Neuromorphic Computing

8. Case Study: Real Hardware-Software Design of 3D-NoC-Based Neuromorphic System

Ben Abdallah Abderazek, Khanh N. Dang
E-mail: {benab, khanh}@u-aizu.ac.jp

This lecture is based on the book ''Neuromorphic Computing Principles and Organization,'' Publisher: Springer; 2022 edition, ISBN-10 : 3030925242, ISBN-13 : 978-3030925246, by Abderazek Ben Abdallah, Khanh N. Dang

Lecture Contents

1. Introduction
2. R-NASH System
3. R-NASH Hardware
4. R-NASH Learning
5. R-NASH Mapping
6. Evaluation results
7. Conclusions

1. Introduction

- This lecture discuss about a case study on the design and evaluation of a reliable three dimensional digital neuromorphic processor (R-NASH)
- RNASH consists of:
  - The design methodology
  - Hardware
  - Learning
  - Mapping

Lecture Contents

1. Introduction
2. R-NASH System
3. R-NASH Hardware
4. R-NASH Learning
5. R-NASH Mapping
6. Evaluation results
7. Conclusions

2. R-NASH system
Overview

- The overview of R-NASH are in four phases:
  1. **Software modeling:** building the software model of the SNN application and perform the offline training to validate the model
  2. **Mapping:** generating the configuration from the software model to be able to port into the R-NASH hardware
  3. **Porting to hardware:** using the configuration to perform inference in the hardware
  4. **Runtime maintenance:** fault management for R-NASH

2. R-NASH system
System Phases

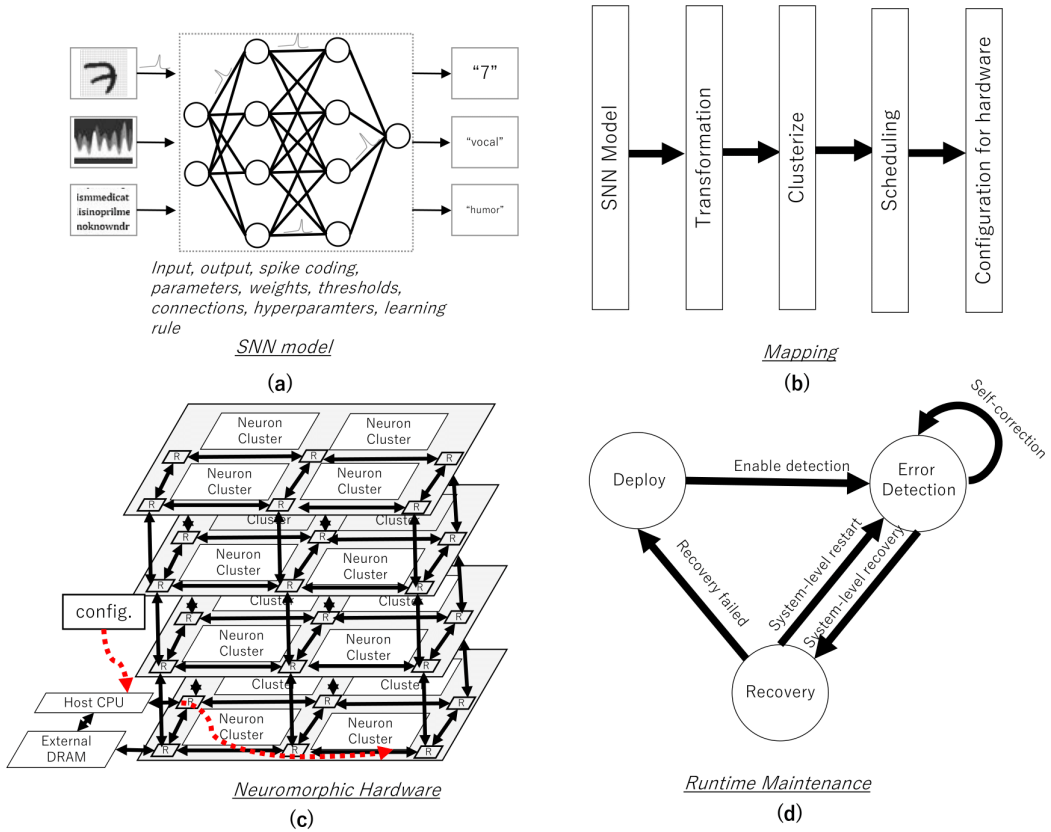

Fig. 8.2: R-NASH system platform. (a) SNN model. (b) Mapping into the neuromorphic hardware. (c) Our neuromophic hardware in 3D-ICs. (d) Runtime maintenance for R-NASH.

Lecture Contents

1. Introduction
2. R-NASH System
3. R-NASH Hardware
4. R-NASH Learning
5. R-NASH Mapping
6. Evaluation results
7. Conclusions

3. R-NASH hardware
Overview

- R-NASH is based on 3D Mesh Network-on-Chip for the communication.
- Communication is based on Address Event Representation protocol.
- Computation is done by the LIF core array for the core (256 LIFs per core).
- SRAM-based synapses are used in R-NASH.

3. R-NASH hardware Overview

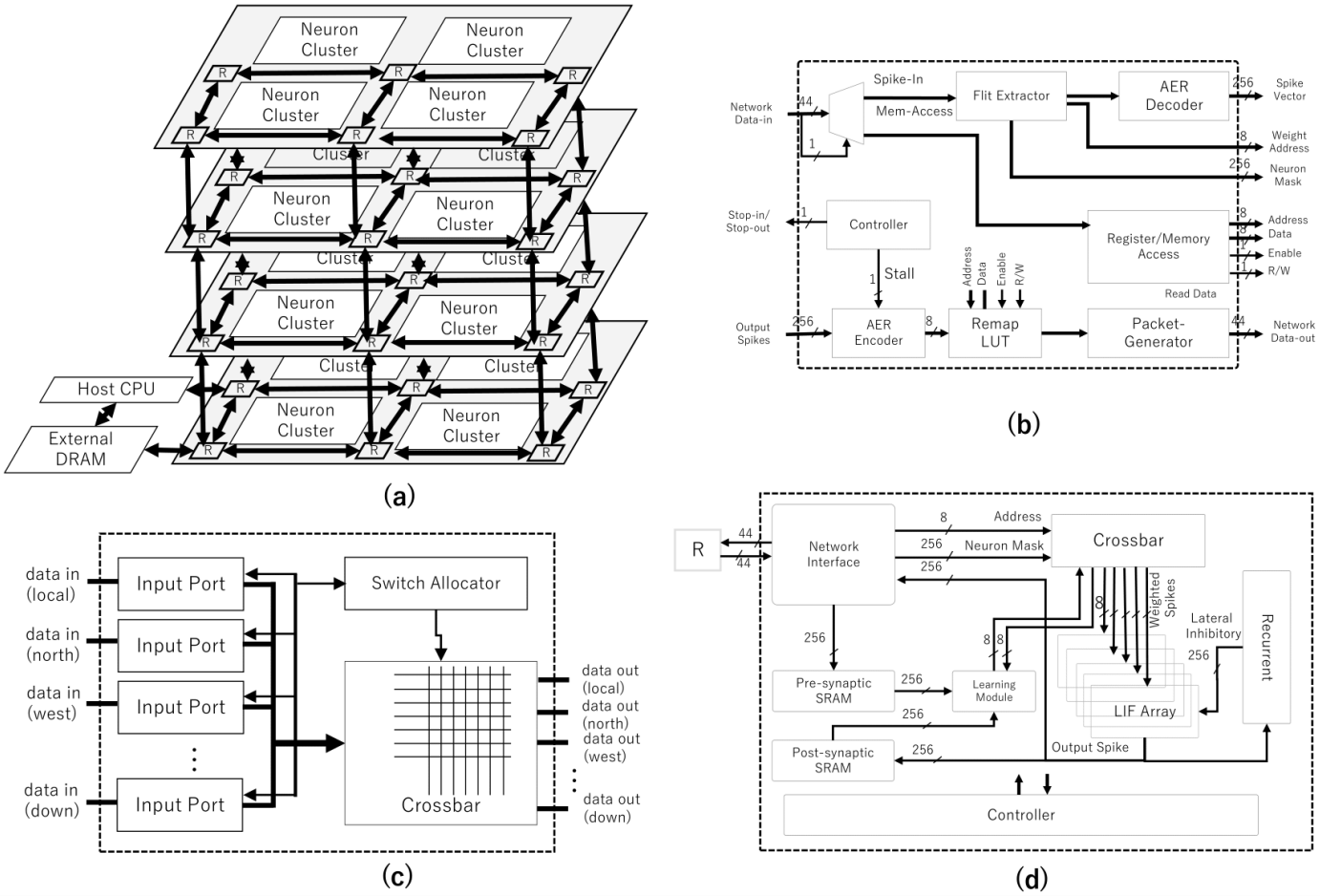

Fig. 8.3: R-NASH architecture. (A) Neuromorphic System. (B) Network-Interface. (C) 3D-Router. (D) SNPC Core

3. R-NASH hardware
Communication

- R-NASH supports two types of flits: **spike** (in AER format) and **memory flits**.
  - The AER format flit is converted to the address of the weight SRAM to extract the weight of the connection
  - Memory flit provides the instruction and the required addresses to read/write to/from the memory cells and registers in the neuron cluster.

3. R-NASH hardware
Flit format

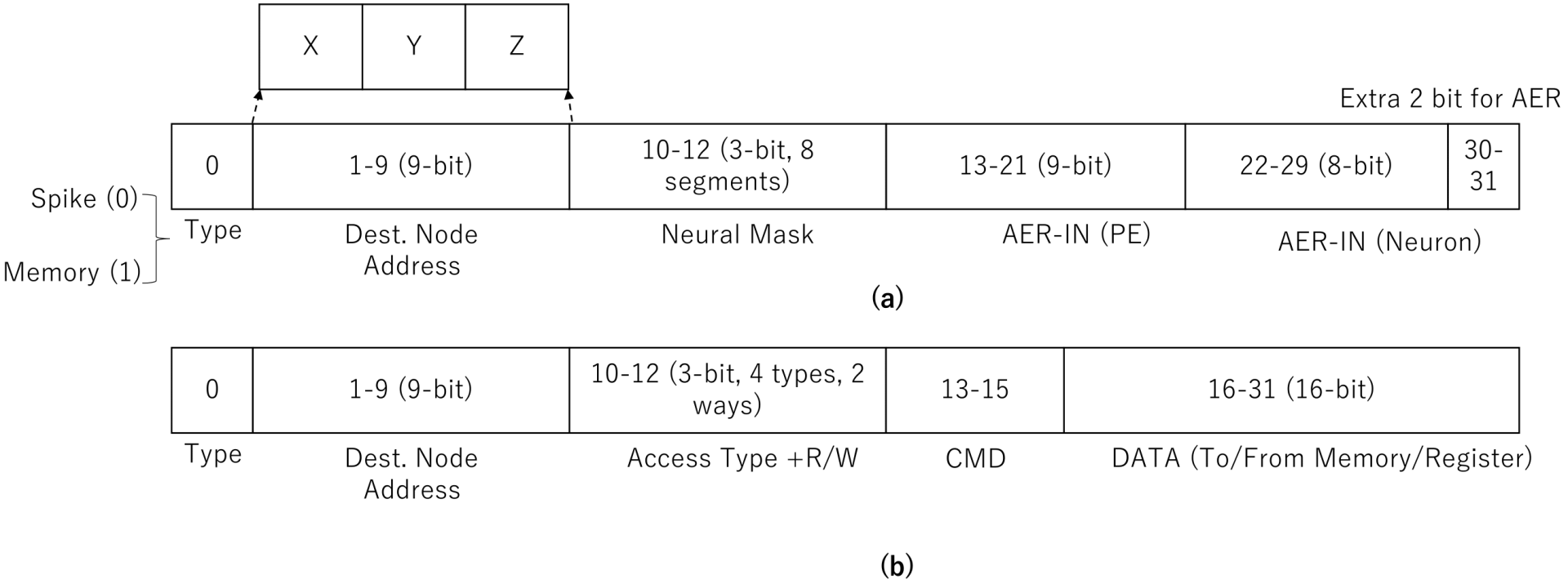

Fig. 8.4: R-NASH Flit Formats

3. R-NASH hardware
Router

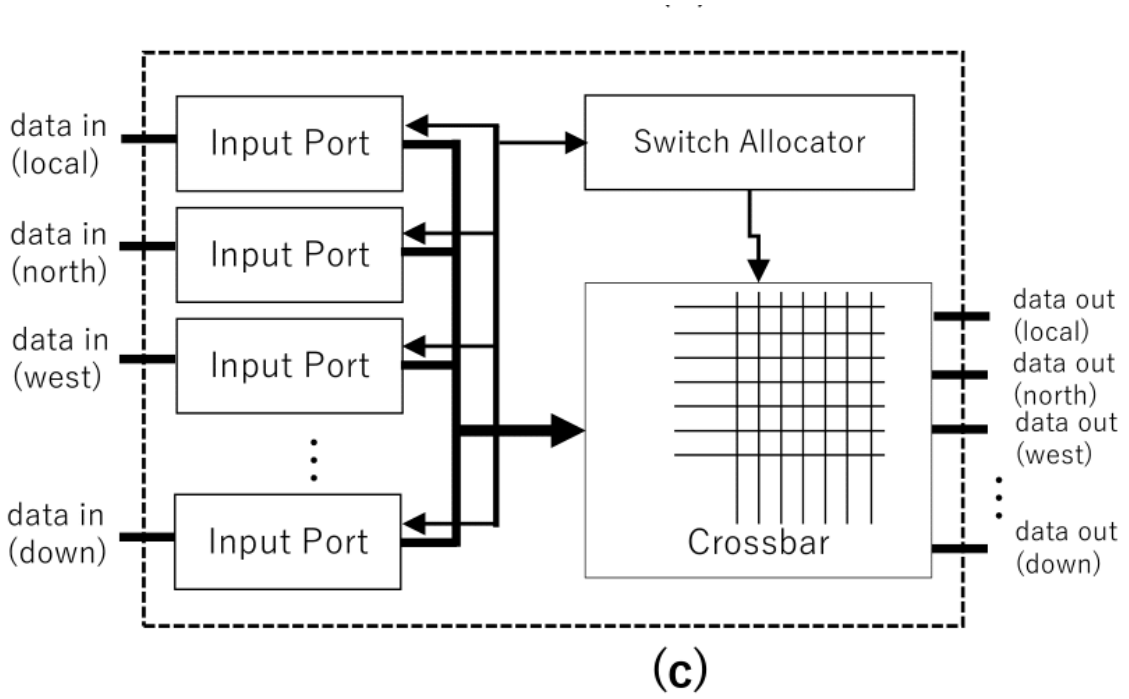

Fig. 8.3(c): Router architecture.

3. R-NASH hardware
Network Interface

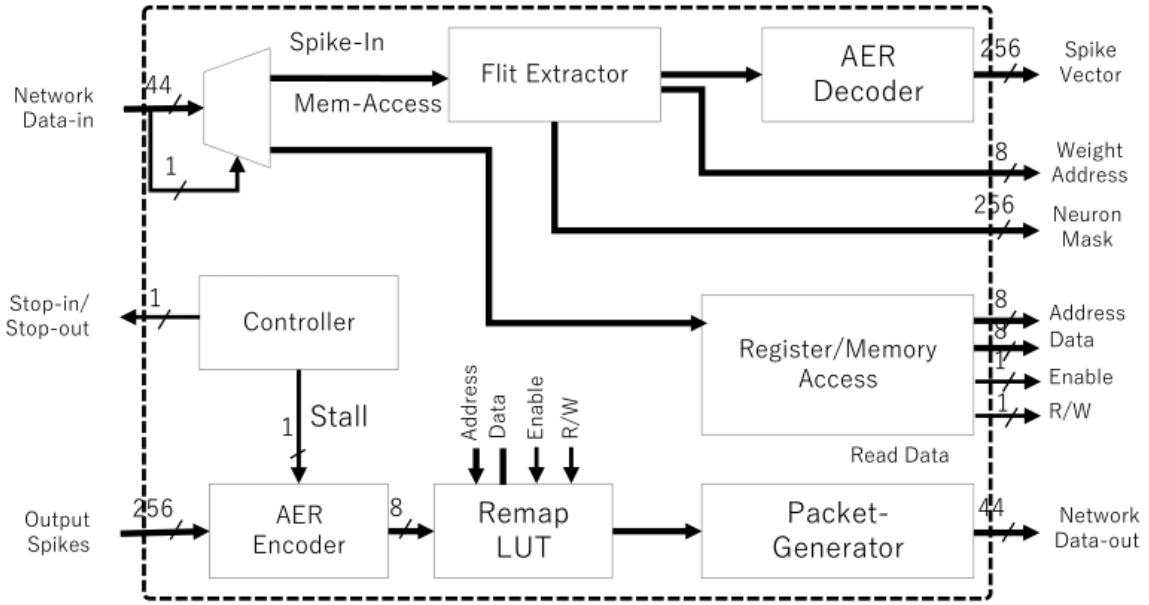

Fig. 8.3 (b): NI architecture.

3. R-NASH hardware
LIF neuron architecture

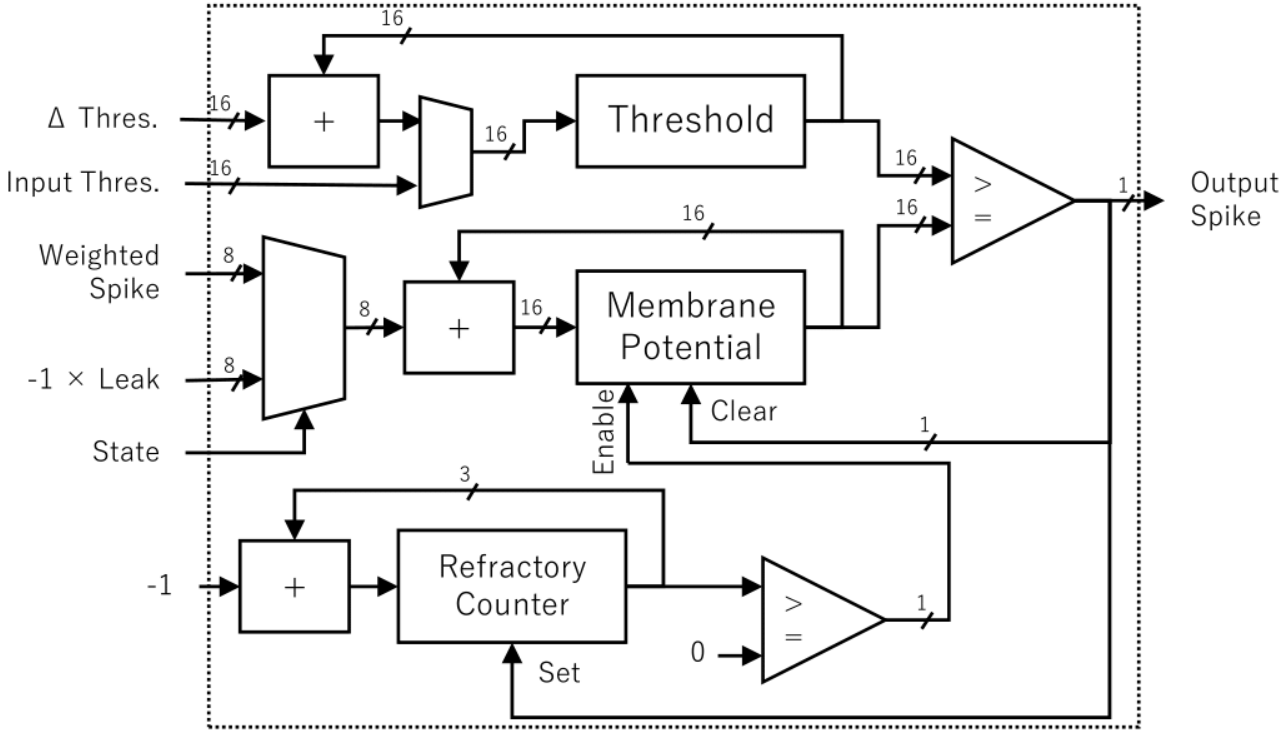

Fig. 8.5: LIF neuron architecture.

3. R-NASH hardware
STDP learning architecture

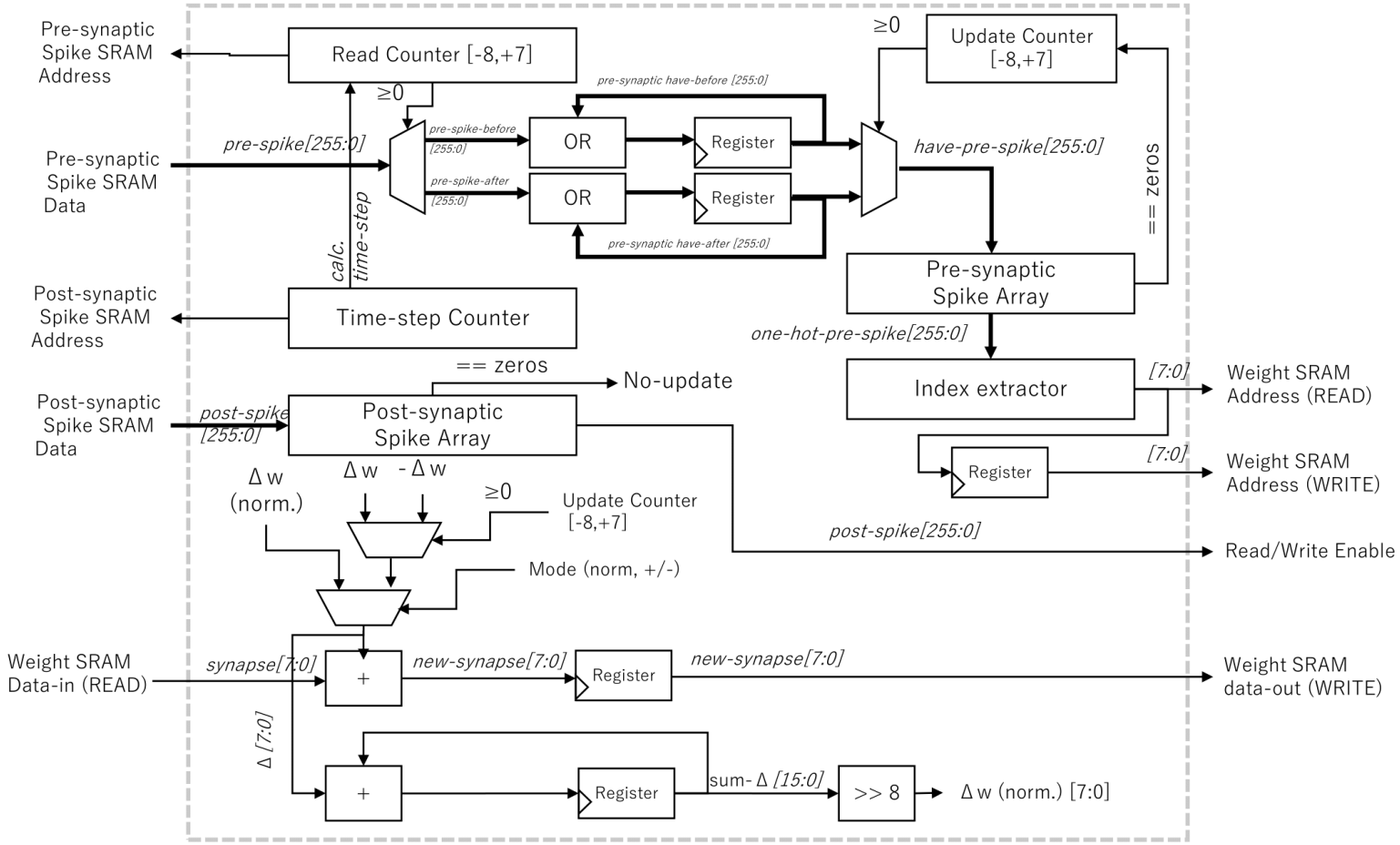

Fig. 8.7: On-chip STDP learning architecture

Lecture Contents

1. Introduction
2. R-NASH System
3. R-NASH Hardware
4. R-NASH Learning
5. R-NASH Mapping
6. Evaluation results
7. Conclusions

4. R-NASH learning
Overview

- R-NASH supports two types of learning:
  - **Off-chip learning:** weights are trained off-line and then download to the chip
  - **On-chip learning with STDP:** R-NASH core support on-chip learning with the ability to update the weight during operation.
- The on-chip learning STDP will follow the simplified rules for unsupervised learning:
  - Weight adjustment will be fixed.
  - Weight normalization will be performed.

4. R-NASH learning
STDP learning model

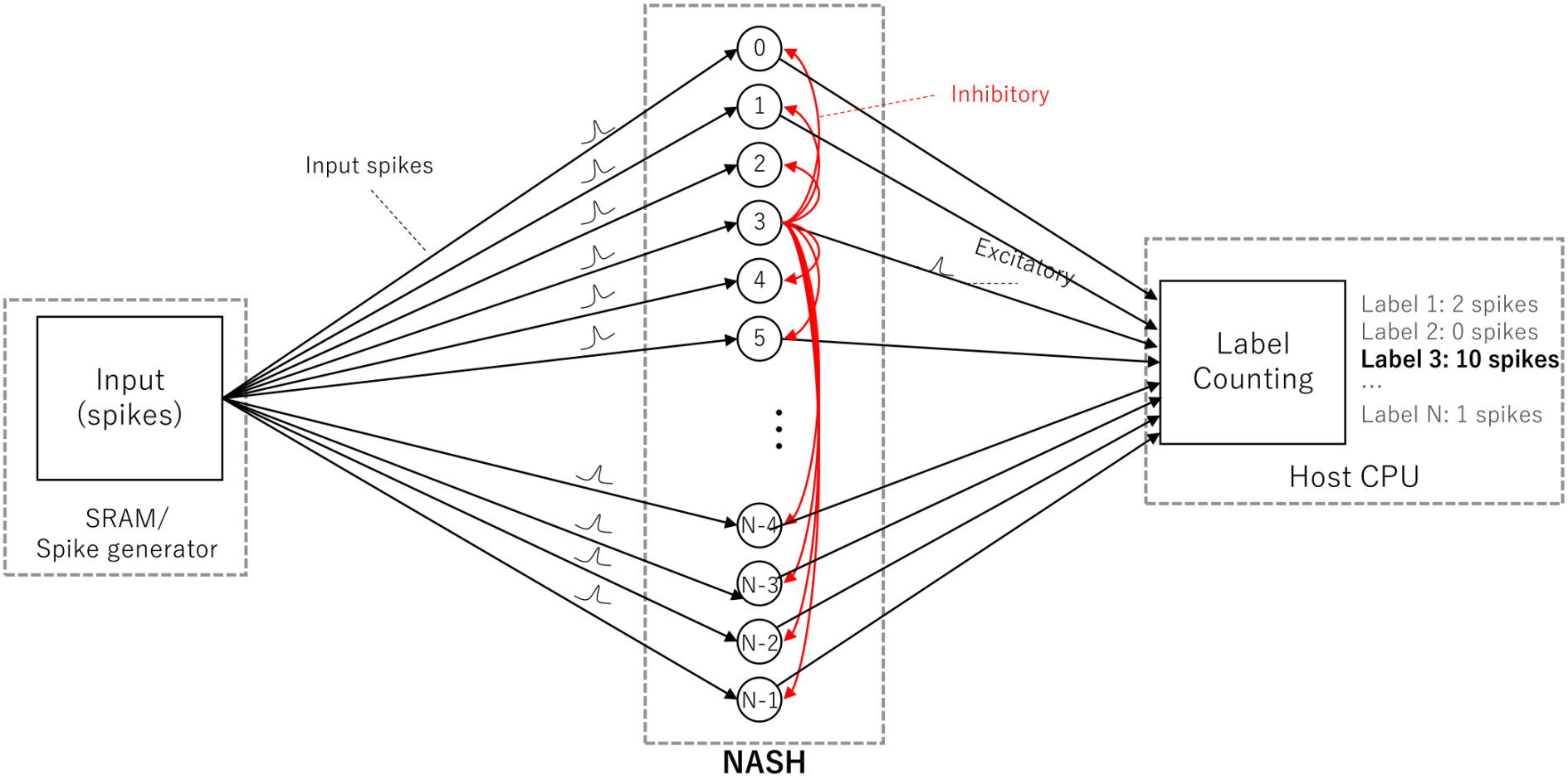

Fig. 8.6: On-chip STDP learning model

4. R-NASH learning
STDP learning weight adjustment

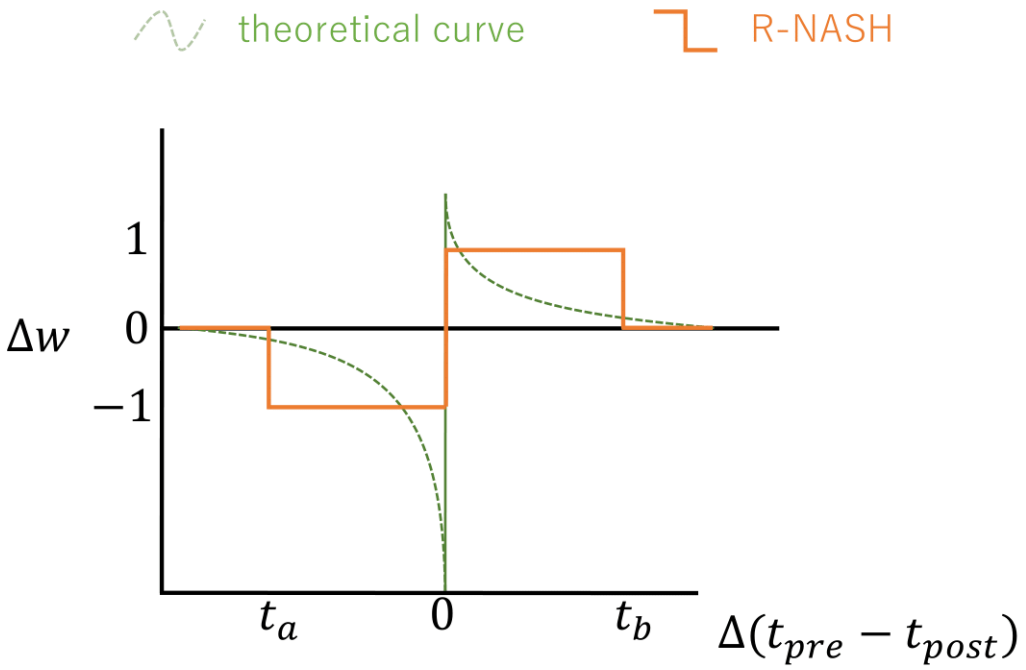

Fig. 8.8: On-chip STDP learning mechanism

Lecture Contents

1. Introduction
2. R-NASH System
3. R-NASH Hardware
4. R-NASH Learning
5. R-NASH Mapping
6. Evaluation results
7. Conclusions

5. R-NASH mapping
Overview

- As we break the neuromorphic system into groups of neurons connected via a Network-on-Chip, dividing and placing are essential issues since they can heavily affect the performance.
  - For instance, placing two connected neurons far apart can lead to a critical delay path in the system.
- In R-NASH, we use Genetic Algorithm to perform mapping.

5. R-NASH mapping
Mapping Algorithm

Algorithm 6 Genetic algorithm for neurons mapping

```
// initialize phase
S1: load the system configuration
S2: randomize the K mapping solutions
// evolve phase
for (generation g_i in 1 to G) do
    S3: remove the wrong mapping solutions
    S4: calculate cost function (communication cost) for each solution of the population
    S5: select the B best out of K solutions based on the cost function
    S6: mutate the B best solutions to have new K solutions
    S6: crossover the new K solutions to have new population
    S7: check if it satisfies the communication cost or does not improve over several generations
// finalize phase
S7: calculate cost function for each solution of the population
S8: select the B = 1 best out of K solutions based on the cost function
```

5. R-NASH mapping
Gene, Cross-over and Mutation

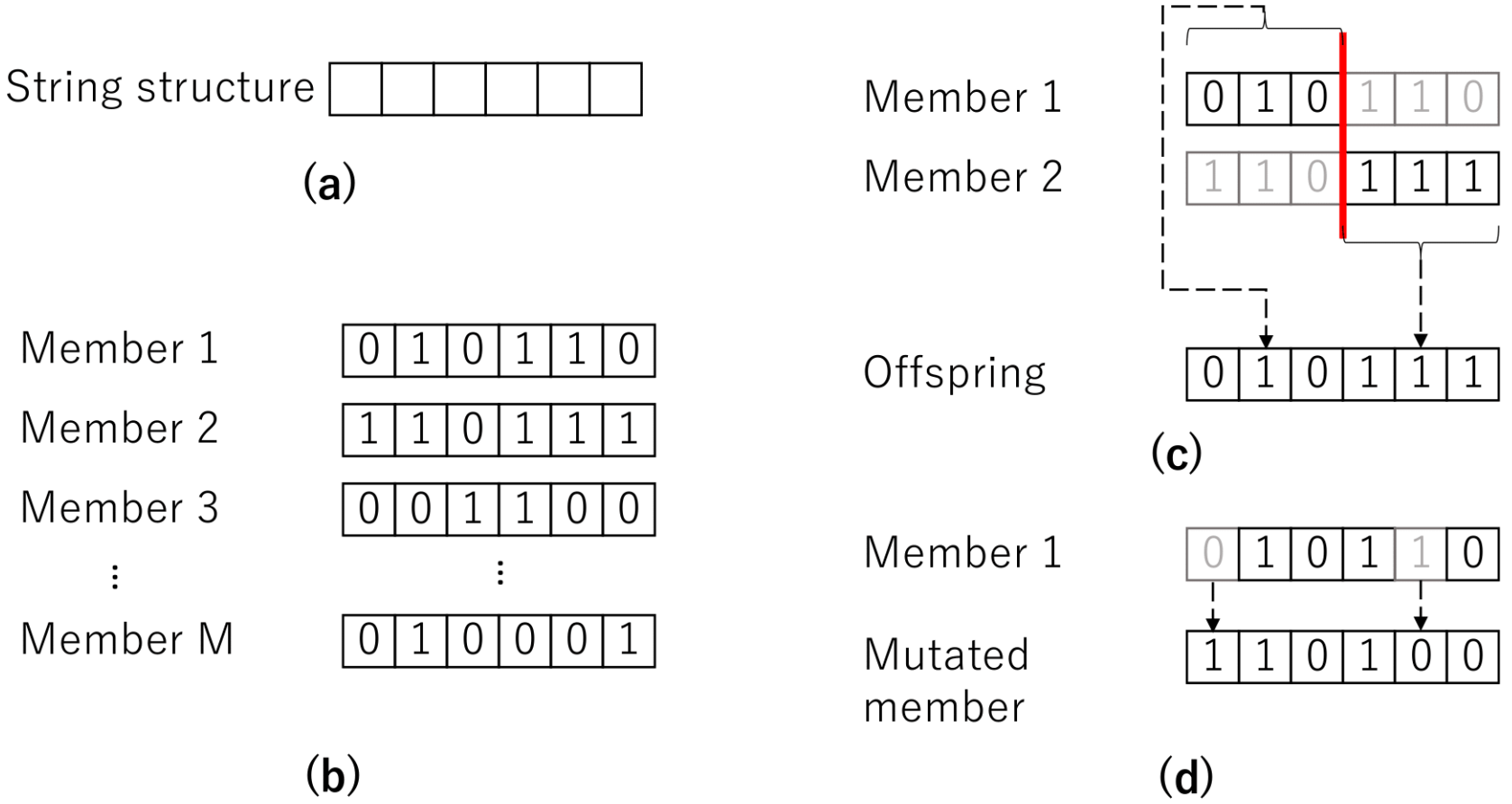

Fig. 8.9: Genetic algorithm (GA): (a) an example string structure; (b) initialization process; (c) cross-over and (d) mutation

5. R-NASH mapping
Example of crossover and mutation

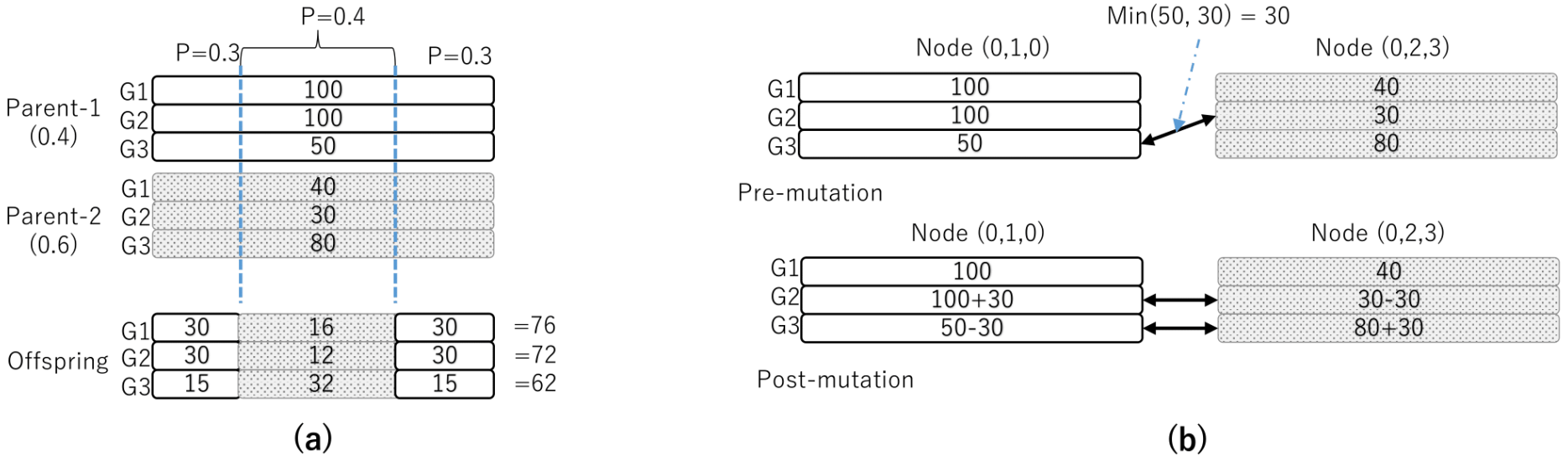

Fig. 8.10: Crossover and mutation method for GA mapping. (A) Crossover. (B) Mutation

5. R-NASH mapping
Fault-tolerance features

- To tackle defective routing, R-NASH offers:
  - Fault tolerance mechanism inside router (input buffer, crossbar).
  - Error Correction Code in the flit (2xSECDED(22, 16)).
  - Soft error tolerance in the routing unit
- To tackle defective neurons, R-NASH offer remapping to use spare neurons as replacements.

Lecture Contents

1. Introduction
2. R-NASH System
3. R-NASH Hardware
4. R-NASH Learning
5. R-NASH Mapping
6. Evaluation results
7. Conclusions

6. Evaluation
Genetic Algorithm for Initial Mapping

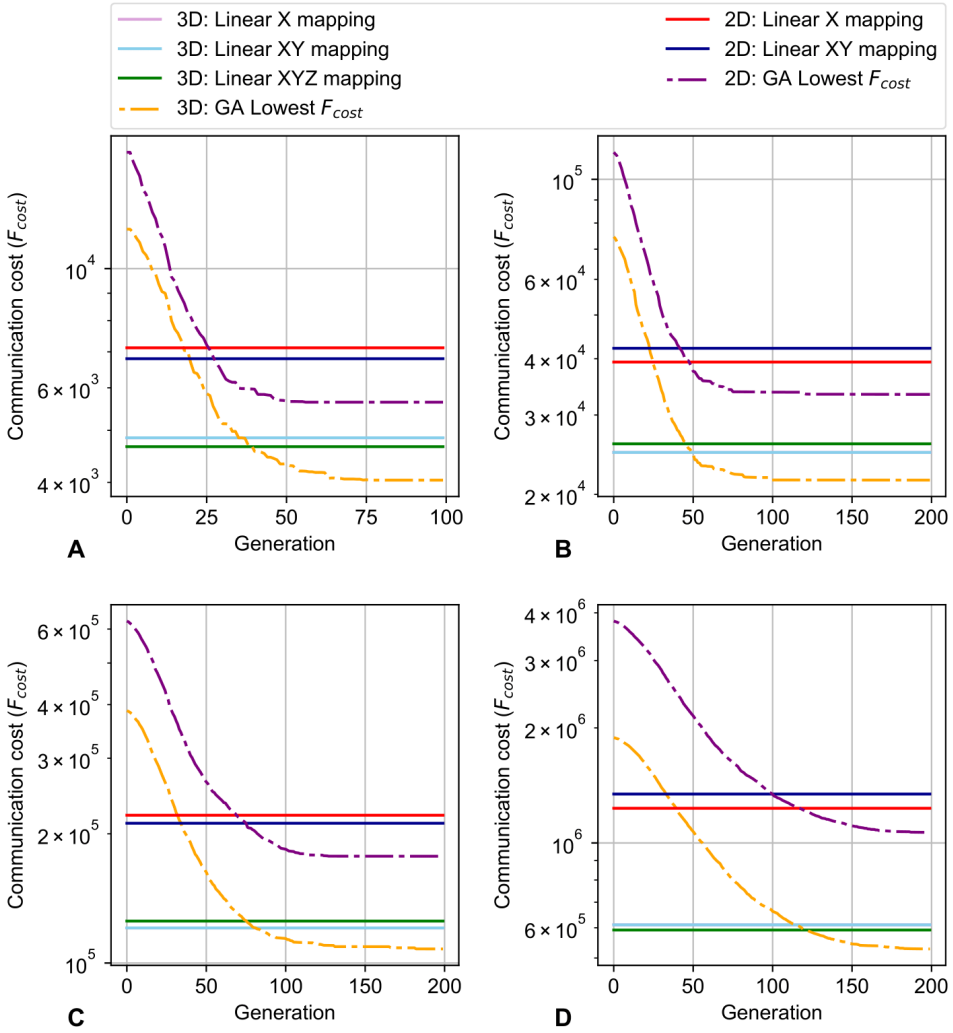

Fig. 8.12: Genetic Algorithm Result for initial mapping. (A) 4 × 4 × 4 NoC-based, 256 neurons/node. (B) 6 × 6 × 6 NoC-based, 256 neurons/node. (C) 8 × 8 × 8 NoC-based, 256 neurons/node.(D) 10 × 10 × 10 NoC-based, 256 neurons/node.

6. Evaluation
Genetic Algorithm for Initial Mapping

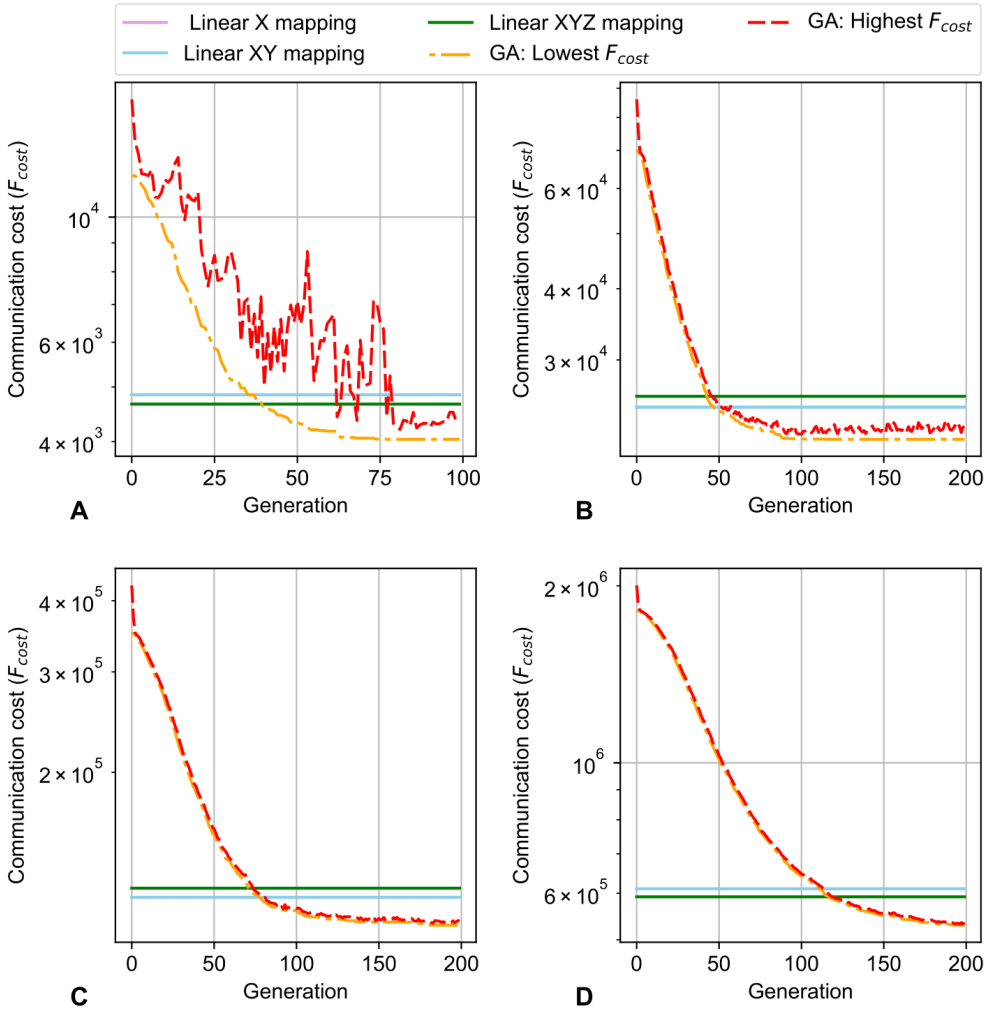

Fig. 8.13: Genetic Algorithm Result of the initial mapping of 3D NoC-based. (A) 4 × 4 × 4, 256 neurons/node. (B) 4 × 4 × 8, 128 neurons/node.(C) 4 × 8 × 8, 64 neurons/node. (D) 4 × 8 × 8, 32 neurons/node.

6. Evaluation

Genetic Algorithm for Initial Mapping

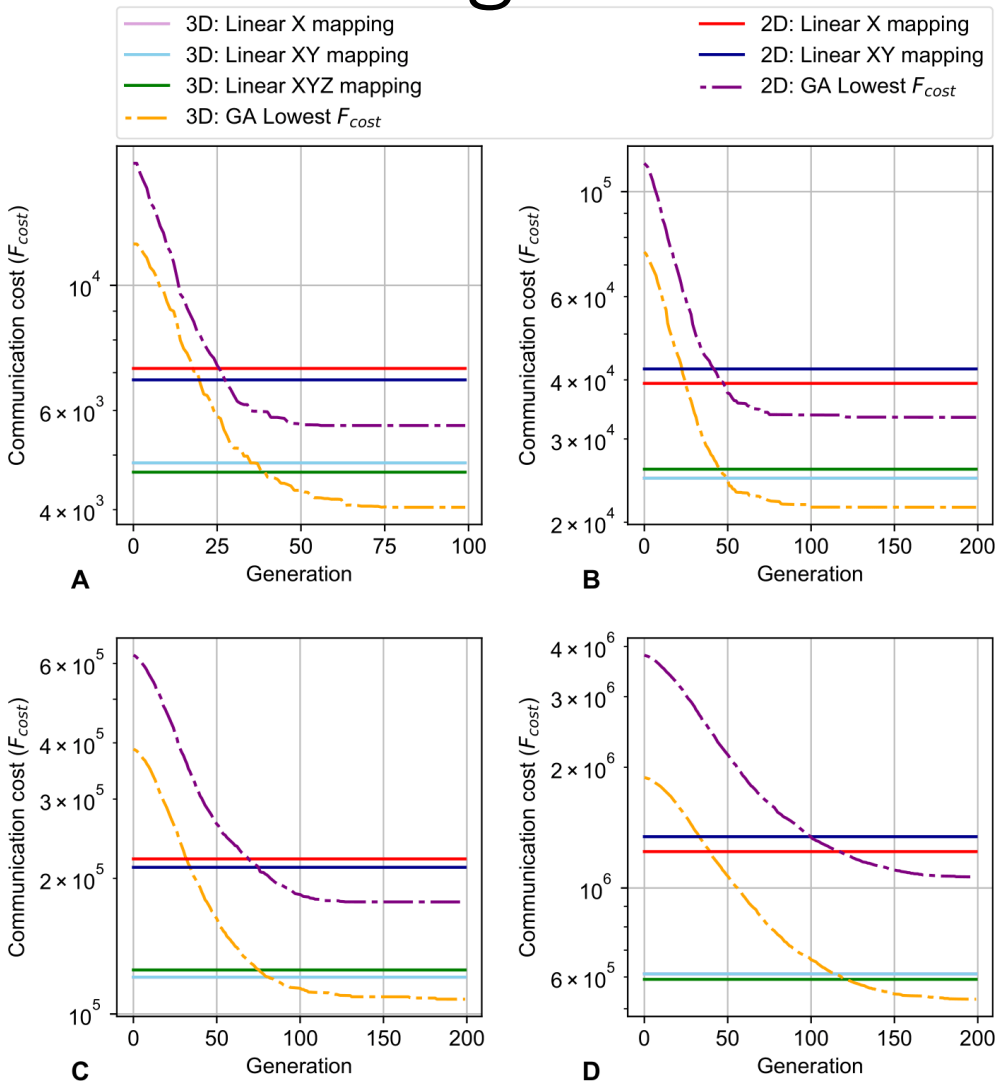

Fig. 8.14: Comparison between 3D and 2D mapping. (A) 64 nodes (4 × 4 × 4 and 8 × 8) NoC-based, 256 neurons/node. (B) 128 nodes (4 × 4 × 8 and 8 × 16) NoC-based, 256 neurons/node. (C) 256 nodes (4 × 8 × 8 and 16 × 16) NoC-based, 256 neurons/node.(D) 512 nodes (8 × 8 × 8 and 16 × 32) NoC-based, 256 neurons/node.

6. Evaluation
Layout

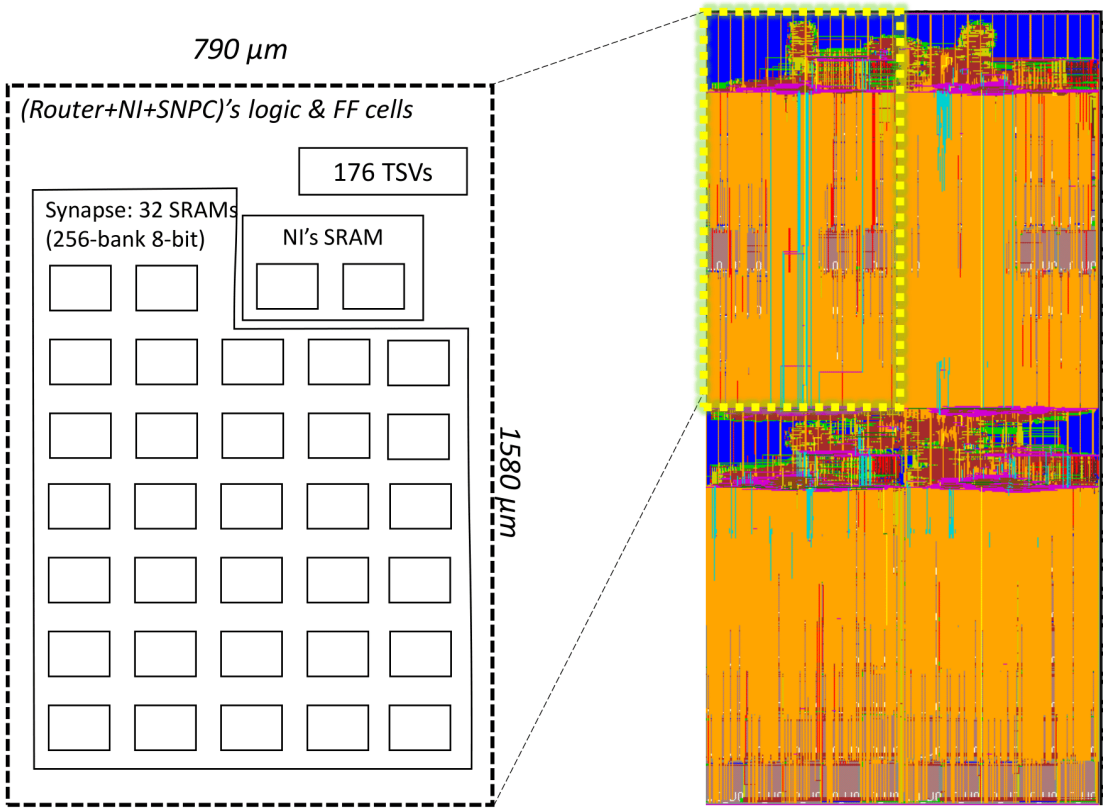

Fig. 8.16: Layout of a 2 × 2 3D-NoC-based R-NASH layer. A tile's size is 790μm ×1580μm

6. Evaluation
Off-chip learning

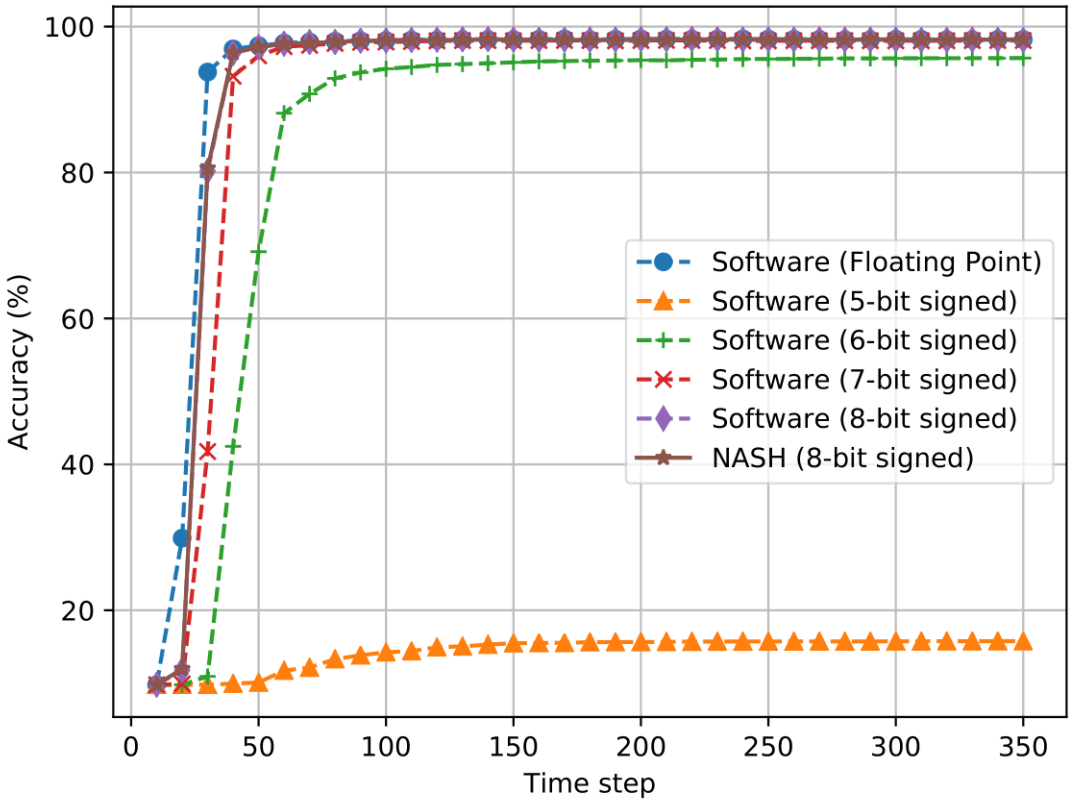

Fig. 8.17: Accuracy result of offline training for MNIST dataset with the network model 784:1024:10.

6. Evaluation
Off-chip learning

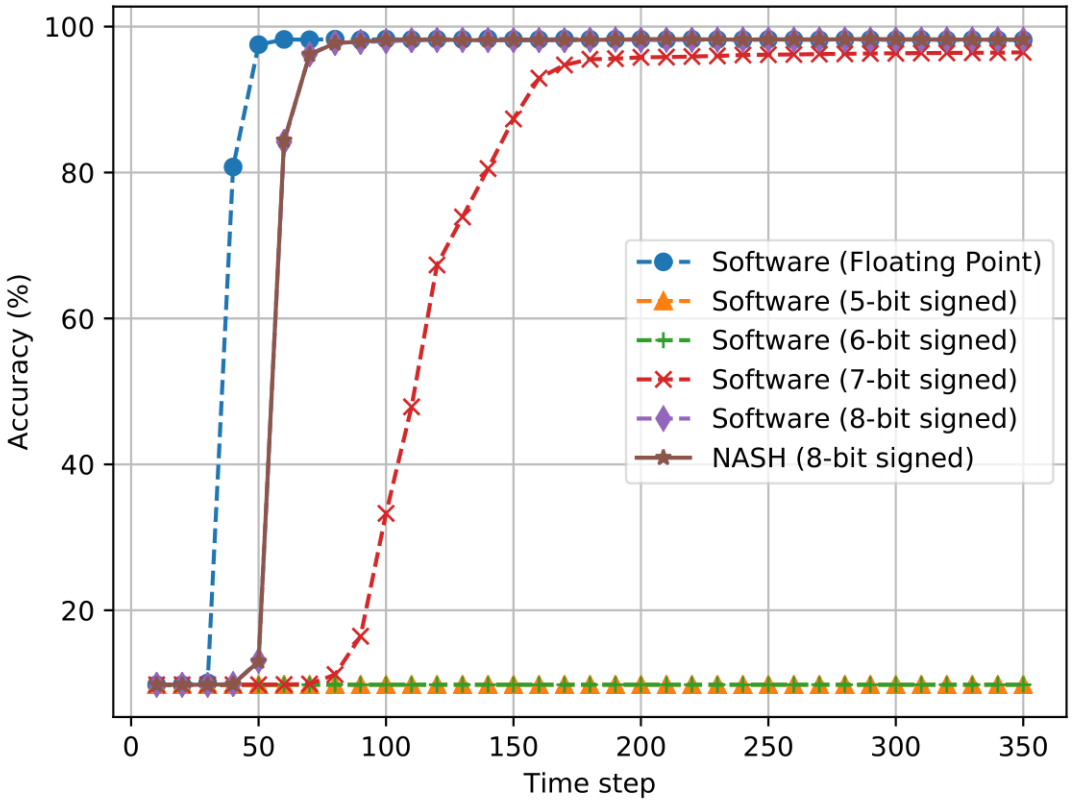

Fig. 8.18: Accuracy result of offline training for MNIST dataset with the network model 784:1024:1024:10.

6. Evaluation
Off-chip learning

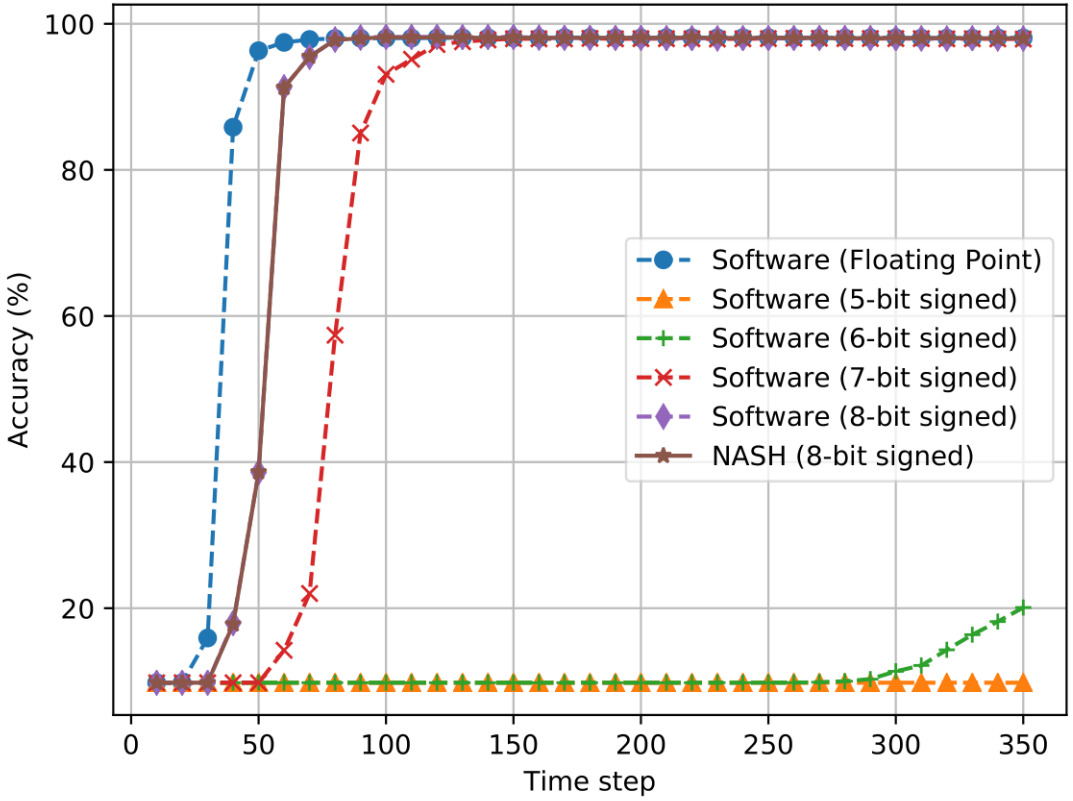

Fig. 8.19: Accuracy result of offline training for MNIST dataset with the network model 784:1024:1024:1024:10.

5. R-NASH mapping
Genetic Algorithm for Initial Mapping

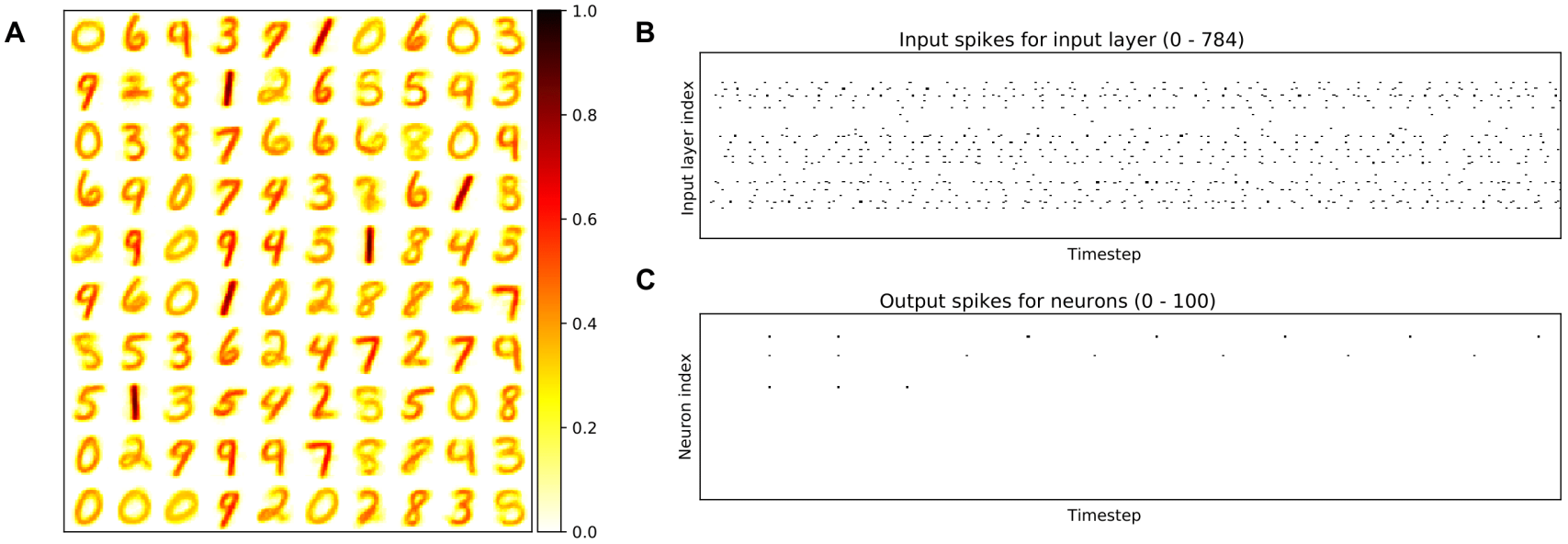

Fig. 8.20: Illustration of the STDP learning model. (A) the final weights. (B) Illustration of input spikes for the first test image (number 8). (C) Illustration of output spikes.

Lecture Contents

1. Introduction
2. R-NASH System
3. R-NASH Hardware
4. R-NASH Learning
5. R-NASH Mapping
6. Evaluation results
7. Conclusions

7. Conclusions

- This lecture shows a case study of R-NASH:
  - 3D NoC based neuromorphic architecture.
  - Fault-tolerance features
- It supports two types of learning:
  - **Off-chip learning & On-chip learning with STDP**
- R-NASH also support initial mapping using GA.
- R-NASH provide protection in the routing unit (routers) and computing unit (neurons)

7. Readings

- F. Akopyan et al., "TrueNorth: Design and Tool Flow of a 65 mW 1 Million Neuron Programmable Neurosynaptic Chip," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 34, no. 10, pp. 1537-1557, Oct. 2015, doi: 10.1109/TCAD.2015.2474396.

https://ieeexplore.ieee.org/abstract/document/7229264