Tackling Hardware Trojan Horses via Hardware-based Methodologies

Alessandro Palumbo
CentraleSupélec, Inria, CNRS, IRISA
alessandro.palumbo@inria.fr

***Abstract*—Hardware-based approaches to detect bit-flips, unauthorized program runs, and tampering.**

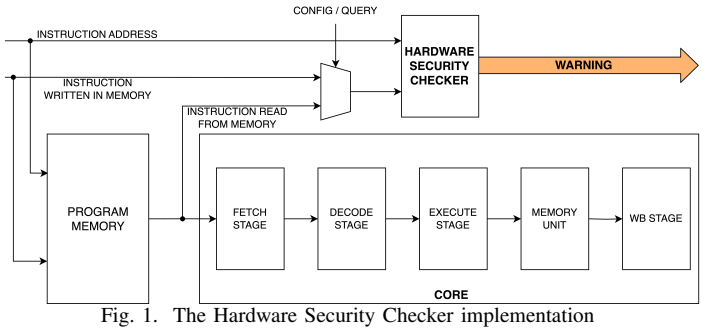

Fig. 1. The Hardware Security Checker implementation

I. The Hardware Security Checker Configuration

Configuration: Recording legitimate instruction-address pairs.

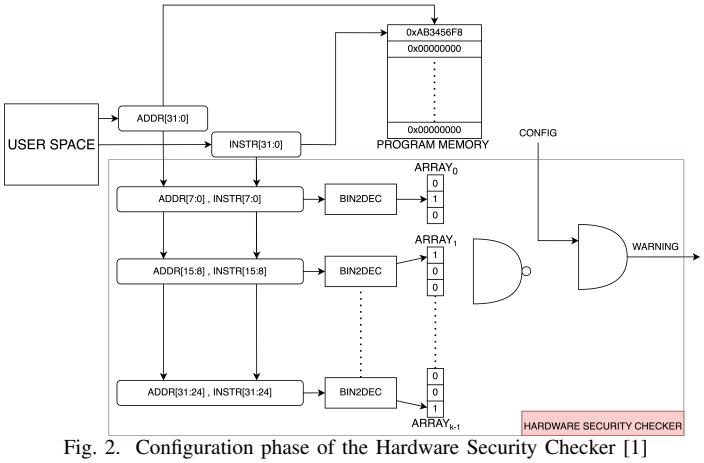

Fig. 2. Configuration phase of the Hardware Security Checker [1]

II. The Hardware Security Checker Verification

Verification: Fetched instructions runtime verification.

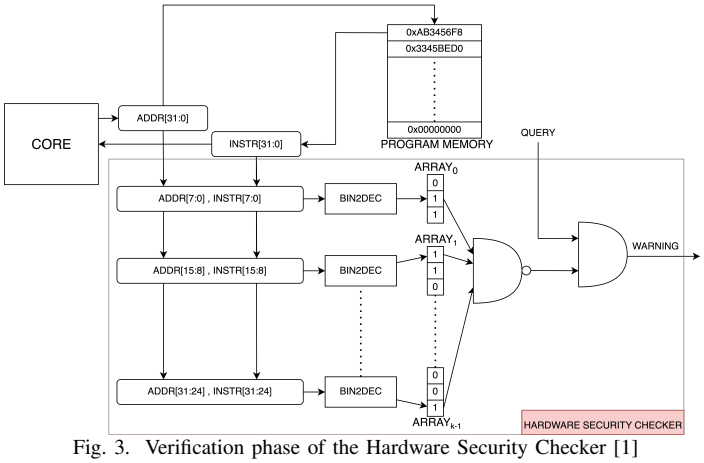

Fig. 3. Verification phase of the Hardware Security Checker [1]

TABLE I
Best anomalies detection rates and overheads

References	Acc.	FP	FN	#LUT	#FF	#BRAM
[1] Fig. 2, 3	100%	0%	0%	75 (0.49%)	31 (0.31%)	8
[2] Fig. 4	100%	0%	0%	82 (0.53%)	31 (0.31%)	8.5
[3] Fig. 5	100%	0%	0%	NA	NA	NA

III. The Hardware Security Checker & Hamming

Enhancing detection capabilities, integrating **Hamming code-based verification**.

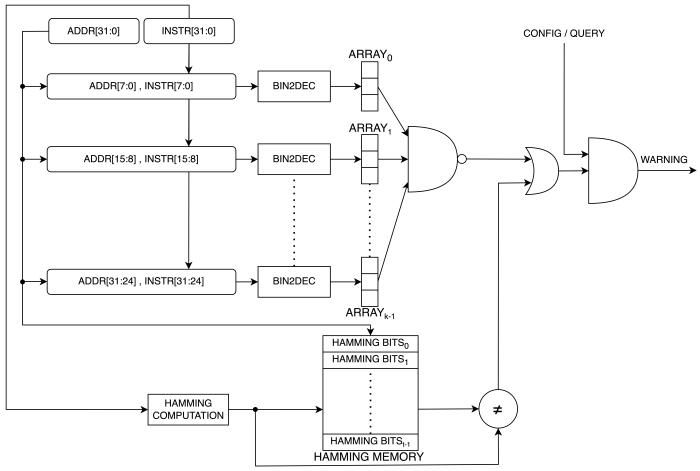

Fig. 4. Improving the Hardware Security Checker [2]

IV. Detecting FPGA Bitstream Tampering with ML

Integrating supply chain phases with **ML-based anomaly detection for FPGA security**.

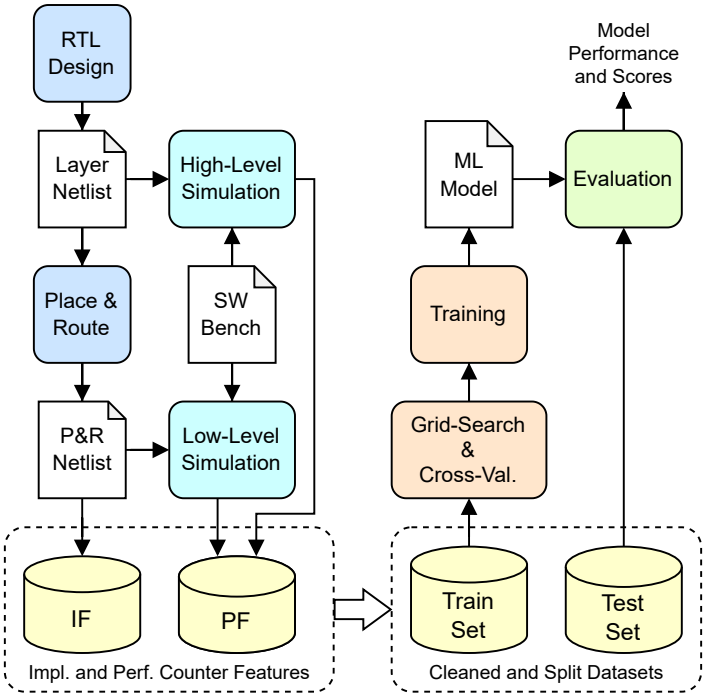

Fig. 5. The workflow for detecting FPGA tampering [3]

References

[1] A. Palumbo et al., "A lightweight security checking module to protect microprocessors against hardware trojan horses," in *DFT*, pp. 1–6, 2021.
[2] A. Palumbo et al., "Improving the detection of hardware trojan horses in microprocessors via hamming codes," in *DFT*, pp. 1–6, 2023.
[3] S. Ribes et al., "Machine learning-based classification of hardware trojans in fpgas implementing risc-v cores.," in *ICISSP*, pp. 717–724, 2024.