6.5930/1

Hardware Architectures for Deep Learning

Advanced Technologies

April 22, 2024

Joel Emer and Vivienne Sze

Massachusetts Institute of Technology
Electrical Engineering & Computer Science

Advanced Storage Technology

- **Bring compute and memory closer together** to reduced the cost of data movement
- **Processing/Compute Near Memory** *(a.k.a. near-data processing)*
  - Embedded DRAM (eDRAM)
    - Increase on-chip storage capacity
  - 3D Stacked DRAM (e.g., Hybrid Memory Cube (HMC), High Bandwidth Memory (HBM))
    - Increase memory bandwidth
- **Processing/Compute In Memory** *(a.k.a. in-memory computing)*
  - Static Random Access Memories (SRAM), Dynamic Random Access Memories (DRAM), and Non-Volatile Memories (NVM)
  - Processing ***integrated into*** memory (more aggressive form of processing near memory) versus processing ***using*** memory (using memory device to perform compute)

SRAM (kB – MB)

- SRAM accounts for majority of on-chip storage
  - SRAM is significant portion of chip area
- Usually, 6 transistors per bit-cell

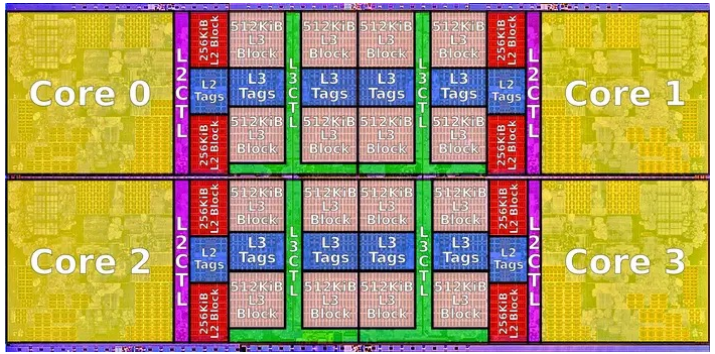
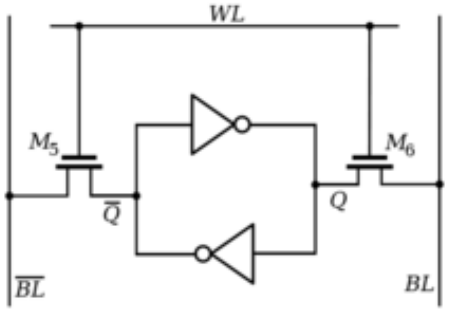
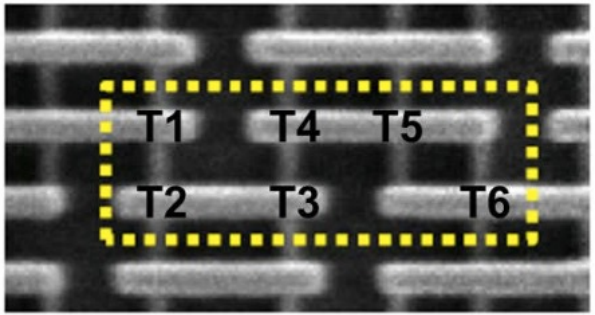


Image Source: https://en.wikichip.org/wiki/amd/microarchitectures/zen

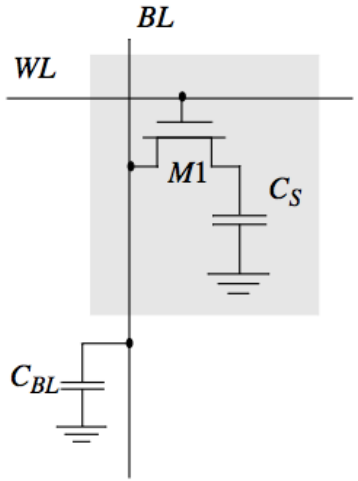


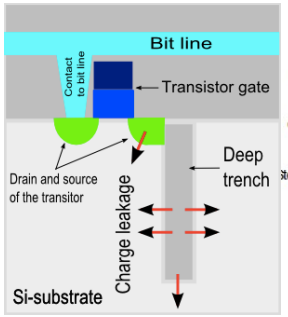


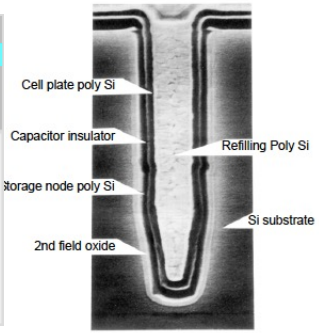
Bit cell size
0.75um^2 in 14nm

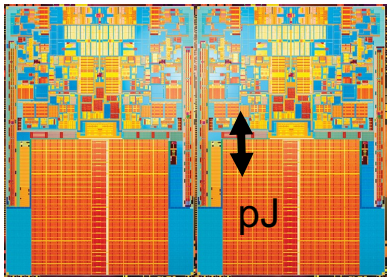
DRAM (GB)

- Higher density than SRAM
  - 1 transistor per bit-cell
  - Needs periodic refresh
- Special device process
  - Usually off-chip (except eDRAM – which is pricey!)
  - Off-chip interconnect has much higher capacitance

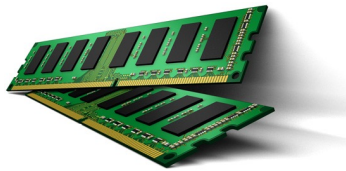






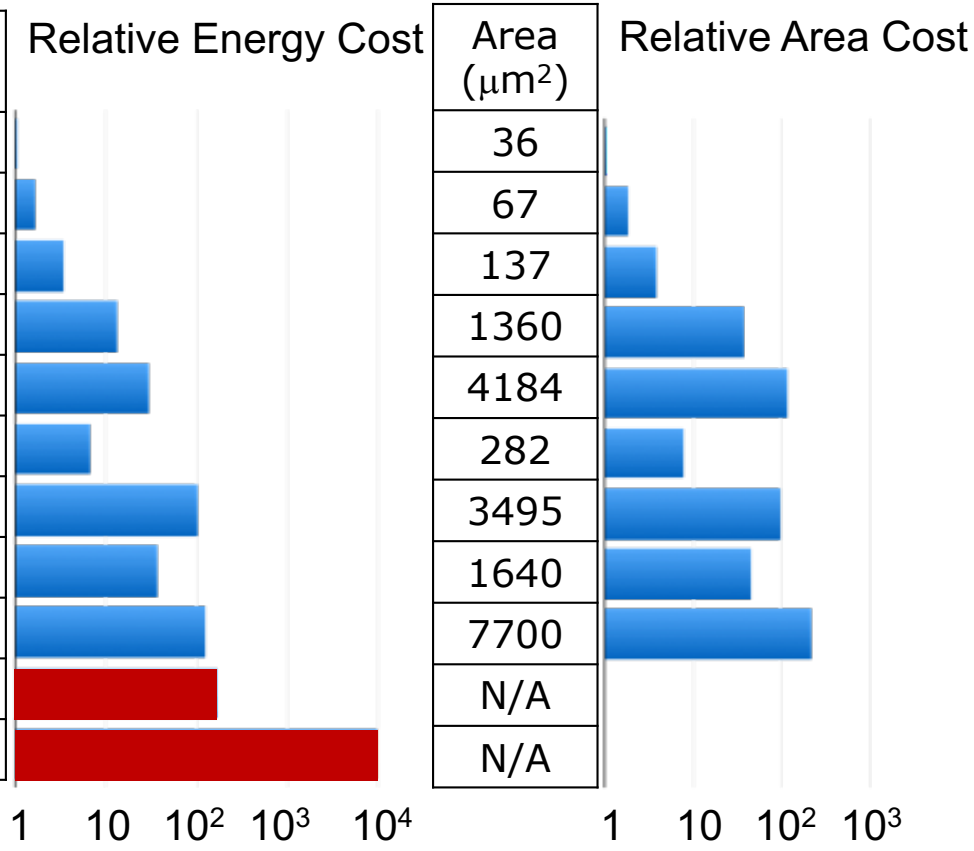


nJ



Memory Access Cost for DRAM is Expensive

Operation:	Energy (pJ)	Area (μm^2)
8b Add	0.03	36
16b Add	0.05	67
32b Add	0.1	137
16b FP Add	0.4	1360
32b FP Add	0.9	4184
8b Mult	0.2	282
32b Mult	3.1	3495
16b FP Mult	1.1	1640
32b FP Mult	3.7	7700
32b SRAM Read (8KB)	5	N/A
32b DRAM Read	640	N/A

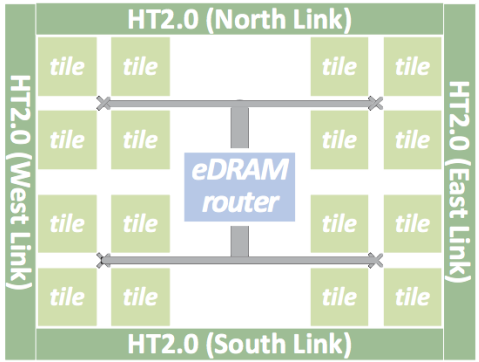


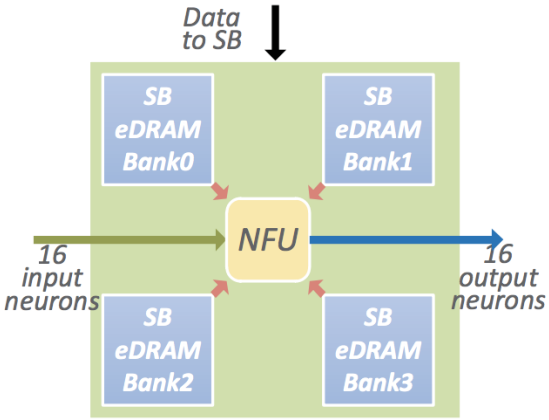
[**Horowitz**, *ISSCC* 2014]

eDRAM (DaDianNao)

- Advantages of eDRAM
  - 2.85x higher density than SRAM
  - 321x more energy-efficient than DRAM (DDR3)
- Store weights in eDRAM (36MB)
  - Target fully connected layers since dominated by weights

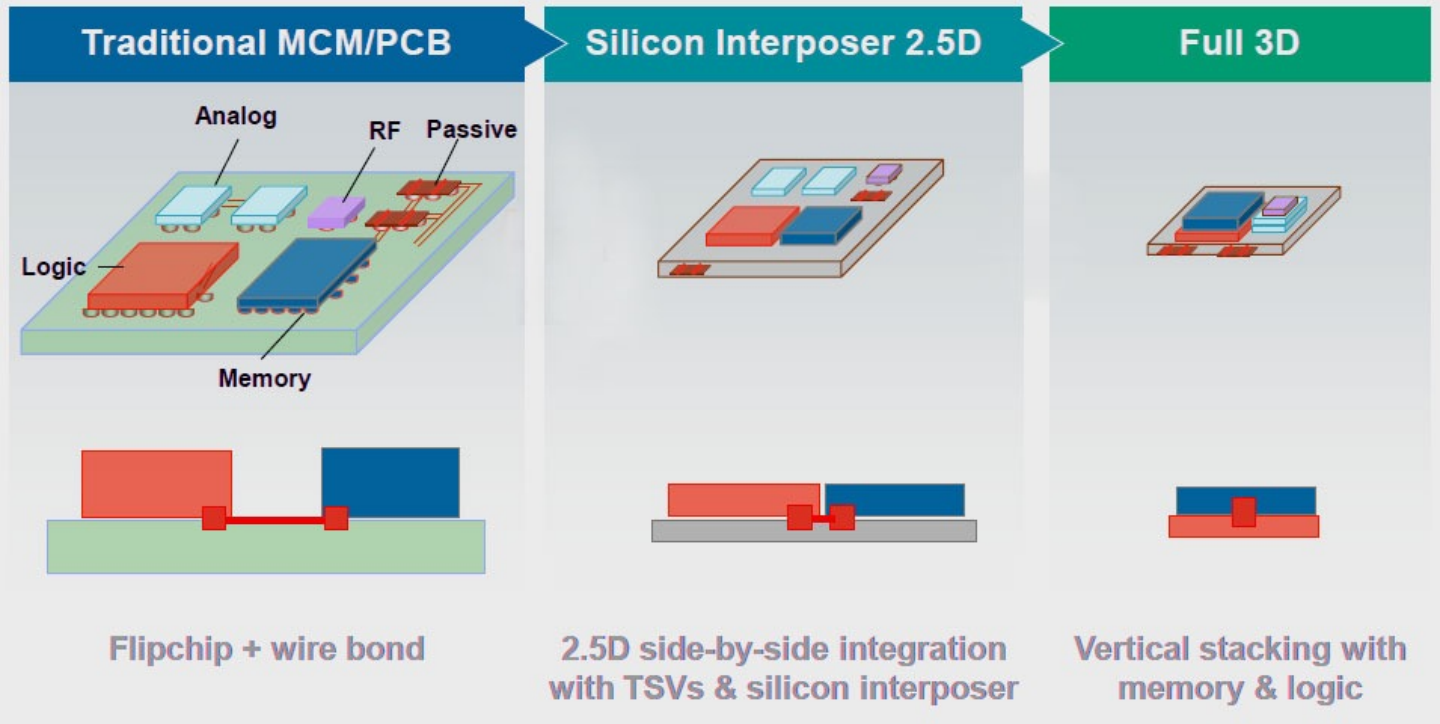
16 Parallel Tiles





[**Chen**, *MICRO* 2014]

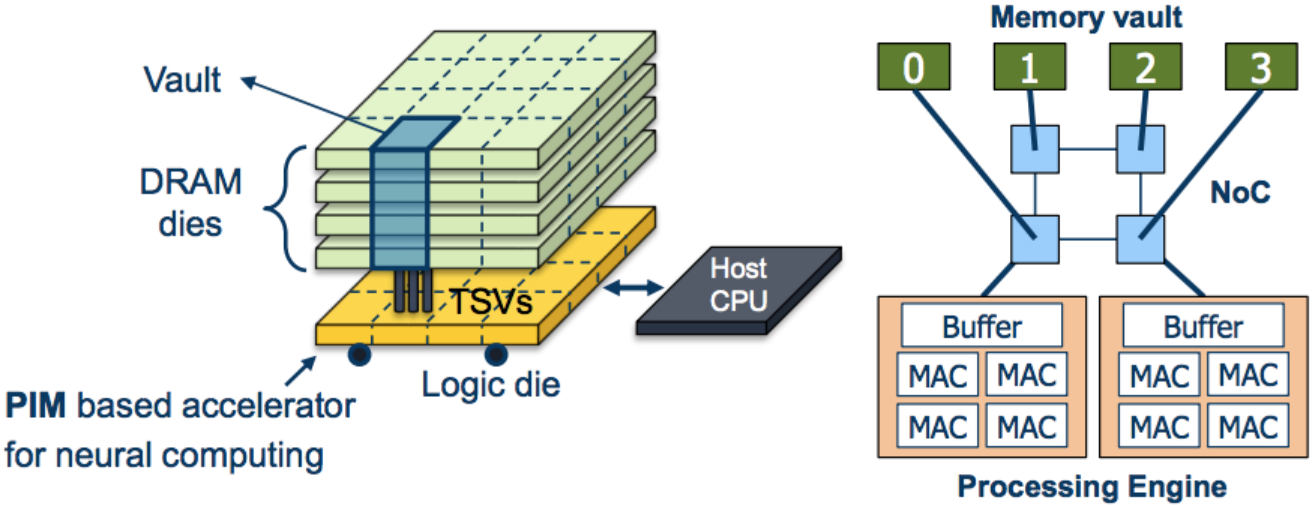
Going 3D!



Source: https://www.einfochips.com/blog/2-5d-3d-ics-new-paradigms-in-asic/

Stacked DRAM (NeuroCube)

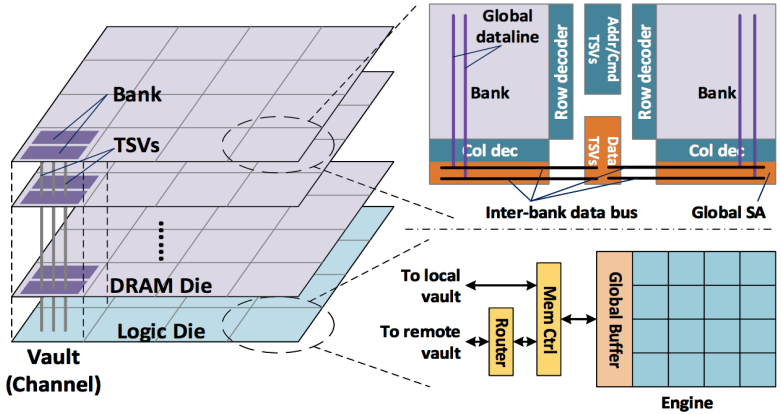
- NeuroCube on Hybrid Memory Cube Logic Die
  - 6.25x higher BW than DDR3
    - HMC (16 ch x 10GB/s) > DDR3 BW (2 ch x 12.8GB/s)
  - Computation closer to memory (reduce energy)



[**Kim**, *ISCA* 2016]

Stacked DRAM (Tetris)

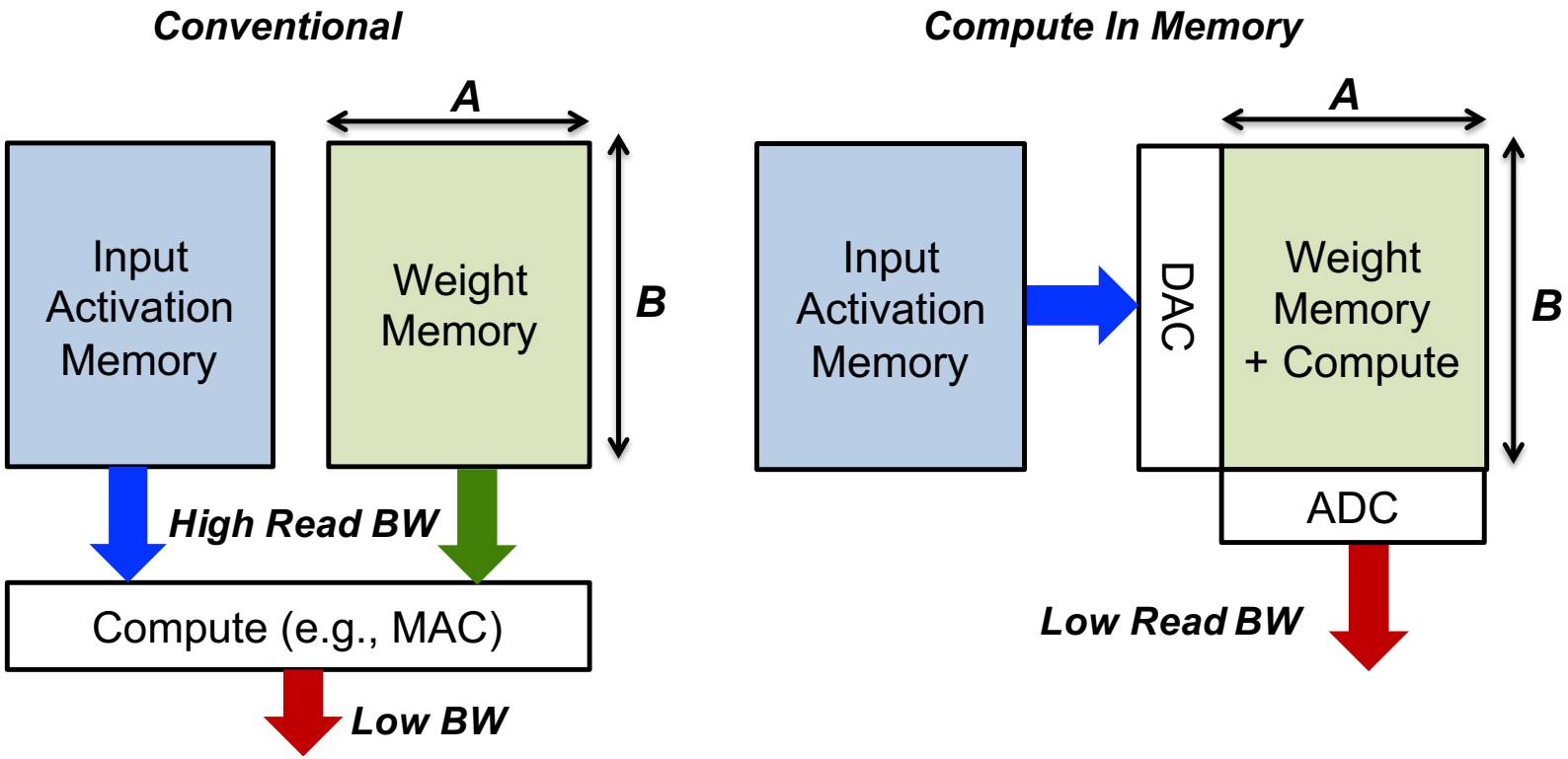
- Explores the use of HMC with the Eyeriss **spatial architecture** and **row stationary dataflow**
- Allocates **more area to the computation (PE array) than on-chip memory (global buffer)** to exploit the low energy and high throughput properties of the **HMC**
  - **1.5x energy reduction, 4.1x higher throughput** vs. 2-D DRAM



[**Gao**, *ASPLOS* 2017]

Eyeriss design

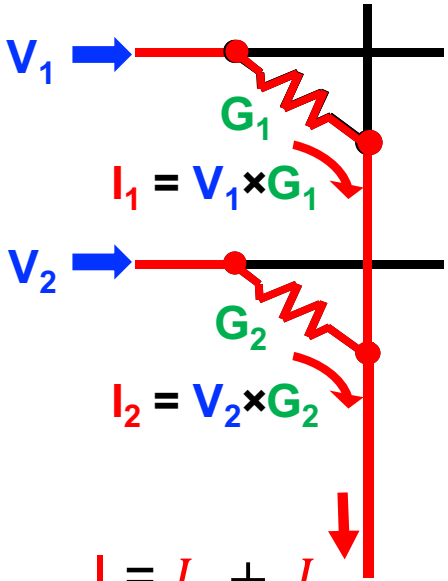
Conventional Processing vs. CiM



Analog Computing

Analog computing is typically required to bring the computation into the array of storage elements or into its peripheral circuits

Activation is input voltage (V_i)



Weight is resistor conductance (G_i) [= 1/resistance]

Many options for "resistance"

Partial sum is output current (I_i)

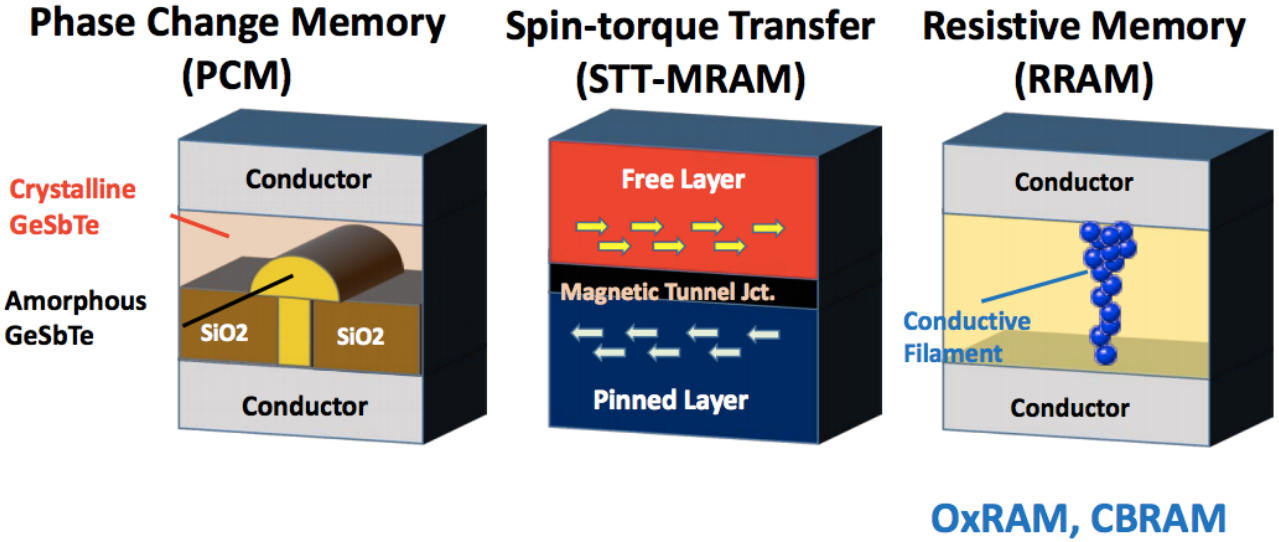
$$I = I_1 + I_2$$

$$I = V_1 \times G_1 + V_2 \times G_2$$

CiM Using Memristors in Non-Volatile Memory

Use memristors as programmable weights (resistance)

Candidate memristor devices



Source: Darsen Lu, MICRO-49 Tutorial on Emerging Memory

Candidate Devices for Memristor

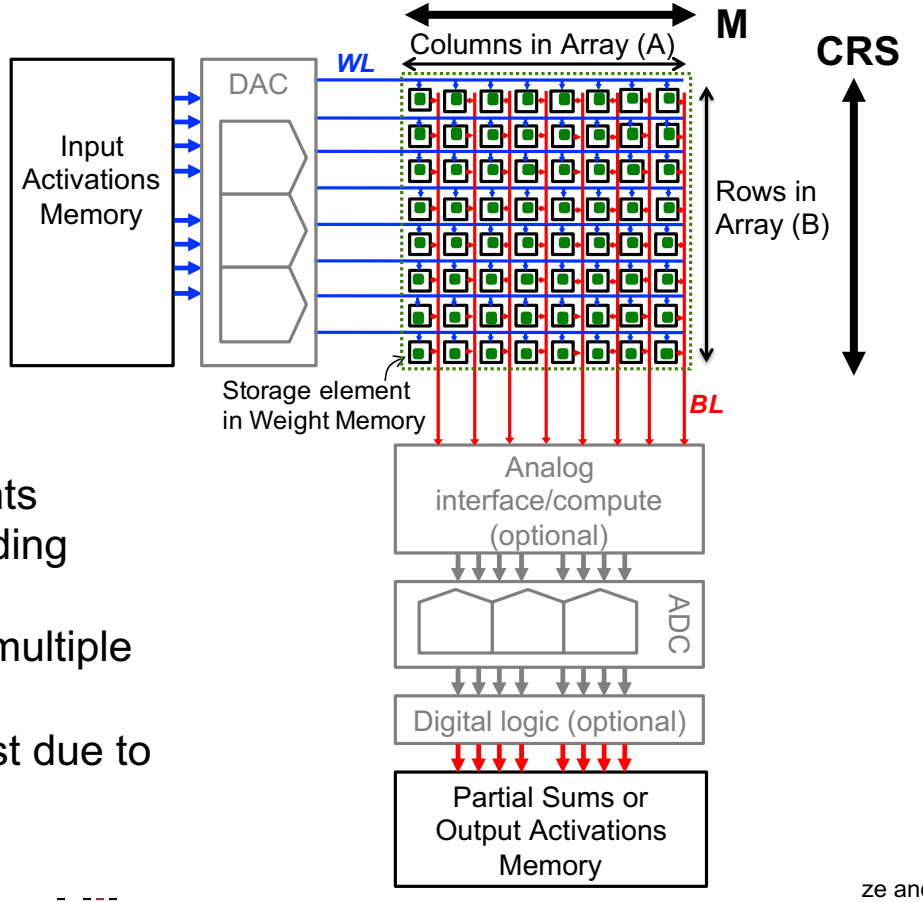
		PCM	STT-MRAM	RRAM	NAND Flash	DRAM
Power	Ewrite/ Bit Iwrite	18pJ [15][1] 100µA [15]	1.0pJ [20] 50µA [33]	1.0pJ [20] 1.0µA [18]	100pJ [31]	<1.0 pJ [9]
Perform-ance[2]	Write Lat.	150ns [15]	5ns [6]	50ns [20]	>100µs	5ns [27]
	Read Lat.	80ns [28]	10ns [5]	<10ns [30]	15-50µs [35]	20–80ns [35]
Reliabi-lity	Program Window	3 bit/cell [12]	Good [3]	Variable [23]	4 bit/cell [32]	Good
	Endurance	10^8-10^9 [4,9]	Unlimited [1]	10^5-10^{10} [1]	10^5-10^6 [8]	Unlimited
	Retention	R-drift[12]	Good [6]	RTN [24]	Good	64ms
Density	Cell size	4 F^2 [15]	12 F^2 [33]	4-6F^2 [21]	<4 F^2 [15]	7 F^2 [15]

Emerging Memory Technologies
(Speed of DRAM; Non-volatility of NAND)

1. Estimated using I_reset * Vdd * t_write
2. Required programming pulse duration

Weight Stationary Dataflow for CiM

Transform into **matrix-vector multiply** with weights in matrix, and input activations in vector



Benefits

- Reduces data movement of weights
- Higher memory bandwidth by reading multiple weights in parallel
- Higher throughput by performing multiple computations in parallel
- Lower input activation delivery cost due to increased density of compute

Loop Nest for CiM

For Fully Connected layer

```
i = Array(CHW)          # Input activations
f = Array(M, CHW)       # Filter weights
o = Array(M)            # Output partial sums

parallel–for m in [0, M):
    parallel–for chw in [0, CHW):
        o[m] += i[chw] * f[m, chw]
```

For Convolutional layer (1-D toy example)

```
i = Array(C, W)         # Input activations
f = Array(M, C, S)      # Filter weights
o = Array(M, Q)         # Output partial sums

parallel–for m in [0, M):
    parallel–for s in [0, S):
        parallel–for c in [0, C]:
            for q in [0, Q):
                w = q + s
                o[m, q] += i[c, w] * f[m, c, s]
```

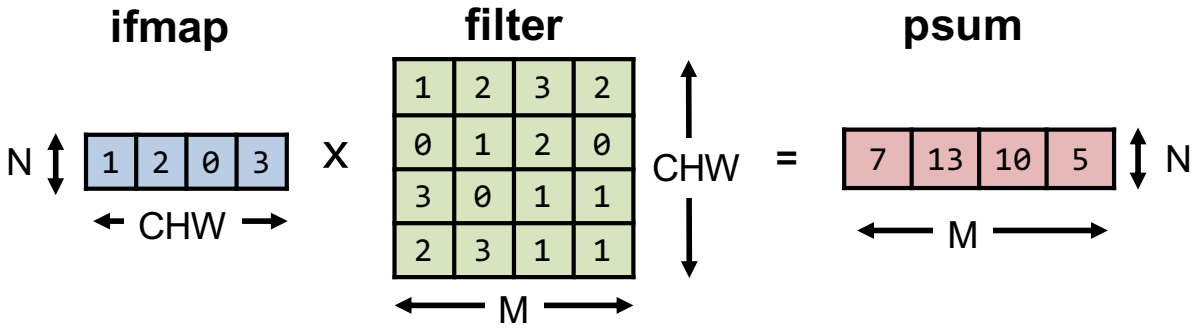
Design Considerations for CiM

- Analog computing has **increased sensitivity** to circuit and device non-idealities (i.e., non-linearity and process, voltage and temperature variations)

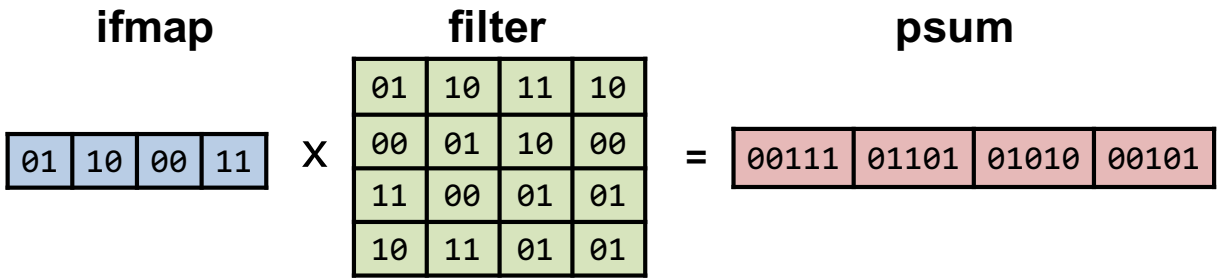
- Requires **trade offs between energy efficiency, throughput, area density, and accuracy**, which reduce the achievable gains over conventional architectures

Toy Example for CiM

Decimal Representation

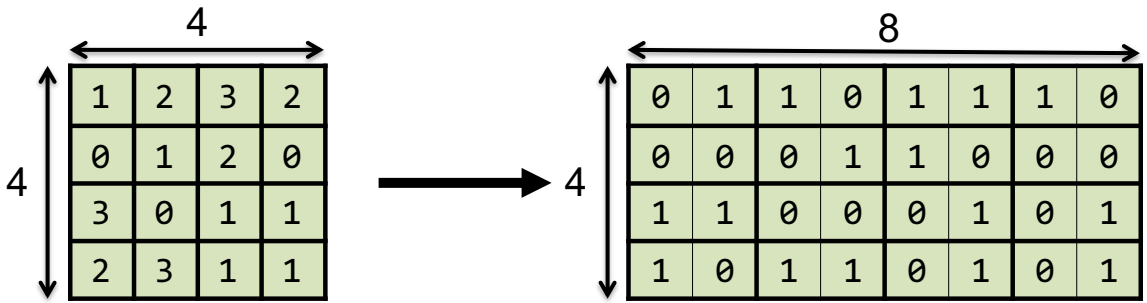


Binary Representation



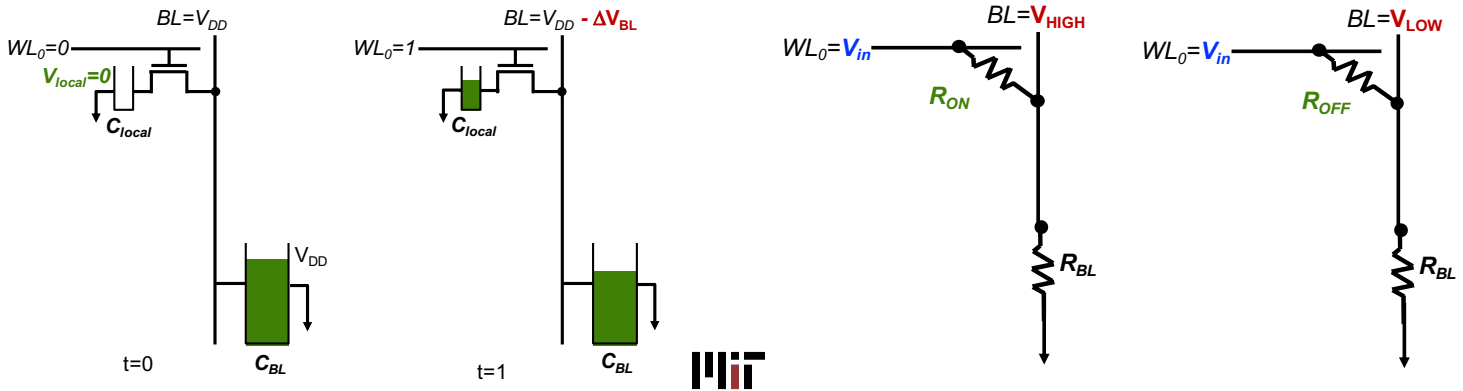
Number of Storage Elements per Weight

- Ideally, it would be desirable to be able to use one storage element (i.e., one device or bit cell) per weight to maximize density.
- In practice, **multiple storage elements are required per weight** due to the limited precision of each device or bit cell (typically on the order of 1 to 4 bits)
  - This is also referred to as **bit slicing** of the weight **(weight slicing)**



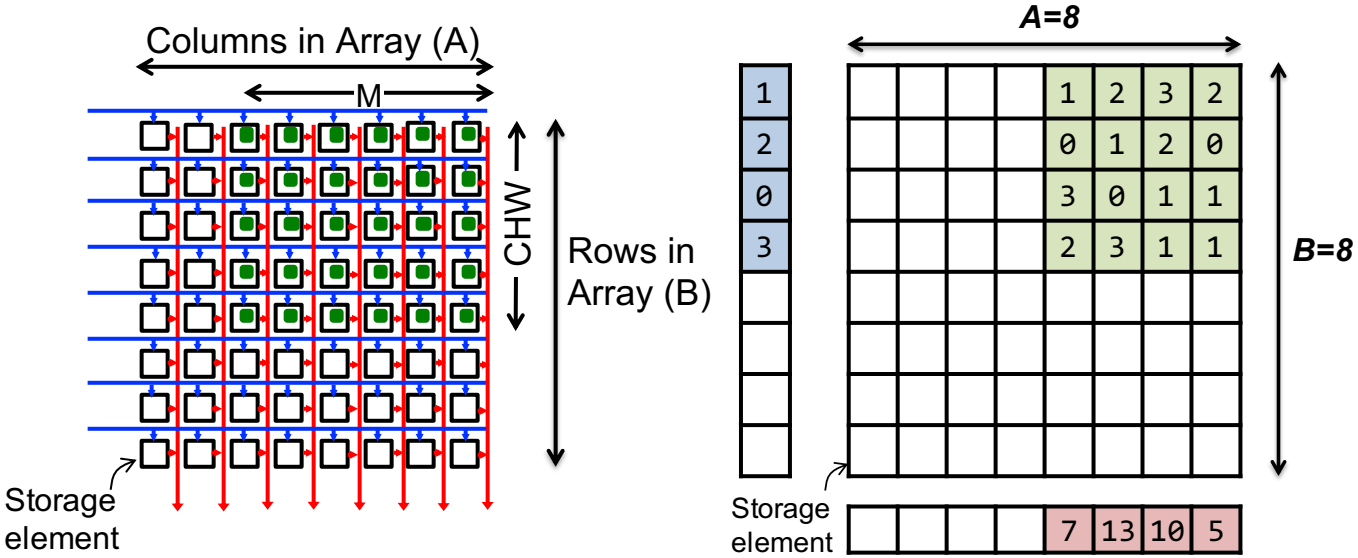
Array Size

- Ideally, it would be **desirable to have a large array size**
  - Increases weight read bandwidth and throughput
  - Increases area density → amortize the cost of the peripheral circuits (e.g., ADC, DAC) which can account for over 50% of NVM designs
- Array size limited by the resistance and capacitance of word line and bit line wires, which impacts robustness, speed and energy consumption
  - If capacitance or resistance of bit line much larger than storage element, it is difficult to sense value in storage element (i.e., change in bit-line voltage/current due to value in storage element is reduced)



Array Size

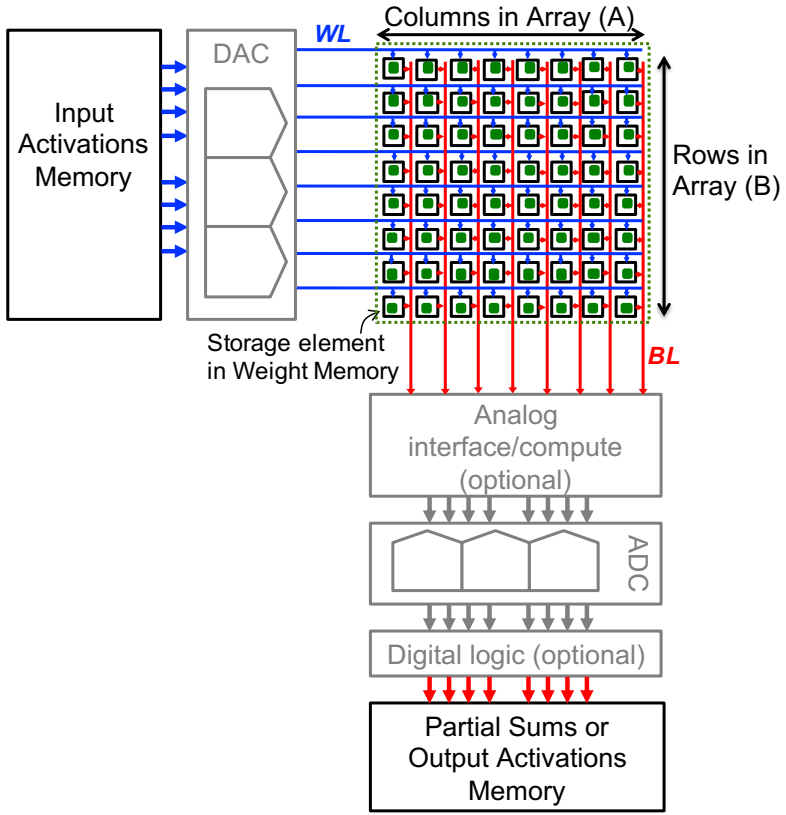
Utilization of the array will drop if the workload cannot fill entire column or entire row



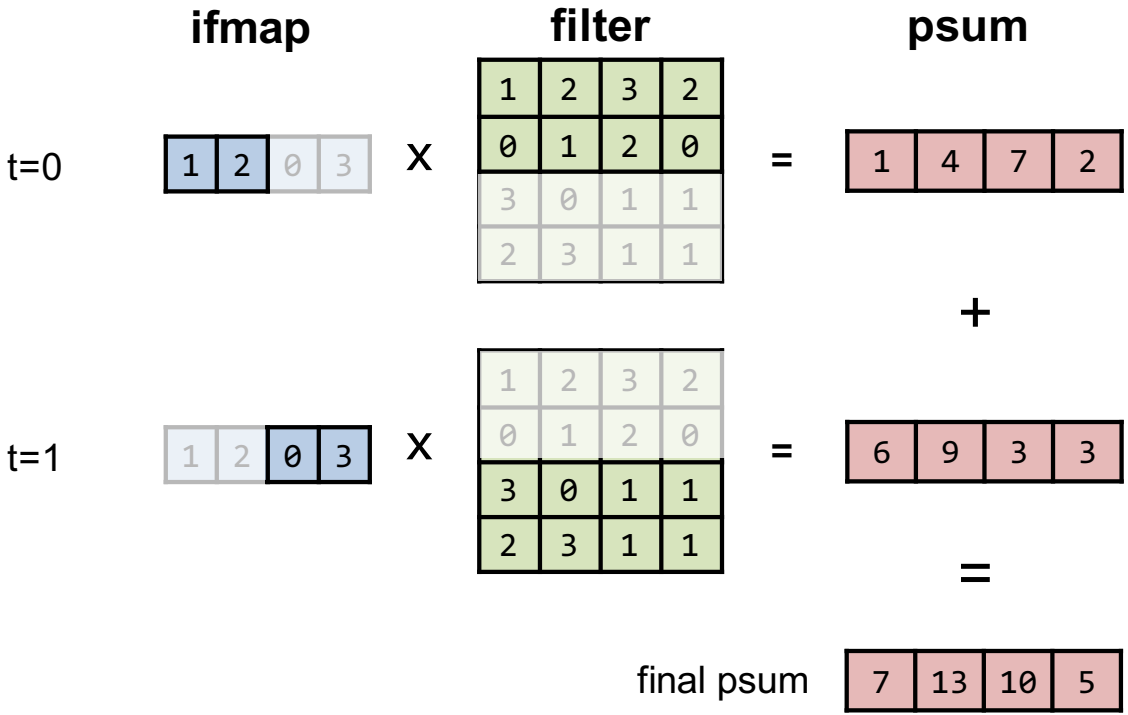
Number of Rows Activated in Parallel

- Ideally, it would be desirable to use all rows at once to maximize parallelism for high bandwidth and high throughput

- In practice, the number of rows that can be used at once is limited due to
  - ADC resolution (number of bits it can resolve)
  - Cumulative effect of the device variations
  - Maximum voltage drop or accumulated current on bit line



Number of Rows Activated in Parallel

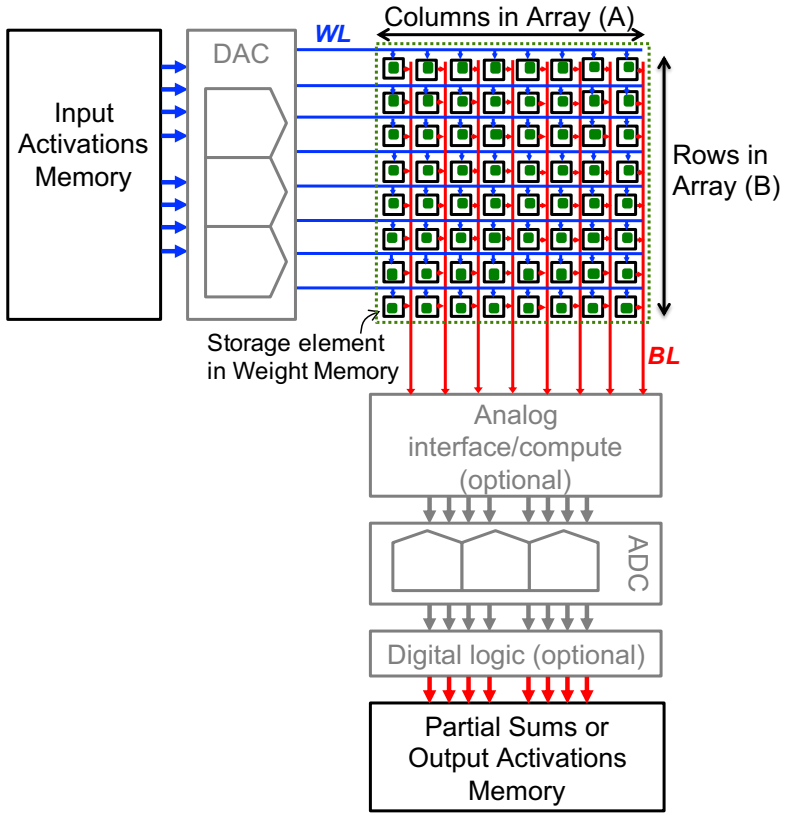


Multiple cycles to complete accumulation across rows → **Reduces Throughput**

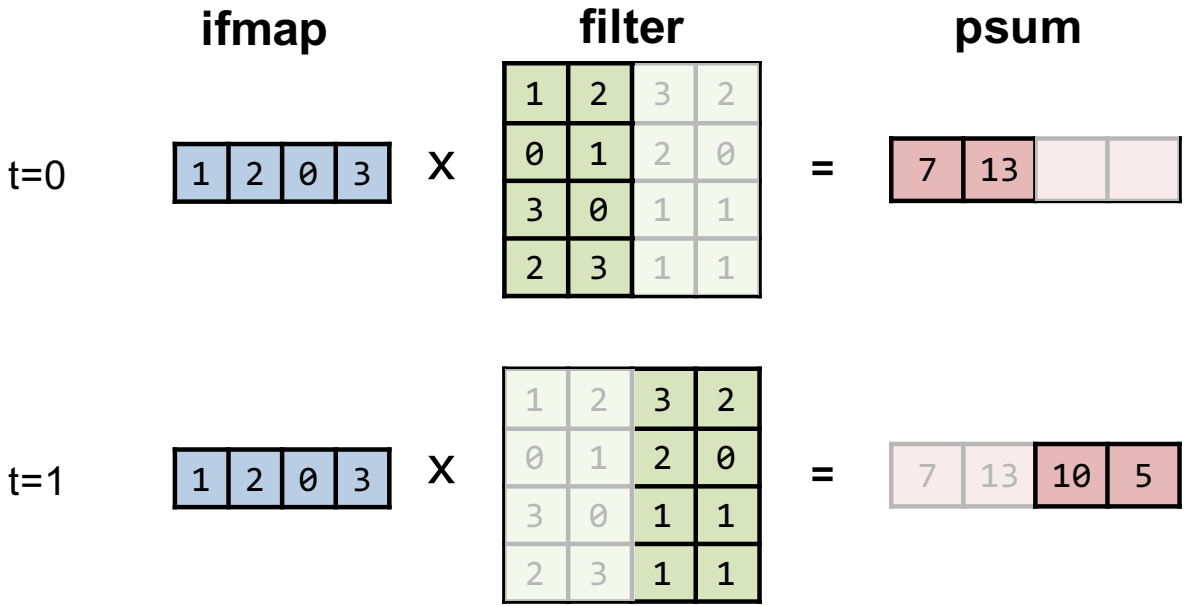
Number of Columns Activated in Parallel

- Ideally, it would be desirable to use all columns at once to maximize parallelism for high bandwidth and high throughput

- In practice, the number of columns that can be used is limited by ADC area



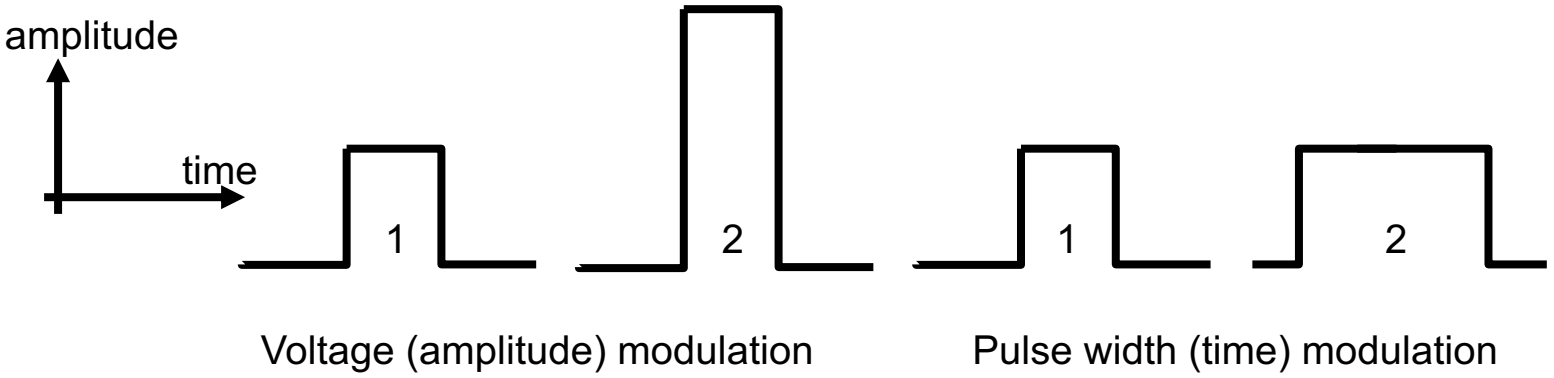
Number of Columns Activated in Parallel



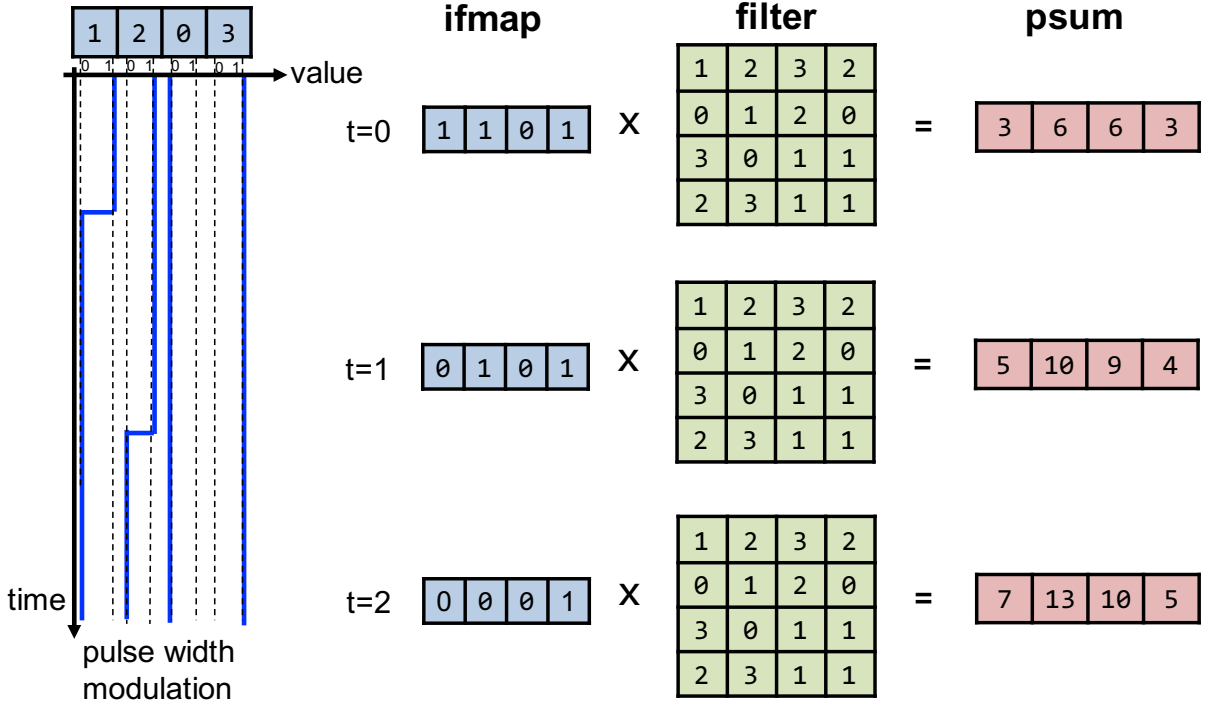
Multiple cycles to complete accumulation for all columns → **Reduces Throughput**

Time to Deliver Input

- Ideally, encode input activations onto the word line in the minimum amount of time for maximum throughput (e.g., voltage modulation)
- In practice, challenging due to non-linearity of devices and complexity of DAC → need to encode in time → **Reduces Throughput**



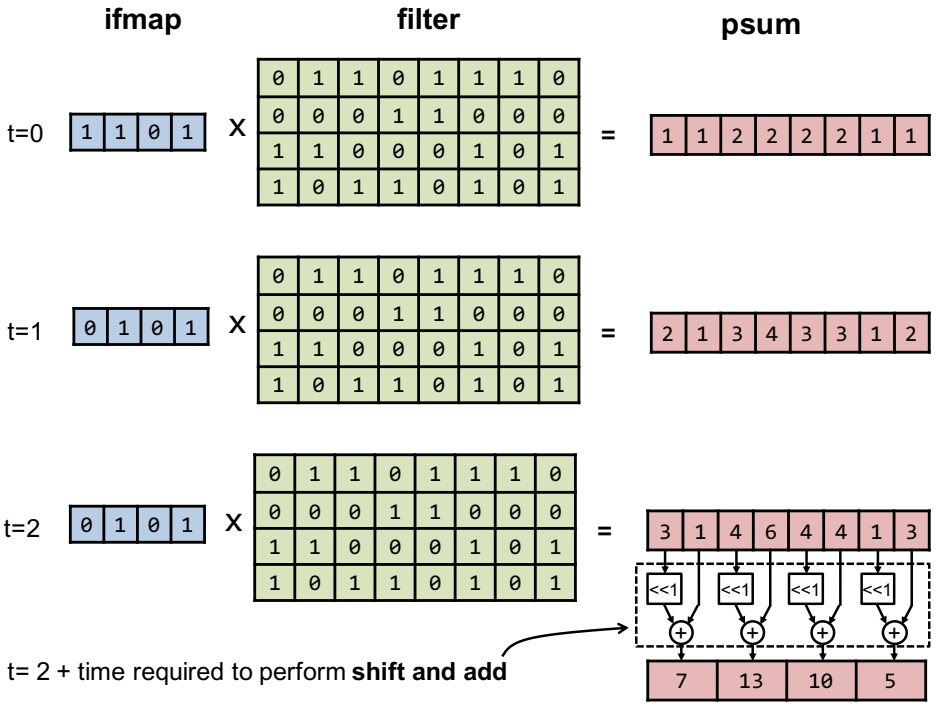
Time to Deliver Input



Time to Compute a MAC

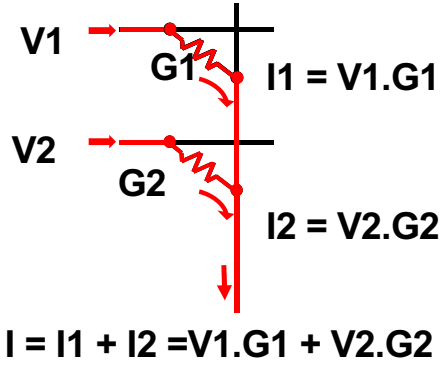
- Ideally, compute MAC in a single cycle for high throughput

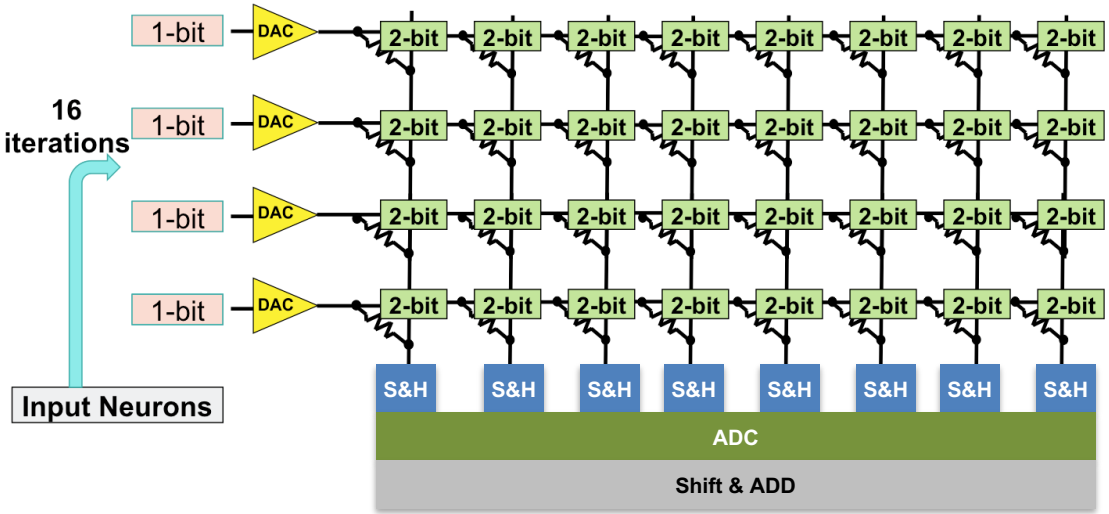
- In practice, the storage element typically can only perform one-bit or two-bit operations → need multiple cycles to build up to a multi-bit operation (temporal accumulation) → **Reduces throughput**



ISAAC

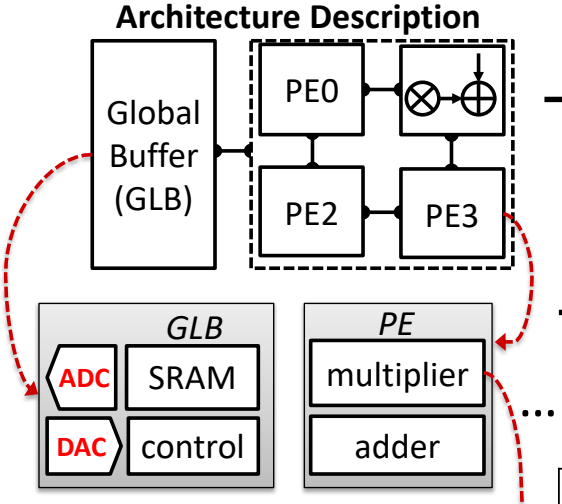
- Replace eDRAM in DaDianNao with memristors
- 16-bit dot-product operation
  - 8 x 2-bits per memristors (weight slicing)
  - 1-bit per cycle computation (input slicing)





[**Shafiee**, *ISCA* 2016]

Modeling CiM with Accelergy



Accelergy

Action Counts

name	action	count
PE0	compute	500
PE1	...	

Energy Estimation

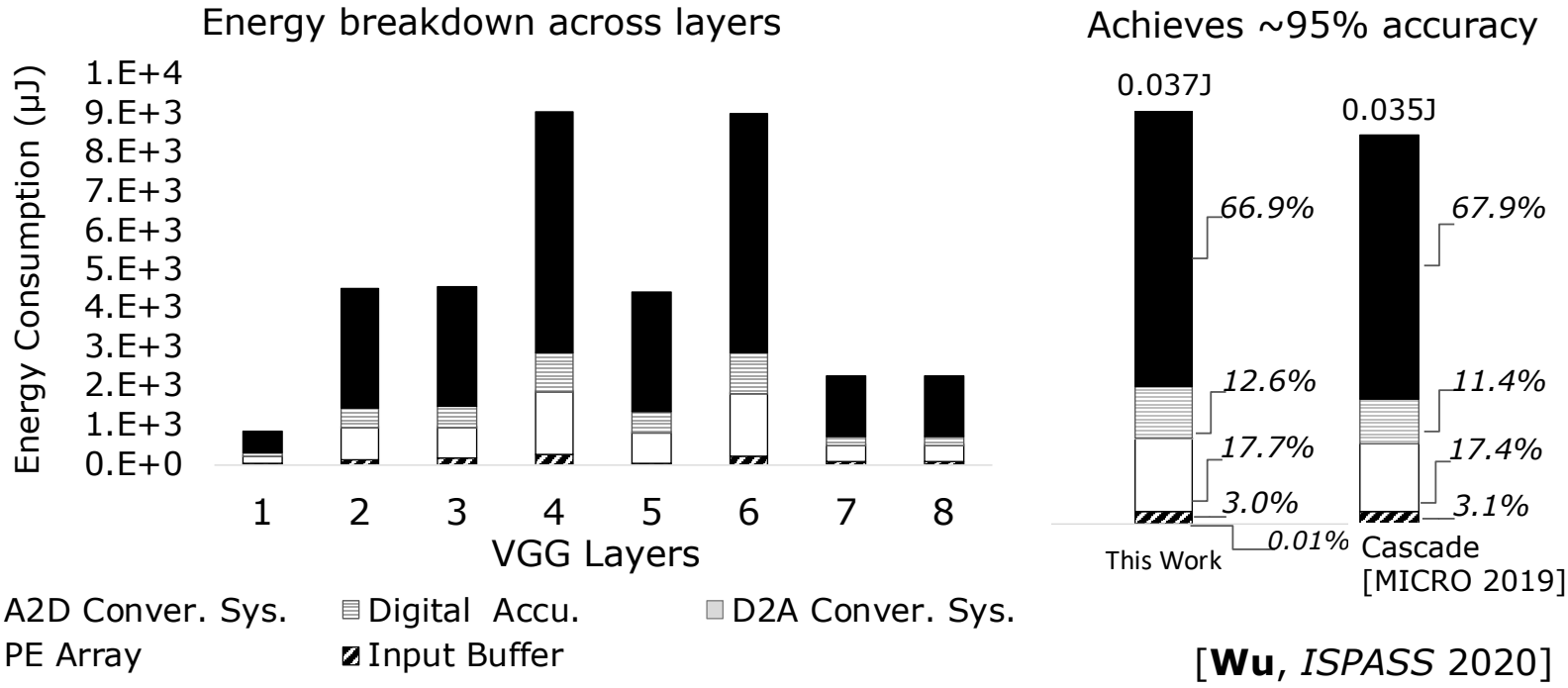
name	energy (pJ)
PE0	~~1500~~ E_{total}
PE1	...

Energy Estimation Plug-in

name	technology	width	action	energy (pJ)
multiplier	~~65nm~~ memristor	16	multiply	~~0.8~~ E_{mult}
adder	...			
ADC				
DAC				

[**Wu**, *ISPASS* 2020]

Modeling CiM with Accelergy



[**Wu**, *ISPASS* 2020]

ADC dominates energy consumption