An In-Memory-Computing Charge-Domain Ternary CNN Classifier

Xiangxing Yang\textsuperscript{1}, Keren Zhu\textsuperscript{1}, Xiyuan Tang\textsuperscript{1}, Meizhi Wang\textsuperscript{1}, Mingtao Zhan\textsuperscript{2}, Nanshu Lu\textsuperscript{1}, Jaydeep P. Kulkarni\textsuperscript{1}, David Z. Pan\textsuperscript{1}, Yongpan Liu\textsuperscript{2}, Nan Sun\textsuperscript{1,2}

\textsuperscript{1}University of Texas at Austin, Austin, TX
\textsuperscript{2}Tsinghua University, Beijing, China
Outline

• Motivations
• Existing Works
• Theoretical Concept of the Proposed Work
• Circuit Implementation
• Measurement results
• Summary
Quest for Energy Efficient Edge Computing System

Increasing need from various applications:

Pattern Recognition    Image Classification    Object recognition
Challenges on Energy Efficient NN Inference

- High computation energy
- High memory access energy

**Total MACs**
**Total Weights**

<table>
<thead>
<tr>
<th>Model</th>
<th>MACs</th>
<th>Weights</th>
</tr>
</thead>
<tbody>
<tr>
<td>LeNet5</td>
<td>60k</td>
<td>341k</td>
</tr>
<tr>
<td>AlexNet</td>
<td>61M</td>
<td></td>
</tr>
<tr>
<td>Reset50</td>
<td></td>
<td>25.5M</td>
</tr>
<tr>
<td>VGG16</td>
<td>138M</td>
<td>15.5G</td>
</tr>
</tbody>
</table>

[V. Sze, Proceedings of the IEEE 105.12 2017]
Challenges on Energy Efficient NN Inference

\[ h = g \left[ \left( \sum_{i=1}^{m} w_i \cdot x_i + b \right) \right] \]

- \( x_i \): Input activation
- \( w_i \): Weight
- \( b \): Bias
- \( h \): Output to next layer
- \( g \): Activation function

Accumulation
Challenges on Energy Efficient NN Inference

1. Higher resolution cost more energy

2. Memory access is expensive

Rough Energy Costs in 45nm 0.9V

- Add
- Mult
- SRAM Access(64b)

<table>
<thead>
<tr>
<th>Format</th>
<th>Add</th>
<th>16b</th>
<th>Mult</th>
<th>16b</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integer</td>
<td>0.03</td>
<td>0.4</td>
<td>0.9</td>
<td>1.1</td>
</tr>
<tr>
<td>Floating Point</td>
<td>0.1</td>
<td>0.2</td>
<td>3.1</td>
<td>3.7</td>
</tr>
</tbody>
</table>

[SRAM Access(64b) 10]

[M. Horowitz, ISSCC 2014]
Solutions to Energy Efficient NN Inference

• Conventional computing:

SRAM/DRAM
Load data for computing/
Store results

Memory

Processing Element(PE)

Compute

Memory access can easily dominate energy/throughput

• In-memory-computing:

Minimized data movement from distributed memory
Solutions to Energy Efficient NN Inference

- Reduced Resolution Network:

\[ \text{32b Floating point} \rightarrow ? \]

Multiplying energy cost
Solutions to Energy Efficient NN Inference

- Reduced Resolution Network:

  CIFAR-10, ResNet-56
  Activations are quantized to 1/2/3/4/8/32b

  [Y. Dong, IJCV 2019]

  Visualization of filters from binary neural network

  [M. Courbariaux, arXiv 2016]
Energy Cost of NN Inference

\[ \text{Power} = \text{Rate} \times \frac{\text{Energy}}{\text{Inference}} = \text{Rate} \times \frac{\text{Operations}}{\text{Inference}} \times \frac{\text{Energy}}{\text{Operation}} \]

[Energy / Inference]  

[Ops/Inference  

Energy/OP]

[B. Murmann, ISSCC 19 Tutorial]
Outline

• Motivations
• Existing Works
• Theoretical Concept of the Proposed Work
• Circuit Implementation
• Measurement results
• Summary
Existing works

• Digital Domain:
  - Bit error free 😊
  - High power from digital adder tree 😞
  - Low throughput 😞

[K. Ando, JSSC 18]
Existing works

- Current Domain:
  - High throughput 😊
  - PVT-robustness 😞
  - Consumes static current 😞

[J. Zhang, JSSC 17]
Existing works

• Charge Domain:
  - High throughput 😊
  - No static current 😊
  - Large operations/inference ☹️

\[ \frac{v_{td}}{V_{DD}} = \frac{C_u}{C_{tot}} \left( \sum_{i=0}^{N-1} w_i x_i + b \right) \]

[D. Bankman, JSSC 18]
Outline

• Motivations
• Existing Works
• Theoretical Concept of the Proposed Work
• Circuit Implementation
• Measurement results
• Summary
Comparison of Model Size

Baseline test: 98% Accuracy on MNIST

1b Resolution
1.38x10^8 OPs

~4x Bigger model size

1.5b Resolution
3.57x10^7 OPs
\{w,x\} from -1,0,1
Mixed Signal BNN vs TNN

SAR ADC

SAR ADC with $V_{CM}$ based switching

Mixed-Signal BNN

1b $W \times X$

Activation function

Input

Output

Mixed-Signal TNN

1.5b $W \times X$

Activation function

Input

Output
Mixed Signal BNN vs TNN

<table>
<thead>
<tr>
<th>Hardware Complexity</th>
<th>Operations Inference (@same accuracy)</th>
<th>Energy Operation (CDAC signal swing)</th>
<th>Energy Inference</th>
</tr>
</thead>
<tbody>
<tr>
<td>BNN</td>
<td>☺ ☺ ☺</td>
<td>☺ ☺ ☺</td>
<td>☺ ☺ ☺</td>
</tr>
<tr>
<td>TNN</td>
<td>☺</td>
<td>☺ ☺</td>
<td>☺ ☺ ☺</td>
</tr>
</tbody>
</table>

VREFP \( \rightarrow \) VREFN
\( V_{REF} \) Output
1b W * X

VREFP \( \rightarrow \) VREFN
\( V_{REF} \) Output
1.5b W * X

Mixed-Signal BNN

Mixed-Signal TNN

OPs/Inference \( \downarrow \) 75%
Energy/Operation \( \downarrow \) 31%
Energy/Inference \( \downarrow \) 82%
Outline

- Motivations
- Existing Works
- Theoretical Concept of the Proposed Work
- Circuit Implementation
- Measurement results
- Summary
On-chip Neural Network Model

<table>
<thead>
<tr>
<th>Layer</th>
<th>Type</th>
<th>Size</th>
<th>Channel</th>
<th>Filter Size</th>
<th>Dilated</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CONV-TN</td>
<td>30x30</td>
<td>1(input)</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>CONV-TN</td>
<td>28x28</td>
<td>32</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2p</td>
<td>MAX POOL</td>
<td>26x26</td>
<td>32</td>
<td>2x2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>CONV-TN</td>
<td>13x13</td>
<td>32</td>
<td>2x2</td>
<td>1</td>
</tr>
<tr>
<td>3p</td>
<td>MAX POOL</td>
<td>12x12</td>
<td>32</td>
<td>2x2</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>FC</td>
<td>(Flatten 6x6x32) 1152 - 10</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Chip Architecture

- **Ternarized Input Image** (From MNIST)
- **Digital CONV1 (32ch * 4)**
- **CONV2 (32ch)**
  - 1 KB Weights
  - 256 Bytes Bias SRAM
- **CONV3 (32ch)**
  - 1 KB Weights
  - 256 Bytes Bias SRAM
- **FC (1152 - 10)**
  - 288 Bytes Image SRAM
  - 2.88 KB Weights SRAM
- **MAXPOOLDING**
- **MAXPOOLDING**
- **Image SRAM**
  - 1352 Bytes

Results (0~9)

Weights and Bias

Preload to chip

0.18uJ/classification @ 97.1% accuracy
CONV1 – Example of One-Channel Convolution

Filter 2x2 Dilated L = 2

Ternarized Input Image 1ch

Output Image 1ch

Output Image 1ch

X_{011} = \text{STEP}(W_{11}X_{i11} + W_{12}X_{i13} + W_{21}X_{i31} + W_{22}X_{i33})

X_{012} = \text{STEP}(W_{11}X_{i12} + W_{12}X_{i14} + W_{21}X_{i32} + W_{22}X_{i34})

W, X \in \{-1, 0, 1\}
CONV1 – Example of 32-Channel Convolution

\( W, X \in \{-1, 0, 1\} \)
CONV1 – Digital Implementation

Slice 0
- 4 input pixels (DFF * 8)
- Filter1 2x2 (DFF * 8)

Slice 1
- 4 input pixels (DFF * 8)

Slice 31

8b 8b 8b 8b

STEP\left(\sum_{i=1}^{4} (W_i \times X_i)\right)

Combinational Logic

ϕ_{EN}

Data I/O
- Loading W
- Loading X

ϕ_{EN}

One Pixel Output (32 ch)

CONV1 Output
CONV1 – Digital Implementation

Slice 0 - 31
- Input Registers
- Filter Registers

Slice 32 - 63

Slice 64 - 95

Slice 96 - 127

Data I/O
- Loading W
- Loading X

$\Phi_{EN}$

CONV2

256b

CONV2$_{EN}$

32ch 2x2 pixels

STEP($\sum_{i=1}^{4} (W \ast X)$)
CONV2 – Example of 32-Channel Convolution

Filter0
2x2x32

Filter1

CONV1
Output
2x2x32

Filter31

CONV2
Output
32ch

\[ \mathbf{X_0} = \text{STEP}(\sum_{i=1}^{128} (\mathbf{W}_i \cdot \mathbf{X}_i) + \text{Bias}) \]

Step Activation Function of CONV2
CONV2 – Implementation of One-Channel SC Neuron

\[ V_x = \frac{C_u}{C_{Total}} \cdot (V_{REFP} - V_{REFN}) \cdot \left( \sum_{i=1}^{128} (W_i \cdot X_i) + \sum_{i=1}^{32} \text{Bias}_i \right) \]

\[ C_{Total} \approx 160 \ C_u \]

\( (W_i \cdot X_i), \text{Bias} \subseteq \{ V_{REFP}, V_{CM}, V_{REFN} \} \)
CONV2 – Synapse Design

### Encoding for simplicity:

<table>
<thead>
<tr>
<th>DEC</th>
<th>BIN</th>
<th>Voltage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>$V_{REFP}$</td>
</tr>
<tr>
<td>-1</td>
<td>11</td>
<td>$V_{REFN}$</td>
</tr>
<tr>
<td>0</td>
<td>0X</td>
<td>$V_{CM}$</td>
</tr>
</tbody>
</table>

$1.5b$ Multiplier $\rightarrow$

\[ \text{IN1}_H, \text{IN2}_H \rightarrow \text{OUT}_H \]
\[ \text{IN1}_L, \text{IN2}_L \rightarrow \text{OUT}_L \]
CONV2 – Comparator Design

Dout = STEP (Vx)

Dout_H

Dout_L

V_{INP}

V_{REF+}

V_{REF-}

V_{INN}

V_{X} = V_{INP} - V_{INN}

V_{+} = V_{REF+} - V_{REF-}

V_{-} = V_{REF-} - V_{REF+}

Differential Comparator

AVDD

COMP OUT

CLK

Moscap Array

5b SRAM

VDD

GND

V_{REF+}

V_{REF-}

Offset Calibration

C2C4C8C16C

5b SRAM

VDD

GND

Moscap Array

Offset Calibration
**CONV2 – Effect of Comparator Offset**

\[ \text{Dout} = \text{STEP} \ (V_x) \]

\[ V_+ = V_{\text{REF}+} - V_{\text{REF}-} \]
\[ V_- = V_{\text{REF}-} - V_{\text{REF}+} \]

Ideal activation function

Actual activation function

Offset \sim 10 \text{ LSB}
CONV2 – Foreground Comparator Offset Calibration

From off-chip DAC

Inputs set to V_CM

Offset code set to minimum

Fire comparator 1000 times

Avg output ≈ 0.5?

Yes

Calibration done

No

Offset code += 1

RST = 1
CONV2 – Foreground Comparator Offset Calibration

- Offset ~ 10 LSB w/o Offset Calibration
- Offset < 1 LSB w/ Offset Calibration
CONV2 – Maxpooling

Architecture of CONV2 (Single-end shown)
Datapath from CONV2 to CONV3

E.g. CONV2 output

CONV3 Filter Window

CONV2 Output Image

Only need to read 2 pixels from SRAM

MUX

CONV3 EN

x144
FC Layer Operation

**CONV3 Output Image**
6x6x32

[Diagram of CONV3 Output Image]

**Flattened**
1152 x 1

**Weights for ‘0-9’**

\[
\text{Logit} = \left( \sum_{i=1}^{1152} (W_i \ast X_i) \right)
\]

\(W_i, X_i \in \{-1,0,1\}\)

‘0’:
‘1’:
‘2’ : 5
‘3’ : 24
‘4’ : 35
‘5’ : -22
‘6’ : 117
‘7’ : -4
‘8’ : -8
‘9’ : 42

Classification Result : 6
FC Layer Implementation

32 Channels

MEMORY DECODER
Scan In
SRAM WRITE DRIVER
640 BITS WEIGHTS SCAN CHAIN

W*X 1.5bit Multiplier

20 bit Weights SRAM

φi Select weights for 0~9

SE<3:0>

1 pixel load after CONV3 conversion

32 Channels

Φ0 Φ0 Φ0

Φ1 Φ1 Φ1

Φ35 Φ35 Φ35

36 Pixels

D_{OUT, CONV3} 64b

DOUT_CONV3 64b

2b 2b 2b

Scan In 640 BITS WEIGHTS SCAN CHAIN

(Single-ended shown)
FC Layer Implementation

![Diagram of FC Layer Implementation](image)

- **RST**
- **ON1**
- **ON2**
- **RST1**
- **RST2**
- **COMP**

- '0' on C1
- '1' on C2
- '0'>='1'?
- '0'<='1'
- Select weights '2'
- '2' on C1
- '1'>='2'?
- '2'<='1'
- Reset C1

- **VCM**
- **VIN**
- **OUT**

- **VCM**
- **RST**
- **COMP**
Outline

• Motivations
• Existing Works
• Theoretical Concept of the Proposed Work
• Circuit Implementation
• Measurement results
• Summary
Die Photo

- 40nm LP CMOS
- Active Area: 0.98mm²
- Supply: 0.8V/0.7V/0.9V
Measurement Results

![Measurement Results Image]

- **Coverage Details**
  - Digital and Memory: 44.1 uW
  - CDAC and Multipliers: 43.7 uW
  - Comparators: 7.8 uW

- **Accuracy**
  - 97.1% accuracy @ 549 FPS

<table>
<thead>
<tr>
<th>Power Domain</th>
<th>Voltage</th>
<th>Energy</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVDD</td>
<td>0.7 V</td>
<td>44.1 uW</td>
</tr>
<tr>
<td>AVDD</td>
<td>0.8 V</td>
<td>7.8 uW</td>
</tr>
<tr>
<td>V_REF</td>
<td>0.9 V</td>
<td>37.8 uW</td>
</tr>
<tr>
<td>V_CM</td>
<td>0.45 V</td>
<td>5.9 uW</td>
</tr>
</tbody>
</table>
## Comparison table

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Technology</td>
<td>40nm</td>
<td>65nm</td>
<td>28nm</td>
<td>55nm</td>
<td>65nm</td>
<td>65nm</td>
</tr>
<tr>
<td>Circuit Type</td>
<td>Mixed-Signal Charge-domain</td>
<td>Digital</td>
<td>Mixed-Signal Charge-domain</td>
<td>Mixed-Signal Current-domain</td>
<td>Mixed-Signal Current-domain</td>
<td>Mixed-Signal Charge-domain</td>
</tr>
<tr>
<td>Bit Precision</td>
<td>1.5b</td>
<td>1/1.5b</td>
<td>1b</td>
<td>1.8b</td>
<td>1.5b</td>
<td>1b</td>
</tr>
<tr>
<td>Area(mm²)</td>
<td>0.98</td>
<td>3.9</td>
<td>4.6</td>
<td>5.85</td>
<td>0.055</td>
<td>12.6</td>
</tr>
<tr>
<td>Area Eff.(GOPS/mm²)</td>
<td>469¹</td>
<td>105</td>
<td>67</td>
<td>N/A</td>
<td>N/A</td>
<td>1498</td>
</tr>
<tr>
<td>Operating VDD(V)</td>
<td>0.8/0.7/0.9</td>
<td>0.55-1.0</td>
<td>0.8/0.8</td>
<td>0.9</td>
<td>0.8/0.45</td>
<td>0.94/0.68/1.2</td>
</tr>
<tr>
<td>Energy Eff.(TOPS/W)</td>
<td>556²</td>
<td>2.3-6.0</td>
<td>532</td>
<td>40.2</td>
<td>490-15.8</td>
<td>866</td>
</tr>
<tr>
<td>Dataset</td>
<td>MNIST</td>
<td>MNIST</td>
<td>CIFAR-10</td>
<td>MNIST</td>
<td>MNIST</td>
<td>MNIST</td>
</tr>
<tr>
<td>Accuracy</td>
<td>97.1%³</td>
<td>90.1%</td>
<td>86.05%</td>
<td>98.56%</td>
<td>96.2%</td>
<td>98.6%</td>
</tr>
<tr>
<td>FPS</td>
<td>549</td>
<td>N/A</td>
<td>237</td>
<td>N/A</td>
<td>N/A</td>
<td>651</td>
</tr>
<tr>
<td>Power(mW)</td>
<td>0.096</td>
<td>N/A</td>
<td>0.899</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Operations / Inference</td>
<td>TNN</td>
<td>BNN (simu)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3.57x10⁷</td>
<td>1.38x10⁸</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MACs Energy / Inference</td>
<td>0.09uJ</td>
<td>0.52uJ</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>0.8uJ</td>
</tr>
<tr>
<td>Total Energy / Inference</td>
<td>0.18uJ</td>
<td>0.7uJ</td>
<td>N/A</td>
<td>3.8uJ</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>All operations on chip</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

¹Based on SC neuron
²Based on MACs energy efficiency
³10 runs average on 10,000 test set images.
Outline

• Motivations
• Existing Works
• Theoretical Concept of the Proposed Work
• Circuit Implementation
• Measurement results
• Summary
Summary

• A 1.5b charge domain ternary CNN classifier is proposed:
  - Fully on-chip NN with lowest energy/inference reported for >97% MNIST accuracy
  - Compared to BNN with same accuracy:
    • 75% \(\frac{\text{Operations}}{\text{Inference}}\) ↓
    • 31% \(\frac{\text{Energy}}{\text{Operation}}\) ↓
    • 82% \(\frac{\text{Energy}}{\text{Inference}}\) ↓
Thank You