Spiking Neural Networks with STDP

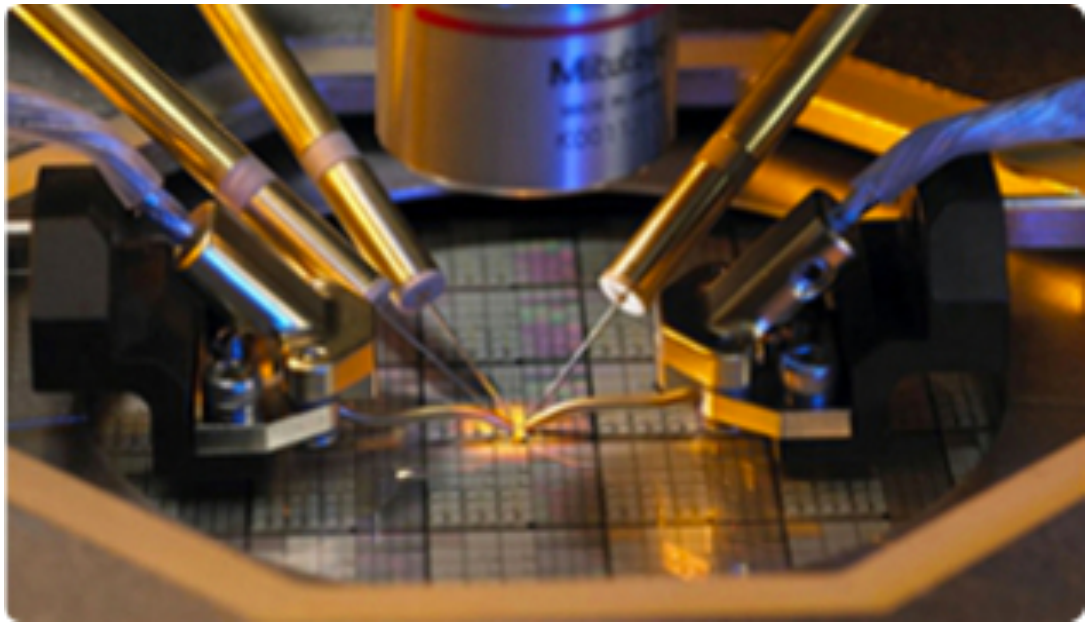
From Robust Hardware Architectures to Testing Strategies

Elena Ioana Vătăjelu, Giorgio Di Natale, Lorena Anghel

TIMA Laboratory, Grenoble, France

Why testing?

- To guarantee quality!
- What about Spiking Neural Networks?
  - Is functional testing enough?
  - Fault modeling?
  - Test methods?



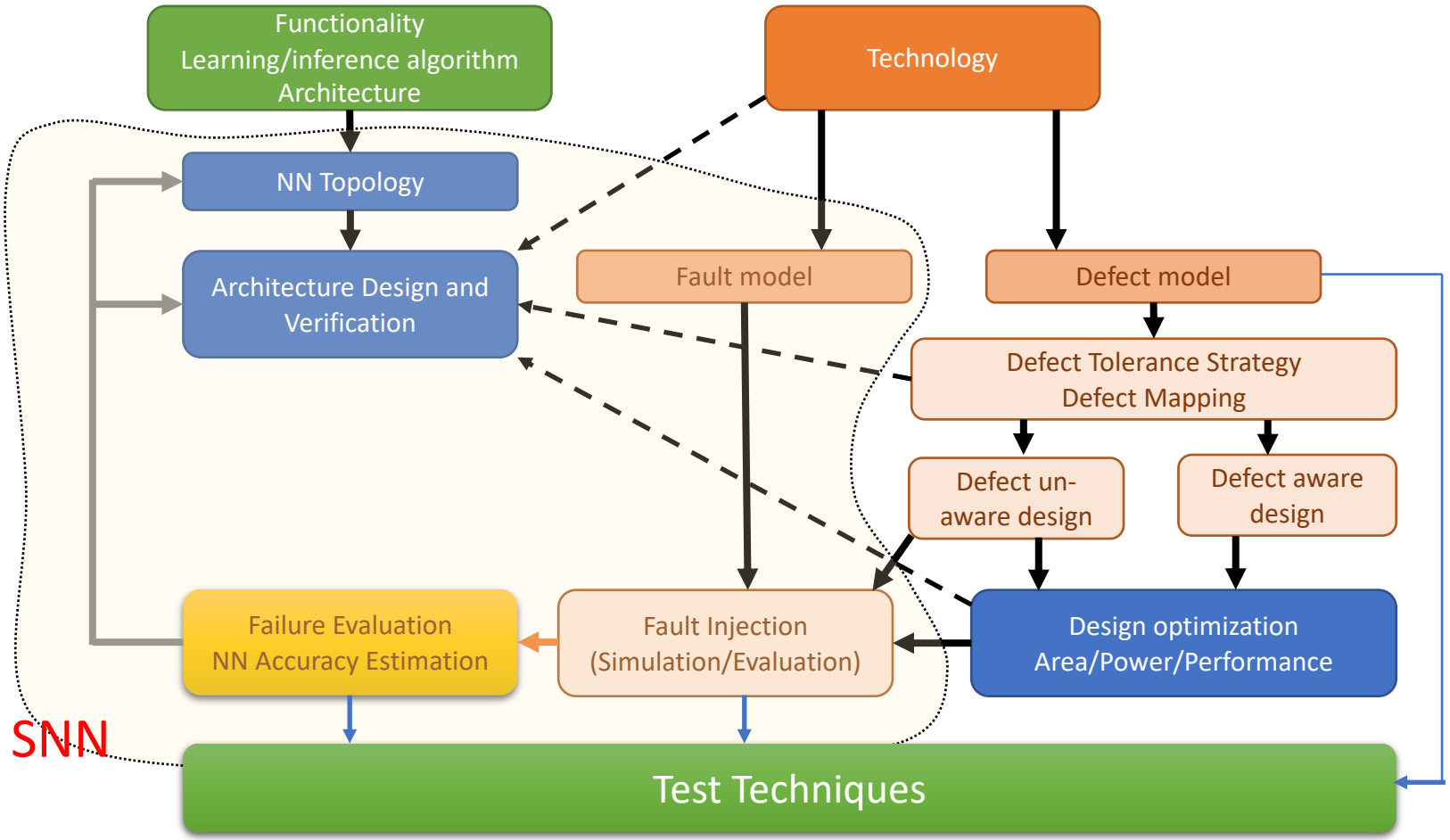
Classical Test: Fault Modeling & ATPG

- Fault modeling: to model the effect of physical defects
- Fault simulation: to simulate the behavior of the circuit in presence of a fault
- Test generation: to generate input vectors able to excite faults and to manifest their presence

- All highly dependent on:
  - The design
  - The technology
  - The operation conditions
  - The application

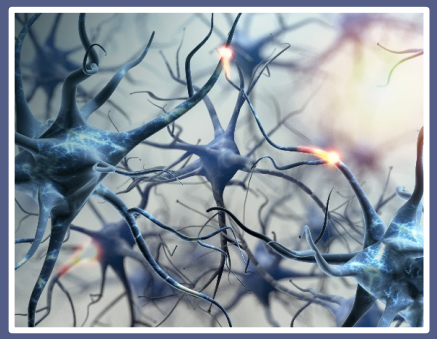
$$P\ of\ failure < \frac{Tolerable\ Risk}{Risk}$$

Our approach to NN study

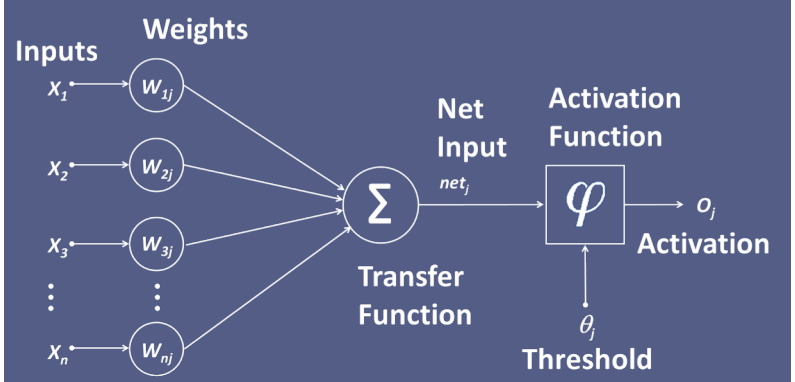


Spiking Neural Network (SNN)

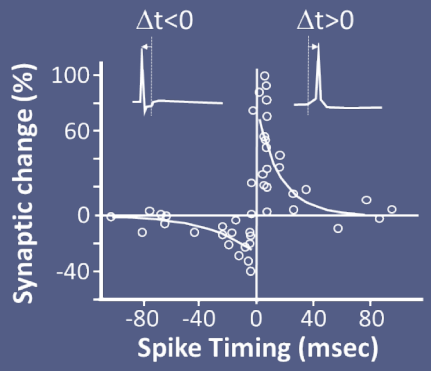
SPIKING NEURONS



LEAKY INTEGRATE-AND-FIRE

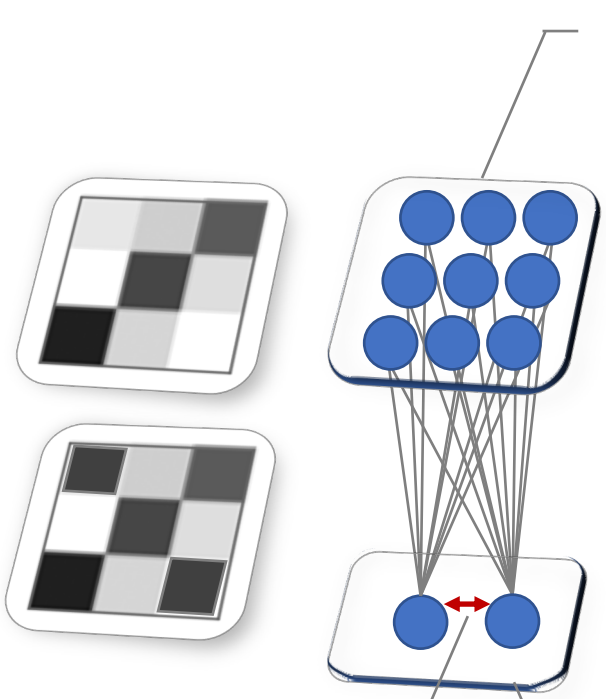


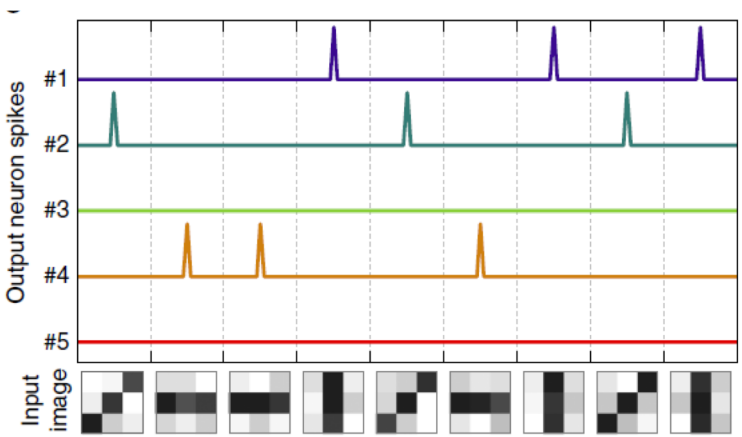
SYNAPSES



Basics of SNN with STDP

Spiking Neural Network Structure - example





Boyn, Sören et al, "Learning through ferroelectric domain dynamics in solid-state synapses," Nature Communications, 2017

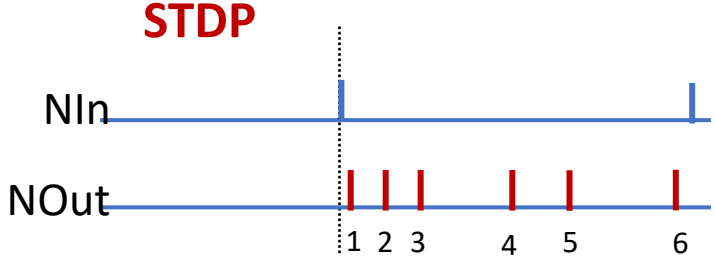
Basics of SNN with STDP

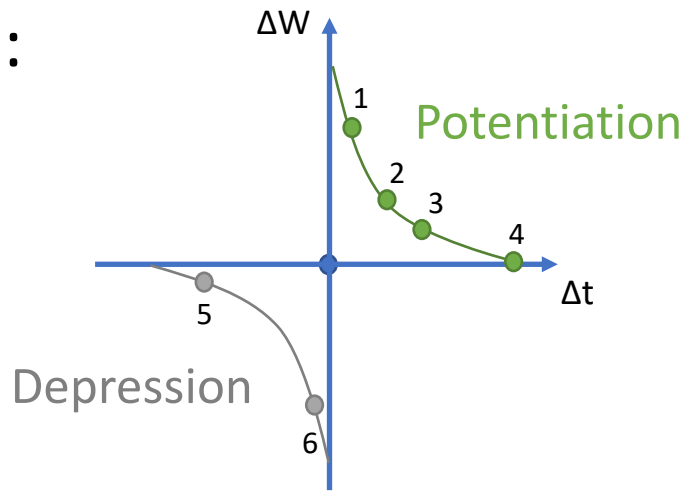
- Has 3 stages of operation:
  - Learning (STDP)
  - Test
  - Inference

Basics of SNN with STDP

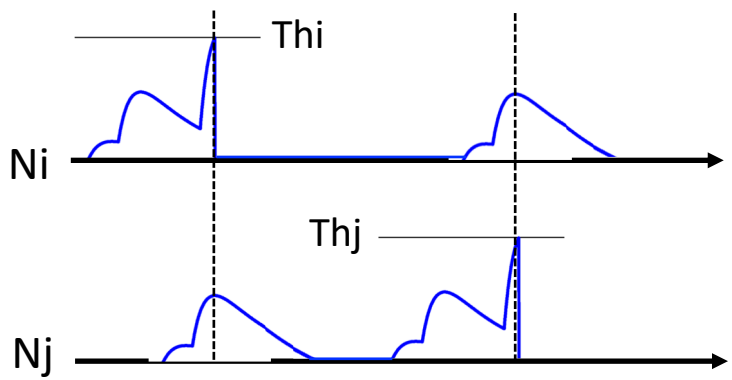
- Has 3 stages of operation:
  - Learning

STDP

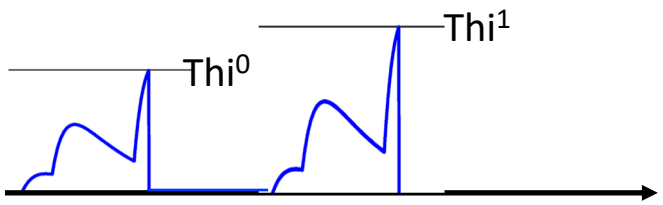




Lateral inhibition

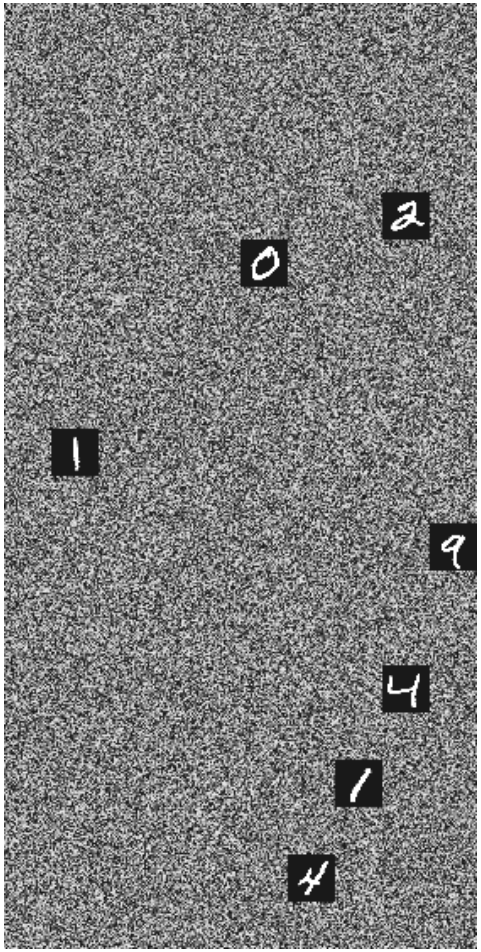


Homeostasis

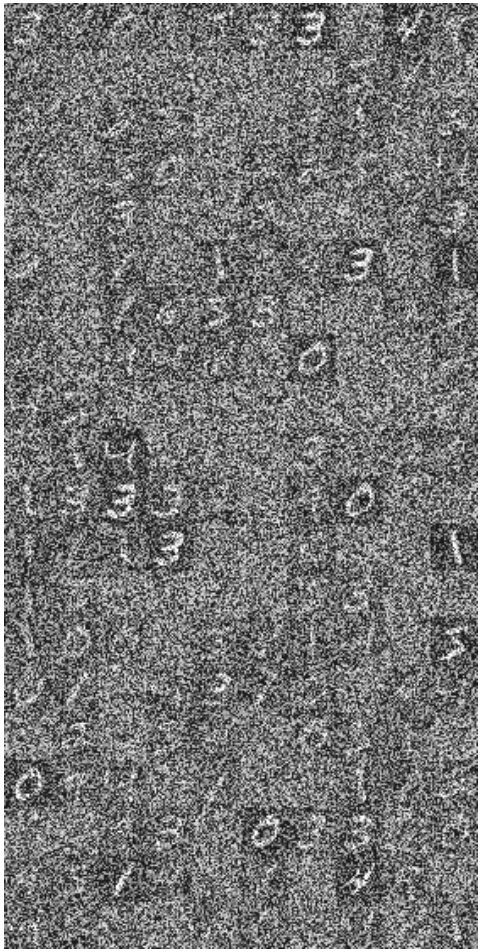


Basics of SNN with STDP

Learning without
lateral inhibition & homeostasis



Full Learning



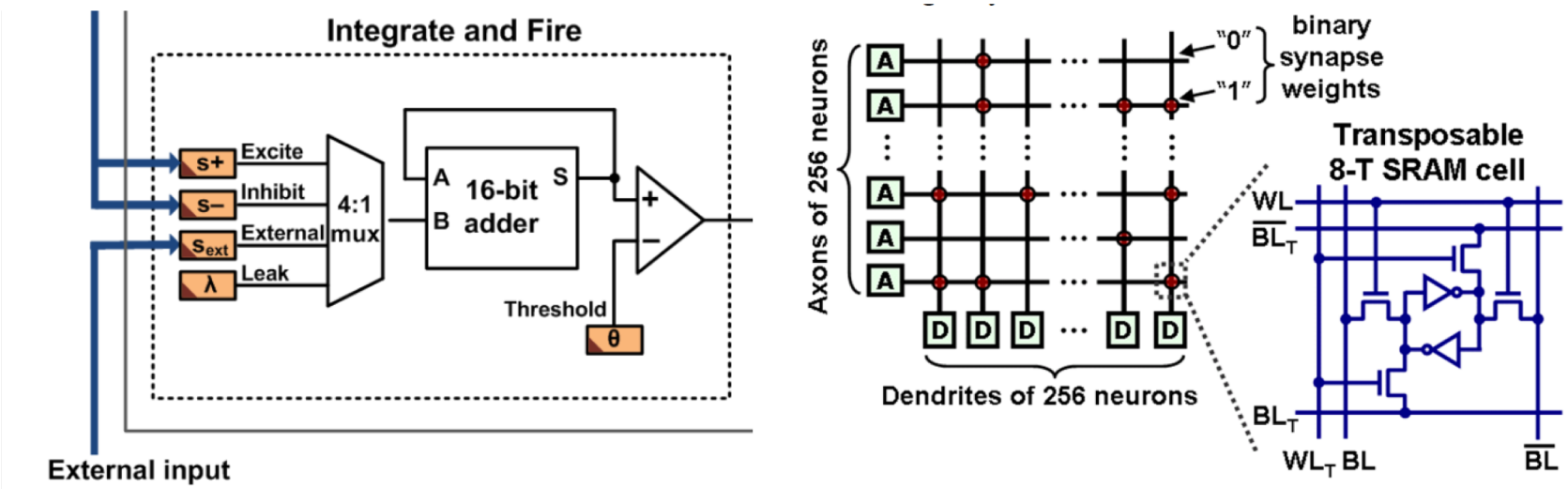
Presentation overview

- Design of HW-implemented SNNs
- Fault modelling
- Fault injection
- Test issues and challenges
- Conclusions

Presentation overview

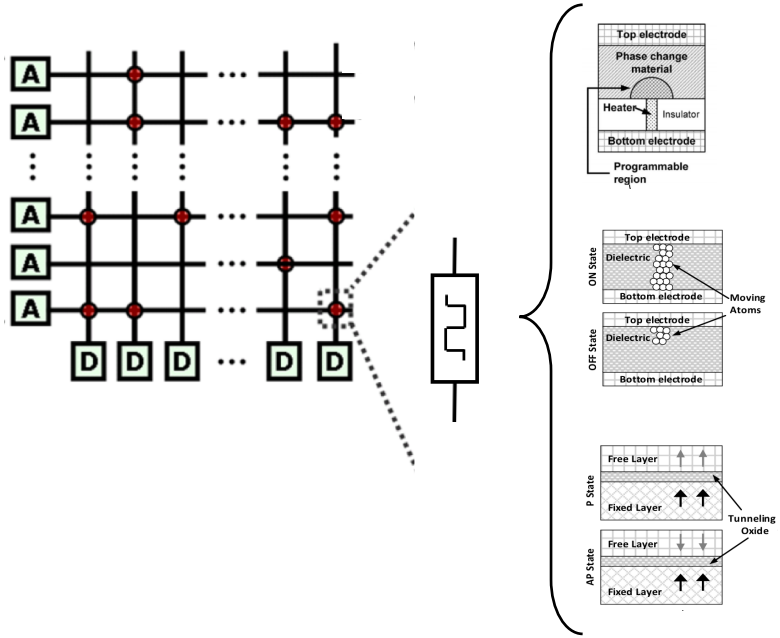
- Design of HW-implemented SNNs
- Fault modelling
- Fault injection
- Test issues and challenges
- Conclusions

Digital SNN implementation

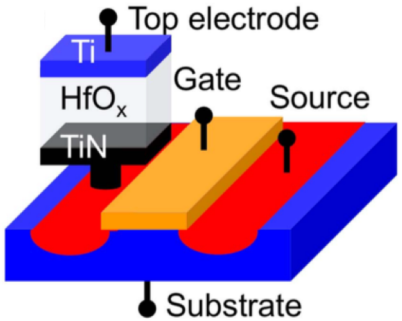


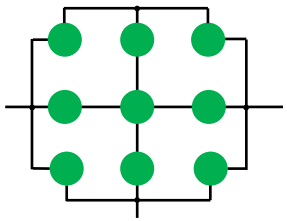
IBM, J. Seo *et al.*, *2011, IEEE CICC*

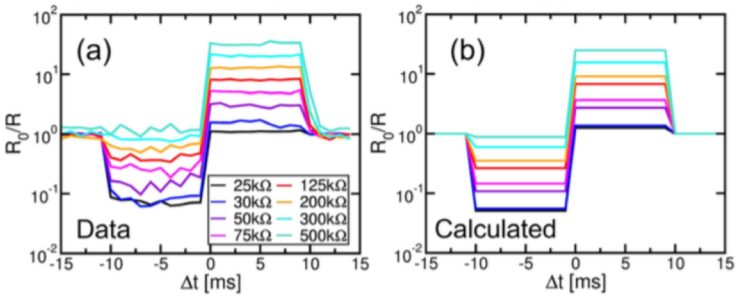
SNN with resistive elements

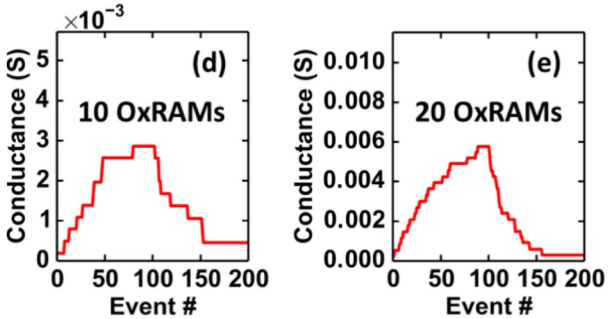


Resistive Synapses







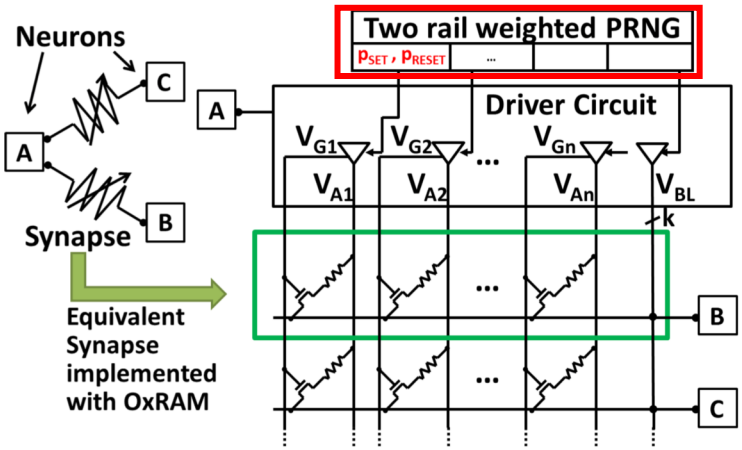


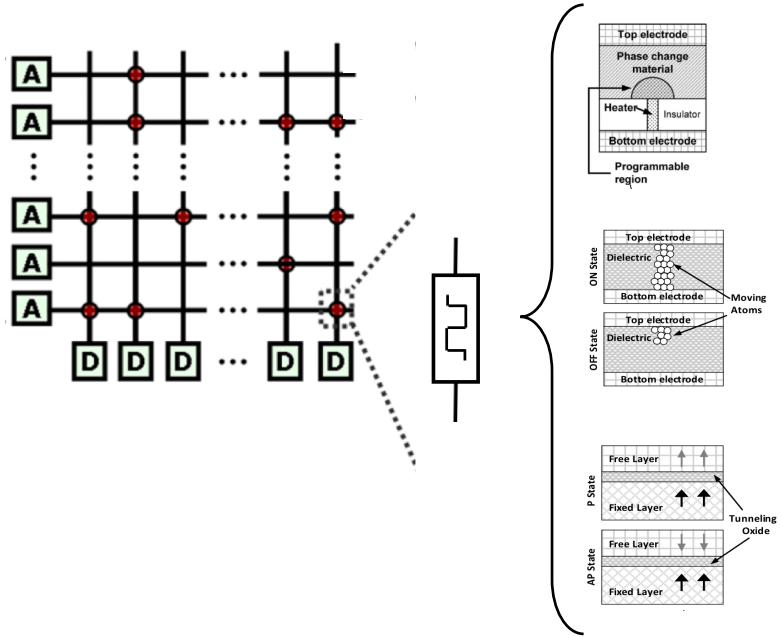
Probabilistic Learning

CEA LETI, D. Garbin, et al., 2015, IEEE T. on Electron Devices
S. Ambrogio, et al., 2016, IEEE T. on Electron Devices

SNN with resistive elements

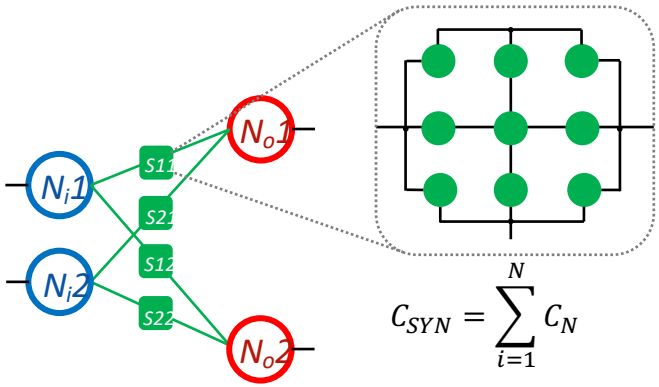
(RRAM/STT-MRAM)
intrinsic stochasticity -> area efficiency





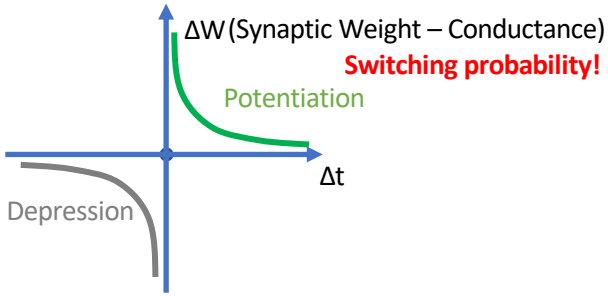
CEA LETI, D. Garbin, et al., 2015, IEEE T. on Electron Devices

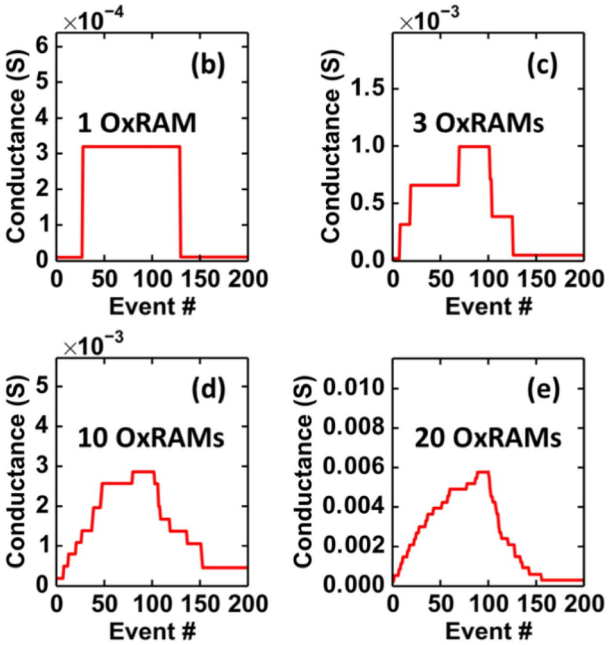
Learning with probabilistic synapses



$$C_{SYN} = \sum_{i=1}^{N} C_N$$

Synaptic levels: N+1

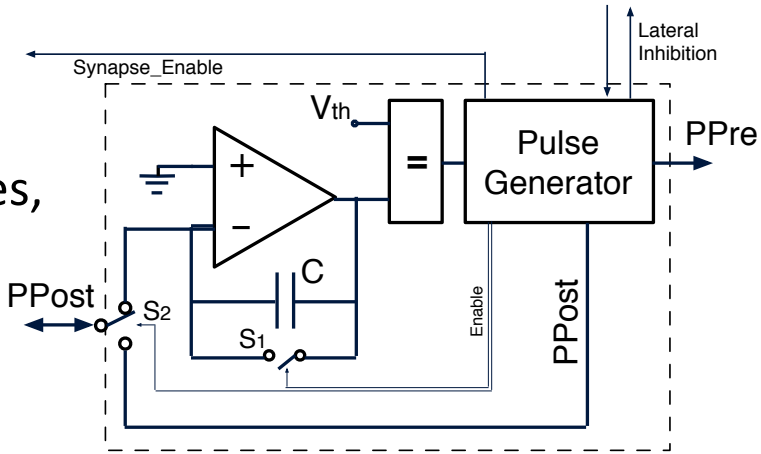




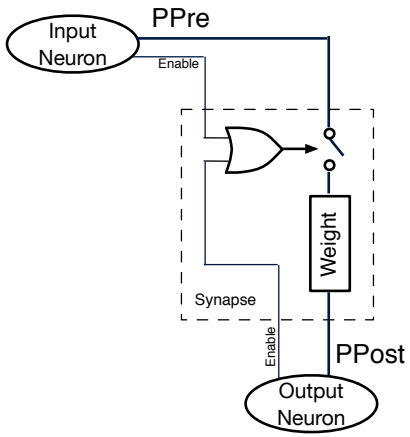
Probabilistic Learning

Design of HW implemented SNNs

- The ***neuron*** has to:
  - translate any signal in train of spikes,
  - have a fast response,
  - have low power consumption.



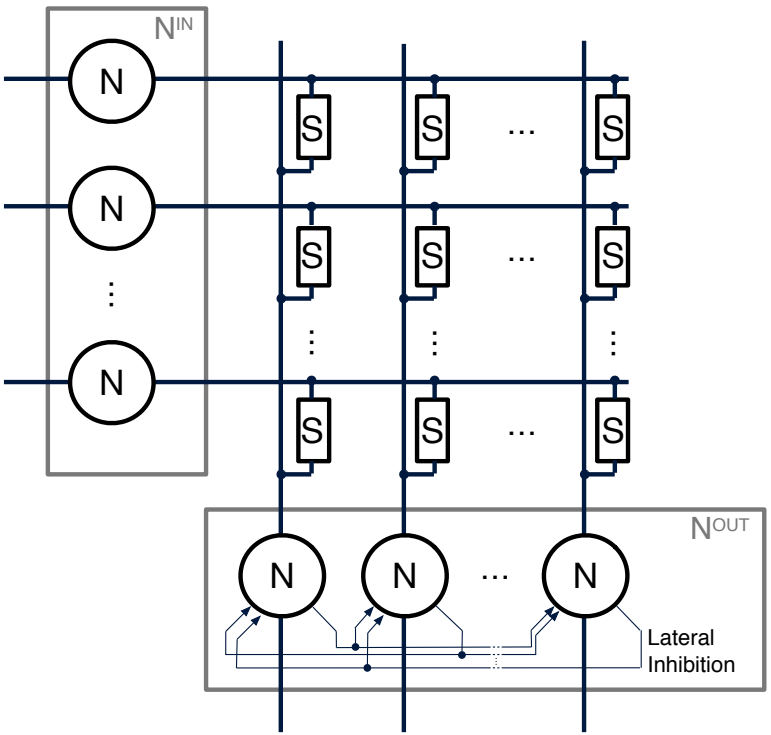
- The ***synaptic weight*** has to:
  - be accessible for learning and inference,
  - be tunable (allow for potentiation/depression),
  - be nonvolatile.



Anghel L., Di Natale G., Miramond B., Vatajelu I., Vianello E., Neuromorphic Computing - From Robust Hardware Architectures to Testing Strategies, VLSI-SoC, 2018

Design of HW implemented SNNs

Spiking Neural Network Design – MNIST example



N^{IN} = 28x28 = 784

$N^{OUT} \geq 10$

S = $N^{IN} x N^{OUT}$

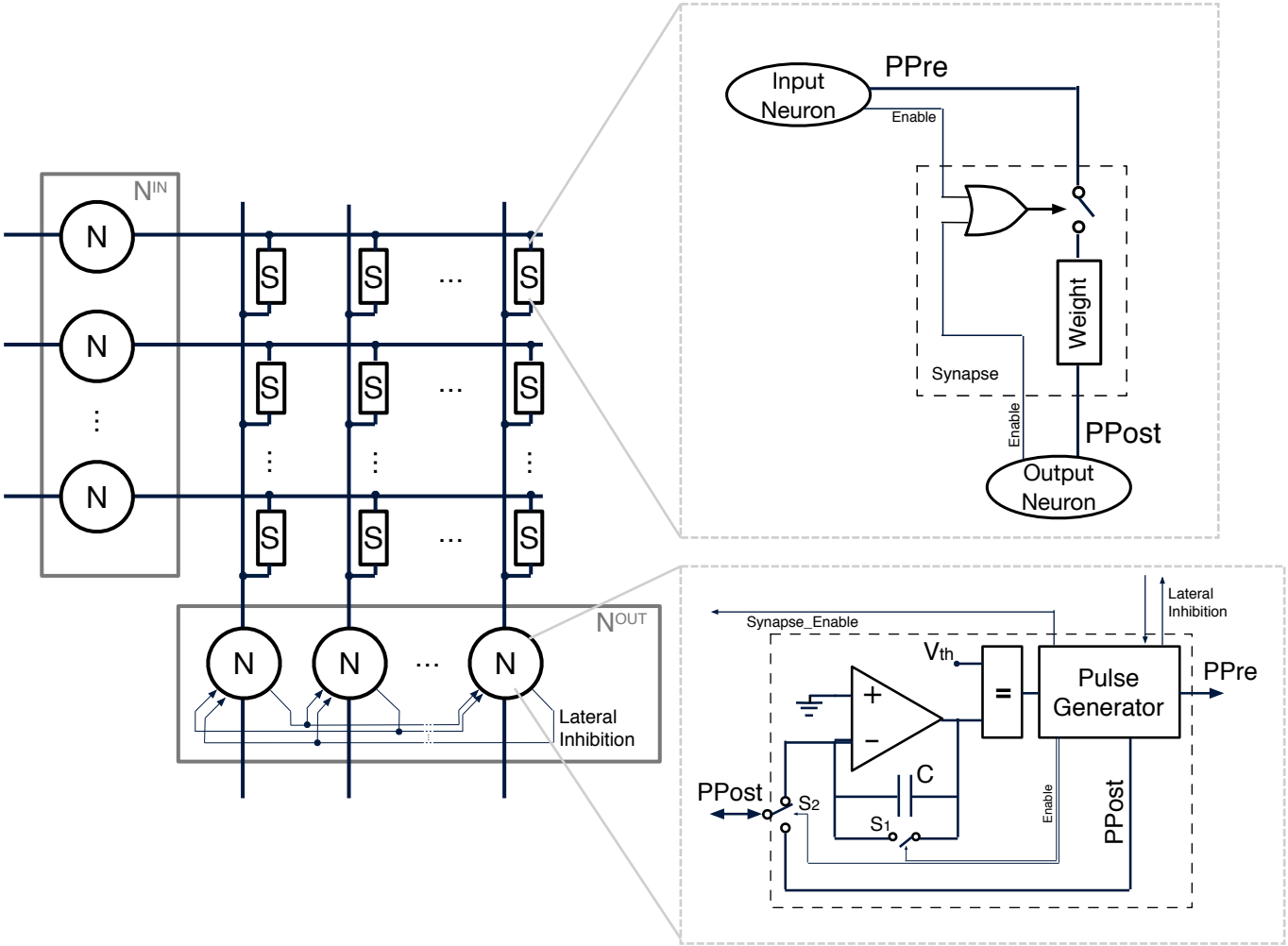
Presentation overview

- Design of HW-implemented SNNs
- Fault modelling
- Fault injection
- Test issues and challenges
- Conclusions

Fault Modelling for SNNs

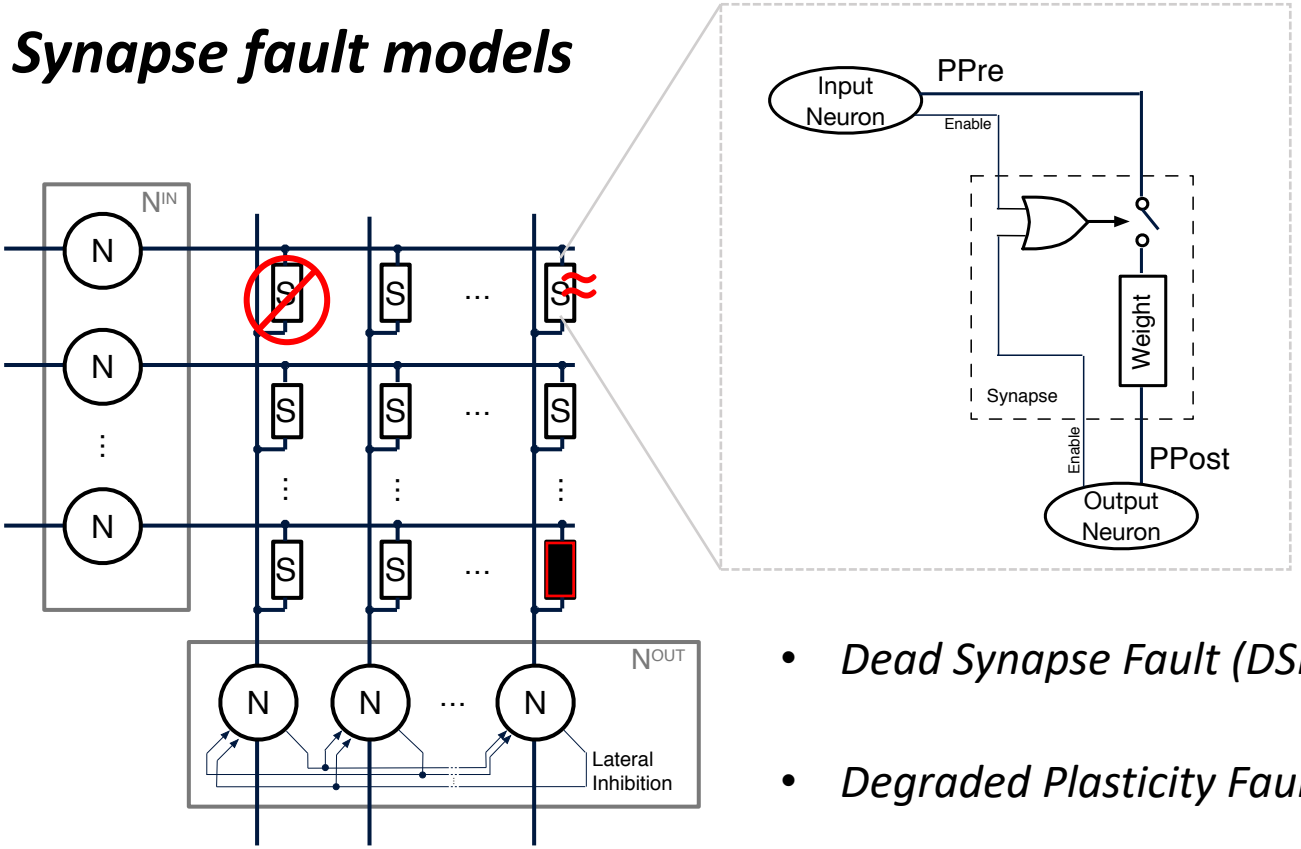
- SNNs with STDP – step towards embedded neuromorphic computing
  - Low-power, compact design, tuneable to application
- High reliability necessary for system integration
  - **Fault modelling**, Test, Design for Dependability

Fault Modelling for SNNs



Fault Modelling for SNNs

Synapse fault models



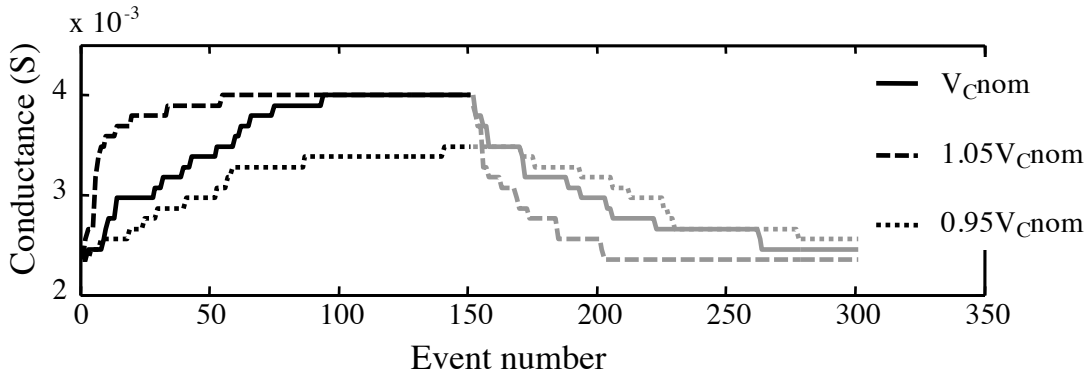
- *Dead Synapse Fault (DSF)*
- *Degraded Plasticity Fault (DPF)*
- *Synapse-Stuck-At (SSA0, SSA1, SSAi)*

Vatajelu I., Di Natale G., Anghel L., Reliability of Hardware-Implemented Spiking Neural Networks (SNN), VTS, 2019

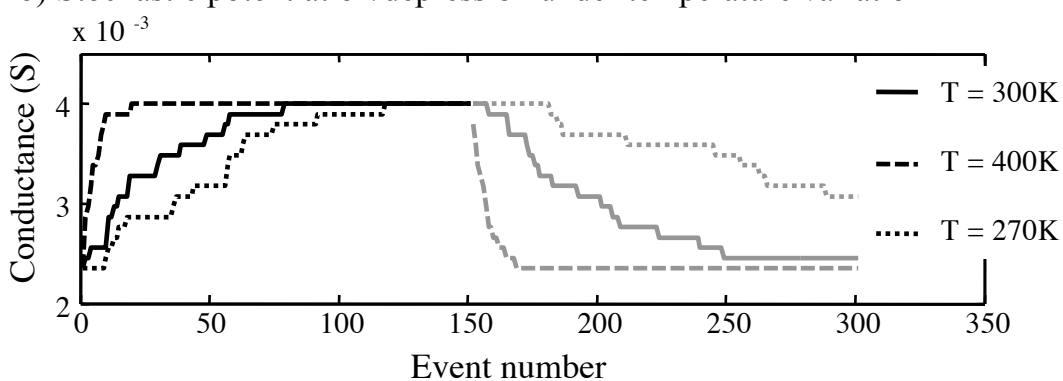
Fault Modelling for SNNs

Synapse fault models – degraded plasticity (nominal 16 conductance levels)

a) Stochastic potentiation/depression under control voltage variation

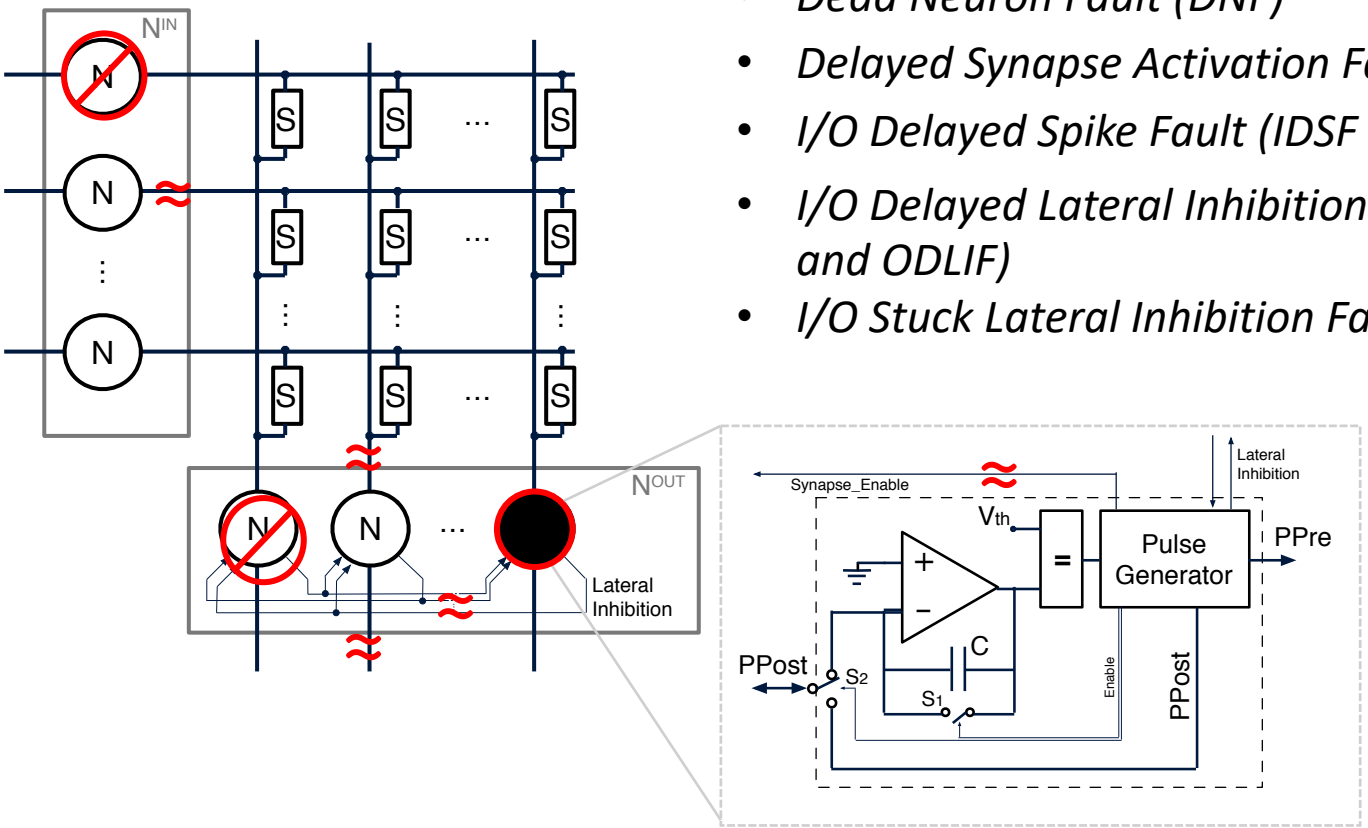


b) Stochastic potentiation/depression under temperature variation



Fault Modelling for SNNs

Neuron fault models

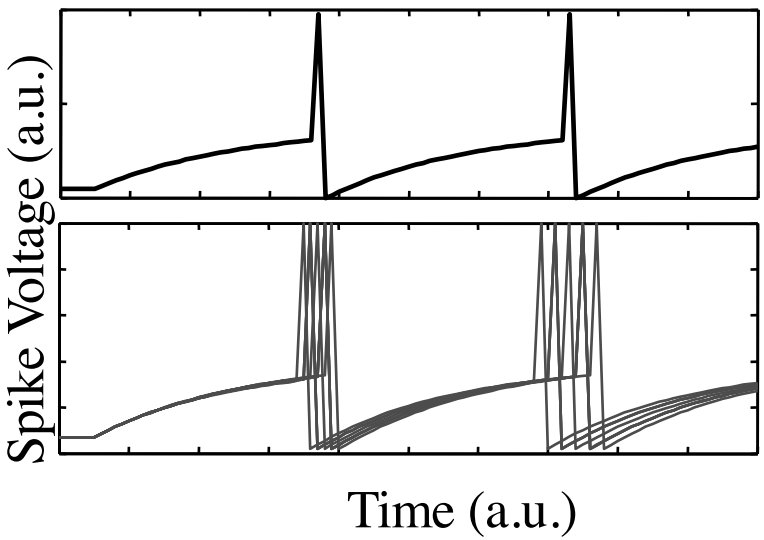


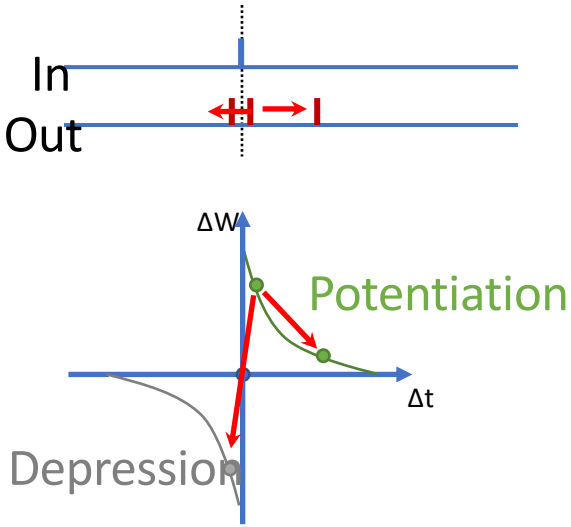
- *Dead Neuron Fault (DNF)*
- *Delayed Synapse Activation Fault (DSAF)*
- *I/O Delayed Spike Fault (IDSF and ODSF)*
- *I/O Delayed Lateral Inhibition Fault (IDLIF and ODLIF)*
- *I/O Stuck Lateral Inhibition Fault (ISLIF, OSLIF)*

Vatajelu I., Di Natale G., Anghel L., Reliability of Hardware-Implemented Spiking Neural Networks (SNN), VTS, 2019

Fault Modelling for SNNs

Neuron fault models - Delayed Spike & Synapse Activation



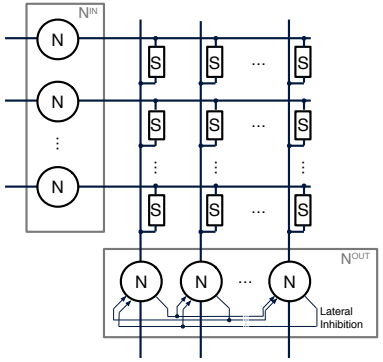


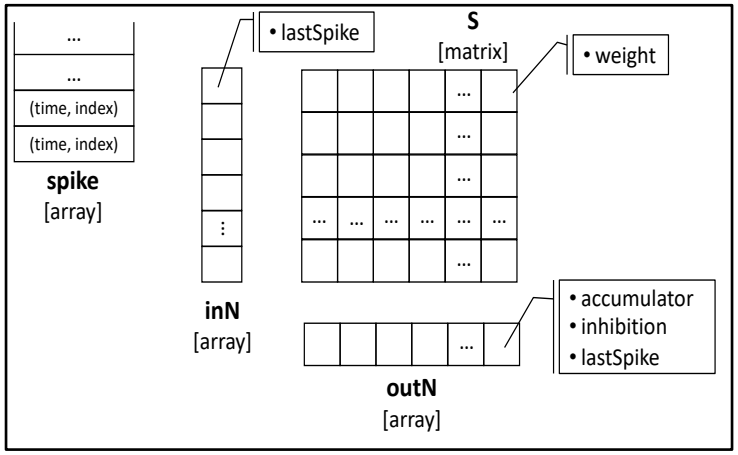
Presentation overview

- Design of HW-implemented SNNs
- Fault modelling
- Fault injection
- Test issues and challenges
- Conclusions

Fault Injection in SNNs

Network simulator (designed for fault-injection)



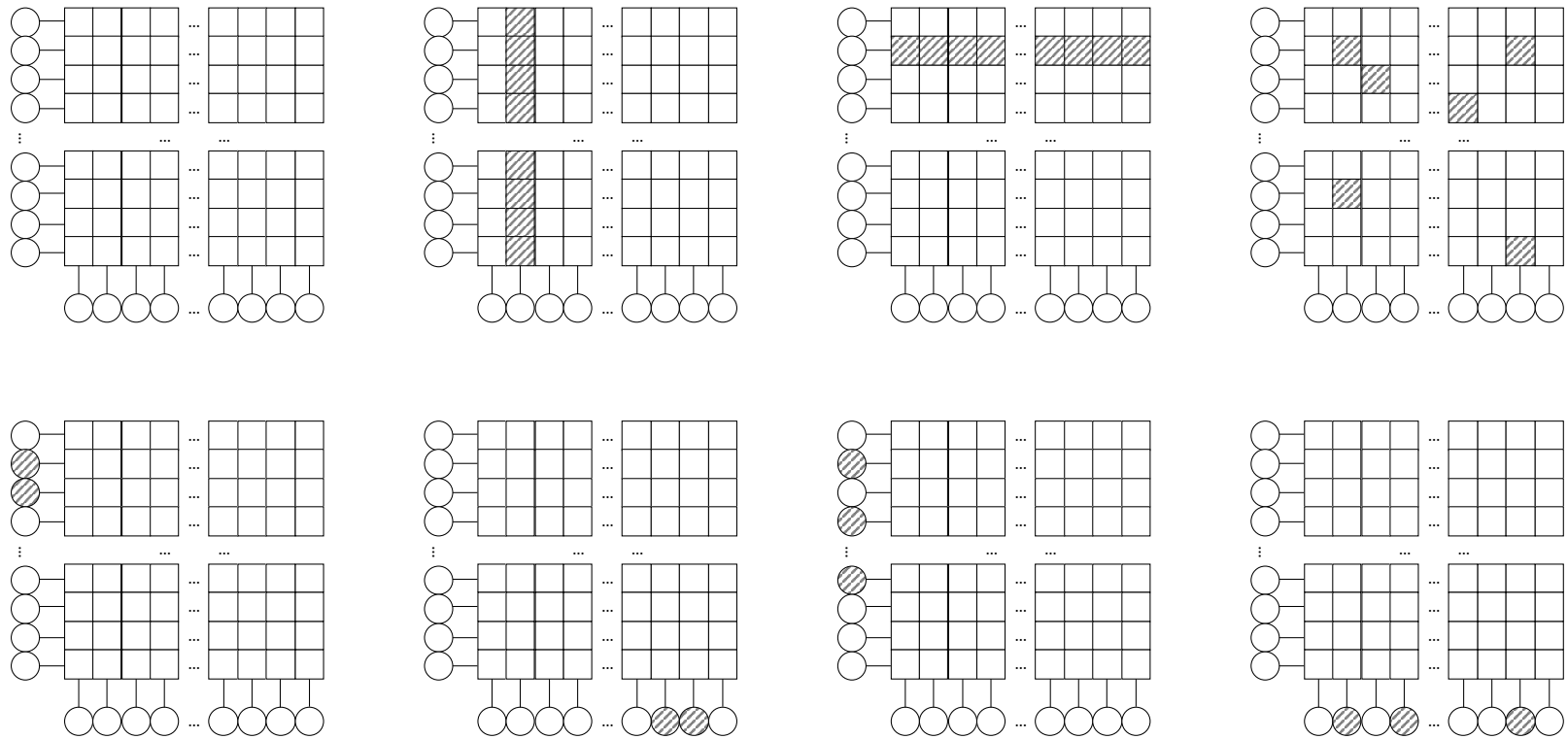



```
foreach spike as (time, index)
    // It updates the lastSpike timing for the input
    // neuron that generated the spike
    inN[index].lastSpike = time

    // It updates the accumulators of the corresponding
    // neurons (if the neuron is not inhibited)
    foreach outNeurons as o
      if (outN[o].inhibition < time)
        outN[o].accumulator += S[index][o].weight

    // If the accumulator of an output neuron
    // overpasses the threshold, it fires the spike
    foreach outNeurons as o
      if (outN[o].accumulator > threshold)
        // for all the output neurons
        // the inhibition time is set
        // and the accumulator is reset
        foreach outNeurons as o2
          outN[o2].inhibition = time + RefractoryPeriod
          outN[o2].accumulator = 0
        // For the neruon that spiked,
        // the lastSpike information is updated
        outN[o].lastSpike = time
        // For all synapses connected to the
        // neuron that spiked, the weight is updated
        foreach inNeurons as i
          S[i][o].weight += potdep (inN[i].lastSpike, time)
```

Vatajelu I., Di Natale G., Anghel L., Reliability of Hardware-Implemented Spiking Neural Networks (SNN), VTS, 2019

Fault Injection in SNNs

Scenarios

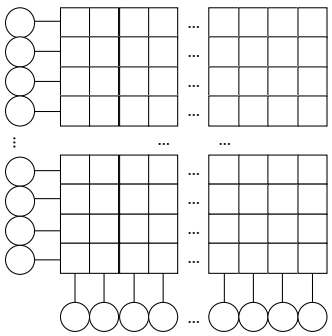


Fault-free element

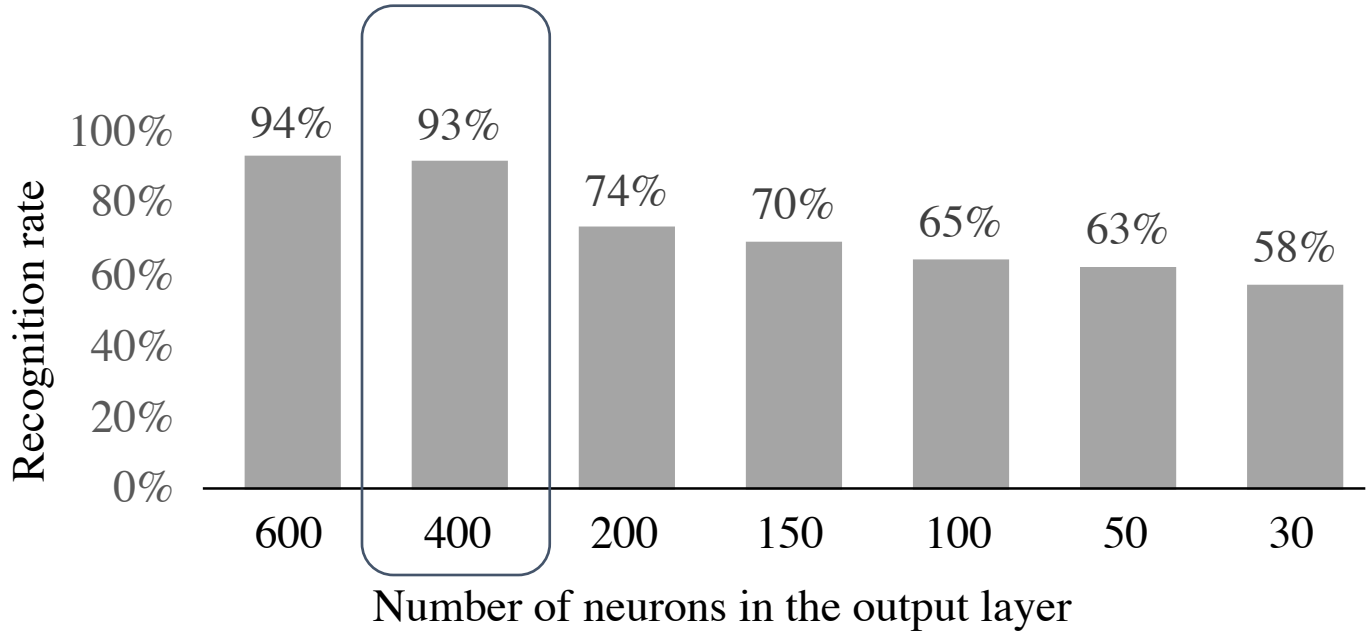
Fault location: synapse or neuron

Vatajelu I., Di Natale G., Anghel L., Reliability of Hardware-Implemented Spiking Neural Networks (SNN), VTS, 2019

Fault Injection in SNNs

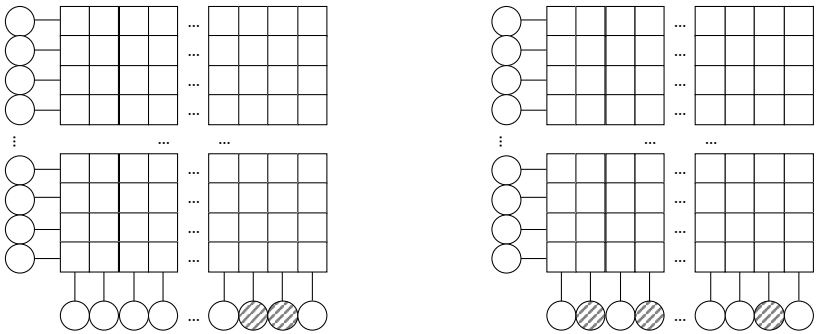


- Fully connected SNN with STDP
- Spike frequency coding
- Winner takes it all – inhibition
- Synaptic device (plasticity) implemented with 9 levels of conductance



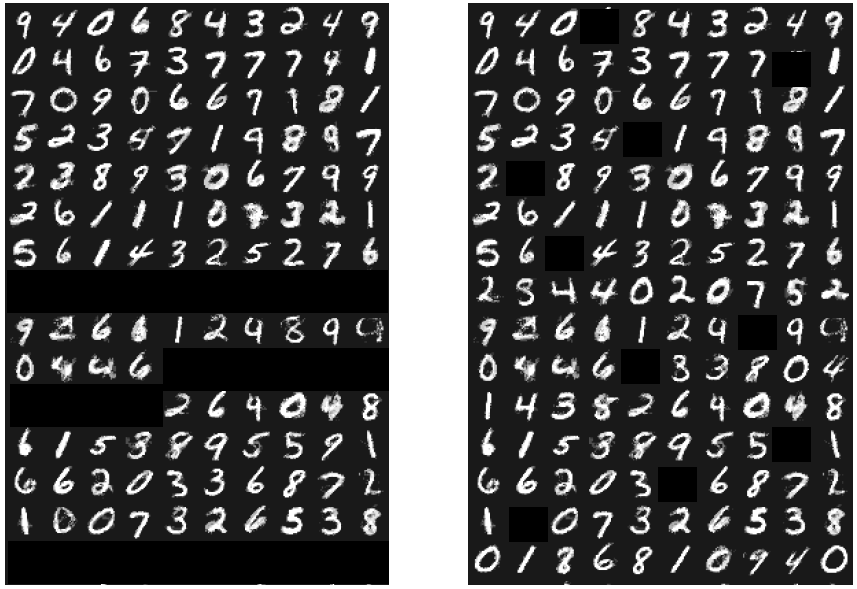
Vatajelu I., Di Natale G., Anghel L., Reliability of Hardware-Implemented Spiking Neural Networks (SNN), VTS, 2019

Fault Injection in SNNs

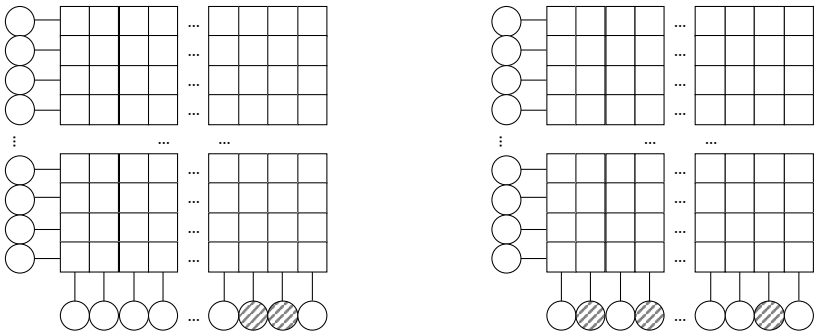


- Dead Neuron Fault (output layer)

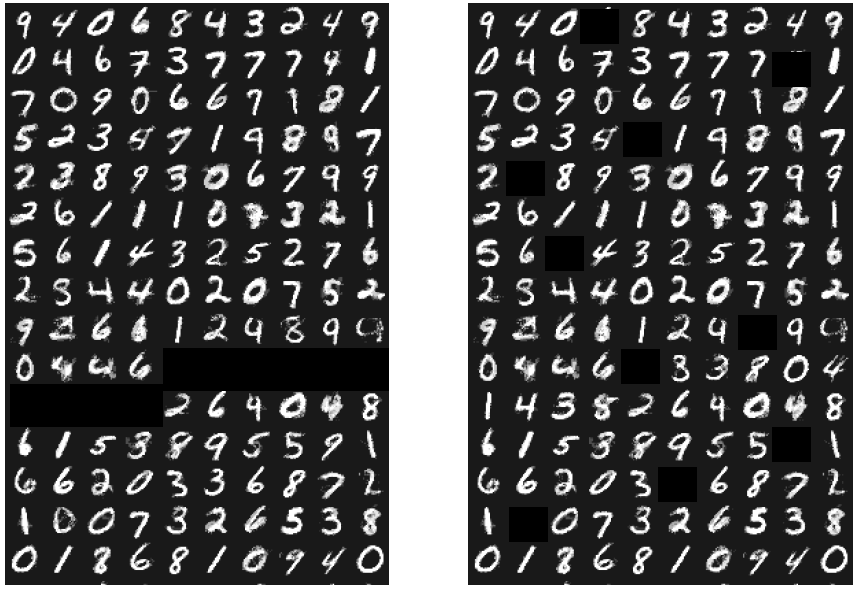
Faults injected at inference



Fault Injection in SNNs

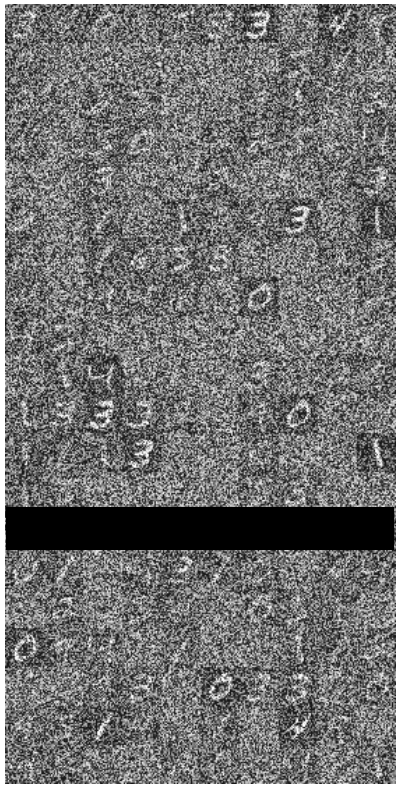


Faults injected at inference

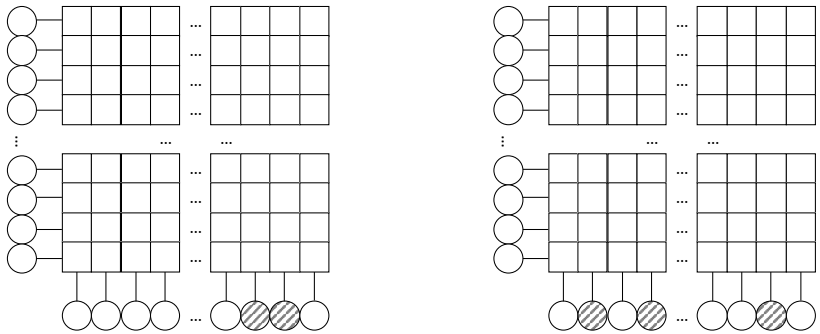


- Dead Neuron Fault (output layer)

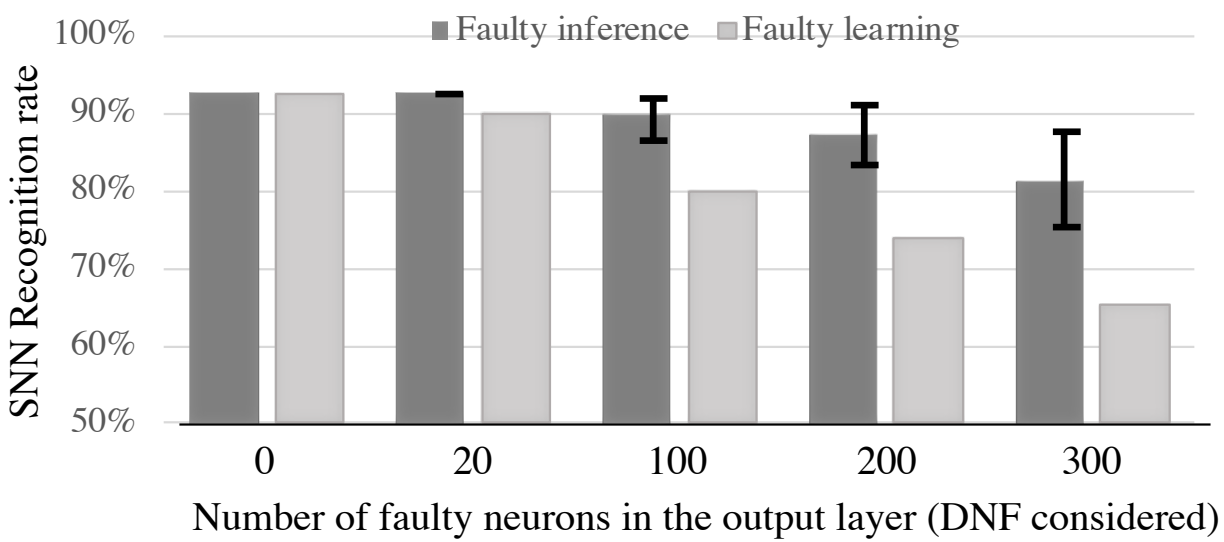
Faults injected at learning



Fault Injection in SNNs

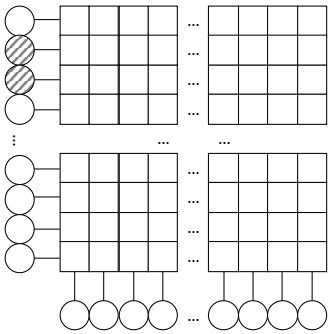


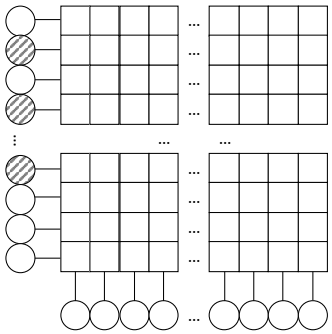
- Dead Neuron Fault (output layer)



Vatajelu I., Di Natale G., Anghel L., Reliability of Hardware-Implemented Spiking Neural Networks (SNN), VTS, 2019

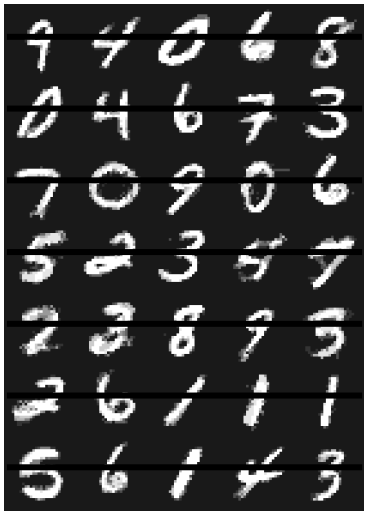
Fault Injection in SNNs



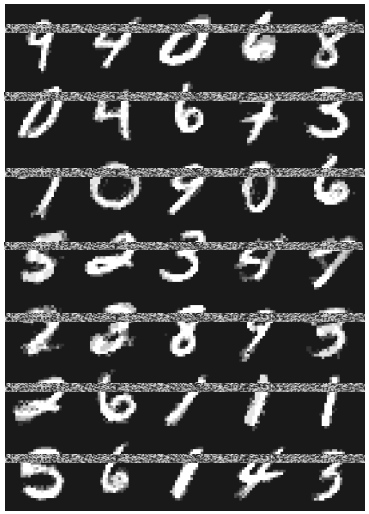


- Dead Neuron Fault (input layer)

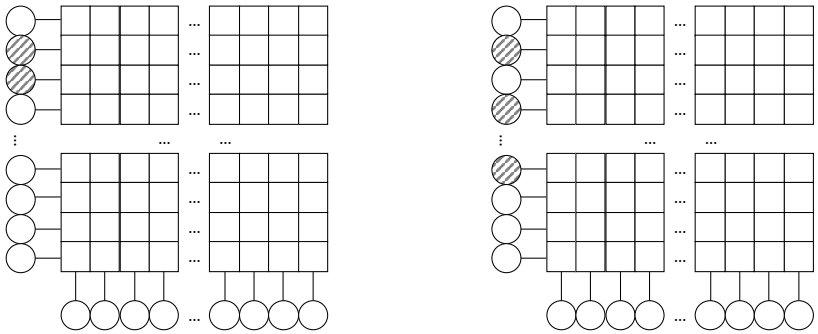
Faults injected at inference



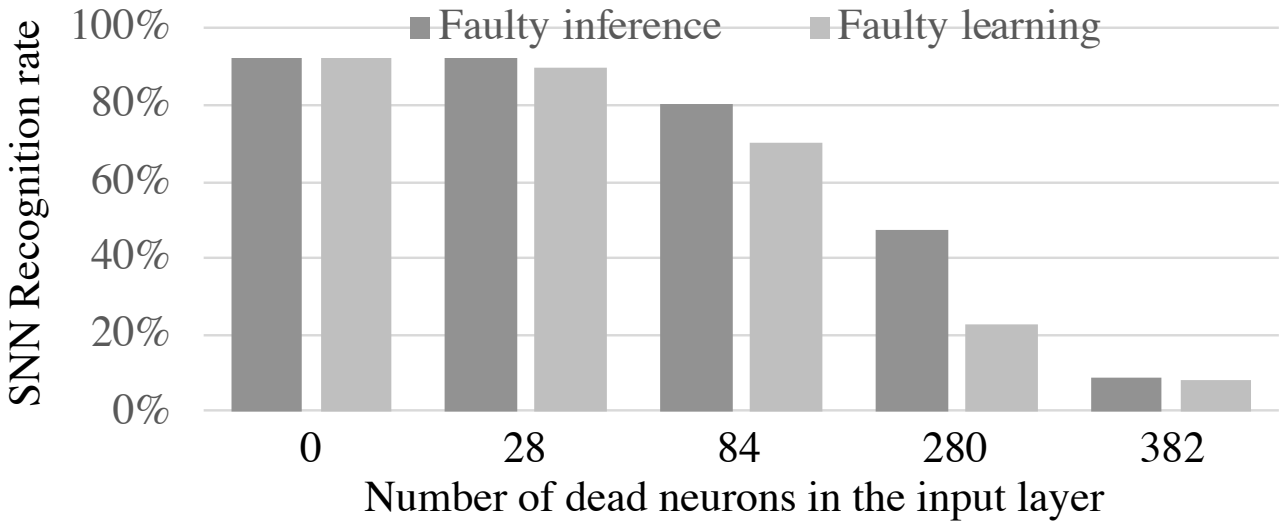
Faults injected at learning



Fault Injection in SNNs

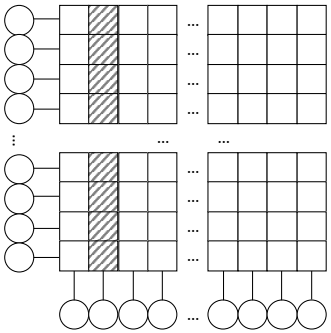


- Dead Neuron Fault (input layer)

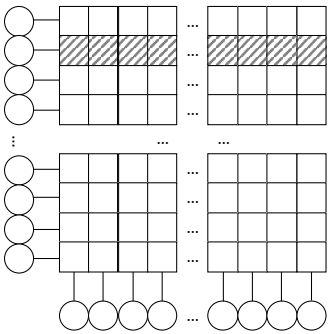


Fault Injection in SNNs

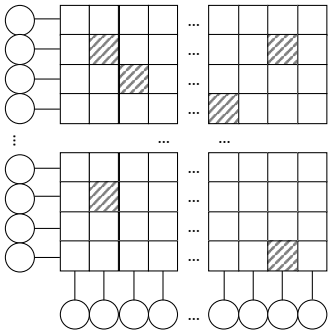
- Dead Synapse Fault



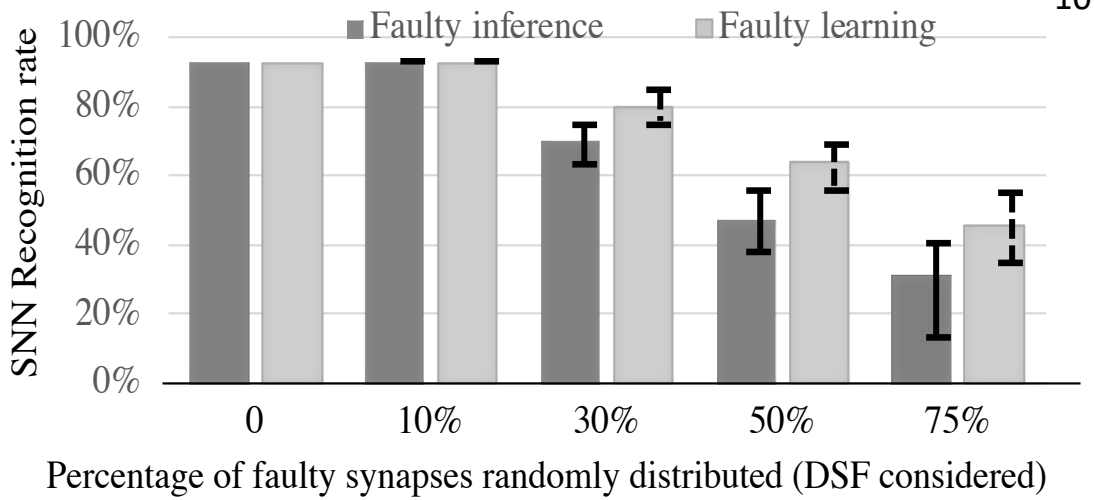
Equivalent 1DNF(out)



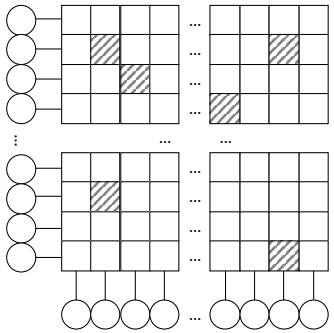
Equivalent 1DNF(in)



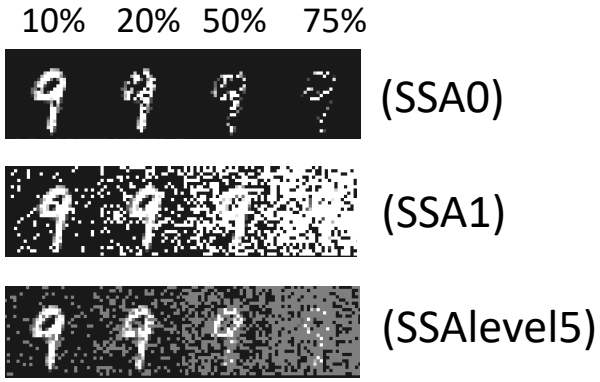
10% 20% 50% 75% (DSF)

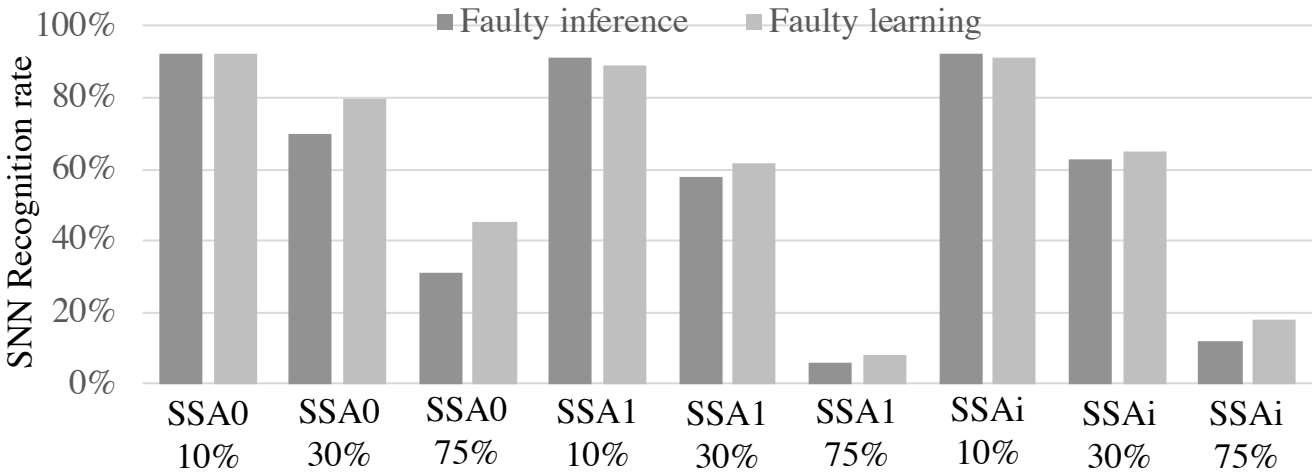


Fault Injection in SNNs



- Synapse-Stuck-At (SSA0, SSA1, SSAi)





Summary

	Fault model	Learning	Inference	Location?
Synapse	DSF			yes
	DPF			yes
	SSAx			yes
Neuron	DNF - in			yes - strong
	DNF - out			no
	IDSF/ODSF			yes
	IDLIF/ODLIF		NA	no
	ISLIF/OSLIF		NA	no

!

Number/magnitude of faults

Recognition rate

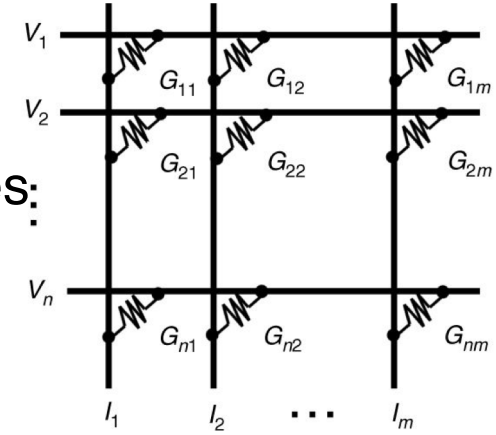
100% 0%

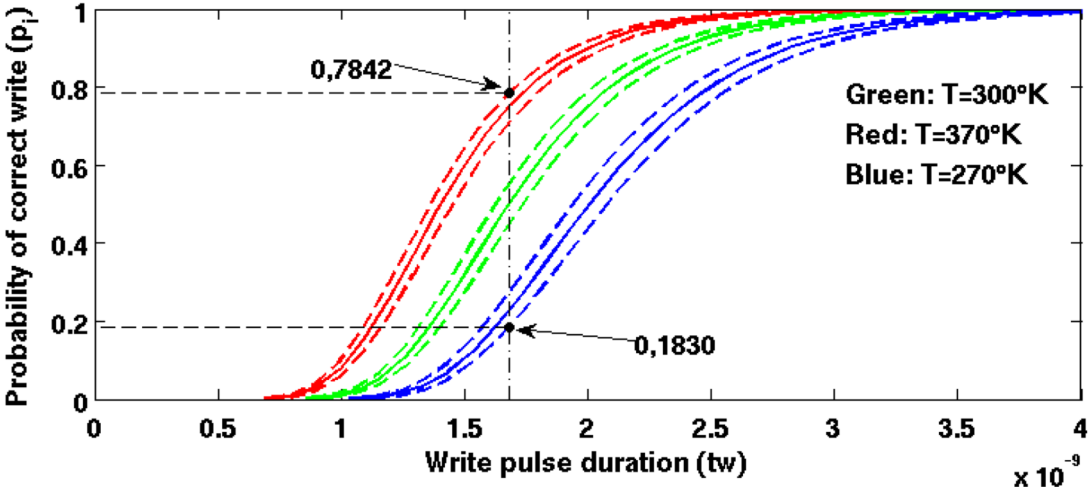
Presentation overview

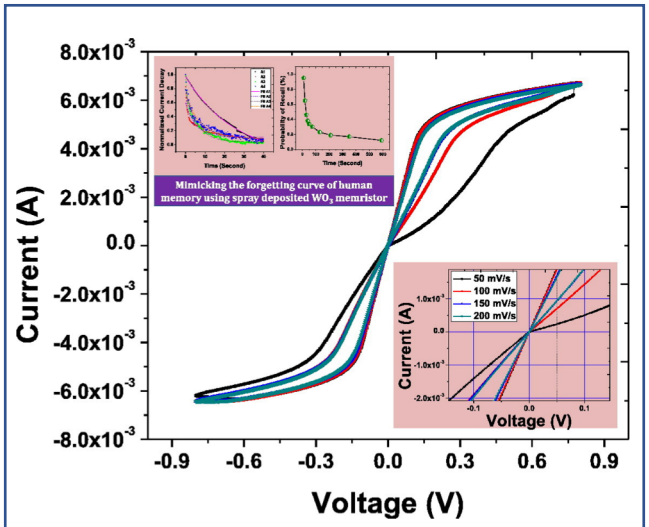
- Design of HW-implemented SNNs
- Fault modelling
- Fault injection
- Test issues and challenges
- Conclusions

Novelties

- Synapses:
  - Analog memories
  - Stochastic memories:
- Regular analog structures







T.D.Dongale, et al., Microelectronic Engineering, 2017

Classical Test: Fault Modeling & ATPG

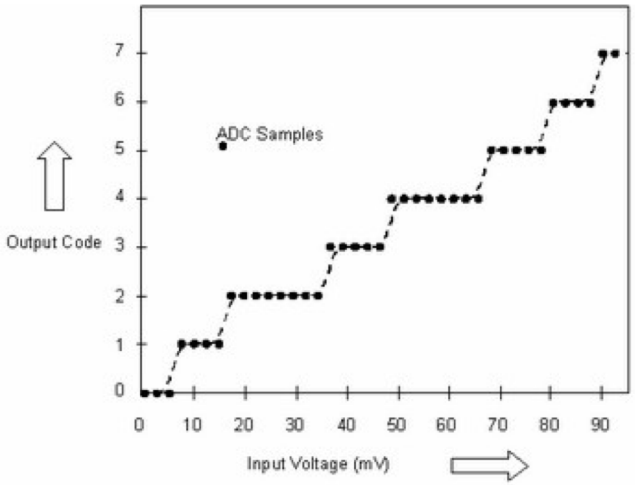
- Fault modeling: to model the effect of physical defects
- Fault simulation: to simulate the behavior of the circuit in presence of a fault
- Test generation: to generate input vectors able to excite faults and to manifest their presence

Not deterministic behavior

Directions

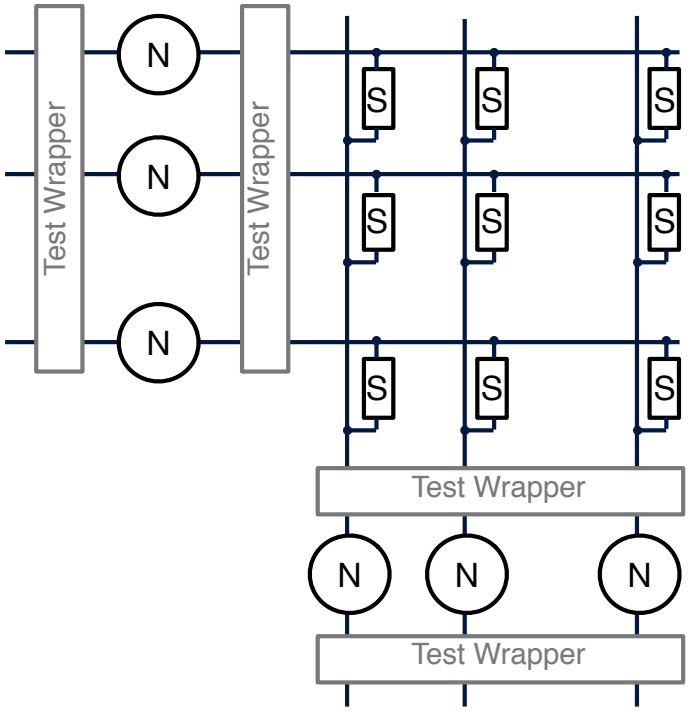
- Get inspired by techniques like:
  - Iddq
  - ADC testing
  - TRNG testing

NIST Test
Frequency
Block frequency
Cumulative Sums
Runs
Longest Runs of 1's
Rank
FFT
Non-overlapping Template
Overlapping Template
Universal Statistical
Approximate Entropy
Random Excursion
Random Excursions Variant
Serial
Linear Complexity



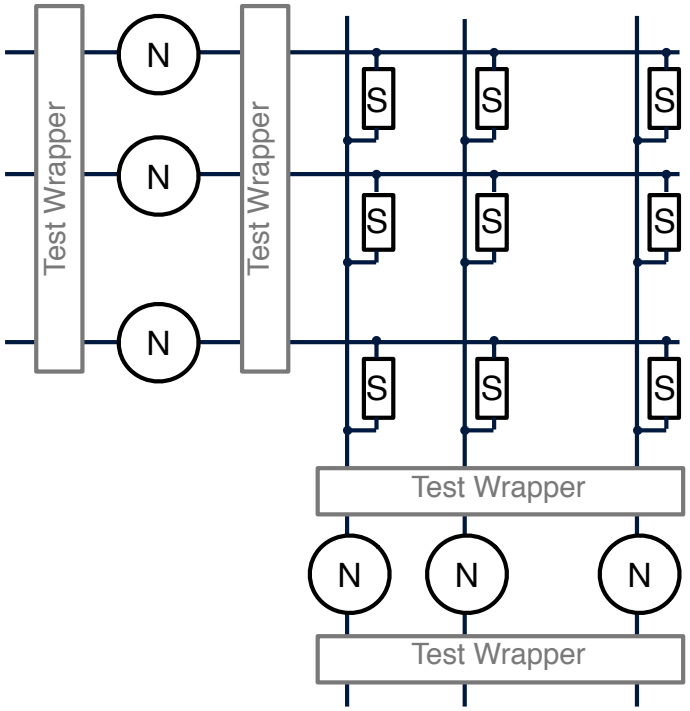
Test Access Mechanism

- Test isolation
- Analog behavior
- Large amount of elements



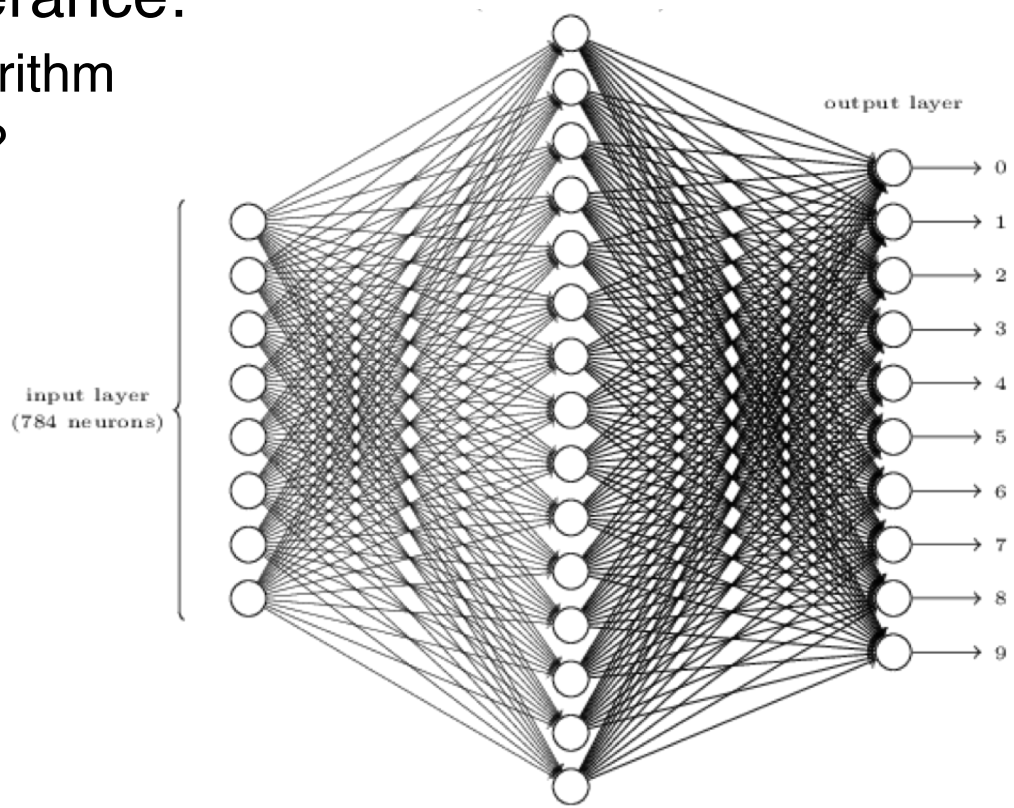
Test Access Mechanism

- Test isolation
- Analog behavior
- Large amount of elements
- BIST
  - e.g., BISTed neurons



Intrinsic Fault Tolerance of SNNs

- Intrinsic Fault tolerance:
  - Very high in algorithm
  - What about HW?



Intrinsic Fault Tolerance of SNNs

- If fault tolerance is guaranteed by the network:
  - Yield/defect levels
  - Identify critical elements – intensified test
  - Application dependency

Conclusions

- SNNs with STDP – step towards embedded neuromorphic computing
  - Low-power, compact design, tuneable to application
- High reliability necessary for system integration
  - **Fault modelling**, Test, Design for Dependability
- Fault models for SNN with STDP
  - Synapse & neuron
  - Learning & inference

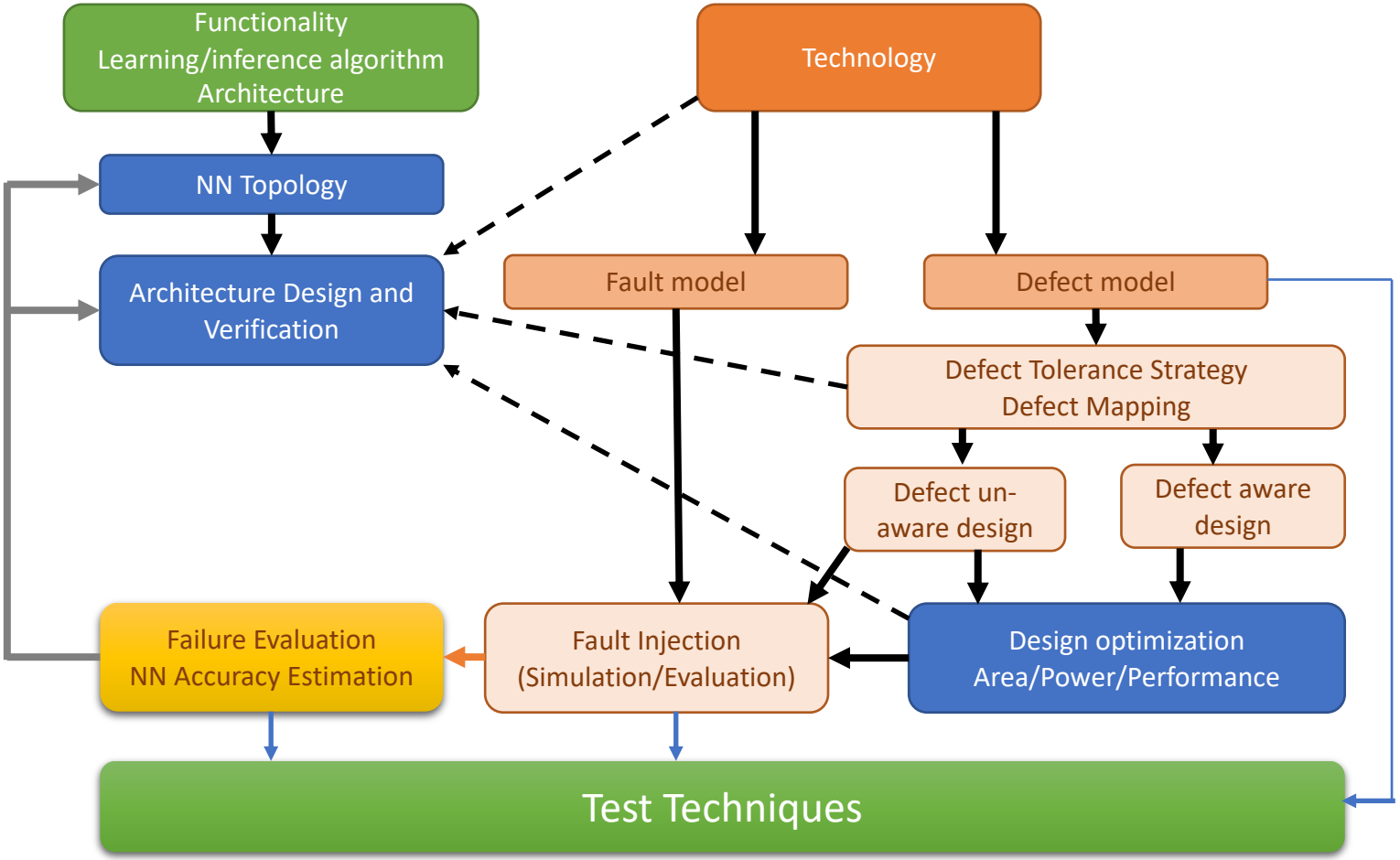
Conclusions

- Testing is challenging
  - No single ”best” design => multiple test strategies
  - Main novelties:
    - Insertion of test wrappers
    - Analog ”memory” testing
    - Probabilistic (random) ”memory” testing
    - Structural or Functional?

Challenging due to analog components

Challenging due to probabilistic behavior

Conclusions



Spiking Neural Networks with STDP

From Robust Hardware Architectures to Testing Strategies

Elena Ioana Vătăjelu, Giorgio Di Natale, Lorena Anghel

TIMA Laboratory, Grenoble, France