2-B: Special Session: FPGA Technologies for Adaptive Computing- II

Memory-Access-Minimized BCNN Accelerator Using Nonvolatile FPGA with Only-Once-Write Shifting

Daisuke Suzuki\textsuperscript{1}, Takahiro Oka\textsuperscript{2}, and Takahiro Hanyu\textsuperscript{2}

\textsuperscript{1}The University of Aizu  

\textsuperscript{2}Research Institute of Electrical Communications, Tohoku University  
http://www.ngc.riec.tohoku.ac.jp
Trend of AI Accelerator

FPGA (Field-Programmable Gate Array)

- Merit
  - 😊 Flexibility (or reconfigurability)
  - 😊 Short design time & low cost
  - Can support state-of-the-art AI

- Demerit
  - 😞 High power consumption
    - Especially, standby power consumption is serious.

How to enhance the energy efficiency of the FPGA?

Speed (GOP/s)

Power (W)

Target Area (10-30TOPS/W)

Download: https://nicsefc.ee.tsinghua.edu.cn/projects/neural-network-accelerator
Nonvolatile FPGA (NV-FPGA)

**SRAM & CMOS FF (Volatile)**
- Always turned ON
  - (PS: Power switch)
- Quickly turned on/off
  - (A: Active I: Idle)

**Nonvolatile memory & FF**
- Nonvolatile FPGA
  - Always on to keep data
  - Instant on/off w/o losing data

**CMOS-only FPGA (volatile)**
- Static
  - Dynamic
  - High standby power

**NV-FPGA -> Suitable for edge-AI accelerator**
Magnetic Tunnel Junction (MTJ) Device

- Resistance value is dependent on M.
- M changes by write current $I_{WH}/I_{WL}$.
- M remains with no power supply.

Nonvolatile storage capability

Spin-Transfer Torque (STT) MTJ


Spin-Orbit Torque (SOT) MTJ


- Device can be fabricated in sub-nm size.
- Read port and write port are separated.

128Mb STT-MRAM

- High-density cache memory
- Fast switching capability
--IoT-oriented MCU has been fabricated.
--Use of NV reduces the power.
--Use of eFPGA improves the performance.

Energy efficiency (OPS/W) has been greatly enhanced.

Great energy efficiency by using NV-FPGA acceleration
Binary Convolutional Neural Network

Iteration of XNOR and bit-count operation

for (to=0; to<M; to++)
  for (row=0; row<R; row++)
    for (col=0; col<C; col++)
      for (ti=0; ti<N; ti++)
        for (i=0; j<K; i++)
          for (j=0; j<K; j++)
            ofm[to][row][col] += weight[to][ti][row+i][col+j] * ifm[ti][row+i][col+j];
Massively parallel architecture with no wasted standby power consumption
Only-Once-Write Shifting [1]


- Lookup table (LUT) circuit: Reconfigurable logic component of FPGA
- Data-shift function: Important function of the LUT circuit

Step 1

Input data stream: D6, D5, D4

M[2] = D2
M[1] = D1
M[0] = D0

Decoder (read/write)

A = 0

Output data stream: D0

Step 2

Input data stream: D6, D5, D4

M[2] = D2
M[1] = D1
M[0] = D4

Decoder (read/write)

A = 1

Output data stream: D1, D0

Active

(1) Read

Memory cell (Active)

(2) Write

Updated

Number of write access per cycle is minimized to one.

By using a single-ended circuitry, the function is compactly embedded.

Reduce energy for data transfer in BCNN accelerator
Estimation of Power Reduction


Input: MNIST 32 × 32pixel (0 padding)
Filter size: 5 × 5
Accuracy: 82%

Symbol legend:

Layer name (# of ifm, # of ofm)
Type of processing
Data format

# of 6-input LUTs for implementing PE: 784

Summary

<table>
<thead>
<tr>
<th>Layer</th>
<th>SRAM-FPGA-based (Conventional)</th>
<th>NV-FPGA-based (proposed)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active energy [a. u.]</td>
<td>0.877</td>
<td>-38.7%</td>
</tr>
<tr>
<td>Standby energy [a. u.]</td>
<td>0.123</td>
<td>-100%</td>
</tr>
<tr>
<td>Total Energy [a. u.]</td>
<td>1</td>
<td>-46.3%</td>
</tr>
</tbody>
</table>

45nm CMOS/MTJ technologies with 200MHz frequency

Potential for energy-efficient BCNN accelerator has been demonstrated.

2021/12/23

MCSoC2021
**Motivation**

**Overall architecture of NV-LUT-based BCNN accelerator**

PG (power-gating) technique is not applied yet.

After data is transferred to the accelerator core block, BRAM block must wait the completion of BCNN computing.

We also want to apply PG technique to BRAM block.

(*) BRAM is implemented as MTJ-based nonvolatile RAM (MRAM).

(*) IBUFs and WBUFs are implemented by using NV-LUT circuits with only-once-write shifting.
Once required data is loaded from BRAM to WBUF/IBUF, PG technique can be applied to BRAM until next data loading.

By reusing the contents of WBUFs/IBUFs, PG period can be increased!

Focus on the data locality of BCNN computing
1) Loop order: M,R,C,N,KW,KH (MRC)  
2) Loop order R,C,M,N,KW,KH (RCM)

IBUF loads data from BRAM for IFM. WBUF loads data from BRAM for weight.

Weights stored in WBUF are reused in R and C loops.

IFMs stored in IBUF are reused in M loop.

Layer 0
Layer 1
Layer 2
Layer 3
Layer 4

IFM size and filter size differ layer by layer -> Layer-wise interchange

Dog -> Cat -> Lion -> Bird

2021/12/23
Case Study (LeNet-5 Like BCNN)

Accuracy = 82%

Network parameters

<table>
<thead>
<tr>
<th>Layer</th>
<th>W, H</th>
<th>N</th>
<th>K</th>
<th>R,C</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>32</td>
<td>1</td>
<td>5</td>
<td>28</td>
<td>6</td>
</tr>
<tr>
<td>1</td>
<td>14</td>
<td>6</td>
<td>5</td>
<td>10</td>
<td>16</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>16</td>
<td>5</td>
<td>1</td>
<td>120</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>120</td>
<td>1</td>
<td>1</td>
<td>10</td>
</tr>
</tbody>
</table>

Loop unroll factors

<table>
<thead>
<tr>
<th>P_N</th>
<th>P_K</th>
<th>P_W, P_H</th>
<th>P_R, P_C</th>
<th>P_M</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>16</td>
</tr>
</tbody>
</table>

Training is performed by using PyTorch.
Layer-Wise Loop Interchange

<table>
<thead>
<tr>
<th>Layer</th>
<th>W, H</th>
<th>N</th>
<th>K</th>
<th>R,C</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>32</td>
<td>1</td>
<td>5</td>
<td>28</td>
<td>6</td>
</tr>
<tr>
<td>1</td>
<td>14</td>
<td>6</td>
<td>5</td>
<td>10</td>
<td>16</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>16</td>
<td>5</td>
<td>1</td>
<td>120</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>120</td>
<td>1</td>
<td>1</td>
<td>10</td>
</tr>
</tbody>
</table>

**Number of loaded pixels**

**Data retention cycle of buffers**

![Table of data retention cycle of buffers]

**Strategy for minimizing memory access**
1) Smaller number of loaded data
2) Longer data retention cycle of buffers

**Total BRAM capacity: 14.6Kb**
## Power Estimation

**Frequency:** 100MHz, **VDD:** 1.0V

<table>
<thead>
<tr>
<th></th>
<th>Conventional</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LUT circuit (6-input)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Logic operation [uW]</td>
<td>14.8 ¹)</td>
<td>14.6 ¹)</td>
</tr>
<tr>
<td>Shift operation [uW]</td>
<td>470 ¹)</td>
<td>215 ¹)</td>
</tr>
<tr>
<td>Standby [uW]</td>
<td>1.58 ¹)</td>
<td>0.01 ¹)</td>
</tr>
<tr>
<td><strong>Memory cell</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cell type</td>
<td>SRAM</td>
<td>MRAM (2T-1MTJ)</td>
</tr>
<tr>
<td>Read operation [uW]</td>
<td>0.3 ²)</td>
<td>0.3</td>
</tr>
<tr>
<td>Write operation [uW]</td>
<td>0.3 ²)</td>
<td>100</td>
</tr>
<tr>
<td>Standby [uW]</td>
<td>0.04 ²)</td>
<td>0.0001</td>
</tr>
<tr>
<td><strong>Multiplier (16-bit)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Logic operation [uW]</td>
<td>300</td>
<td></td>
</tr>
<tr>
<td>Standby [uW]</td>
<td>150</td>
<td></td>
</tr>
</tbody>
</table>

¹) SPICE simulation under 45-nm CMOS technology

Estimate total energy consumption for image inference
## Result

### Total energy per image

<table>
<thead>
<tr>
<th>Values in ( ) : With PG</th>
<th>SRAM-FPGA-based (Conventional)</th>
<th>NV-FPGA-based (proposed)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active energy [uJ]</td>
<td>15.6</td>
<td>14.4</td>
</tr>
<tr>
<td>Standby energy [uJ]</td>
<td>27.4</td>
<td>4.97 (0)</td>
</tr>
<tr>
<td>Total energy [uJ]</td>
<td>42.9</td>
<td>19.4 (14.4)</td>
</tr>
</tbody>
</table>

**Total energy@30FPS (activity ratio 2.05%)**

<table>
<thead>
<tr>
<th></th>
<th>SRAM-FPGA-based (Conventional)</th>
<th>NV-FPGA-based (proposed)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active energy [uJ]</td>
<td>15.6</td>
<td>14.4</td>
</tr>
<tr>
<td>Standby energy [uJ]</td>
<td>1347.6</td>
<td>250.5 (0)</td>
</tr>
<tr>
<td>Total energy [uJ]</td>
<td>1363.2</td>
<td>264.9 (14.4)</td>
</tr>
</tbody>
</table>

99.0% total power reduction is estimated NV-BCNN accelerator with PG
Summary

- MTJ-based NV-LUT circuit with only once-write data shifting
  - Suitable for ultra-low power BCNN accelerator

- Memory PG technique using layer-wise loop interchange
  - Further energy reduction

Future works

- Application to larger CNN and design space exploration
- Tool flow for implementing large scale NV-FPGA

Acknowledgements

This research is supported by JST-CREST(JPMJCR19K3), JST-OPERA, JSPS KAKENHI Grant No.JP16H06300 and JP20K11725, CIES cons. program, and VDEC.

2021/12/23