Trojan Attack (Time Bomb) on Computing Hardware and Machine Learning Accelerators

Professor Anirban Sengupta, *FIET (UK), FBCS, FIETE*

IEEE Distinguished Visitor, IEEE Distinguished Lecturer, ACM India Eminent Speaker

Chair, IEEE Distinguished Visitor Selection Committee

Department of Computer Science and Engineering

Indian Institute of Technology (IIT) Indore

Keynote

26-SEP-2025

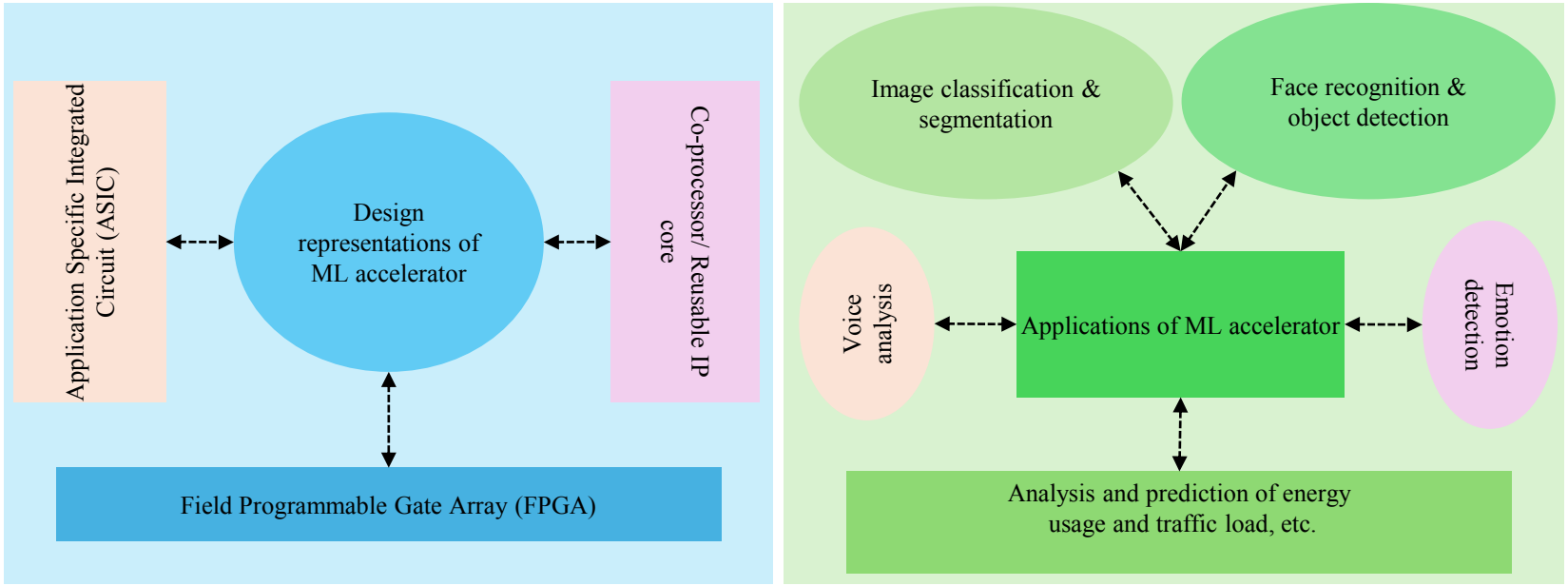
Introduction

- Key machine learning algorithms, such as linear regression (LR), convolutional neural networks (CNNs), etc., are widely used in the computing industry.
  - For example, CNN is known for their high accuracy, which is applied in face recognition, object detection, image segmentation, voice recognition, and emotion analysis [1].
  - Another example is LR, which is utilized for forecasting temperature, data estimation and prediction in smart systems. Similarly, other ML algorithms play vital roles in modern applications, including autonomous driving, smart home energy management, etc. [1], [6], [9].
- ML algorithms are essentially probabilistic models that perform extensive computations on input data for tasks like classification/prediction.
  - For instance, a CNN consists of layers such as convolutional, pooling, flattening, and fully connected layers, with the convolutional layer being highly computation intensive.
  - This necessitates the use of dedicated coprocessors/ accelerators for data-centric applications.
  - Similarly, the LR algorithm handles large datasets during training, requiring substantial computational resources.
- Dedicated computing platforms, such as coprocessors/ hardware accelerators, field-programmable gate arrays (FPGAs), and application- specified integrated circuits (ASICs), efficiently manage these computational demands. Dedicated accelerators can be designed to meet specific power and performance constraints.
- These ML accelerators can be designed by using the high-level synthesis (HLS) framework [3]. HLS accepts the behavioral description of the ML algorithms and produces its corresponding register transfer-level (RTL) data path.

[1] C. Jiang, D. Ojika, B. Patel and H. Lam, "Optimized FPGA-based Deep Learning Accelerator for Sparse CNN using High Bandwidth Memory," *IEEE 29th Annual International Symposium on Field-Programmable Custom Computing Machines*, USA, 2021, pp. 157-164.
[3]Why you Need HLS for Machine Learning Accelerators, accessed in 2024, Available: https://resources.sw.siemens.com/en-US/video-why-you-need-hls-for-machine-learning-accelerators.
[6] A. Sengupta, R. Chaurasia, M. Rathor: HLS-based swarm intelligence driven optimized hardware IP core for linear regression-based machine learning, *IET Journal of Engineering*, e12299 (2023).
[8] S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network," *International Conference on Engineering and Technology*, Turkey, 2017, pp. 1-6.
[9] A. Sengupta and R. Chaurasia, "Secured Convolutional Layer IP Core in Convolutional Neural Network Using Facial Biometric," *IEEE Transactions on Consumer Electronics*, vol.68, no. 3, pp. 291-306, 2022.

ML/Hardware Accelerators

- ML/hardware accelerators enhance data-intensive tasks like image recognition, natural language processing, and autonomous driving, enabling faster data processing and improved efficiency in real-life applications.
  - Some real-life examples of ML accelerators are NVIDIA Deep Learning Accelerator (NVDLA) [4], LR hardware accelerator [6], convolutional layer hardware accelerator [9], etc.



(a) (b)

Fig. 1. (a) Different design representations for machine learning (ML) accelerators, and (b) Different applications of ML accelerators

[4] N. Gupta, A. Jati and A. Chattopadhyay, AI Attacks AI: Recovering Neural Network architecture from NVDLA using AI-assisted Side Channel Attack, *Cryptology {ePrint} Archive*, Paper 2023/368, 2023, url = https://eprint.iacr.org/2023/368.

[6] A. Sengupta, R. Chaurasia, M. Rathor: HLS-based swarm intelligence driven optimized hardware IP core for linear regression-based machine learning, *IET Journal of Engineering*, e12299 (2023).

[9] A. Sengupta and R. Chaurasia, "Secured Convolutional Layer IP Core in Convolutional Neural Network Using Facial Biometric," *IEEE Transactions on Consumer Electronics*, vol.68, no. 3, pp. 291-306, 2022.

Security Vulnerability in ML/Hardware Accelerators

➢ **How do hackers (attackers) exploit security vulnerability in ML/Hardware accelerators?**

- ML accelerators or in general hardware accelerators may be designed using HLS framework/RTL designing [3], [6], [9]. In various steps of HLS/RTL design, attackers (within the design house) can compromise and exploit a computer-aided design (CAD) software tool and/or RTL design to covertly inject backdoor Trojans.
- As shown in Fig. 1. (c), a hardware Trojan attack on a crypto-accelerator has capability to bypass the encryption circuit and leak confidential information. On rare even triggering, the encryption is bypassed easily.
- Moreover, security vulnerability of NVIDIA accelerators (NVDLA) has been exposed in [4], as shown in Fig. 1. (d). Power and side channel leakage information from CNN models have been used to train CNN based attack models.

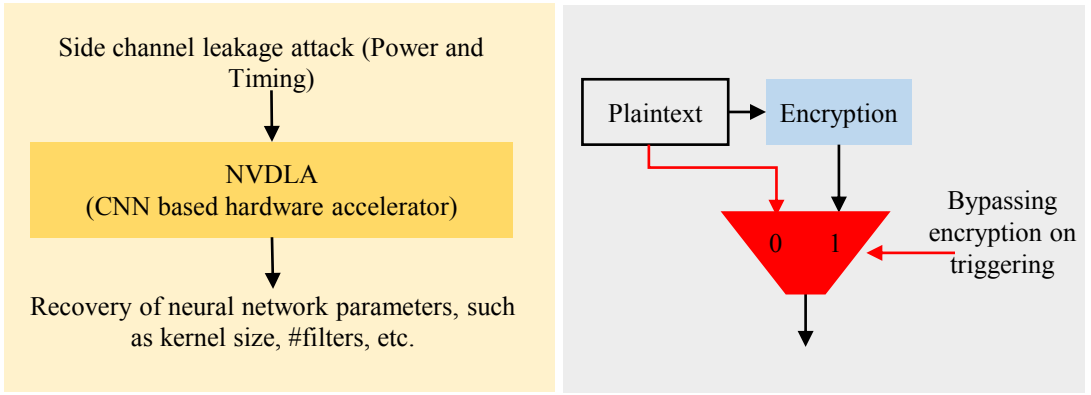


Fig. 1. (c) Example of Trojan attack and security vulnerability in a cryptographic accelerator, and (d) Example of Trojan attack and security vulnerability of a real-world application (NVDLA accelerator)

[3]Why you Need HLS for Machine Learning Accelerators, accessed in 2024, Available: https://resources.sw.siemens.com/en-US/video-why-you-need-hls-for-machine-learning-accelerators.
[4] N. Gupta, A. Jati and A. Chattopadhyay, AI Attacks AI: Recovering Neural Network architecture from NVDLA using AI-assisted Side Channel Attack, *Cryptology {ePrint} Archive*, Paper 2023/368, 2023, url = https://eprint.iacr.org/2023/368.
[6] A. Sengupta, R. Chaurasia, M. Rathor: HLS-based swarm intelligence driven optimized hardware IP core for linear regression-based machine learning, *IET Journal of Engineering*, e12299 (2023).
[9] A. Sengupta and R. Chaurasia, "Secured Convolutional Layer IP Core in Convolutional Neural Network Using Facial Biometric," *IEEE Transactions on Consumer Electronics*, vol.68, no. 3, pp. 291-306, 2022.

Security Vulnerability in ML/Hardware Accelerators

- **How do hackers (attackers) exploit security vulnerability in ML/Hardware accelerators?**
  - In another real-life scenario, an attacker can accelerate the aging process of a computing device, such as digital signal processing (DSP) accelerator, by exploiting negative bias temperature instability (NBTI) stress as hardware Trojan.
  - By applying NBTI stress based Trojan attack, an attacker puts stress on PMOS transistors by increasing their threshold voltage. This causes them to degrade in terms of performance delay and can expedite the aging related performance degradation. This has been established in [5].
  - Furthermore, an attacker can inject a trojan during scheduling phase, allocation phase and max interconnect design phase. For example, a hacker can also secretly insert Trojan (pseudo/fake) operations during the scheduling phase of the HLS design process (resulting in a battery exhaustion attack) [17].
  - Further, a hacker can also exploit the Mux-based interconnect design stage during HLS to secretly insert Trojans (such as denial-of-service hardware Trojan (DoS HT), performance degradation hardware Trojan (PD-HT), data damage hardware Trojan (DD-HT)) into the ML accelerators (adopted from [7]).

  The goal of the attacker while launching such Trojan attacks is to maliciously exploit any unused free port or under-utilized resources to inject Trojan logic.

[5] D. Kachave and A. Sengupta, "Digital Processing Core Performance Degradation Due to Hardware Stress Attacks," *IEEE Potentials*, vol. 38, no. 2, pp. 39-45, March-April 2019.
[7] A. Sengupta, A. Anshul, V. Chourasia and N. Kumar, "M-HLS: Malevolent High-Level Synthesis for Watermarked Hardware IPs," *IEEE Embedded Systems Letters*, 2024, doi: 10.1109/LES.2024.3416422.
[17] C. Pilato, K. Basu, F. Regazzoni and R. Karri, "Black-Hat High-Level Synthesis: Myth or Reality?," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 4, pp. 913-926, 2019.

Security Vulnerability in ML/Hardware Accelerators

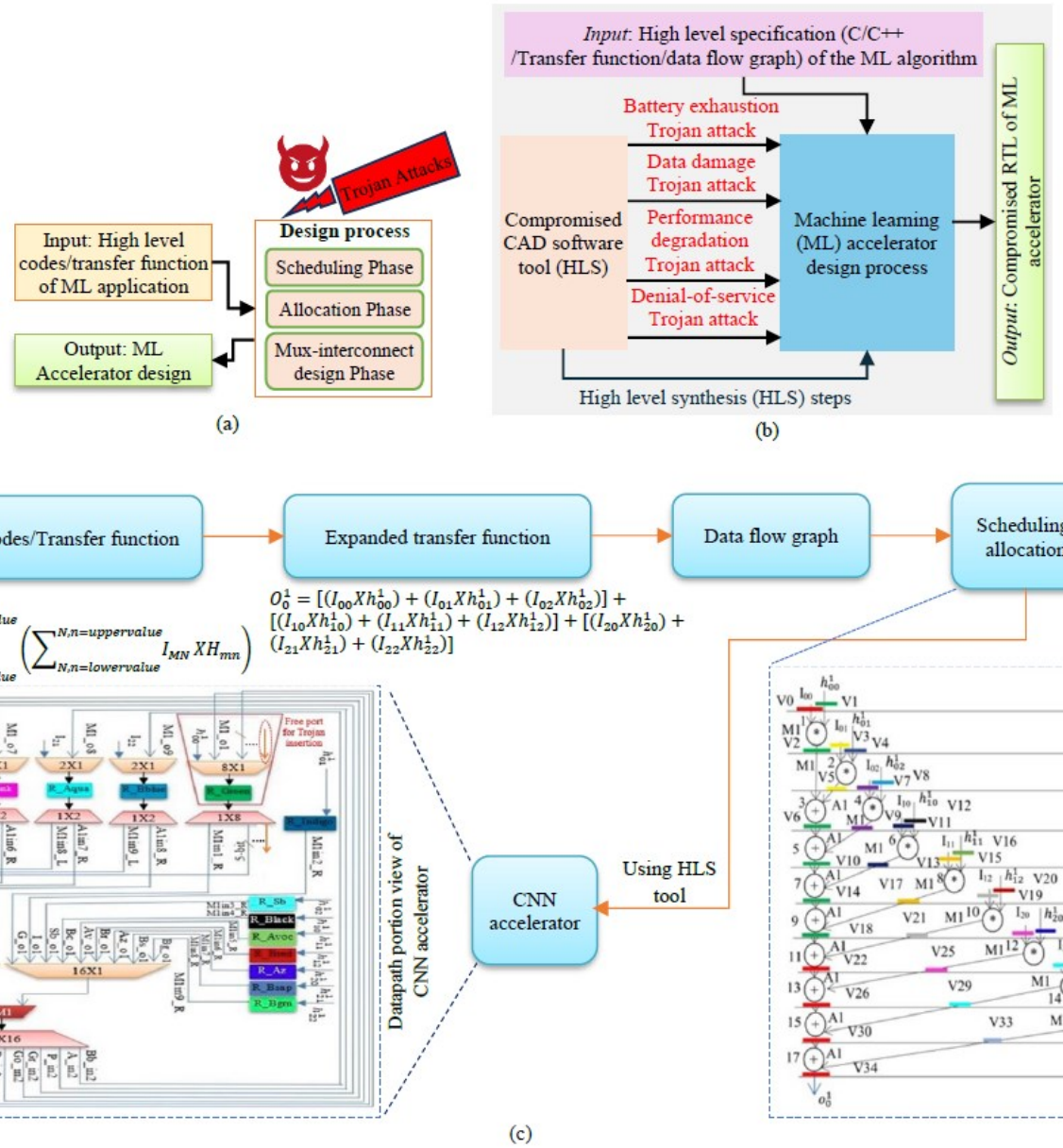


Fig. 2. (a) Overview of Trojan attack during ML accelerator design process, (b) Established Trojan attacks on ML accelerator, and (c) Design flow of ML (CNN) co-processor/accelerator (adopted from [9]).

*Note: '$\boldsymbol{I_{MN}}$' and '$\boldsymbol{H_{mn}}$' in the transfer function represents the input image of size **MxN** and kernal of size **mxn** respectively. $\boldsymbol{O_y}$ denotes the output value of each element/pixel corresponding to output feature map; further in the expanded transfer function, each pixel value of the input image matrix and each kernel value of kernel matrix 't' is represented by $\boldsymbol{I_{ab}}$ and $\boldsymbol{h_{pq}^t}$ respectively*

[9] A. Sengupta and R. Chaurasia, "Secured Convolutional Layer IP Core in Convolutional Neural Network Using Facial Biometric," *IEEE Transactions on Consumer Electronics*, vol.68, no. 3, pp. 291-306, 2022.

Security Vulnerability in ML/Hardware Accelerators

- As shown in Fig. 2.(c), initially, the high-level code/transfer function of the ML application is taken as input. For example, the CNN convolution layer's transfer function is shown in Fig. 2.(c).
- Further, the expanded transfer function is generated. Next, the corresponding data flow graph (DFG/CDFG) is generated [9]. Subsequently, the DFG is fed as input to the HLS scheduling and allocation block. Finally, ML accelerator RTL datapath is generated post datapath synthesis.
- Fig. 2.(c) also depicts the datapath portion view of the CNN convolutional layer accelerator [9]. As evident in the datapath portion view, some input ports are free (unutilized) in the Mux-based interconnect design of the shown datapath (ports shown in orange).
- It has been established in the literature [7], that these unused free ports can be exploited by the attacker during compromising a CAD HLS tool (to secretly insert the Trojan), without the knowledge of the ML accelerator designer (who is using the tool), causing different payloads (such as denial-of-service hardware Trojan (DoS HT), performance degradation hardware Trojan (PD-HT), data damage hardware Trojan (DD-HT)).
- Additionally, it has also been established in the literature [17], that an attacker can also exploit the scheduling phase of the HLS framework to insert pseudo/fake operations to launch a battery exhaustion attack.
- The various Trojan payloads [2] can cause different types of adversarial effects in ML accelerators.

[2] Xue, M., Gu, C., Liu, W., Yu, S. and O'Neill, M. (2020), Ten years of hardware Trojans: a survey from the attacker's perspective. *IET Comput. Digit. Tech*., 14: 231-246.
[7] A. Sengupta, A. Anshul, V. Chourasia and N. Kumar, "M-HLS: Malevolent High-Level Synthesis for Watermarked Hardware IPs," *IEEE Embedded Systems Letters*, 2024, doi: 10.1109/LES.2024.3416422.
[9] A. Sengupta and R. Chaurasia, "Secured Convolutional Layer IP Core in Convolutional Neural Network Using Facial Biometric," *IEEE Transactions on Consumer Electronics*, vol.68, no. 3, pp. 291-306, 2022.
[17] C. Pilato, K. Basu, F. Regazzoni and R. Karri, "Black-Hat High-Level Synthesis: Myth or Reality?," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 4, pp. 913-926, 2019.

Types of Trojan Attacks

- The four established Trojan attacks on ML accelerators are as follows:
  - Performance degradation hardware Trojan (PD-HT),
  - Data damage hardware Trojan (DD-HT),
  - Denial-of-service hardware Trojan (DoS-HT), and
  - Battery exhaustion Trojan (BE-HT).

Note: *Here, the orange-colored components indicate Trojan logic inserted by the attacker and orange free port input on top of 8x1 multiplexer indicates unutilized port exploited by the hacker for secret Trojan insertion.*

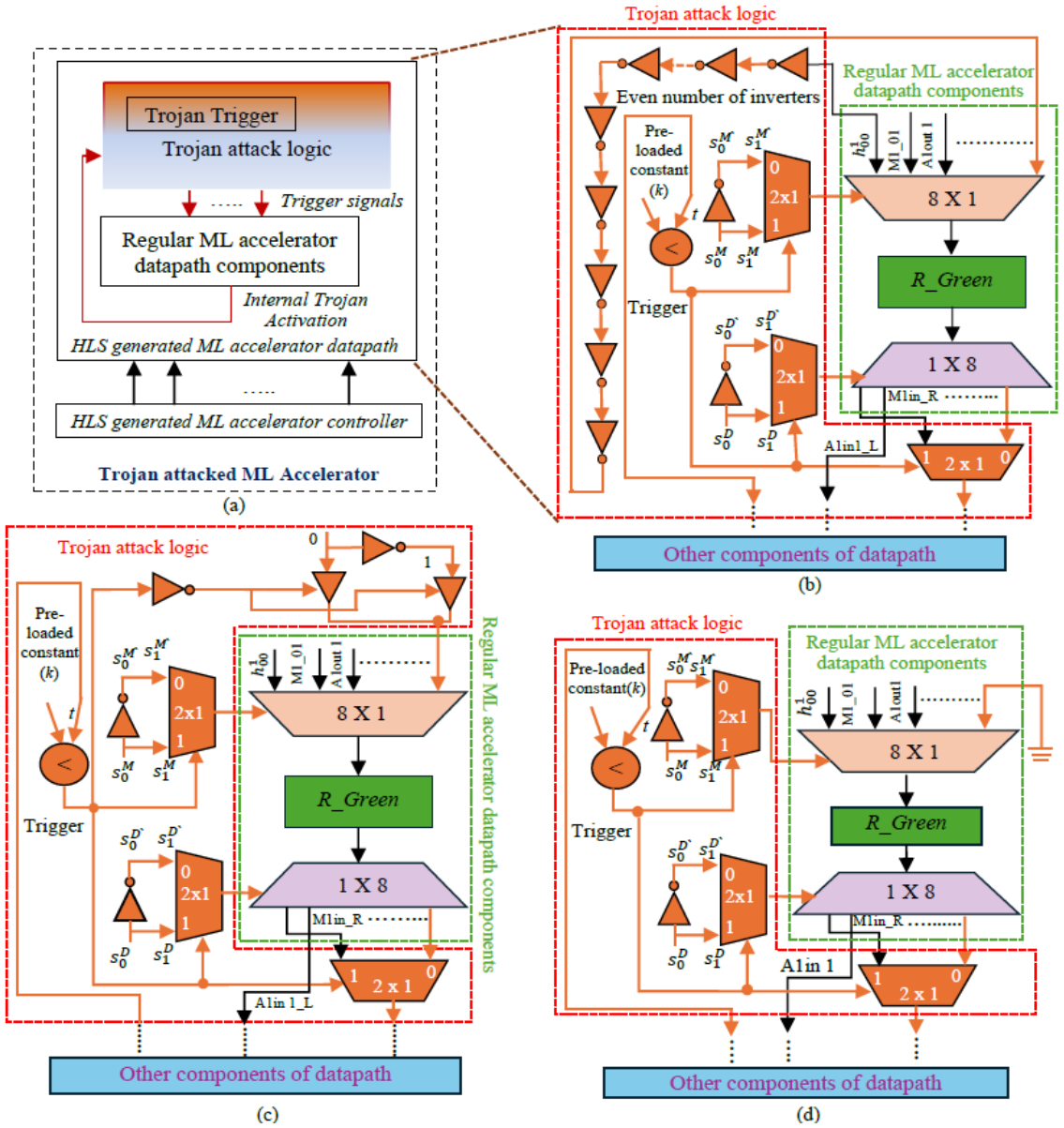
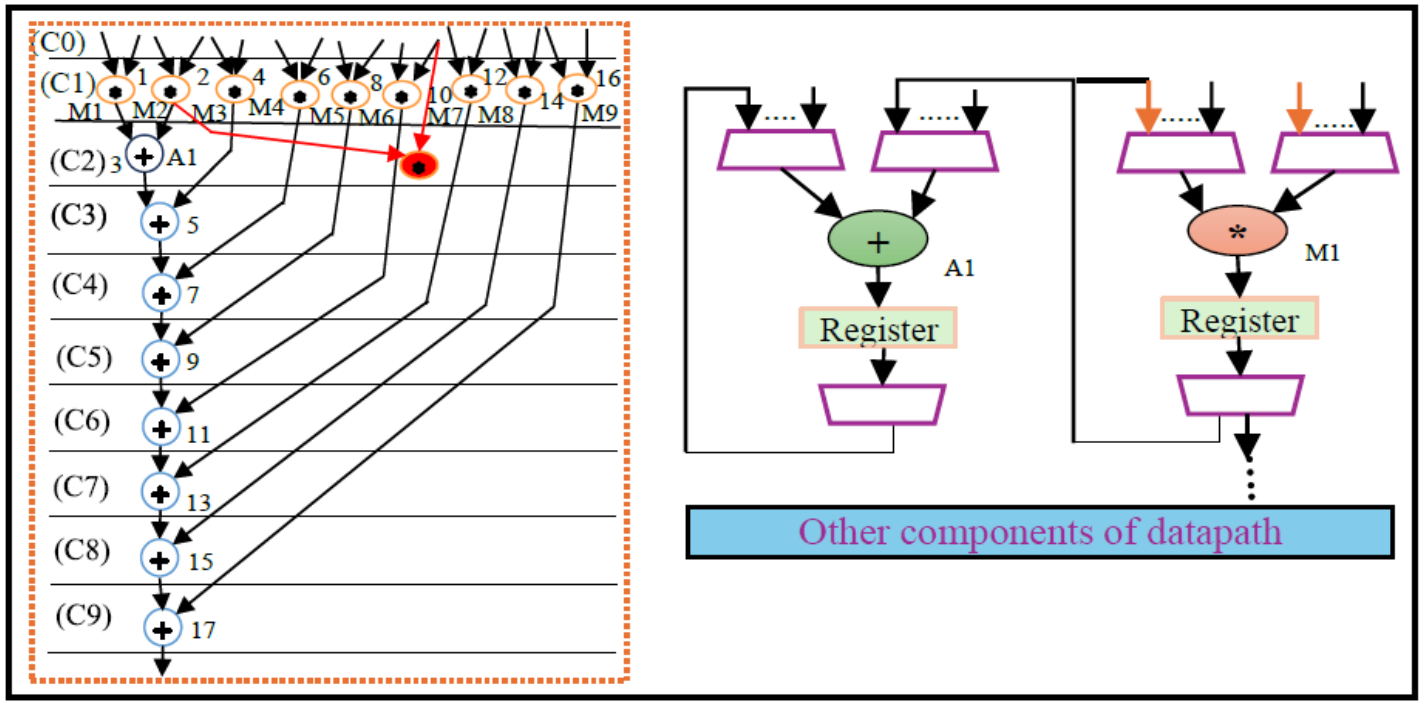


Fig. 3. (a) Trojan attacked ML Accelerator design, (b) Performance degradation Trojan attack, (c) Denial-of-service Trojan attack, and (d) Data Damage trojan attack

Types of Trojan Attacks (Contd.)

- Fig. 3.(e) demonstrates the integration of a BE-HT in the ML accelerator, designed to increase power consumption and speed up battery depletion.
  - The primary goal is to reuse the idle functional units (FUs) in the ML accelerator datapath. Multipliers, which have larger power dissipation, are chosen to increase the overall power consumption of the accelerator. Such modifications in the datapath have nominal area and power overhead.
  - This technique does not affect the final computational output while concurrently not enhancing the power overhead of the design substantially.



(e)

Fig. 3. (e) Battery exhaustion Trojan attack in CNN convolutional layer accelerator

Triggering of Backdoor Trojan

- Hardware Trojans are stealthily inserted into systems that become active only when a specific rare condition (predetermined by the attacker) is met.
- This activation has been managed through comparator logic that switches on the Trojan's payload when the rare condition is satisfied.
- These Trojans are extremely difficult to detect because they stay inactive (dormant and undetected until triggered by specific conditions) during regular system operations. During the insertion of the Trojan, an attacker programs a constant value 'k' into memory (electrically programmable) [7].
- As shown in Figures 3. (b), (c), and (d), the first input (t) of the comparator is connected internally to the functional units (such as adders, multipliers, etc.) of remaining ML accelerator datapath, and the second input is connected to memory holding pre-loaded constant 'k'. Once the system's state (t) matches this constant (k), the Trojan becomes triggered, causing it to execute its intended malicious effects.
- The above-explained trigger condition is the same for PD-HT, DD-HT, and DoS-HT. However, BE-HT has been designed in the literature to become triggered after a certain count value of the counter [17].
- These Trojans not only compromise the security of ML designs but also erode the trust between the ML accelerator vendors and CAD software communities [5].

[5] D. Kachave and A. Sengupta, "Digital Processing Core Performance Degradation Due to Hardware Stress Attacks," *IEEE Potentials*, vol. 38, no. 2, pp. 39-45, March-April 2019.
[7] A. Sengupta, A. Anshul, V. Chourasia and N. Kumar, "M-HLS: Malevolent High-Level Synthesis for Watermarked Hardware IPs," *IEEE Embedded Systems Letters*, 2024, doi: 10.1109/LES.2024.3416422.
[17] C. Pilato, K. Basu, F. Regazzoni and R. Karri, "Black-Hat High-Level Synthesis: Myth or Reality?," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 4, pp. 913-926, 2019.

Time Bomb Trojan Attack

- Fig. 4(a) shows the insertion stage of the proposed time-bomb triggered Trojan during the HLS design process.
- For explanation and demonstration of the proposed HLS Trojan, we use an HLS generated convolution filter IP design (Fig. 4(b)).

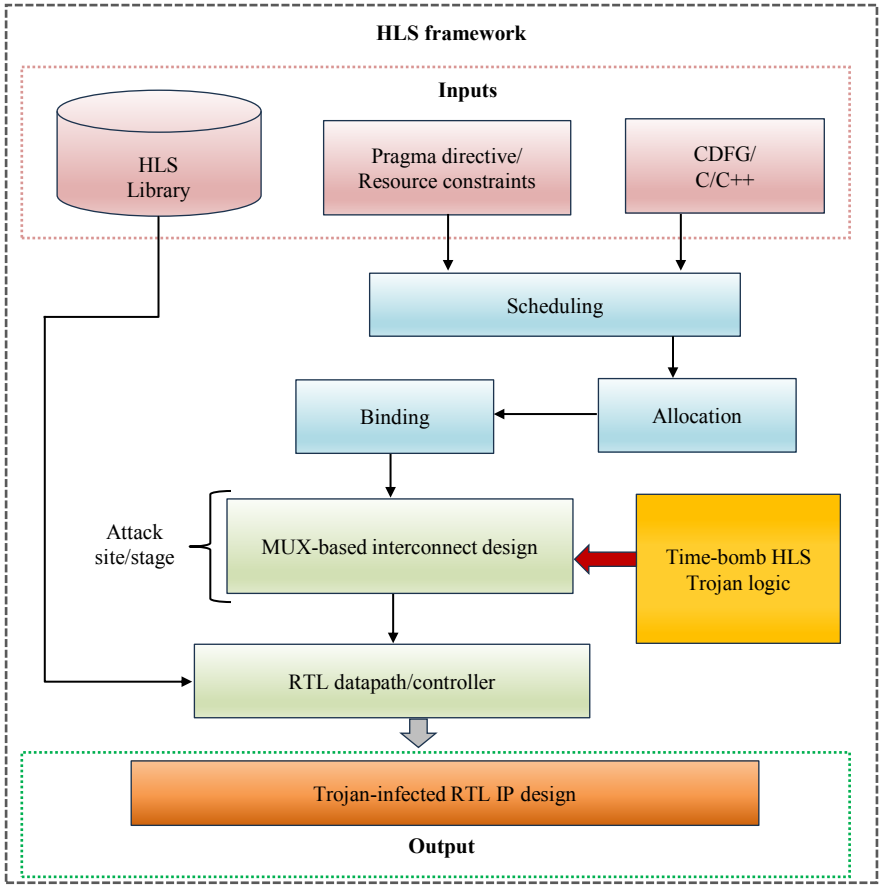


Fig. 4(a). Proposed Trojan insertion in the HLS design flow

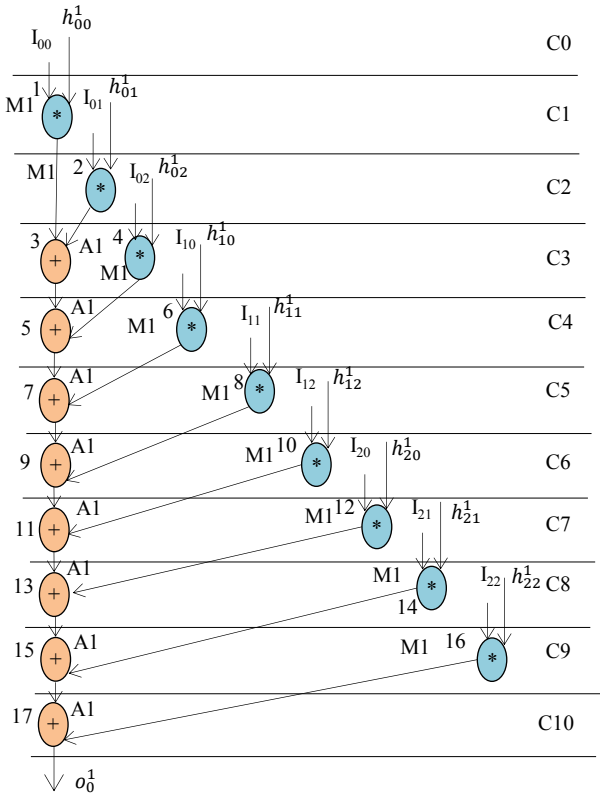


Fig. 4(b). Scheduled DFG of proposed convolutional layer IP core with kernel of size 3x3 based on 1M, 1A resources

Time Bomb Trojan Attack

➢ **Overview of the Proposed Trojan Insertion**

- The work presents novel time-bomb triggering driven performance degradation hardware Trojan (PD-HT) that an attacker can secretly implant by exploiting a free (vacant) input port in the mux-based interconnect design of HLS process.
- During the mux-based interconnect design of HLS, appropriate number and type of multiplexer (mux) units are determined and generated.
- In almost all IP datapath designs, the generated muxes have at least a single free (vacant) port, which can be easily exploited by an attacker to covertly inject Trojan.
- The proposed PD-HT refers to a malicious alteration within the IP design that achieves performance degradation payload under a specific rare-event time-duration based triggering condition.
- The proposed HLS Trojan exploits a time-bomb based trigger which indicates that the Trojan logic only gets activated (by an attacker) when a pre-defined time interval has elapsed.
- The proposed time-bomb Trojan trigger is designed in such a way that the activation only occurs when the modulus up-counter reaches the same state value as pre-defined in the in-built memory (or register).
- Since the proposed HLS Trojan is only activated under a specific rare-event and it only affects the performance, hence it is very challenging to identify this Trojan.

- Fig. 4(a) shows the insertion stage of the proposed time-bomb triggered Trojan during the HLS design process.

Time Bomb Trojan Attack

➢ Details of the Proposed Trojan Insertion

- Fig. 4(c) shows the proposed performance (delay) degradation hardware Trojan inserted into the convolutional filter IP datapath design, during the mux-based interconnect design stage of HLS process.
- In Fig. 4(c), the red colored components or logic indicates the proposed performance (delay) degradation hardware Trojan logic, while the other components or logic are part of the regular IP datapath design of convolutional filter.
- More is the Tri-State buffer (TSB) chain length, greater is the performance degradation payload achieved by an attacker. The non-red color components are the regular units of convolutional filter IP datapath design.
- The behavioral table of the proposed time-bomb HLS Trojan with its trigger possibilities, triggering conditions and relevant outputs is described in the next slide.

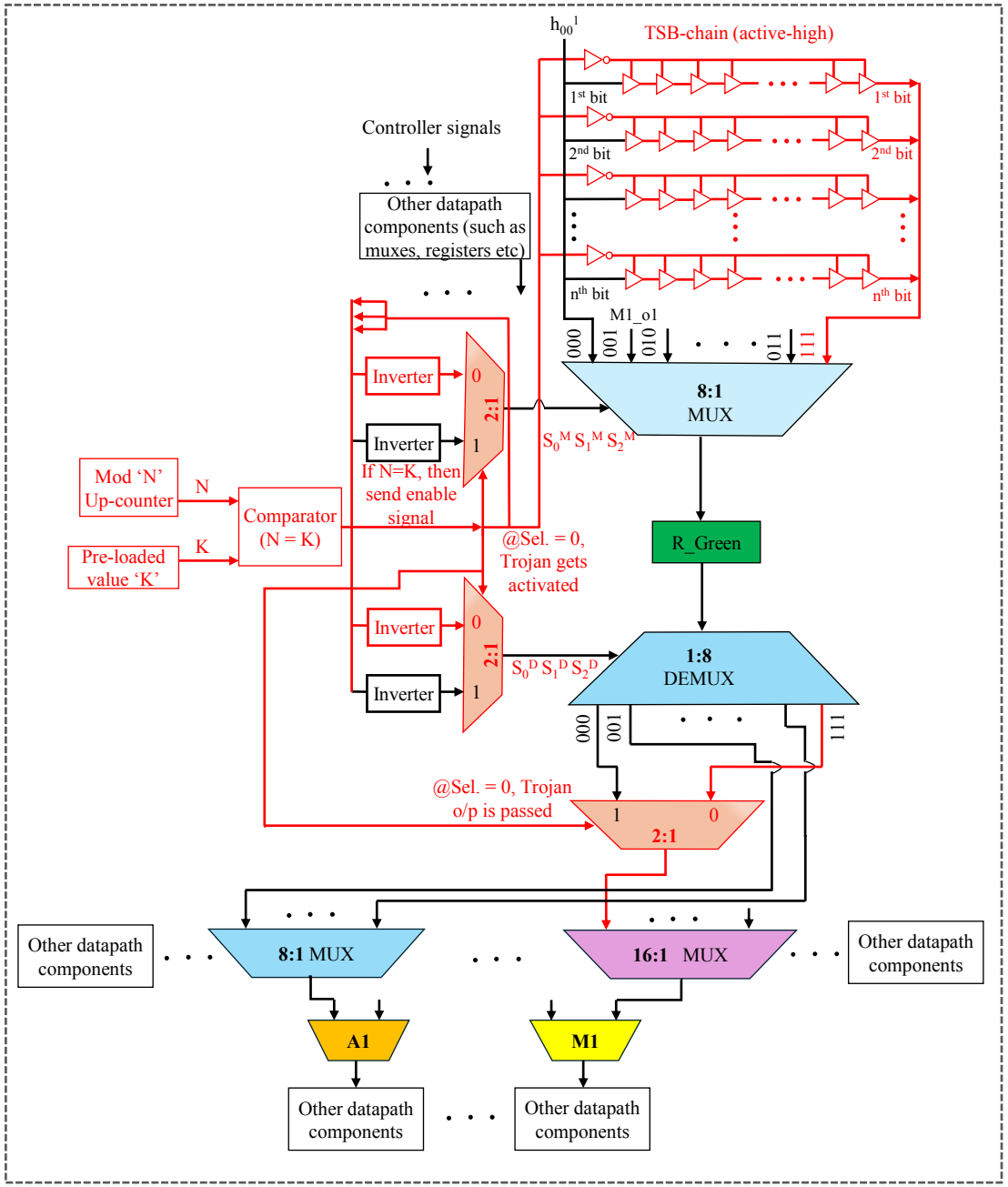


Fig. 4(c). HLS based time-bomb Trojan inserted convolution filter IP datapath design – partial view

Detection Techniques Employed for Backdoor Trojans in ML Accelerators

S. No.	Different detection techniques	Performance degradation Trojan attack	Data damage Trojan attack	Denial-of-service Trojan attack	Battery (Power) exhaustion Trojan attack
1	C to RTL Equivalence checking [10]	×	✓	×	✓
2	TL-HLS (DMR based security-aware scheduling) [11]	×	×	×	×
3	Side channel analysis [12]	×	×	×	×
4	Detection using reverse engineering [13]	×	×	×	×
5	Detection using path delay fingerprint [14]	×	×	×	×
6	GNN based detection [15]	×	×	×	×
7	HLT based detection [16]	×	×	×	✓

(a)

Fig. 5. (a) Analysis of different detection techniques on Trojan infected ML accelerator designs (Note: '×' indicates "not detectable")

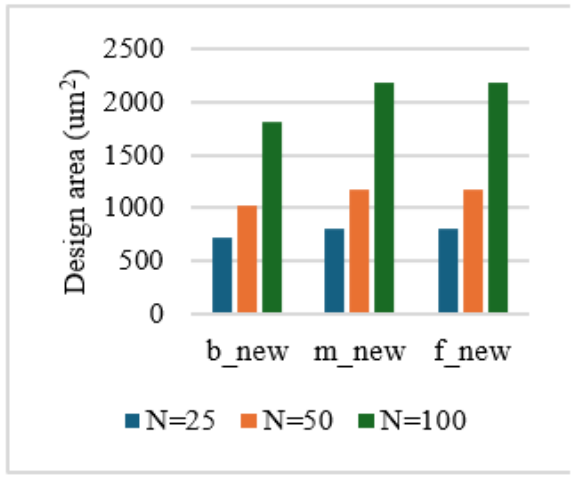
[10] M. Abderehman, R. Gupta, R. R. Theegala and C. Karfa, "BLAST: Belling the Black-Hat High-Level Synthesis Tool," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 11, pp. 3661-3672, 2022.
[11] A. Sengupta, S. Bhadauria and S. P. Mohanty, "TL-HLS: Methodology for Low Cost Hardware Trojan Security Aware Scheduling With Optimal Loop Unrolling Factor During High Level Synthesis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 36, no. 4, pp. 655-668, 2017.
[12] Y. Huang, S. Bhunia and P. Mishra, "Scalable Test Generation for Trojan Detection Using Side Channel Analysis," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, pp. 2746-2760, 2018.
[13] M. Ludwig, A. -C. Bette and B. Lippmann, "ViTaL: Verifying Trojan-Free Physical Layouts through Hardware Reverse Engineering," IEEE Physical Assurance and Inspection of Electronics, USA, 2021, pp. 1-8.
[14] Y. Jin and Y. Makris, "Hardware trojan detection using path delay fingerprint," *IEEE International Workshop on HOST*, 2008, pp. 51–57.
[15] R. Yasaei, L. Chen, S. -Y. Yu and M. A. A. Faruque, "Hardware Trojan Detection using Graph Neural Networks," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2022.
[16] M. Rathor and A. Sengupta, "Revisiting Black-Hat HLS: A Lightweight Countermeasure to HLS-Aided Trojan Attack," *IEEE Embedded Systems Letters*, Volume: 16, Issue: 2, 2024, pp. 170-173.

Detection Techniques Employed for Backdoor Trojans in ML Accelerators (Contd.)

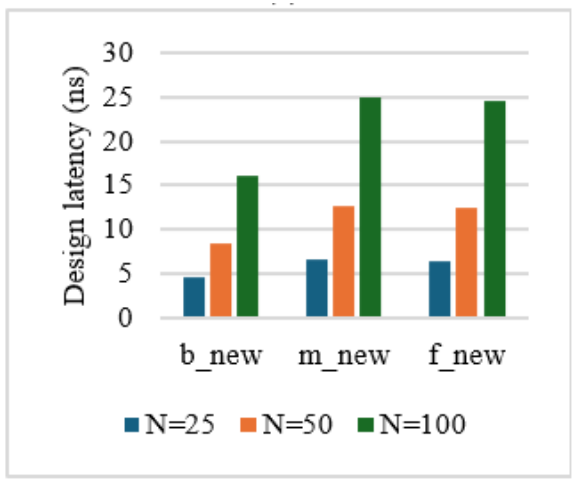
- Different types of Trojans, like PD-HT and DoS-HT, impacting performance and operational state, without altering functionality, makes them difficult to detect using equivalence checking [7], [10].
- On the other hand, DD-HT affects data output under rare condition triggering, making detection somehow possible through equivalence analysis [10].
- Further, BE-HT has been successfully detected using C to RTL equivalence checking based on finite state machine datapath (FSMD) extraction. Further, these Trojans remain undetected through side-channel analysis [12] as they don't leak significant parametric information (such as delay and power).
- Techniques like path delay fingerprinting [14], attempt to differentiate normal designs from those compromised by Trojans, however, they become impractical for complex HLS-generated ML accelerator.
- Detection tools relying on Graph Neural Networks (GNN) [15] face limitations in accurately detecting Trojans within ML accelerators, because its performance/accuracy for complex ML accelerators is lower due to weaker learning behavior.
- Moreover, the detection technique [16] is only capable of handling BE-HT attacks, as PD-HT, DD-HT, and DoS-HT do not induce Trojan payload using fake operation insertion.
- Therefore, based on the published detection techniques for Trojans, C to RTL functional equivalence checking [10] has been the most effective technique as it is capable of detecting both BE-HT and DD-HT.

[10] M. Abderehman, R. Gupta, R. R. Theegala and C. Karfa, "BLAST: Belling the Black-Hat High-Level Synthesis Tool," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 11, pp. 3661-3672, 2022.

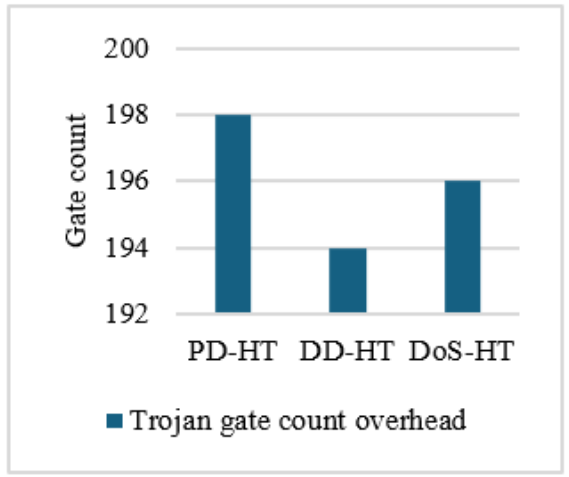
Analysis in terms of Design Area, Latency, and Resources



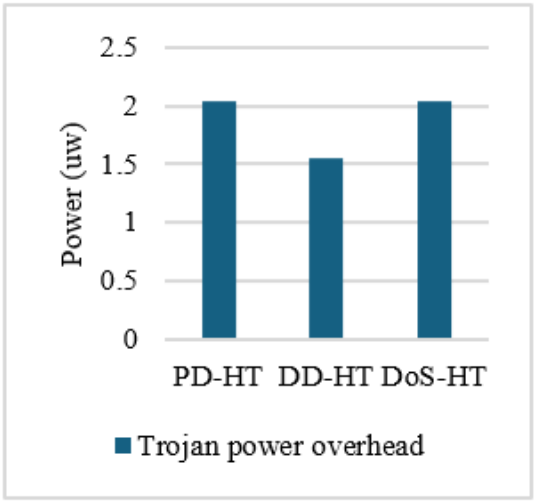
(b)



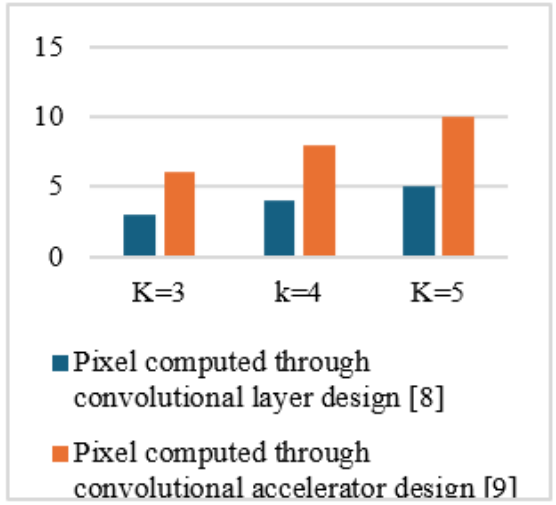
(c)



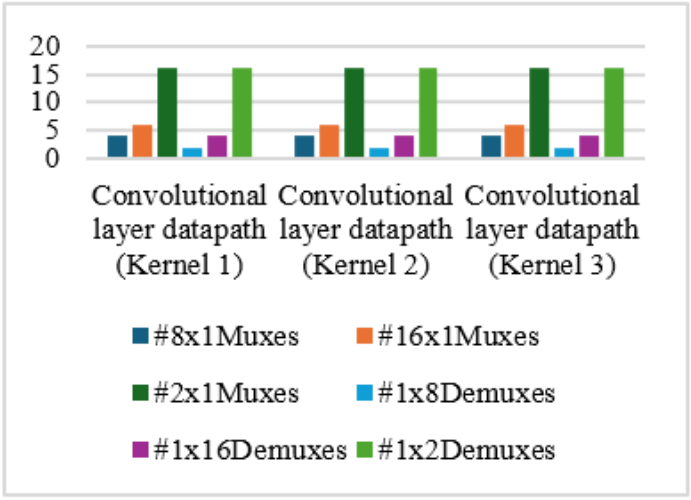
(d)



(e)



(f)



(g)

Fig. 5. (b) Design area for LR-ML accelerator corresponding to different numbers of datasets (N) [6], (c) Design latency for LR-ML accelerator corresponding to different numbers of datasets (N) [6], (d) Trojan design area overhead (in terms of gate count) corresponding to convolutional layer CNN accelerator [7], (e) Trojan power overhead corresponding to convolutional layer CNN accelerator [7], (f) Comparison of number of pixels computed between [8] and [9] for different convolutional kernel filters (K), and (g) Resources required for convolutional layer datapath w.r.t. three different kernels

Analysis of ML Accelerators in terms of Design Area, Latency, and Resources (Contd.)

- This section presents the analysis of different ML accelerator designs from the literature. Figures 4.(b) and (c) depict the design area and latency for the LR-ML accelerator corresponding to different numbers of datasets (N), respectively [6].
- The design area and latency are directly proportional to the number of datasets it handles. Subsequently, figures 4. (d) and (e) show the design area (in terms of gate count) and power overhead corresponding to convolutional layer CNN accelerator after Trojan injection, respectively [7].
- The Trojan-infected design, on average, incurs a minimal increase in the ~196 gate count value and ~1.6 μw power as compared to the baseline ML-accelerator design [7].
- Next, Fig. 4. (f) shows the comparison of pixel computation between [8] and [9] for different convolutional kernel filters (K).
- Approach [9] surpasses [8] in terms of pixel computation value due to parallel pixel computation process owing to loop unrolled architecture. Finally, Fig. 4 (g) highlights required resources for convolutional layer accelerator datapath w.r.t. three different kernels [9].

[6] A. Sengupta, R. Chaurasia, M. Rathor: HLS-based swarm intelligence driven optimized hardware IP core for linear regression-based machine learning, *IET Journal of Engineering*, e12299 (2023).
[7] A. Sengupta, A. Anshul, V. Chourasia and N. Kumar, "M-HLS: Malevolent High-Level Synthesis for Watermarked Hardware IPs," *IEEE Embedded Systems Letters*, 2024, doi: 10.1109/LES.2024.3416422.
[8] S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network," *International Conference on Engineering and Technology*, Turkey, 2017, pp. 1-6.
[9] A. Sengupta and R. Chaurasia, "Secured Convolutional Layer IP Core in Convolutional Neural Network Using Facial Biometric," *IEEE Transactions on Consumer Electronics*, vol.68, no. 3, pp. 291-306, 2022.

References

[1] C. Jiang, D. Ojika, B. Patel and H. Lam, "Optimized FPGA-based Deep Learning Accelerator for Sparse CNN using High Bandwidth Memory," *IEEE 29th Annual International Symposium on Field-Programmable Custom Computing Machines*, USA, 2021, pp. 157-164.

[2] Xue, M., Gu, C., Liu, W., Yu, S. and O'Neill, M. (2020), Ten years of hardware Trojans: a survey from the attacker's perspective. *IET Comput. Digit. Tech*., 14: 231-246.

[3]Why you Need HLS for Machine Learning Accelerators, accessed in 2024, Available: https://resources.sw.siemens.com/en-US/video-why-you-need-hls-for-machine-learning-accelerators.

[4] N. Gupta, A. Jati and A. Chattopadhyay, AI Attacks AI: Recovering Neural Network architecture from NVDLA using AI-assisted Side Channel Attack, *Cryptology {ePrint} Archive*, Paper 2023/368, 2023, url = https://eprint.iacr.org/2023/368.

[5] D. Kachave and A. Sengupta, "Digital Processing Core Performance Degradation Due to Hardware Stress Attacks," *IEEE Potentials*, vol. 38, no. 2, pp. 39-45, March-April 2019.

[6] A. Sengupta, R. Chaurasia, M. Rathor: HLS-based swarm intelligence driven optimized hardware IP core for linear regression-based machine learning, *IET Journal of Engineering*, e12299 (2023).

[7] A. Sengupta, A. Anshul, V. Chourasia and N. Kumar, "M-HLS: Malevolent High-Level Synthesis for Watermarked Hardware IPs," *IEEE Embedded Systems Letters*, 2024, doi: 10.1109/LES.2024.3416422.

[8] S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network," International Conference on Engineering and Technology, Turkey, 2017, pp. 1-6.

[9] A. Sengupta and R. Chaurasia, "Secured Convolutional Layer IP Core in Convolutional Neural Network Using Facial Biometric," *IEEE Transactions on Consumer Electronics*, vol.68, no. 3, pp. 291-306, 2022.

[10] M. Abderehman, R. Gupta, R. R. Theegala and C. Karfa, "BLAST: Belling the Black-Hat High-Level Synthesis Tool," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 11, pp. 3661-3672, 2022.

[11] A. Sengupta, S. Bhadauria and S. P. Mohanty, "TL-HLS: Methodology for Low Cost Hardware Trojan Security Aware Scheduling With Optimal Loop Unrolling Factor During High Level Synthesis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 36, no. 4, pp. 655-668, 2017.

[12] Y. Huang, S. Bhunia and P. Mishra, "Scalable Test Generation for Trojan Detection Using Side Channel Analysis," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, pp. 2746-2760, 2018.

[13] M. Ludwig, A. -C. Bette and B. Lippmann, "ViTaL: Verifying Trojan-Free Physical Layouts through Hardware Reverse Engineering," *IEEE Physical Assurance and Inspection of Electronics*, USA, 2021, pp. 1-8.

[14] Y. Jin and Y. Makris, "Hardware trojan detection using path delay fingerprint," *IEEE International Workshop on HOST*, 2008, pp. 51–57.

[15] R. Yasaei, L. Chen, S. -Y. Yu and M. A. A. Faruque, "Hardware Trojan Detection using Graph Neural Networks," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2022.

[16] M. Rathor and A. Sengupta, "Revisiting Black-Hat HLS: A Lightweight Countermeasure to HLS-Aided Trojan Attack," *IEEE Embedded Systems Letters,* Volume: 16, Issue: 2, 2024, pp. 170-173.

[17] C. Pilato, K. Basu, F. Regazzoni and R. Karri, "Black-Hat High-Level Synthesis: Myth or Reality*?," IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 4, pp. 913-926, 2019.

[18] K. I. Gubbi et al., "Securing AI hardware: Challenges in detecting and mitigating hardware trojans in ML accelerators," *Proc. IEEE 66th Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Tempe, AZ, USA, pp. 821–825, 2023.

Thank You !!!