IEEE Xplore® Digital Library

ISCAS 2020

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

CAS

# Optimization of Analog Accelerators for Deep Neural Networks Inference

**Andrea Fasoli**, Stefano Ambrogio, Pritish Narayanan, Hsinyu Tsai,
Charles Mackin, Katherine Spoon, Alexander Friz, An Chen, Geoffrey W Burr

*IBM Research - Almaden*

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

# Objectives

- Iso-accuracy of analog hardware implementations compared to software
- Improved energy efficiency and throughput per area over conventional architectures

IEDM 2019

**Reducing the Impact of Phase-Change Memory Conductance Drift on the Inference of large-scale Hardware Neural Networks**

S. Ambrogio, M. Gallot, K. Spoon, H. Tsai, C. Mackin, M. Wesson, S. Kariyappa, P. Narayanan, C.-C. Liu*, A. Kumar**, A. Chen, and G. W. Burr

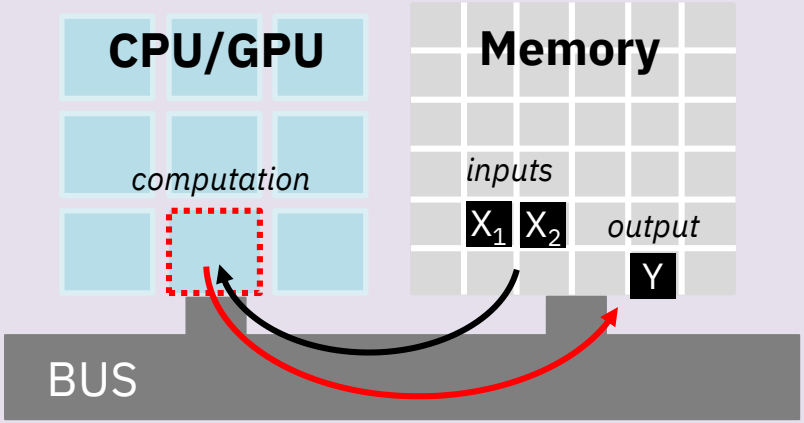
VLSI 2019

**Inference of Long-Short Term Memory networks at software-equivalent accuracy using 2.5M analog Phase Change Memory devices**

H. Tsai, S. Ambrogio, C. Mackin, P. Narayanan, R. M. Shelby, K. Rocki, A. Chen and G. W. Burr

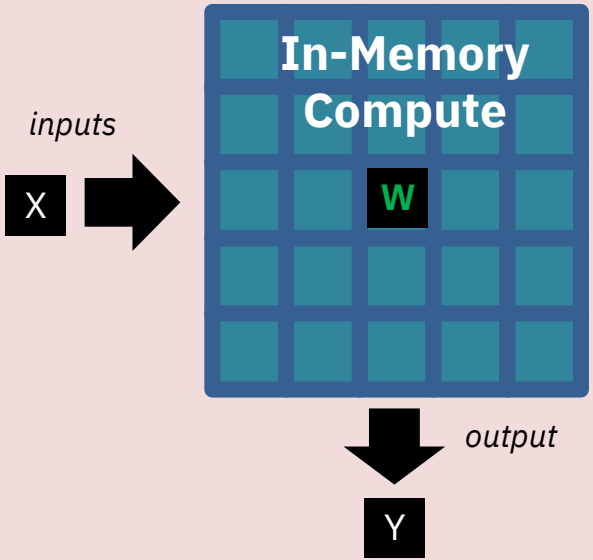
# Accelerators Architecture

## Digital Accelerators

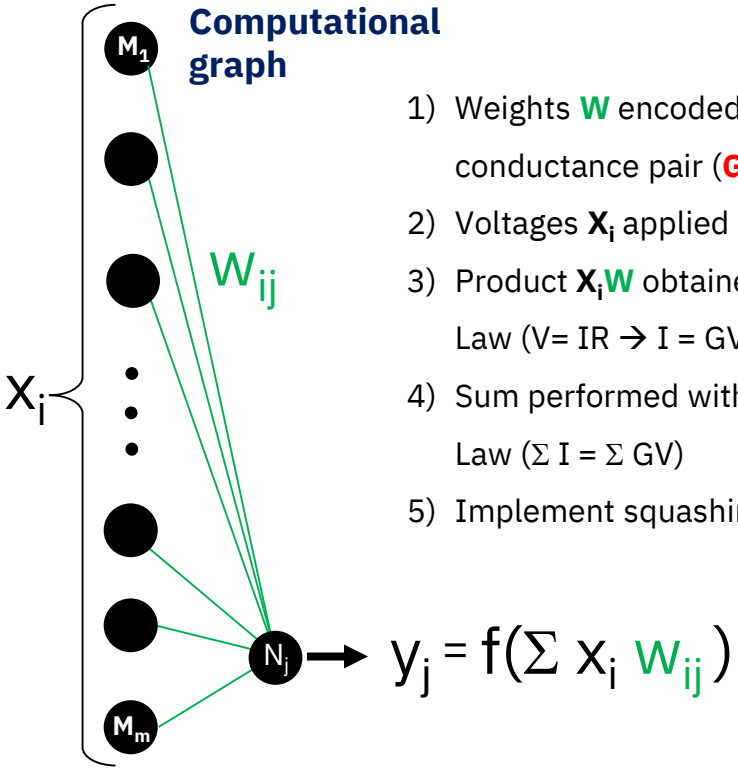



- ✕ Consumes high energy in data movement
- ✕ Bus has limited bandwidth & can be a bottleneck

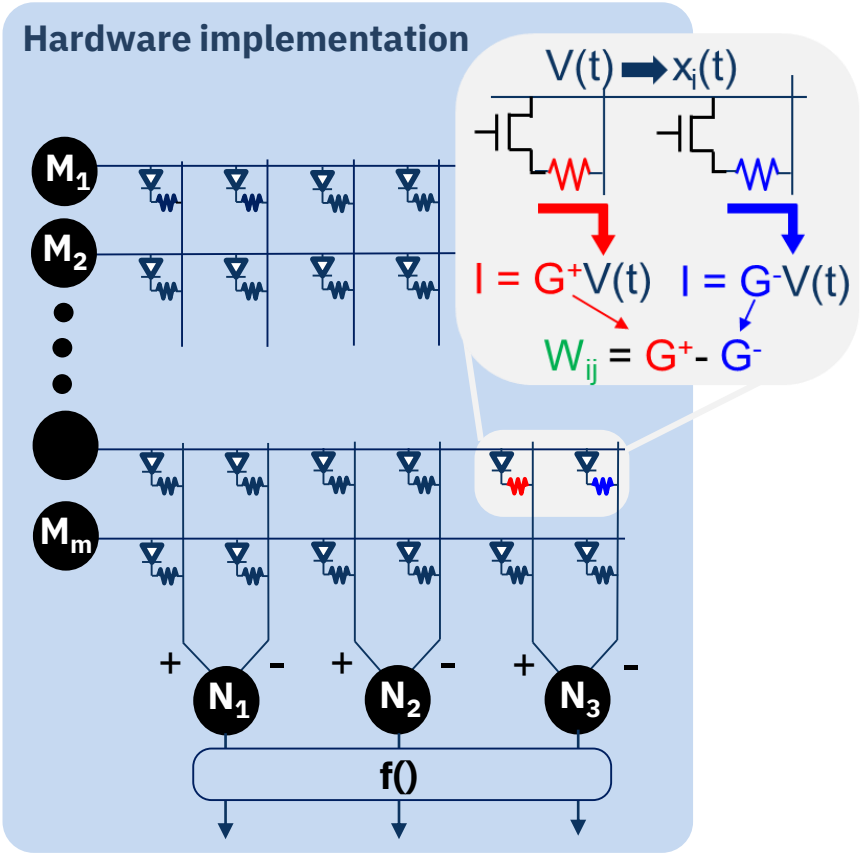
## Analog Accelerators



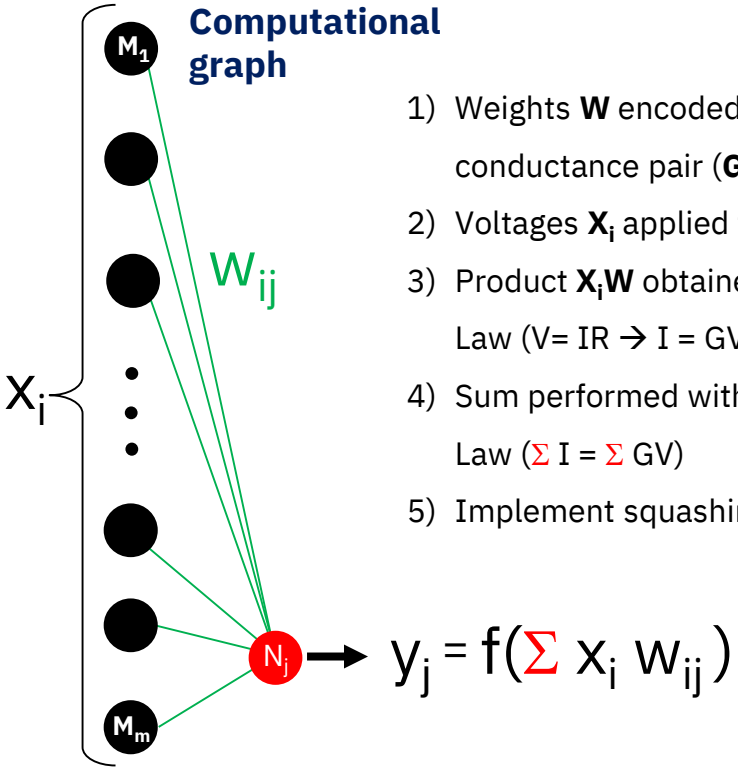
# Mapping Multiply-Accumulate to Analog Hardware



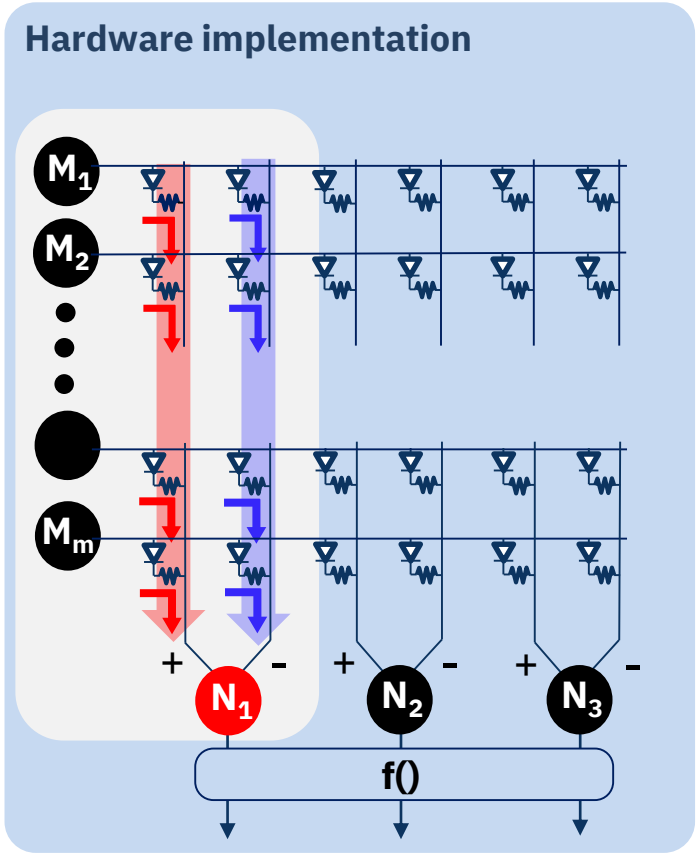

1) Weights **W** encoded with a conductance pair (**G⁺**, **G⁻**)
2) Voltages $\mathbf{X_i}$ applied to every row
3) Product $\mathbf{X_i W}$ obtained by Ohm's Law (V= IR → I = GV)
4) Sum performed with Kirchhoff's Law (Σ I = Σ GV)
5) Implement squashing function **f**



# Mapping Multiply-Accumulate to Analog Hardware

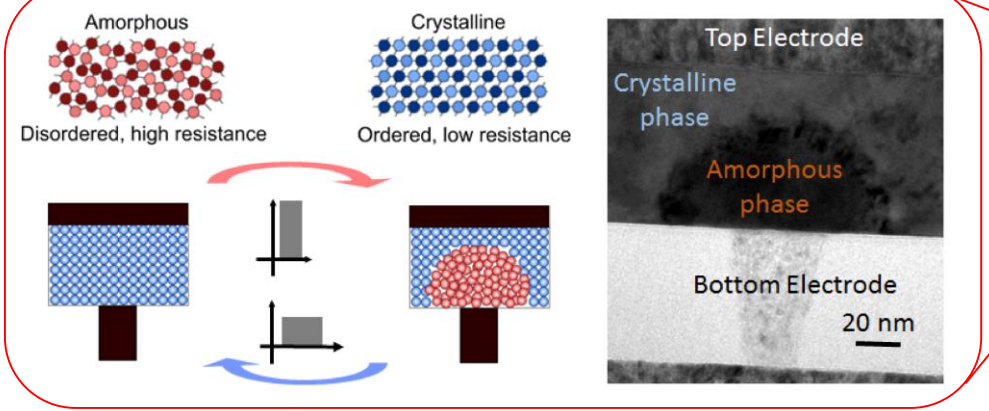


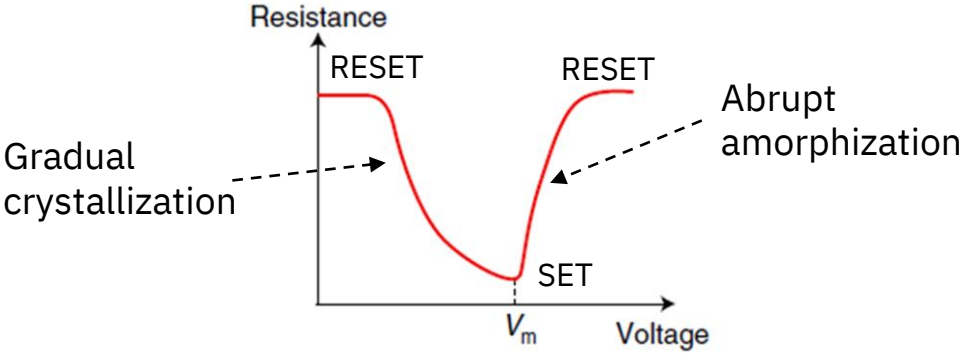

1) Weights **W** encoded with a conductance pair (**G⁺, G⁻**)
2) Voltages $\mathbf{X_i}$ applied to every row
3) Product $\mathbf{X_iW}$ obtained by Ohm's Law (V= IR → I = GV)
4) Sum performed with Kirchhoff's Law (Σ I = Σ GV)
5) Implement squashing function **f**



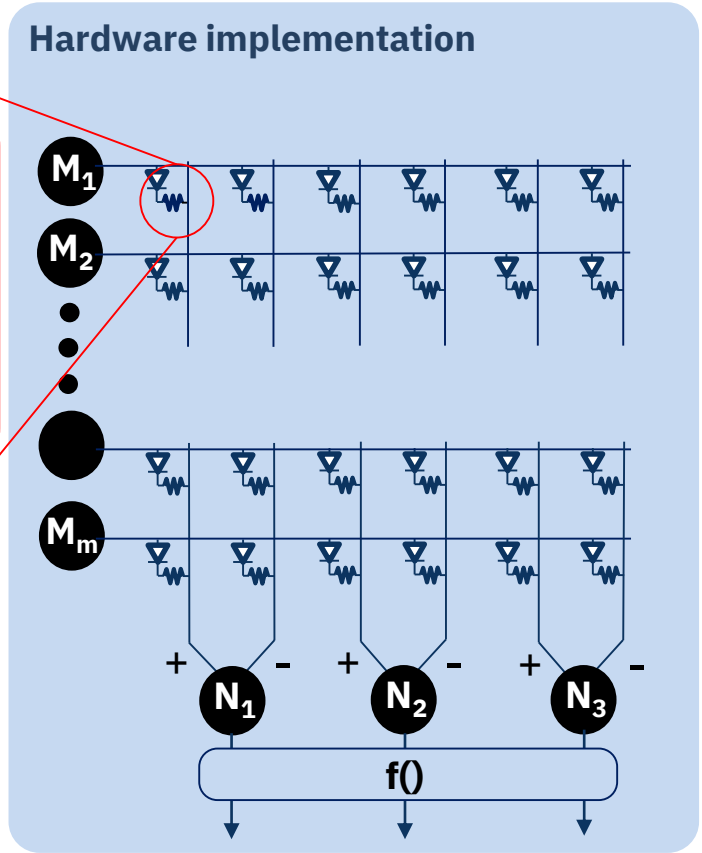
# Phase Change Memory (PCM)

## Mushroom Cell PCM



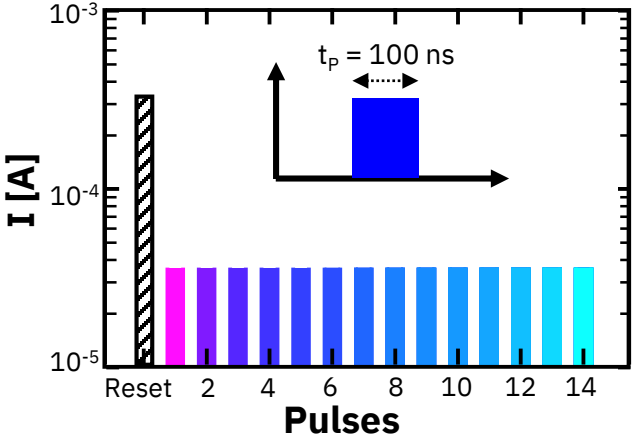



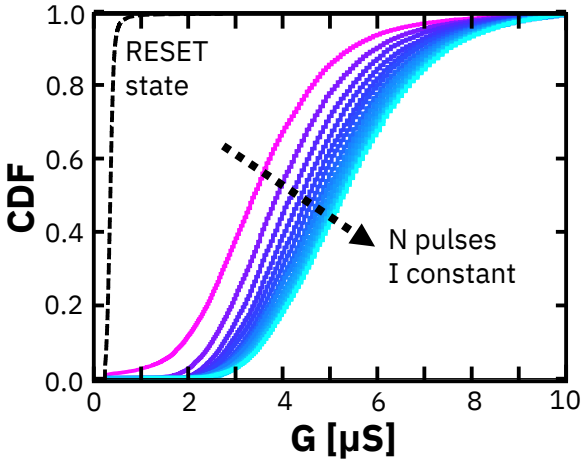

## Hardware implementation

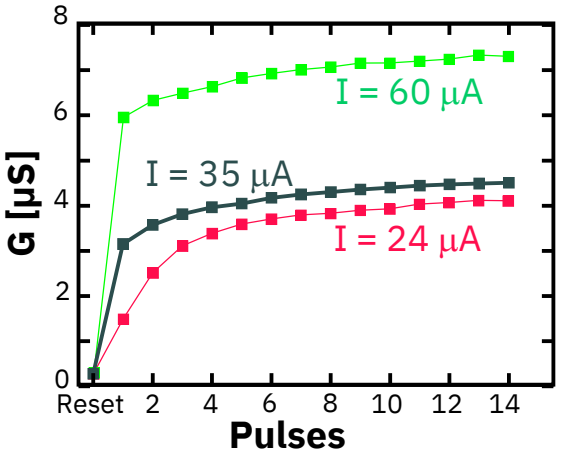


# Programming analog G states in PCM

- PCM devices programmed using train of pulses
- Current kept constant across SET pulses
- Range of accessible conductances is limited

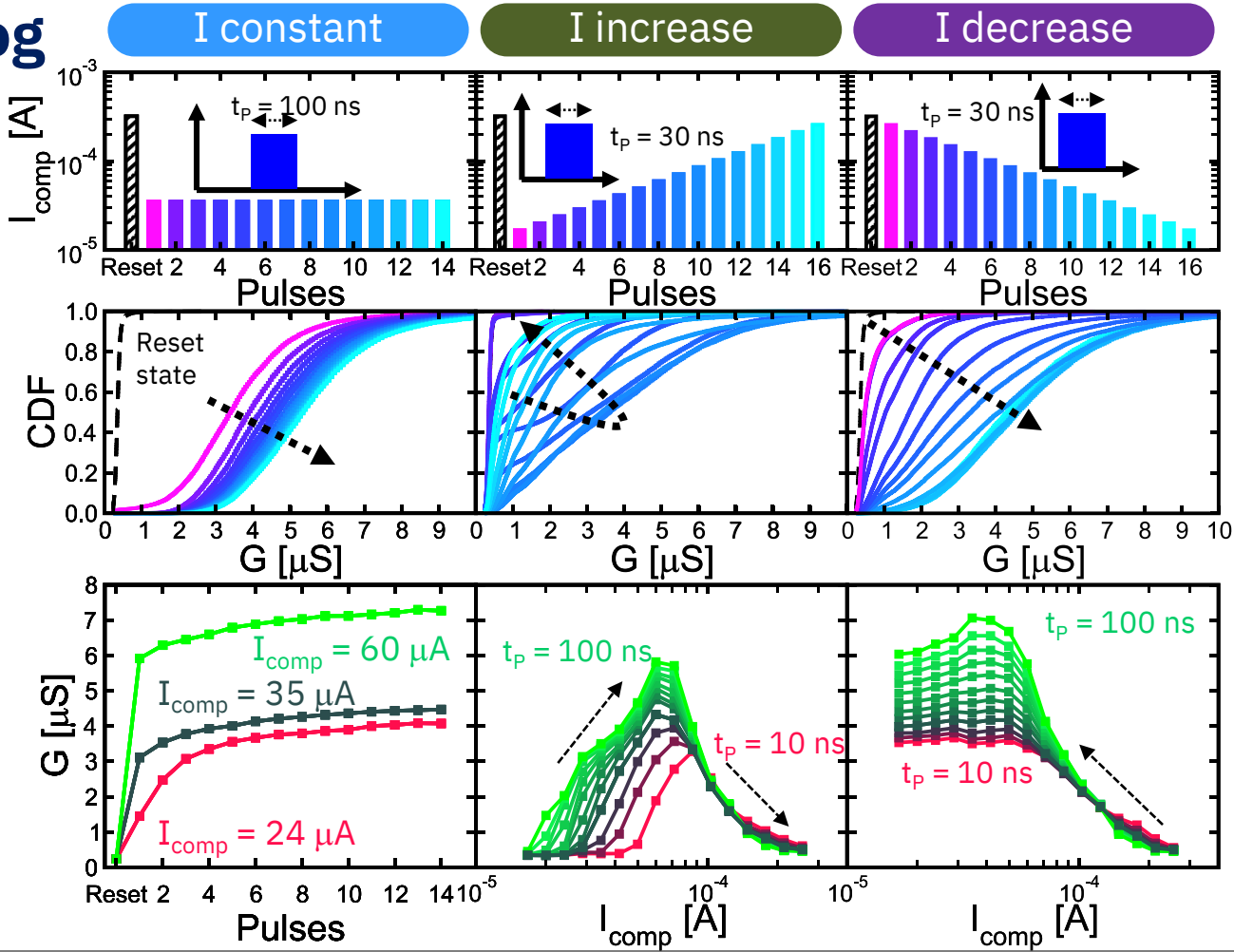


# Programming analog G states in PCM

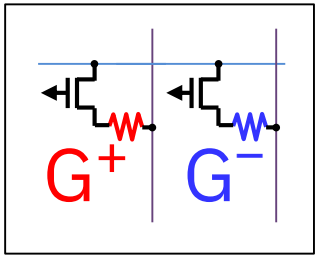
- Three different current programming schemes
  - Constant Current
  - Increase Current
  - Decrease Current
- Varying pulse duration extends dynamic range
- Decreasing current provides most desirable convergence trend toward analog G states




Ambrogio et al., IEDM 2019

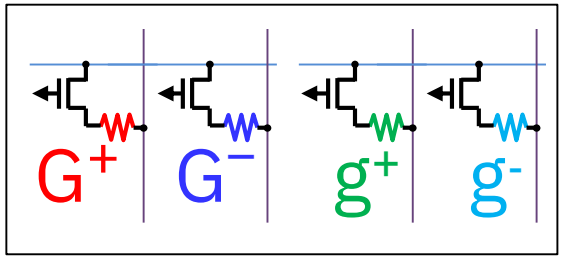
# Full 4-Analog Memory structure

## 2-PCM cell



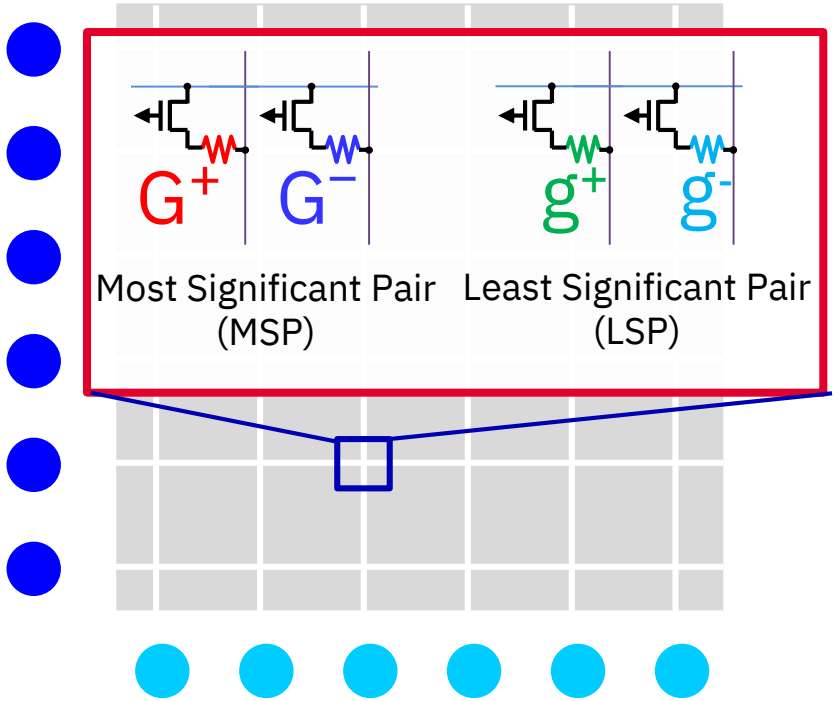

$W = G^{+} - G^{-}$

## 4-PCM cell



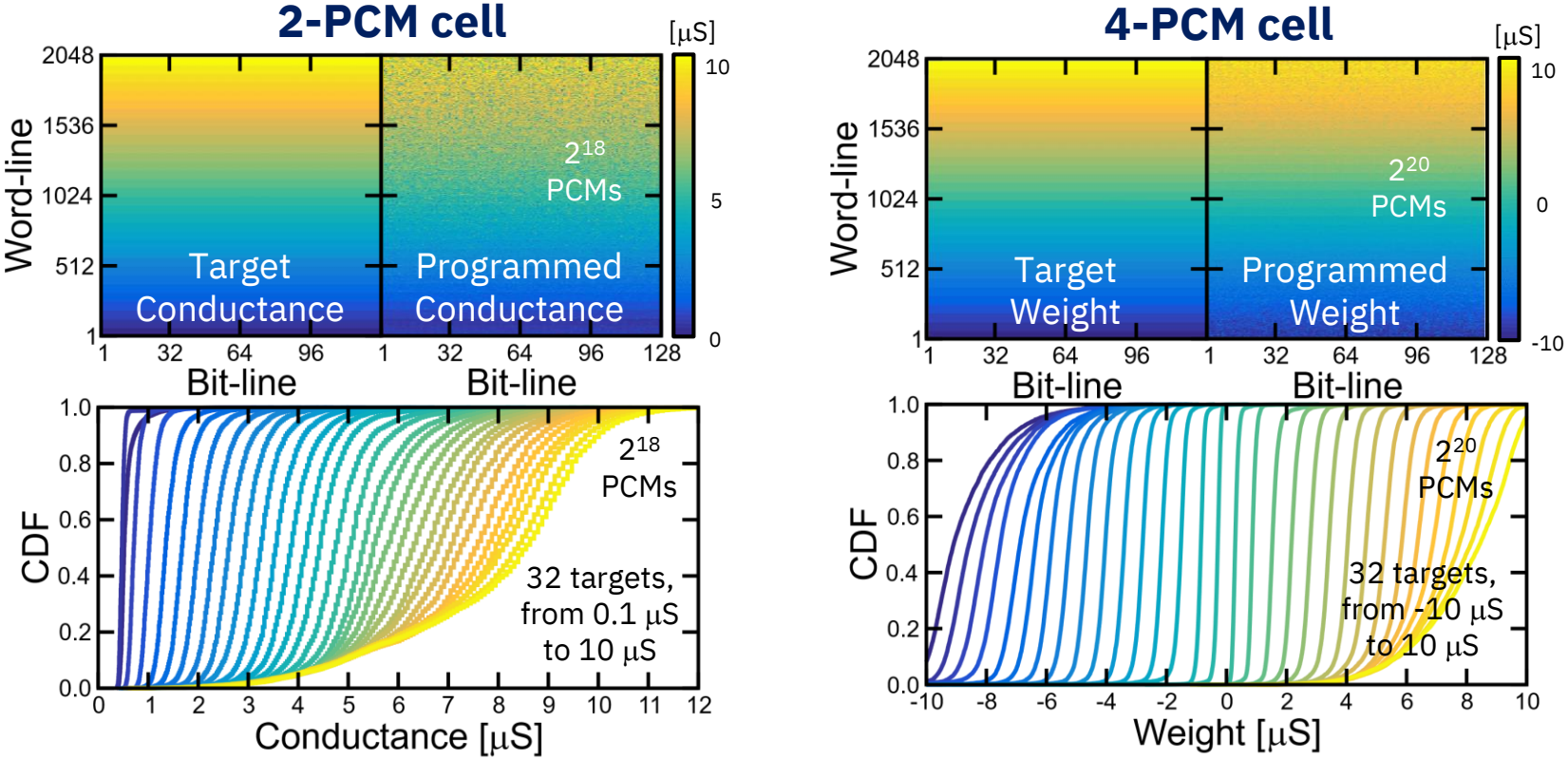

$W = G^{+} - G^{-} + (g^{+} - g^{-}) / F$

## Inference chip



# Writing Analog Weights in PCM

Tsai et al., VLSI 2019

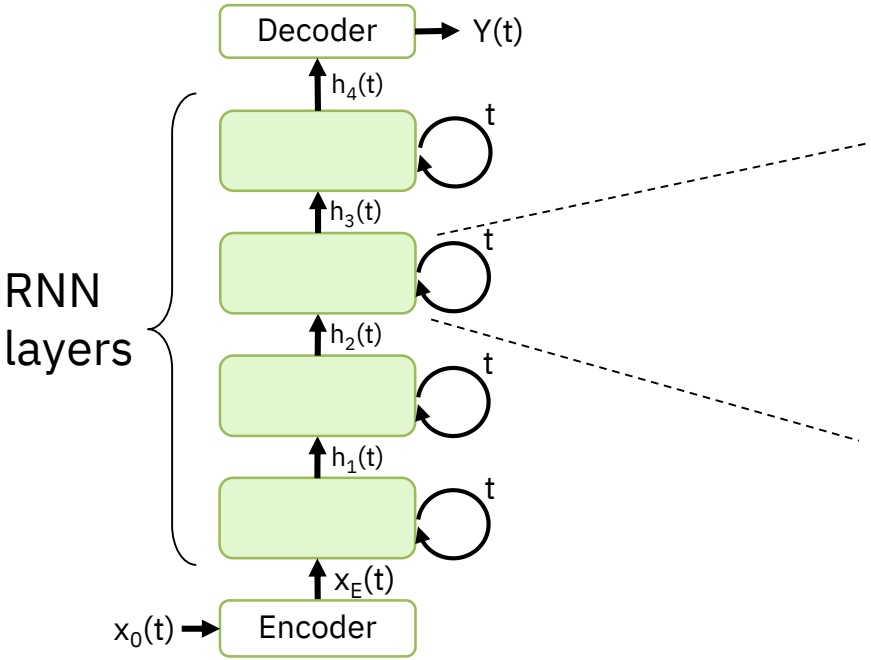



4-PCM cell design provides better resilience to **write noise** and **conductance saturation**

# Optimized design and operation on an NLP task

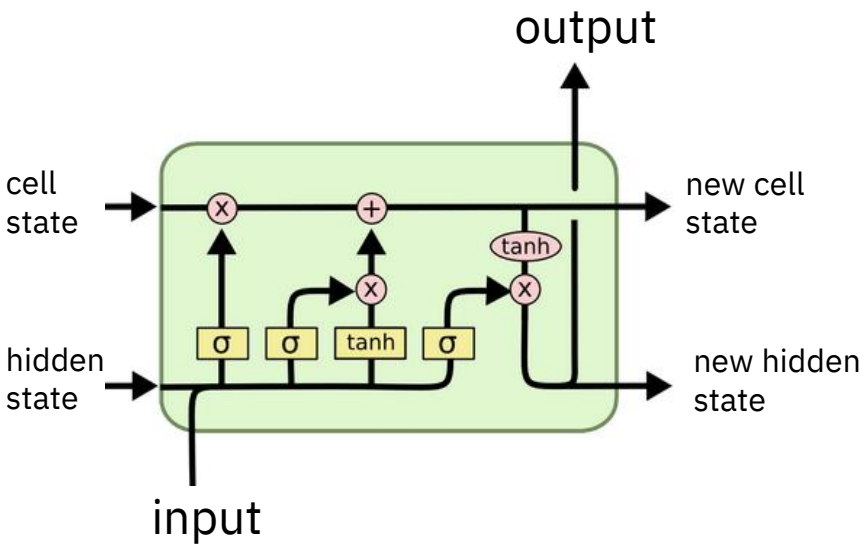
## Recurrent Neural Networks

Input processed over many time steps



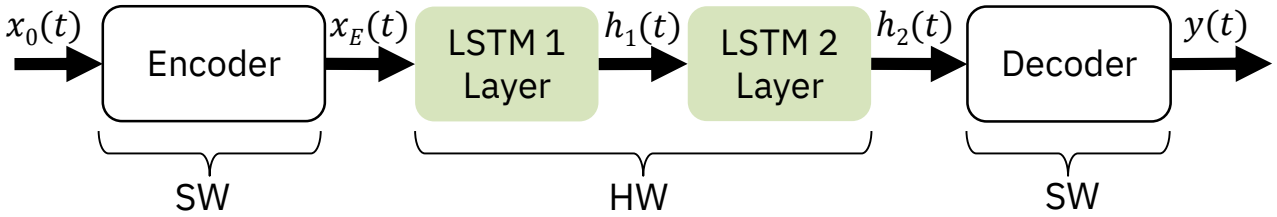

## Long Short-Term Memory (LSTM) cell

Can propagate information over many time steps

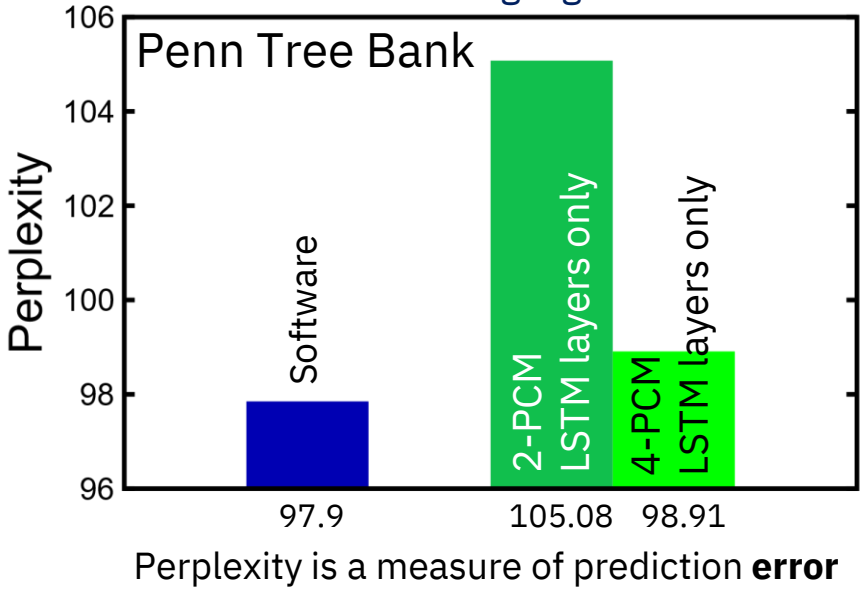


# Software-Equivalent Accuracy on LSTM/PTB

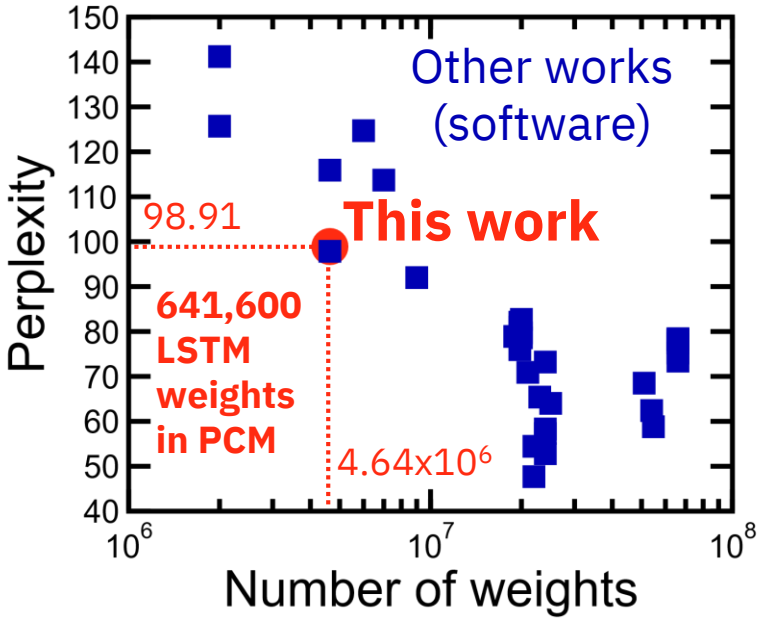
Tsai et al., VLSI 2019




Word-level language model




Perplexity is a measure of prediction **error**



# Conclusions

- PCM programming scheme based on progressively decreasing current in subsequent pulses improves control on PCM conductance
- 4-PCM design improves resilience to write noise and conductance saturation
- Software-equivalent accuracy achieved on 2-layer LSTM and word-level language model on Penn Treebank

# Thank you!

andrea.fasoli@ibm.com