RRAM: In-memory Computing from Device x large-scale Integration System Perspectives

CIM Computing-in-memory

PIM Process-in-memory

Types of Memory: SRAM (only Volatile*) Flash, MRAM.
↓
highlight fast write speed

Racetrack extremely high density:
↓
Data is stored in an array of
Magnetic-nanowires
↓
Racetracks

PCM
linear conductance update characteristic

RRAM =
Versatility, High Resistivity (MΩ order of magnitude)

Non-volatile memory technology
↓
higher density < 100 F^2

support of 3D-integration
Stochastic programming

Why ? for matrix-multiplication-dependent neural networks

↳ bcz of Von-Neumann Bottleneck

→ Analog Synapse (VMM → Vector-matrix Multiplication) ↓ in DNN (Deep Neural Network)

→ Neuron Implementation

Von-Neumann Architecture:

Separate CPU from the Storage device.

→ Problem → As Data Volume ↑

↓

Solution

↓

fusing CPU & memory

↓

So that less transportation of Data.

ⓐ High latency ↓ Delay

ⓑ High energy consumption

How we'll implement?

implementation of RRAM device in weight Matrices & in realizing activation functions

↓

then det

↓

then in-memory Computing

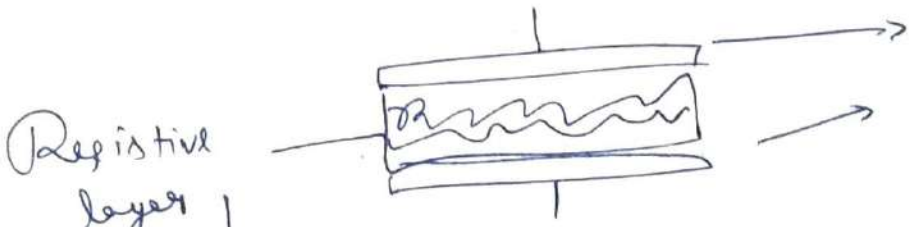
Where ? for DNN acceleration weighting function

RRAM → aka Memristor

↓

two-terminal device → whose Resistance value can be programmed by applying external voltage/current ← with an appropriate Configuration

* Logic



→ Transition Metal oxides

RRAM

Filamentary

↓

relies on

↓

Formation & Dissolution of Conductive filaments

Forming ← operation is required.

→ High voltage required

→ generates initial filaments before Normal use

Interfacial

↓

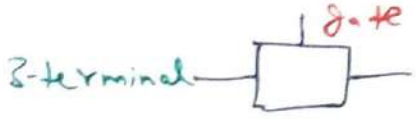
Redistributes the Oxygen vacancies at the heterogeneous interface to change overall Resistance

How to change overall Resistance?

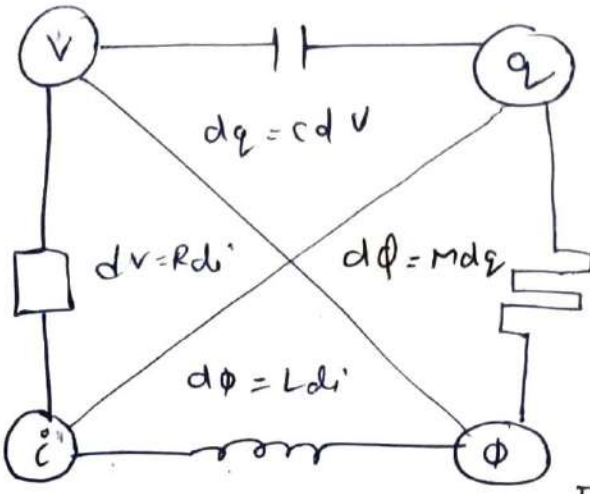
Set Operation : ↑ the device conductance (R → ↓)

Reset " : ↓ (R → ↑)

1T1R → 1 Transistor 1 memristor



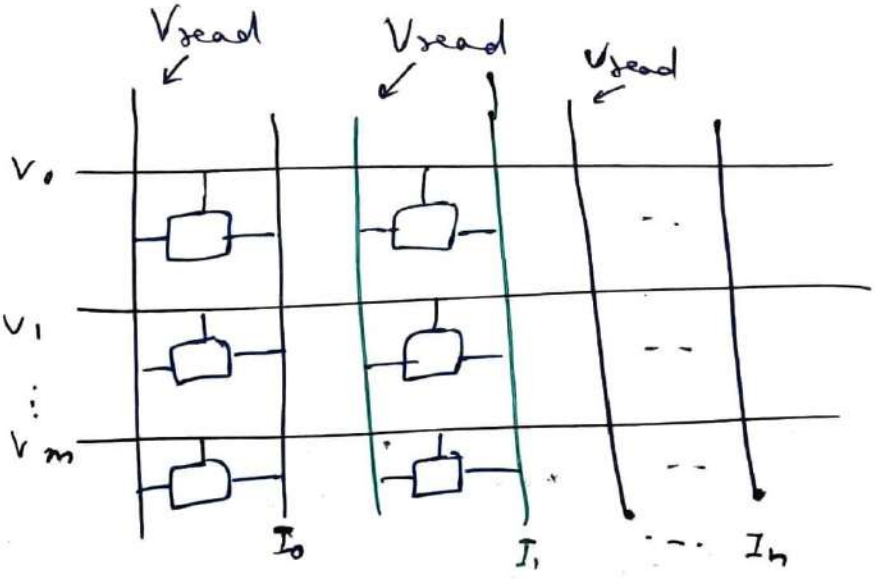
Theoretical Quartet



} Memristor

Main logic of RRAM: Resistance value can be programmed

Example of RRAM Array Topology



Input vector V

output vector I

Cell Conductance Matrix G

Vector - Matrix Multiplication:

$I = V \cdot G$

Analog of RRAM array to VMM, where Synaptic weights are stored as a Conductance Matrix of RRAM array. (*)

RRAM → Non-volatile

Off-computing system is possible.
↓
Power consumption Reduction

Analog
↓
Value programmed to any value b/w HRS & LRS
↓
that is where Analog comes?

Binary
↓
Normal memory device
↓
Stable LRS & HRS

HRS — Highest Resistance State → 1
LRS — Lowest Resistance State → 0
Binary values

Benefits of RRAM: High data Storage as well as Compatability with CMOS *

First Break-through: Logic gate implementation with memory devices

We can make all type of Logic operations (gates)

& more importantly: Material Implication.

How ?

Binary I/Ps are stored into RRAM devices as the Resistance (Conductance) before performing Computation

Sensing voltage is applied

The amplitude of O/P Current defines the logic operation result.

Applying voltage pulses to RRAM device can gradually change its conductance.

Bilogical Synapse vs RRAM Synapse!

SET voltage pulses: used to realize potentiation (excitatory)

RESET " —— —— enable depressing (inhibitory)

What is Synapse? A Structure that permits a neuron (or nerve cell) to pass an electrical signal or chemical signal to another neuron or to the target effector cell.

App: RRAM Synapse + Voltage Spike Generator = Unsupervised Learning Applications

AI learns from data w/o Human supervision.

Also → Weight Representation

- ANALOG → Synaptic weights → Conductive Values → required the device to be programmed to any value in a certain Conductance Range
- DIGITAL → can accomodate only a certain or limited number of Resistance State

RRAM Arrays (1st thing) → Weight Segmentation Technique → for Various Neural Network Models.

Synapses not only transfer information but also process it

Large-Scale System integration:

RRAM Arrays + Peripheral Circuitary = ↑

new → Timing Control & Data Conversion circuitaries.

RRAM-based in Memory computing MACRO design

① ADC/DAC-based

→ computing operates in an Analog format & requires Data Conversion for such a macro to interface with its surrounding Digital Systems.

Why Analog + Digital = MIXED SIGNALS

eg # Pure Analog dot-Product engine DPE using a 128×64 RRAM array.

How? Current changed to voltage by a Trans impedance Amplifier

② Level-Sense-Amplifier-Based Design

Subsequently Changed to Digital O/P vector by ADC

Pure Analog Approach is using the RRAM array & fine-Tune the Analog RRAM cell Resistance

Comparison:

Digital approaches can simplify the data conversion inside in-memory computing macro.

Making → 1T1R cell ⟶ Making array

↓

Fine tune cell Resistance ⟵ for using in Different Applications