



HLS Register Allocation Attack for Compromising Hardware IP Designs using HLS-aided Trojan

Anirban Sengupta, Vishal Chourasia, Vipin Verma "HLS Register Allocation Attack for Compromising Hardware IP Designs using HLS-aided Trojan", **IEEE Design & Test (D&T)**, doi: 10.1109/MDAT.2026.3715573, 2026

Indian Institute of Technology Indore

Introduction

- Modern electronics and computing systems rely heavily on complex System-on-Chips (SoCs) for their operation.
- SoCs integrate a wide range of reusable Intellectual Property (IP) cores.
- The IP designs used in SoCs of complex applications are typically designed using HLS [1], [2], [10], [5]. .
- HLS design framework depends on third-party HLS tools, which are considered untrustworthy due to potential security risks.
- HLS frameworks expose several attack points that adversaries can exploit to covertly insert backdoor hardware Trojans [1], [2], [10], [5].
- Trojan Payloads compromising functional integrity, denial of service, data damage, leakage of secret key, performance degradation etc. [15], [11].
- This paper presents a novel HLS-aided Trojan for register allocation tampering (RAT) attack, without affecting the functionality of the IP design.
- It introduces a new variant of HLS Trojan called '*RAT-HT*' that can be secretly inserted by an HLS tool developer during datapath design stage.

The Proposed HLS Trojan Framework: RAT-HT

Threat Model and Motivation

- HLS frameworks are prone to security vulnerabilities across multiple phases of the design process [1]-[3].
- These vulnerabilities in the HLS flow are exploited through the covert insertion of hardware Trojans.
- First, HLS tools can be unreliable or compromised, potentially containing hidden backdoor Trojans.
- Second, malicious logic may be stealthily embedded within various HLS design phases, remaining inactive during regular operations but triggering under specific rare event condition.
- Part of international toolkit: Due to globalization in design supply chain, an HLS-tool developed in one country may be employed in another nation.
- Since RTL IP designs generated through HLS are often significantly larger and more complex than their corresponding high-level specifications, detecting such HLS Trojan logic becomes an extremely difficult task.

Overview

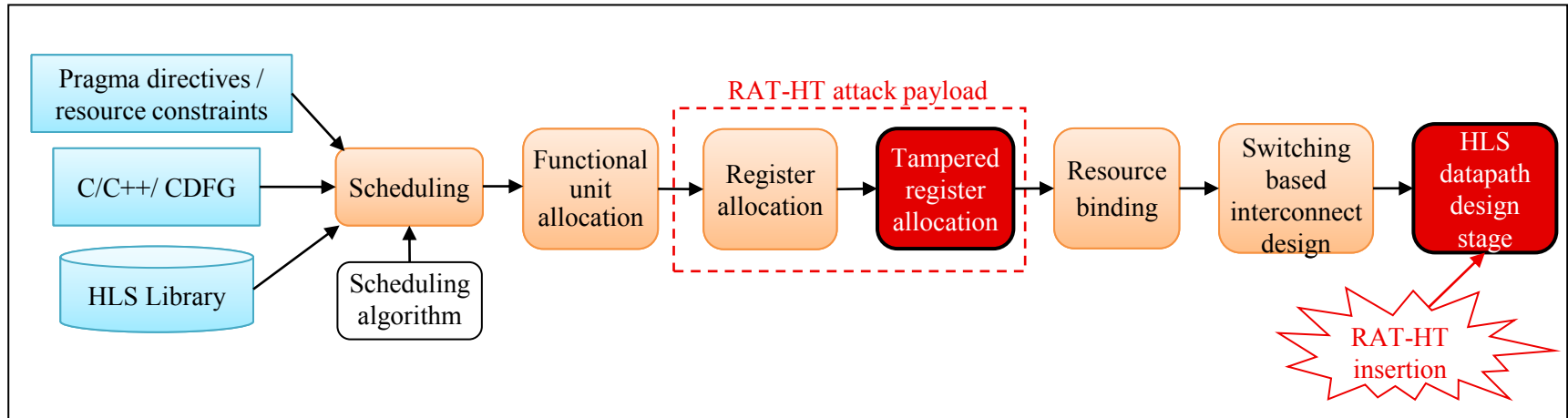
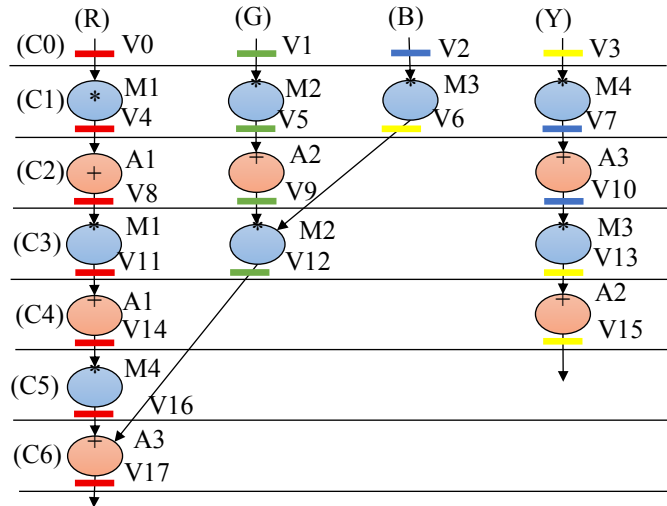


Fig.1. Proposed RAT-HT insertion in HLS Flow

- RAT-HT in the HLS datapath implements swapping of storage variables to different register while causing excess power exhaustion.
- The Trojan is designed to trigger under a rare, time-sensitive condition, using timer-based activation logic.
- It employs an integrated logic comprising of modulus-based up-counter, stored pre-defined value in register and comparator to implement the rare event triggering condition.

Proposed Methodology

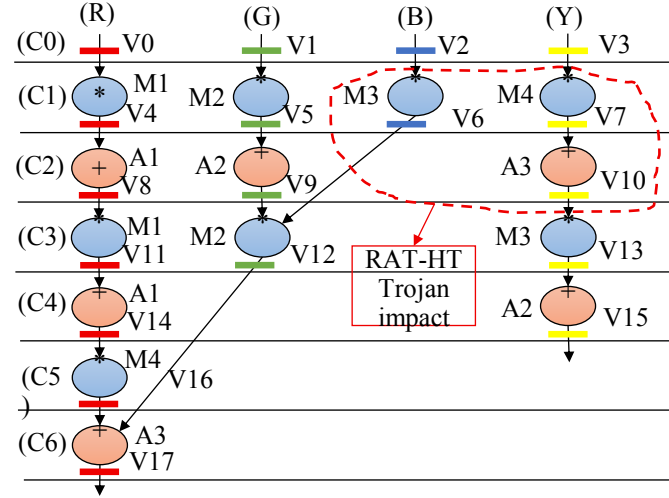


REGISTER ALLOCATION FOR MESA BEFORE TROJAN INSERTION

Control steps	Red	Green	Blue	Yellow
0	V0	V1	V2	V3
1	V4	V5	V7	V6
2	V8	V9	V10	V6
3	V11	V12	-	V13
4	V14	V12	-	V15
5	V16	V12	-	V15
6	V17	-	-	V15

Red	Green	Blue	Yellow
8:1 Mux 1:8 Dmux	4:1 Mux 1:4 Dmux	4:1 Mux 1:4 Dmux	4:1 Mux 1:4 Dmux

Fig.2(a). MESA Schedule and allocated graph (R: red, G: green, B: blue, Y: yellow)



TAMPERED REGISTER ALLOCATION FOR MESA DUE TO RAT-HT INSERTION

Control steps	Red	Green	Blue	Yellow
0	V0	V1	V2	V3
1	V4	V5	V6	V7
2	V8	V9	V6	V10
3	V11	V12	-	V13
4	V14	V12	-	V15
5	V16	V12	-	V15
6	V17	-	-	V15

Red	Green	Blue	Yellow
8:1 Mux 1:8 Dmux	4:1 Mux 1:4 Dmux	2:1 Mux 1:2 Dmux	8:1 Mux 1:8 Dmux

Fig.2(b). RAT-HT Trojan affected MESA Schedule and allocated graph (R: red, G: green, B: blue, Y: yellow).

Proposed Methodology

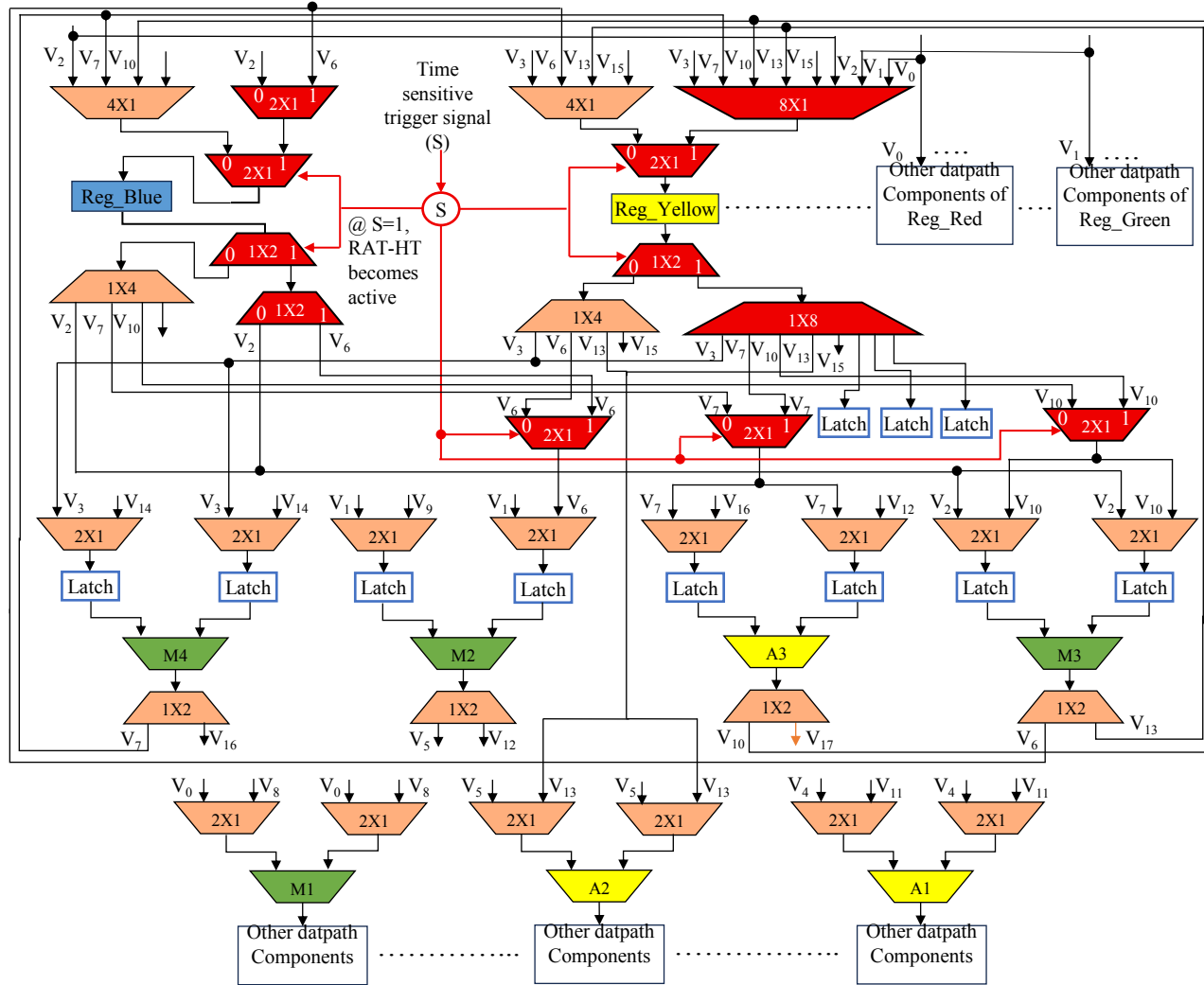


Fig.2(c). Proposed RAT-HT infected MESA Datapath

Results and Analysis

TABLE 1A

POWER DRAINAGE ACHIEVED BY THE PROPOSED RAT-HT INFECTED IP DATAPATH FOR DIFFERENT BENCHMARKS

Benchmarks [2]	Trojan Free Datapath IP		Trojan induced datapath IP (post-activation)		Power exhaustion achieved by the proposed framework	Average power exhaustion achieved by the proposed framework
	Datapath Components	Power (μ W)	Datapath Components	Power (μ W)		
MESA	Register $\rightarrow$ 4 MUX 2:1 $\rightarrow$ 14 DEMUX 1:2 $\rightarrow$ 7 MUX 4:1 $\rightarrow$ 2 DEMUX 1:4 $\rightarrow$ 2 Adder (A) $\rightarrow$ 3 Multiplier (M) $\rightarrow$ 4	30.13	Register $\rightarrow$ 4 MUX 2:1 $\rightarrow$ 20 DEMUX 1:2 $\rightarrow$ 10 MUX 4:1 $\rightarrow$ 2 DEMUX 1:4 $\rightarrow$ 2 MUX 8:1 $\rightarrow$ 1 DEMUX 1:8 $\rightarrow$ 1 Adder (A) $\rightarrow$ 3 Multiplier (M) $\rightarrow$ 4	41.93	39.16%	36.70 %
Sharpening Filter	Register $\rightarrow$ 22 MUX 2:1 $\rightarrow$ 13 DEMUX 1:2 $\rightarrow$ 11 MUX 4:1 $\rightarrow$ 5 DEMUX 1:4 $\rightarrow$ 3 MUX 8:1 $\rightarrow$ 6 DEMUX 1:8 $\rightarrow$ 4 Adder (A) $\rightarrow$ 4 Multiplier (M) $\rightarrow$ 2	90.77	Register $\rightarrow$ 22 MUX 2:1 $\rightarrow$ 21 DEMUX 1:2 $\rightarrow$ 16 MUX 4:1 $\rightarrow$ 7 DEMUX 1:4 $\rightarrow$ 5 MUX 8:1 $\rightarrow$ 7 DEMUX 1:8 $\rightarrow$ 5 Adder (A) $\rightarrow$ 4 Multiplier (M) $\rightarrow$ 2	110.97	22.25%	
LED Filter	Register $\rightarrow$ 14 MUX 2:1 $\rightarrow$ 8 DEMUX 1:2 $\rightarrow$ 6 MUX 4:1 $\rightarrow$ 8 DEMUX 1:4 $\rightarrow$ 6 Adder (A) $\rightarrow$ 2 Multiplier (M) $\rightarrow$ 2	49.39	Register $\rightarrow$ 14 MUX 2:1 $\rightarrow$ 21 DEMUX 1:2 $\rightarrow$ 14 MUX 4:1 $\rightarrow$ 8 DEMUX 1:4 $\rightarrow$ 6 MUX 8:1 $\rightarrow$ 2 DEMUX 1:8 $\rightarrow$ 2 Adder (A) $\rightarrow$ 2 Multiplier (M) $\rightarrow$ 2	74.66	51.15%	

Results and Analysis

TABLE 1A

POWER DRAINAGE ACHIEVED BY THE PROPOSED RAT-HT INFECTED IP DATAPATH FOR DIFFERENT BENCHMARKS

Benchmarks [2]	Trojan Free Datapath IP		Trojan induced datapath IP (post-activation)		Power exhaustion achieved by the proposed framework	Average power exhaustion achieved by the proposed framework
	Datapath Components	Power (μ W)	Datapath Components	Power (μ W)		
FIR Filter	Register $\rightarrow$ 21 MUX 2:1 $\rightarrow$ 3 DEMUX 1:2 $\rightarrow$ 3 MUX 4:1 $\rightarrow$ 3 DEMUX 1:4 $\rightarrow$ 3 MUX 8:1 $\rightarrow$ 4 DEMUX 1:8 $\rightarrow$ 2 MUX 16:1 $\rightarrow$ 3 DEMUX 1:16 $\rightarrow$ 2 Adder (A) $\rightarrow$ 1 Multiplier (M) $\rightarrow$ 2	99.66	Register $\rightarrow$ 21 MUX 2:1 $\rightarrow$ 12 DEMUX 1:2 $\rightarrow$ 11 MUX 4:1 $\rightarrow$ 5 DEMUX 1:4 $\rightarrow$ 5 MUX 8:1 $\rightarrow$ 4 DEMUX 1:8 $\rightarrow$ 2 MUX 16:1 $\rightarrow$ 4 DEMUX 1:16 $\rightarrow$ 3 Adder (A) $\rightarrow$ 1 Multiplier (M) $\rightarrow$ 2	130.55	30.98%	36.70 %
JPEG DCT	Register $\rightarrow$ 16 MUX 2:1 $\rightarrow$ 15 DEMUX 1:2 $\rightarrow$ 11 MUX 8:1 $\rightarrow$ 2 DEMUX 1:8 $\rightarrow$ 1 MUX 16:1 $\rightarrow$ 1 DEMUX 1:16 $\rightarrow$ 1 Adder (A) $\rightarrow$ 1 Multiplier (M) $\rightarrow$ 4	47.34	Register $\rightarrow$ 16 MUX 2:1 $\rightarrow$ 27 DEMUX 1:2 $\rightarrow$ 16 MUX 4:1 $\rightarrow$ 1 DEMUX 1:4 $\rightarrow$ 1 MUX 8:1 $\rightarrow$ 3 DEMUX 1:8 $\rightarrow$ 2 MUX 16:1 $\rightarrow$ 1 DEMUX 1:16 $\rightarrow$ 1 Adder (A) $\rightarrow$ 1 Multiplier (M) $\rightarrow$ 4	66.12	40%	

Results and Analysis

TABLE 1B
AREA OVERHEAD ANALYSIS OF THE PROPOSED RAT-HT
INSERTED HLS IP DESIGNS

Benchmarks	IP design without Trojan (Area in μm^2)	IP design with RAT-HT (Area in μm^2)	Area Overhead (%)
MESA	384.763	400.875	3.8%
Sharpening Filter	864.155	881.408	1.9%
LED Filter	664.281	678.339	2.1%
FIR Filter	277.269	304.107	9.6%
JPEG DCT	381.82	401.35	5%

TABLE 1D
POWER EXHAUSTION COMPARISON OF THE PROPOSED RAT-HT WITH [1]

Benchmarks	Power consumed due to [1] (μW)	Power consumed due to the proposed Trojan (μW)	% enhancement in power drainage due to the proposed approach w. r. t. [1]
MESA Filter	33.09	41.93	26.71%
Sharpening Filter	96.71	110.97	14.36%
LED Filter	55.32	74.66	34.96%
FIR Filter	120.78	130.55	8.09%
JPEG DCT	47.7	66.12	39%

TABLE 1C
DESIGN TIME OVERHEAD OF THE PROPOSED RAT-HT
INFECTED IP DATAPATH W. R. T. TROJAN FREE DESIGN FOR
DIFFERENT BENCHMARKS

Benchmarks	Trojan free datapath IP (ps)	Trojan induced datapath IP (ps)	Overhead %
MESA Filter	993.64	1019.12	2.5%
Sharpening Filter	794.91	820.39	3.2%
LED Filter	728.67	779.63	6.9%
FIR Filter	1722.31	1760.53	2.2%
JPEG DCT	927.38	952.86	2.74%

TABLE 1E
POWER EXHAUSTION COMPARISON OF THE PROPOSED RAT-HT WITH [2]

Benchmarks	Power consumed due to [2] (μW)	Power consumed due to the proposed Trojan (μW)	% enhancement in power drainage due to the proposed approach w. r. t. [2]
MESA Filter	32.66	41.93	28.38%
Sharpening Filter	93.30	110.97	18.93%
LED Filter	51.92	74.66	43.79%
FIR Filter	102.19	130.55	27.75%
JPEG DCT	47.7	66.12	39%

1. C. Pilato et al., "Black-Hat High-Level Synthesis: Myth or Reality?," IEEE TVLSI, vol. 27 (4), pp. 913-926, 2018.
2. A. Sengupta et al., "M-HLS: Malevolent High-Level Synthesis for Watermarked Hardware IPs," IEEE ESL, vol. 16 (4), pp. 497-500, 2024.

Results and Analysis

TABLE 2A

COMPARISON OF PROPOSED RAT-HT WITH M-HLS [2] AND BLACK-HAT HLS [1] IN TERMS OF TRIGGER LOGIC, HT INSERTION SITE/ PAYLOAD PHASE, DETECTION TECHNIQUES, DETECTION PROBABILITY PARAMETERS, TROJAN IMPACT PAYLOAD AND AREA OVERHEAD

HLS-HT Framework		Trigger Logic	HT insertion site/ payload phase	Detection using Techniques					Detection probability parameters	Trojan Impact Payload (Power drainage)	Area Overhead (avg.)
				[9]	[3]	[7]	[10]	[5]			
Proposed RAT-HT		Time Sensitive unique solitary condition	HLS register allocation and Datapath	X	X	X	X	X	Trojan detection probability = 0 %	High	4.3%
M-HLS [2]	DoS-HT	Rare Event	HLS-Mux-interconnect design	X	X	X	✓	X	FP>0% FN>0%	Low	~ 2.6%
Black-HAT HLS [1]	DG-HT	Rare Event	HLS scheduling /datapath	X	X	X	✓	X	FP>0% FN>0%	Medium	4%
	BE-HT			X	X	X	✓	✓	For [5]:FP=0% FN=0% For [10]: FP>0% FN>0%		
	DA-HT			X	X	X	✓	X	FP>0% FN>0%		

TABLE 2B

COMPARISON OF PROPOSED RAT-HT WITH M-HLS [2] AND BLACK-HAT HLS [1] IN TERMS OF AVG. DESIGN TIME OVERHEAD, AVG. RUN TIME IMPACT, TRIGGER ROBUSTNESS

HLS-HT Framework	Avg. Design Time Overhead	Avg. Run Time Impact	Trigger Robustness (Trigger Probability)
Proposed RAT-HT	3.5 %	~6.5 secs	2.32E-10
M-HLS [2]	9.8%	7 secs	1.5E-5
Black-HAT HLS [1]	>19%	~7 secs	1.5E-5

1. C. Pilato et al., "Black-Hat High-Level Synthesis: Myth or Reality?," IEEE TVLSI, vol. 27 (4), pp. 913-926, 2018.
2. A. Sengupta et al., "M-HLS: Malevolent High-Level Synthesis for Watermarked Hardware IPs," IEEE ESL, vol. 16 (4), pp. 497-500, 2024.

Results and Analysis

TABLE 3
RAT-HT EVASION CAPABILITY AGAINST KNOWN DETECTION TECHNIQUES
(NOTE: DMR – DUAL MODULAR REDUNDANCY FROM [3], BIPs: BEHAVIORAL IPS, 3PIP: THIRD PARTY IP)

Trojan Detection approach	Proposed Trojan Detection	Annotations
3PIP Defense [9]	X	Unable to detect proposed RAT-HT that causes power exhaustion. It is only capable of handling functional Trojan (i.e. Trojans that change the output of the circuit) using redundancy logic with minimum three distinct IP vendor allocation policy.
TL-HLS [3]	X	Unable to detect proposed RAT-HT that causes power exhaustion. TL-HLS detection technique with DMR design is only able to handle HLS functional trojan.
Property Checking for BIPs [7]	X	Unable to detect proposed RAT-HT that causes power exhaustion. Only capable to detect functional HW Trojans in 3PBIPs where the modelling of the HW Trojan is done using ternary operators. Needs property checker feature in HLS tool for detection. Only applicable for HLS tool from CyberWorkBench. Does not work for other HLS tools/frameworks.
Equivalence checking [10]	X	As this equivalence checking relies on creation of finite state machine with datapath (RTL) for comparing with C code, hence it is only capable to detect BE-based HLS trojan and DA HLS trojan.

3. A. Sengupta et al., “TL-HLS: Methodology for Low Cost Hardware Trojan Security Aware Scheduling with Optimal Loop Unrolling Factor during High Level Synthesis,” IEEE TCAD, 2017.
7. N. Veeranna et al., “Hardware Trojan Detection in Behavioral Intellectual Properties (IP’s) Using Property Checking Techniques,” IEEE TETC, 2017.
9. S. Rajmohan et al., “Hybrid Evolutionary Design Space Exploration Algorithm With Defence Against Third Party IP Vulnerabilities,” IEEE TCAD, 2020.
10. M. Abderehman et al., “BLAST: Belling the Black-Hat High-Level Synthesis Tool,” IEEE TCAD, 2022.

Results and Analysis

TABLE 4A
DEMONSTRATION OF TROJAN EVASION CAPABILITY AGAINST KNOWN DETECTION TECHNIQUES
(NOTE: UOG AND UDP INDICATE ORIGINAL AND DUPLICATE UNITS FROM [3])

Detection approach	Features in proposed IP (baseline) - Design A (U ^{OG})	Features in proposed IP (with RAT-HT) – Design B (U ^{DP})	Justification of evasion	Trojan Detection
3PIP Defense [9]	Baseline design output for LED Filter $O_0 = 5$ Baseline design output for LED Filter $O_1 = 5$ (Note: a sample value is reported based on a specific test pattern provided. Large exhaustive set of test patterns were tested)	Trojan infected design output for LED Filter $O_0 = 5$ Trojan infected design output for LED Filter $O_1 = 5$ (Note: a sample value is reported based on a specific test pattern provided. Large exhaustive set of test patterns were tested)	Difference in functional o/p between designs A & B using comparator: Nil	X
	Sample test vectors for U ^{OG} : $\{i_1 \dots i_{14} = 1\}$ U ^{OG} : allocated to 3PIP vendor 1	Sample test vectors for U ^{DP} : $\{i_1 \dots i_{14} = 1\}$ U ^{DP} : allocated to 3PIP vendor 2		
HLS Trojan detection and recovery [18]	U ^{OG} o/p for LED Filter $O_0 = 20$ U ^{OG} o/p for LED Filter $O_1 = 20$ Sample test vectors for U ^{OG} : $\{i_1 \dots i_{14} = 2\}$	U ^{DP} o/p for LED Filter $O_0 = 20$ U ^{DP} o/p for LED Filter $O_1 = 20$ Sample test vectors for U ^{DP} : $\{i_1 \dots i_{14} = 2\}$	Difference in functional o/p between designs A & B using comparator: Nil	X
TL-HLS [3]	Baseline design output for LED Filter $O_0 = 45$ Baseline design output for LED Filter $O_1 = 45$ (Note: a sample value is reported based on a specific test pattern provided. Large exhaustive set of test patterns were tested)	Trojan infected design output for LED Filter $O_0 = 45$ Trojan infected design output for LED Filter $O_1 = 45$ (Note: a sample value is reported based on a specific test pattern provided. Large exhaustive set of test patterns were tested)	Difference in functional o/p between designs A & B using comparator: Nil	X
	Sample test vectors for U ^{OG} : $\{i_1 \dots i_{14} = 3\}$	Sample test vectors for U ^{DP} : $\{i_1 \dots i_{14} = 3\}$		

3. A. Sengupta et al., "TL-HLS: Methodology for Low Cost Hardware Trojan Security Aware Scheduling with Optimal Loop Unrolling Factor during High Level Synthesis," IEEE TCAD, 2017.
9. S. Rajmohan et al., "Hybrid Evolutionary Design Space Exploration Algorithm With Defence Against Third Party IP Vulnerabilities," IEEE TCAD, 2020.
18. X.Cui, et al., "High-level synthesis for run-time hardware Trojan detection and recovery," 51st ACM/IEEE DAC, CA, 2014, pp. 1-6.

Results and Analysis

TABLE 4B

POWER SCA ANALYSIS ON RAT-HT INFECTED HLS BASED IP DESIGNS

Benchmarks	Trojan Free Datapath IP (Power in μ W)	Trojan inserted datapath IP before activation/triggering (Power in μ W)	Power Overhead
MESA Filter	30.13	30.99	3 %
Sharpening Filter	90.77	93.86	3.4%
LED Filter	49.39	51	4%
FIR Filter	99.66	103.56	3.9%
JPEG DCT	47.34	51.24	4%

TABLE 4D

CONFUSION MATRIX OF THE GNN MODEL FOR DIFFERENT HLS TROJAN VARIANCE

NOTE: TEST_SIZE=0.20, RANDOM_STATE=42, LR=0.01, EPOCHS = 200

	Predicted Label 0	Predicted Label 1	Predicted Label 2	Predicted Label 3	Predicted Label 4	Predicted Label 5	Predicted Label 6	Predicted Label 7
Actual 0	0	0	0	0	0	0	0	0
Actual 1	0	16	0	0	0	0	0	0
Actual 2	0	0	2	0	0	0	0	0
Actual 3	1	0	0	2	0	0	0	0
Actual 4	1	1	0	0	2	0	0	0
Actual 5	0	0	0	0	0	2	0	0
Actual 6	0	0	0	0	0	0	2	0
Actual 7	1	1	0	0	0	0	0	0

TABLE 4C

EPOCHS AND LOSSES AS WELL AS ACCURACY EVALUATION OF GNN MODEL [6] FOR RAT-HT AND OTHER INSTANCES
NOTE: TEST_SIZE=0.20, RANDOM_STATE=42 LR=0.001 EPOCHS = 200
LABEL 1: RAT-HT INSTANCES IN THE DATABASE
LABEL 0: OTHER INSTANCES (NON RAT-HT)

Epoch 000	Loss = 0.859542
Epoch 010	Loss = 0.288541
Epoch 020	Loss = 0.124131
Epoch 030	Loss = 0.069241
Epoch 040	Loss = 0.047515
Epoch 050	Loss = 0.041266
Epoch 060	Loss = 0.034195
Epoch 070	Loss = 0.031278
Epoch 080	Loss = 0.025559
Epoch 090	Loss = 0.024116
Epoch 100	Loss = 0.014943
Epoch 110	Loss = 0.011846
Epoch 120	Loss = 0.012652
Epoch 130	Loss = 0.011814
Epoch 140	Loss = 0.008823
Epoch 150	Loss = 0.008354
Epoch 160	Loss = 0.007240
Epoch 170	Loss = 0.007378
Epoch 180	Loss = 0.005602
Epoch 190	Loss = 0.004923
Label 0	Accuracy: 1.0
Label 1	Accuracy: 0.0

Results and Analysis

TABLE 4E
ACCURACY EVALUATION OF GNN MODEL FOR
DIFFERENT HLS TROJAN VARIANCES (INCLUDING
PROPOSED RAT-HT)

Labels	Instances	Accuracy
Label 0	Non-infected	0.00
Label 1	PD-HT	1.00
Label 2	DoS-HT	1.00
Label 3	BE-HT	0.66
Label 4	DA-HT	0.50
Label 5	F-HT	0.66
Label 6	TB-HT	1.00
Label 7	RAT-HT	0.00

TABLE 5
FPGA SYNTHESIS OF MESA TROJAN INFECTED IP DESIGN

Parameters	Value
Quartus II 64-Bit Version	13.1.0 Build 162 10/23/2013 SJ
Family	Cyclone IV GX
Device	EP4CGX150DF31I7AD
Total logic elements	498 / 149,760 (< 1 %)
Embedded Multiplier 9-bit elements	4 / 720 (< 1 %)

References

1. C. Pilato et al., "Black-Hat High-Level Synthesis: Myth or Reality?," IEEE TVLSI, vol. 27 (4), pp. 913-926, 2018.
2. A. Sengupta et al., "M-HLS: Malevolent High-Level Synthesis for Watermarked Hardware IPs," IEEE ESL, vol. 16 (4), pp. 497-500, 2024.
3. A. Sengupta et al., "TL-HLS: Methodology for Low Cost Hardware Trojan Security Aware Scheduling with Optimal Loop Unrolling Factor during High Level Synthesis," IEEE TCAD, Vol.36(4), pp.655-668, 2017.
4. Hassan Rakibul et al., "Circuit topology-aware vaccination-based hardware trojan detection." IEEE TCAD, 42(9), 2023: 2852-2862.
5. M. Rathor, A. Sengupta., "Revisiting Black-Hat HLS: A Lightweight Countermeasure to HLS-Aided Trojan Attack," IEEE ESL, vol.16(2), pp. 170-173, 2024.
6. R. Yasaei, S. -Y. Yu and M. A. Al Faruque, "GNN4TJ: Graph Neural Networks for Hardware Trojan Detection at Register Transfer Level," 2021 Design, Automation & Test in Europe Conference & Exhibition (DATE), Grenoble, France, 2021, pp. 1504-1509.
7. N. Veeranna et al., "Hardware Trojan Detection in Behavioral Intellectual Properties (IP's) Using Property Checking Techniques," IEEE TETC, vol. 5, no. 4, pp. 576-585, 2017.
8. A. Sengupta, S. P. Mohanty "High-Level Synthesis of Digital Circuits in the Nanoscale, Mobile Electronics Era", IET Book: Nano-CMOS and Post-CMOS Electronics: Circuits and Design, Book Chapter, e-ISBN: 9781785610004, pp: 219 - 261, June 2016.
9. S. Rajmohan et al., "Hybrid Evolutionary Design Space Exploration Algorithm With Defence Against Third Party IP Vulnerabilities," IEEE TCAD, vol. 39(10), pp. 2602-2614, 2020.
10. M. Abderehman et al., "BLAST: Belling the Black-Hat High-Level Synthesis Tool," IEEE TCAD, vol. 41(11), pp. 3661-3672, 2022.
11. M. Rostami et al., "A Primer on Hardware Security: Models, Methods, and Metrics," Proceedings of IEEE, vol. 102(8), pp. 1283-1295, 2014.
12. Open Cell NanGate Library, accessed on March 2026, Available: <https://si2.org/open-cell-library/>
13. Jiang, Zijing et al.. "A framework for hardware trojan detection based on contrastive learning." Scientific Reports 14, no.1 (2024): 30847.
14. A. Sengupta et al., "A High Level Synthesis design flow with a novel approach for Efficient Design Space Exploration in case of multi parametric optimization objective", Elsevier Microelectronics Reliability, Vol. 50 (3), 2010, pp. 424-437.
15. Naveenkumar, R., et al.,. "Hardware trojans detection and prevention techniques review." Wireless Personal Communications 136, no. 2 (2024): 1147-1182.
16. https://github.com/phd2401201010-hash/Design_and_Test.
17. A. Sengupta, N. Bhui "Time-Bomb HLS Trojan for Performance Degradation Payload", IEEE Design & Test (D&T), Volume: 42, Issue: 5, pp. 83 - 92, October 2025.
18. X.Cui, et al., "High-level synthesis for run-time hardware Trojan detection and recovery," 51st ACM/IEEE DAC, CA, 2014, pp. 1-6.