A Design of Scalable Deep Neural Network Accelerator Cores with 3D Integration

Masaaki Kondo

The University of Tokyo / RIKEN

(This talk is partly based on the following conference paper: R. Sakamoto, R. Takata, J. Ishii, M. Kondo, H. Nakamura, T. Ohkubo, T. Kojima and H. Amano, “The Design and Implementation of Scalable Deep Neural Network Accelerator Cores”, IEEE 11th International Symposium on Embedded Multicore/Many-core Systems-on-Chip, 2017.)

Convolutional Neural Network (CNN)

- Typically, there are three major layers in CNN
  - Convolutional, fully connected, and pooling layers

- Convolutional layer

$$y_{n,i,j} = \sum_{n=0}^{N-1}\sum_{m=0}^{M-1}\sum_{p=0}^{K-1}\sum_{q=0}^{K-1} \omega_{n,m,p,q}\, x_{m,i+p,j+q}$$

  - 2-dimentional convolutional operations
  - multiply-add operations with quadruple nested loops
  - The convolution layers tend to be computation bottleneck
- Fully connected layer

$$y_i = \sum_j w_{i,j} x_j$$

  - Matrix-vector multiplication
  - Little data reusability
  - The fully connected layer tends to be memory bottleneck

- These layers exhibit different execution characteristics

Research Objective and Concept

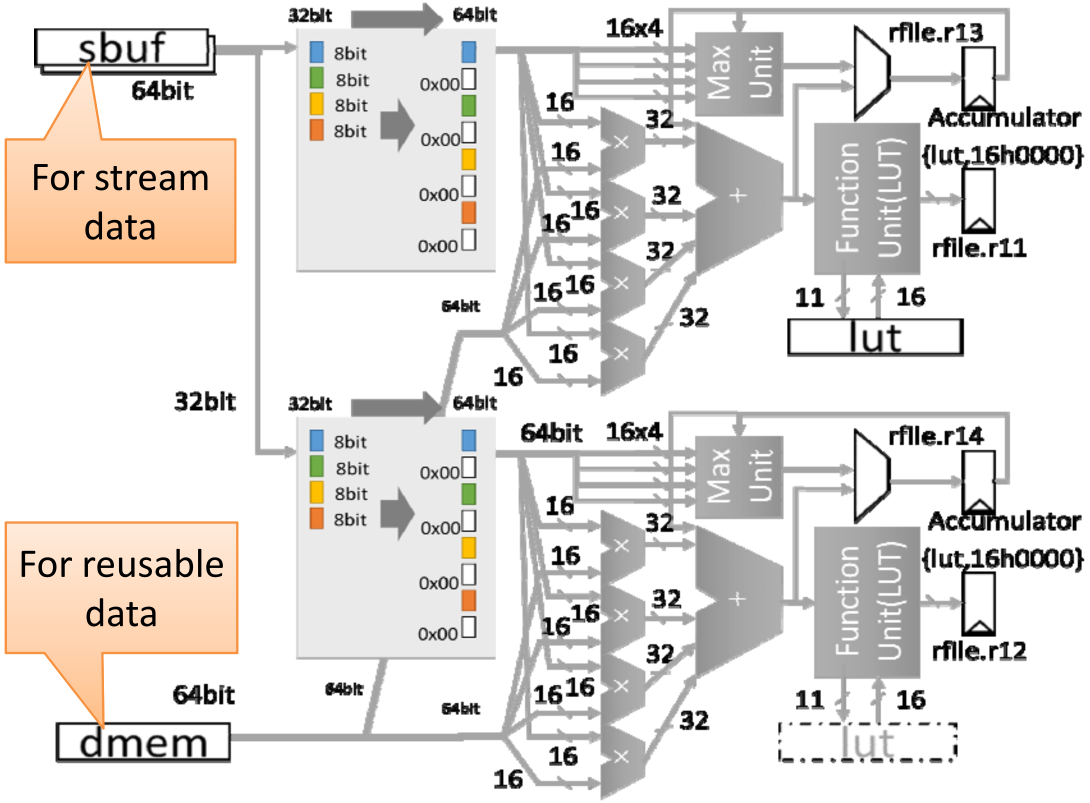
- R&D of flexible and power efficient DNN accelerator
- The concept of architectural design

<table>
<tr><th></th><th>Convolutional layer
【computing bottleneck】</th><th>Fully connected layer
【memory bottleneck】</th></tr>
<tr><td>Basic concepts</td><td colspan="2">1. SIMD MAC operation with reduced bit width
→ Power saving
2. Instruction-based exec. control w/ a microcontroller
→ Flexibility</td></tr>
<tr><td rowspan="2">Features</td><td colspan="2">1. Data reusability aware data allocation
→ Efficient use of double buffering
→ Distributed memories</td></tr>
<tr><td>2. Custom mutli-cycle SIMD
→ Reducing control overhead</td><td>3. Short bit-width mode
→ Reducing memory Bottleneck</td></tr>
</table>

DNN Accelerator: SNACC

- SNACC（*Scalable Neural Acceleration Cores with Cubic Integration*）
  – Efficient - custom SIMD, 8/16bit fixed-point arithmetics
  – Flexible - small microcontroller with custom ISA
  – Extensible - multicore, 3D-stackable

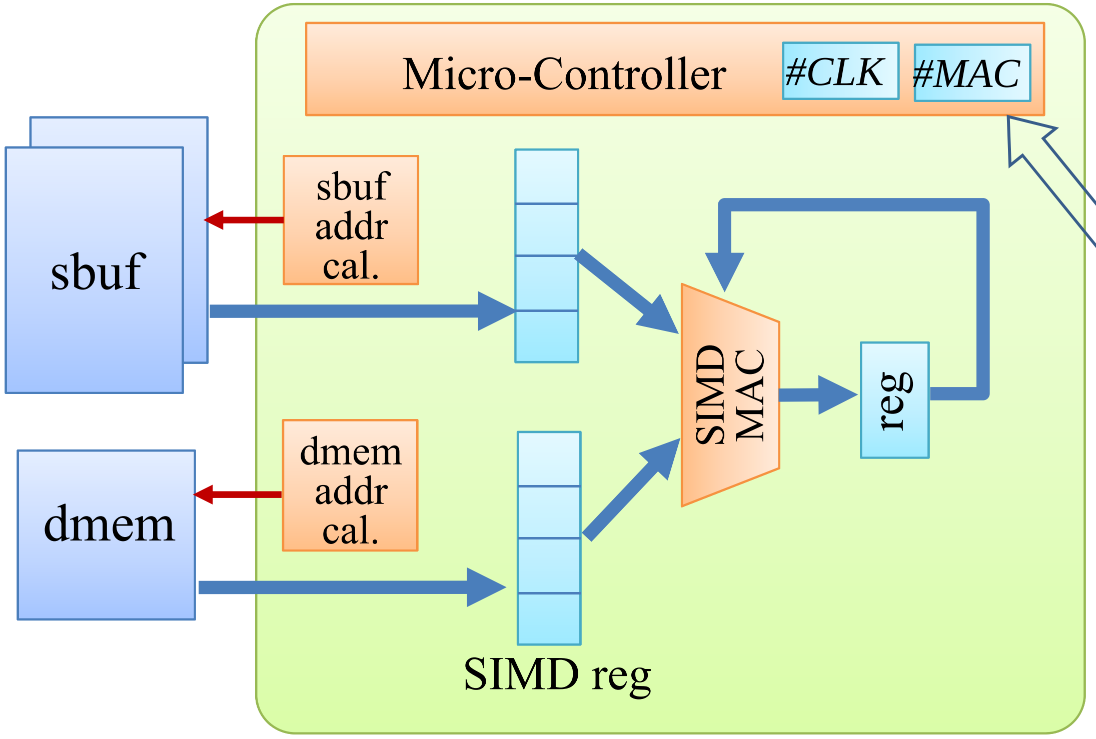
	Description
Overall structure	Multi-core accelerator
One core	Micro-controller + SIMD MAC operation unit
Instruction set	Custom instructions
MAC function	4 or 8-way SIMD/core
Memory	5 local memories/core Distributed memory Only an output memory is shared
Interconnect	Shared bus


For stream data

For reusable data

Custom Instruction for CNN Operations

- MAC loops in convolutional layers have many operations with nested loops → high overhead for instruction fetch and control
- A special loop instruction (madlp): executes convolutional operation loops with only a single instruction



Control overhead

1. Controlling SIMD loop
2. Address calculation
3. Updating registers
4. LUT lookup to calculate activate functions

Special control registers to specify the parameters of SIMD loop such as num. of loops, address offsets, etc.

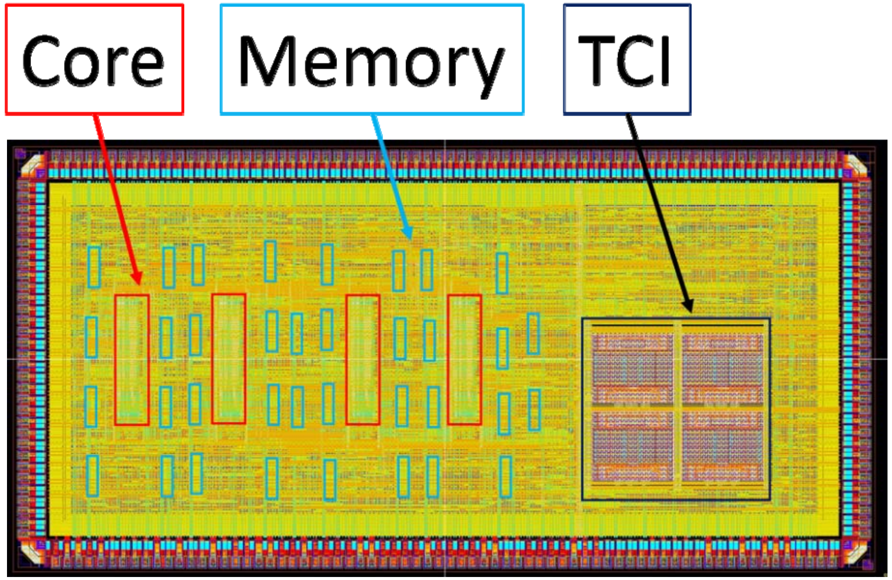
Prototyping an LSI chip for PoC

- Taped out a prototype LSI chip

Num. cores	4
Chip area	3mm x 6mm
Process	Renesas Electronics 65nm SOTB(VDEC)
Num. gate	18864/core
Vcc	0.55V
Frequency	50MHz
Local Memory	SRAM 68KB[1)]
TCI	ThruChip Interface[2)]

1): 16KB/core x 4-core + 4KB(shared)

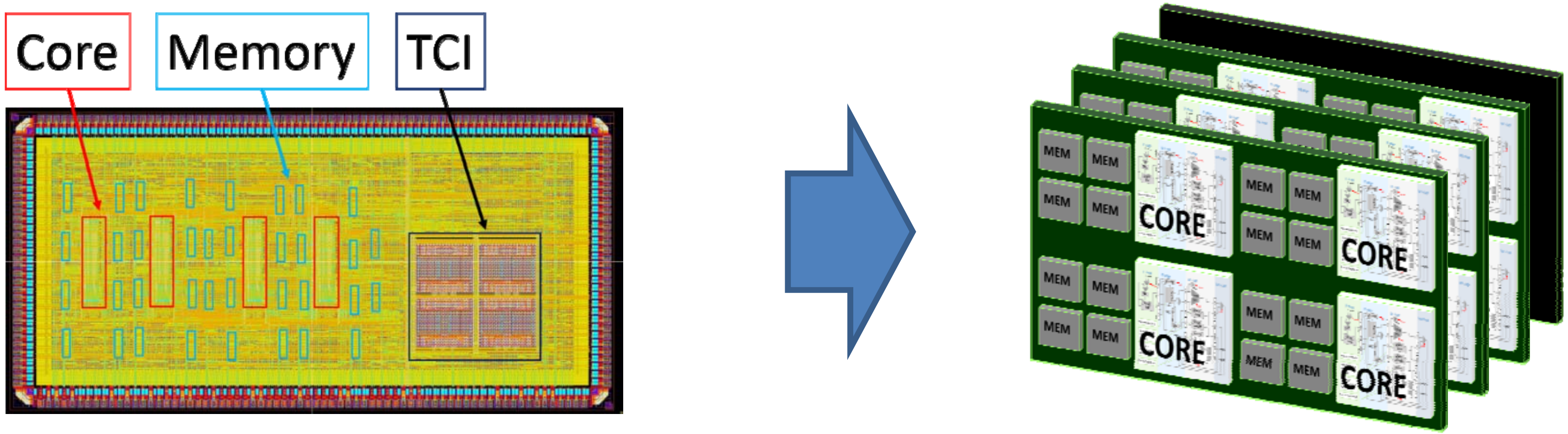
2): Communication interface for 3D-stacked chips with inductive coupling



- Co-developed with Keio Univ.
- Chip specifications
  - Renesas Electronics 65nm SOTB
  - Chip area: 3mm × 6mm
  - 4core, 4SIMD/core, 36memory modules

Evaluation of Scalability

- Benefit of 3D LSI integration with TCI
  - The number of cores can be varied easily by change the number of stacked chips
- Evaluation of scalability for prototype chip
  - Finding the best condition for maximizing energy efficiency
  - Based on simulation
- Parameters
  - Number of cores and bandwidth of off-chip memory

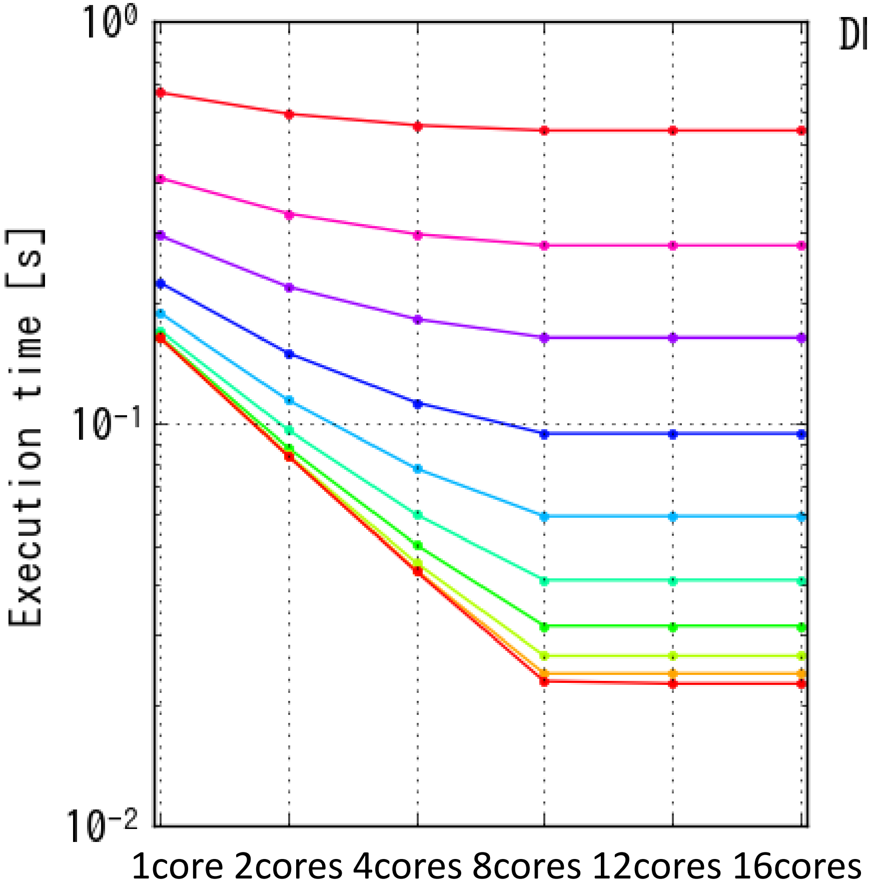


Single accelerator chip

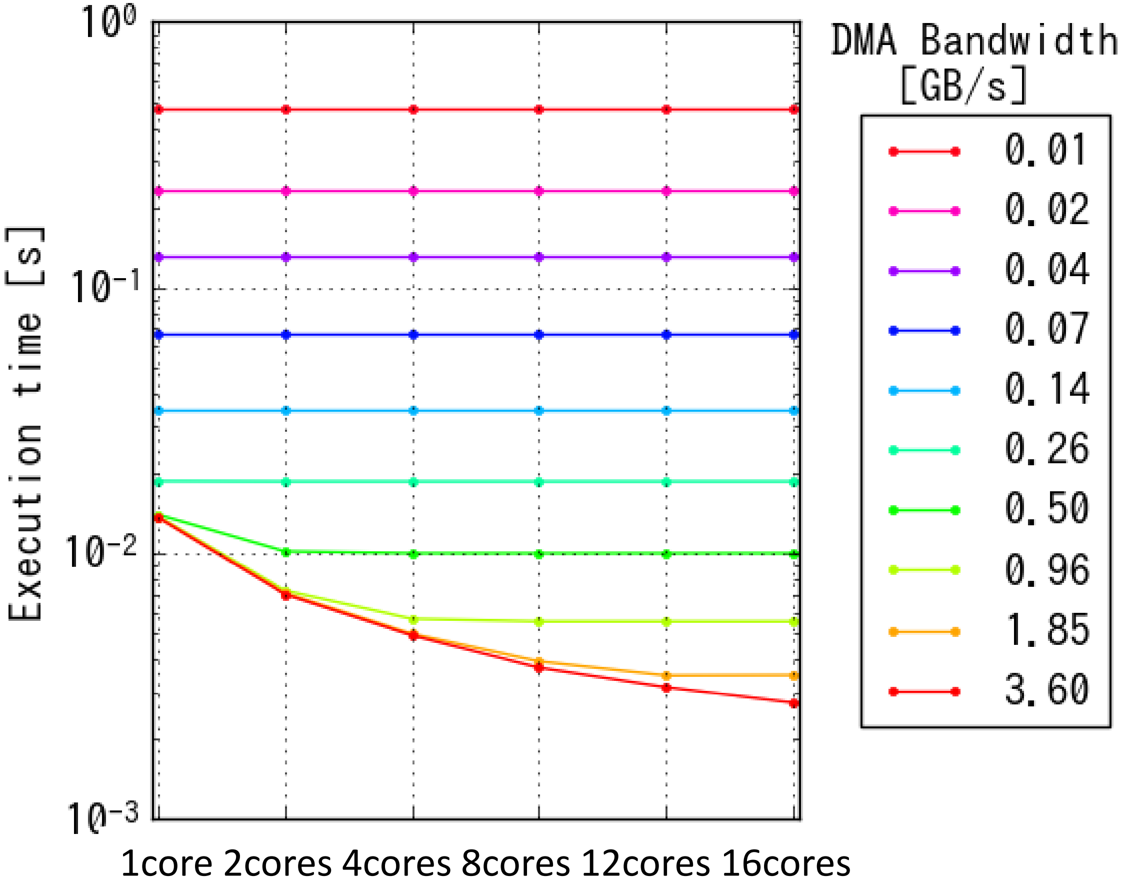
e.g.: Three accelerator chips and one general purpose chip（12 cores）

Evaluation Result：Execution time

Overall time of 7-layers CNN

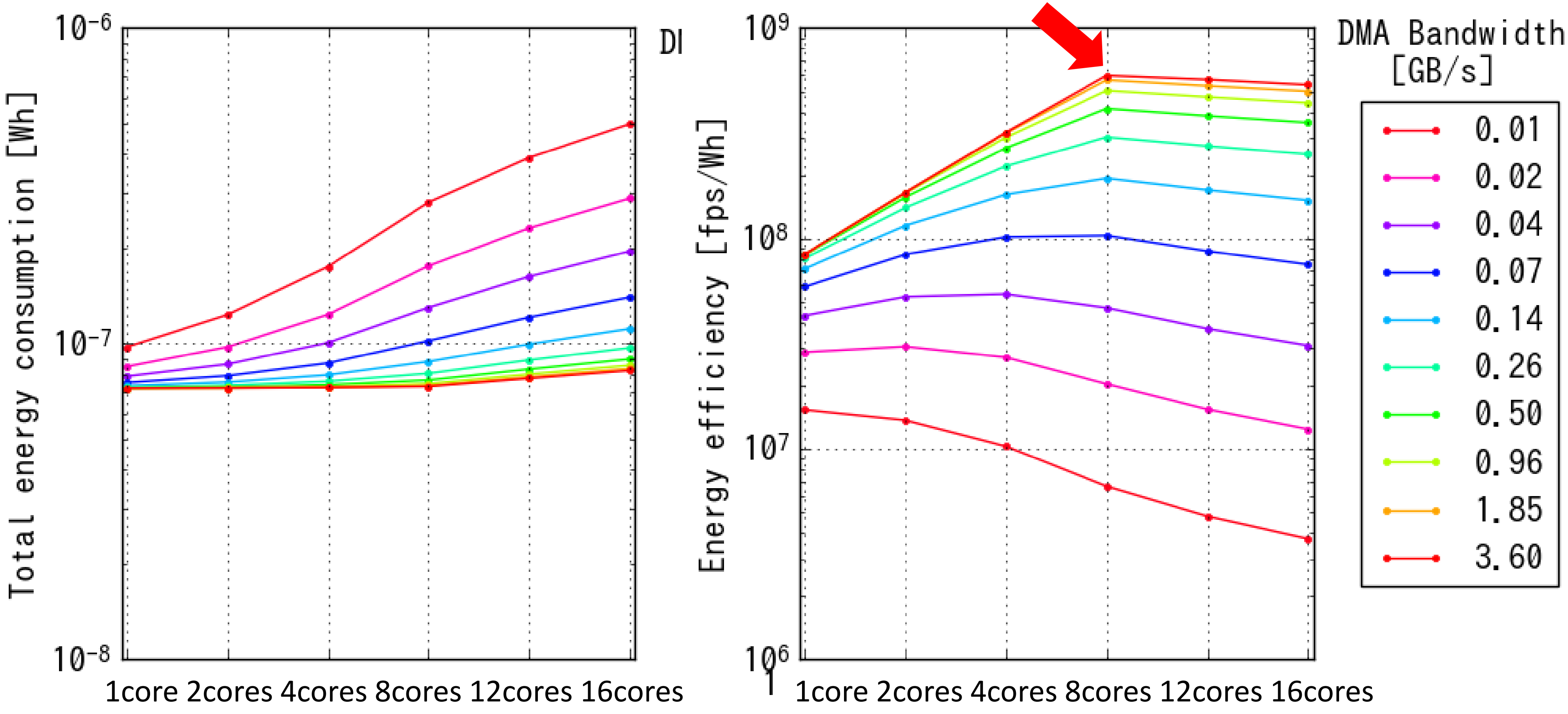


Execution time of fully connected layers part



- Sufficient with 8-cores for conv. layer, 0.5GB/s BW for FC layer
  – The bottleneck changes from FC to conv. layer

Evaluation Result: Power efficiency

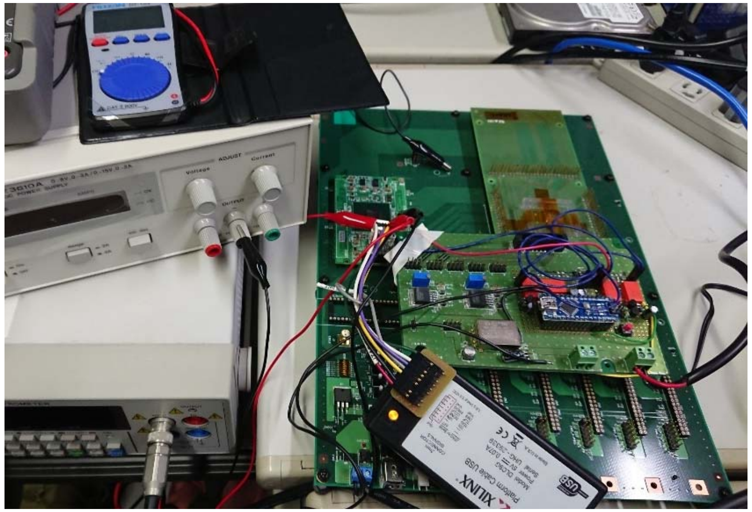

Consumed energy

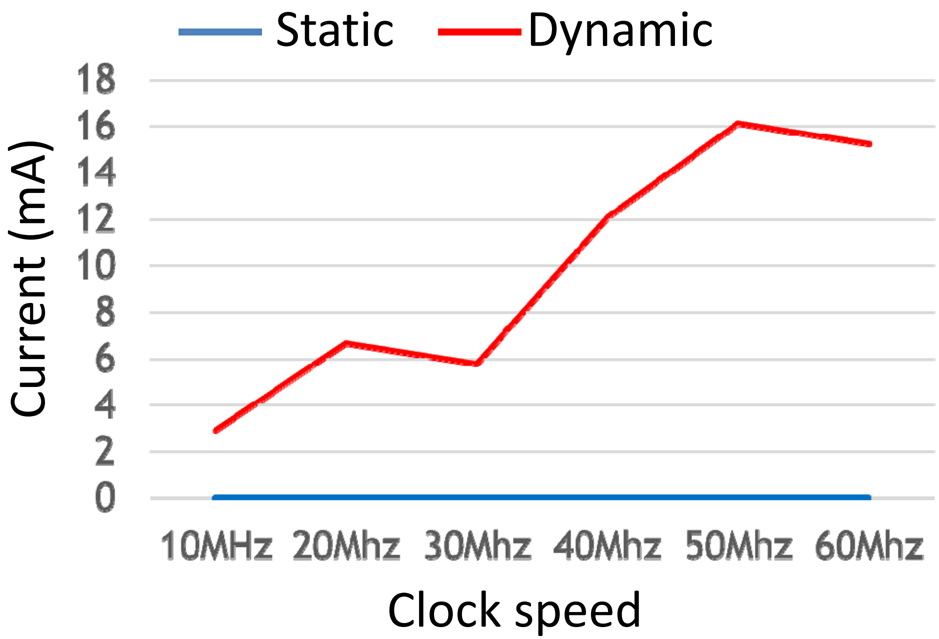
Energy efficiency

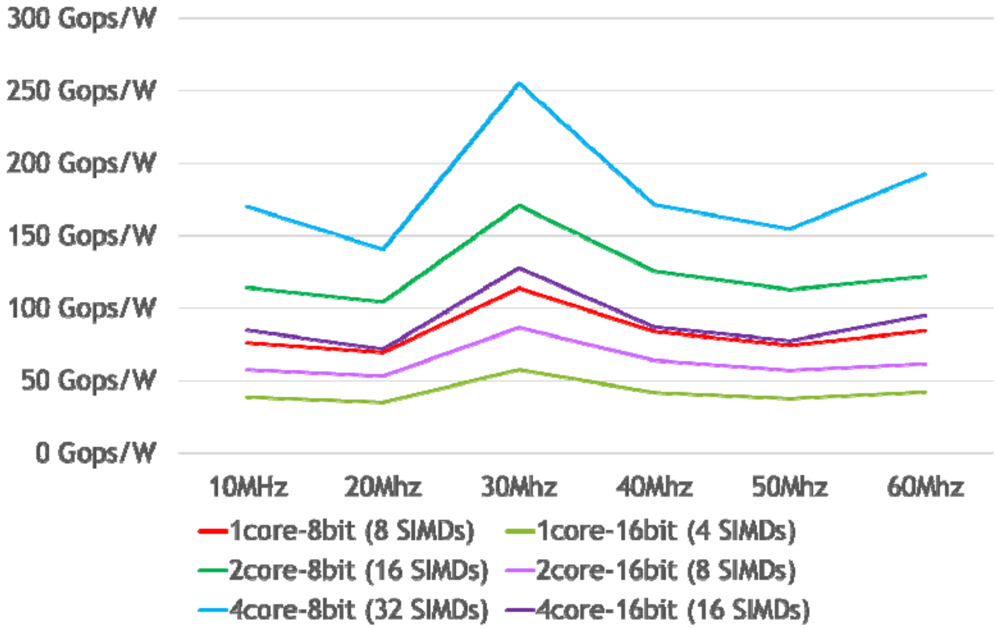
- The best configuration of maximizing energy efficiency:
  - 8-core with 0.5GB/s off-chip memory BW

Power Efficiency Evaluation on Real Chip

- Test program
  - MAC calculation by the custom instruction
- Parameters
  - Num. of cores: 1,2,4-core
  - Clock speed: 10-60MHz
  - SIMD mode：16-bit 4-SIMD, 8-bit 8-SIMD



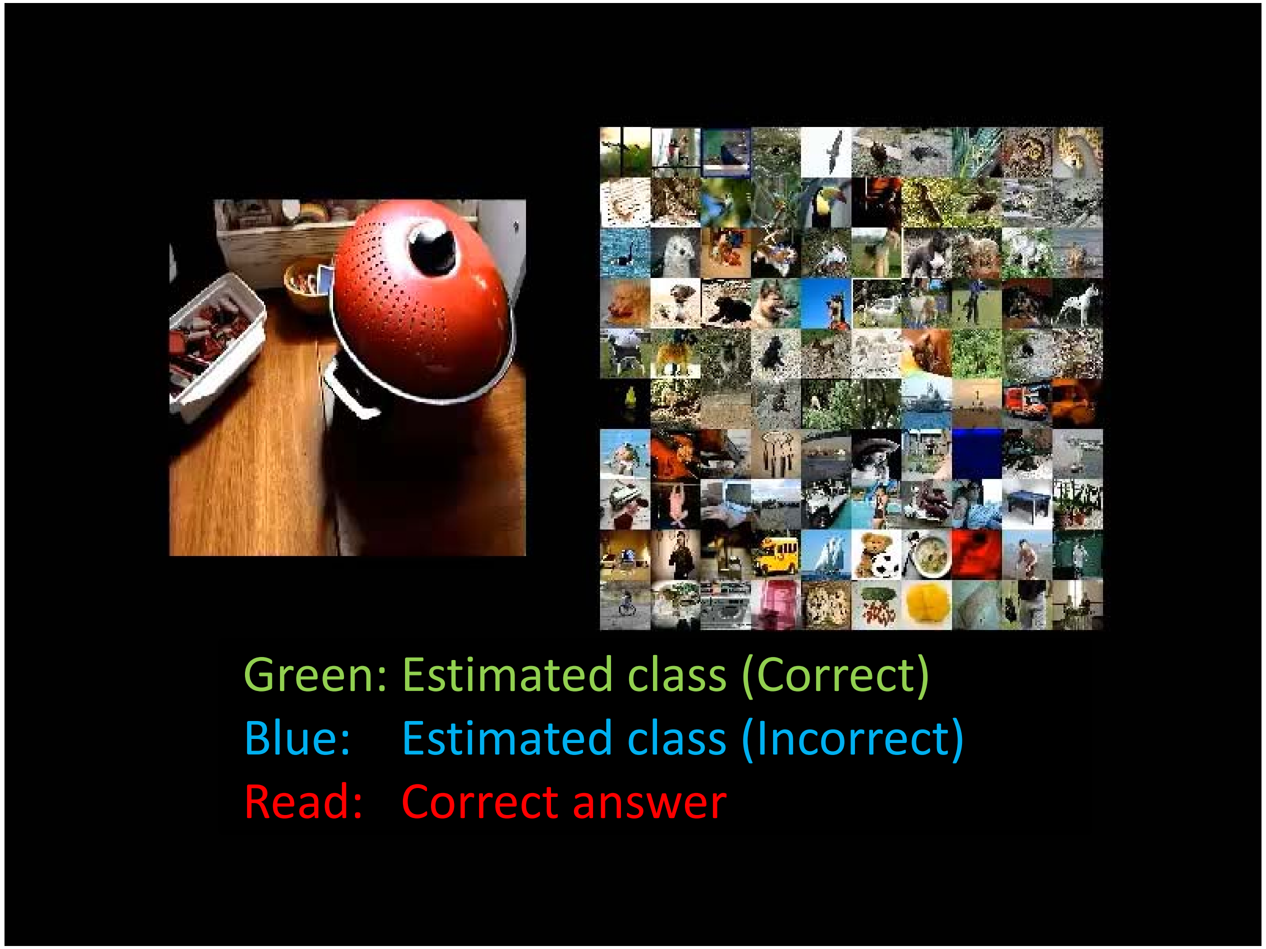




FPGA Prototyping and Demo

- Implemented AlexNet into the accelerator core on an FPGA

Zynq-7000 XC7Z020
　･ARM Cortex-A9 （667MHz）
　･DNN accelerator (1-core)
　in programmable logic
　（25/50MHz）



Green: Estimated class (Correct)
Blue: Estimated class (Incorrect)
Read: Correct answer

Summary

- SNACC (Scalable Neural Acceleration Cores with Cubic Integration)
  - DNN Accelerator Cores with 3D Integrated LSIs
  - Very power efficient and programmable accelerator
  - Taped out prototype LSI chip
  - Optimized design parameters with simulation
- Evaluation with a prototype chip
  - Achieves 256GOPS/W power efficiency
- Future work
  - Evaluation with real 3D stacked chips
  - Testing with CNN applications