Thinker-IM: An Energy-Efficient Mixed Signal RNN Engine with Computing-in-Memory Techniques and Predictive Execution

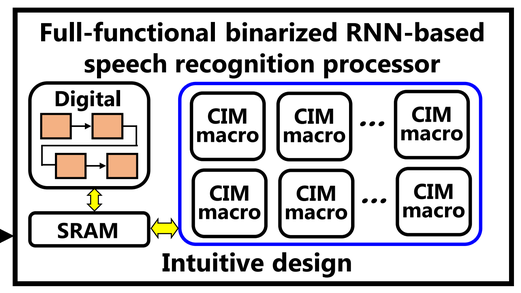
Ruiqi Guo[1], Yonggang Liu[1], Shixuan Zheng[1], Ssu-Yen Wu[2], Peng Ouyang[3], Win-San Khwa[2], Xi Chen[1],Jia-Jing Chen[2], Xiudong Li[3], Leibo Liu[1], Meng-Fan Chang[2], Shaojun Wei[1], Shouyi Yin[1]*

[1]Tsinghua University, Beijing; [2]National Tsing Hua University, Hsinchu; [3]TsingMicro Tech, Beijing; *yinsy@tinghua.edu.cn

Motivation

To achieve the ultra-low energy:
1: Binary neural networks;
2: Computing-in-memory techniques.

Bottleneck of conventional digital implementation → Computing-in-Memory techniques → *Integrate*



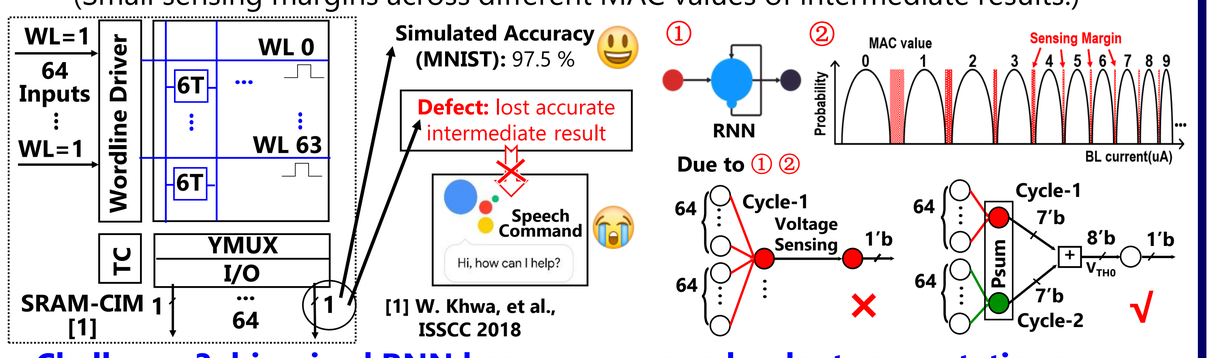
Challenge 1: Requirement for multi-CIM architecture and dataflow design

A. Sufficient memory capacity is needed to fix all weights.
B. A big SRAM-CIM macro will mismatch the network shapes and computing resources.
(Weights stored in the same WL need to be multiplied by the same input; Accumulated currents through the same BL contribute to the same output.) [1]

- **SRAM-CIM unit-macro is needed to constitute multi-CIM architecture.**
- **Dedicated computing flow is needed to schedule multiple SRAM-CIM macros.**

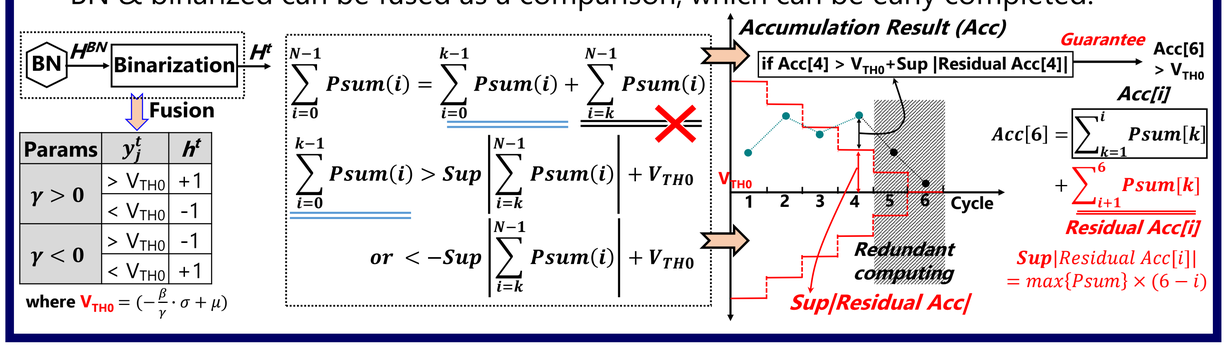
Challenge 2: Requirement for multi-bit output SRAM –CIM design

A. RNNs are sensitive to bit-precision of intermediate results.
(the current iteration is determined by the current input and previous hidden state.)
B. RNNs should be partitioned to generate partial sums (Psums).
(Small sensing margins across different MAC values of intermediate results.)



Challenge 3: binarized RNN has numerous redundant computations

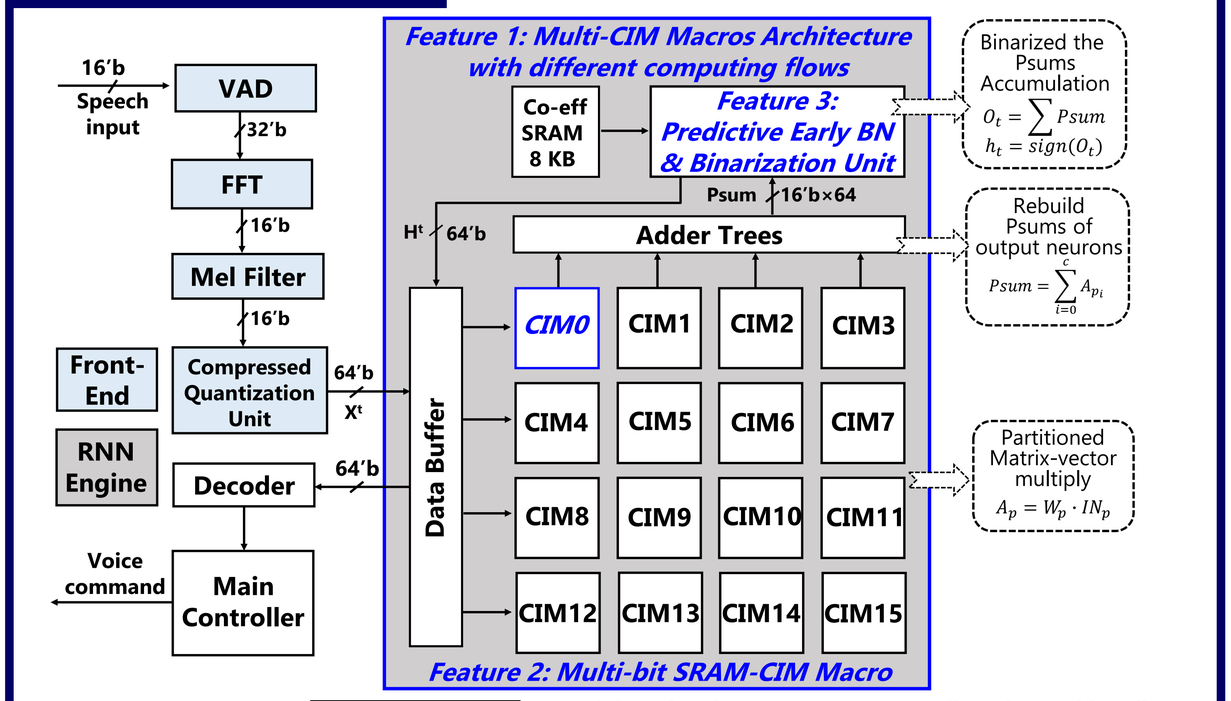
- BN & binarized can be fused as a comparison, which can be early completed.



Params	y_j^t	h^t
$\gamma > 0$	> V_{TH0}	+1
	< V_{TH0}	-1
$\gamma < 0$	> V_{TH0}	-1
	< V_{TH0}	+1

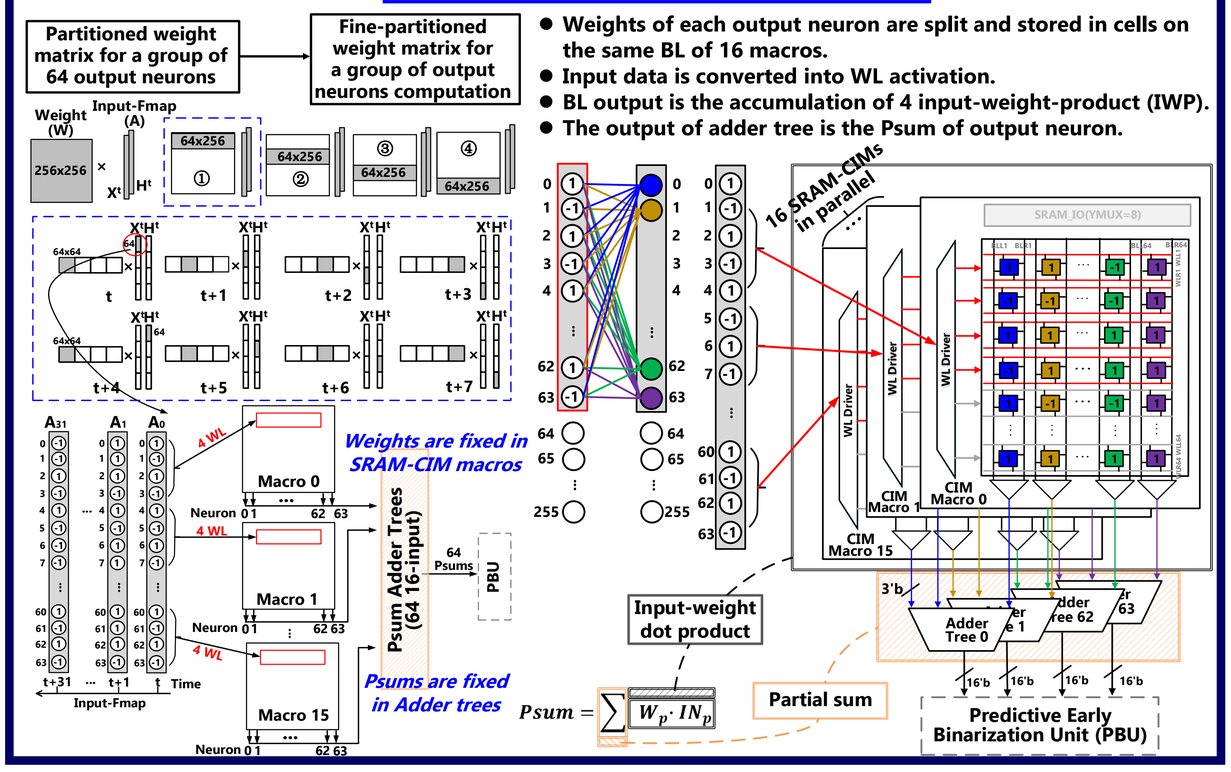
where $V_{TH0} = (-\frac{\beta}{\gamma} \cdot \sigma + \mu)$

Architecture



Partitioned weight matrix for a group of 64 output neurons → Fine-partitioned weight matrix for a group of output neurons computation

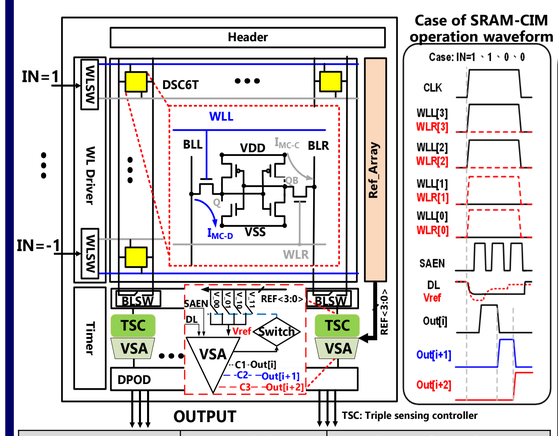
- Weights of each output neuron are split and stored in cells on the same BL of 16 macros.
- Input data is converted into WL activation.
- BL output is the accumulation of 4 input-weight-product (IWP).
- The output of adder tree is the Psum of output neuron.



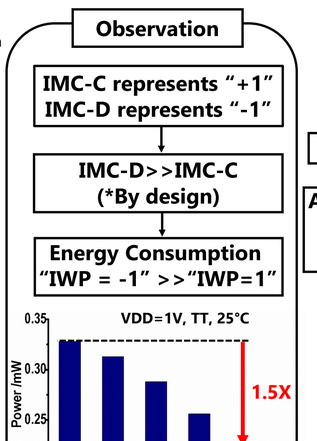
Low Power Techniques

1. SRAM-CIM Macro Design and CIM-aware Weights Adaptation

Key features of SRAM-CIM macro:
A. Dual-split-control 6T memory cell to achieve XNOR; B. Serial-phase triple sensing controller to support 3-b output



Input			Weight			Sum of Product		
Value	WLL	WLR	Value	Q	Q_B	Value	BLL	BLR
+1	1	0	-1	0	1	-1	IMC-D	0
+1	1	0	+1	1	0	+1	IMC-C	0
-1	0	1	-1	0	1	+1	0	IMC-C
-1	0	1	+1	1	0	-1	0	IMC-D



2. Predictive early BN & Binarization Mechanism and Processing Unit

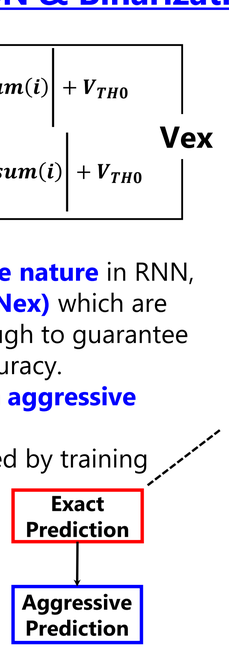
- **Exact Prediction**

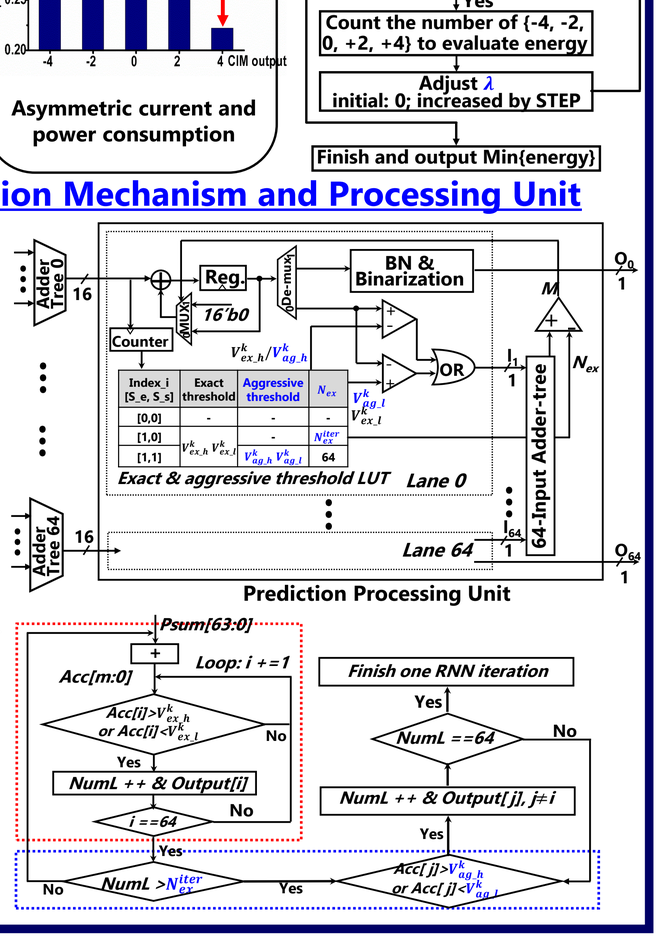
$$\sum_{i=0}^{k-1} Psum(i) > \left| Sup \left| \sum_{i=k}^{N-1} Psum(i) \right| + V_{TH0} \right| \quad or \quad < \left| -Sup \left| \sum_{i=k}^{N-1} Psum(i) \right| + V_{TH0} \right| \quad \text{Vex}$$

- **Aggressive Prediction**

A. Due to the **fault-tolerate nature** in RNN, **a portion of Neurons (Nex)** which are exactly predicted is enough to guarantee the final recognition accuracy.
B. Vex can be relaxed to an **aggressive threshold (Vag)**
(Nex and Vag are determined by training offline)

Predictive early BN & Binarization Mechanism



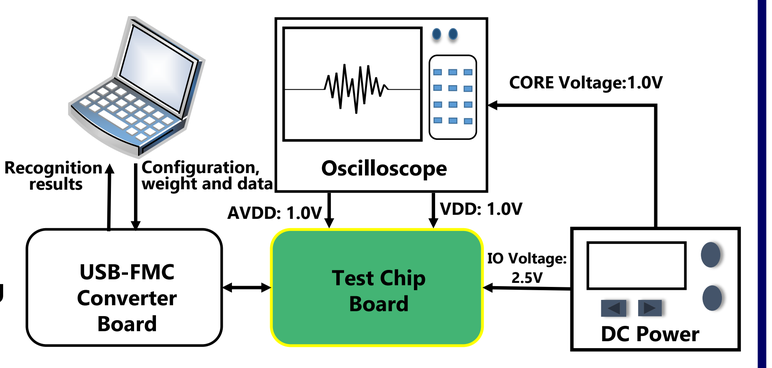


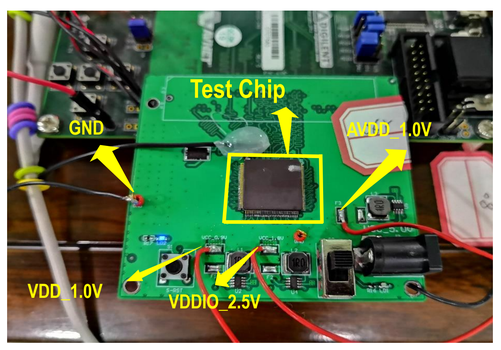
Verification

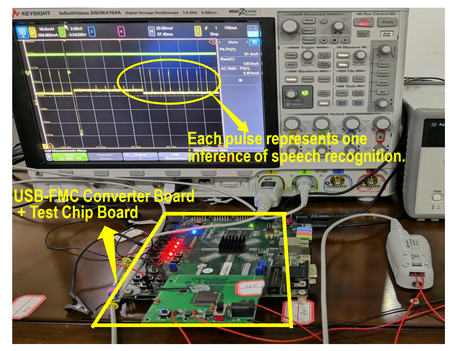
Demonstration System

Test chip communicates with PC
- neural network weights, BN parameters, *etc*;
- testing data;
- configurations;
- recognition result.

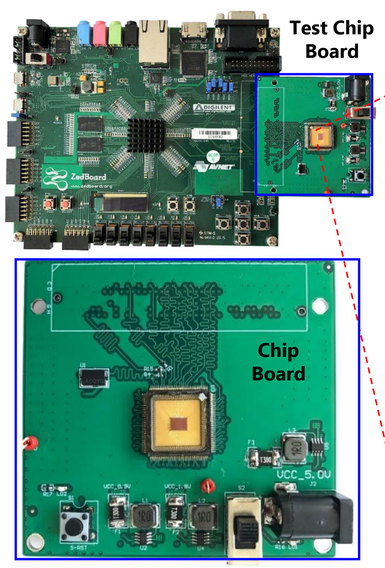
Oscilloscope measures the working current

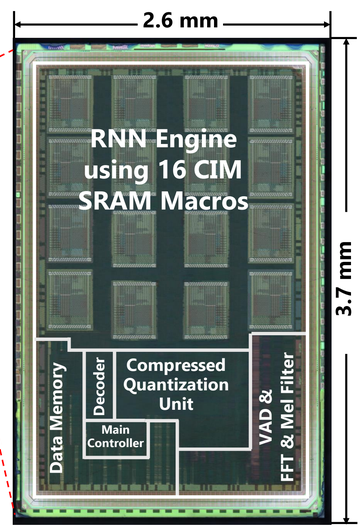






Chip Photograph





Chip Summary:

- Process: 65 nm CMOS
- Supply Voltage: 0.9 – 1.1 V
- Frequency: 5 – 75 MHZ
- Core Size: 3.1×2 mm^2
- Die Size: 3.7 ×2.6 mm^2
- Neural Energy Efficiency: 5.1 pJ/Neuron @0.9 V, 75 MHZ
- Arithmetic Energy Efficiency: 11.7 TOPS/W @0.9 V, 75 MHZ

Key Features:

A. Multiple SRAM-CIM architecture
B. Muti-bit output SRAM-CIM
C. Low-current training flow for SRAM-CIM architecture
D. Predictive early BN and binarization method