UNIVERSITÄT ZU LÜBECK
INSTITUT FÜR TECHNISCHE INFORMATIK

Fault Analysis in Early Design Steps for AI Application

18.11.2021

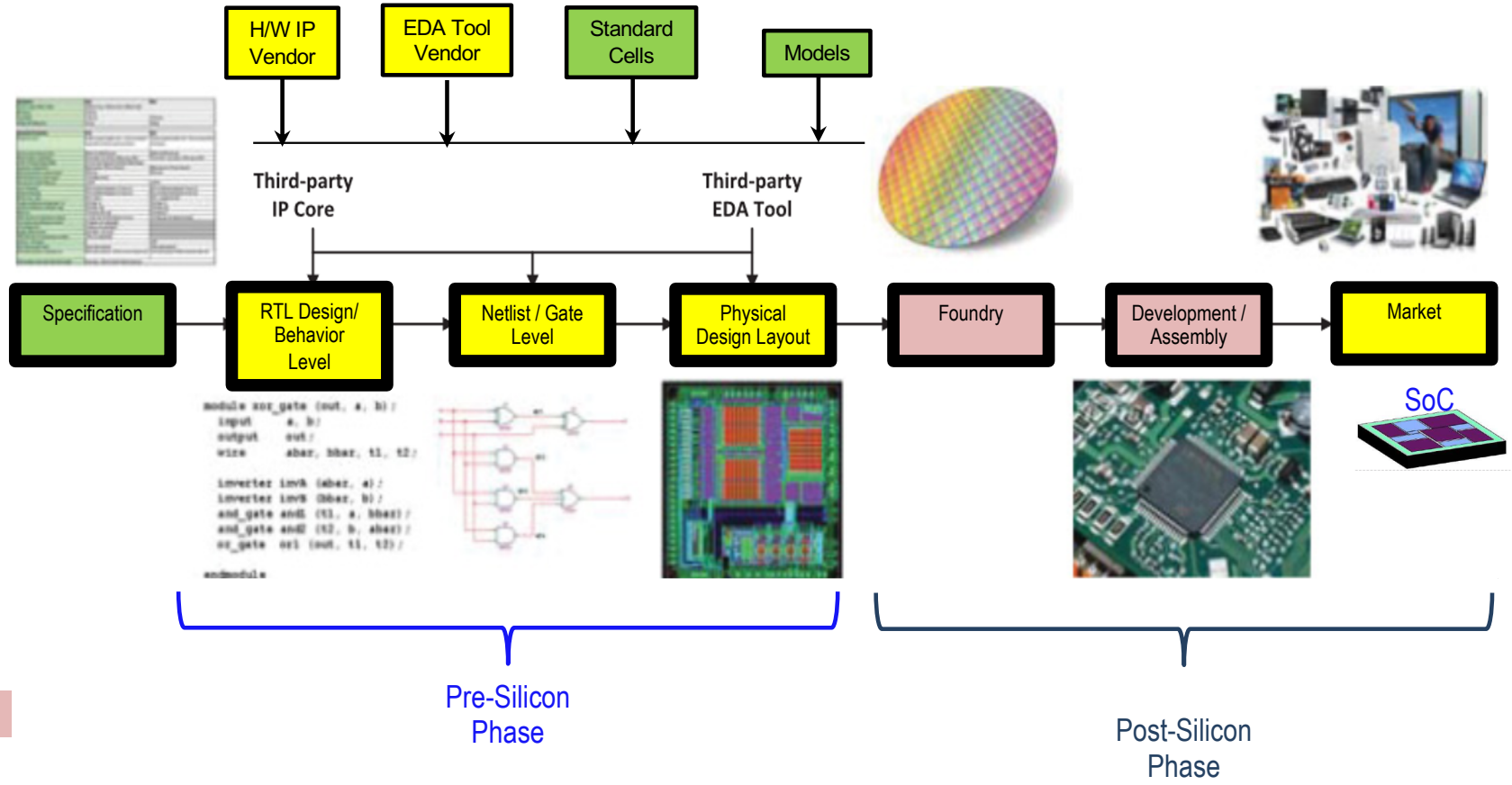
Prof. Dr.-Ing. Mladen Berekovic

Universität zu Lübeck
Institut für Technische Informatik
www.iti.uni-luebeck.de

Outline

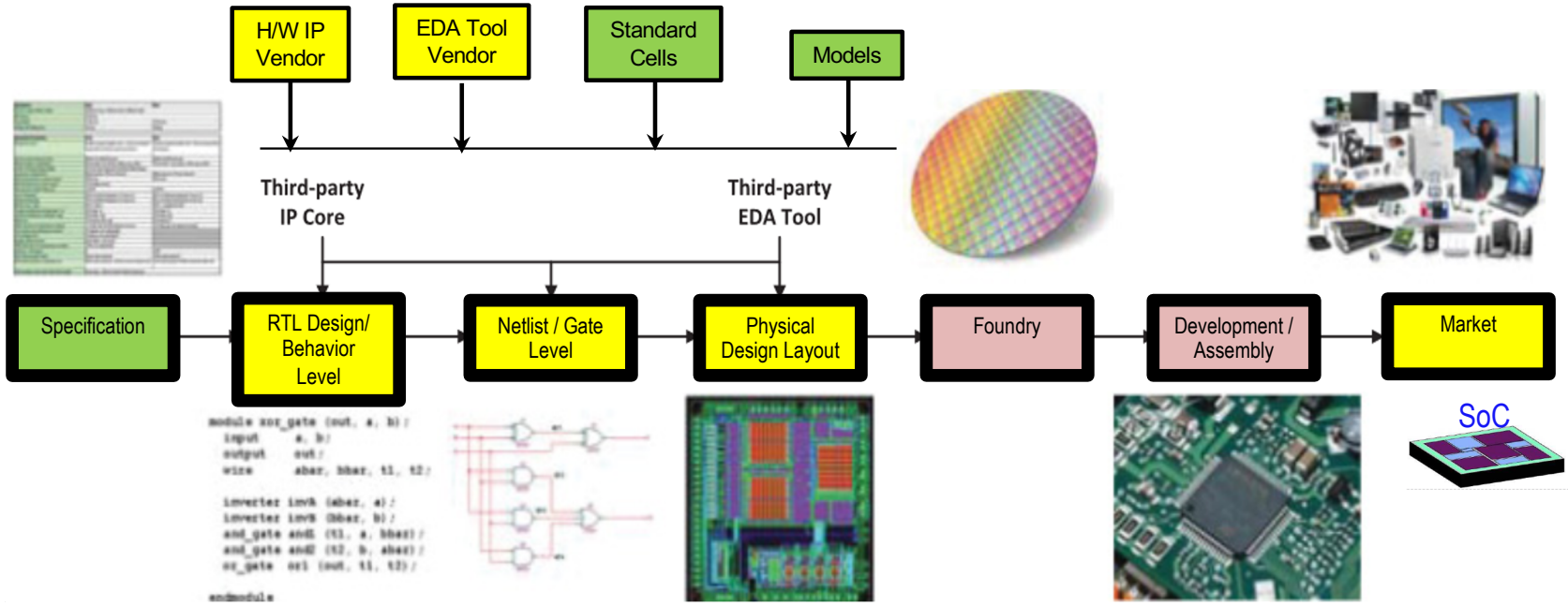
- Semiconductor Supply Chain
- Verification Life Cycle
- Classical EDA Flow
- SystemC model in EDS: Neural Network for XOR function
- Fault Injection and Verification: XOR Trained Model.
- KI-Pro as a Case Studay

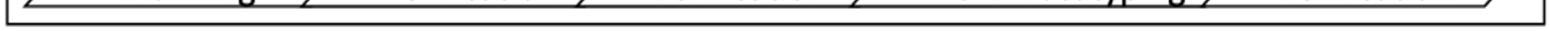
Semiconductor Supply Chain (SSC)



[XIA 16]

Verification Life Cycle



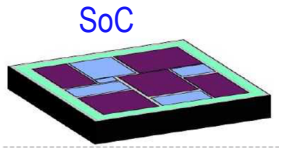


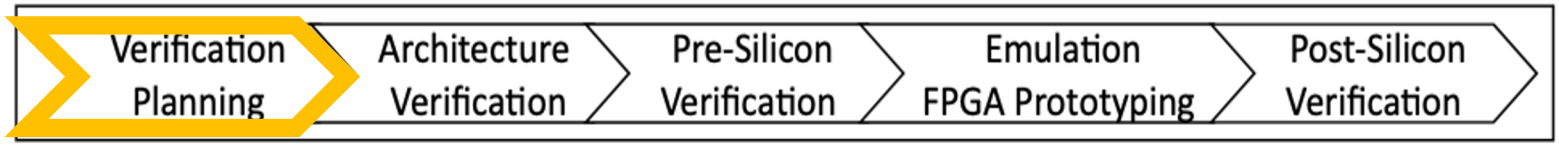
[CHE 17]

[XIA 16]

Verification Life Cycle

- Verification planning
  - starts with the product planning, and continues during the system development phase.
  - defines necessary IPs.
  - defines connection and communication interfaces,
  - determines various power, performance, security, and energy targets

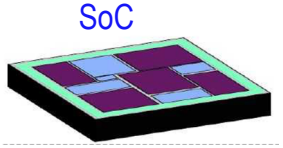


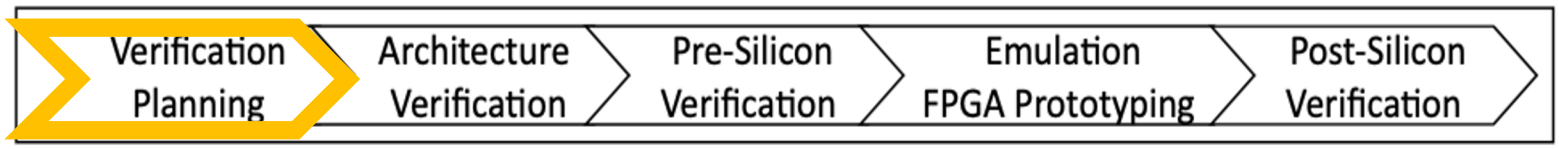


[CHE 17]

Verification Life Cycle

- Verification planning
  - starts with the product planning, and continues during the system development phase.
  - defines necessary IPs.
  - defines connection and communication interfaces,
  - determines various power, performance, security, and energy targets
  - creates appropriate test plans, test cards, and various monitors, checker, exercisers, etc.

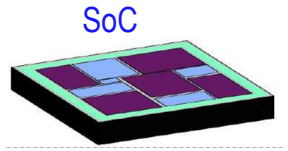


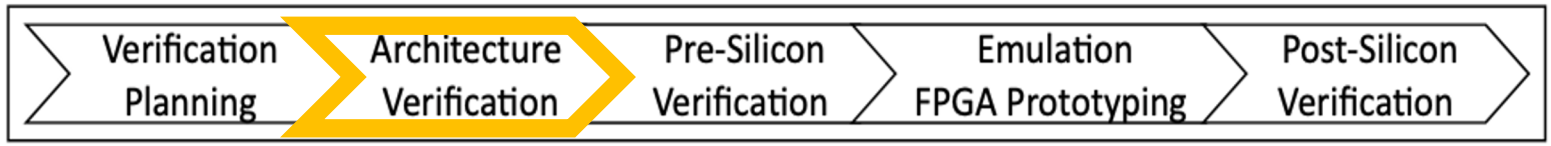


[CHE 17]

Verification Life Cycle

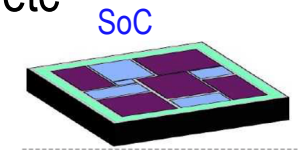
- Architecture verification
  - defines functional parameters of the design (cache size, pipeline depth,etc).
  - determines communication protocols among IPs
  - determines power and performance management schemes, etc



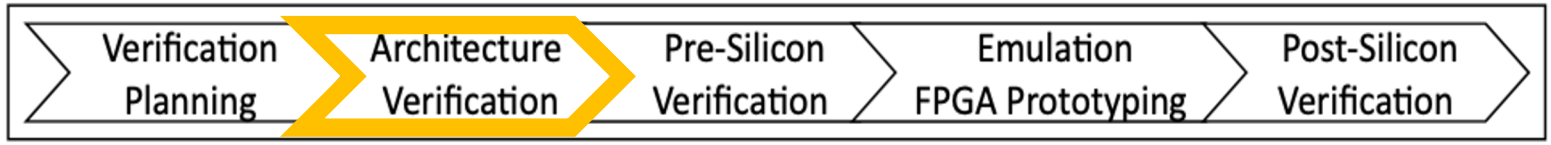


Verification Life Cycle

- Architecture verification
  - defines functional parameters of the design (cache size, pipeline depth,etc).
  - determines communication protocols among IPs
  - determines power and performance management schemes, etc

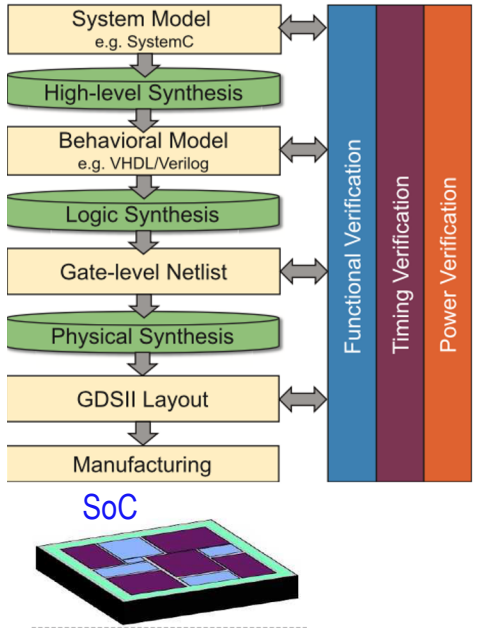


- Most important verification activities:
  - Functional verification: to verify communication protocols using system Model.



Verification Life Cycle

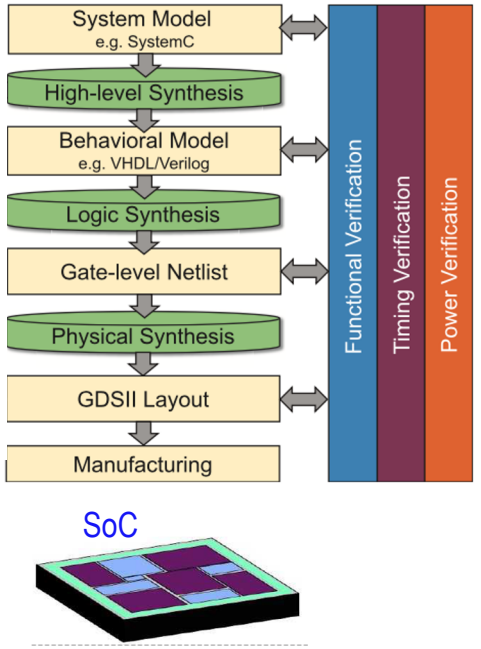
- IP verification team:
  - performs the verification of the IP.
  - The objective is to ensure that the IP on its own functions as expected.



[CHE 17] [KNE 20]

Verification Life Cycle

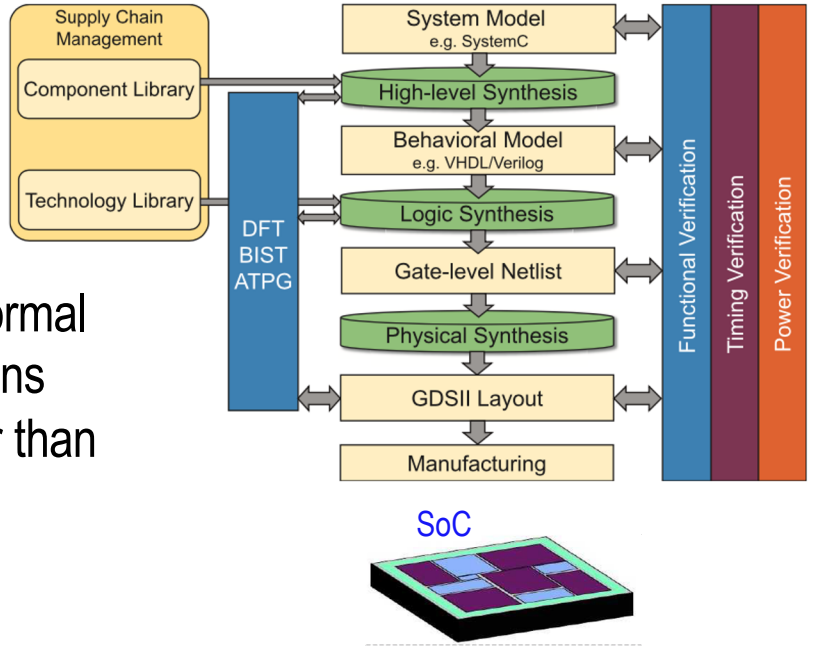
- IP verification team:
  - performs the verification of the IP.
  - The objective is to ensure that the IP on its own functions as expected.
- SoC team:
  - Integrates the IPs into an (evolving) SoC model
  - Perform system-level verification.



[CHE 17] [KNE 20]

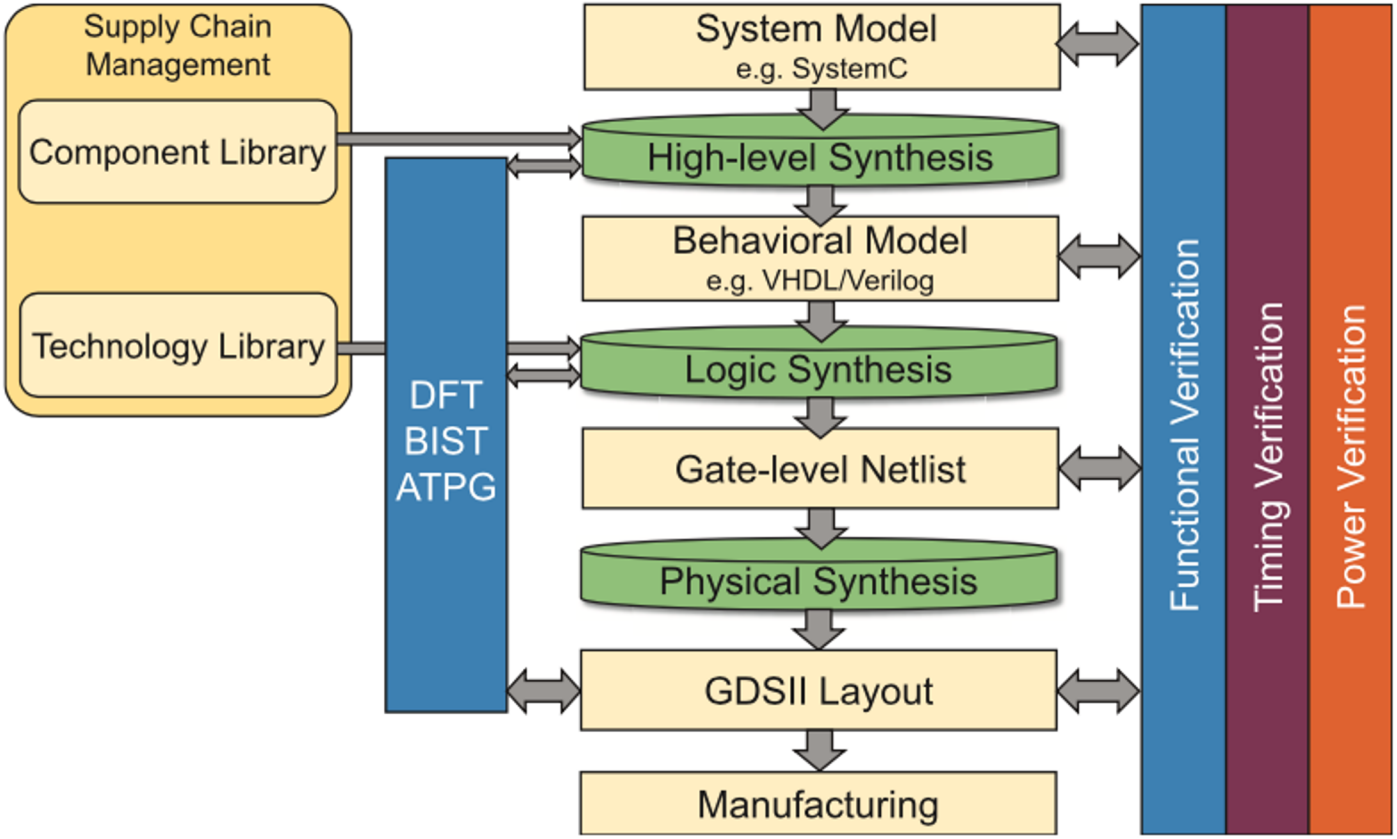
Verification Life Cycle

- Shrinking verification time
- Limited tool scalability
  - there has been a growing trend in formal methods to target specific applications (e.g., security, deadlock, etc.) rather than a complete proof of functional correctness
- Specification capture
  - A key challenge in the applicability of verification today is the lack of specifications
- Power management challenges



[CHE 17] [KNE 20]

Classical EDA Flow



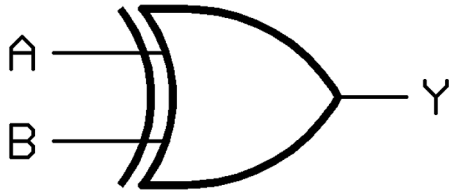
[KNE 20]

SystemC model in EDS for AI

- Training of Neural Network for XOR function
- MLP was chosen for its simplicity
- The inference will be run on a SystemC model
- Faults will be injected in different points in the HW (registers)
- Fault propagation behavior will be shown

SystemC model in EDS for AI

- Neural network will be trained for ***XOR gate.***
- Gate input/output have a 4-bit precision each.
- Dataset contains $2^4 x 2^4 = 256$ samples.
- Training was done in Keras/TensorFlow.



A				B				Y			
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	0	1	0	0	0	1	0
⋮								⋮			
1	1	1	1	1	1	1	1	0	0	0	0

SystemC model in EDS for AI: Neural Network training

- The chosen architecture is 2 neurons at the input with

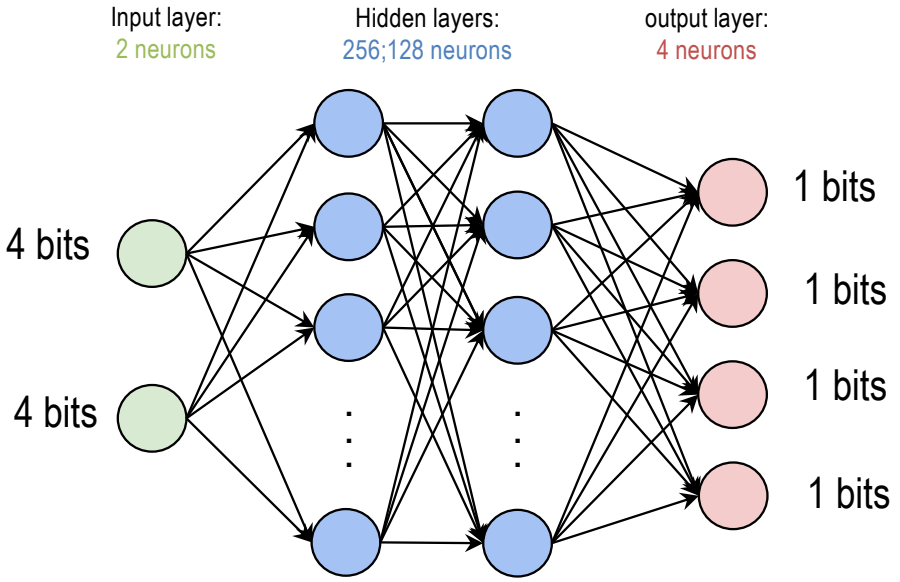
two hidden layers of 256 and 128 neurons and 4 output neurons.

- Rectified Liner (ReLu) as activation function for hidden layers.
- Sigmoid as activation function for output layer.
- Data set not divided into training/test batches because

the Neural Network is doing a specific XOR logic on a tiny dataset.

  → Overfitting is not a problem for this example.

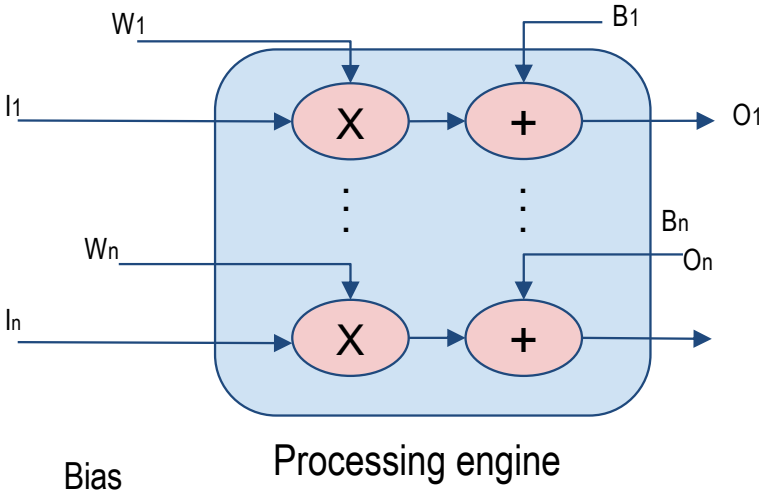
- Weights are extracted on 32-bit floats

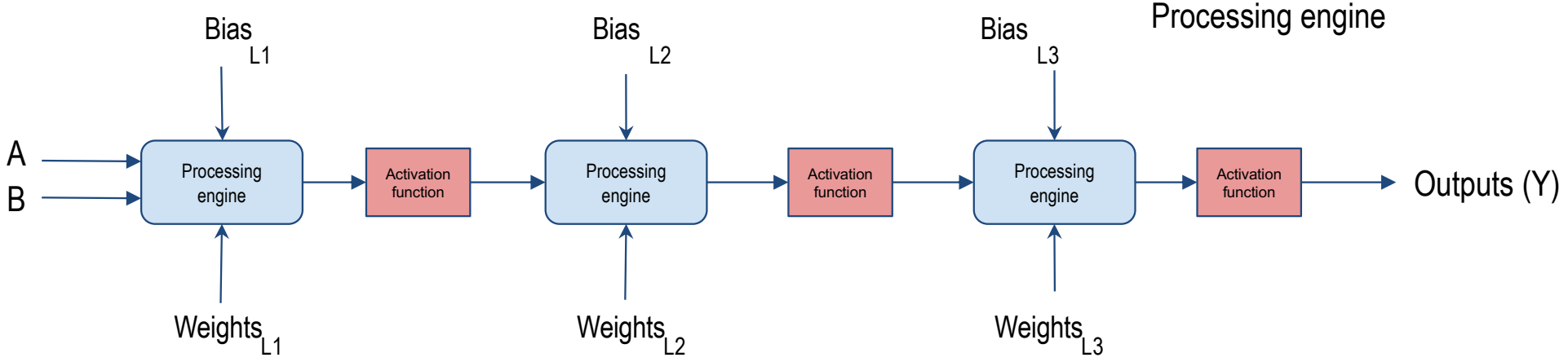


Each weight is a 32-bit floats

SystemC model in EDS for AI: Model architecture

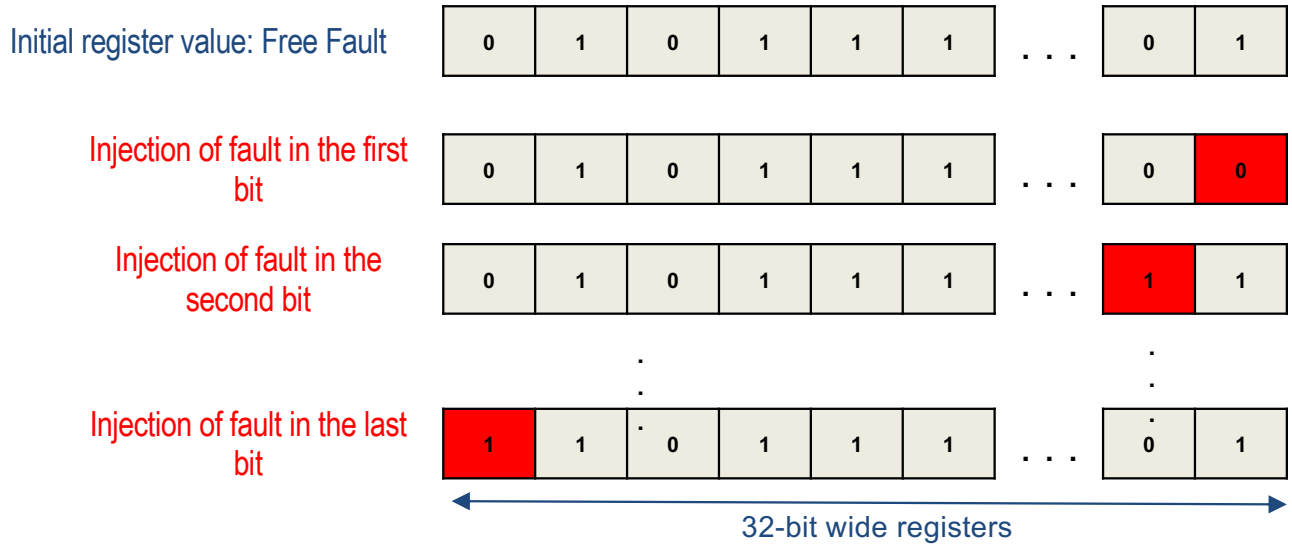
- Inference was done in SystemC.
- A hardware engine for matrix multiplication is implemented.
- Weights and biases are extracted from the trained model





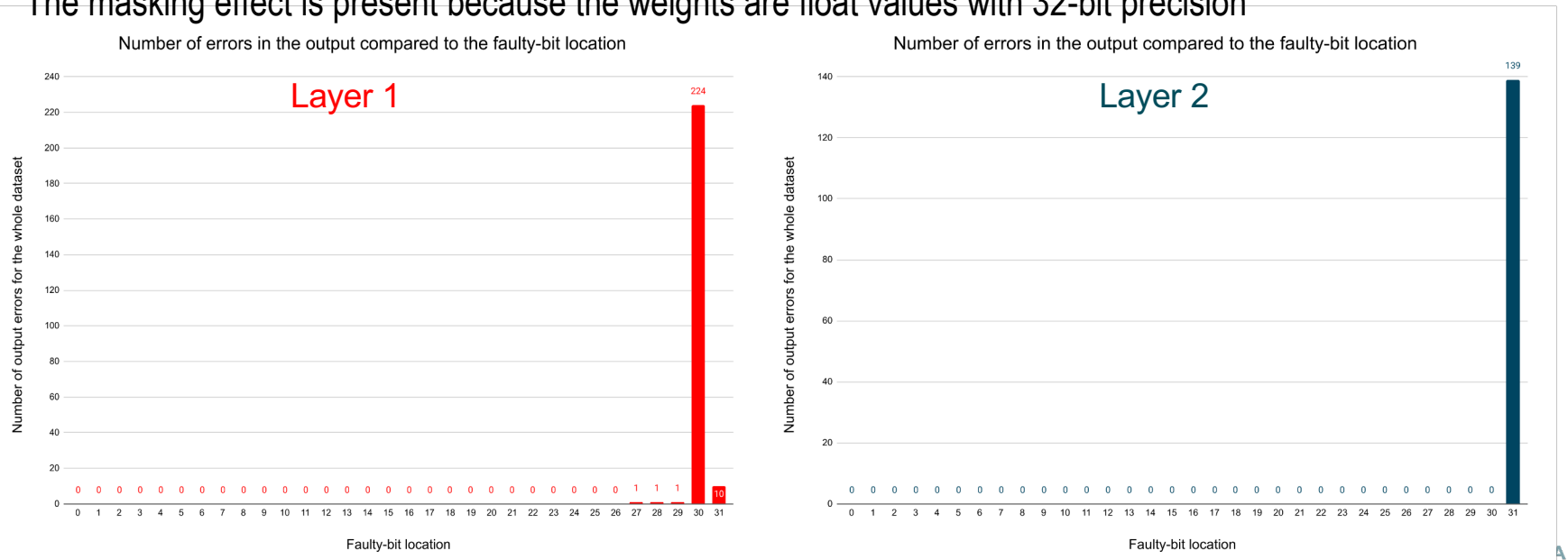
Fault injection

- Weights are represented on 32-bit floats.
- Faults were injected in random weights at different layers.
- Every time one single fault is injected.
- For one weight, a single fault (one bit-flip) is injected, going from LSB to MSB.



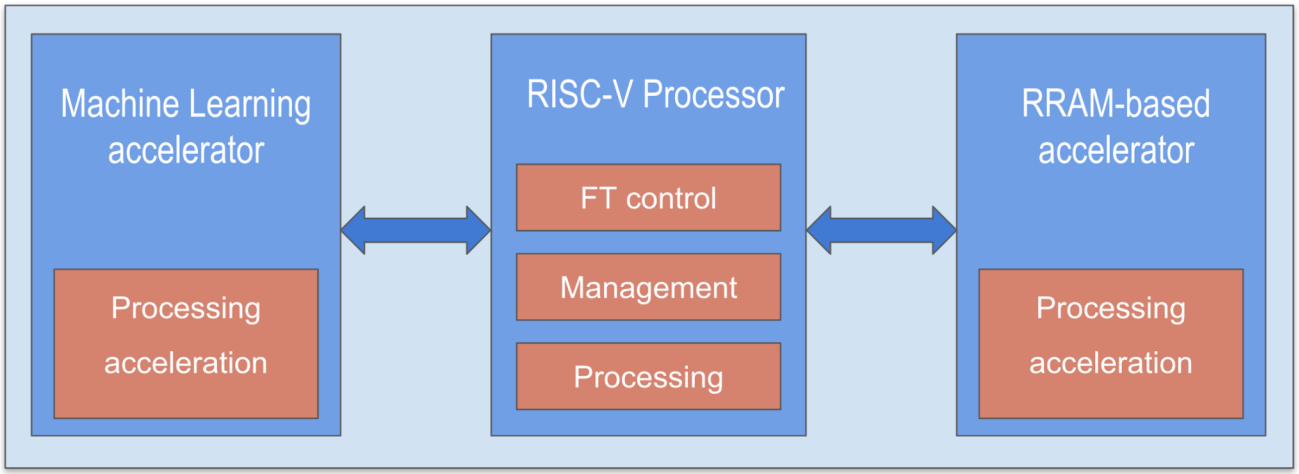
Fault injection

- Injecting faults in the LSB does not lead to errors in the output, the faults are masked.
- Faults injected in the MSB propagate and affect the behavior of the SystemC model.
- Faults injected near the output (at the last layer) are masked.
- The masking effect is present because the weights are float values with 32-bit precision



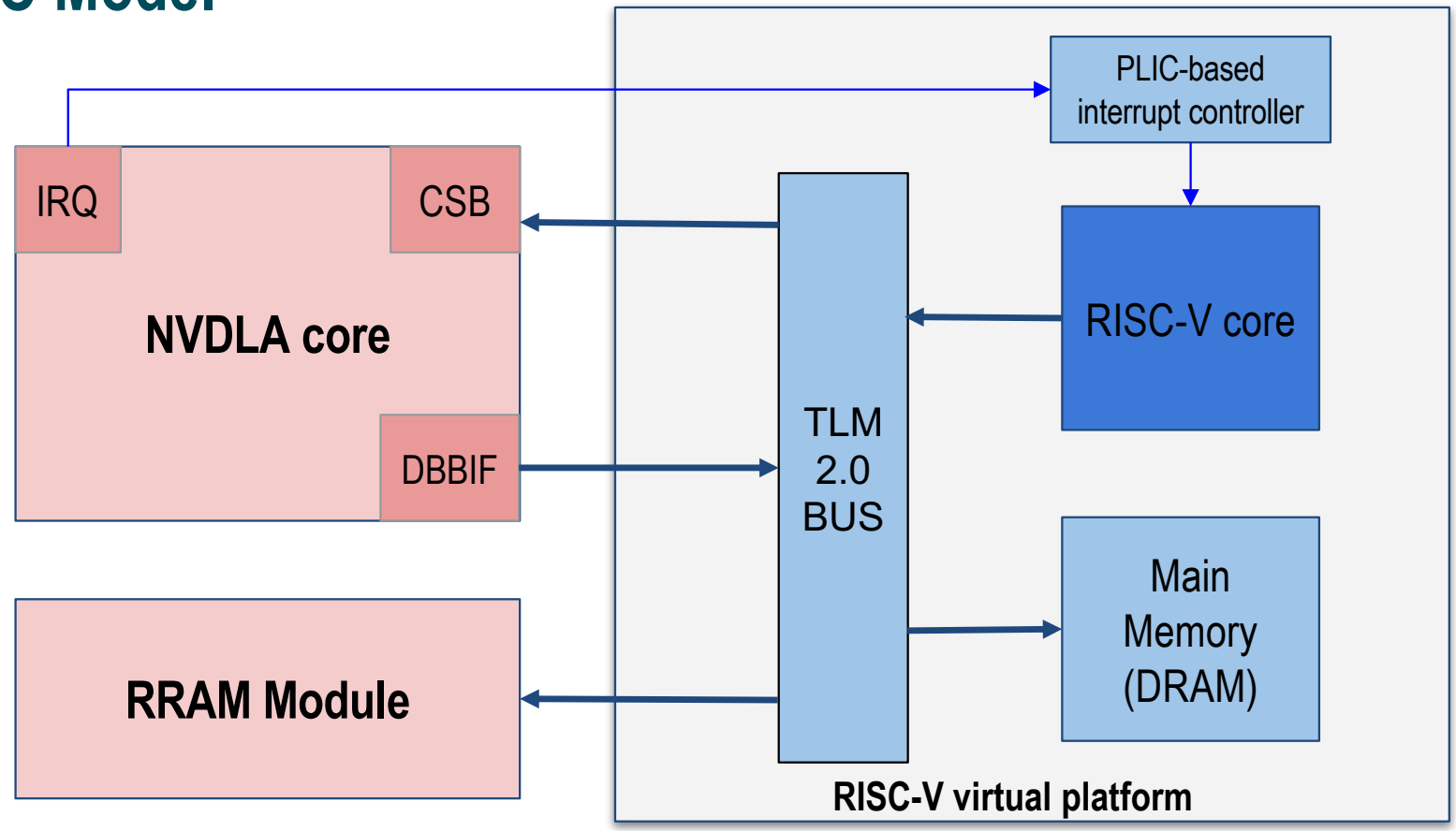
KI-Pro Platform

- RISC-V core based on the RISC-V VP.
- RRAM-based AI accelerator.
- NVIDIA Deep Learning Accelerator as a Machine Learning accelerator



KI-Pro: SystemC Model

The simulation platform uses transaction-based modelling (SystemC/TLM2.0) to achieve the adaptation and interfacing of the different components



References

[KNE 20] J. Knechtel et al., "Towards Secure Composition of Integrated Circuits and Electronic Systems: On the Role of EDA," 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE), 2020, pp. 508-513, doi: 10.23919/DATE48585.2020.9116483.

[XIA 16] Xiao, Kan, et al. "Hardware trojans: Lessons learned after one decade of research." ACM Transactions on Design Automation of Electronic Systems (TODAES) 22.1 (2016): 1-23.

[CHE 17] Chen, Wen, et al. "Challenges and trends in modern SoC design verification." IEEE Design & Test 34.5 (2017): 7-22.