

T-HAC: Time-Sensitive HLS Trojan Attack for Affecting Controller Timing Signal of Hardware IP Designs

Anirban Sengupta, Vishal Chourasia, Nabendu Bhui and Rimil Mandi , "T-HAC: Time-Sensitive HLS Trojan Attack for Affecting Controller Timing Signal of Hardware IP Designs," *IEEE Embedded Systems Letters*, doi: 10.1109/LES.2026.3696630

Threat Model



- A malicious HLS framework/tool is capable of inserting hardware Trojans (HT) during the Mux-based interconnect stage of watermarked/non watermarked IP design.
- Some Trojans target change in computational output value in 3PIPs. It does not steal information/data.
- The impact of a malicious HLS tool may be performance degradation or denial of service.

[1] 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 Transaction on Computer-Aided Design of Integrated Circuits and Systems*, Vol.36 (4), pp.655-668, 2017.

[2] A. Sengupta et al., "M-HLS: Malevolent High-Level Synthesis for Watermarked Hardware IPs," *IEEE Embedded Systems Letters*, vol. 16 (4), pp. 497-500, 2024.

Threat Model



- The malicious HLS tool may be unreliable/untrusted and can be compromised containing potential backdoor hardware Trojan without the awareness of the original IP designer or vendor.
- Malicious logic can be covertly inserted/embedded into intermediate stages of the HLS flow, which remains dormant during normal operation and activates only when a specific adversarial trigger is met (predefined by the attacker/adversary).
- Compromised HLS tool may be distributed as part of large-scale or nation-wide toolkit, where tool developed in one country can be used/integrated in a design flow of another country.
- More explicitly, the target of designing malevolent HLS tool is to create inferior IP components in the market.

HLS Trojans



- HLS-IP Trojan is classified as per the following design stage insertion points:
 - HLS tool library (as components)— functional Trojan (F-HT) [1],
 - HLS interconnect design stage—denial of service hardware Trojan (DOS-HT)[2],
 - HLS scheduling stage—performance degradation hardware Trojan (PD-HT)/degradation hardware Trojan (DG-HT) [2],
 - HLS finite-state-machine (FSM) design stage—battery exhaustion hardware Trojan (BE-HT) [10],
 - HLS hidden directives—downgrade attack hardware Trojan (DA-HT) [10].

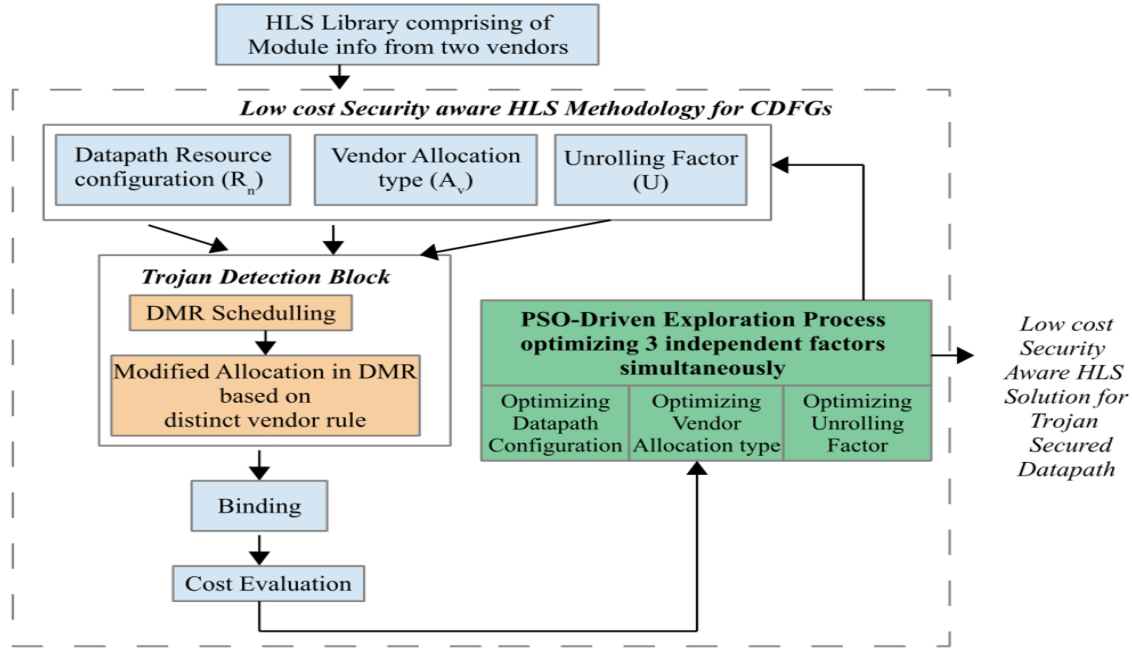
[1] A. Sengupta et. al., “TL-HLS: Methodology for Low-Cost Hardware Trojan Security Aware Scheduling with Optimal Loop Unrolling Factor during High Level Synthesis,” *IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems*, Vol.36 (4), pp.655-668, 2017.

[2] A. Sengupta et al., “M-HLS: Malevolent High-Level Synthesis for Watermarked Hardware IPs,” *IEEE Embedded Systems Letters*, vol. 16 (4), pp. 497-500, 2024.

[10] C. Pilato et al., “Black-Hat High-Level Synthesis: Myth or Reality?,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27 (4), pp. 913-926, 2019

[14] Sengupta, A., Chourasia, V., & Bhui, N. (2025). “HLS Trojan Classification and Its Detection Techniques for CAD-Based IP Designs.” *IEEE IT Professional*, 27(5), 66–74.

State-of-the-art: Trojan Detection Methodology



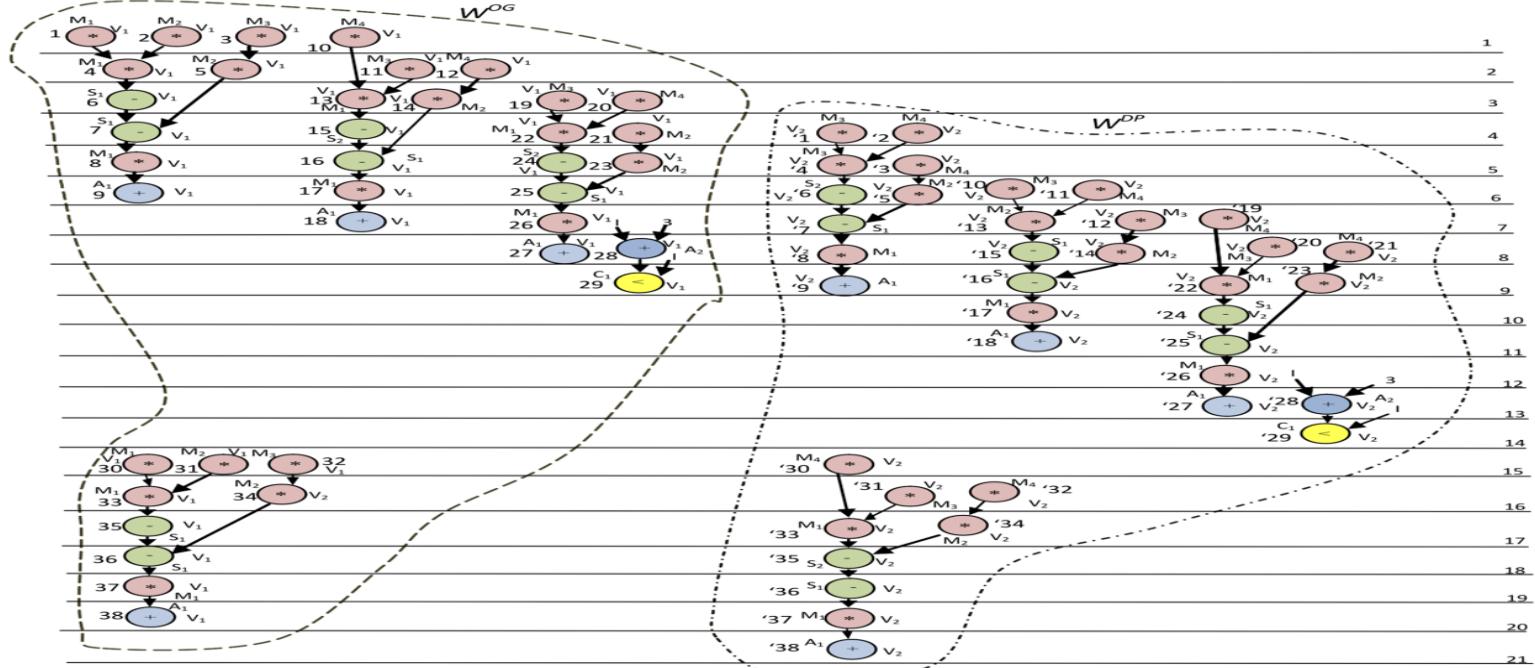
Key idea behind low cost Trojan secured schedule during HLS.

Flow of the Trojan Detection Method



- This technique allocated two distinct vendors within the DMR schedule controls the final delay and area of the design.
- Distinct vendor allocation
 - Vendor Allocation Procedure [Type 1 ($P_v = 1$)]:
 - All operations of a specific unit (original/duplicate), are strictly assigned to same vendor type.
 - Similar operations of both original unit (W^{OG}) and duplicate unit (W^{DP}) being assigned to two different vendors.

Vendor Allocation Procedure [Type 1 ($P_v = 1$)]:

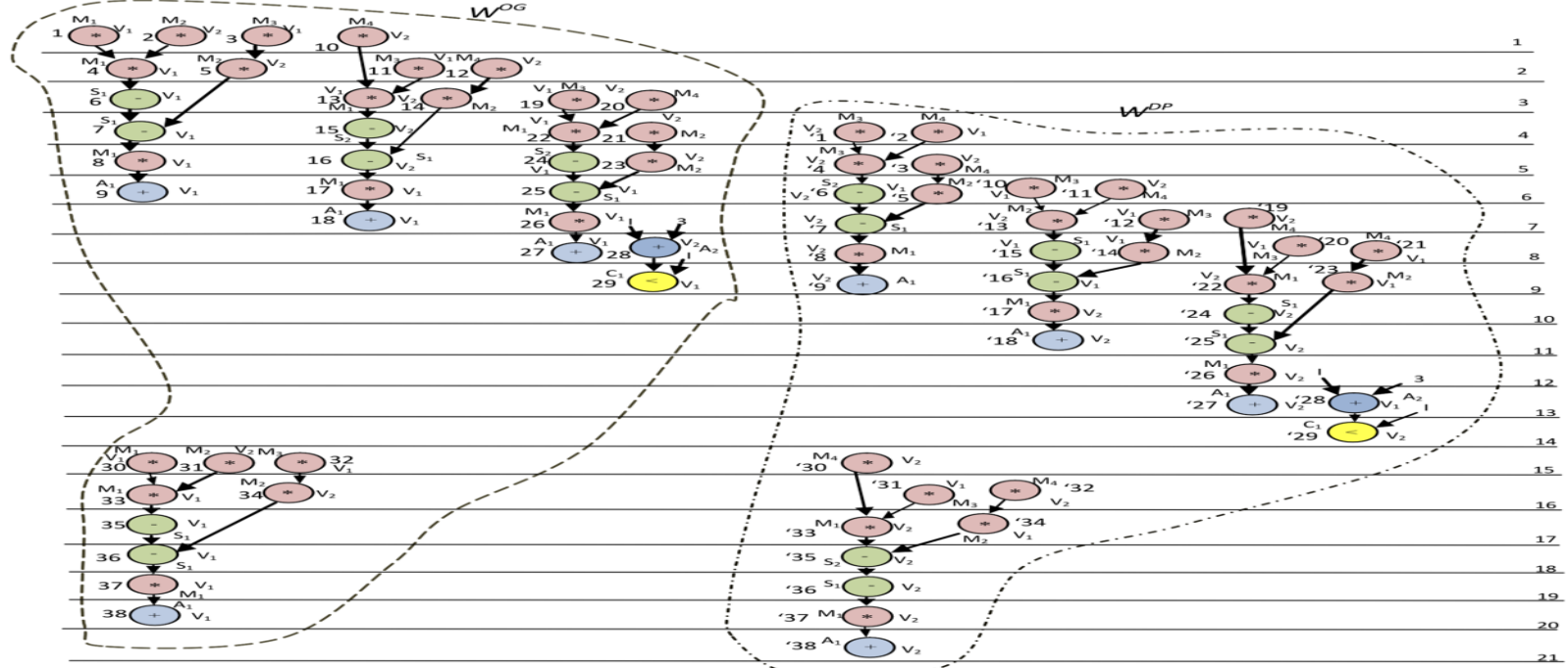


Scheduled and bounded DMR design of differential equation for $X_i = \{2(+), 4(*), 2(-), 2(<), U=3\}$ with $P_v = 1$;

Flow of the Trojan Detection Method(contd.)

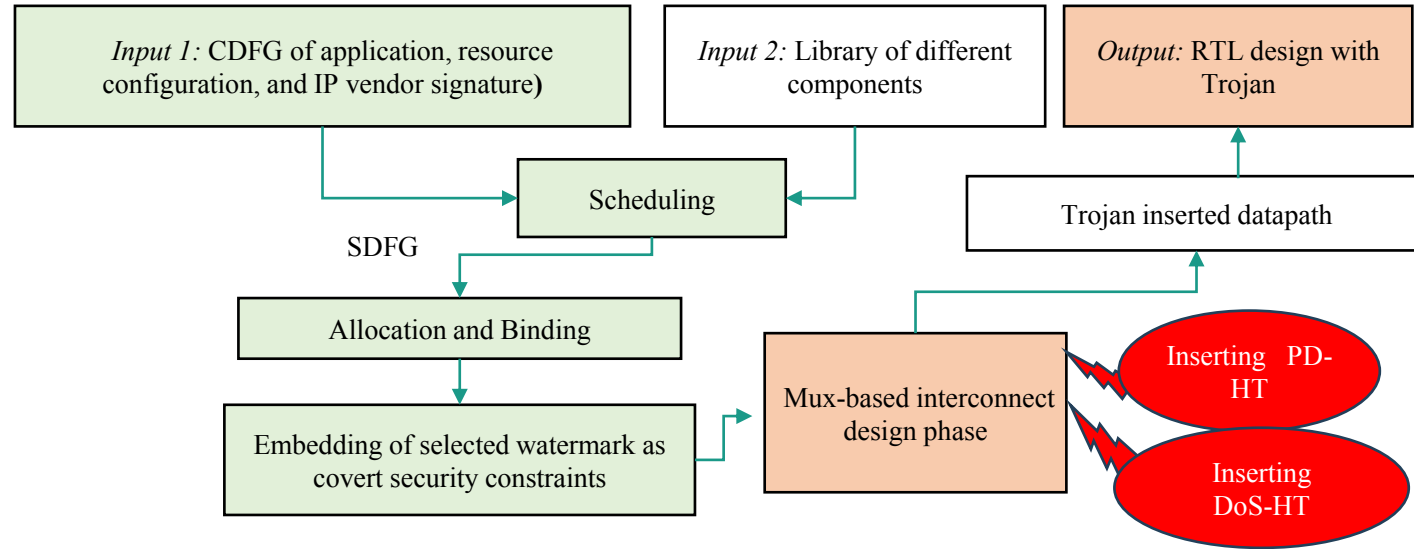
- 
- Vendor Allocation Procedure [Type 2 ($P_v = 0$)]:
 - Alternate vendor assignment to operations of a resource type in a CS of a unit. Moreover, in a CS if an operation with another resource type is encountered then independently alternate vendor assignment is performed.
 - Similar operations of both W^{OG} and W^{DP} being assigned to two different vendors.

Vendor Allocation Procedure [Type 2 ($P_v = 0$)]:



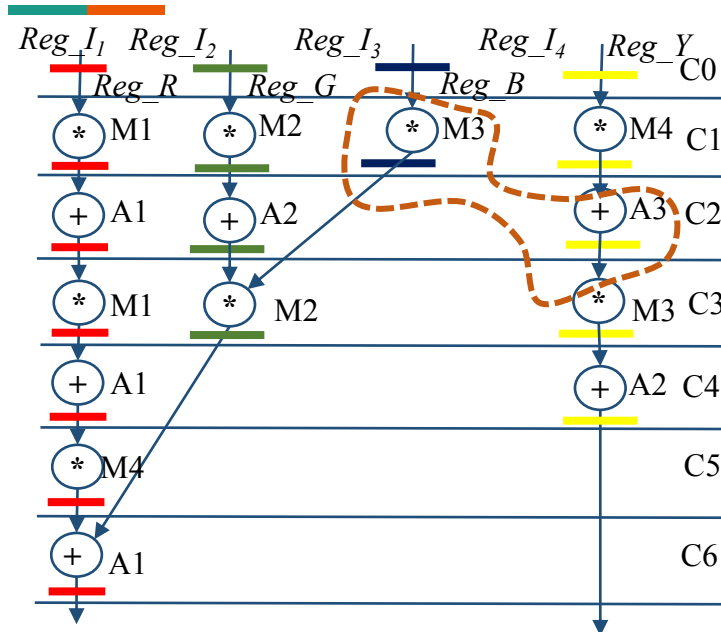
Scheduled and bounded DMR design of differential equation for $\bar{X}_i = \{(2(+), 4(*), 2(-), 2(<), U=3), P_v\}$ with $P_v = 0$ and $I=4$;

State-of-the-art: Malevolent HLS Framework

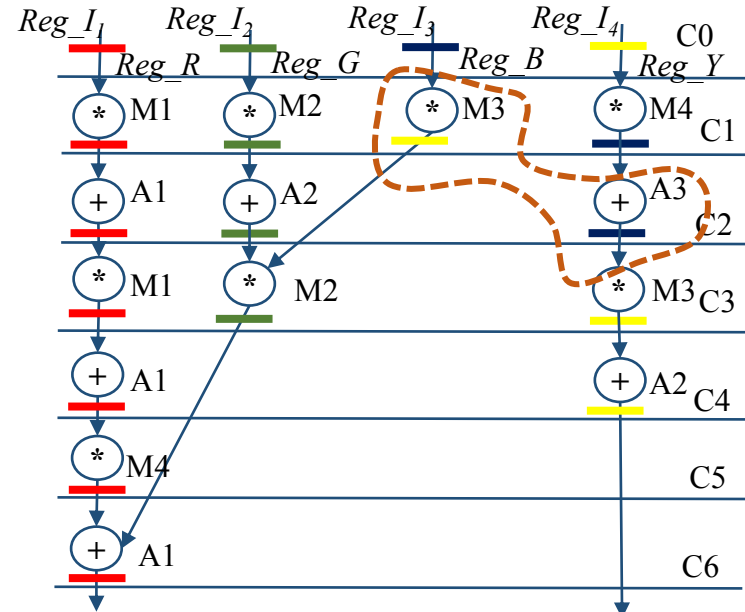


Malevolent High-Level Synthesis framework

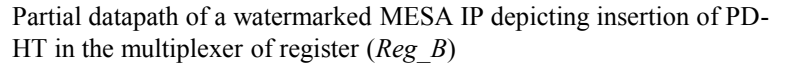
Trojan Vulnerability in HLS-Based Watermarked IPs



Scheduled data flow graph of MESA Horner Bezier without watermark



Scheduled data flow graph of MESA Horner Bezier with embedded watermark



Variance of Trojan Detection Approach

Comparison of Trojan Defense Techniques. Note: FN: false negative, FP: false positive, NA: not available

Trojan detection approach	Trojan handling capability					Defense capability	Additional resources/overhead/tools required	Justifications
	F-HT	PD-HT	DOS-HT	BE-HT	DA-HT			
HLT based detection [15]	X	X	X	√	X	Detection	HLT tool for operation comparison	HLT based detection technique is only able to handle fake operations as trojans (therefore is able to detect only HLS based battery exhaustion trojan attacks)
						FN = 0% FP = 0%		
TL-HLS [1]	√	X	X	X	X	Detection	Single distinct third-party IP vendor (3PIP)	TL-HLS detection technique with DMR design is only able to handle HLS functional trojan (unable to detect trojan with performance degradation payload)
						FN= 0% FP= 0%		
DMR based [16]	√	X	X	X	X	Detection	Single distinct third-party IP vendor (3PIP)	DMR design is only able to handle HLS functional trojan
						FN= 0% FP= 0%		
Trojan resistance [17]	√	X	X	X	X	Detection and isolation of F-HT only	Two distinct third-party IP vendors (3PIP)	Trojan resistance technique with TMR design is only able to handle HLS functional trojan (unable to detect trojan with performance degradation payload)
						FN= 0% FP= 0%		
C to RTL Equivalence checking [7]	X	X	X	√	√	Detection	Generation of RTL level finite-state machine with datapaths (RTL-FSMD) from the HLS generated RTL	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 battery exhaustion-based HLS trojan and downgrade attack HLS trojan
						FN>0% FP >0%		

Proposed Work



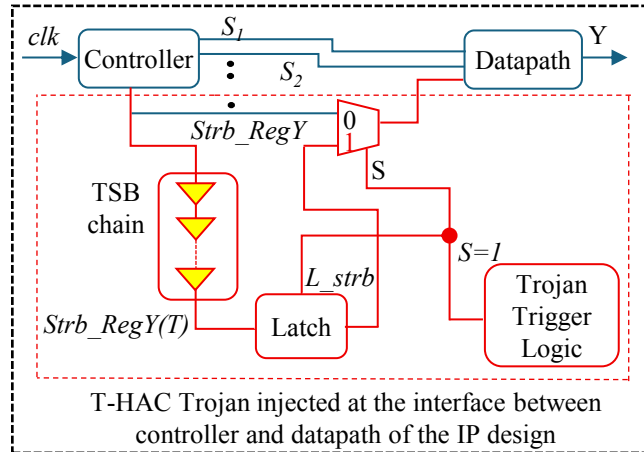
- A new class of HLS-aided Trojan attack called time- sensitive HLS Trojan attack for affecting controller timing signal of hardware IP designs has been developed.
- It is being automatically injected/embedded in a stealthy manner by an adversary/attacker during HLS datapath designing stage at the interface between controller and datapath of the IP design.
- The goal of the hardware Trojan (HT) is to expose the vulnerability of hardware IP design in terms of yielding significant timing signal delay (degradation), at nominal design area/power overhead.

Proposed T-HAC: Trojan Trigger Logic

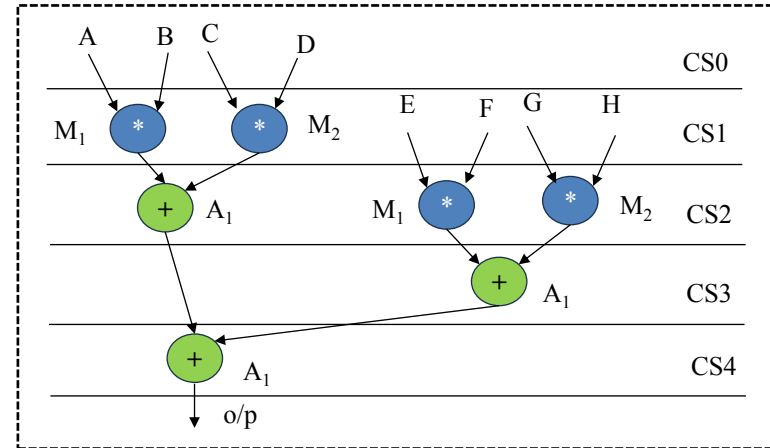


- A predefined value 'U' is loaded in the register, and a mod 'V' up-counter increments during normal operation.
- When, the modulus (mod) 'V' up-counter value reaches the predefined value 'U' under a rare-event condition (*i.e.*, $U=V$)
- Then, the time-driven trigger signal $S=1$ is generated, activating the Trojan trigger logic.
- In other conditions, the T-HAC Trojan stays dormant as long as the attacker intends to.

The Proposed HLS-aided T-HAC Trojan Framework



Proposed HLS-aided T-HAC Trojan



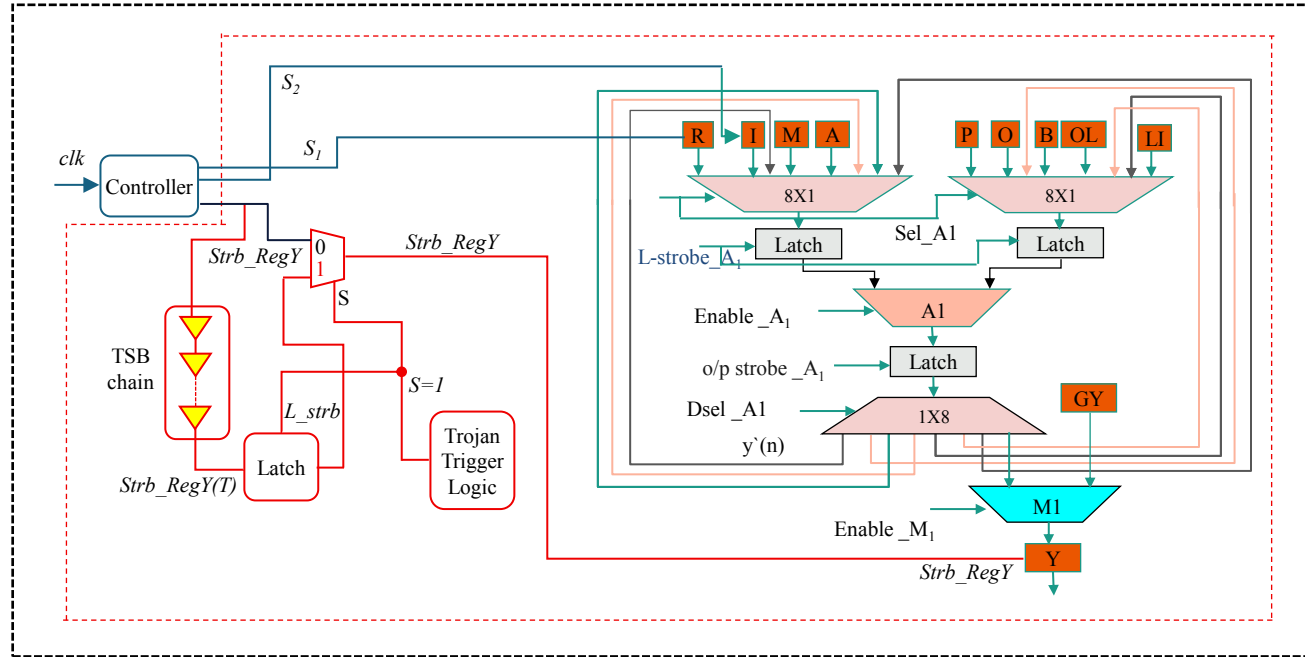
SDFG of 4-point DCT

Proposed T-HAC: Trojan Logic



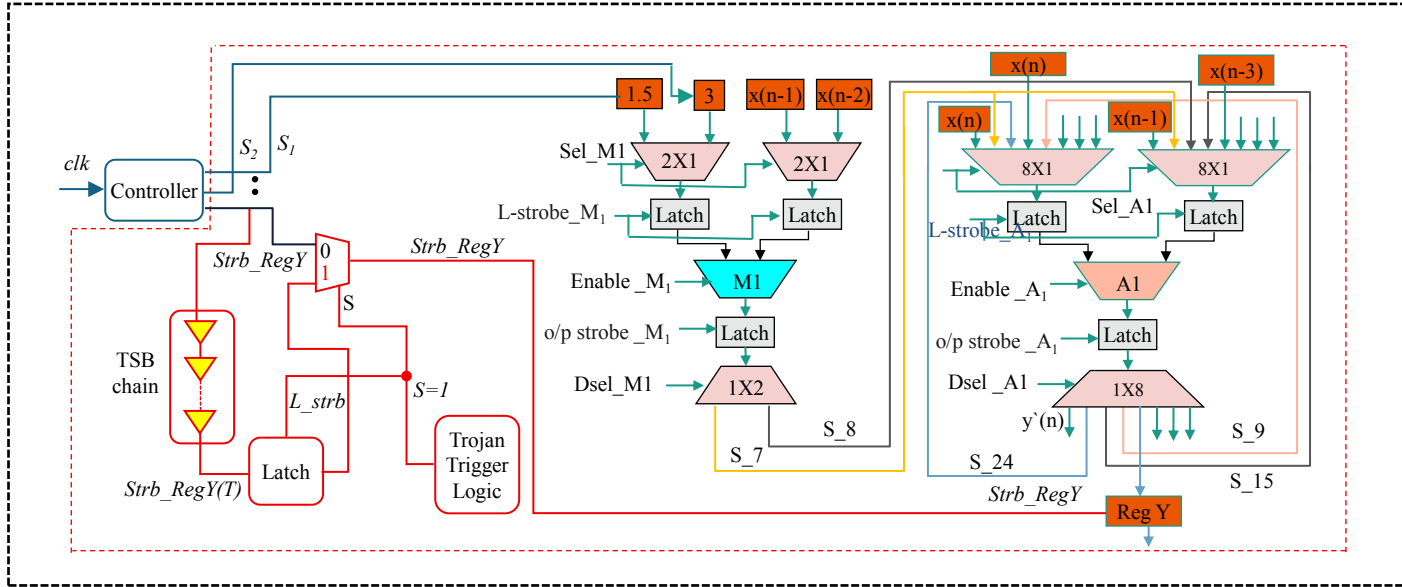
- When trigger signal $S=1$, activates the 2:1 mux (select line 1) and latch (L_strb signal).
- This causes the Trojan-induced Strb_RegY timing signal (Strb_RegY(T)) to pass through the TSB chain.
- The TSB chain adds unwarranted timing delay to the Strb_RegY signal (while $S=1$).
- The TSB chain length (α) is selected based on the amount of intended timing delay to be induced by the attacker.

The Proposed HLS-aided T-HAC Trojan Framework



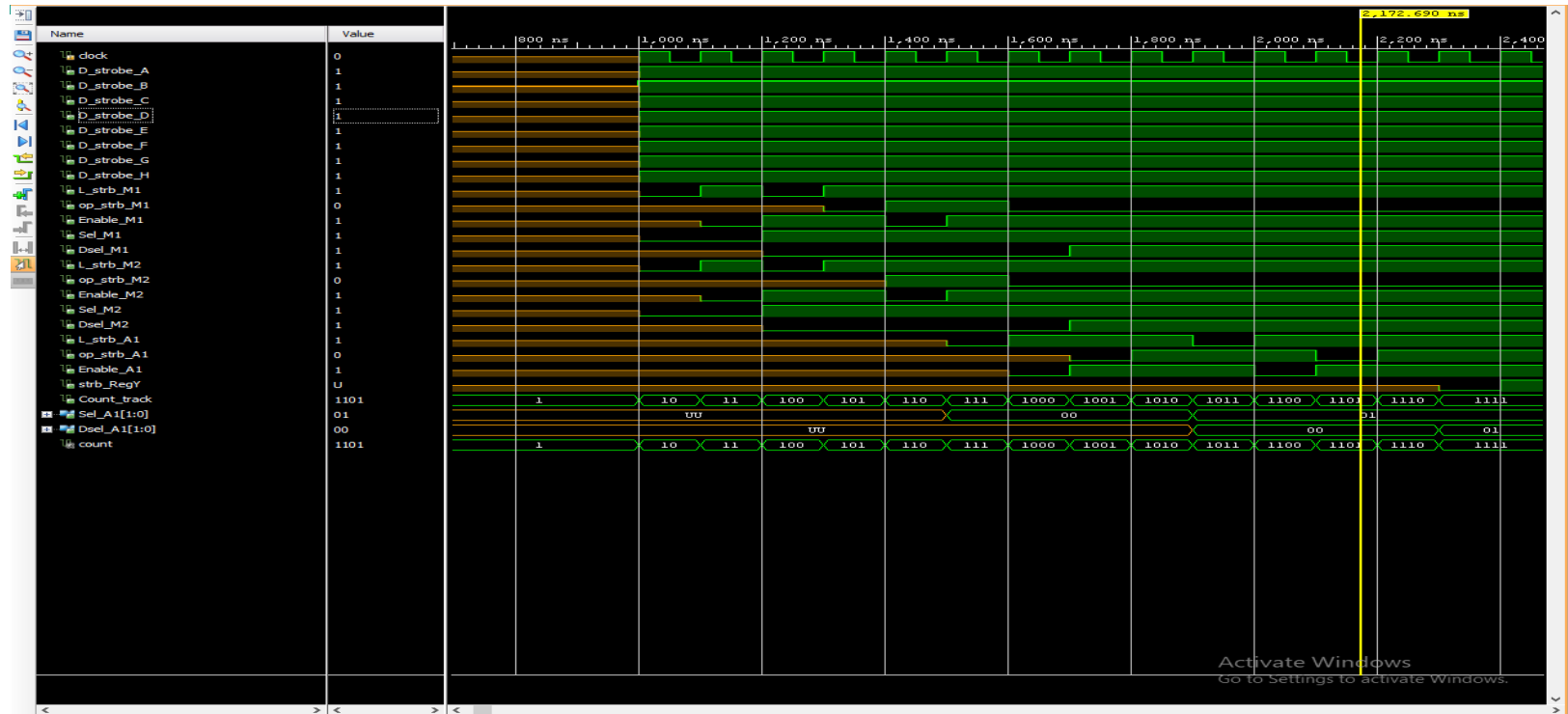
Proposed HLS-aided T-HAC Trojan injected IP design of Blur Filter Datapath benchmark

The Proposed HLS-aided T-HAC Trojan Framework



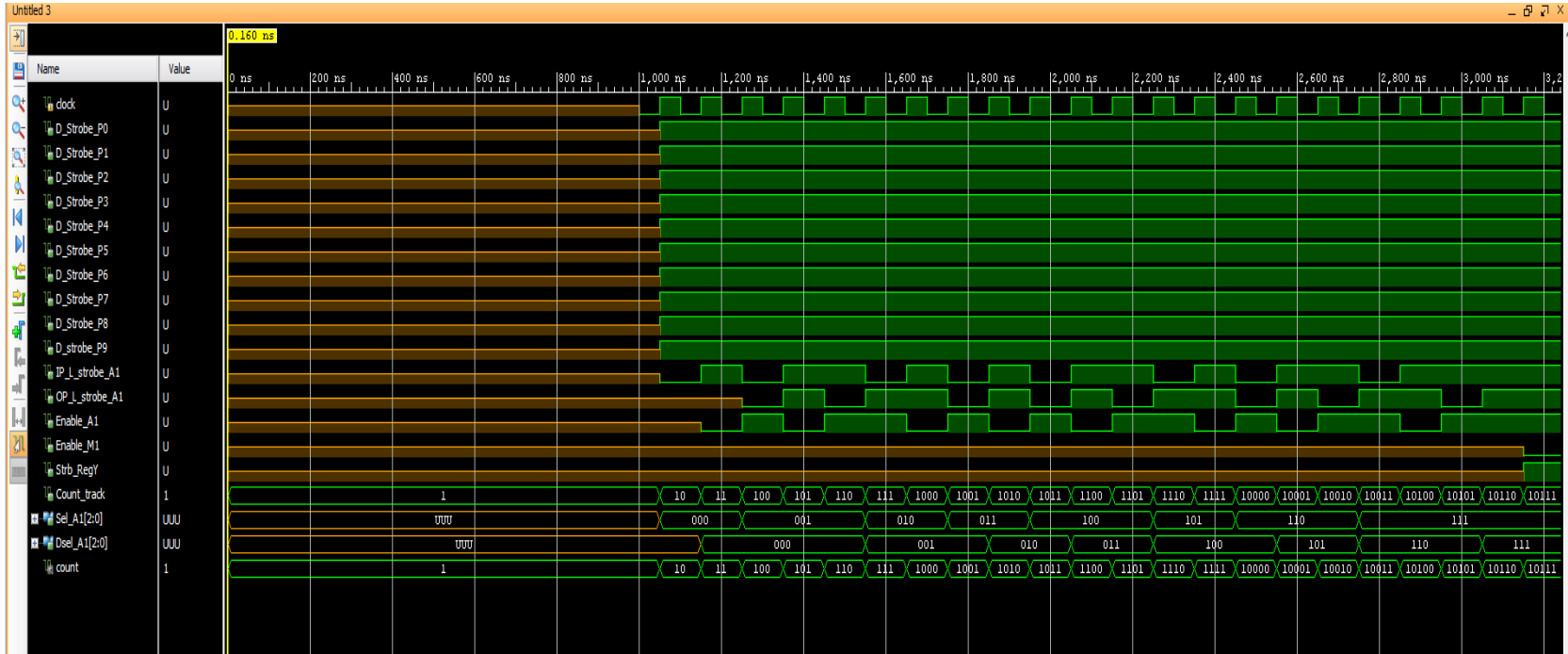
Proposed HLS-aided T-HAC Trojan injected IP design of Cardiac Pacemaker Datapath benchmark

Controller Design and Simulation



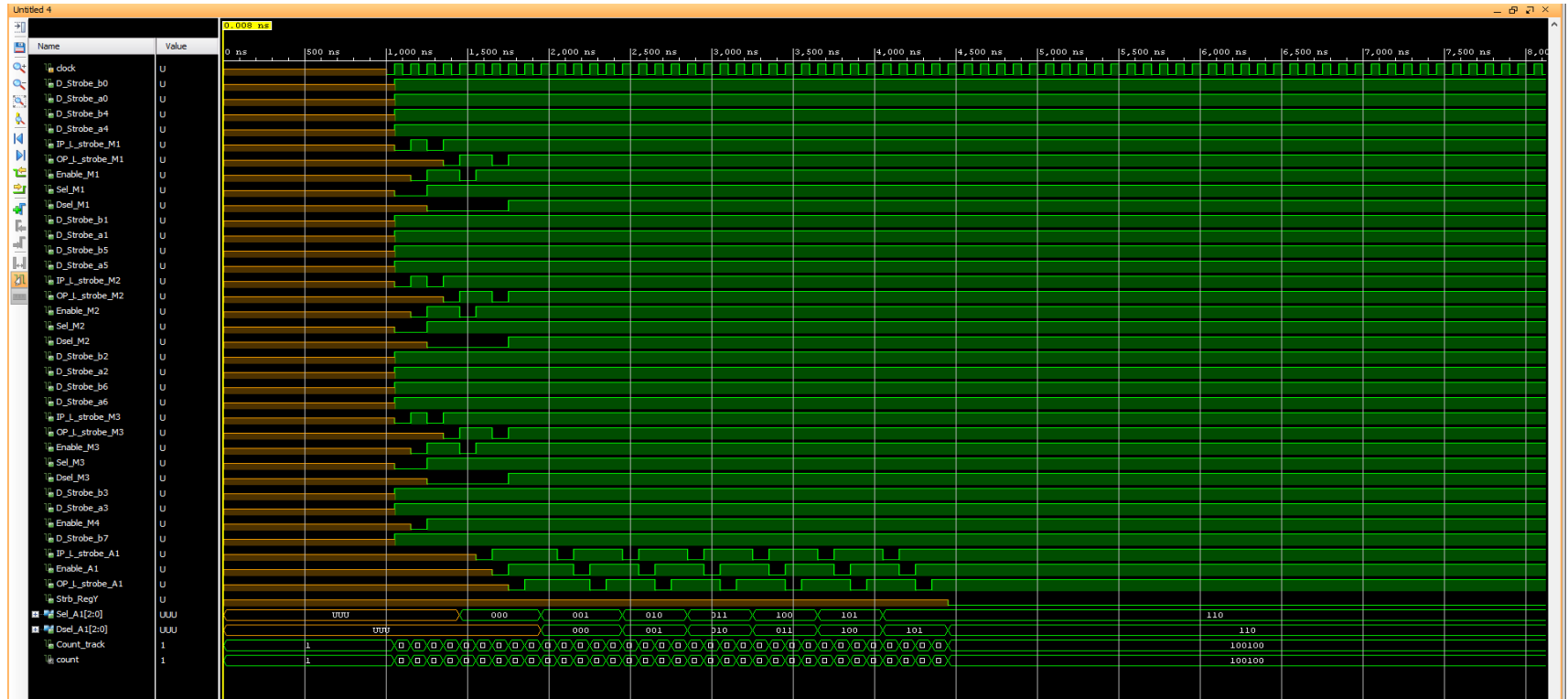
Controller simulation of 4-point DCT

Controller Design and Simulation



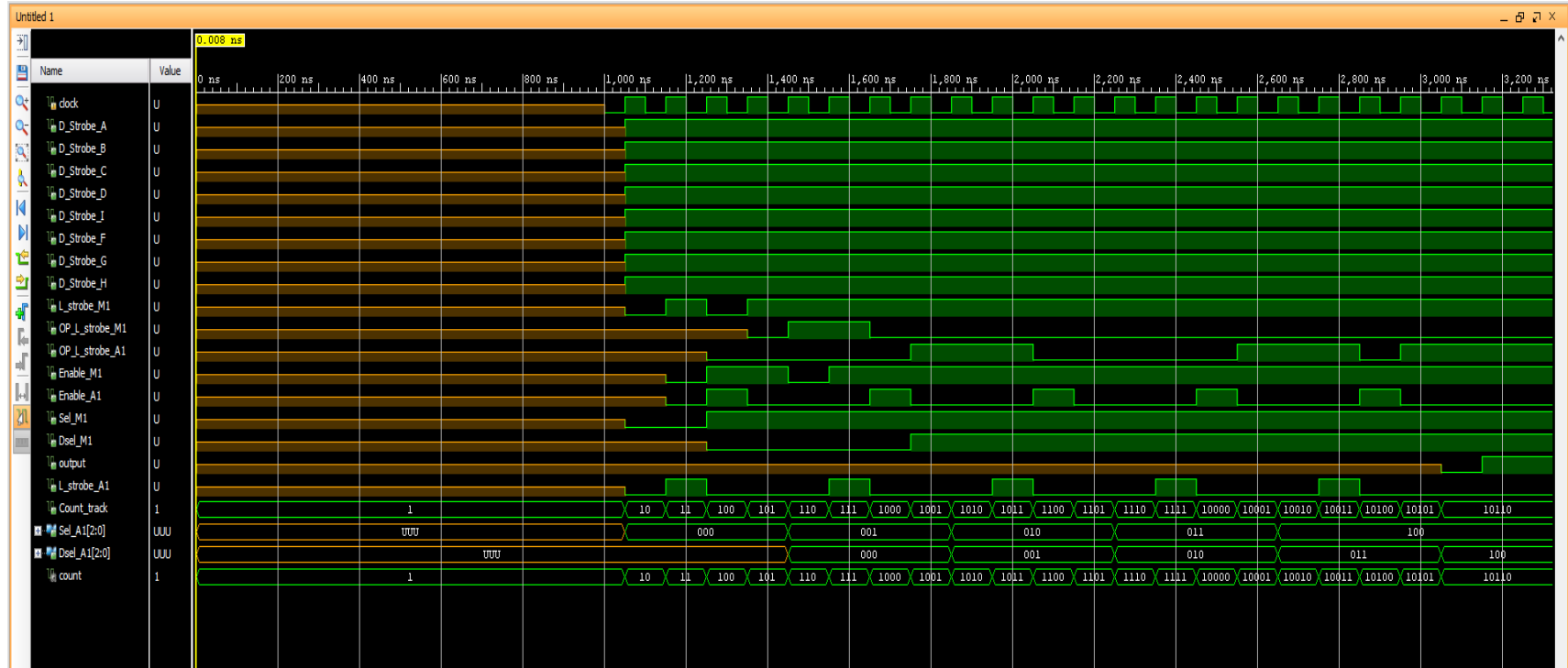
Controller simulation of Blur Filter

Controller Design and Simulation



Controller simulation of FIR

Controller Design and Simulation



Controller simulation of LPF

Results and Analysis

Timing Degradation Achieved due to Proposed T-HAC Trojan for Different Benchmarks (in Terms of Clock Cycle)

Benchmarks	Controller design timing in clock cycles (Trojan free)	Controller design timing in clock cycles (with Trojan) ($\alpha \rightarrow$ TSB chain length)				Timing degradation due to proposed T-HAC				Percentage degradation due to proposed T-HAC			
		$\alpha=4$	$\alpha=6$	$\alpha=8$	$\alpha=10$	$\alpha=4$	$\alpha=6$	$\alpha=8$	$\alpha=10$	$\alpha=4$	$\alpha=6$	$\alpha=8$	$\alpha=10$
4-point DCT	18 cc	24 cc	26 cc	28 cc	30 cc	6	8	10	12	~33%	~44%	~56%	~67%
Cardiac Pacemaker LPF	21 cc	27 cc	29 cc	31 cc	33 cc	6	8	10	12	~29%	~38%	~48%	~57%
FIR Filter (UF=7)	36 cc	42 cc	44 cc	46 cc	48 cc	6	8	10	12	~17%	~22%	~28%	~33%
Blur Filter	23 cc	29 cc	31 cc	33 cc	35 cc	6	8	10	12	~26%	~35%	~43%	~52%

Results and Analysis

Comparison Between Controller Design Schedule Latency (Trojan Free) vs Controller Design Schedule Latency (with Trojan (T-HAC))

Benchmarks	Controller design driven schedule latency (Trojan free)	Controller design driven schedule latency (with Trojan) (in pico-secs (ps))				Percentage (%) degradation due to proposed T-HAC (in pico-secs (ps))			
		$\alpha=4$	$\alpha=6$	$\alpha=8$	$\alpha=10$	$\alpha=4$	$\alpha=6$	$\alpha=8$	$\alpha=10$
4-point DCT	662.43 ps	733.66	756.32	778.98	801.64	10.75	14.17	17.59	21.02
Pacemaker LPF	728.67 ps	799.90	822.56	845.22	867.88	9.78	12.89	15.99	19.10
FIR Filter (UF=7)	927.40 ps	998.63	1021.29	1043.95	1066.61	7.68	10.12	12.57	15.01
Blur Filter	794.91 ps	866.14	888.80	911.46	934.12	8.96	11.81	14.66	17.51

Comparison of Delay Degradation and Area Overhead Achieved by an Attacker using M-HLS[5], Black-Hat[9], and Proposed T-HAC

Parameter	M-HLS [5]	Black-Hat HLS [9]	Proposed T-HAC
Delay degradation	Up to 15%	Up to 17%	Up to 21.02%
Area overhead	Up to 1.89%	Up to 4%	Up to 1.68%

Results and Analysis

Area and Power Overhead Due to Insertion of Proposed T-HAC Trojan For Different IP Design

Benchmarks (IP design)	Parameters		Trojan free design		IP design with T-HAC ($\alpha = 4$)		IP design with T-HAC ($\alpha = 6$)		IP design with T-HAC ($\alpha = 8$)		IP design with T-HAC ($\alpha = 10$)	
4-point DCT	Area (gates)	Power (μ w)	3044.00	32.43	3088	32.93	3090	32.95	3092	32.96	3094	32.98
	Area Ovh.	Power Ovh.	--	--	1.45%	1.56%	1.51%	1.60%	1.58%	1.65%	1.64%	1.70%
Cardiac Pacemaker LPF	Area (gates)	Power (μ w)	3252.00	31.57	3296	32.08	3298	32.09	3300	32.11	3302	32.12
	Area Ovh.	Power Ovh.	--	--	1.35%	1.60%	1.41%	1.65%	1.48%	1.70%	1.54%	1.75%
FIR Filter (UF=7)	Area (gates)	Power (μ w)	5448.00	53.93	5492	54.44	5494	54.45	5496	54.47	5498	54.48
	Area Ovh.	Power Ovh.	--	--	0.81%	0.94%	0.84%	0.96%	0.88%	0.99%	0.92%	1.02%
Blur Filter	Area (gates)	Power (μ w)	2980.00	28.94	3024	29.44	3026	29.46	3028	29.48	3030	29.49
	Area Ovh.	Power Ovh.	--	--	1.48%	1.74%	1.54%	1.80%	1.61%	1.85%	1.68%	1.90%

Results and Analysis

This table demonstrates the ability of the proposed HLS-aided T-HAC Trojan framework to evade several existing detection techniques.

Evasion Capability of Proposed T-HAC Trojan Against Known detection Techniques [1] [5] [6]

(Note: U^{OG} and U^{DP} indicate original and duplicate units from [1], and '⊗' means 'Not detected')

Detection approach	Features in proposed IP (Trojan free) - Design X (U^{OG})	Features in proposed IP (with T-HAC Trojan) - Design Y (U^{DP})	Justification of evasion	Trojan Status
Trojan detection and recovery [5]	U^{OG} (4-point DCT): allocated to 3PIP vendor 1 U^{OG} o/p = 16	U^{DP} (4-point DCT): allocated to 3PIP vendor 2 U^{DP} o/p = 16	Difference in functional o/p between designs X & Y using comparator: Nil	⊗
	Sample test vectors for U^{OG} : {a=2, b=2, c=2, d=2, e=2, f=2, g=2, and h=2}	Sample test vectors for U^{DP} : {a=2, b=2, c=2, d=2, e=2, f=2, g=2, and h=2}		
3PIP Defence [6]	Baseline design o/p = 30 (For a specific test pattern o/p is reported. Exhaustive test patterns were tested.)	Trojan infected design o/p = 30 (For a specific test pattern o/p is reported. Exhaustive test patterns were tested.)	Difference in functional o/p between designs X & Y using comparator: Nil	⊗
	Sample test vectors for U^{OG} : {a=1, b=1, c=2, d=2, e=3, f=3, g=4, and h=4}	Sample test vectors for U^{DP} : {a=1, b=1, c=2, d=2, e=3, f=3, g=4, and h=4}		
TL-HLS [1]	Baseline design o/p = 28 (For a specific test pattern o/p is reported. Exhaustive test patterns were tested.)	Trojan infected design o/p = 28 (For a specific test pattern o/p is reported. Exhaustive test patterns were tested.)	Difference in functional o/p between designs X & Y using comparator: Nil	⊗
	Sample test vectors for U^{OG} : {a=1, b=2, c=3, d=4, e=4, f=3, g=2, and h=1}	Sample test vectors for U^{DP} : {a=1, b=2, c=3, d=4, e=4, f=3, g=2, and h=1}		

References

- 
- [1] 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 Transaction on Computer-Aided Design of Integrated Circuits and Systems*, Vol.36 (4), pp.655-668, 2017.
- [2] A. Sengupta et al., “M-HLS: Malevolent High-Level Synthesis for Watermarked Hardware IPs,” *IEEE Embedded Systems Letters*, vol. 16 (4), pp. 497-500, 2024.
- [3] A. Nahiyani, M. Tehranipoor, “Code Coverage Analysis for IP Trust Verification,” In: Mishra, P., Bhunia, S., Tehranipoor, M. (eds) *Hardware IP Security and Trust*. Springer, Cham., 2017.
- [4] Silvaco’s open cell library [Online], Available: <https://si2.org/open-cell-and-free-pdk-libraries/>, accessed on December 2025.
- [5] X. Cui, K. Ma, L. Shi and K. Wu, "High-level synthesis for run-time hardware Trojan detection and recovery," *2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC)*, San Francisco, CA, USA, 2014, pp. 1-6.
- [6] S. Rajmohan et al., “Hybrid Evolutionary Design Space Exploration Algorithm With Defence Against Third Party IP Vulnerabilities,” *IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 10, pp. 2602-2614, 2020.
- [7] M. Abderehman et al., “BLAST: Belling the Black-Hat High-Level Synthesis Tool,” *IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 11, pp. 3661-3672, 2022.
- [8] R. Yasaei et al., “Hardware Trojan Detection Using Graph Neural Networks,” *IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems*, vol. 44 (1), pp. 25-38, 2024.
- [9] Yier Jin and Y. Makris, "Hardware Trojan detection using path delay fingerprint," *2008 IEEE International Workshop on Hardware-Oriented Security and Trust*, Anaheim, CA, 2008, pp. 51-57.
- [10] C. Pilato et al., “Black-Hat High-Level Synthesis: Myth or Reality?,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27 (4), pp. 913-926, 2019.

References (Contd.)

- 
- [11] A. Sengupta, "Hardware Security of CE Devices [Hardware Matters]," in *IEEE Consumer Electronics Magazine*, vol. 6, no. 1, pp. 130-133, Jan. 2017.
- [12] A. Sengupta and D. Roy, "Antipiracy-Aware IP Chipset Design for CE Devices: A Robust Watermarking Approach," *IEEE Consumer Electronics Magazine*, vol. 6, no. 2, pp. 118-124, April 2017.
- [13] I. Hong and M. Potkonjak, "Behavioral synthesis techniques for intellectual property protection," *Proceedings 1999 Design Automation Conference (Cat. No. 99CH36361)*, New Orleans, LA, USA, 1999, pp. 849-854.
- [14] Sengupta, Anirban & Chourasia, Vishal & Bhui, Nabendu. (2025). "HLS Trojan Classification and Its Detection Techniques for CAD-Based IP Designs". *IEEE IT Professional*. 27. 66-74. 10.1109/MITP.2025.3584043.
- [15] M. Rathor and A. Sengupta, "Revisiting Black-Hat HLS: A Lightweight Countermeasure to HLS-Aided Trojan Attack," in *IEEE Embedded Systems Letters*, vol. 16, no. 2, pp. 170-173, June 2024.
- [16] J. J. Rajendran, O. Sinanoglu and R. Karri, "Building Trustworthy Systems Using Untrusted Components: A High-Level Synthesis Approach," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no. 9, pp. 2946-2959, Sept. 2016.
- [17] A. Sengupta, A. Anshul, and R. Chaurasia, "Exploration of optimal functional Trojan-resistant hardware intellectual property(IP) core designs during high level synthesis," *Microprocess. Microsyst.*, vol. 103, Nov. 2023.
- [18] A. Sengupta, V. Chourasia, N. Bhui and R. Mandi, "T-HAC: Time-Sensitive HLS Trojan Attack for Affecting Controller Timing Signal of Hardware IP Designs," in *IEEE Embedded Systems Letters*, doi: 10.1109/LES.2026.3696630



Thank you