A 5.1pJ/Neuron 127.3us/Inference RNN-based Speech Recognition Processor using 16 Computing-in-Memory SRAM Macros in 65nm CMOS

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@tsinghua.edu.cn

Abstract

This work presents a 65nm CMOS speech recognition processor, named Thinker-IM, which employs 16 computing-in-memory (SRAM-CIM) macros for binarized recurrent neural network (RNN) computation. Its major contributions are: 1) A novel digital-CIM mixed architecture that runs an output-weight dual stationary (OWDS) dataflow, reducing 85.7% memory accessing; 2) Multi-bit XNOR SRAM-CIM macros and corresponding CIM-aware weight adaptation that reduces 9.9% energy consumption in average; 3) Predictive early batch-normalization (BN) and binarization units (PBUs) that reduce at most 28.3% computations in RNN. Measured results show the processing speed of 127.3us/Inference and over 90.2% accuracy, while achieving neural energy efficiency of 5.1pJ/Neuron, which is 2.8× better than state-of-the-art.

Introduction

Ultra-low energy consumption is critical for speech recognition processor to improve battery's lifetime of mobile and wearable devices. Recently, computing-in-memory (CIM) techniques show great potentials in low-energy neural network computation. However, previous works [1, 2] focus on macro-level CIM design and only demonstrate the basic computations in CONV and FC layers. When integrating SRAM-CIM techniques in a full-functional RNN-based speech recognition processor, some challenges exist (Fig. 1). 1) Since the weights are fixed on several SRAM-CIM macros, a dedicated computing dataflow and corresponding architecture is required for CIM-based RNN engine. 2) Since RNNs are sensitive to bit precision of intermediate results, SRAM-CIM macros that supports multi-bit bit-line outputs are required. 3) Since RNN iterations are quite time-consuming, some unnecessary operations need to be eliminated. To the best of our knowledge, Thinker-IM is the first digital-CIM mixed processor for speech recognition. The major contributions are: 1) An OWDS architecture to maximize parallelism and data reuse of SRAM-CIM macros array; 2) SRAM-CIM macros with serial-phase triple sensing controller to support 3-bit outputs and weight adaptation to redistribute the BL currents; 3) An aggressive predictive execution mechanism and corresponding PBUs to reduce the operations of binary RNN inferences.

CIM-based RNN engine with OWDS dataflow

Fig.2 shows the overall architecture including digital front-end, CIM-based RNN engine and decoder. In digital front-end, 25ms audio frames enter VAD, FFT, mel-filter and compressed quantization unit serially to generate 256-dimentional binary speech features, which are the inputs of RNN processing. The CIM-based RNN engine mainly contains 16 64×64b SRAM-CIM macros connected in parallel and a PBU. Each bitline (BL) of 16 SRAM macros complete MAC operations in full parallel and the corresponding BLs are connected to a 16-input adder tree. Therefore, weights are fixed in SRAM cells and outputs are fixed in adder trees, which forms the OWDS dataflow. Compared to pure weight-stationary dataflow, OWDS reduces 85.7% memory accessing. In each RNN iteration, 64 output neurons are regarded as a group. Weights of each output neuron are split and stored in cells on the same BL of 16 macros, and input data (IN) is converted into wordline (WL) activation. Every 4 adjacent WLs are activated simultaneously in each cycle. Therefore, 64b-IN are scattered along 4 WLs by 16 parallel macros. After that, each of the 64 adder trees receives 16 3b outputs from 16 macros and produces 16b Psum. The PBU accumulates 16b-Psums for 64 output neurons and conducts predictive execution of RNN. After RNN inference, the decoder generates final recognition result.

3b-output SRAM-CIM and CIM-aware weight adaptation

Fig. 3 shows the area-efficient 3b-output XNOR SRAM-CIM macro. In XNOR operation, WLL=1 represents IN=1 and WLR=1 represents IN=-1. When a WLL or WLR is activated, the read current of each activated memory cell represents the input-weight-product (IWP=IN×W). An IWP=1 cell generates a charging current (I_{MC-C}) on BL, an IWP=-1 cell generates a discharging current (I_{MC-D}) on BL. With a voltage-divider type sensing scheme, the BL voltage (V_{BL}) is the MAC value which refers to the summation of 4 IWPs. Serial-phase triple sensing controller (TSC) selects reference voltages (V_{REF1}~V_{REF3}) with the front sensing result and controls a voltage sense amplifier (VSA). The VSA compares V_{BL} with V_{REF1}~V_{REF3} in three sequential sensing phases to generate 3b MAC output without increased area. To sense V_{BL} with enough margins, I_{MC-C} and I_{MC-D} are set as $I_{MC-D}>>I_{MC-C}$, which means energy consumed when IWP=-1 is larger than that of IWP=1. To reduce the energy of BL accumulating, a CIM-aware RNN training flow is proposed to adapt weights to increase the occurrence of IWP=1. A regularizer which includes the summation value of RNN outputs is added to the loss function for training. As minimizing the loss function, the ratio of IWP=1 is increased. It achieves average 9.9% energy reduction with over 90.2% speech recognition accuracy on various benchmarks.

Predictive early BN and binarization mechanism

Fig. 4 shows the predictive binarization technique fused with BN to reduce the operation count in RNN iterations. Binarization means comparing the final neuron output with zero. If an intermediate accumulation result is large enough, which exceeds the upper bound of remaining accumulations (*exact threshold*), the binarized result can be early determined without error. Due to the fault-tolerant nature in RNN, a portion of neurons (N_{ex}) which are exactly predicted is enough to guarantee the final recognition accuracy. Therefore, the exact threshold can be relaxed to an *aggressive threshold.* The aggressive threshold and N_{ex} are determined by offline training. When fused with BN, the exact and aggressive threshold are revised by a bias (V_{TH0}). The PBU contains 64 predict lanes, a 64-input adder tree and a comparator. Each lane includes 3 comparators, an accumulator and a look-up-table (LUT) providing predictive parameters. Each lane accumulates Psums from the adder tree in RNN engine and predicts binary result by exact threshold first. The 64-input adder tree counts the number of exact predicted neuron (*NumL*). When *NumL>N_{ex},* the current RNN iteration is stopped and the rest neurons are predicted by aggressive threshold. With over 90.2% accuracy, 24.5% computations can be reduced on average.

Measurement results

Thinker-IM is fabricated in a 65 nm CMOS technology with 6.2mm^2 die area. Fig. 5 shows measurement results and the comparison with state-of-the-art. The average recognition accuracy is 92.4% on three benchmarks. Due to SRAM-CIM technique, the peak area efficiency and arithmetic energy efficiency are 156GOPs/mm^2 and 11.7TOPS/W, which are at least 4.6× and 1.2× better than previous works [3-5], respectively. Varying supply voltage and frequency, the minimal energy consumed per neuron and per inference are 5.1pJ and 3.36uJ, outperforming [4, 5] by over 2.8× and 1.9×, respectively. The improvements mainly result from the reduced memory accessing by OWDS dataflow and the low current oriented weights adaptation for CIM macros. Benefited from the aggressive predictive execution, the minimum latency per inference is 127.3us, which is more than 3.9× shorter than state-of-the-art. Fig. 6 shows the chip photograph and summary.

References

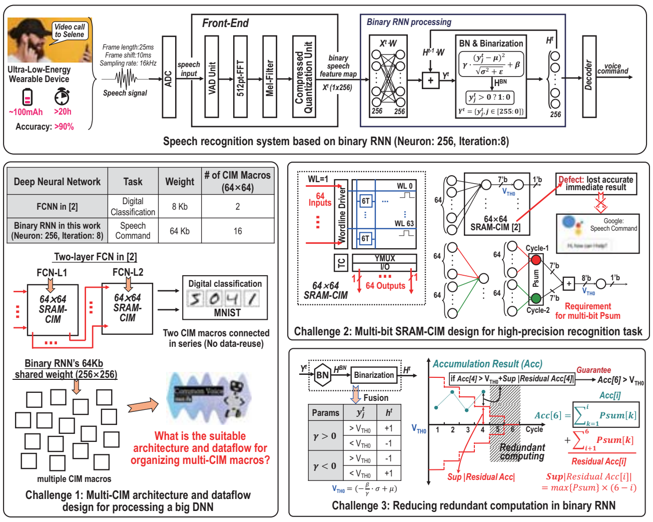
[1] W. Khwa, *et al., ISSCC*, 2018.

[2] S. Gonugondla, *et al, ISSCC*, 2018.

[3] M. Price, *et al., ISSCC*, 2017.

[4] S. Bang, *et al., ISSCC*, 2017.

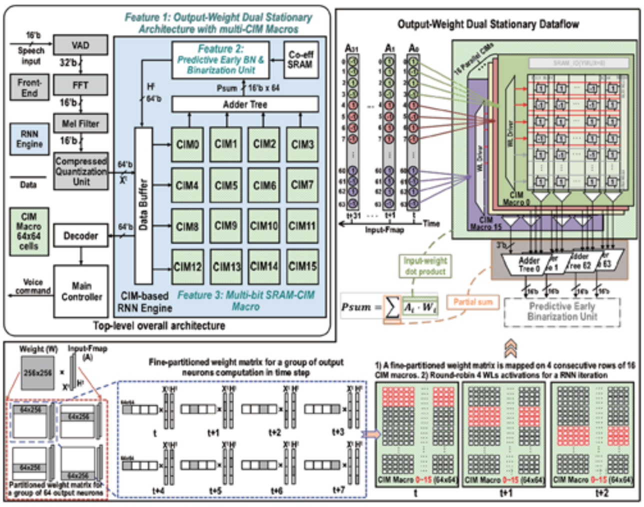
[5] S. Yin, *et al. VLSI Symp.*, 2018.



Video call to Selene
Ultra-Low-Energy Wearable Device
~100mAh >20h
Accuracy: >90%
Front-End
Speech signal
ADC
VAD Unit
512pt-FFT
Mel-Filter
Compressed Quantization Unit
binary speech feature map
Binary RNN processing
BN & Binarization
Decoder
voice command
Speech recognition system based on binary RNN (Neuron: 256, Iteration:8)
Two-layer FCN in [2]
FCN-L1
FCN-L2
64×64 SRAM-CIM
Digital classification
MNIST
Two CIM macros connected in series (No data-reuse)
Binary RNN's 64Kb shared weight (256×256)
multiple CIM macros
What is the suitable architecture and dataflow for organizing multi-CIM macros?
Challenge 1: Multi-CIM architecture and dataflow design for processing a big DNN
Defect: lost accurate immediate result
Requirement for multi-bit Psum
Challenge 2: Multi-bit SRAM-CIM design for high-precision recognition task
Accumulation Result (Acc)
Guarantee
Redundant computing
Challenge 3: Reducing redundant computation in binary RNN

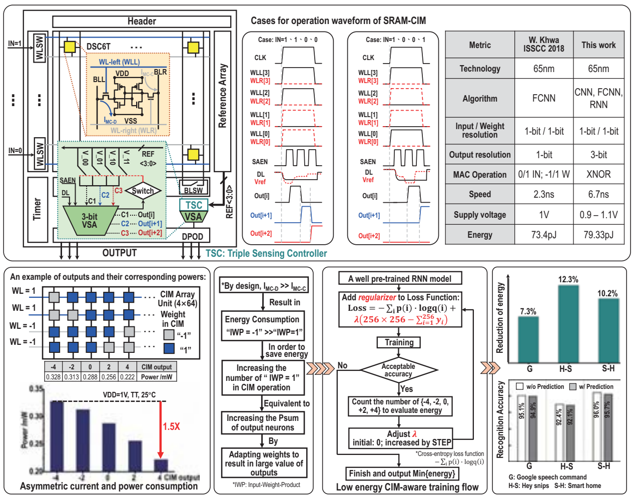
Deep Neural Network	Task	Weight	# of CIM Macros (64×64)
FCNN in [2]	Digital Classification	8 Kb	2
Binary RNN in this work (Neuron: 256, Iteration: 8)	Speech Command	64 Kb	16

Fig. 1. Challenges in CIM-based speech recognition processor.



Feature 1: Output-Weight Dual Stationary Architecture with multi-CIM Macros
Feature 2: Predictive Early BN & Binarization Unit
Co-eff SRAM
Adder Tree
VAD
FFT
Mel Filter
Compressed Quantization Unit
Data Buffer
CIM0 CIM1 CIM2 CIM3 CIM4 CIM5 CIM6 CIM7 CIM8 CIM9 CIM10 CIM11 CIM12 CIM13 CIM14 CIM15
Decoder
Main Controller
CIM-based RNN Engine
Feature 3: Multi-bit SRAM-CIM Macro
Top-level overall architecture
Output-Weight Dual Stationary Dataflow
Input-Fmap
Input-weight dot product
Partial sum
Predictive Early Binarization Unit
Fine-partitioned weight matrix for a group of output neurons computation in time step
1) A fine-partitioned weight matrix is mapped on 4 consecutive rows of 16 CIM macros. 2) Round-robin 4 WLs activations for a RNN iteration
Partitioned weight matrix for a group of 64 output neurons

Fig. 2 Top-level architecture and OWDS dataflow.



Header
Reference Array
Timer
OUTPUT
TSC: Triple Sensing Controller
Cases for operation waveform of SRAM-CIM
An example of outputs and their corresponding powers:
Asymmetric current and power consumption
VDD=1V, TT, 25°C
1.5X
By design, I_{MC-0} >> I_{MC-1}
Energy Consumption "IWP = -1" >> "IWP=1"
In order to save energy
Increasing the number of "IWP = 1" in CIM operation
Increasing the Psum of output neurons
Adapting weights to result in large value of outputs
A well pre-trained RNN model
Training
Acceptable accuracy
Low energy CIM-aware training flow
12.3%
7.3%
10.2%
Reduction of energy
Recognition Accuracy
G: Google speech command
H-S: Hey snips S-H: Smart home

Metric	W. Khwa ISSCC 2018	This work
Technology	65nm	65nm
Algorithm	FCNN	CNN, FCNN, RNN
Input / Weight resolution	1-bit / 1-bit	1-bit / 1-bit
Output resolution	1-bit	3-bit
MAC Operation	0/1 IN; -1/1 W	XNOR
Speed	2.3ns	6.7ns
Supply voltage	1V	0.9 – 1.1V
Energy	73.4pJ	79.33pJ

Fig. 3 3b-output CIM macro and weight adaptation method.

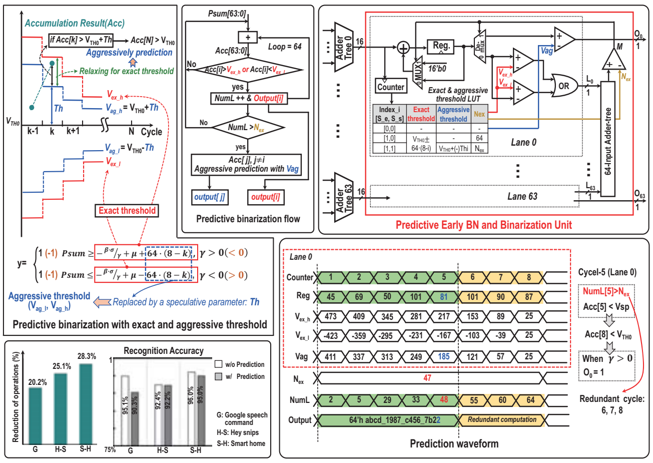
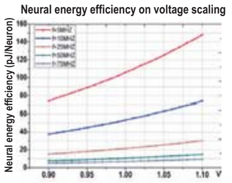


Fig. 4 Predictive execution mechanism and PBU.



Speech Recognition Results on Benchmarks

Dataset	# of Keywords	Accuracy	False Alarm	Recall
Google Speech command [a]	10	90.2%	1.50%	71.8%
Hey snips [b]	1	91.9%	1.65%	69.3%
Smart home [c]	11	95.0%	1.48%	71.0%

a: http://download.tensorflow.org/data/speech_commands_v0.02.tar.gz
b: Coucke A, *et al.* arXiv preprint arXiv:1811.07684, 2018
c: Self-made data set

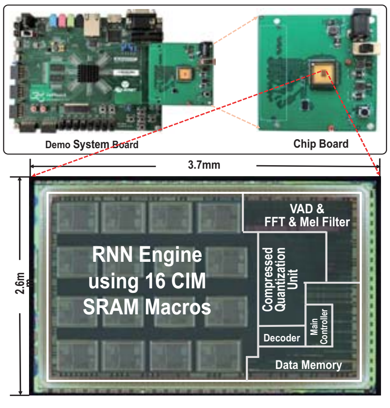
Metric	ISSCC 2017 [3]	ISSCC 2017 [4]		VLSI 2018 [5]		This work
Technology	65nm	40nm		28nm		65nm
Algorithm	DNN+HMM	DNN		CNN+DNN		RNN
Dataset	WSJ	/		TIMIT/ TIDGIT/ Smart home		Google/ Hey snips/Smart home
Architecture	Digital	Digital		Digital		Digital and CIM
Die Area (mm^2)	9.61mm^2	7.1mm^2		2mm^2		6.2mm^2
On-chip SRAM (Mb)	5.84Mb	144KB		27KB		10KB (SRAM) + 64Kb (CIM-SRAM)
Supply Voltage (V)	0.6 V – 1.2 V	0.63 V – 0.9 V		0.57 V – 0.9 V		0.9 V -1.1 V
Clock Freq. (MHz)	3 – 86 MHz	1.9 – 19.3 MHz		2.5 – 50 MHz		5 – 75 MHz
Max. Accuracy	92%	/		95%		95%
Peak Performance (GOPS)	22.0	/		307.2	132.3 [a]	614.4
Peak Area Efficiency (TOPS/mm^2)	0.00293	/		0.421	0.0337 [b]	0.156
Arithmetic Energy Efficiency (TOPS/W)	/	0.318 @ 0.65V, 3.9MHz	0.1021 [c]	54.4 @ 0.57V, 2.5MHz	9.39 [c]	11.7 @ 0.9V, 75MHz
Neural Energy Efficiency * (pJ/Neuron)	36 @ 0.9V, 10.2MHz	2540 @ 0.65V, 3.9MHz	7910 [d]	2.46 @ 0.57V, 2.5MHz	14.23 [d]	5.1 @ 0.9V, 75MHz
Min. Latency per Inference * (us)	/	6500 @ 3.9MHz		500 @ 50MHz		127.3 @ 75MHz
Min. Energy per Inference * (uJ)	10.1 @ 0.9V, 3MHz	2.086 @ 0.65V, 3.9MHz	6.50 [d]	1.47 @ 0.57V, 2.5MHz	8.51 [d]	3.36 @ 0.9V, 75MHz

For fair comparison, scaling the previous work's results to 65nm, 0.9V :
a: Scaled peak performance = Absolute peak performance×(technology/65nm)
b: Scaled peak area efficiency = Absolute peak area efficiency×(technology/65nm)^3
c: Scaled arithmetic energy efficiency = Absolute arithmetic energy efficiency×(technology/65nm)×(voltage/0.9V)^2
d: Scaled energy per inference(neuron) = Absolute energy per inference(neuron)/(technology/65nm)/(voltage/0.9V)^2
*: the lower the better

Fig. 5 Measurement results and comparison.

Chip Summary	
Process	65nm CMOS
Supply Voltage	0.9 - 1.1V
Frequency	5 - 75MHz
Core Size	3.1×2mm^2
Die Size	3.7×2.6mm^2
Logic Gates (NAND2)	1M
SRAM Capacity	10KB
SRAM-CIM Macro Capacity	16×4Kb
Latency per Inference	127.3us - 1.91ms
Energy per Inference	3.36uJ - 49.2uJ
Energy per Neuron	5.1 - 148.2 pJ/Neuron
Arithmetic Energy Efficiency	6.45 - 11.7 TOPS/W

(a) Chip Summary



(b) Chip Photograph

Fig. 6 Chip summary and photograph.