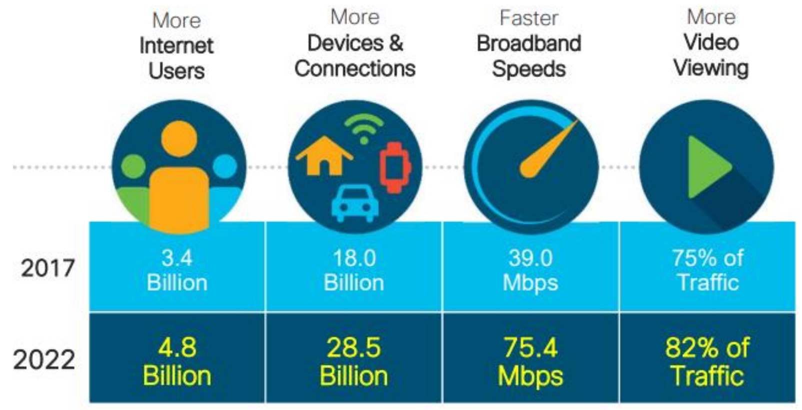
The system of a typical computer vision application



Introduction: Computer Vision

Trends:

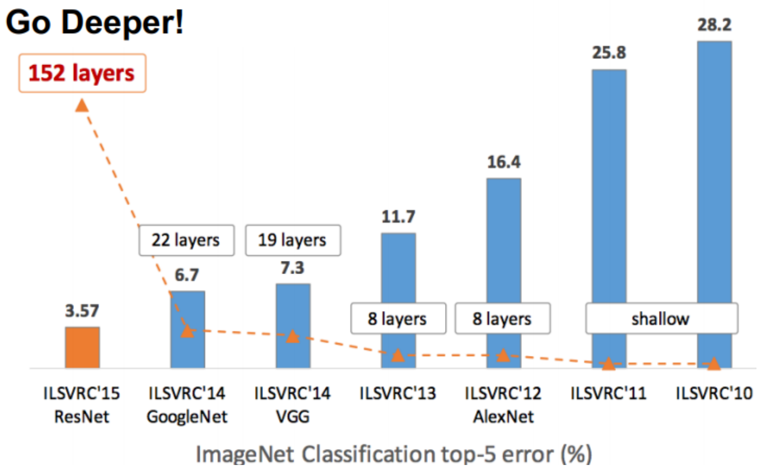
(1) Bigger Data



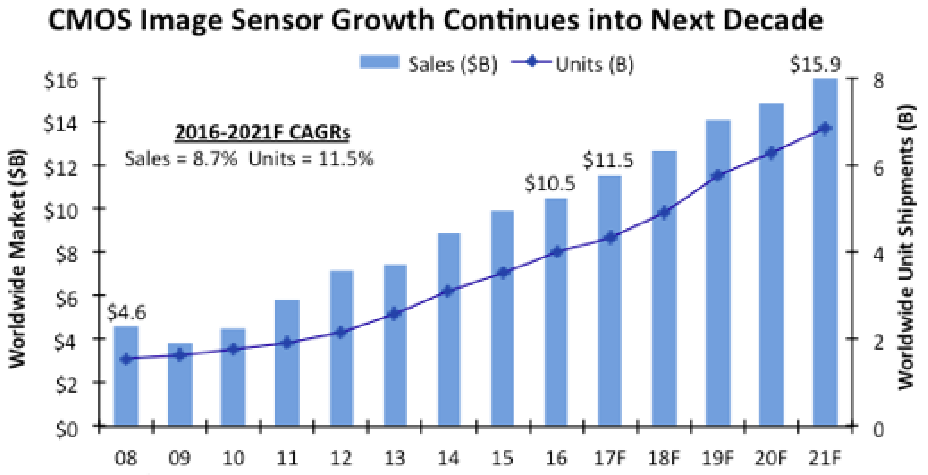
	More Internet Users	More Devices & Connections	Faster Broadband Speeds	More Video Viewing
2017	3.4 Billion	18.0 Billion	39.0 Mbps	75% of Traffic
2022	4.8 Billion	28.5 Billion	75.4 Mbps	82% of Traffic

Source: Cisco VNI Global IP Traffic Forecast, 2017-2022

(2) Larger Model

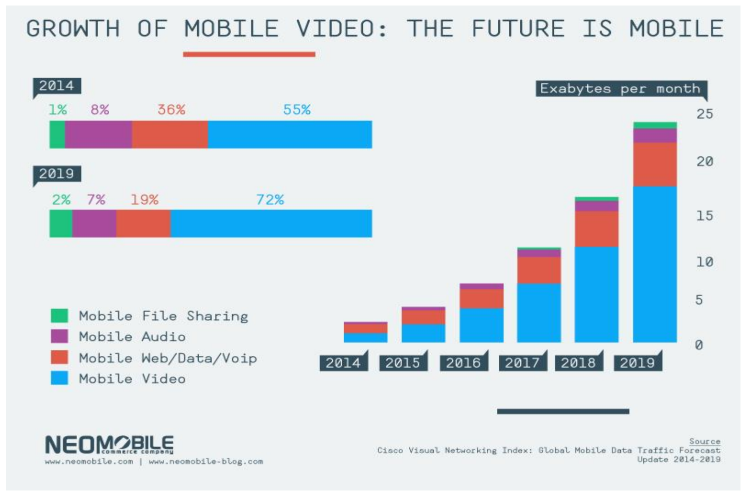


(3) More Sensors (IoT)



Source: IC Insights

(4) More Mobiles



Introduction: Computer Vision

Challenges:

(1) Bigger Data

(2) Larger Model

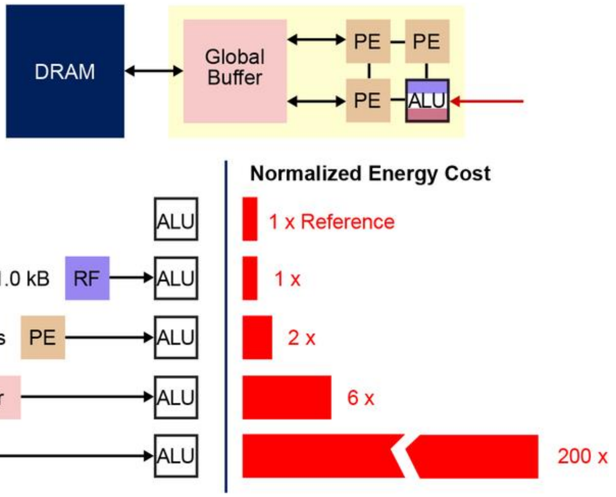
→ Massive computation & long time

(3) More Sensors (IoT)

(4) More Mobiles

→ Low energy

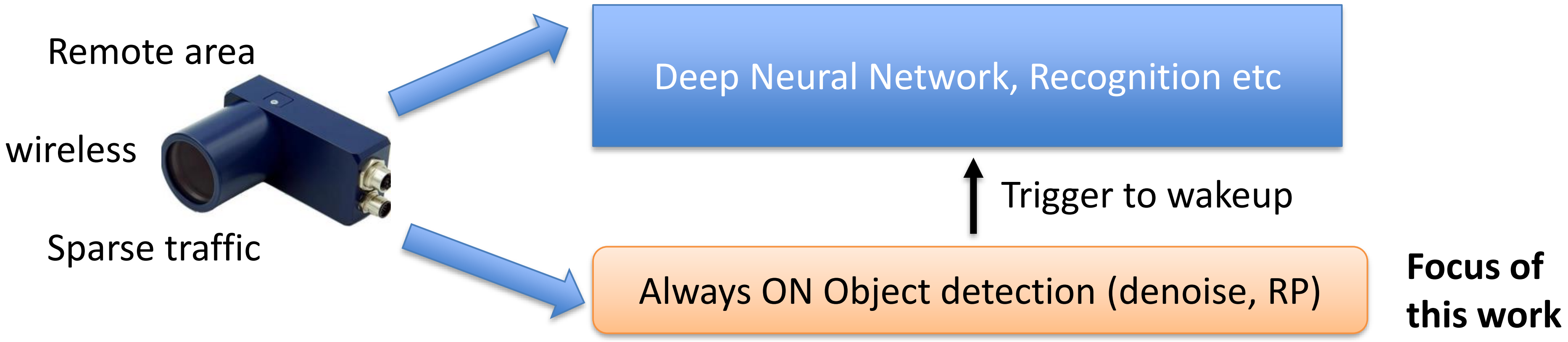
(5) Real-time interaction → Low latency



Goal of this work

Develop novel low-energy solutions for **Internet of Video Things** (IoVT) application like traffic monitoring by innovations at the sensor and processor.

- **Sensor: Event driven**
  - ✓ Event driven neuromorphic sensor → Advantage of reduced data but processing may be problem since data format is different
- **Processor: Processing Circuit + Memory**
  - ✓ Memory access is dominant for image processing & neural networks → In-memory computing as a solution


Remote area
wireless
Sparse traffic
Deep Neural Network, Recognition etc
Trigger to wakeup
Always ON Object detection (denoise, RP)
Focus of this work

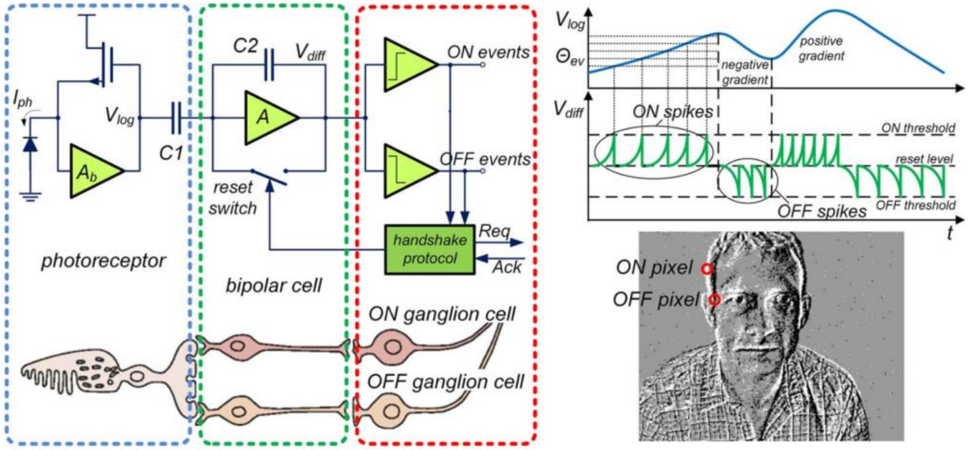
Goal of this work

Solutions:

1. Data
Simplify data by using <u>event based</u> sensor.
Only dynamic pixels are recorded

Asynchronous change detecting pixels
—Simulating human retina!!

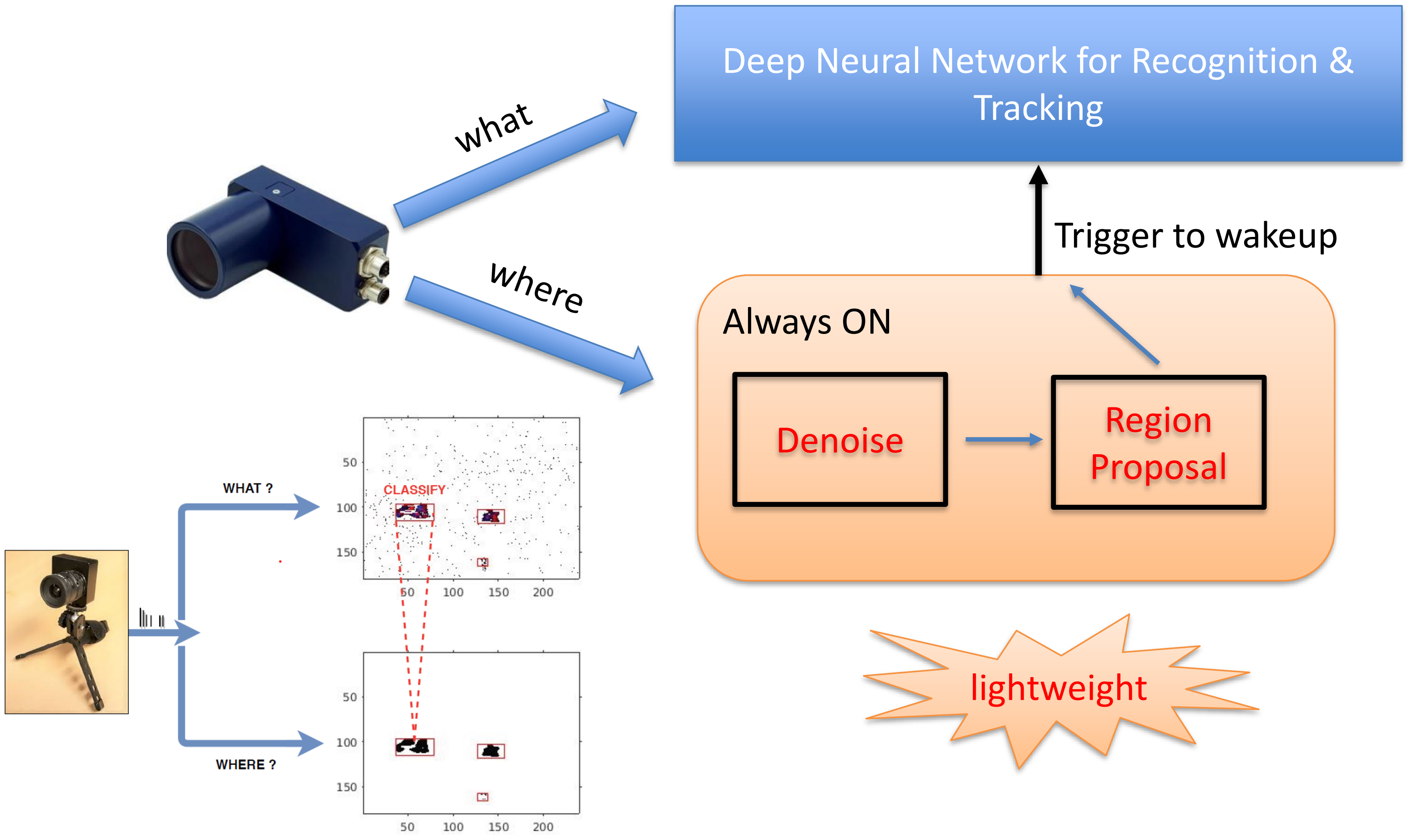
- High dynamic range
- Data reduction
- Fast reaction



2. Computing
By reducing or even eliminating the data movement to improve energy efficiency

Digital Computing	Near-Memory Computing	In-Memory Computing
☺ Flexibility ☺ Full digital accuracy	☺ Higher on-chip BW ☺ Localized computation	☺ Highest on-chip BW ☺ Better energy efficiency
☹ Limited by on-chip BW ☹ Computation efficiency ☹ Large area and energy overhead	☹ Less flexible ☹ Area efficiency	☹ Least flexible ☹ Computation accuracy

Processing pipeline



D. Singla et al., “HyNNA: Improved Performance for Neuromorphic Vision Sensor….” ISCAS, 2020.

Literature review: Denoise operations

Traditional method

(1) Nearest neighbour (NNb) filtering

We can define a set of indices for past events in the Nearest Neighbourhood, N_{NN} , of event e_i as

$$N_{NN}(e_i) = \{j \mid j < i,\ D_{i,j} <= \sqrt{2}\} \qquad e_i = \{x_i, y_i, p_i, t_i\}, \quad i \in \mathbb{N}$$

$$D_{i,j} = \left\| \begin{pmatrix} x_i \\ y_i \end{pmatrix} - \begin{pmatrix} x_j \\ y_j \end{pmatrix} \right\|$$

The set of events, F_{NN} , passed through the nearest neighbour filter will then be

$$F_{NN} = \{e_i \mid t_i - \max(t_{N_{NN}(e_i)}) < \underbrace{T}_{\text{threshold}}\}$$

(2) Refractory filtering

$$N_{RP}(e_i) = \{j | j < i, D_{i,j} <= 0, e_i \in F_{RP}(e_{i-1})\}$$
$$F_{RP}(i) = \{e_i | t_i - \max(t_{N_{RP}(e_i)}) > \underbrace{T}_{\text{Refractory period}}\}$$

Czech, et al., “Evaluating noise filtering for event-based asynchronous change detection image sensors,” BioRob, 2016.

Literature review: Denoise operations

Recent method – Hybrid event + frame approach
-- Event based binary image (EBBI) + Median filtering
-- Still need to read through entire image ☹

NN filt + Refractory

$$C_{NN-filt} = (2(p^2 - 1) + B_t) \times \overline{n}$$
$$M_{NN-filt} = B_t \times A \times B$$
$$\overline{n} \;=\; \beta$$

EBBI+ Median filt

$$C_{EBBI} \approx (\alpha p^2 + 2) \times A \times B(\because p << A, B)$$
$$M_{EBBI} = 2 \times A \times B$$

$C_{NN-flit}$ (ko	BBI (kb)
691	4.375

ory X8↓

Can we do In-memory Computing (IMC) based EBBI?

p: neighbourhood size for noise filtering (p=3)
α: the fraction of active pixels in a patch on average (α=0.5)
β: average number of times an active pixel fires in frame duration (β=1)
Bt: timestamp bits (Bt=16)
A×B: image resolution (A=240, B=180)

Acharya, J., et al., "EBBIOT: A Low-complexity Tracking Algorithm for Surveillance in IoVT ..". SOCC, 2019

Outline

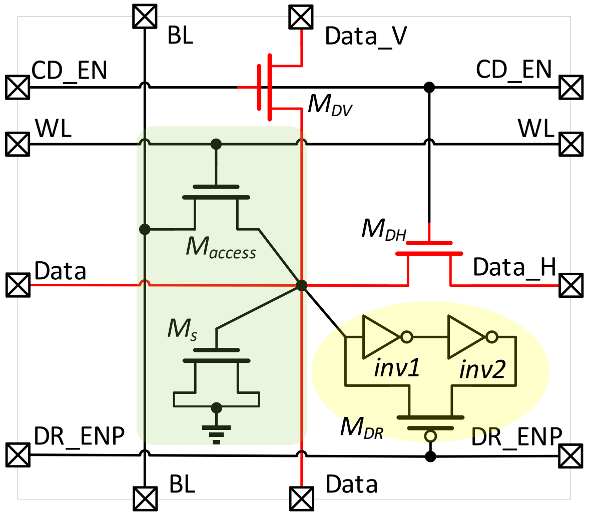
- Introduction
- CRAM structure
  - Memory mode
  - In-memory computing (IMC) mode
- Results and discussion
- Conclusion
- Future Work

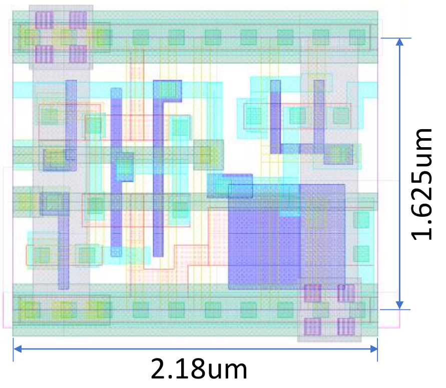
Denoise operation in-memory

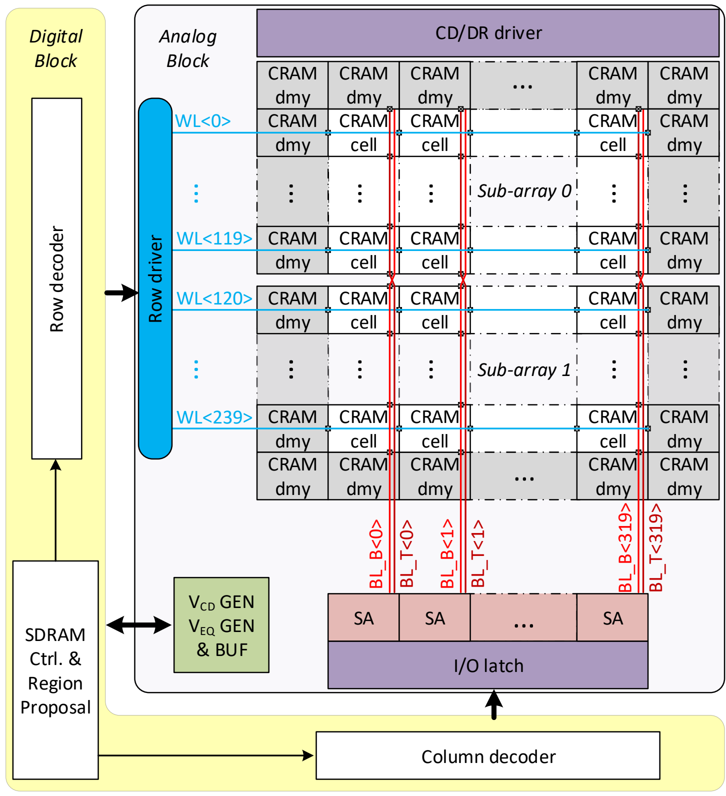
- **We use CRAM to perform image denoising and filling**
- **What is CRAM**
  - **CRAM: Collocated SRAM and DRAM**
  - **We use SRAM-like to latch the storage data**
    - **no need fresh circuit in conventional DRAM**
  - **We use DRAM-like to redistribute/diffuse the stored charge**
    - **by charging/discharging to do averaging all over the array**
    - **Globally parallel computing**
  - **We use SRAM-like to detect and do ADC (based on its threshold voltage)**
    - **to remove noise and recover image data**

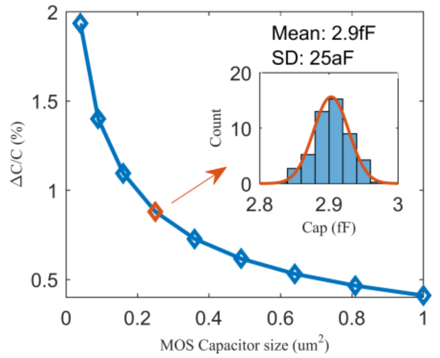
CRAM

- **CRAM Cell & 320x240 Array**
  - **Collocated SRAM and DRAM (9T bit cell)**
  - **Memory mode + IMC mode**









MOS capacitor mismatch with different size

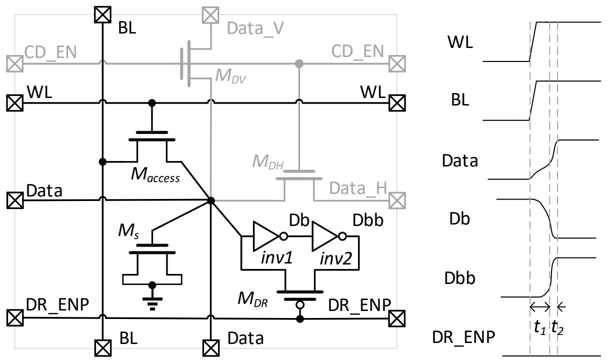
The 9T CRAM cell occupies 1.21× larger area, compared with the 9T bit-cell area in [1].

[1] M.-H. Tu, et al., "A single-ended disturb-free 9T subthreshold SRAM..." IEEE JSSC, 2012

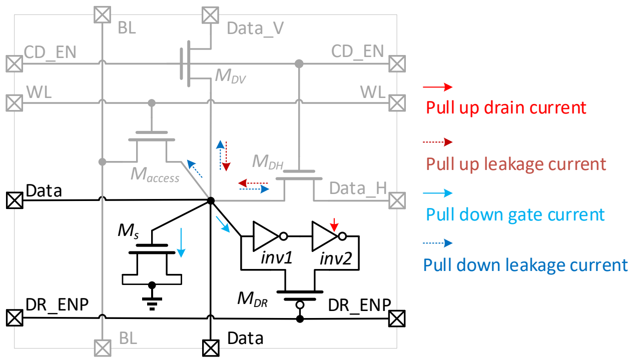
Analog blocks:	**Digital blocks:**
• **CRAM array**	• **Controller**
• **SA**	• **Decoder**
• **Driver**	• **RP algorithm**

CRAM

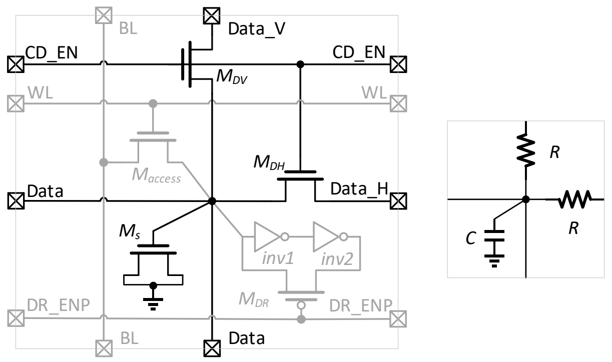
- **CRAM Operations**



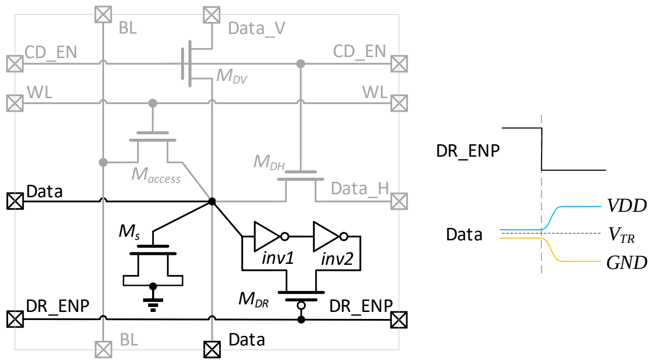
(a) Write mode



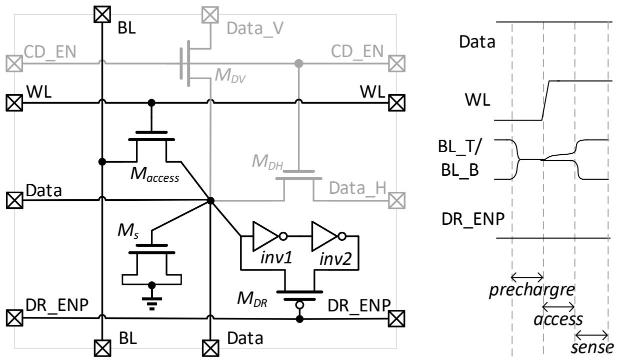
(b) Retention mode



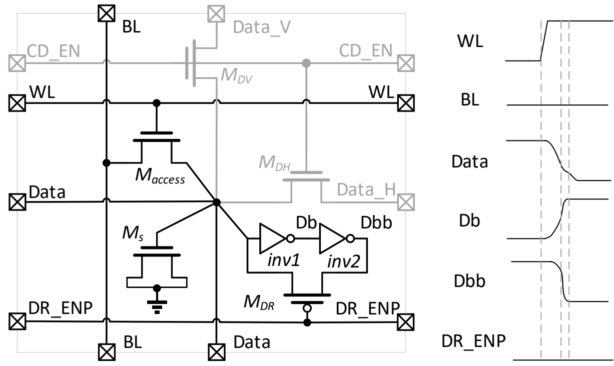
(c) IMC mode



(d) Data recovery mode



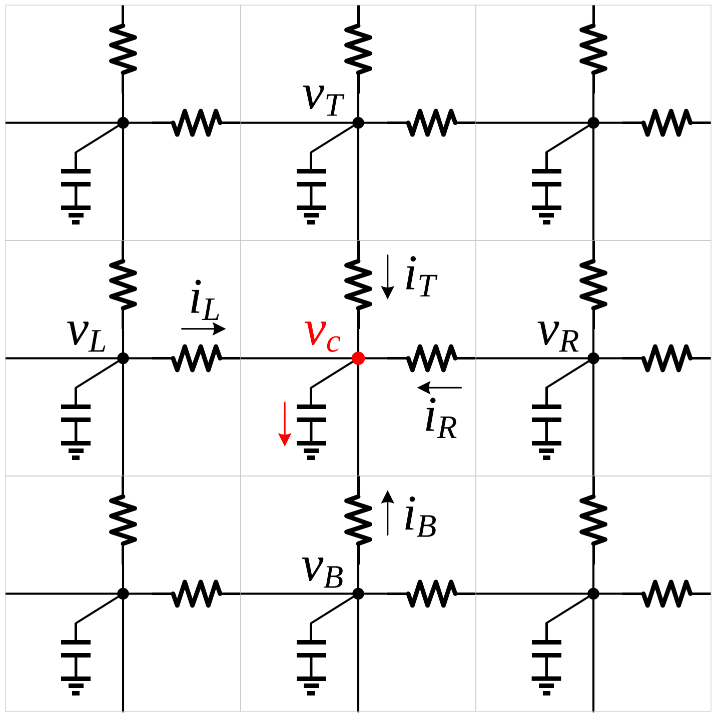
(e) Read mode



(f) Reset mode

CRAM – in memory computing

- **CRAM equivalent 2D RC network**



KCL,

$$\frac{v_T - v_c}{r_T} + \frac{v_B - v_c}{r_B} + \frac{v_L - v_c}{r_L} + \frac{v_R - v_c}{r_R} = C\frac{dv_c}{dt}. \quad (1)$$

Assume: $r_T = r_B = r_L = r_R = R$

$$\frac{dv_c}{dt} + \frac{4}{RC}v_c - \frac{1}{RC}(v_T + v_B + v_L + v_R) = 0. \quad (2)$$

$$v_c(t) = Ae^{-\frac{4t}{RC}} + \frac{v_T + v_B + v_L + v_R}{4}. \quad (3)$$

$$A = v_c(0) - \frac{v_T(0) + v_B(0) + v_L(0) + v_R(0)}{4}. \quad (4)$$

In IMC mode, charges redistribute all over the array
- **Nearest Neighbour Filter (Median Filter)**
- **Edge Smoothing**
- **Filling (Image Recovery)**

☺ **Low power consumption**
Charge diffusion/redistribution happens naturally and consumes no extra energy
☺ **Lower latency**
No data access (R/W) required
Globally parallel computing

CRAM – in memory computing

- **CRAM equivalent 2D RC network**

Charge conservation:

$$Q_{tot} = \sum_{i=1}^{M\times N} C_i(t)\cdot v_i(t) = Constant \qquad (1)$$

$$\sum_{i=1}^{M\times N} v_i(t) = Constant \qquad (2)$$

- ***$v_i(t)$*** **is transient voltage of the *i^{th}* node**
- ***C_i*** **is the capacitor of the *i^{th}* node**
- ***M * N*** **is the size of the array**

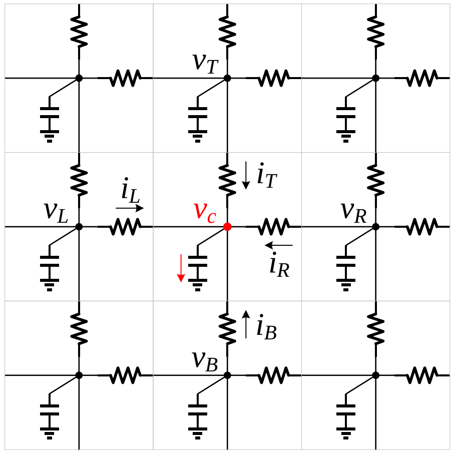
Charge diffusion time:

Denoising: $$t_{diff} > \frac{C(VDD - V_{TR,min})}{4i_{discharge}} \qquad (3\text{-}1)$$

Filling: $$t_{diff} > \frac{CV_{TR,max}}{4i_{charge}} \qquad (3\text{-}2)$$

Erosion: $$t_{diff} < \frac{C(VDD - V_{TR,max})}{i_{discharge}} \qquad (3\text{-}3)$$

Dilation: $$t_{diff} < \frac{CV_{TR,min}}{i_{charge}} \qquad (3\text{-}4)$$



- ***$V_{TR,min}$* is the minimum value of trip voltage *V_{TR}* due to devices mismatch.**
- ***$V_{TR,max}$* is the maximum value of trip voltage *V_{TR}* due to devices mismatch.**
- ***$i_{discharge}$* is the discharge current of the high pixel to one direction**
- ***i_{charge}* is the charge current of the high pixel to one direction**

Outline

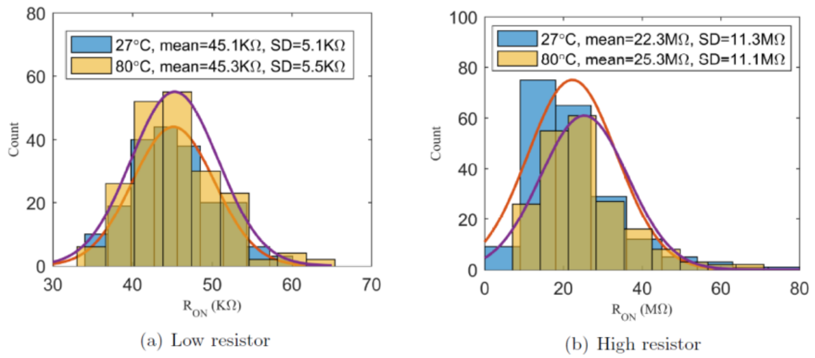
- Introduction
- CRAM structure
  - Memory mode
  - In-memory computing (IMC) mode
- **Results and discussion**
- Conclusion
- Future Work

Results and discussion

- **CRAM performance**

SPICE simulation (65nm CMOS):

(1) Monte Carlo for MOS Res

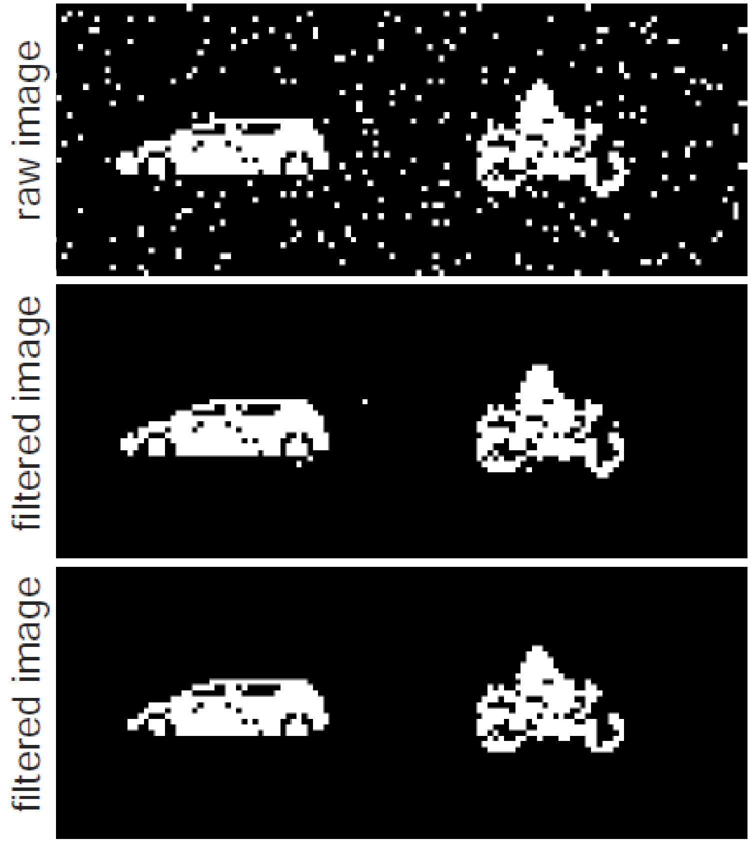


The equivalent Res is programmable by gate voltage.

Comparison with Different Filter Implementations

Type	Process	Area/Cell (um^2)	Latency (ns/bit)	Energy (pJ/bit)
Spatio-temporal Filter [12]	180nm	400	10	20
Median Filter	65nm	4.89	95	228
Proposed IMC-based Analog Natural Filter	65nm	3.54	0.11	0.02/ 0.001

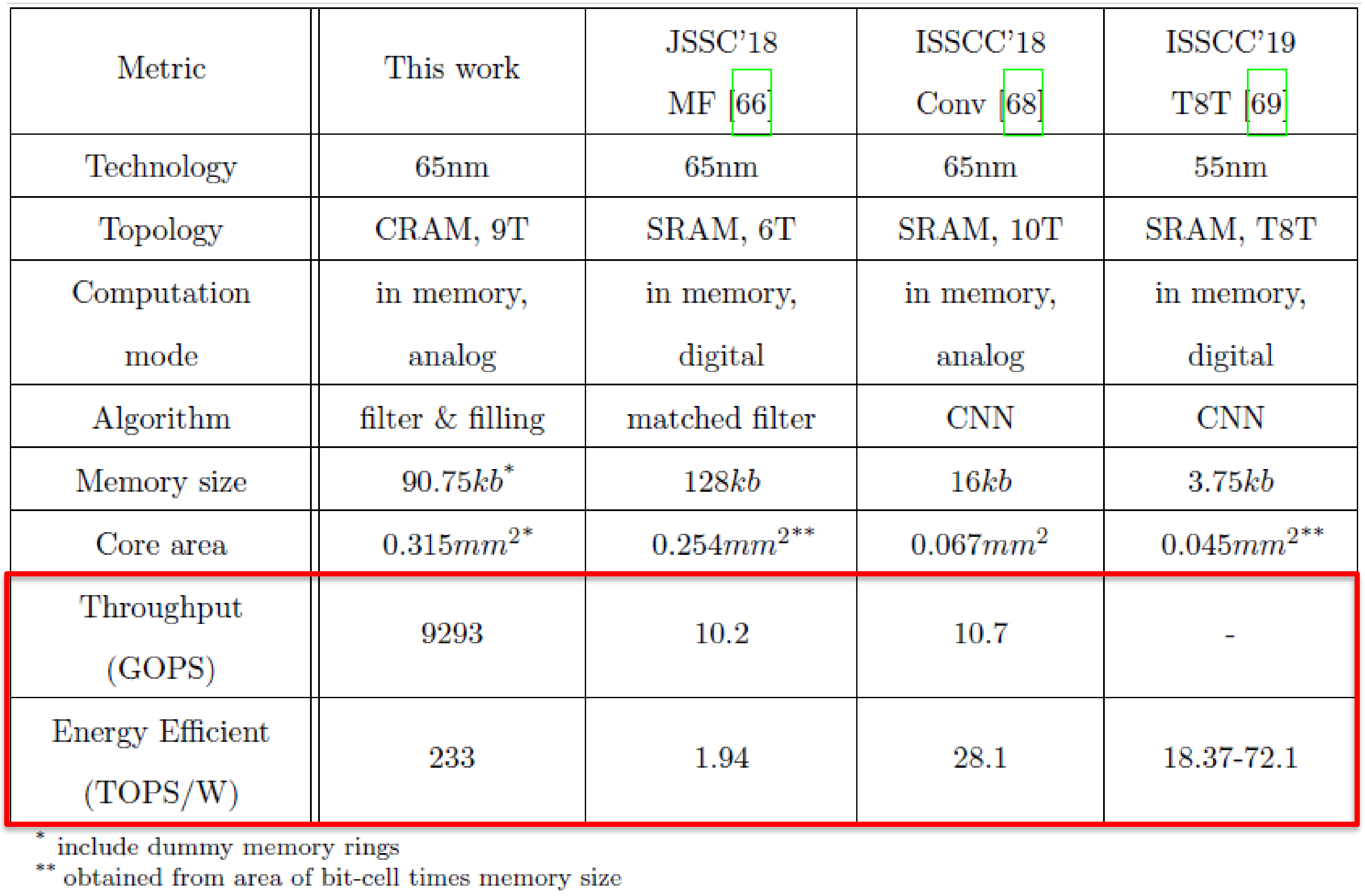
(2) IMC for denoising



Lower equivalent Res, severer filtering

Results and discussion

- **CRAM performance**



Metric	This work	JSSC'18 MF [66]	ISSCC'18 Conv [68]	ISSCC'19 T8T [69]
Technology	65nm	65nm	65nm	55nm
Topology	CRAM, 9T	SRAM, 6T	SRAM, 10T	SRAM, T8T
Computation mode	in memory, analog	in memory, digital	in memory, analog	in memory, digital
Algorithm	filter & filling	matched filter	CNN	CNN
Memory size	$90.75kb^*$	$128kb$	$16kb$	$3.75kb$
Core area	$0.315mm^{2*}$	$0.254mm^{2**}$	$0.067mm^2$	$0.045mm^{2**}$
Throughput (GOPS)	9293	10.2	10.7	-
Energy Efficient (TOPS/W)	233	1.94	28.1	18.37-72.1

* include dummy memory rings
** obtained from area of bit-cell times memory size

Outline

- Introduction
- CRAM structure
  - Memory mode
  - In-memory computing (IMC) mode
- Results and discussion
- **Conclusion**
- Future Work

Conclusion

- A novel CRAM based analog IMC architecture is presented to fulfil image denoising and filling tasks.
- Charge domain In-memory computing
  - Ultra low power
  - Parallel computing

Outline

Future work

- Improve the robust of CRAM
  - Writability and read stability
- Reduce the distribution of diffusion resistor
  - Resistor mismatch degrades the uniformity of the RC network
- Region proposal algo
  - In/near-memory computing for region proposal

THANKS!

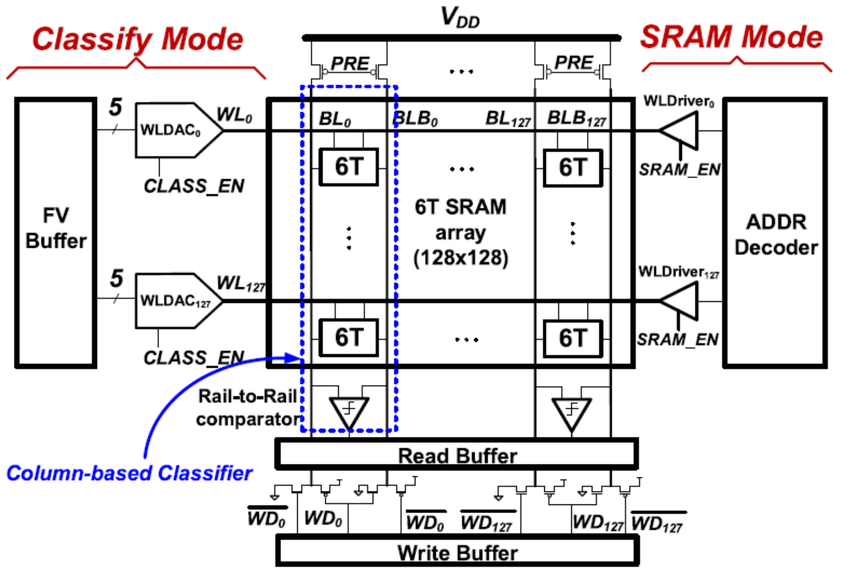
Q&A

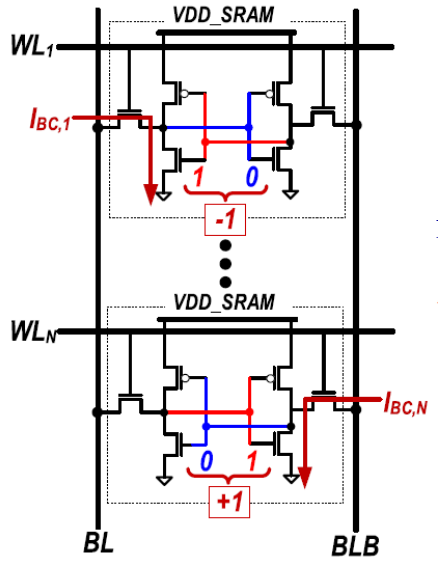
Literature Review: In-memory computing

Analog IMC (6T SRAM)

Binary weights and analog feature inputs (through DAC)

- **SRAM mode: like normal memory to store weights**
- **IMC mode: all wordlines (WLs) are driven at once to analog voltages corresponding to input feature vectors.**





BL/BLB discharge

$I_{cell} \propto (V_{WL} - V_t)$

$V_a \propto I_{cell}$

☹ **Write-disturb issue**

☹ **variation and nonlinearity => nonideal => weak classifier**

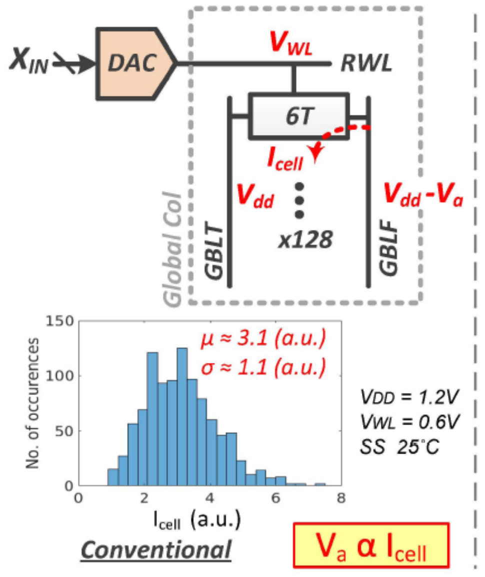
J. Zhang, Z. Wang and N. Verma, "In-Memory Computation of a Machine-Learning Classifier in a Standard 6T SRAM Array," in IEEE Journal of Solid-State Circuits, vol. 52, no. 4, pp. 915-924, April 2017.

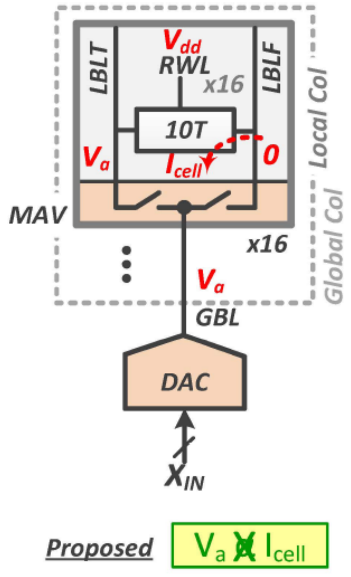
Literature Review: In-memory computing

Analog IMC (10T SRAM)

Binary weights and <u>signed</u> analog feature inputs (through DAC)

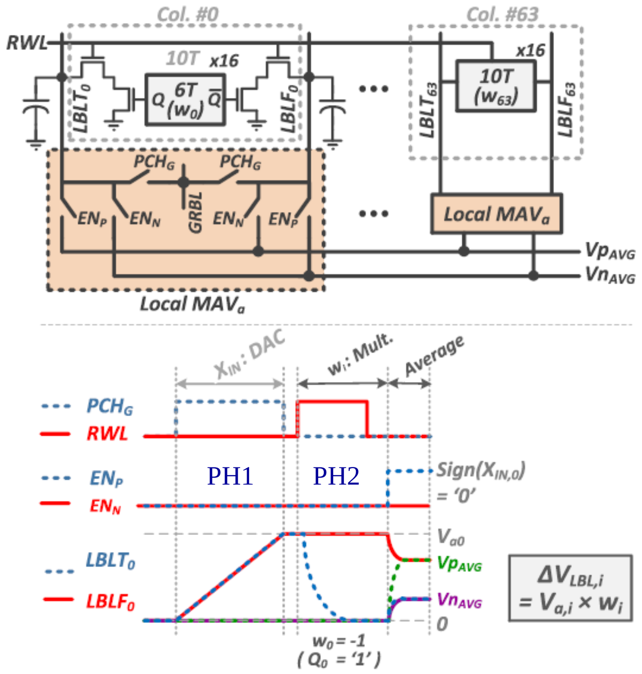
- **SRAM mode: like normal memory to store weights**
- **IMC mode: phase1– input to DAC, precharge GRBL; phase2– active RWL, discharge LBLT/LLBF to 0, get ΔV_{LBL}, and obtain the average horizontally.**





Va various widely due to Icell variation

Va has no variation due to Icell



A. Biswas and A. P. Chandrakasan, "CONV-SRAM: An Energy-Efficient SRAM With In-Memory Dot-Product Computation for Low-Power Convolutional Neural Networks," in IEEE Journal of Solid-State Circuits, vol. 54, no. 1, pp. 217-230, Jan. 2019.