High Performance Computing 4th Lecture

11/Oct/2016

Yuya Kobayashi

Reviewed Paper

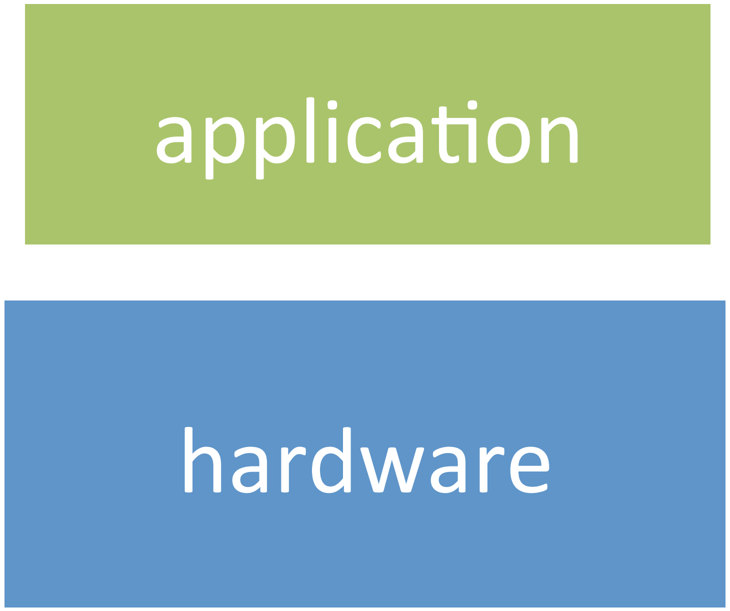
- Suyog Gupta, Ankur Agrawal, Kailash Gopalakrishnan, Pritish Narayanan. **Deep Learning with Limited Numerical Precision.** Proceedings of the 32nd International Conference on Machine Learning (ICML-15)

Background

- Natural error resiliency of neural network (NN) [Bottou & Bousquet, 2007].
  - In the presence of statistical approximation and estimation errors, high-precision computing is not necessary for DNN.
- Large scale systems specialized for DNN do not utilize natural error resiliency, except for Asynchronous SGD.

- This paper shows a performance of NN and a prototype hardware with 16-bit fixed point number.
  - Fixed point compute units are faster, consume less resources and power.
  - A data is of smaller data size.

Idea of system



mitigating impacts of error

low-precision fixed point arithmetic

- simpler component
- smaller memory

Limited Precision Arithmetic
fixed-point number type

bit-length for integer part

<IL, FL>

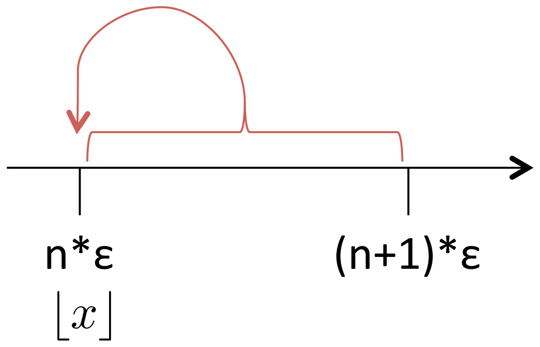
bit-length for fraction part

This notation provides how long bit is assigned to integer part and fraction part in a decimal number.

Rounding Mode

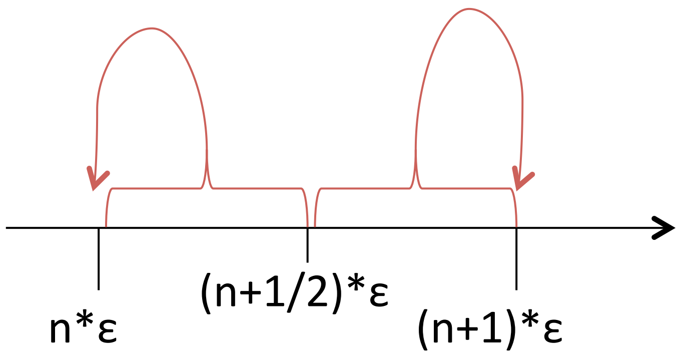
$\varepsilon = 2^{-FL}$ (minimum value in <FI, FL>)

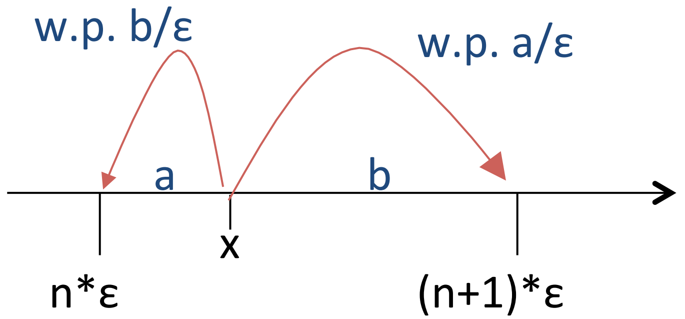
$\lfloor x \rfloor$ = max{ y | (y / ε) is integer, y <= x}



$Round\,(x, \langle \texttt{IL}, \texttt{FL} \rangle)$

- Round-to-nearest(RtN)
- Stochastic rounding





Rounding Mode

If a calculated result is outside the range of <IL, FL>, then we saturate it to upper or lower bound of <IL, FL>.

$$Convert\,(x, \langle \mathtt{IL}, \mathtt{FL} \rangle) = \begin{cases} -2^{\mathtt{IL}-1} & \text{if } x \leq -2^{\mathtt{IL}-1} \\ 2^{\mathtt{IL}-1} - 2^{-\mathtt{FL}} & \text{if } x \geq 2^{\mathtt{IL}-1} - 2^{-\mathtt{FL}} \\ Round(x, \langle \mathtt{IL}, \mathtt{FL} \rangle) & \text{otherwise} \end{cases} \tag{1}$$

Multiply and accumulate (MACC) operation

Calculating $\mathbf{c}_0 = \mathbf{a} \cdot \mathbf{b}$ by 2 steps.

– **a**, **b** : <*IL*, *FL*> fixed-point number *d*-dimension vector

– $\mathbf{c}_0$: $\langle \tilde{\mathtt{IL}}, \tilde{\mathtt{IF}} \rangle$ fixed-point number

1. Compute $z = \sum_{i=1}^{d} a_i b_i$
   – $a_i\ b_i$: ⟨2 IL, 2 FL⟩ fixed–point
   – z : $\{\log_2 d + 2\,(\mathrm{IL} + \mathrm{FL})\}$ bit length fixed–point
2. Convert: $c_0 = Convert(z, \langle \tilde{\mathtt{IL}}, \tilde{\mathtt{IF}} \rangle)$

Multiply and accumulate (MACC) operation

- advantage of this 2-steps methodology
  - easy to implement with FPGA
  - one rounding per one multiplying operation
  - easy to simulate in CPU/GPU, BLAS library

Evaluation

Going to evaluate error of network with 16-bit fixed point arithmetic by comparing with 32-bit floating point one.

- Network
  - DNN
  - Convolutional Neural Network(CNN)
- Data set
  - MNIST
  - CIFAR10

Evaluation

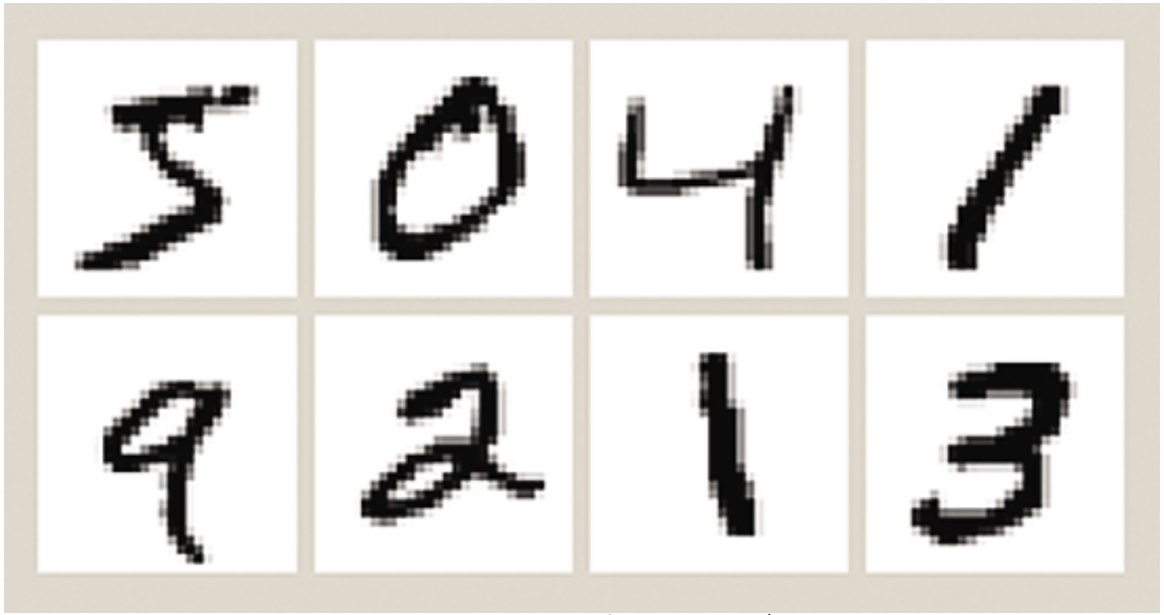
- Weights and Biases in network are to be initialized randomly.
- HyperParameters (e.g. number of layer, momentum, learning rate, ...) is the same between baseline experiment and 16-bit fixed point one.

- Fixed-point number is represented in 16 bits.

error in DNN for MNIST

MNIST

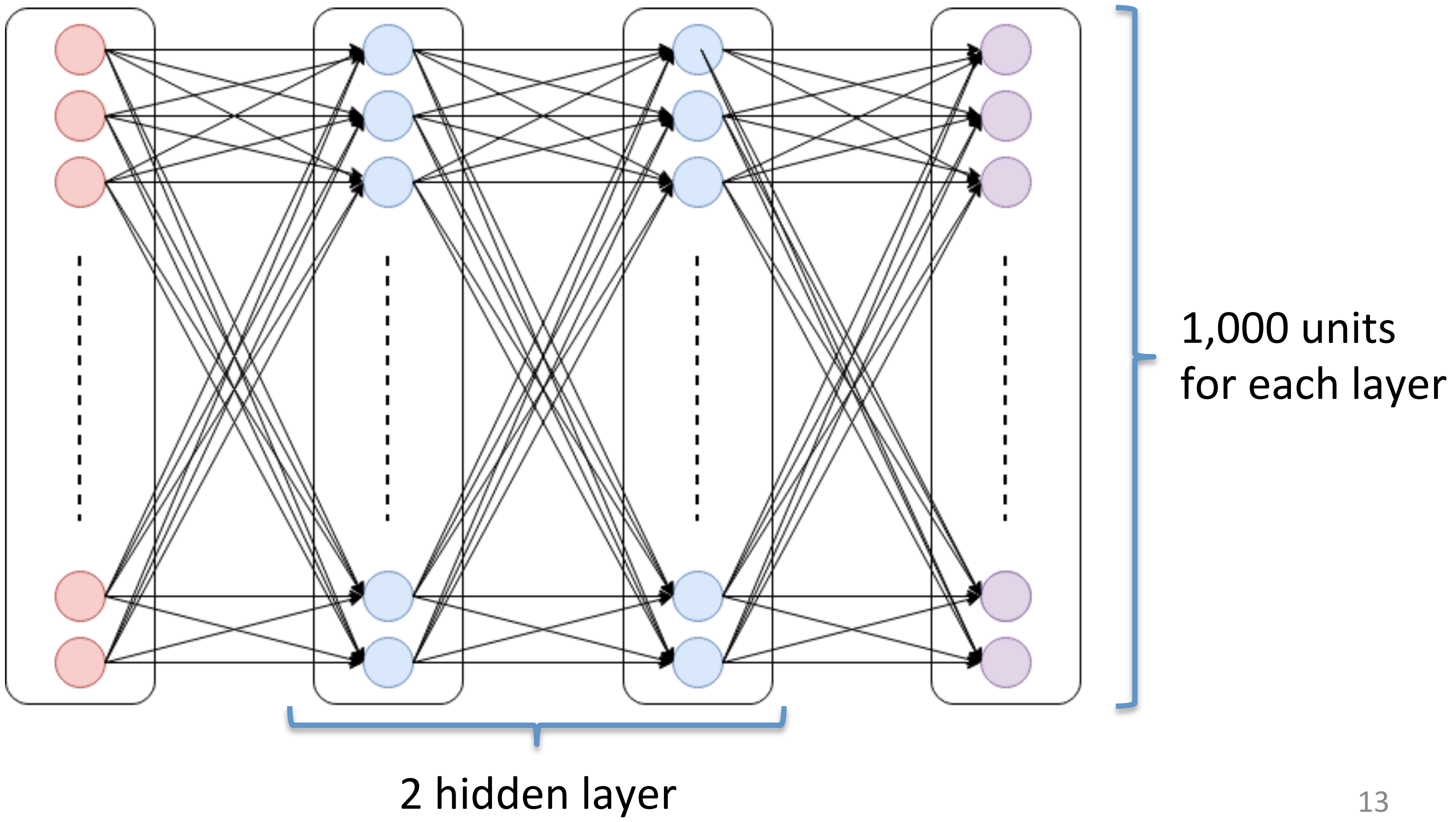
- 60,000 training images/ 10,000 test images
- 28 × 28 pixels in a image
- Each pixel in the images has a value in [0,1].



from テストの実行 - MNIST 画像認識データ セットに取り組む (https://msdn.microsoft.com/ja-jp/magazine/dn745868.aspx)

error in DNN for MNIST

- Fully connected network
- Each weight is initialized randomly from *N*(0, 0.01). The bias vector initialized to 0.



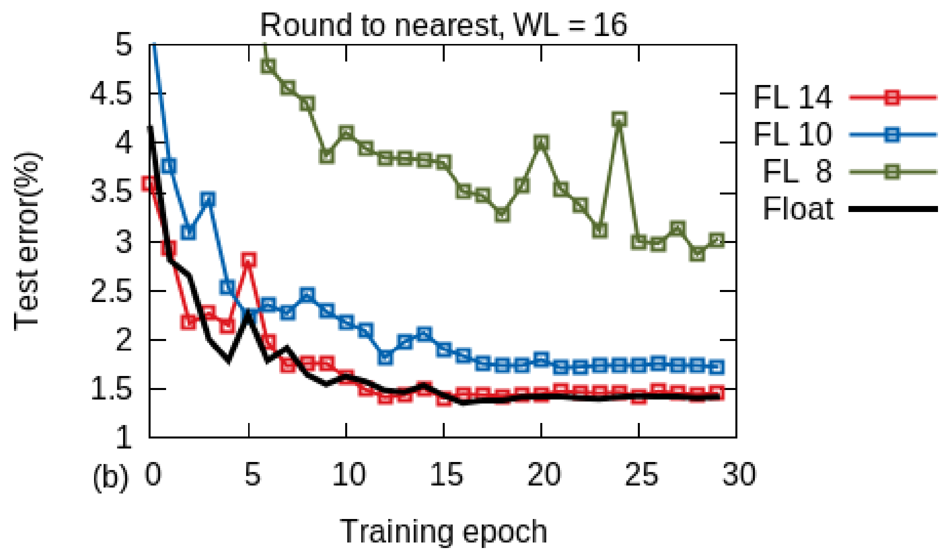
error in DNN for MNIST

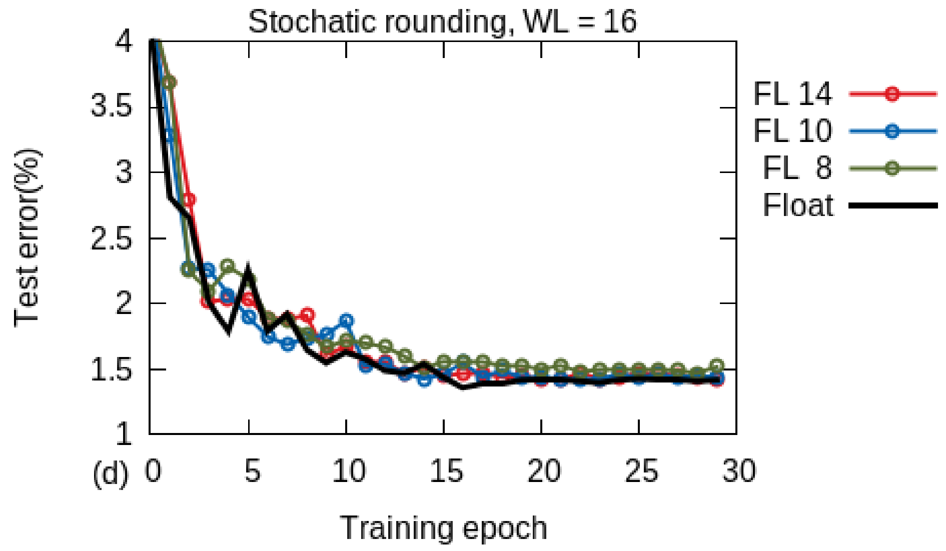
Training DNN by minibatch SGD method.

- the minibatch size is 100

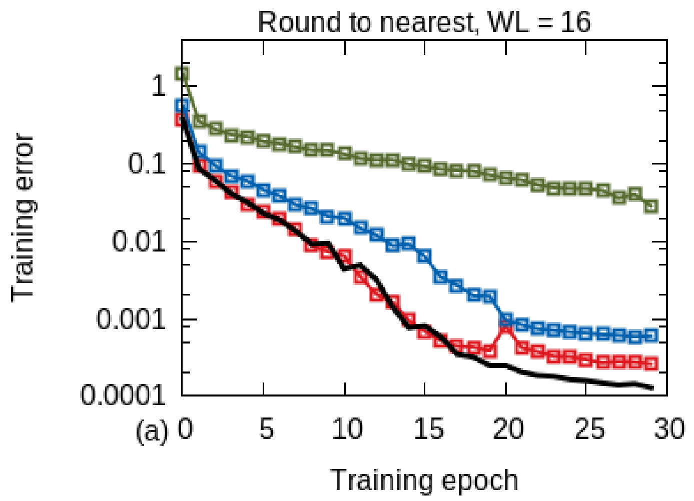
error in DNN for MNIST

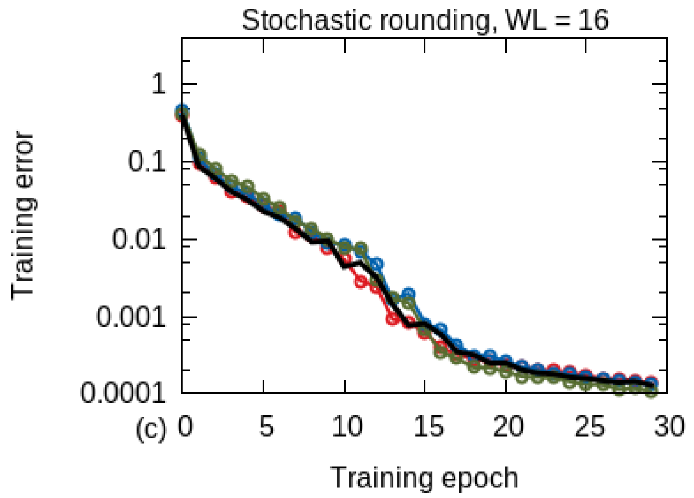
- Precision of fixed point in which test error is close to the one with float is <2,14> in RtN scheme, or <8,8> in Stochastic rounding scheme.
  - RtN lose gradient information more readily, then some weights are not updated.





error in DNN for MNIST

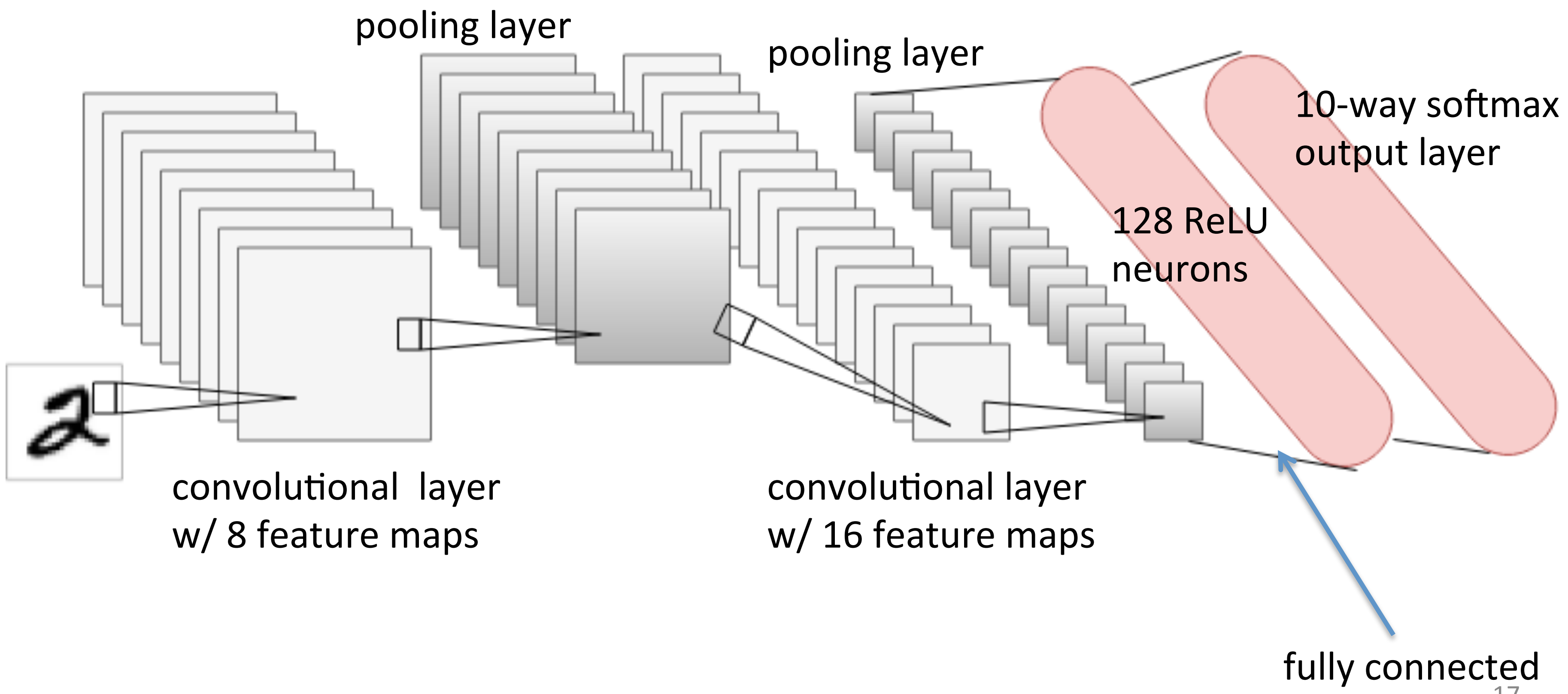




error in CNN for MNIST

The network is similar to LeNet-5.

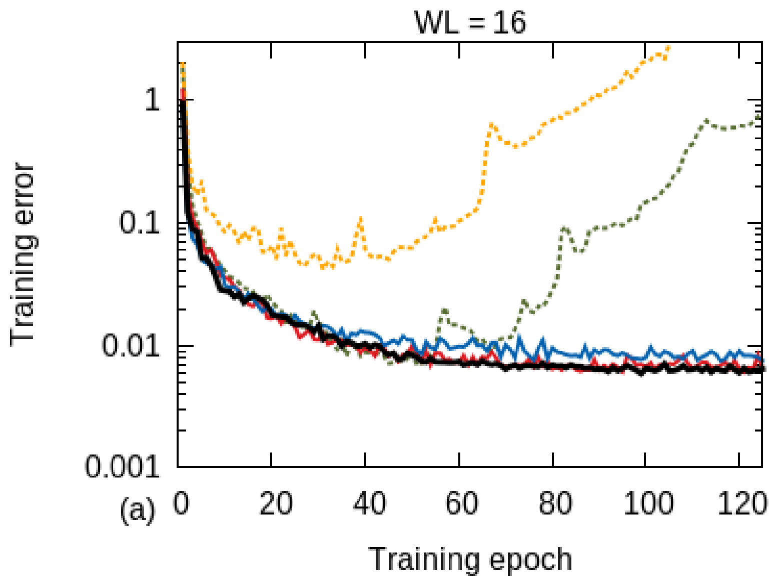
- 5*5 filter, 2*2 non-overlapped pooling

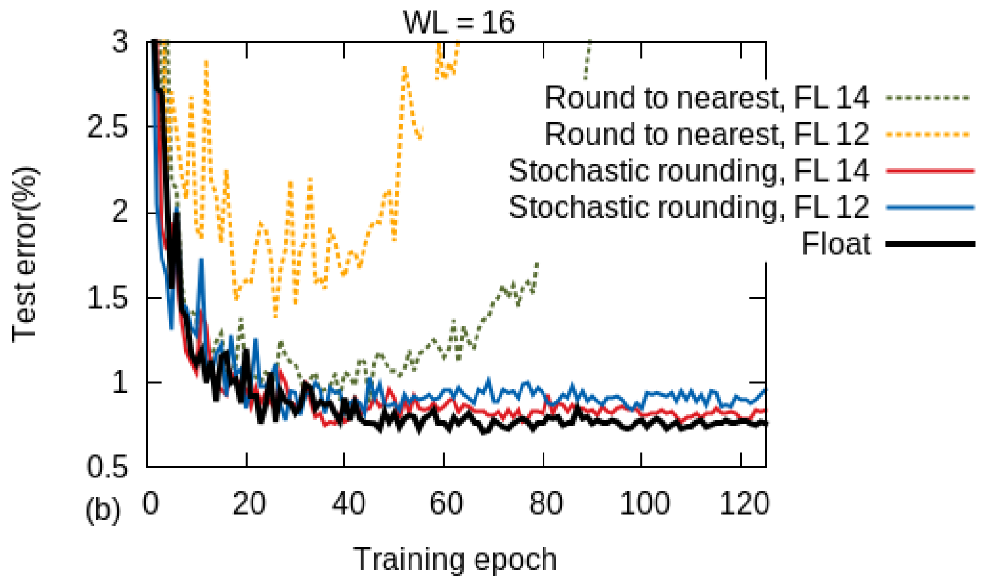

pooling layer
pooling layer
10-way softmax output layer
128 ReLU neurons
convolutional layer w/ 8 feature maps
convolutional layer w/ 16 feature maps
fully connected

error in CNN for MNIST

- parameter
  - learning rate = $0.1 * (0.95)^{(\# \text{ of completed epoch})}$
  - momentum = 0.9
  - weight decay = 0.0005
- Output from layer is represented in <6,10> fixed-point.
  - If IL < 6, outputs from convolutional layers are higher than a number the fixed-point can represent.

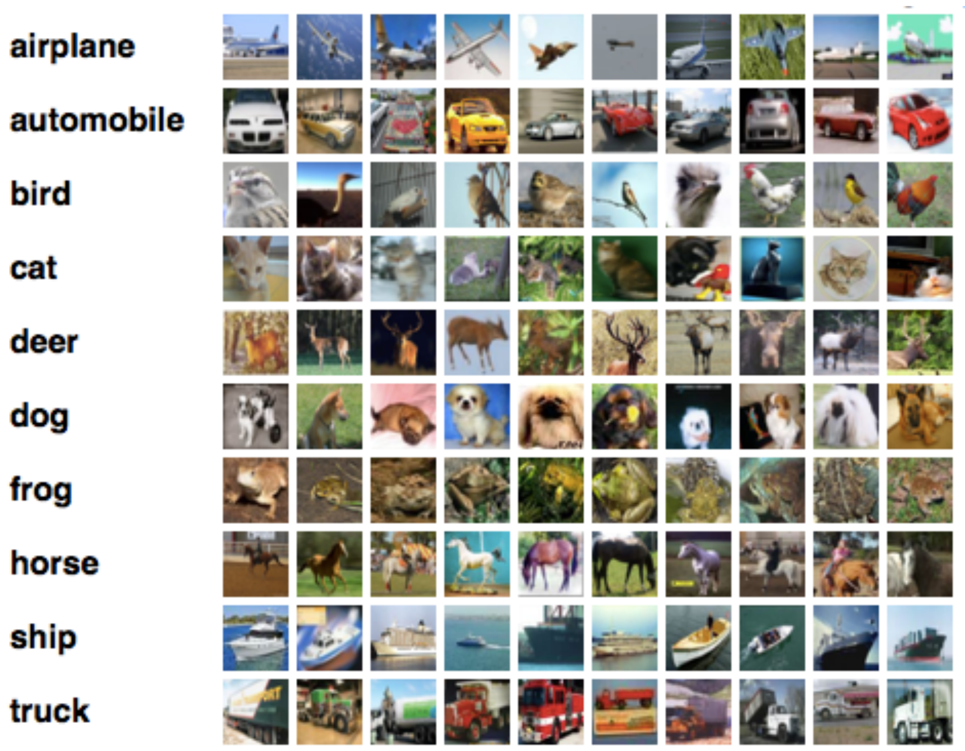
error in CNN for MNIST





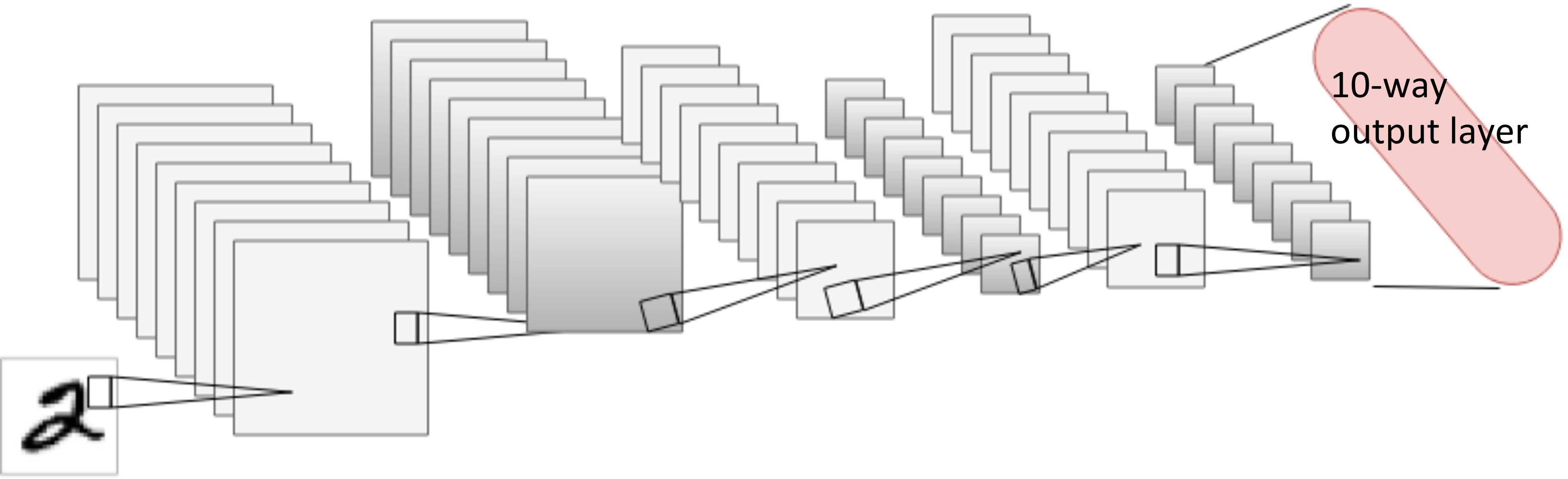
error in CNN for CIFAR10

- The CIFAR-10 dataset consists of 60000 32x32 color images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images.
- The image RGB values are scaled to [0,1] for the evaluation.



error in CNN for CIFAR10

- 64 5*5 filters
- 3*3 pooling window using a stride of 2



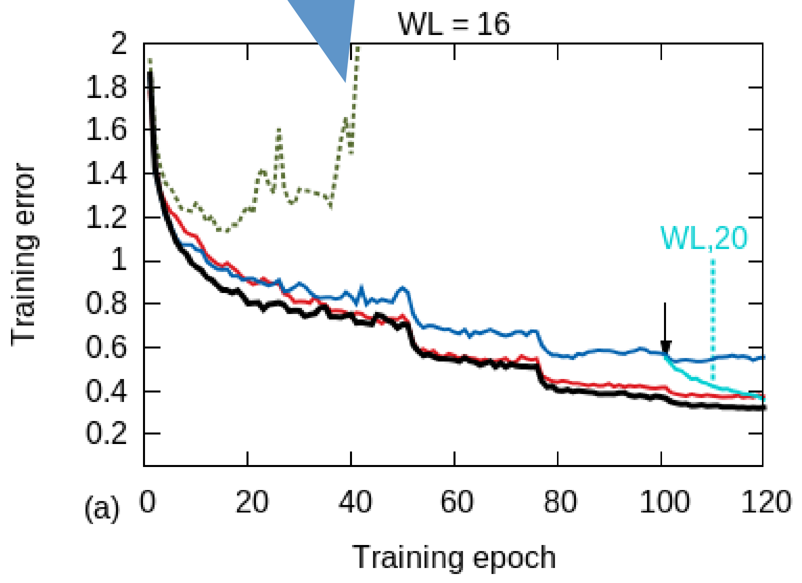
error in CNN for CIFAR10

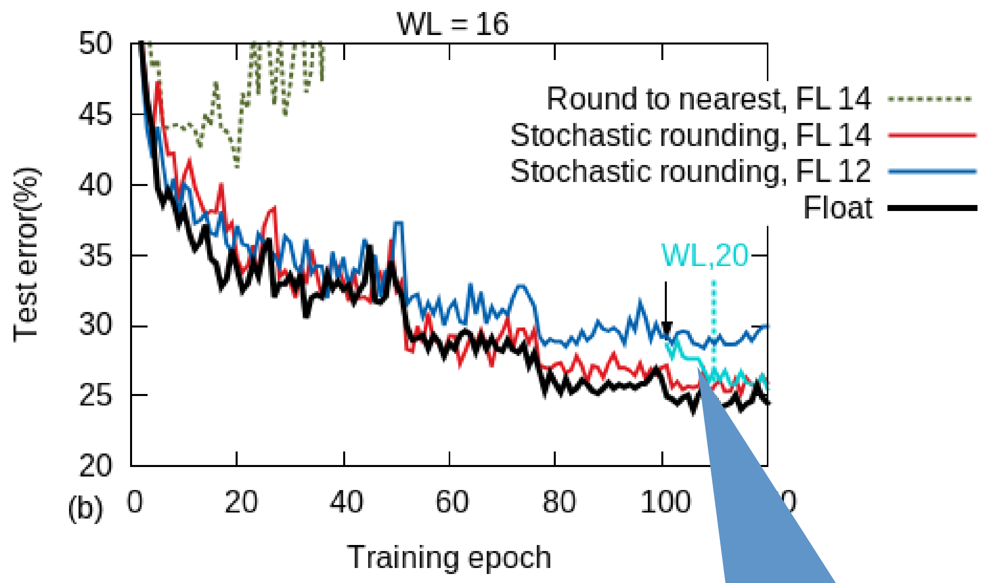
- Parameter
  - learning rate is 0.01 (at begin), 0.005(after 50 epoch), 0.0025(after 75 epoch), 0.00125(after 100 epoch).

- Outputs from layers are represented in the <4,12> format.

error in CNN for CIFAR10

RtN scheme results in divergence





Changing the precision to <4, 16> improves the network performance

Hardware Prototyping

- FPGA-based hardware accelerator for matrix-matrix multiplication
  - FPGA contains DSP units that are well-suit to implement fixed point arithmetic.
  - FPGA has potential in performance and power efficiency.

Components of the prototype

- Xilinx Kintex325T FPGA
  - 840 DSP multiply-accumulate unit
  - 2MB on-chip lock RAM
    - cache of matrix data
- 8GB DDR3
  - to store matrix data
- PCIe Bus between the FPGA and the Host
  - The data bandwidth between the off-chip DDR3 memory and the FPGA is 6.4 (GB/s) .

Inside of the accelerator

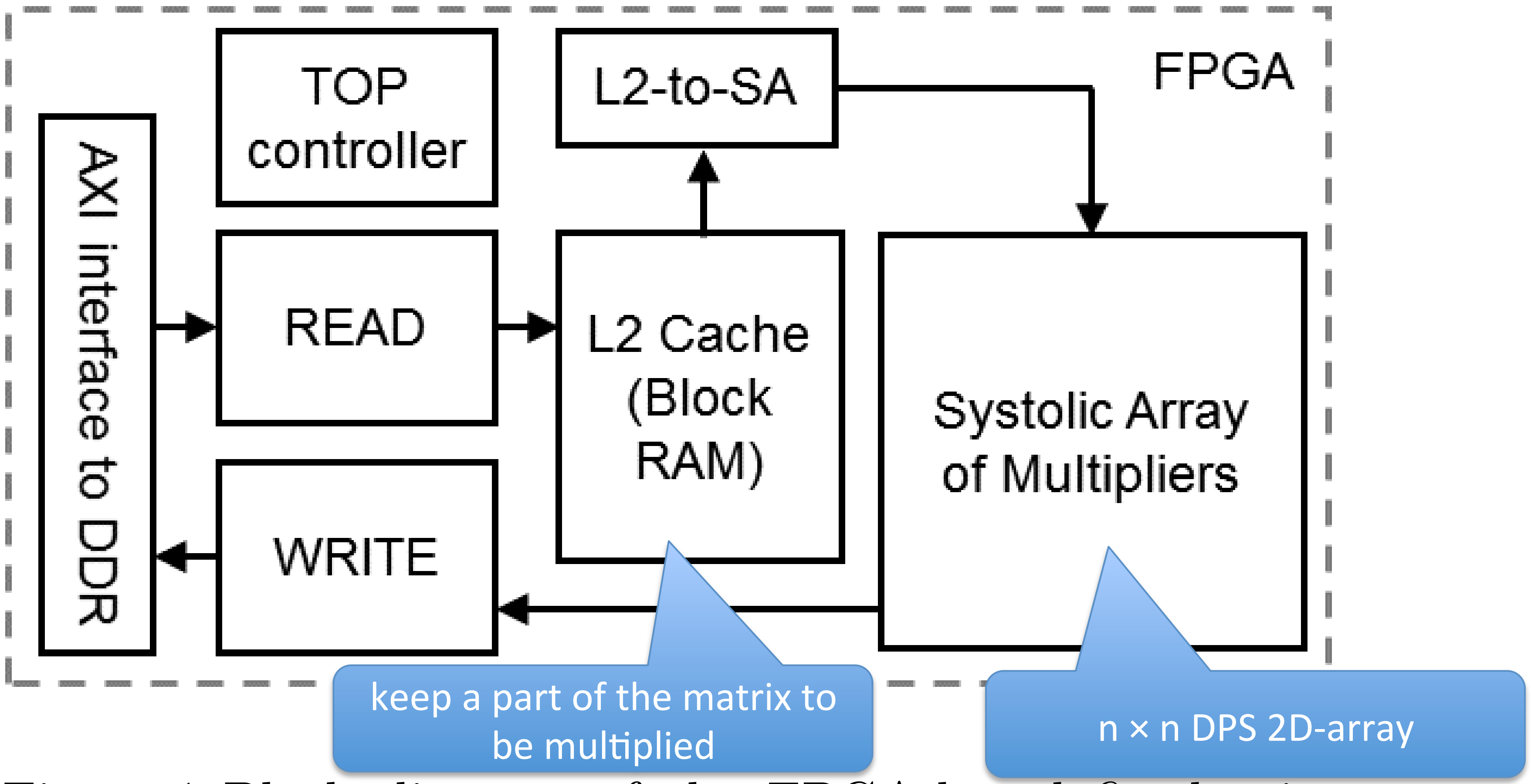
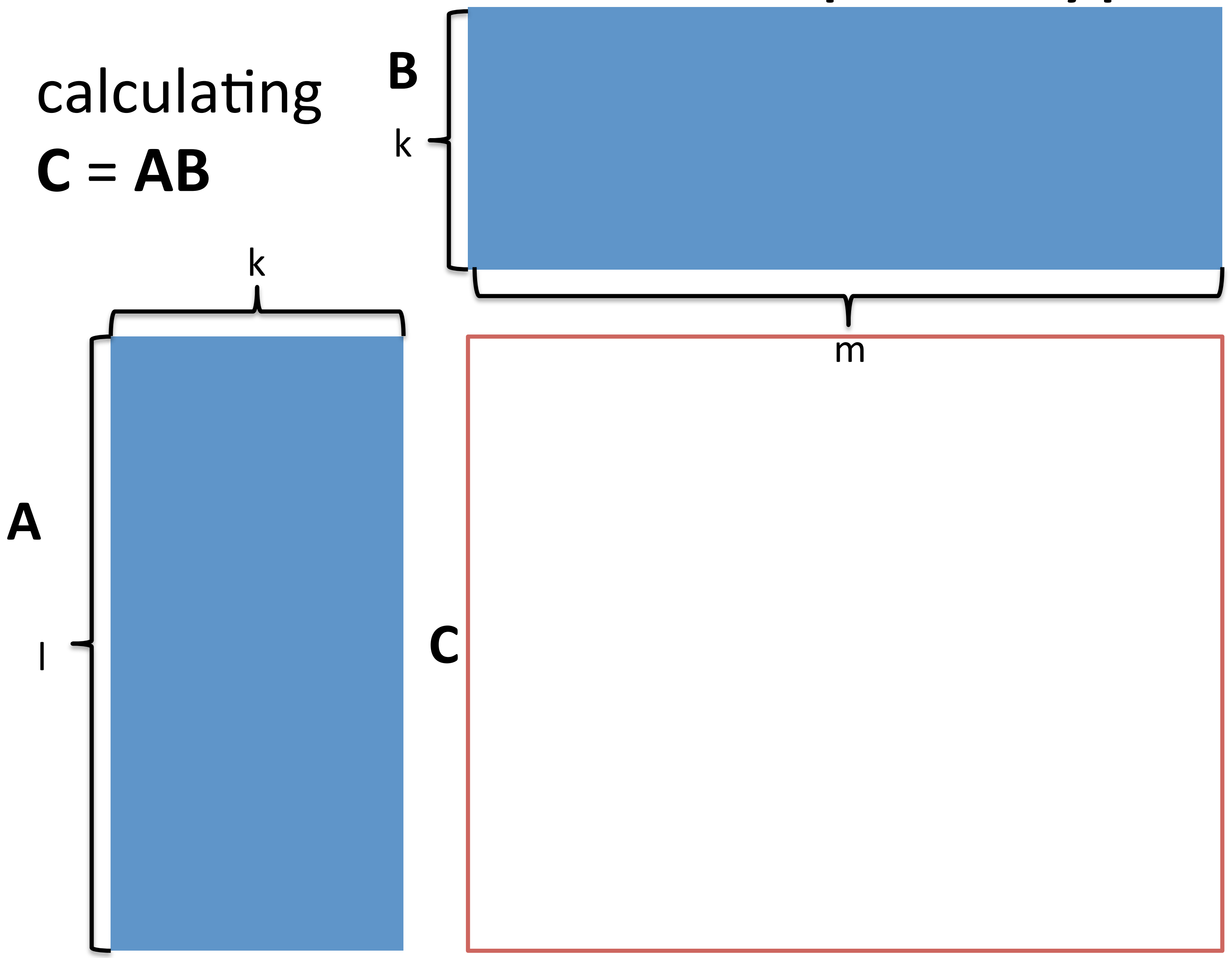


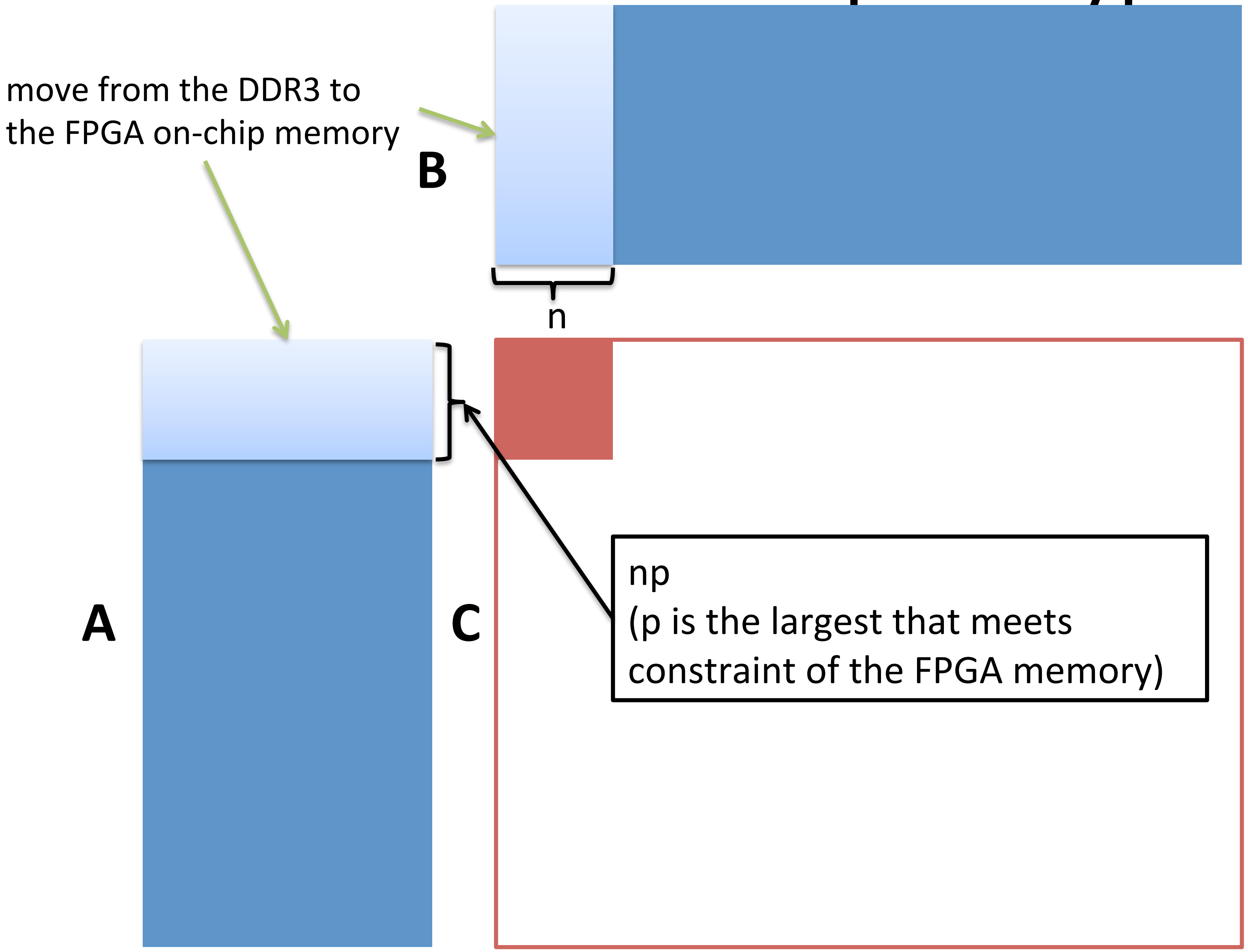
Figure 4. Block diagram of the FPGA-based fixed-point matrix multiplier.

calculation in the prototype

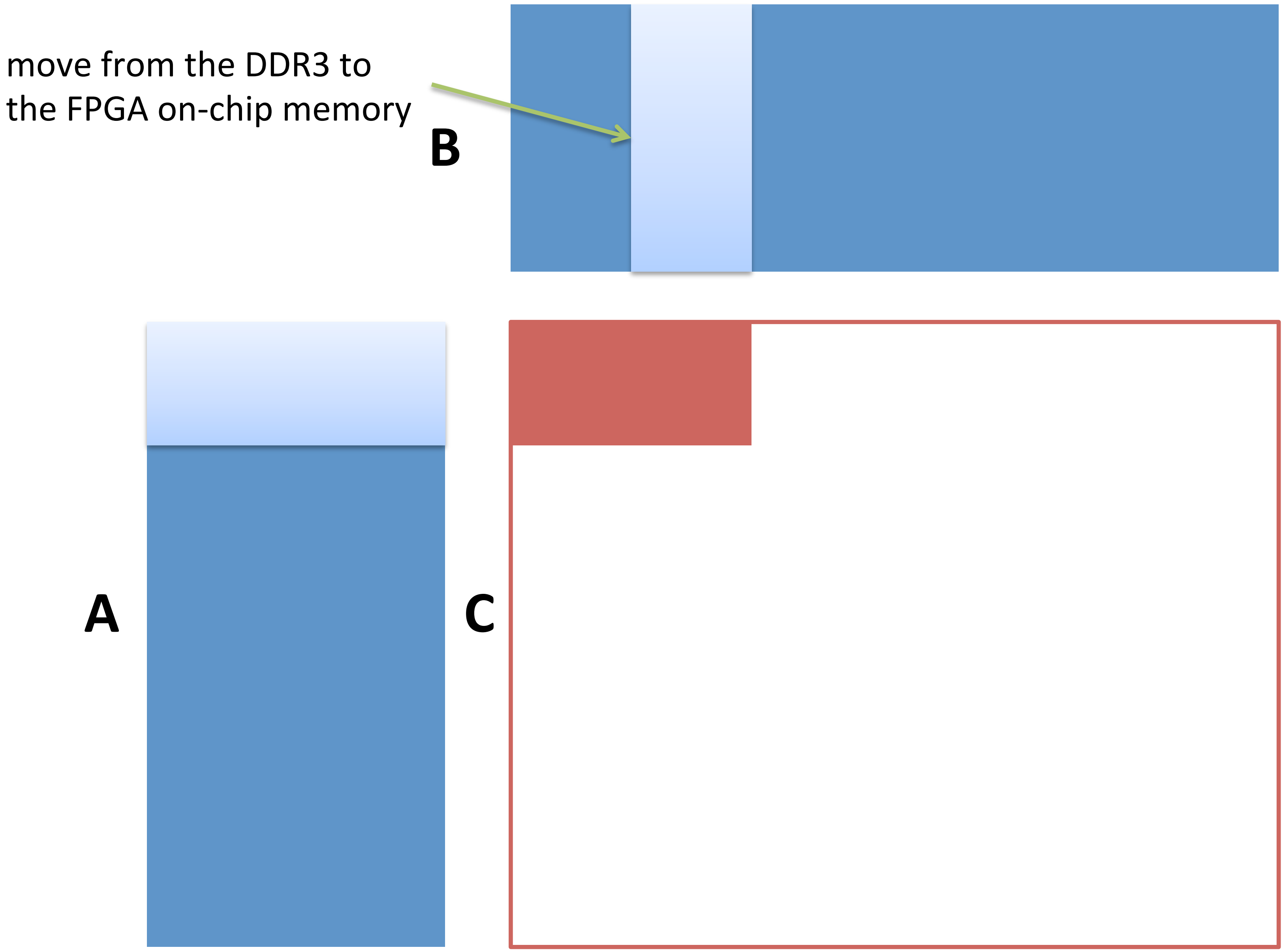
- calculating
  C = AB



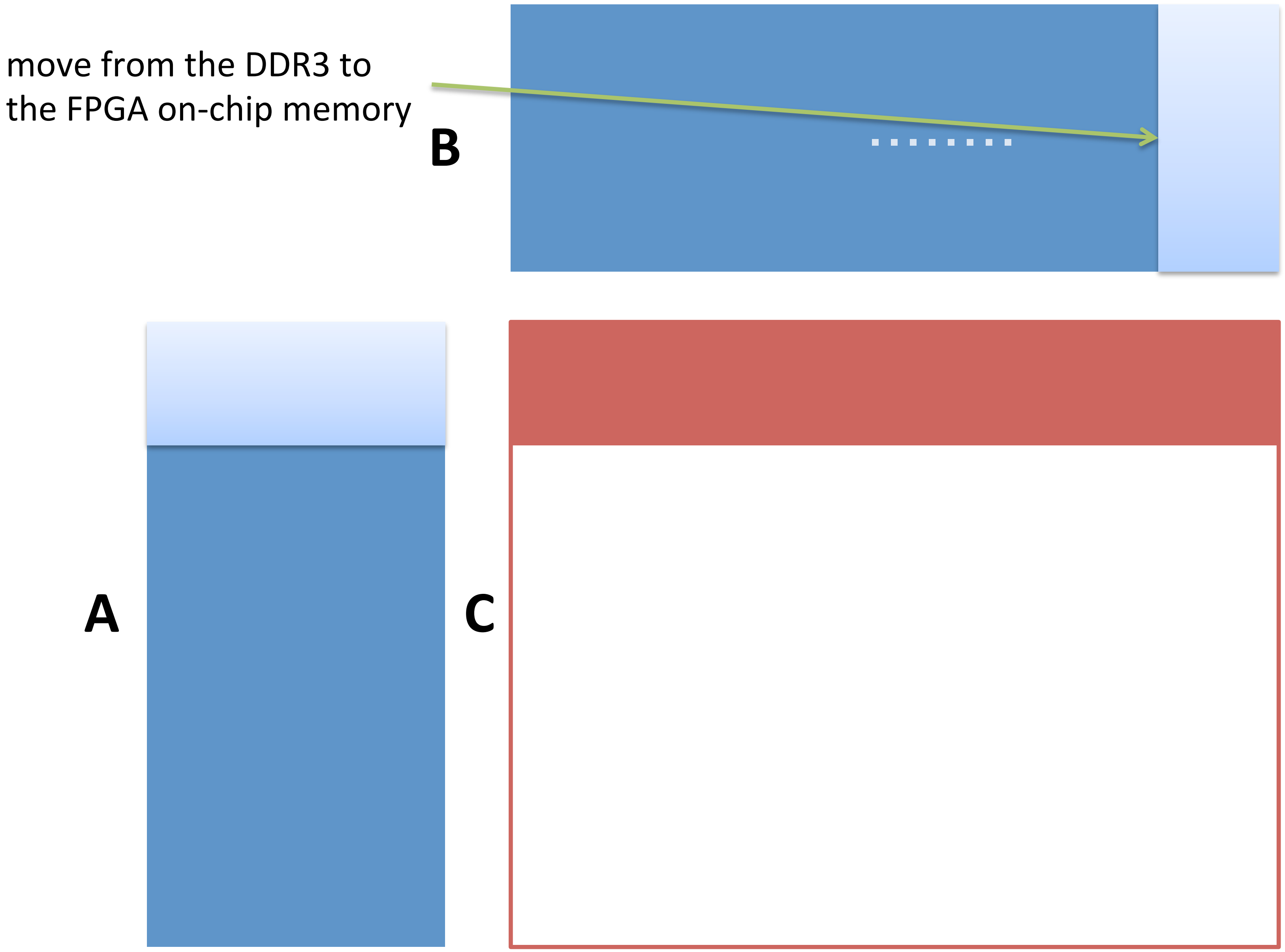
calculation in the prototype



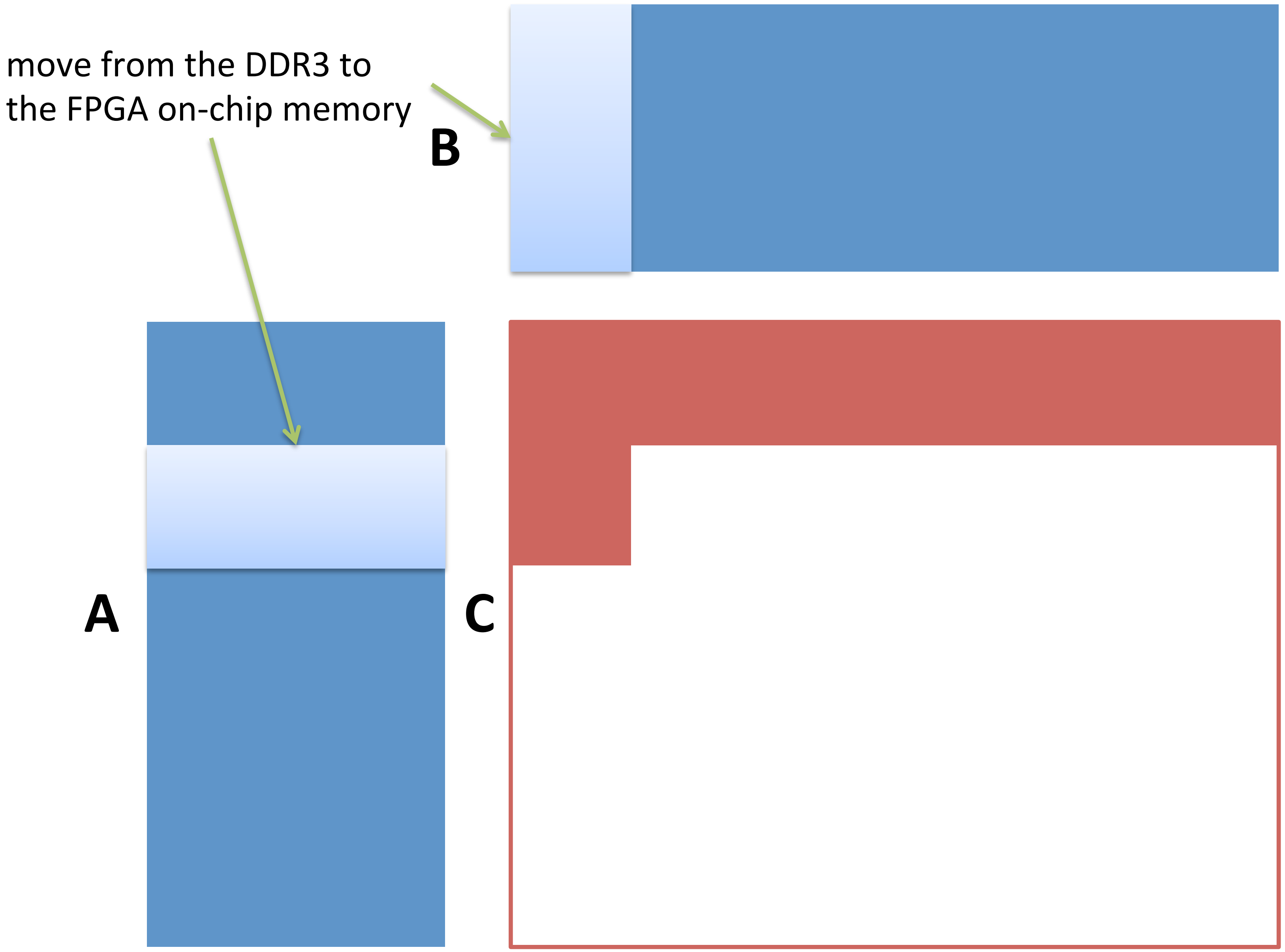
calculation in the prototype



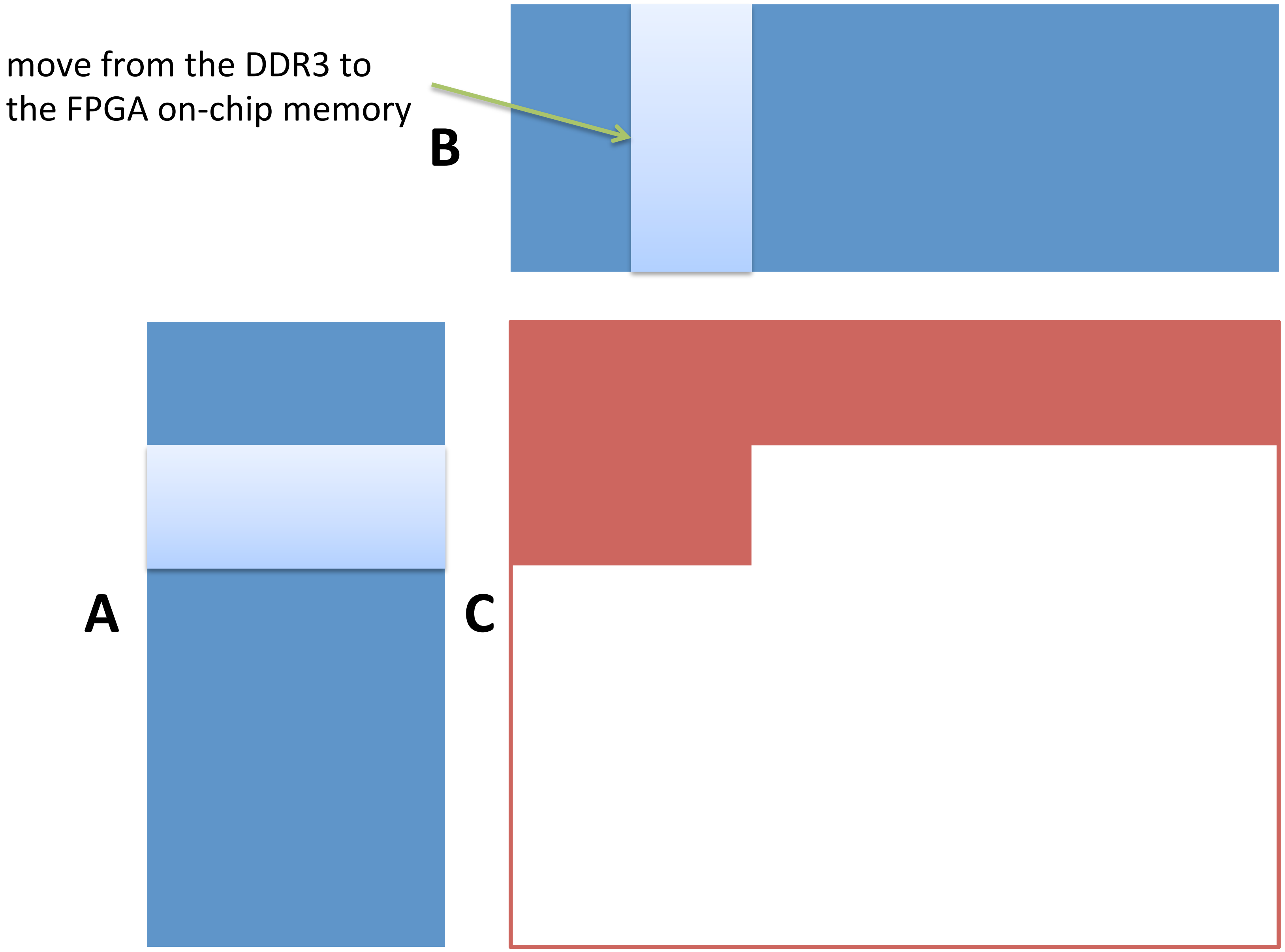
calculation in the prototype



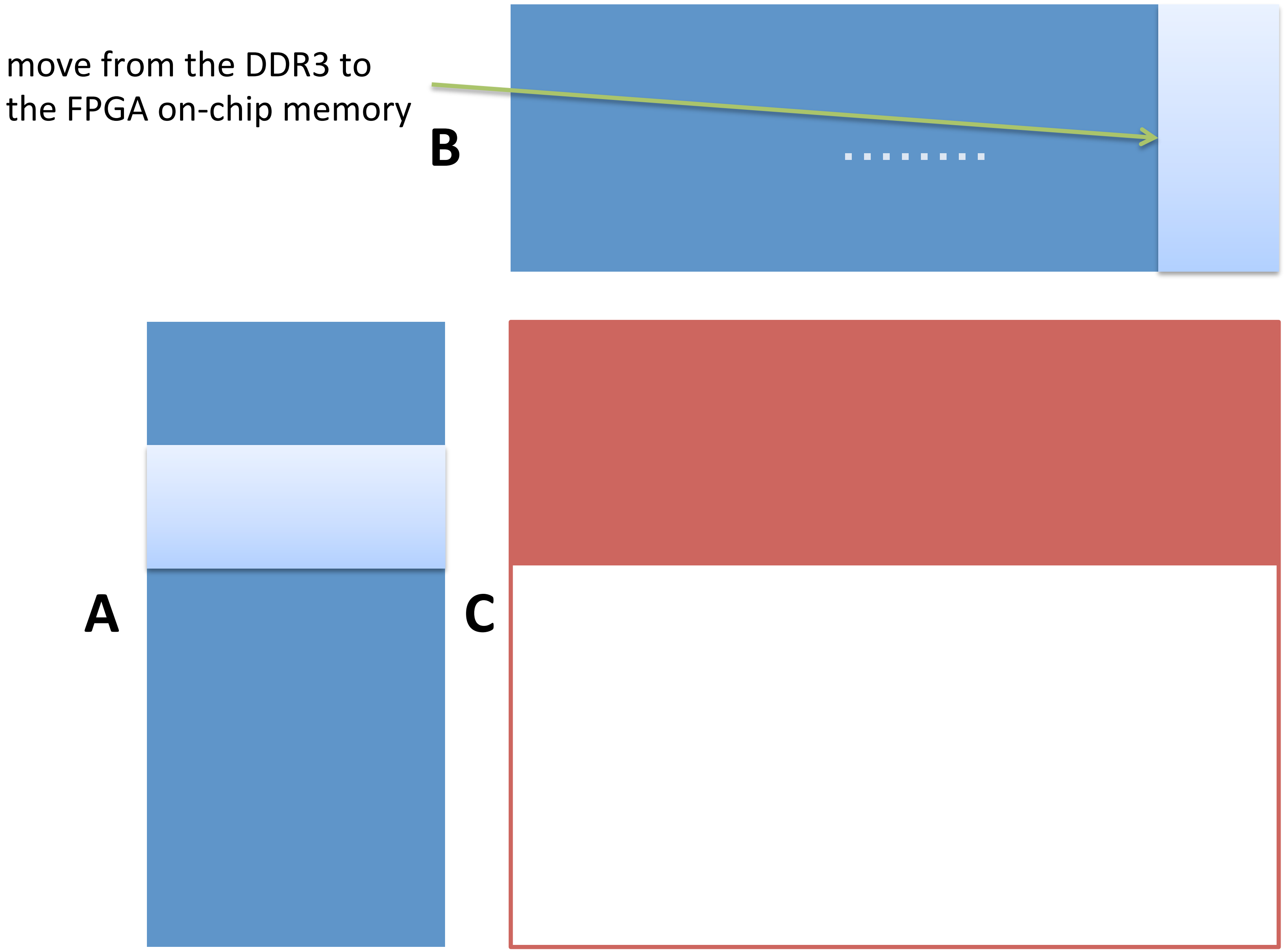
calculation in the prototype



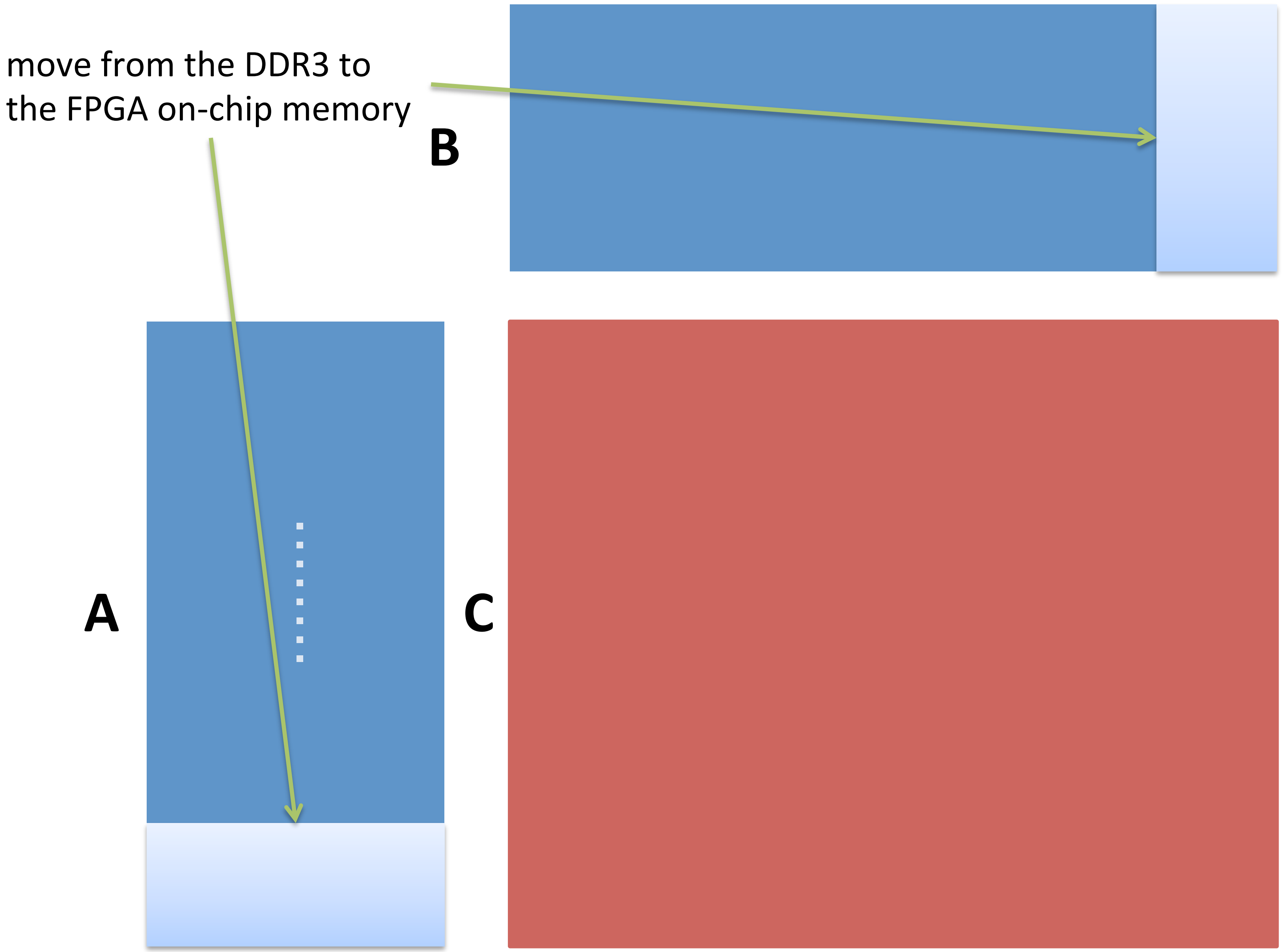
calculation in the prototype



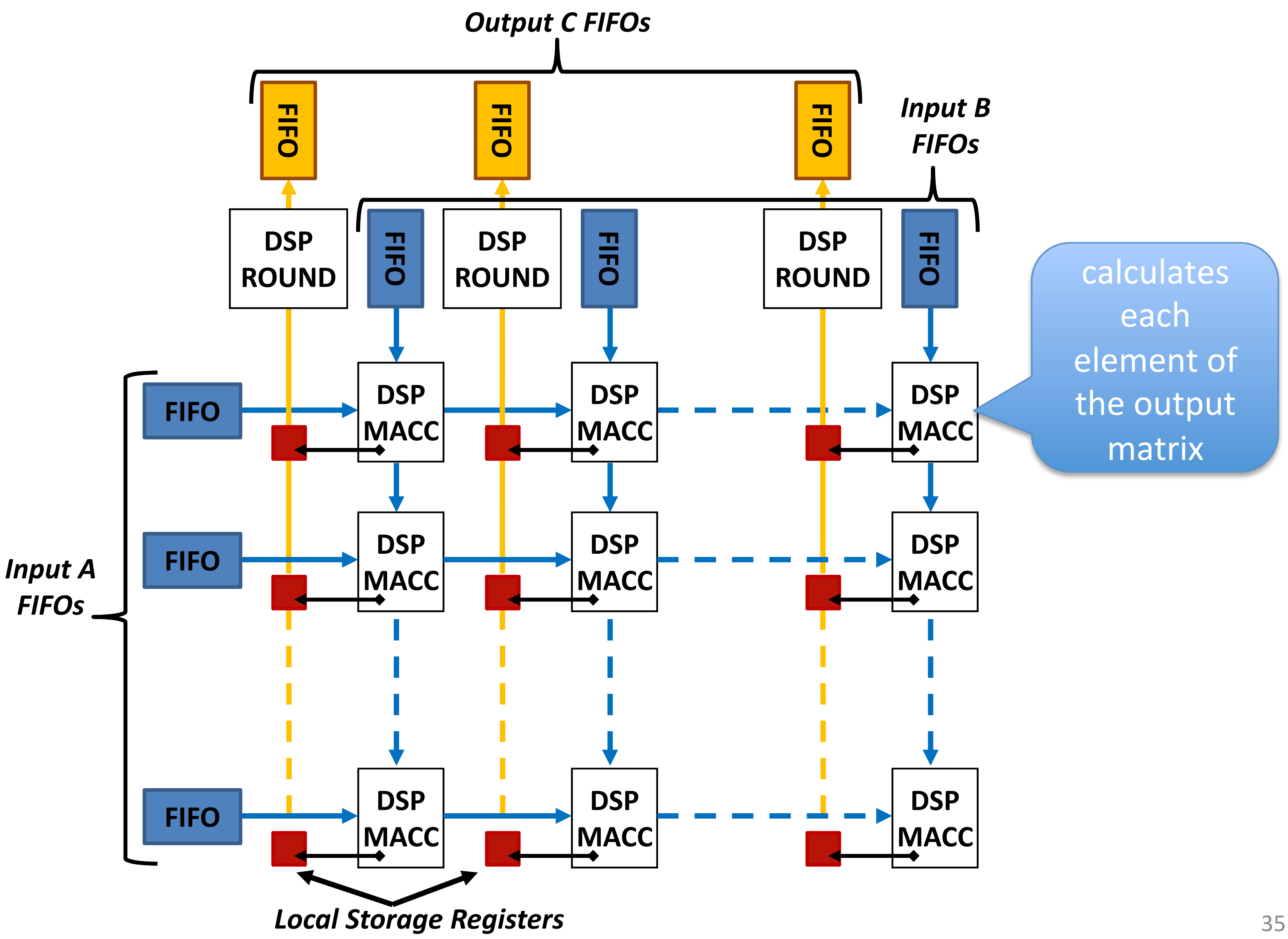
calculation in the prototype



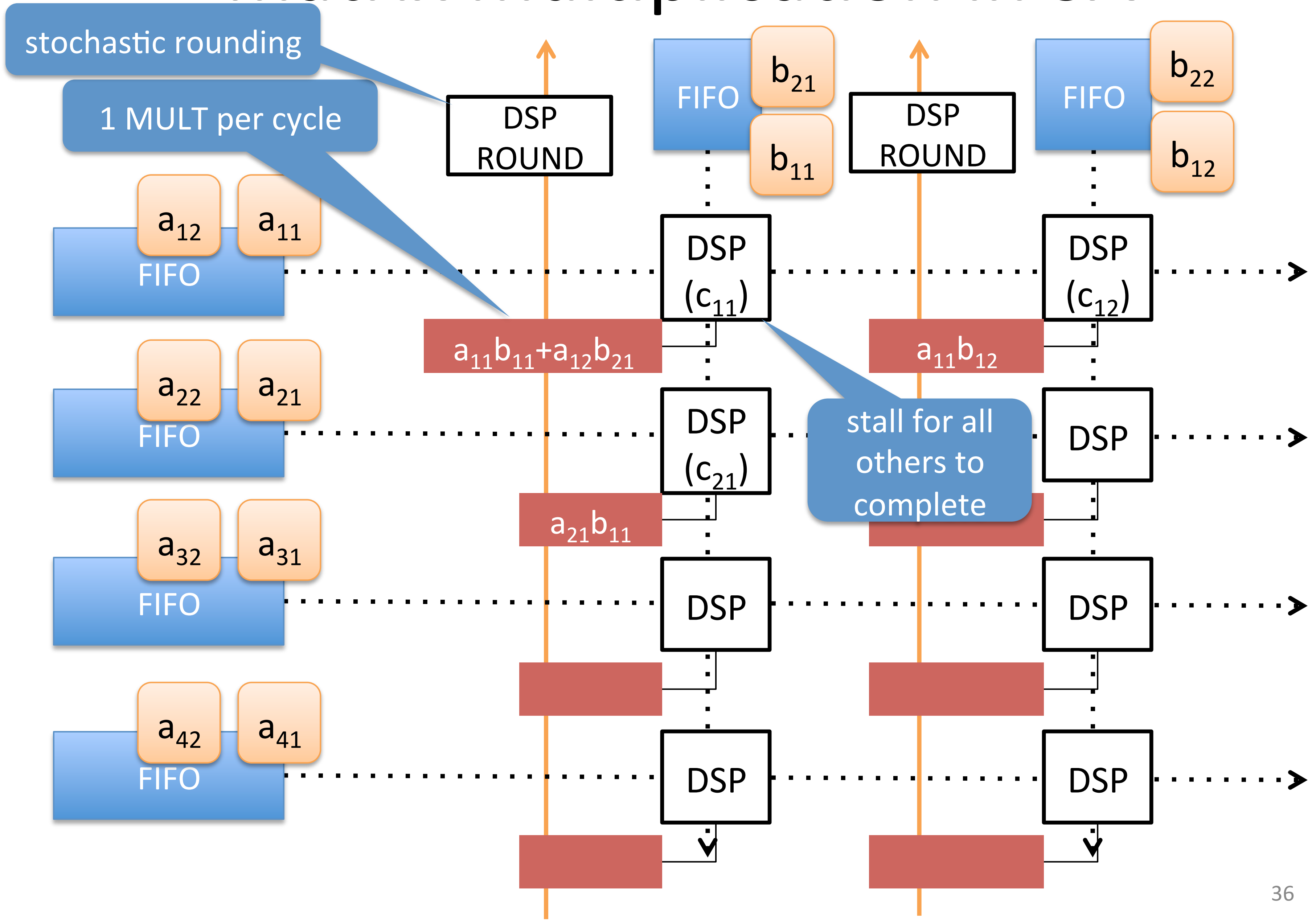
calculation in the prototype



Systolic Array(SA) Architecture



matrix multiplication in SA



Evaluating the prototype

- 28×28 SA is implemented on the FPGA.
  - The throughput is 260 G-ops/s.
  - The power efficiency is 37 G-ops/s/W.
    - The range of power efficiency of NVIDIA GT650m and GTX780, the Intel i7-3720QM is 1~5 G-ops/s/W

以降、予備スライド