17th IEEE Dallas Circuits and Systems Conference
April 19-21, Richardson, TX

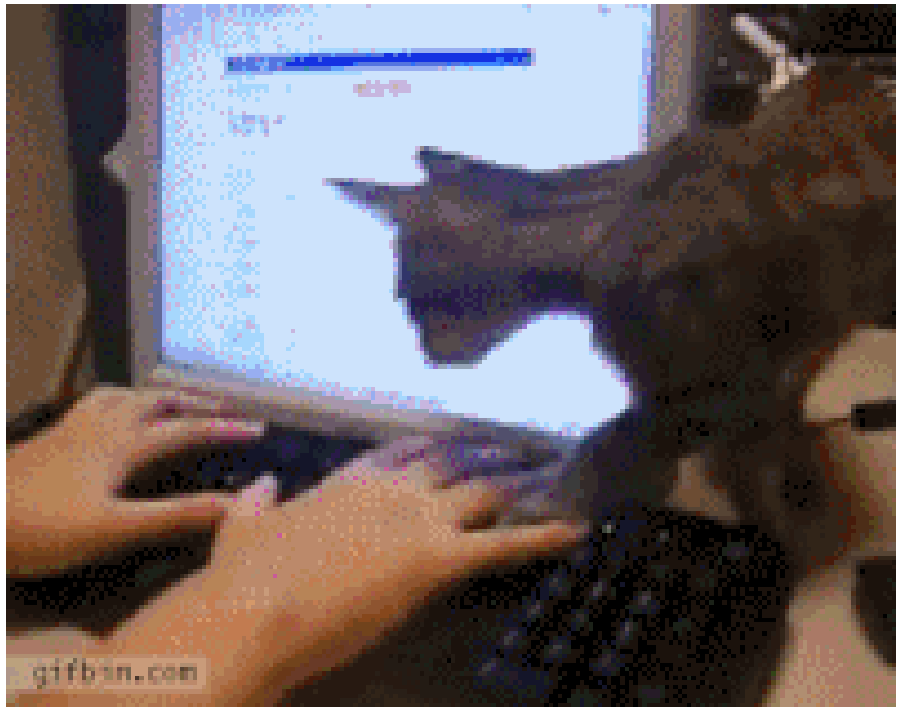
Reconfigurable Run-Time Hardware Trojan Mitigation for Logic-Locked Circuits

Jordan Maynard & Amin Rezaei

California State University Long Beach

Overview

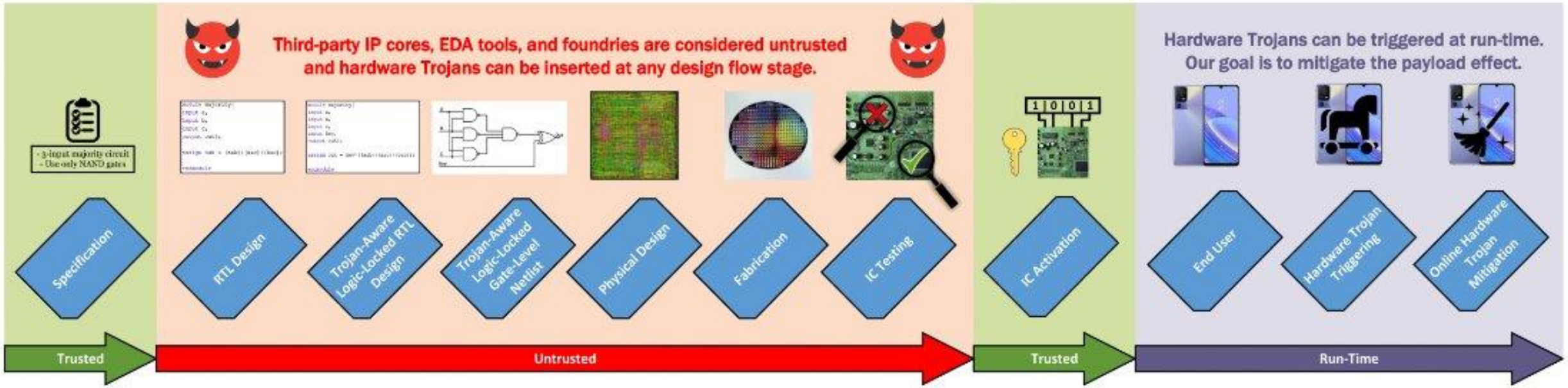
- Introduction
  - Problem Statement
  - Threat Model
  - Contributions
- Related Works
- Methodology
  - Choosing States
  - Cloning States
- Experimental Results
  - Security
  - Overhead
- Conclusion



Computer Attack Thwarter (CAT)

Introduction

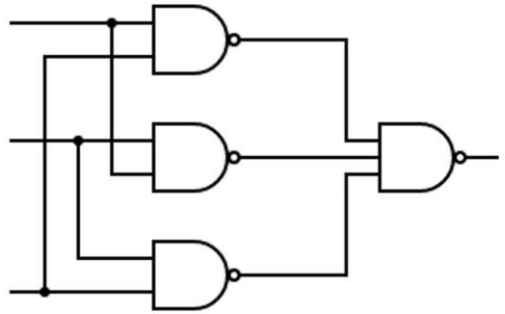
- Security threats exist throughout the IC design flow
- Hardware Trojans – targeted malicious defects
  - Rare triggers cause activation
  - Payloads become active once triggered



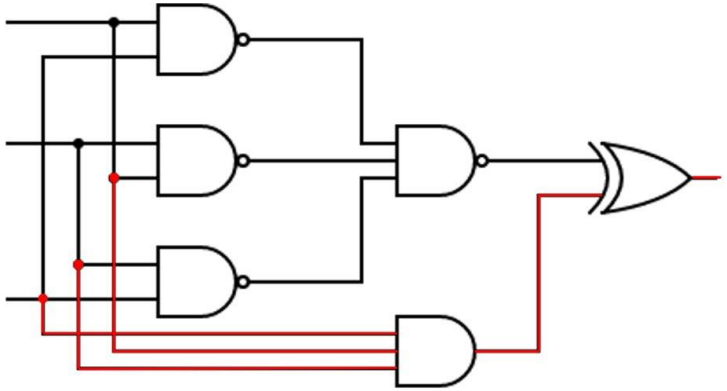
Threat Model

- We assume a **zero trust** environment
- Untrusted sources include:
  - Third-party IP vendors
  - EDA Tools
  - Manufacturing foundries
  - IC testing entities
- Trojan types we aim to defend against:
  - Data leakage
  - Functional modification
  - DoS

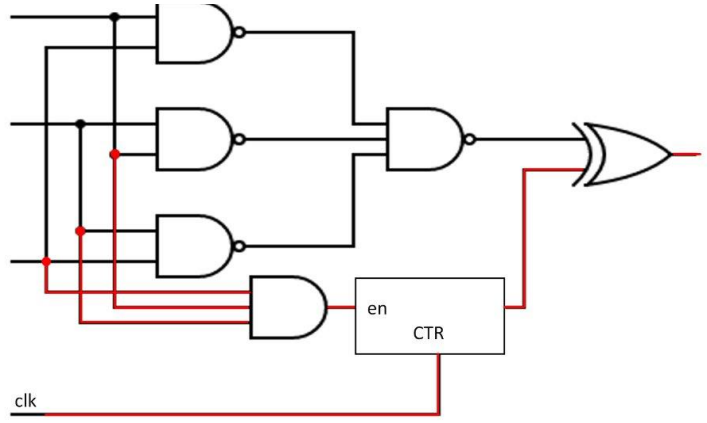
Majority Circuit



Majority Circuit w/ Combinational Trojan



Majority Circuit w/ Sequential Trojan



Problem Statement

$$\exists k^* \in K : g(X, S, k^*) \equiv f(X, S) \tag{1}$$

$$\exists x_{HT} \in X, \exists s_{HT} \in S, \exists k_{HT} \in K : \\ g_{HT}(x_{HT}, s_{HT}, k_{HT}) \not\equiv g(x_{HT}, s_{HT}, k_{HT}) \tag{2}$$

$$\exists K^* \subset K : \\ \forall k^* \in K^* : g(X, S, k^*) \equiv f(X, S) \\ \exists k^*_{HT} \in K^* : g_{HT}(X, S, k^*_{HT}) \equiv f(X, S) \tag{3}$$

DETECTION

VS

MITIGATION

Seeks out HT in design

Invasive and destructive

IC scrapped and remanufactured

Trojan insertion prevented

Circumvents trigger activation or payload

IC usable and secure

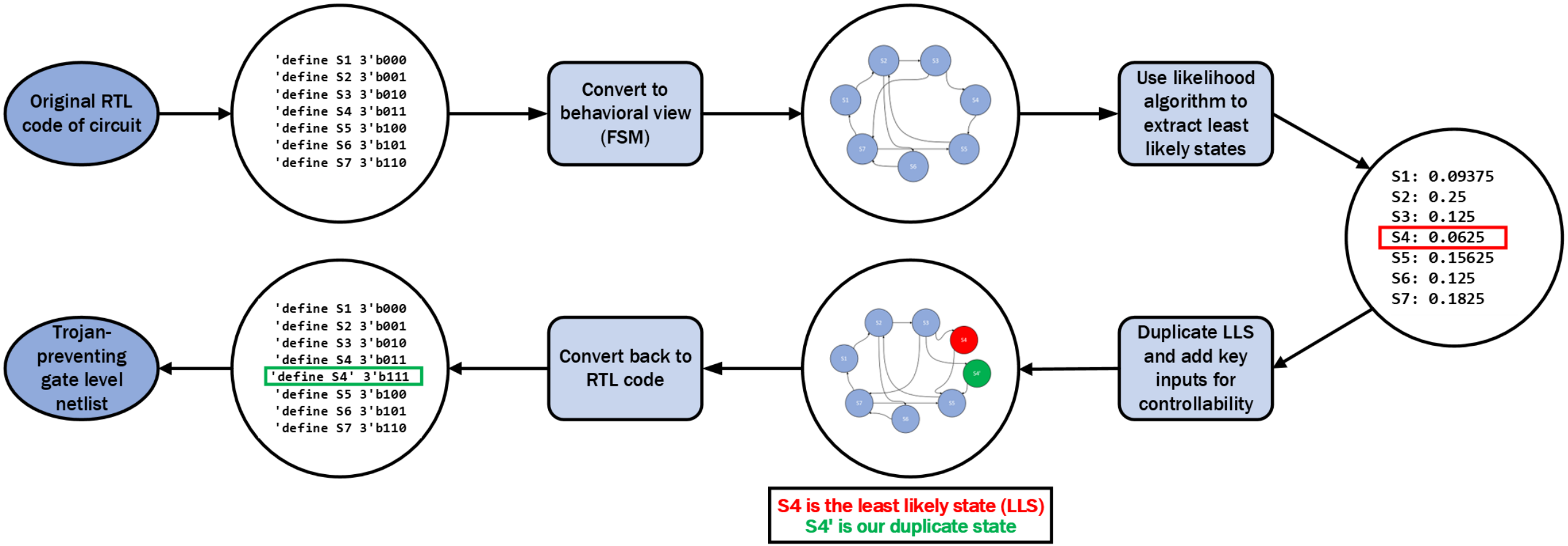
Related Works

Approach	Mitigation Method	Defends Against	Susceptible To
Hardware Sandboxes	Untrusted components are given indirect access to system resources through "sandboxes"	Trojans isolated in untrusted components	Zero-trust environment
Run-Time Mitigation in NoC	Bit shuffling encoders and decoders prevent malicious packet modification	Data leakage, Some DoS	Advanced DoS, Functional modification
Hardware-based Software Obfuscator	Software instructions are replaced by equivalent instructions to avoid Trojan triggers	Data leakage, Some DoS	Advanced DoS, Functional modification
Mitigation in MPSoC	Detects Trojan activity and reroutes around infected components with backup components	Data leakage, DoS, Functional modification	High manufacturing cost and PPA overhead
Hardware Enlightening	Increases controllability and observability of rare nets to prevent Trojan stealthiness	Stealthy combinational Trojans	Stealthy sequential Trojans
Evolutionary Encryption and Mitigation	Increases controllability and observability of rare nets using an evolutionary algorithm	Stealthy combinational Trojans	Stealthy sequential Trojans
Hardware and Software Fault Tolerance	Increases fault tolerance by introducing component redundancy to the architecture	Functional modification, DoS	Data leakage, High manufacturing cost and PPA overhead
Reconfigurable Run-Time Mitigation (this work)	Architectural redundancies with reconfigurable reachability to circumvent the Trojan payload	All Trojans with stealthy triggers	N/A

Our Contributions

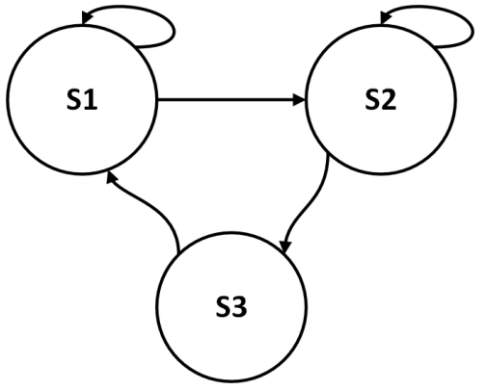
- Defining security in terms of reconfigurability.
- Algorithmic identification and duplication of Trojan-likely states from behavioral circuit specifications.
- A novel run-time mitigation approach for any logic-locked sequential circuit.
- Security gain and overhead results via implementations on several FSM benchmark circuits.
- Publicly available scripts to find state reachability, duplicate states, and insert trojans.

Methodology



Choosing States

1.



2.

Deterministic	S1	S2	S3
S1	1	1	0
S2	0	1	1
S3	1	0	0

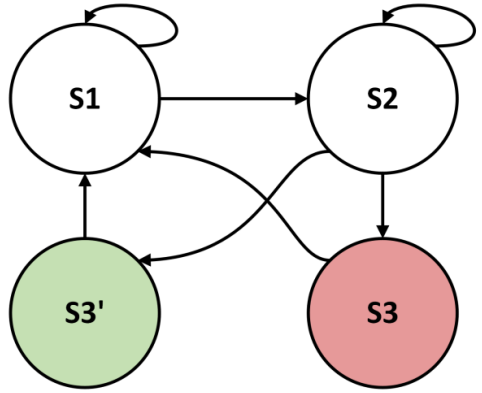
3.

Probabilistic	S1	S2	S3
S1	0.5	0.5	0
S2	0	0.5	0.5
S3	1	0	0

4.

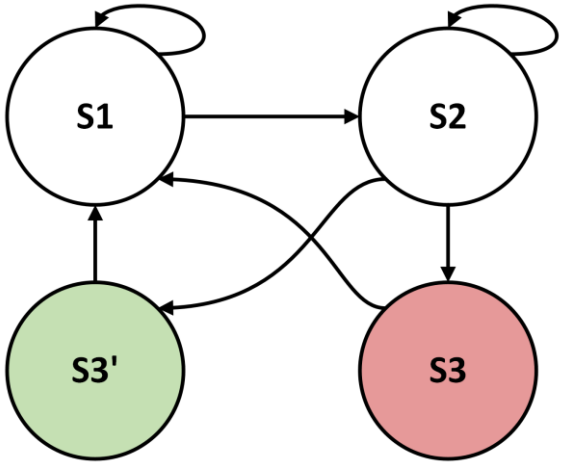
Steady State	S1	S2	S3
P	0.4	0.4	0.2

5.



Cloning States

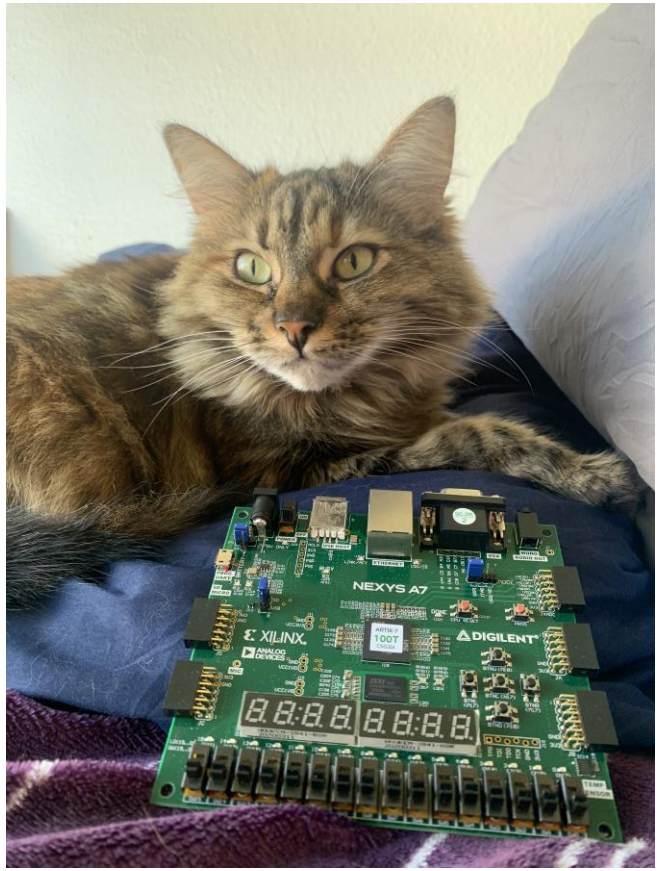
- Key bit is added to choose between original and cloned state logic.
  - Prevents merging during logic optimization.
- Trojan payload is avoided by using different logic.
- Least-likely cloned states become less-likely under equivalent operation.



Steady State	S1	S2	S3	S3'
P	0.4	0.4	0.1	0.1

Experimental Results

- 43 benchmarks were tested with our mitigation method:
  - Synthezza – 41 FSM benchmarks
  - Custom example FSM
  - ITC'99 - b02



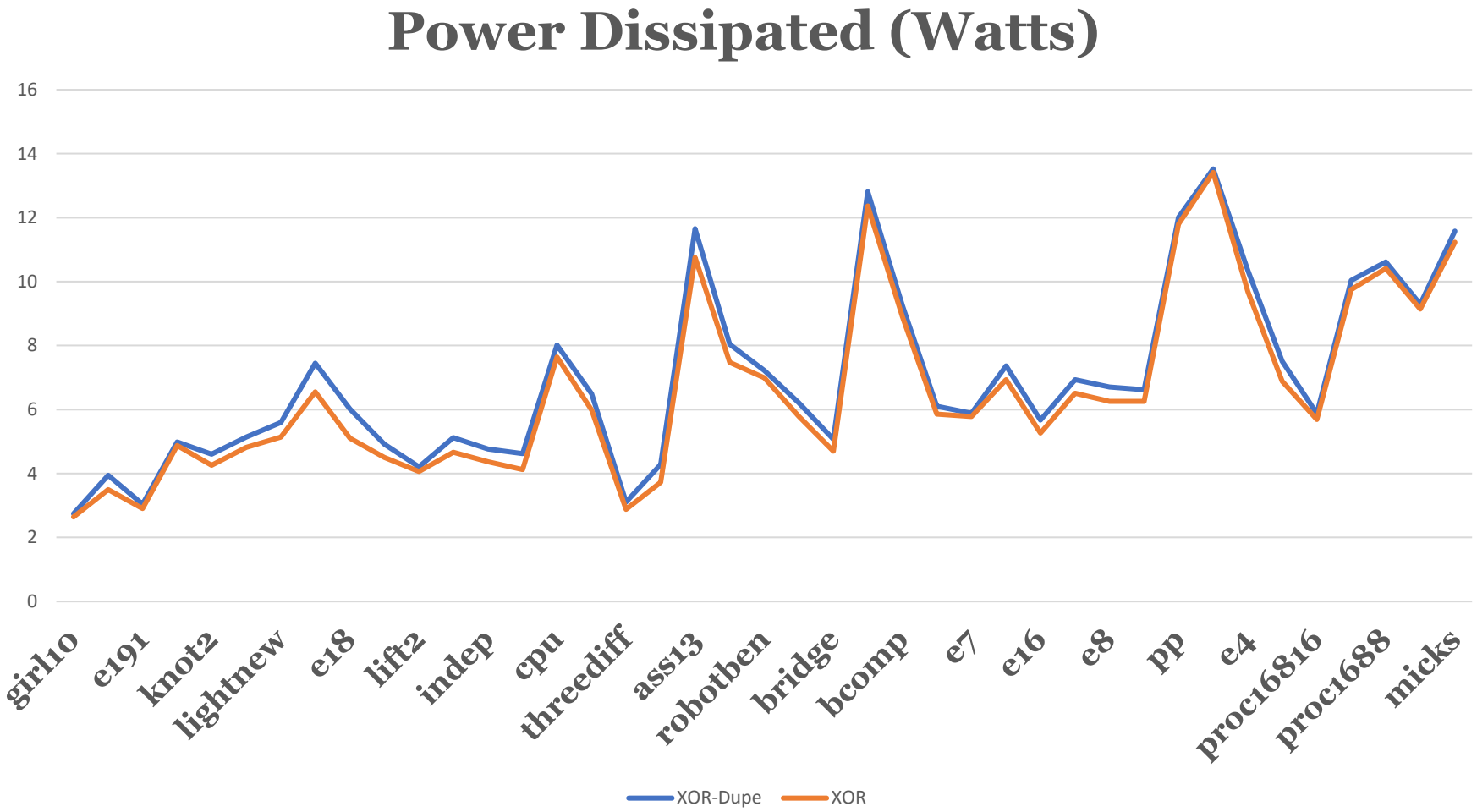
Unpaid research assistant collecting data

Benchmarks

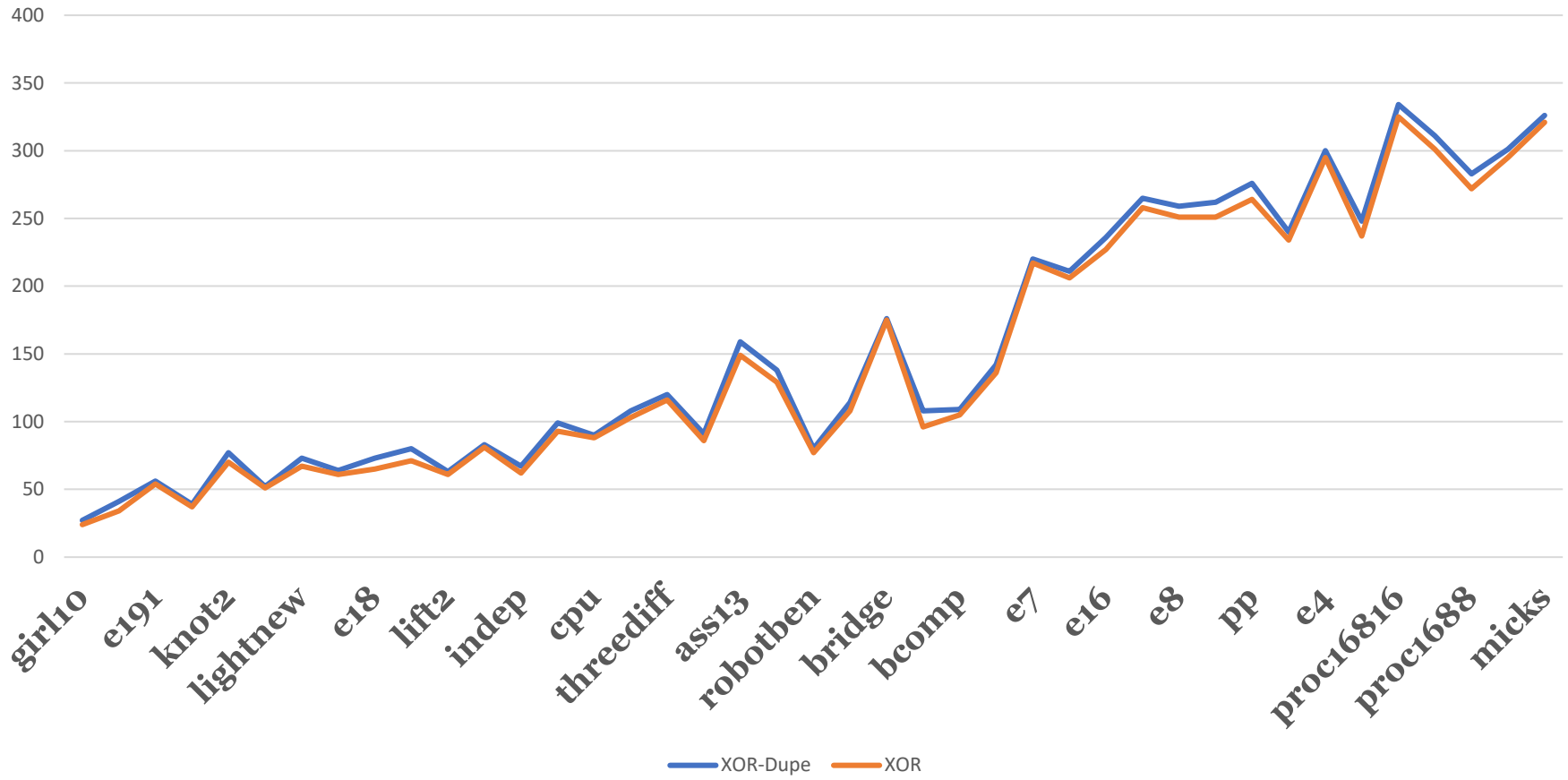
Benchmark	Acronym	Description
Original	ORG	Used as oracle for security metrics and baseline for power and utilization overhead metrics.
Duplicated	DUPE	Low-controllable states are duplicated with added key input to choose between state. Functionality is equivalent to ORG under all the keys.
Duplicated Trojan-Inserted	DUPE-TI	Same as DUPE, but out of every two duplicated states, one has inverted outputs or wrong state transitions. Functionality is equivalent to ORG only under one correct key.
Duplicated Trojan-Inserted XOR-Locked	DUPE-TI-XOR	Same as DUPE-TI, but an added XOR lock showcases its ability to be used in compound locking. Functionality is equivalent to ORG only under one correct key.
Duplicated Trojan-Inserted Secure-Locked	DUPE-TI-SEC	Same as DUPE-TI, but an added SAT-secure lock showcases its ability to be used as multi-objective security. Functionality is equivalent to ORG only under a pair of keys.

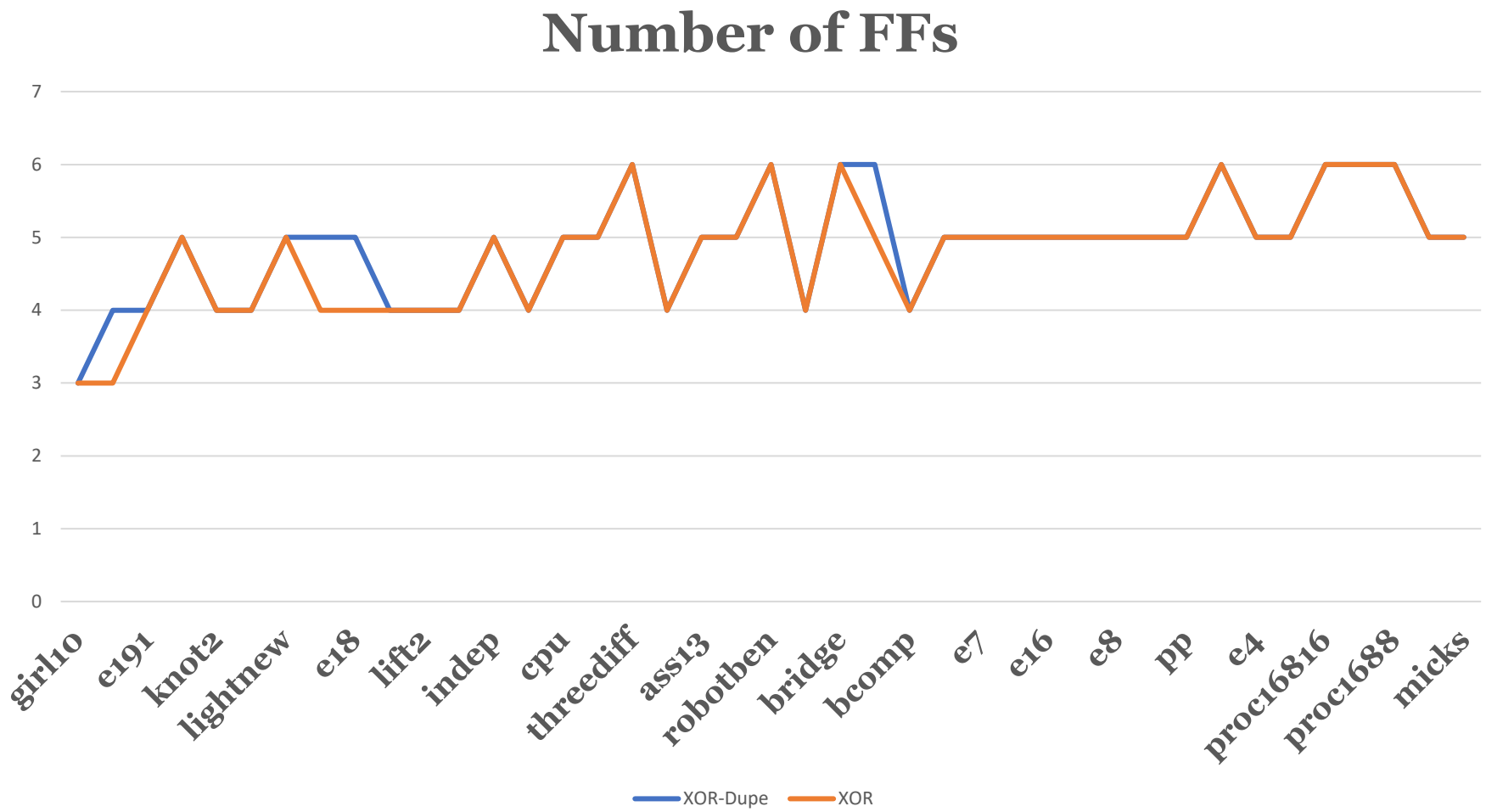
NEOS Attack Results

Benchmark	Key	DUPE	DUPE-TI	DUPE-TI-XOR	DUPE-TI-SEC
Small: <100 LUTs custom, b02, girl10, robm, sortmax, e191, lightnew, ex6, knot2, cat, e18, e17, lift2, e161, indep, cpu	Key Size	3 bits	3 bits	13 bits	I: 13 bits, F: 13 bits
	Reported Key	Correct	Correct	Correct	No Result
Medium: >=100 & <250 LUTs lift, checker9, ass13, e10, e16, robotben, bridge, bech, dmac, bcomp, pilot, e7, lcu, e2	Key Size	3 bits	3 bits	13 bits	I: 13 bits, F: 13 bits
	Reported Key	Correct	Correct	Correct	No Result
Large: >= 250 LUTs e8, e15, pp, v16, e4, sara, proc81616, proc16816, proc1688, max, micks	Key Size	5 bits	5 bits	15 bits	I: 15 bits, F: 15 bits
	Reported Key	Correct	Correct	Correct	No Result


Power Dissipated (Watts)
16
14
12
10
8
6
4
2
0
girl10
e191
knot2
lightnew
e18
lift2
indep
cpu
threediff
ass13
robotben
bridge
bcomp
e7
e16
e8
pp
e4
proc16816
proc1688
micks
XOR-Dupe
XOR

Number of LUTs




Number of FFs
7
6
5
4
3
2
1
0
girl10
e191
knot2
lightnew
e18
lift2
indep
cpu
threediff
ass13
robotben
bridge
bcomp
e7
e16
e8
pp
e4
proc16816
proc1688
micks
XOR-Dupe
XOR

Conclusion & Future Works

- Our mitigation method provides tunable reconfigurability-based security against stealthy hardware Trojans in sequential designs.
- Experiments confirm this method to be effective when combined with secure logic locking to provide multi-objective security.
- In future works, structural realization of our behavioral method can be pursued to apply this method directly to gate-level netlists.