

**JUNE 23-27, 2024**

MOSCONE WEST CENTER  
SAN FRANCISCO, CA, USA



# ReRAM: Enabling New Low-power AI Architectures in Advanced Nodes

Gideon Intrater

*Weebit Nano*



# Agenda

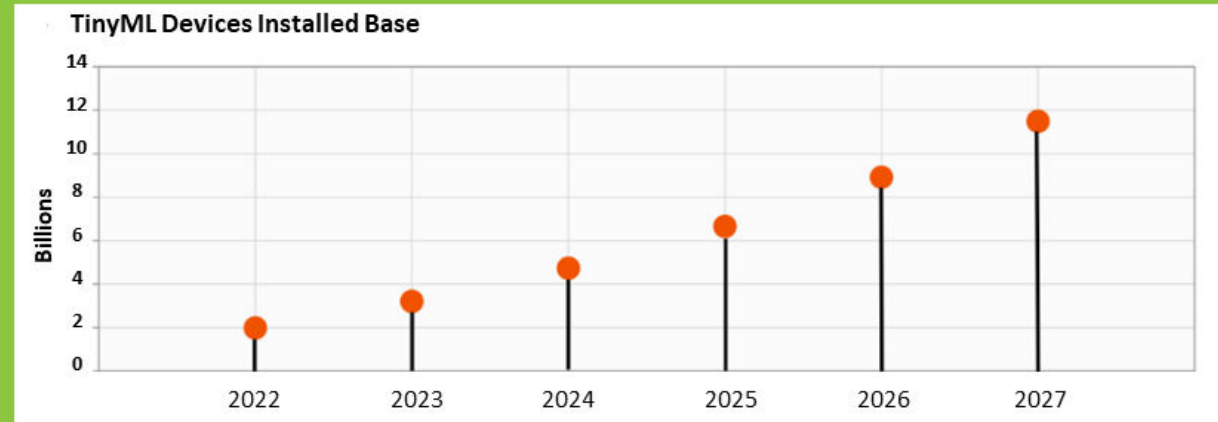
- AI inference at the edge
- Compute Solutions based on Volatile Memories
- Introduction to ReRAM
- ReRAM-based Near-Memory Compute
- ReRAM-based In-Memory Compute
- Conclusions



# AI at the Edge

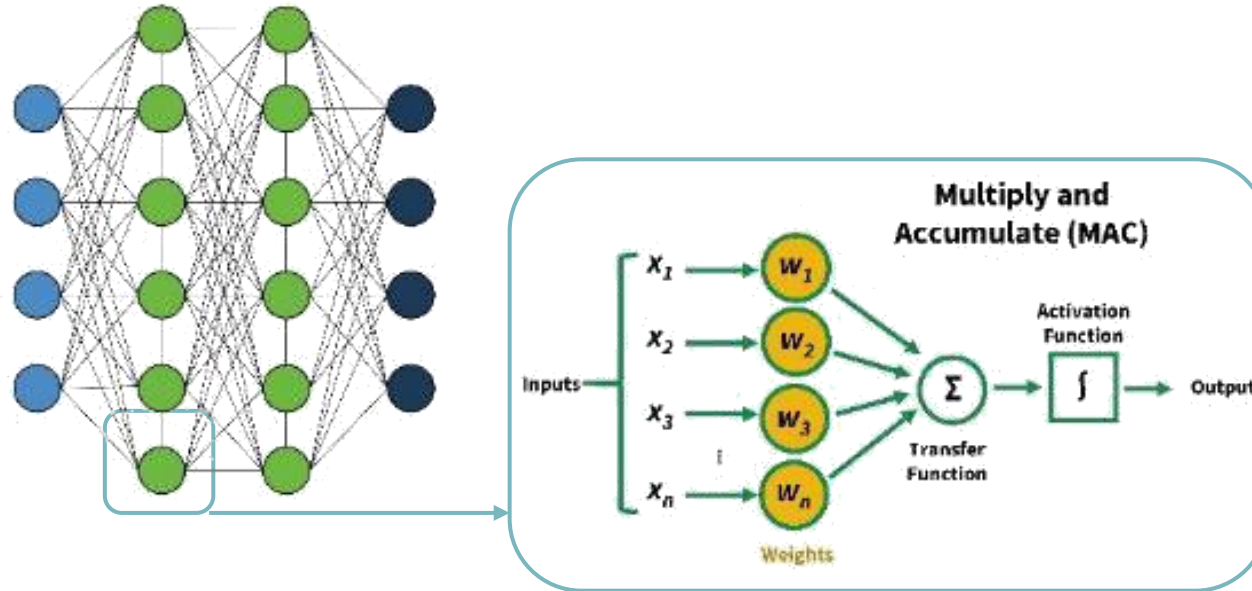
- Traditionally, a significant portion of AI inference has been performed in the cloud
- AI inference is increasingly local (edge), driven by:
  - Power efficiency
  - Ultra-low latency
  - Low bandwidth
  - Security/privacy
  - Smarter products; new applications
- Simple edge AI tasks are handled in software by MCUs
  - Flash-based MCUs implemented at  $\geq 40\text{nm}$  technologies
  - Limited to the simplest algorithms due to power and performance limitations

## Machine learning (ML) in IoT sensors & devices is growing rapidly



Source: <https://go.abiresearch.com/lp-37-technology-stats-you-need-to-know-for-2023>

# AI Inference Basics



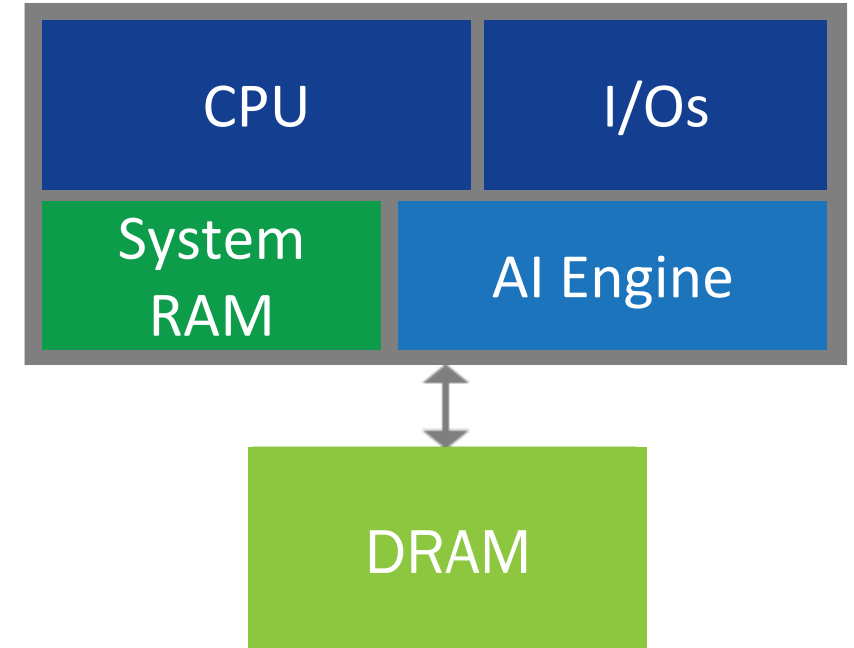
Most of the processing in AI are multiplications of vectors times a fixed weight matrix

- Weight matrix is large, typically up to 10s of MBs
- Weights need to be modified from time to time for updating the algorithms



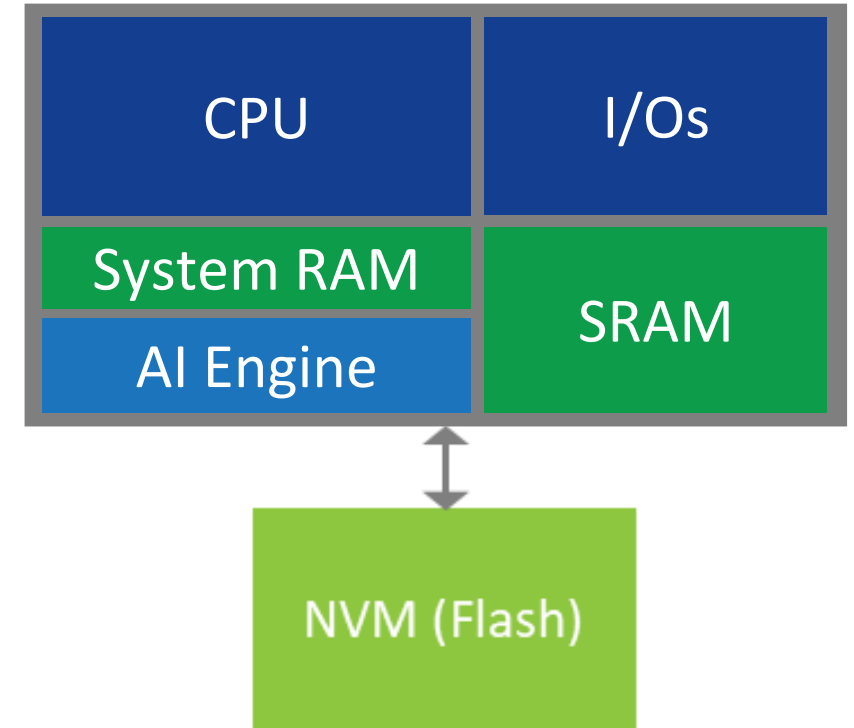
# Building Edge Devices with AI Capabilities

- AI accelerators enable advanced algorithms
- AI engines are complex circuits
  - Require large logic circuits
  - Consume a lot of power
  - Dictate advanced processes, 22nm and below to reduce the impact of the circuits' area and power
- Weights are held in DRAM
  - Computations are performed directly out of DRAM
  - Weights are being fetched from DRAM continuously



# SRAM-based Near-Memory Compute

- At start-up, weights are copied from external NVM to on-chip SRAM
  - Reducing the power and latency associated with frequent DRAM accesses
  - Requires large and costly on-chip SRAM
  - During power-down
    - SRAM must be kept powered – high leakage
    - Alternatively, SRAM must be reloaded upon power-up



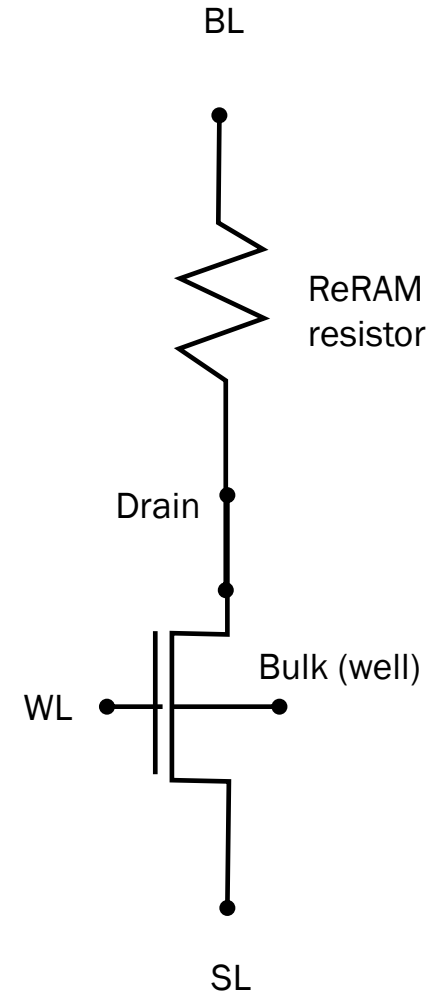
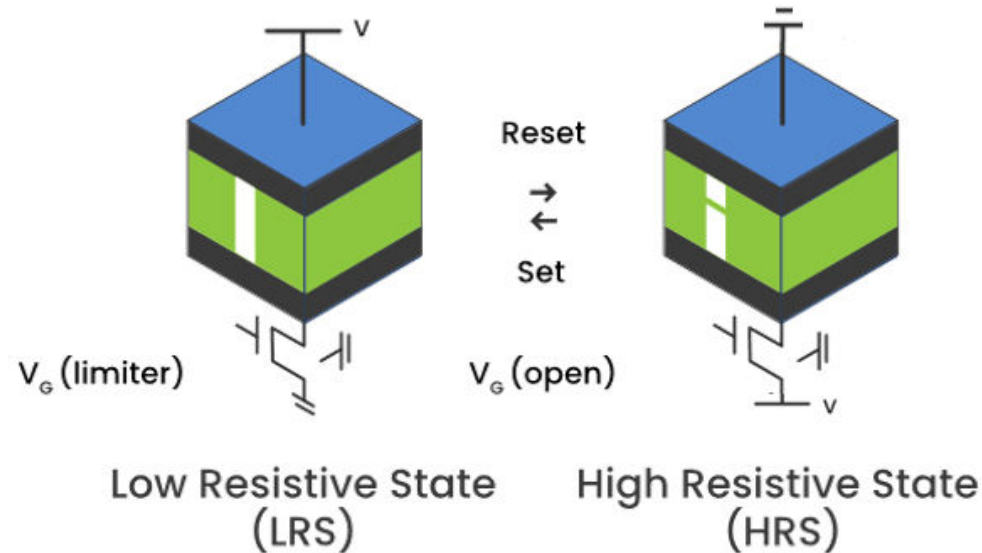
# Introducing ReRAM for Edge AI Applications

# ReRAM Basic Operation

- ReRAM is based on oxygen vacancies filament (OxRAM)
  - By applying different voltage levels on the resistive layer, a filament is created or dissolved
    - RESET (Erase) – Partial dissolution of the Conductive Filament: LRS → HRS
    - SET (Program) – Recreation of the Conductive Filament: HRS → LRS
  - Data retained within the stack is resilient to many environmental conditions

## Low Power Consumption

- ✓ Low read voltage <1V
- ✓ Low write voltage <3V
- ✓ Low currents
- ✓ Zero standby power
- ✓ Fast operation





# The Most Cost-Effective NVM Solution

## 2-mask adder

- Very few added steps compared to other NVM technologies
- Lower wafer cost than competing NVM technologies

## Fab-friendly materials

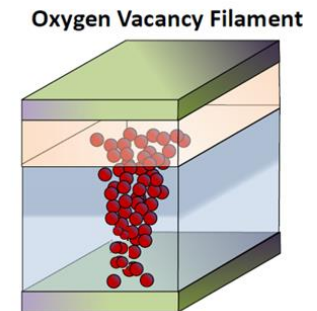
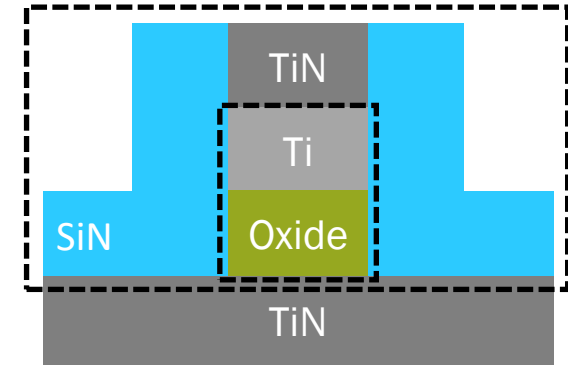
- No contamination risk, No special handling, etc.

## Using existing deposition techniques and tools

- Easy to integrate into any CMOS fab

## BEOL technology

- Stack between any 2 metal layers
- No interference with FEOL – Easier to embed with existing Analog and RF circuits
- Easy to scale from one process variation to another



# ReRAM Advantages



**3-4x**

**Lower added wafer cost**  
**vs. embedded flash**

- ✓ 2-mask adder (vs. ~10)
- ✓ Shorter CT, fewer steps



**10x**

**Better endurance**  
**vs. embedded flash**

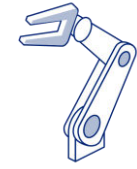
- ✓ Demo 100K-1M write cycles



**~100x**

**More energy efficient**  
**vs. embedded flash**

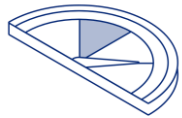
- ✓ Low voltage, low current write operations



**<28nm**

**Scales to processes far**  
**below limits of flash**

- ✓ Proven @ 28nm and 22nm
- ✓ Scalable **below**



**>10x**

**Faster program time**  
**than embedded flash**

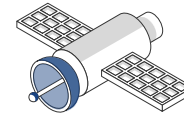
- ✓ Bit/byte addressable
- ✓ No sector erase



**150°C**

**Reliable for**  
**Automotive designs**

- ✓ Grade-0 conditions and profiles



**~350x**

**Better radiation tolerance**  
**vs. flash<sup>1</sup>**

- ✓ Also tolerant to EMI



**0**

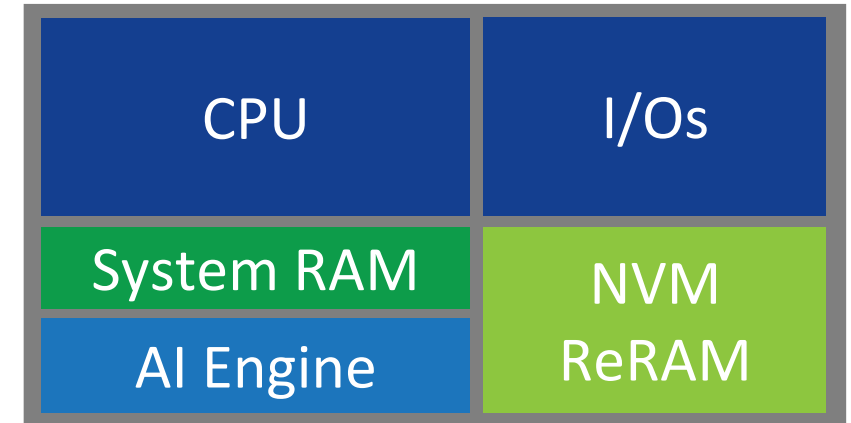
**Interference w/ analog**  
**& power devices**

- ✓ Best NVM for PMIC & mixed-signal

<sup>1</sup> Refers to ReRAM cell array

# ReRAM in Near Memory Edge Inference

- ReRAM is available in 22nm with a roadmap to smaller geometries
- ReRAM can be implemented on the same die as the rest of the MCU and hold:
  - AI weights
  - CPU's firmware
- The resulting MCU will have:
  - Higher AI and CPU performance
  - Lower power; longer battery life
  - Lower cost; reduced SRAM and no external memory
  - Enhanced security
  - Better system integration



# Analog In-Memory Compute Basics

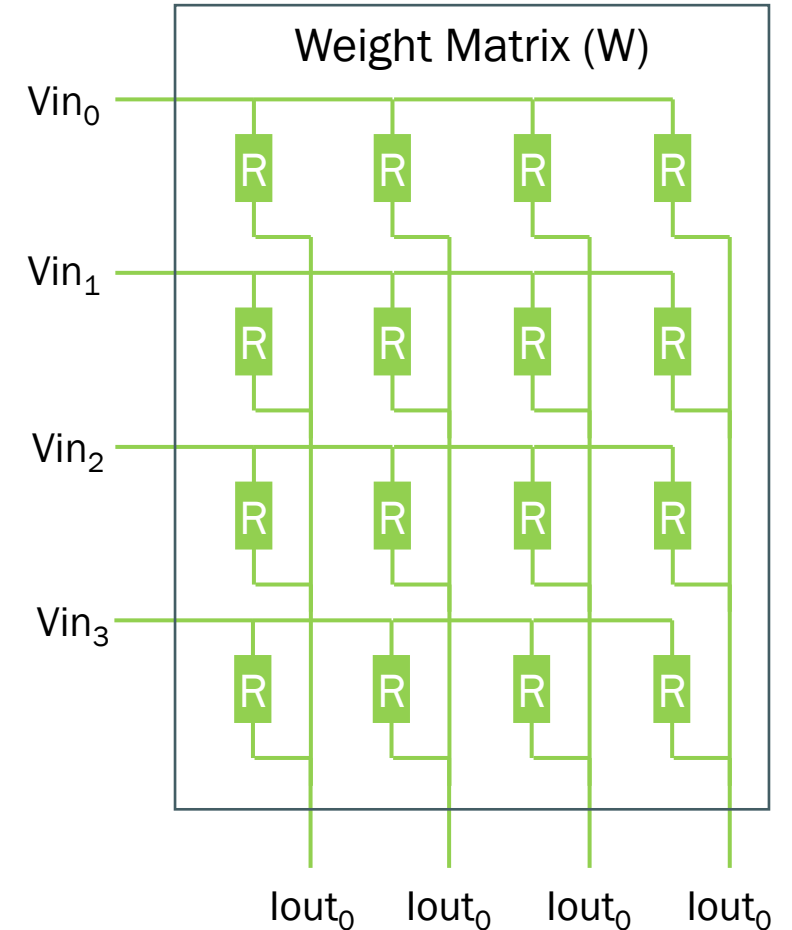
- Most of the processing in AI are multiplications of the input vector times a fixed weight matrix
- Resistor arrays built out of ReRAM elements perform these operations instantly:

- The weights are represented by  $\frac{1}{R}$
- Using Ohm's law each the current through each resistor is:





















$$I_{out} = V_{in} \times \left(\frac{1}{R}\right) = V_{in} \times W$$

- Using Kirchhoff's law, the current in each column is:

$$I_{out_j} = \sum_{j=1}^{j=n} W_{ij} V_{in_j}$$



# How the Various Technologies Compare

	DRAM-based Compute	SRAM-based Near-Memory Compute	ReRAM-based Near-Memory Compute	Analog In-Memory Compute
High performance				
Low power				
Low cost				
Instant-on				
Available today?				



# Summary

- ReRAM-based Near-Memory Compute is superior to DRAM and SRAM-based alternatives
  - Offering better cost, performance, power and always-on availability
- ReRAM-based In-Memory Compute is a promising technology with advantages over all existing technologies



# Thank You!

[www.weebit-nano.com](http://www.weebit-nano.com)



 **Weebitnano**  
THE NEXT NVM IS HERE