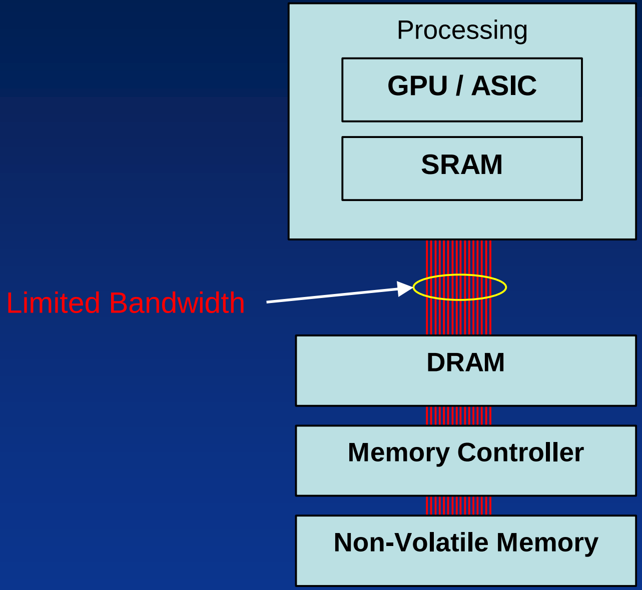
Forging the Way in AI Architecture with ReRAM Based Computational Memory

Hagop Nazarian

VP of Engineering, Cofounder

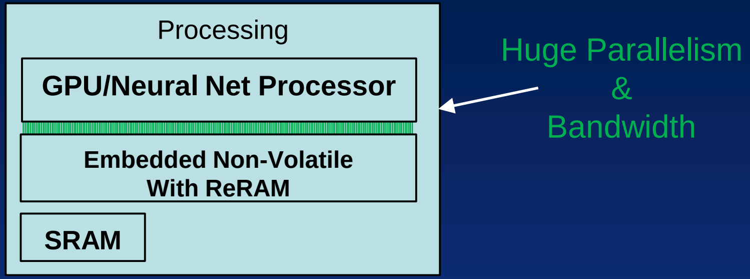
Crossbar Technology Inc.

AI Architectural Evolution



Memory separated from the computational unit

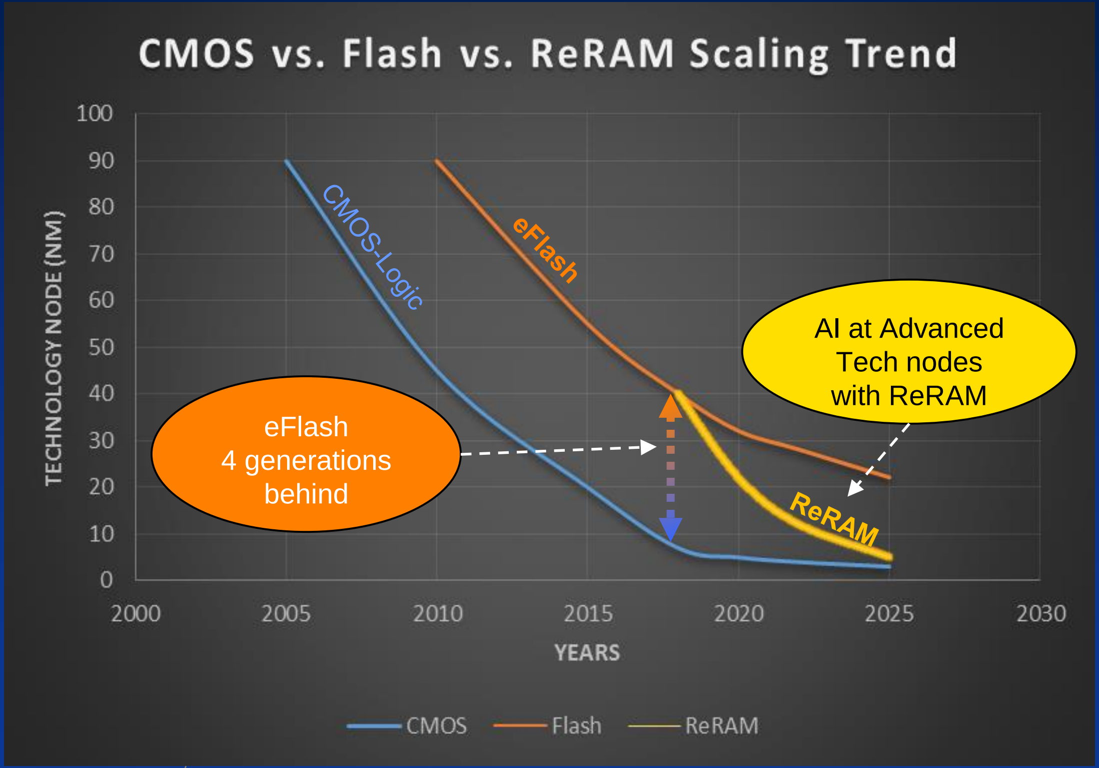
John Von Neumann Architecture



Embedded ReRAM with NN Processors

- ✓ Computational ReRAM array
- ✓ Monolithic solution at advanced CMOS nodes
- ✓ Interface bus size defined by AI System architects – not by memory or GPU manufacturers
- ✓ Large parallelism directly coupled to the processor
  - ✓ Yields huge bandwidth increase
  - ✓ Drastically reduces overall system energy consumption

ReRAM Bridging The Generational Gap



- eFlash is ~4 generations behind advanced Logic !!
- Flash technology does not scale with CMOS Logic process
- ReRAM is the non-Volatile memory choice for the advanced nodes in major foundries
- ReRAM Development already in Progress at least in 3 major foundries

What does an AI system Do?

AI systems can think and learn

- Training
- Inferring
- Classifying information
- Evaluating
- Use Low Energy and Low Latency
- Do all above in real-time at the edge

How does an AI system operate?

- To Train
  - Learns from the massive data
  - Establish relationships and trends
  - Interpolation and extrapolation models to be used for optimum solutions
  - Store them

- To Infer
  - Deduce estimated solutions or trends from observations and the trained model
  - Adapts to the environment and optimize the system
  - Store the new scheme

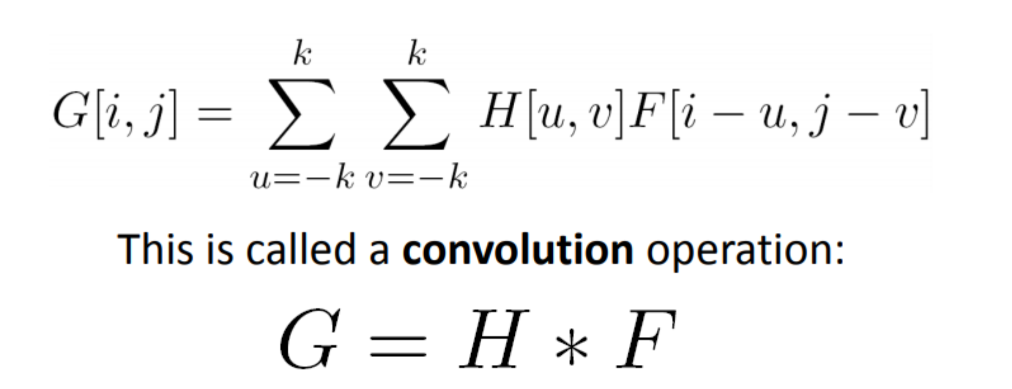
- To Classify information
  - Compares information, establishes relations, and stores them

- To Evaluate
  - Calculates, compares and finds the best given match conditions

AI Operations

- Add/Subtract/Multiply/Divide
- Comparison => Classification
- Randomization => to speed up searches
- Best fit, Best match
- Matrix operations – Convolution – Sparse
- Energy efficiency of the AI system

Example: Convolution



$$G[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u,v]F[i-u,j-v]$$

This is called a **convolution** operation:

$$G = H * F$$

- F is the image Matrix
- H is the Kernel Matrix
- G is the output

1	1	1
1	1	1
1	1	1

Mean Filter (Blurring)

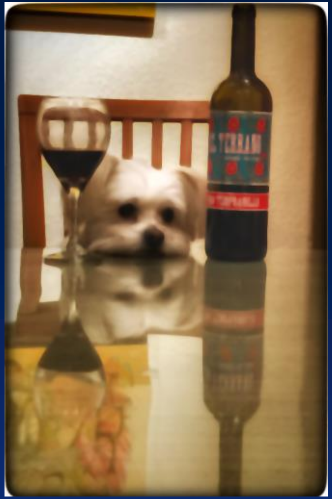
*

-1	-1	-1
-1	9	-1
-1	-1	-1

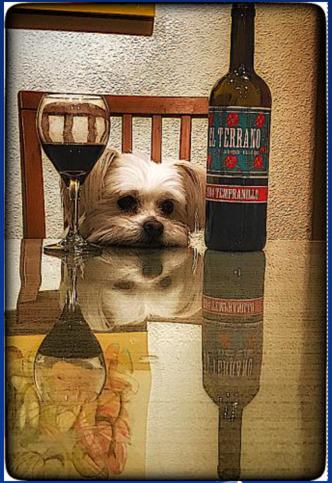
Sharpening Filter

*

Blurring



Sharpening



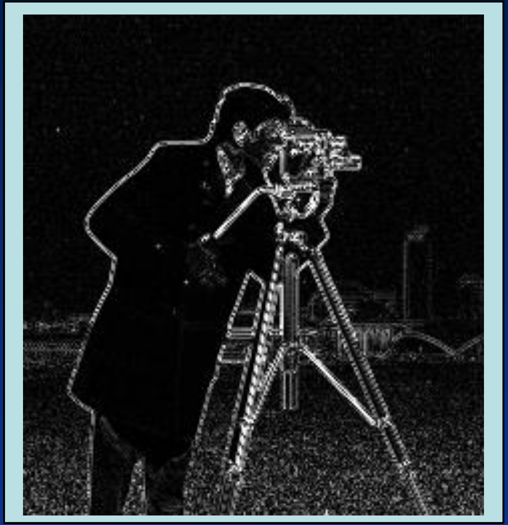
$$H \ast F = G$$

Edge Detection

0	1	0
1	-4	1
0	1	0

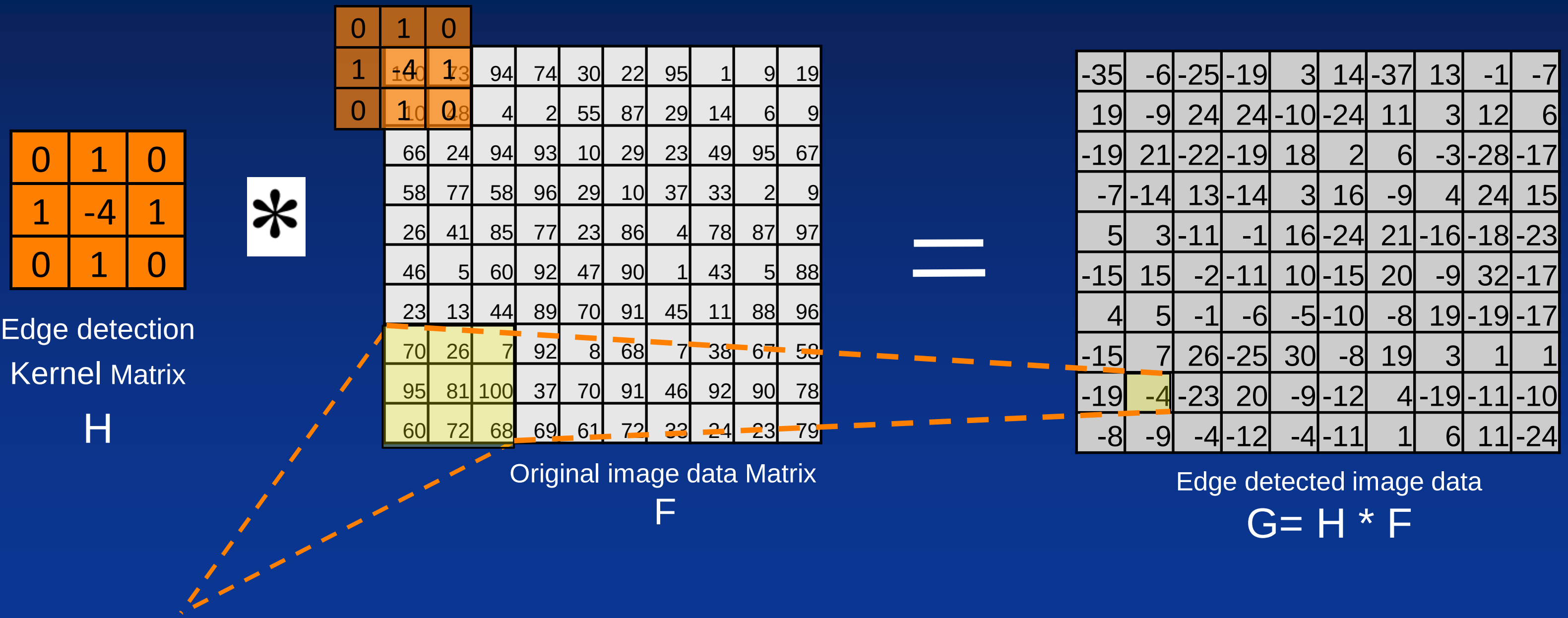
*

Edge Detection →



Source: Matlabtricks.com

Edge Detection Calculation



0	1	0
1	-4	1
0	1	0

Edge detection Kernel Matrix

H

*

0	1	0
1	-4	1
0	1	0

100	73	94	74	30	22	95	1	9	19
10	48	4	2	55	87	29	14	6	9
66	24	94	93	10	29	23	49	95	67
58	77	58	96	29	10	37	33	2	9
26	41	85	77	23	86	4	78	87	97
46	5	60	92	47	90	1	43	5	88
23	13	44	89	70	91	45	11	88	96
70	26	7	92	8	68	7	38	67	58
95	81	100	37	70	91	46	92	90	78
60	72	68	69	61	72	33	24	23	79

Original image data Matrix

F

=

-35	-6	-25	-19	3	14	-37	13	-1	-7
19	-9	24	24	-10	-24	11	3	12	6
-19	21	-22	-19	18	2	6	-3	-28	-17
-7	-14	13	-14	3	16	-9	4	24	15
5	3	-11	-1	16	-24	21	-16	-18	-23
-15	15	-2	-11	10	-15	20	-9	32	-17
4	5	-1	-6	-5	-10	-8	19	-19	-17
-15	7	26	-25	30	-8	19	3	1	1
-19	-4	-23	20	-9	-12	4	-19	-11	-10
-8	-9	-4	-12	-4	-11	1	6	11	-24

Edge detected image data

G= H * F

{70*(0)+26*(1)+7*(0)+95*(1)+81*(-4)+100*(1)+60*(0)+72*(1)+68*(0)} * (1/9) = -4

Santa Clara, CA
August 2018

Parallelisms in Computation

0	1	0
1	-4	1
0	1	0

100	73	94	74	30	22	95	1	9	19
10	48	4	2	55	87	29	14	6	9
66	24	94	93	10	29	23	49	95	67
58	77	58	96	29	10	37	33	2	9
26	41	85	77	23	86	4	78	87	97
46	5	60	92	47	90	1	43	5	88
23	13	44	89	70	91	45	11	88	96
70	26	7	92	8	68	7	38	67	58
95	81	100	37	70	91	46	92	90	78
60	72	68	69	61	72	33	24	23	79

Sequential Computation

1700 operations

900 multiplications and 800 additions

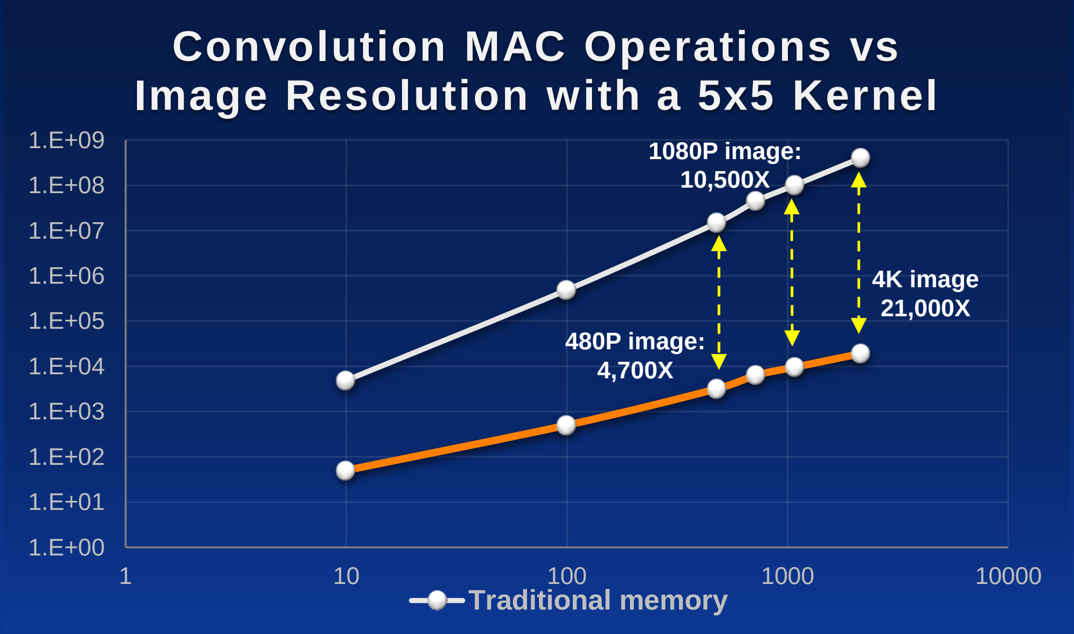
0	1	0	0	1	0	0	1	0	0	1	0
1	-4	1	1	-4	1	1	-4	1	1	-4	1
0	1	0	0	1	0	0	1	0	0	1	0

100	73	94	74	30	22	95	1	9	19
10	48	4	2	55	87	29	14	6	9
66	24	94	93	10	29	23	49	95	67
58	77	58	96	29	10	37	33	2	9
26	41	85	77	23	86	4	78	87	97
46	5	60	92	47	90	1	43	5	88
23	13	44	89	70	91	45	11	88	96
70	26	7	92	8	68	7	38	67	58
95	81	100	37	70	91	46	92	90	78
60	72	68	69	61	72	33	24	23	79

Parallel Computation

30 operations

Operations vs Image resolution



- ReRAM computational array architectures provides magnitudes orders of performance & energy improvement

ReRAM Based IPs & Arrays

ReRAM Computational Memory Arrays & IPs	Usage
Highly Parallel Memory	Classification
Computational Arrays	Matrix Operations Multiply Accumulate Sparse Matrix
Comparison/Evaluation	Matching, Best Fit, statistics
Configurable logic	Configuration bits, FPGA,
Power management	Memory Shadowing
Embedded memory, standalone memory, & OTP	Embedded at Code/Data Memory at advanced nodes

Highly Parallel Memory

Face Recognition with ReRAM

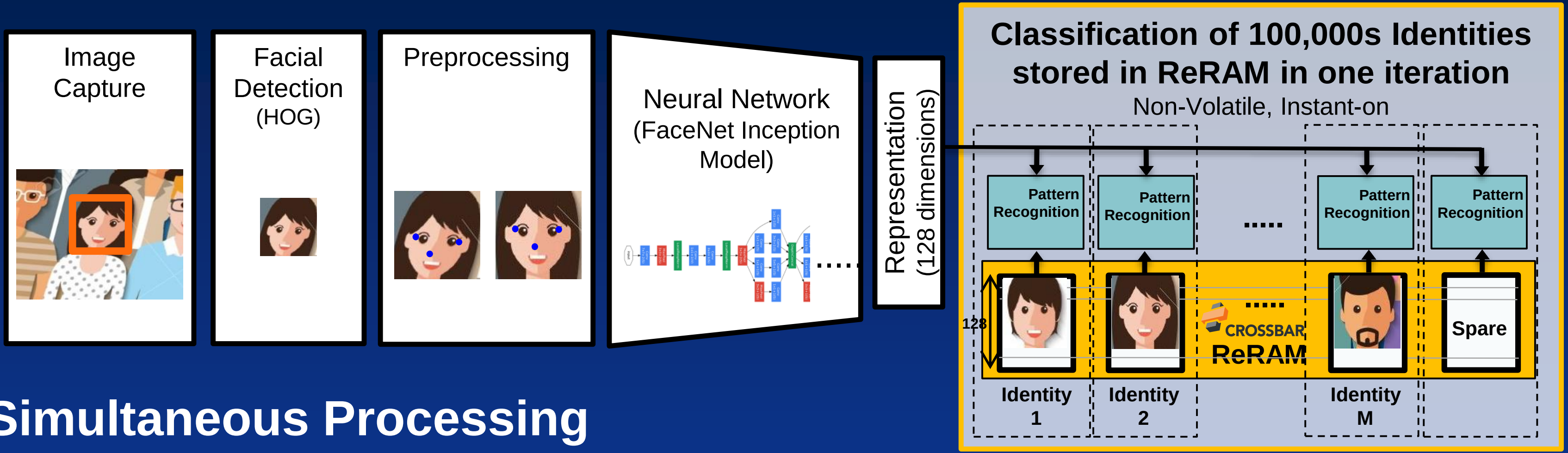


Image Capture

Facial Detection (HOG)

Preprocessing

Neural Network (FaceNet Inception Model)

Representation (128 dimensions)

Classification of 100,000s Identities stored in ReRAM in one iteration

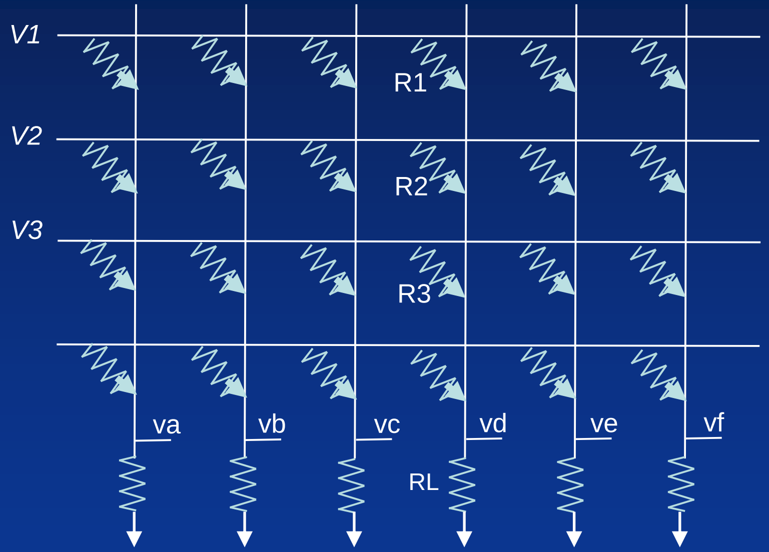
Non-Volatile, Instant-on

Simultaneous Processing with Deterministic Performance

- Parallel comparison against all identities
- If no match, new identity created (learning)
- Classification performed in one cycle independent of number of identities

Computational ReRAM Array Convolution - MAC

ReRAM Computational Array Properties - MAC



- High Bandwidth Multiply Accumulate operations (MAC) are performed with crossbar ReRAM array architectures
- Many MAC operations are simultaneously calculated
- Multiple Word lines are activated simultaneously
- Linear equations are solved with low latency, and low energy consumption

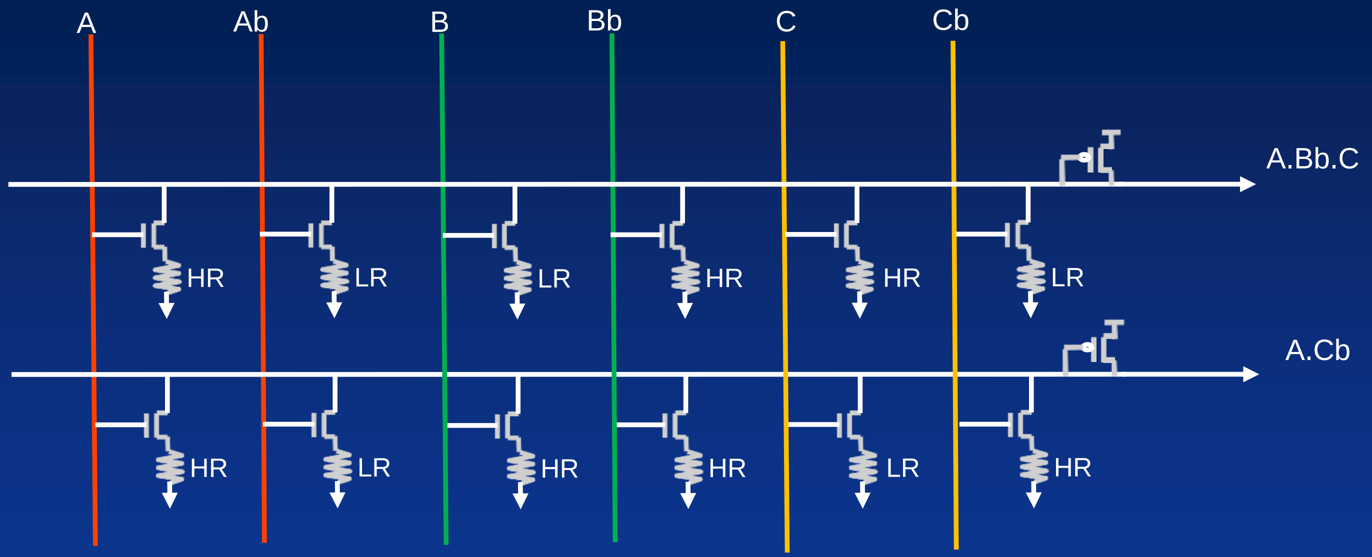
For example:

$$Vd = \left\{\frac{(V1 \cdot Gm1) + (V2.Gm2) + (V3.Gm3)}{Gl.(1 + Gm1 + Gm2 + Gm3)}\right\}$$

where G= $\frac{1}{R}$

Computational ReRAM Array Comparison – Evaluation – Best Fit

ReRAM Computational Array Properties - Comparison

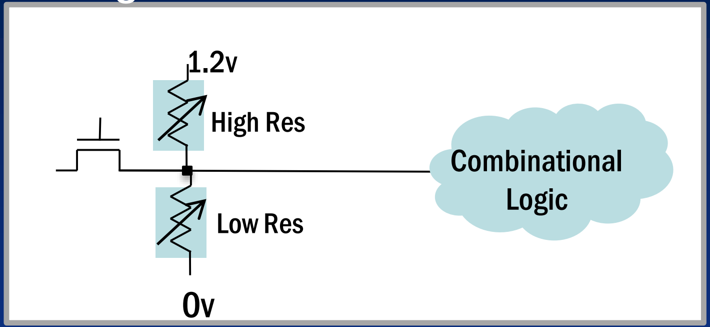


- Large vector comparison/detection (i.e. > 512bits) is performed in few nanoseconds
- Vector evaluation performed within the memory array
- Providing major system energy savings and reduced latency

Configurable Logic – Power management

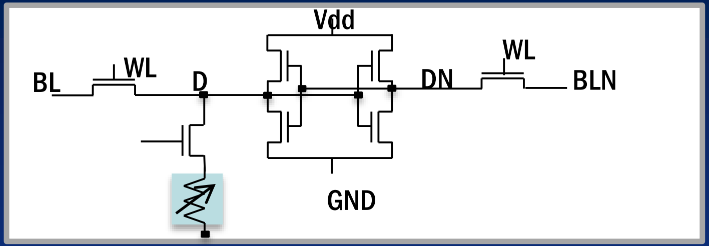
RRAM for FPGA Configuration Bits, NVRAM, State Retainer

Configuration Bit

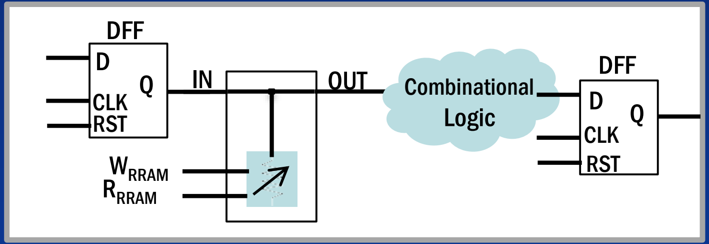


- Instant On
- Eliminates external non-volatile memory
- Security

NVRAM



State Retainer



- Stores data at power down
- Recalls at power up
- Power saving

ReRAM based Computational Arrays and IPs

- ReRAM Technology provides AI architects:
  - Breakthrough computational ReRAM memory arrays with
    - High computation bandwidth and high parallelism
    - Low energy
    - low latency
  - Freedom to architect
  - Monolithic integration with advanced CMOS & FPGAs

Everyone says AI is the key,
but we know the key to AI is
ReRAM
Don’t be left behind

Rethink Artificial intelligence with ReRAM