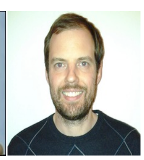
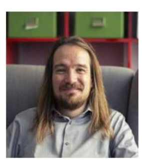
# High-Level Approaches to Hardware Security

Ramesh Karri
Polytechnic School of Engineering
rkarri@nyu.edu
http://cyber.nyu.edu

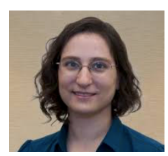
## http://cyber.nyu.edu/


H. AlKhzaimi, AD, Crypto

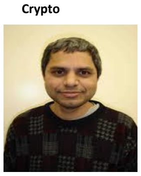

J. Cappos, Tandon, Sys Security

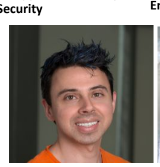

B. Dolan-Gavitt, Tandon, Emb. Security


S. Garg, Tandon, H/W Security


R. Greenstadt, Tandon, Security


R. Milch, Law, Security


R. Karri, Tandon, H/W Security


D. Mccoy, Tandon, Security &Privacy


M. Maniatakos, AD, H/W Security


N.Memon, Tandon, Forensics, Security


R. Song, Biochip Security


O. Nov, MOT, Security


C. Popper, AD, Wireless Security


S. Raskoff, Law


K. Ross, Tandon, Soc Networks Privacy


O. Sinanoglu, AD, H/W Security


Q. Zhu , Tandon, Game theory


M. Rasras, AD, Photonics

## Mission

NEW YORK UNIVERSITY

NYU Center for Cybersecurity (CCS) is an interdisciplinary center dedicated to

- Research technical and other means to secure the cyber infrastructure
- Educate the next generation of cybersecurity professionals and
- Shape public discourse on the policy and legal aspects of cybersecurity.

NYU TANDON SCHOOL OF ENGINEERING Polytechnic Institute

جامعة نيويورك ابوظبي
NYU ABU DHABI

NYU LAW

NYU ROBERT F. WAGNER GRADUATE SCHOOL OF PUBLIC SERVICE

CENTER FOR CYBER SECURITY

NYU STEINHARDT

NYU JOURNALISM ARTHUR L. CARTER JOURNALISM INSTITUTE

NYU STERN

## NYU has a Reputation in Cyber Sec

NEW YORK UNIVERSITY

- One of the earliest to offer degrees in Cyber Security (circa 1998)
- Triple distinction
  - NSA Center of Excellence in Information Assurance Education
  - NSA Center of Excellence in Information Assurance Research
  - NSA Center of Excellence in Cyber Operations
- MS in Cybersecurity (Cyberscholars for US residents)
- MS in Cyberrisk
- Bridge to Cyber
- Significant funding for research/education over 10 years.
- Scholarship for Service
  - Strong research and training partnership with federal agencies.
  - Placed over 100 students in all agencies of the Govt.
- Signature programs and partnerships.

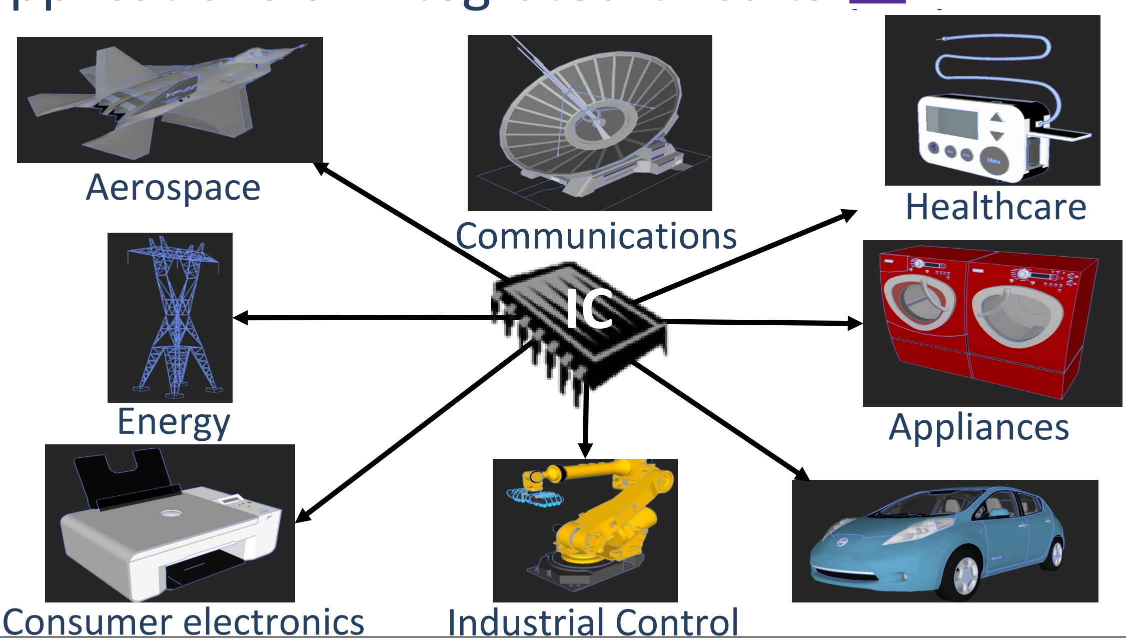
# Signature Outreach Programs

NEW YORK UNIVERSITY

- Cyber Security Awareness Week (CSAW)
  - Celebrating its 14th year
  - Largest student cyber competition in US
  - Largest Capture the Flag
  - 20,000+ HS and college students
  - CTF, ESC, Best paper, High school forensics,…
  - MENA(NYU-AD), India (IIT Kanpur), Europe (Valence France and Israel (U. Haifa)
- Summer Cyber Boot-camp High School STEM Educators
- Sloan Speaker Series
- Hackers in Residence from Industry
- Hosts NSF/NSA CyberCorps Program ~ 100 in government service

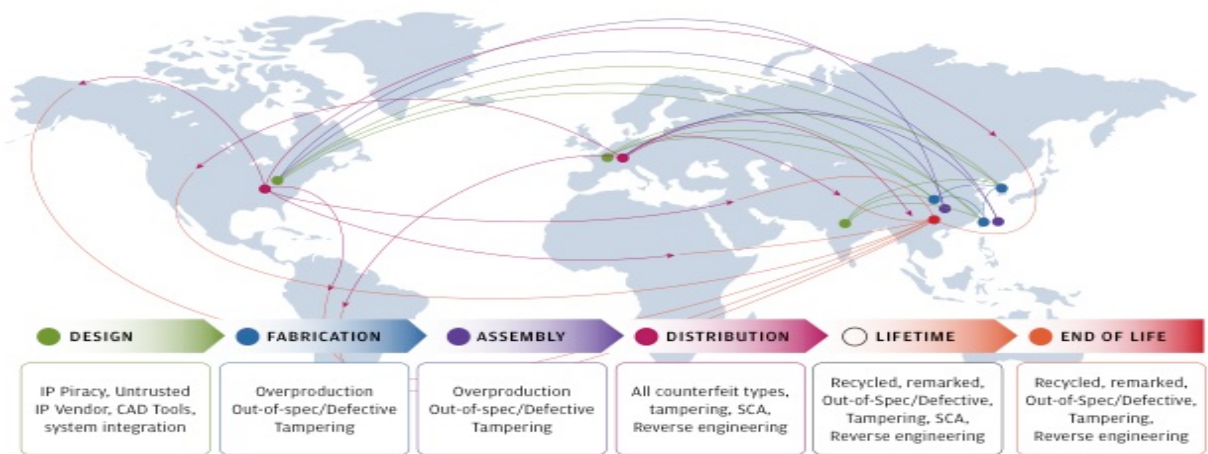
# Applications of Integrated Circuits

NEW YORK UNIVERSITY



## Securing Electronics Supply Chain

NEW YORK UNIVERSITY

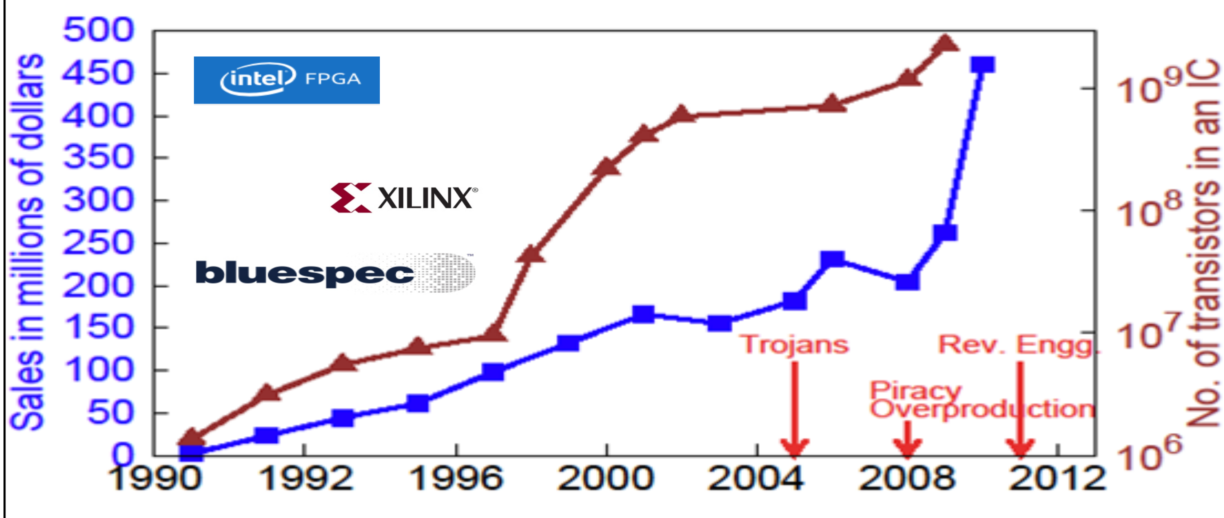






## High-Level Synthesis (HLS/ESL)

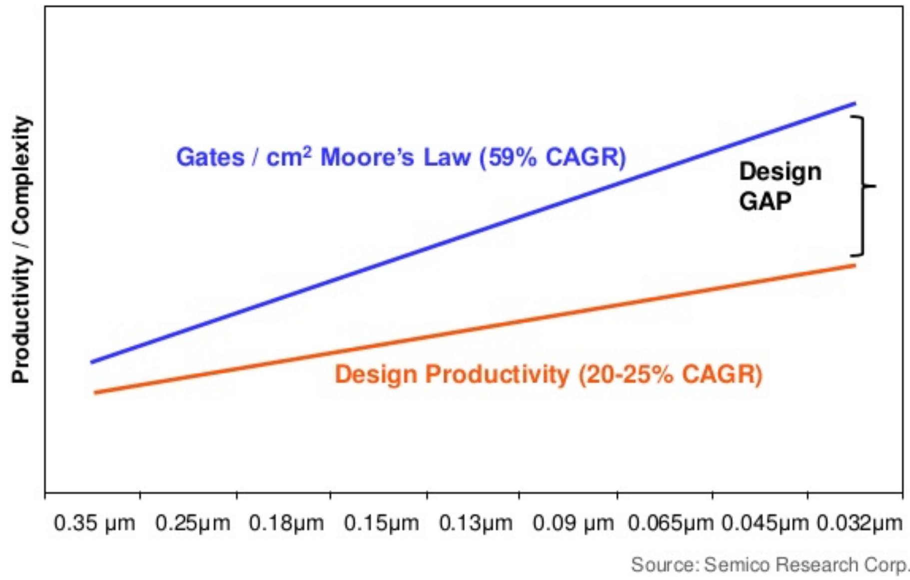
NEW YORK UNIVERSITY




CALYPTO ↔ Mentor Graphics ↔ Oasys FORTE Design Systems ↔ cadence

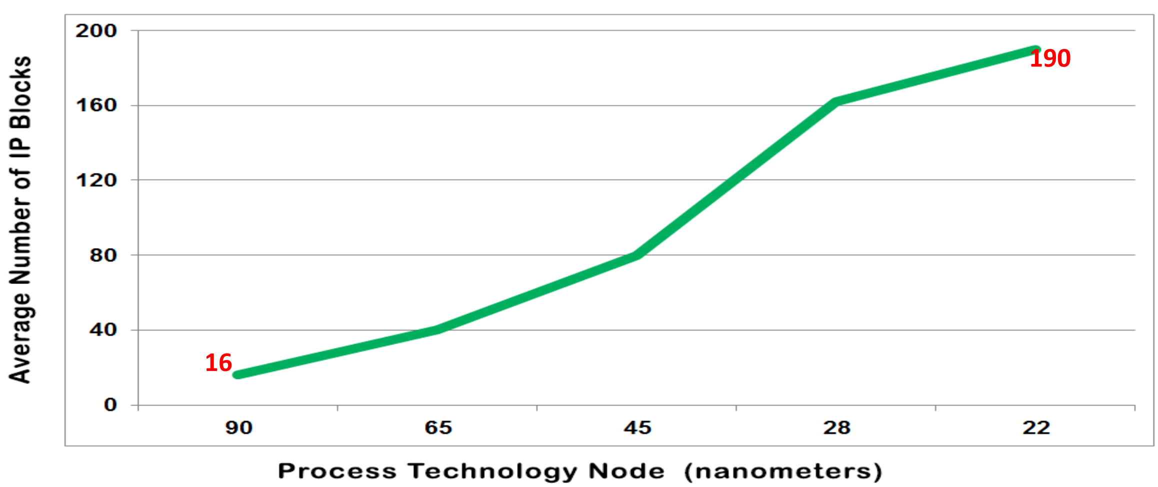
# HLS is a Productivity Tool

NEW YORK UNIVERSITY




# 3rd Party IPs in a Design

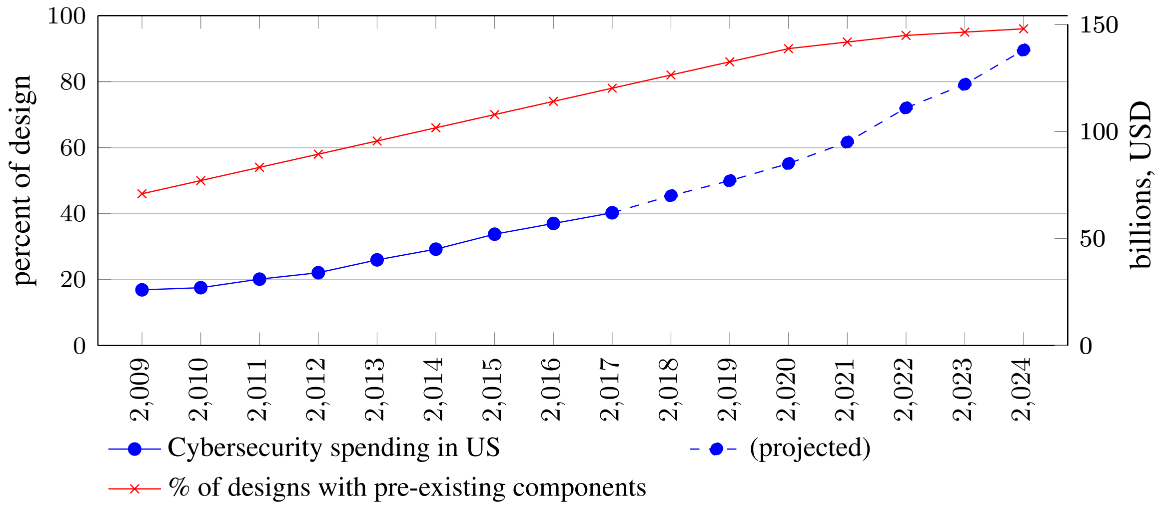
NEW YORK UNIVERSITY




(International Business Strategies, 2012)

# Accelerator-based Design!

NEW YORK UNIVERSITY

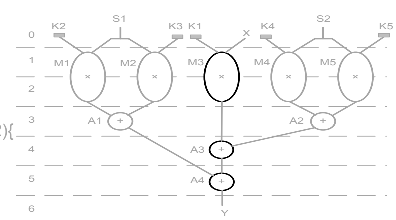



# HLS Design Flow

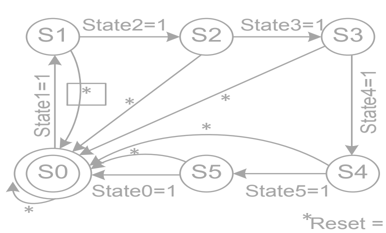
NEW YORK UNIVERSITY

```
int main (int X, int *Y, int *Z1, int *Z2 : num16) {
        int  in1 = (X * K1);
        Y = biquad(in1, K2, K3, K4, K5, *Z1, *Z2);
        return Y;
}
int biquad(int in, int a1, int a2, int b1, int b2, int *Z1, int *Z2){
        int state = in + (a1 × *Z1) + (a2 × *Z2);
        return state + (b1 × *Z1) + (b2 × *Z2);
}
```

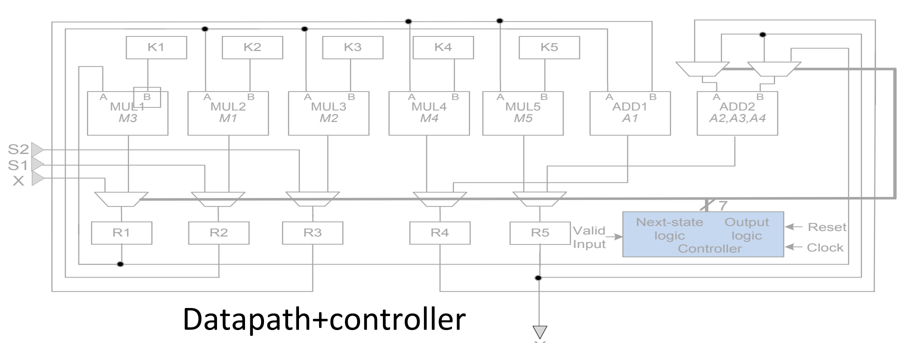
c-specification of biquad filter



Scheduling and binding



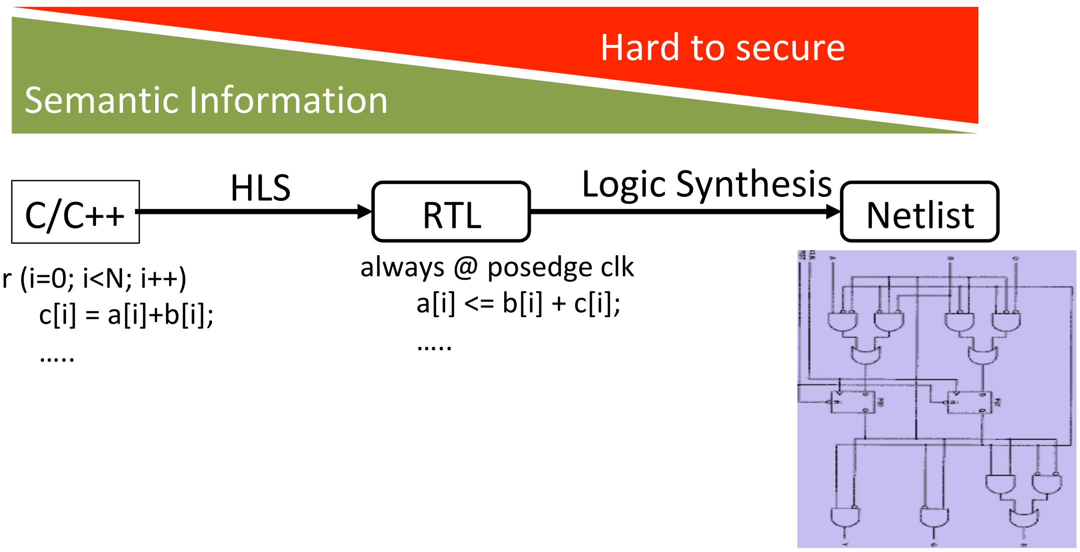

Finite state machine




Datapath+controller

# Security-Aware HLS

NEW YORK UNIVERSITY




# High-Level View of Security

NEW YORK UNIVERSITY

- Promising to add security constraints
- HLS in Hardware vs Programming Language/Compilers in Software
- Semantics: constants, operators, control flow, dependencies (sensitive IP)

| Hardware | Software |
|---|---|
| Algorithm-Level (HLS) | Programming Lang (Compiler) |
| RT Level | |
| Gate Level | Assembly (HEX) |
| Layout | Binary |

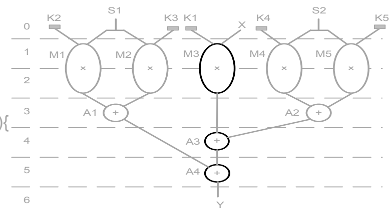
Hard to secure

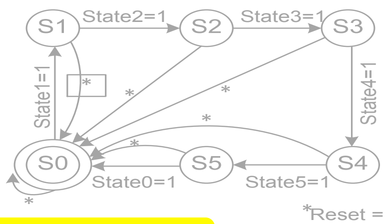
Semantic info

# HLS Design Flow

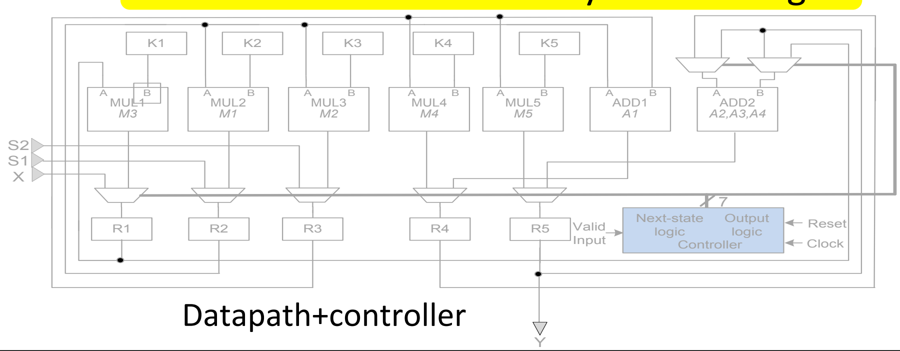
NEW YORK UNIVERSITY

```
int main (int X, int *Y, int *Z1, int *Z2 : num16) {
        int  in1 = (X * K1);
        Y = biquad(in1, K2, K3, K4, K5, *Z1, *Z2);
        return Y;
}
int biquad(int in, int a1, int a2, int b1, int b2, int *Z1, int *Z2){
        int state = in + (a1 × *Z1) + (a2 × *Z2);
        return state + (b1 × *Z1) + (b2 × *Z2);
}
```





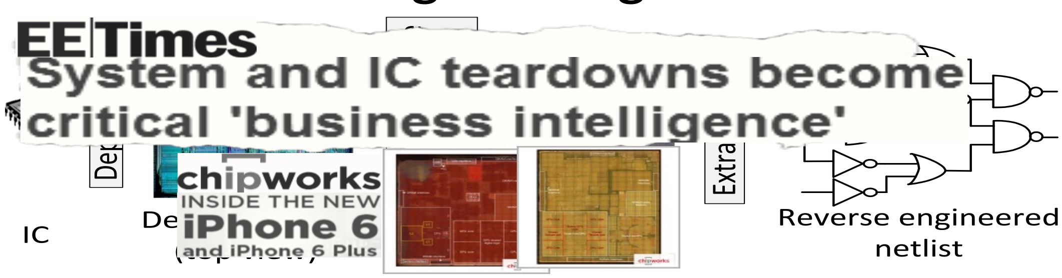

c-specification of biq Can ESL undermine security of the design? te machine




Datapath+controller

# Threat: Reverse Engineering

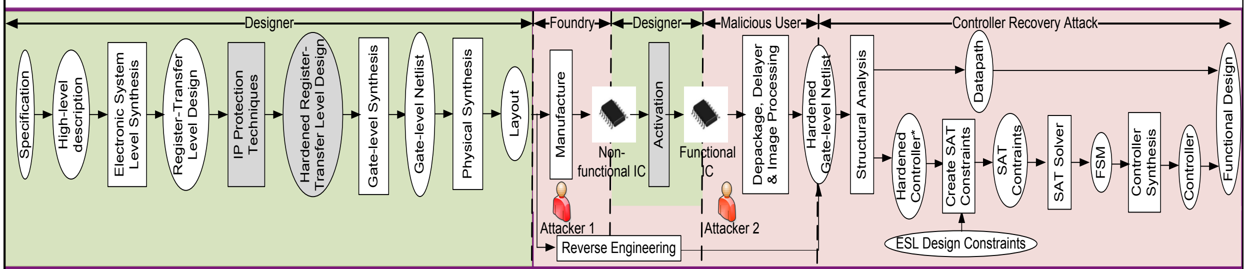
NEW YORK UNIVERSITY




- Legal: to detect piracy
  - Identify device technology, functionality, design
  - Chipworks
- Illegal: piracy, IP theft and Trojan insertion
  - Malicious user or  Malicious SoC integration house or Malicious foundry

# Attack: HLS-informed Rev. Engg.

NEW YORK UNIVERSITY

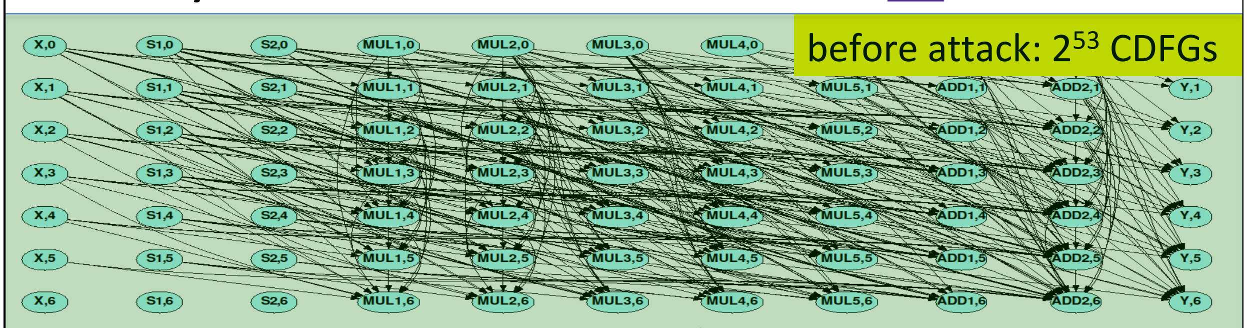


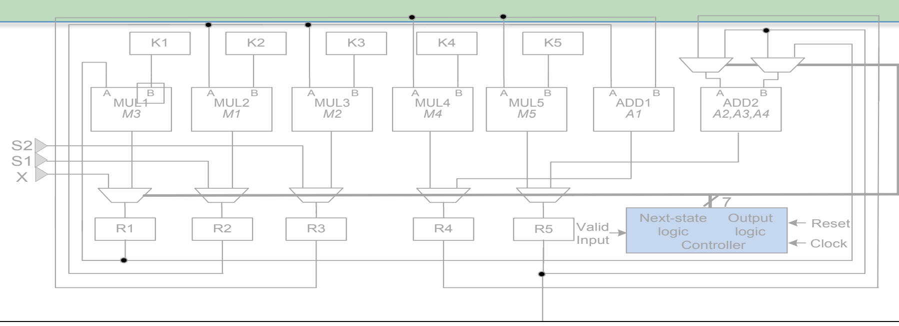

J. Rajendran, A. Ali, O. Sinanoglu and R. Karri, Belling the CAD: Toward Security-Centric Electronic System Design, IEEE Transactions on CAD, Vol 34, No. 11, pp. 1756-1769, November, 2015.

# Security Metric: # of CDFGs

NEW YORK UNIVERSITY

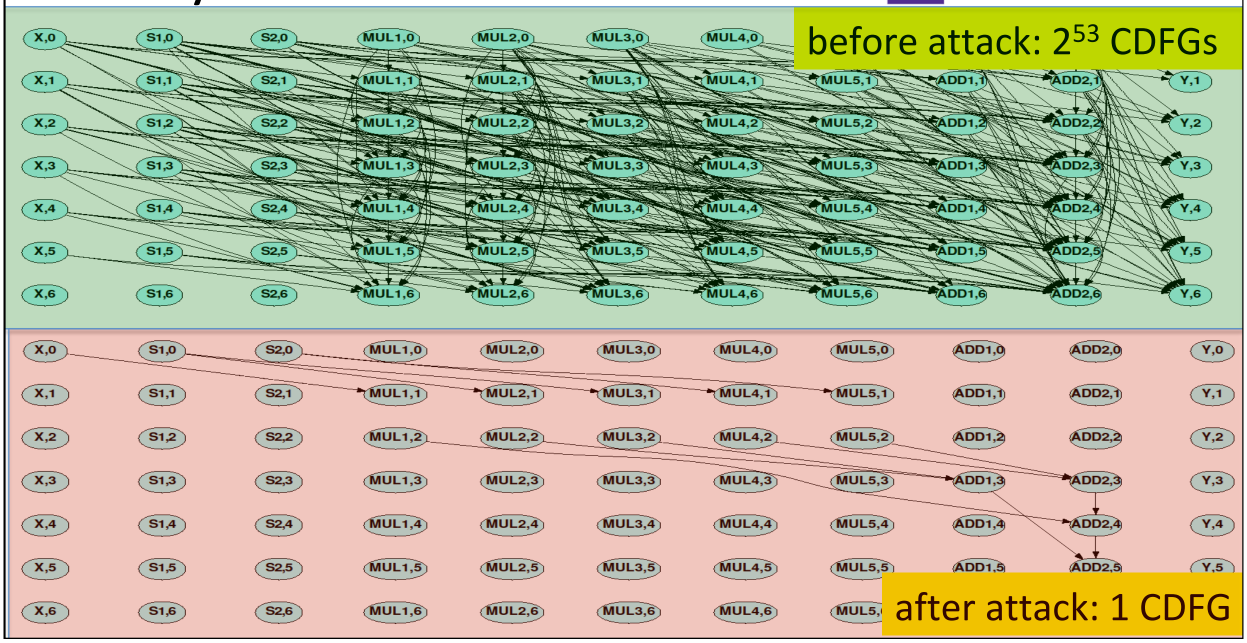
before attack: $2^{53}$ CDFGs





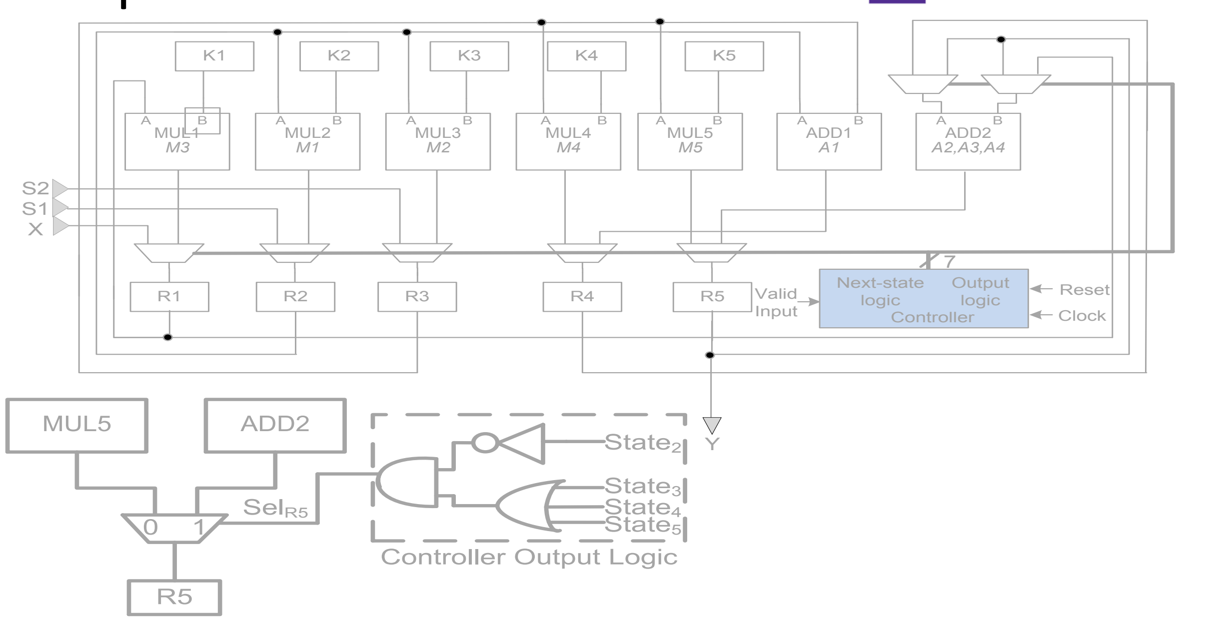
# Security Metric: # of CDFGs

NEW YORK UNIVERSITY



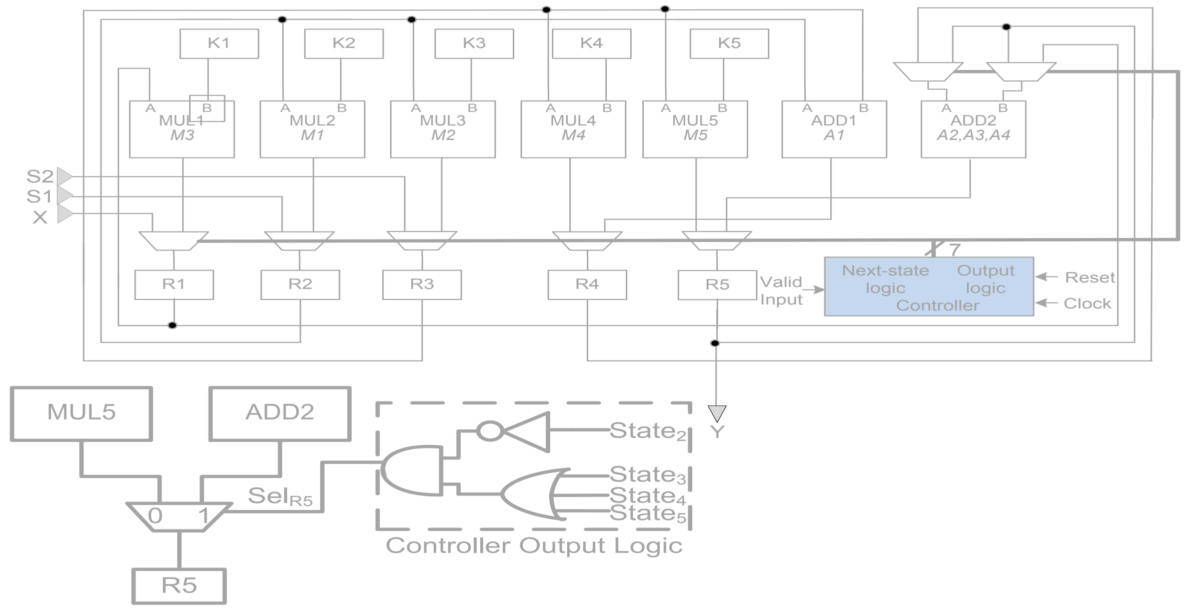

# Datapath Constraints

NEW YORK UNIVERSITY



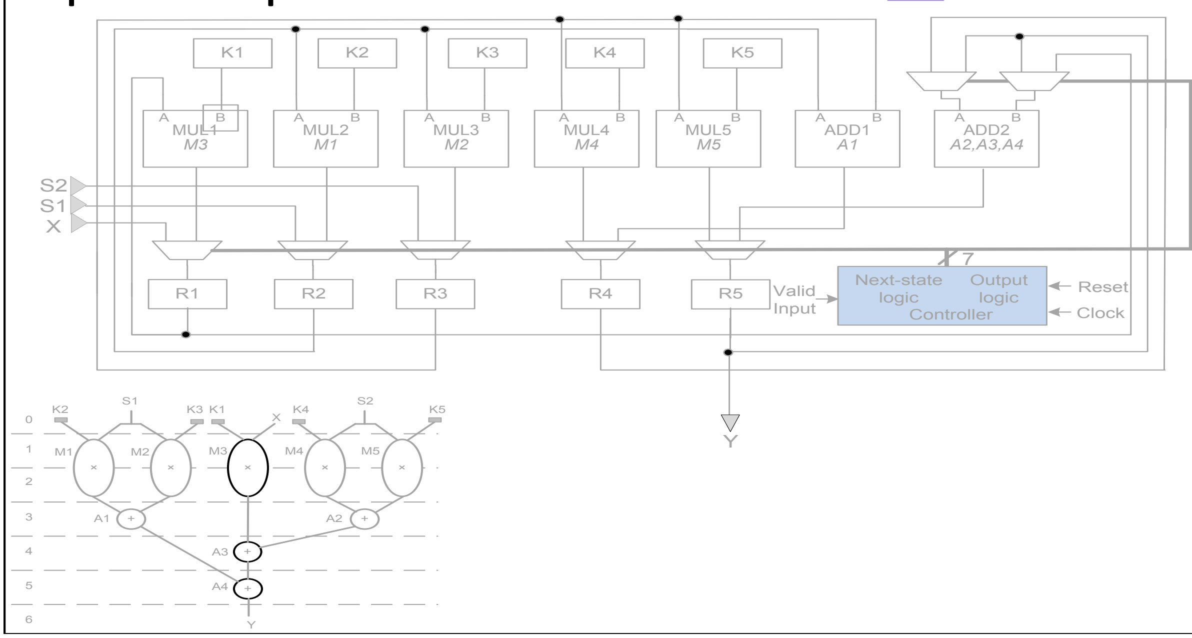
# Controller Constraints

NEW YORK UNIVERSITY




# Input-Output Constraints

NEW YORK UNIVERSITY



# Security Metric: # of CDFGs

NEW YORK UNIVERSITY

| Design | ESL Constraints | | | |
|---|---|---|---|---|
| | # 1 | # 1 – # 4 | # 1 – # 6 | # 1 – # 7 |
| BQF | $2^{53}$ | $2^{52}$ | $2^{33}$ | $2^{2}$ |
| Arai | $2^{246}$ | $2^{160}$ | $2^{118}$ | $2^{3}$ |
| Chem | $2^{3526}$ | $2^{717}$ | $2^{606}$ | $2^{4}$ |
| Dir | $2^{731}$ | $2^{160}$ | $2^{118}$ | $2^{3}$ |
| Feig_dct | $2^{3790}$ | $2^{606}$ | $2^{512}$ | $2^{4}$ |
| Honda | [illegible] | [illegible] | [illegible] | [illegible] |
| Lee | [illegible] | [illegible] | [illegible] | [illegible] |
| Mcm | $2^{716}$ | $2^{160}$ | $2^{118}$ | $2^{3}$ |
| Pr | $2^{319}$ | $2^{216}$ | $2^{160}$ | $2^{3}$ |
| Wang | $2^{321}$ | $2^{215}$ | $2^{160}$ | $2^{3}$ |
| Snow3g | $2^{383}$ | $2^{80}$ | $2^{53}$ | $2^{3}$ |
| Kasumi | $\geq 2^{1000000}$ | $2^{757749}$ | $2^{752363}$ | $2^{9}$ |
| MD5c | $\geq 2^{1000000}$ | $2^{722105}$ | $2^{717134}$ | $2^{9}$ |
| AES | $\geq 2^{1000000}$ | $2^{598662}$ | $2^{594179}$ | $2^{9}$ |

# of CDFGs reduce drastically using HSL constraints

# Belled the CAD!

NEW YORK UNIVERSITY

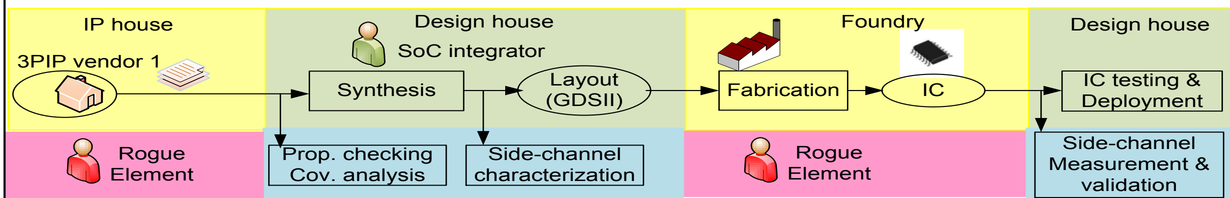
| Design | Tools A,B, C, D & E: Non-pipelined and Resource-Constrained | | | | |
|---|---|---|---|---|---|
| | Attack Success | | | Attack Cost | |
| | No. of compare points | % compare points matched | Equivalence checking | # of SAT literals | Time for solving SAT (s) |
| BQF | 16 | 100 | Pass | 1050 | 0.01 |
| Arai | 128 | 100 | Pass | 5166 | 0.02 |
| Chem | 240 | 100 | Pass | 2415264 | 43 |
| Dir | 128 | 100 | Pass | 131328 | 0.75 |
| Feig_dct | 1024 | 100 | Pass | 517545 | 5.17 |
| Honda | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| Lee | 128 | 100 | Pass | 10374 | 0.05 |
| Mcm | [illegible] | [illegible] | [illegible] | [illegible] | 0.35 |
| Pr | 128 | 100 | Pass | 12320 | 0.01 |
| Wang | 128 | 100 | Pass | 11520 | 0.04 |
| Snow3g | 32 | 100 | Pass | 27720 | 0.17 |
| Kasumi | 64 | 100 | Pass | 8090016 | 143 |
| MD5c | 128 | 100 | Pass | 2536050 | 32 |
| AES | 128 | 100 | Pass | 33353948 | 1321 |

All benchmarks reverse engineered in <30 minutes

Functionally equivalent and structurally identical!

# Threat: Malicious 3PIP (Trojans)

NEW YORK UNIVERSITY

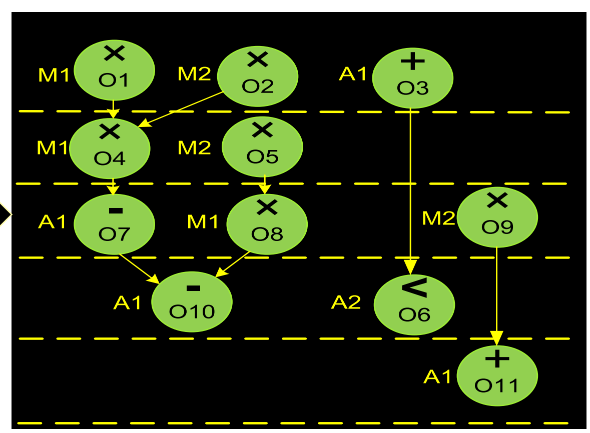



- 3PIP vendors are not trusted; may insert trojans
  - Trojans cause wrong outputs
  - Distributed: in different modules from same vendor may collude
- SoC integrator is trusted
  - SoC integrator uses components from 3PIP vendors
  - 3PIPs are integrated into a system and synthesized
- SoC is manufactured at an off-shore foundry
- The manufactured hardware is tested and deployed

# HLS-based Trojan Detection

NEW YORK UNIVERSITY

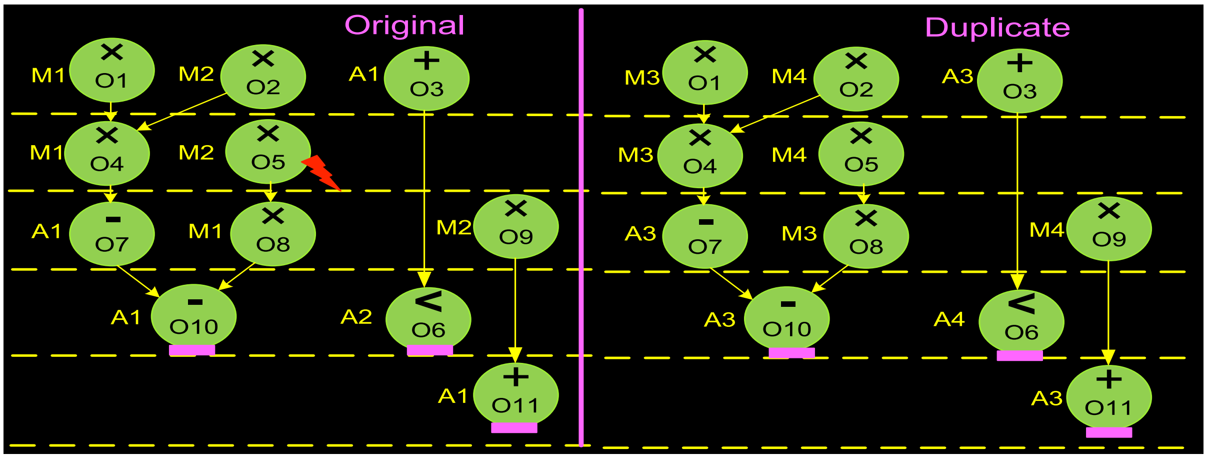
```
While (x < a) {
    x1 = x + dx
    u1 = u – 3xudx – 3ydx
    y1 = y + udx
    x = x1; u = u1; y = y1
    }
```




Control Data Flow Graph

# Detect "Natural" Faults

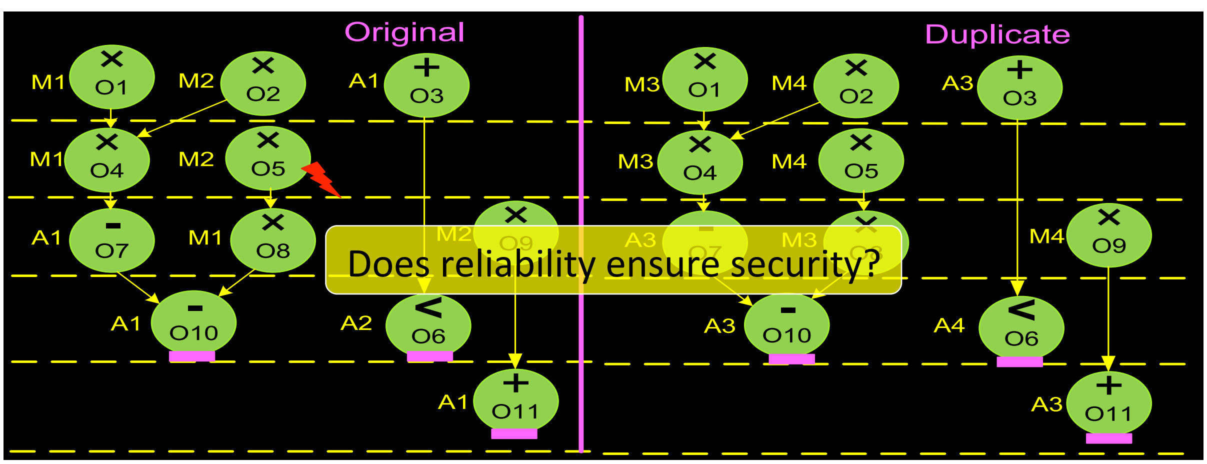
NEW YORK UNIVERSITY





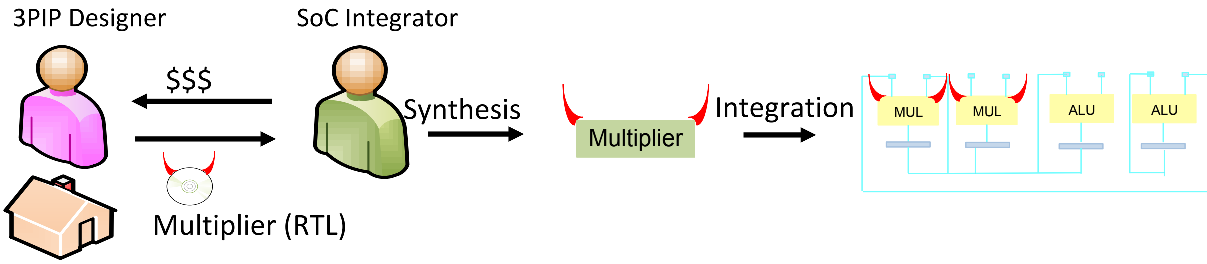
# Detect "Natural" Faults

NEW YORK UNIVERSITY

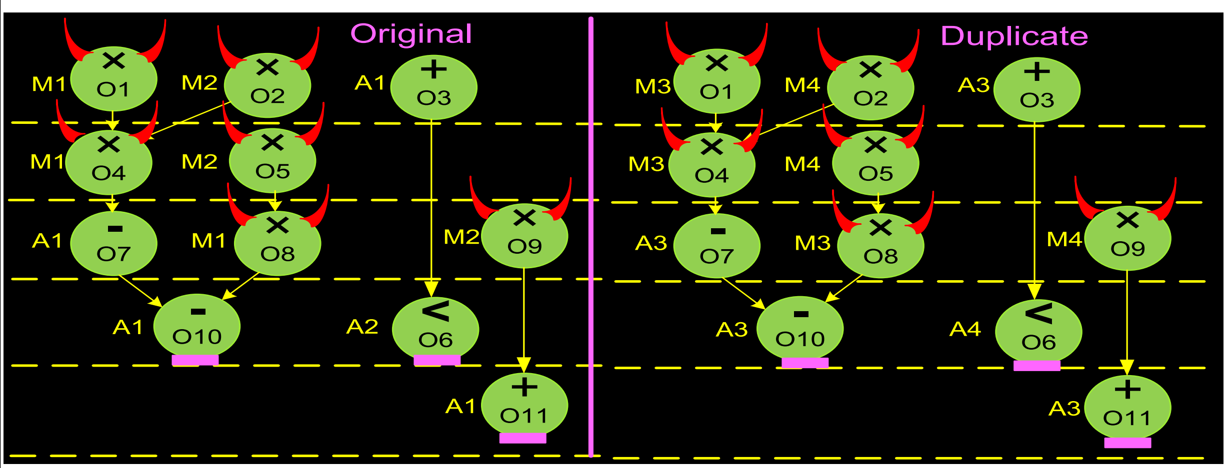
# Malicious 3PIPs

NEW YORK UNIVERSITY




# Detect 3PIPs: Duplicate and Check

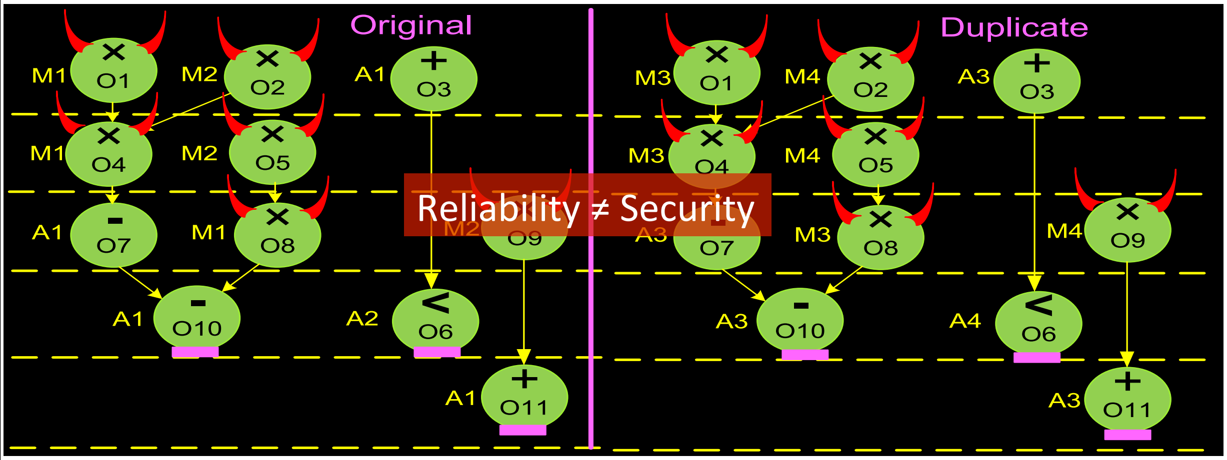
NEW YORK UNIVERSITY




- Components with Trojans produce same “malicious” outputs
- Checkers cannot detect malicious outputs
- Violates assumption for reliability

# Detect 3PIPs: Duplicate and Check

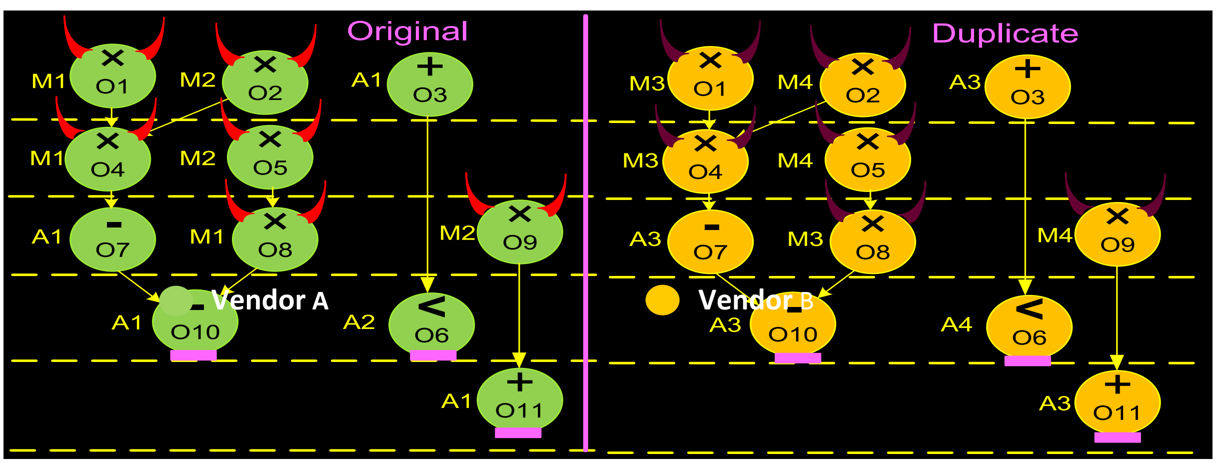
NEW YORK UNIVERSITY




- Components with Trojans produce same "malicious" outputs
- Checkers cannot detect malicious outputs
- Violates assumption for reliability

# Detect 3PIPs: Duplicate+Diversify

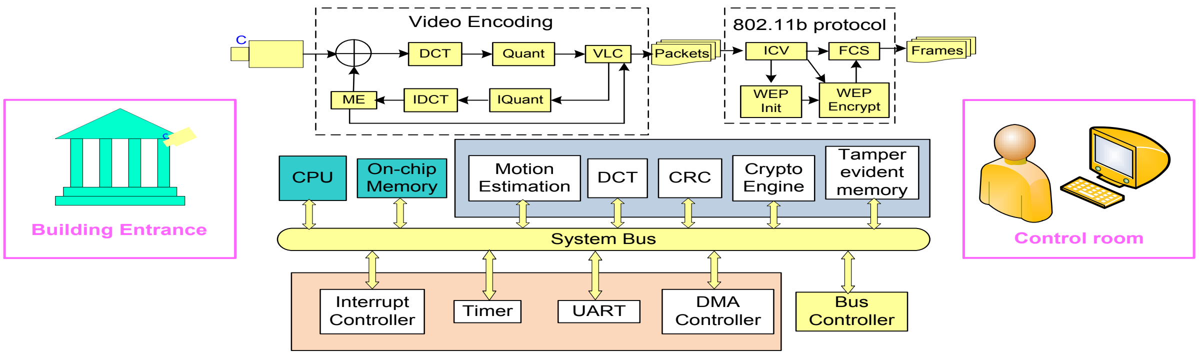
NEW YORK UNIVERSITY




J. Rajendran, O Sinanoglu, R Karri: Building Trustworthy Systems Using Untrusted Components: A High-Level Synthesis Approach. IEEE Trans. VLSI Syst. 24(9): 2946-2959 (2016)

# Collude (a.k.a Distributed Trojans)

NEW YORK UNIVERSITY

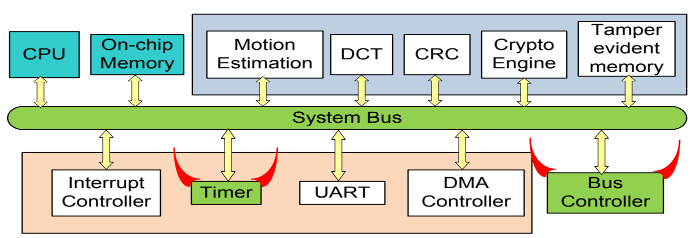



- Wireless Video Capture SoC monitors a building entrance
- Normal: CPU processes camera output → generates video frames → crypto engine encrypts frames → UART transmits to control room
- In the control room, the frames are decrypted and viewed

From: V. Joy, et. al., "Recovery-based design for variation-tolerant SoCs," DAC, 2012

# (Parent-Child) Collusion

NEW YORK UNIVERSITY




- Timer and bus controller obtained from malicious vendor
- Normal operation: Bus contr. controls bus when timer expires
- Malicious operation
  - Timer sends a trigger (within its packet) to bus contr.
  - Trojan in the bus contr. puts the bus in tri-state
  - Output of the SoC freezes
  - Attacker sneaks into the building
- Timer (parent module) colludes with bus contr. (child module)

# Prevent Collusion

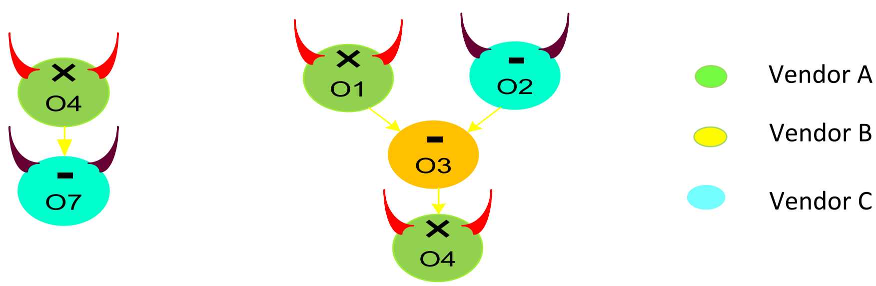
NEW YORK UNIVERSITY


Parent-Child Collusion

- Prevent collusions: Map operations to diverse components
- Parent-Child collusion: Map parent, child ops on diverse components

# Prevent Collusion

NEW YORK UNIVERSITY




Parent-Child Collusion    Parent-Parent Collusion

- Prevent collusions: Map operations to diverse components
- Parent-Child collusion: Map parent, child ops on diverse components
- Parent-Parent collusion: Map at least one parent on a component from a different vendor

# # of Potential Vulnerabilities

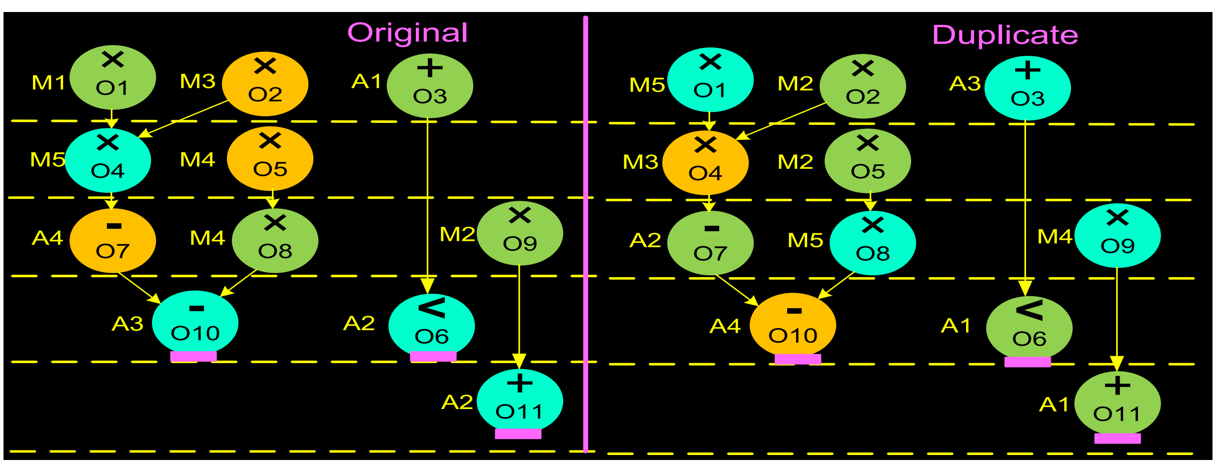
NEW YORK UNIVERSITY

| Design | # of Ops | # of Comm. Paths | # of Potentially Untrustworthy IPs | # of Parent to Child Collusion | # of Parent to Parent Collusion |
|---|---|---|---|---|---|
| $Diff_2Eq$ | 17 | 8 | 17 | 8 | 6 |
| Conv3X3 | 514 | 413 | 514 | 413 | 204 |
| Cordic | 194 | 338 | 194 | 338 | 247 |
| DCT32 | 519 | 612 | 519 | 612 | 306 |
| FIR16 | 63 | 30 | 63 | 30 | 15 |
| Polynom | 8 | 4 | 8 | 4 | 2 |
| Sobel | 391 | 670 | 391 | 670 | 536 |
| Ellipticlass | 37 | 39 | 37 | 39 | 19 |

Opportunities to produce malicious outputs or opportunities to collude

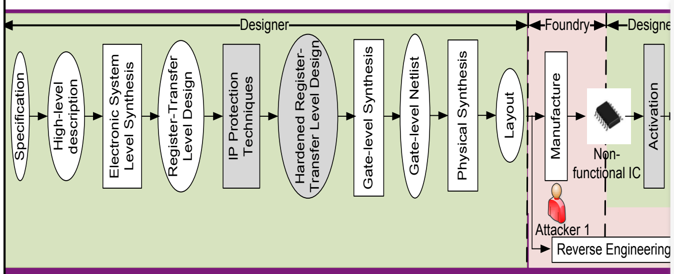
# Detect 3PIPs: Duplicate+Diversify

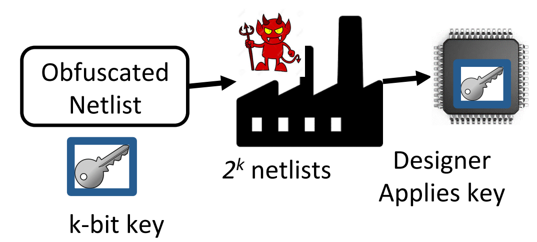
NEW YORK UNIVERSITY




Duplicate + Diversify: 3 vendors; 3 mults 4 adder/comparators/subs
Prevent Parent-Child Collusion and Parent-Parent Collusion

# Untrusted Foundry




- Attacker capabilities
  - Is (in) the Foundry
  - Has the GDSII
  - Does not have access to a (activated/)functional IC
- Objective: Recover the design

C. Pilato, F. Reggazoni, S. Garg and R. Karri, “TAO: Techniques for Algorithm Level Obfuscation During High-Level Synthesis,” Proc IEEE/ACM Design Automation Conf, June 2018.

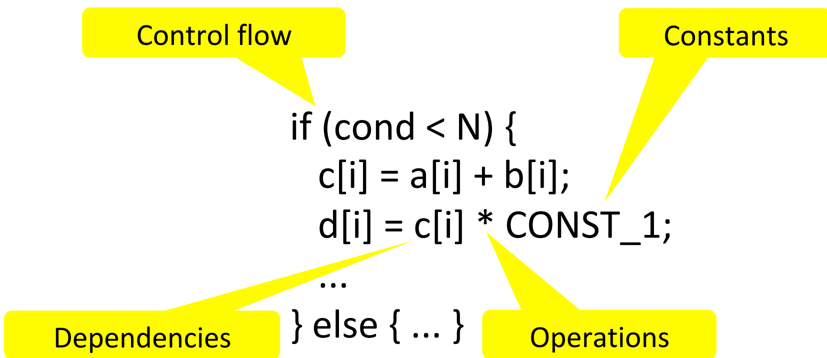
# Algorithm Obfuscation

```
if (cond < N) {
  c[i] = a[i] + b[i];
  d[i] = c[i] * CONST_1;
  ...
} else { ... }
```

Several ways to obfuscate an algorithm

# Algorithm Obfuscation

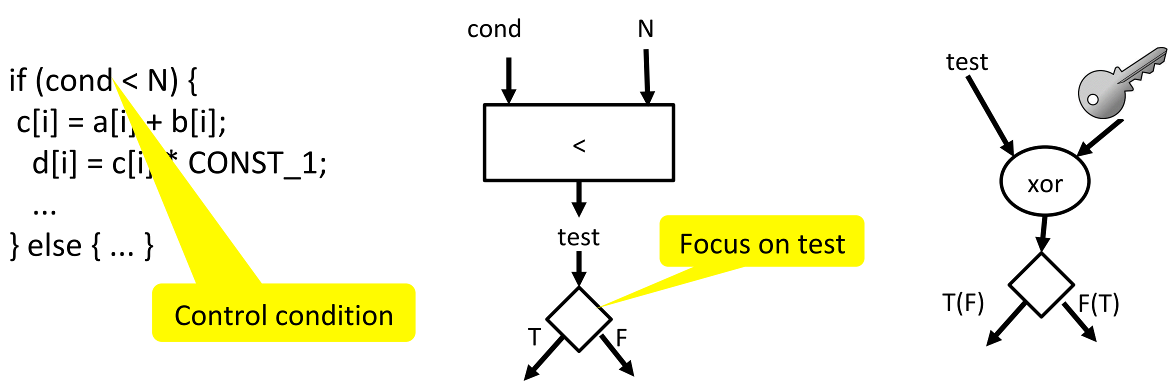
NEW YORK UNIVERSITY




# Obfuscate Control Flow

NEW YORK UNIVERSITY

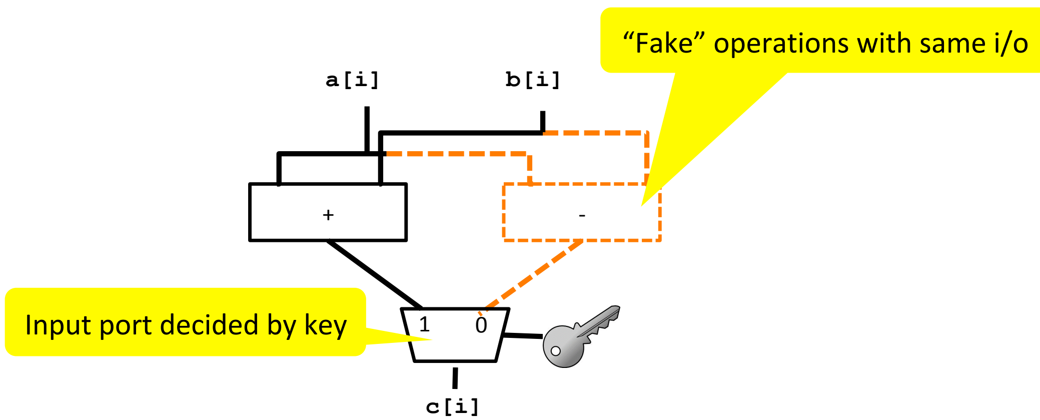
- Mask control condition with key bit
- Correct branch is taken only with correct key
- Reorder Branch: Ensures semantic equivalence + confuse attacker



# Obfuscate Operations

NEW YORK UNIVERSITY

- Gives intelligence on what the algorithm does
- Operator variants can camouflage correct operation
- Correct result is propagated only with the correct key

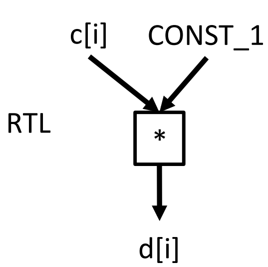



# Obfuscate Constants

NEW YORK UNIVERSITY

- Hard-coded values used by algorithm (coefficients, thresholds, …)
- Information is maintained at RTL
- Extensively optimized during logic synthesis

C/C++: d[i] = c[i] * CONST_1;




No impact on security, less keys

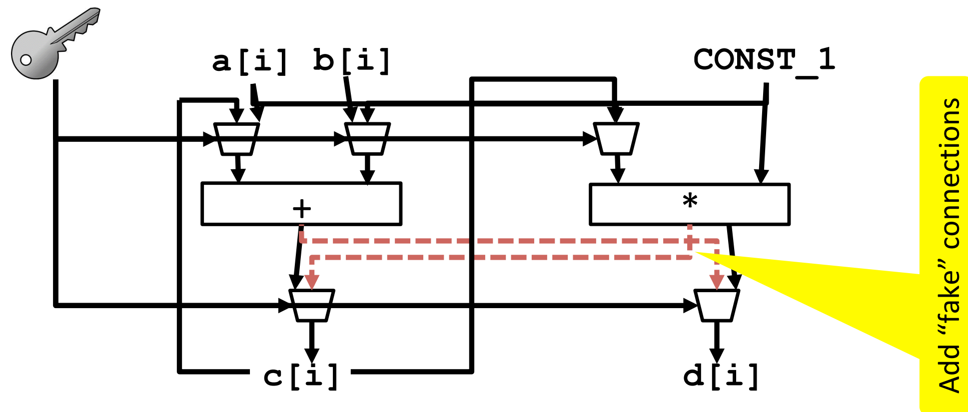
| Obfuscated | Not obfuscated |
|---|---|
| Data co-efficients | Reset values |
| Signal extensions | Signal polarity |
| Mask values | |

No impact on semantics

# Obfuscate Dependencies

NEW YORK UNIVERSITY

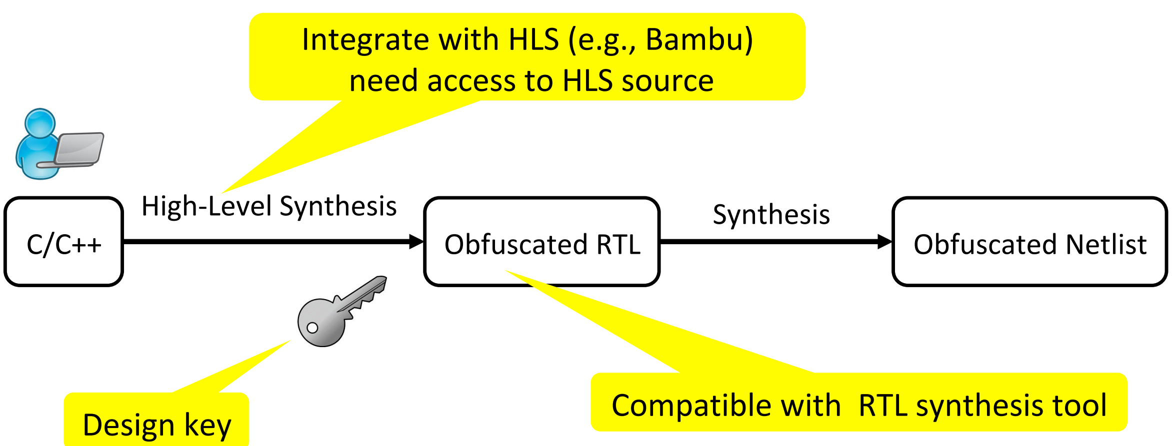
- K-bit key is used to select $2^k$ DFG variants




Correct paths are activated only with the correct key

# HLS Obfuscation

NEW YORK UNIVERSITY




Semantic Obfuscation: Branches, Dependencies, Operations, Constants

# Results

NEW YORK UNIVERSITY

| Design name | Obfuscation | | | # of key bits |
|---|---|---|---|---|
| | Constant | Branch | DFG Variants | |
| GSM | 4 / 128 | 4 | 88 / 352 | 484 |
| ADPCM | 5 / 160 | 5 | 100 / 400 | 565 |
| SOBEL | 2 / 64 | 2 | 11 / 44 | 110 |
| BACKPROP | 12 / 384 | 11 | 123 / 492 | 887 |
| VITERBI | 117 / 3,744 | 9 | 98 / 392 | 4,145 |

Obfuscated consts / used key bits

Obfuscated branches

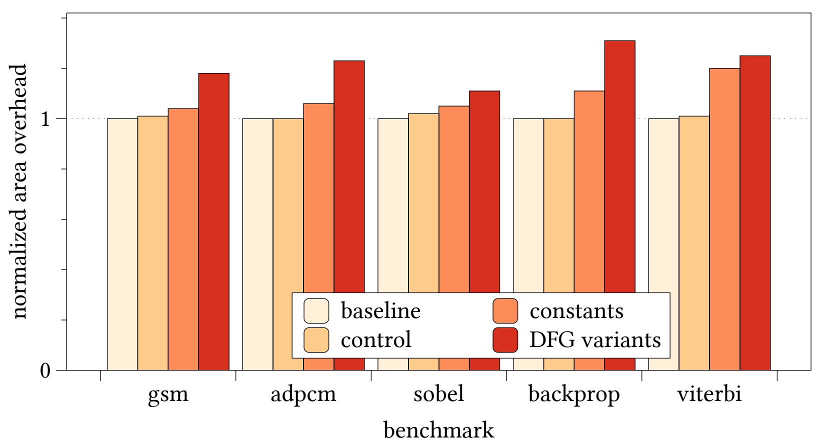
# of Basic Blocks / key bits

# of key bits

Bambu Open Source HLS (automatic generation from C-to-HDL)

# Overhead

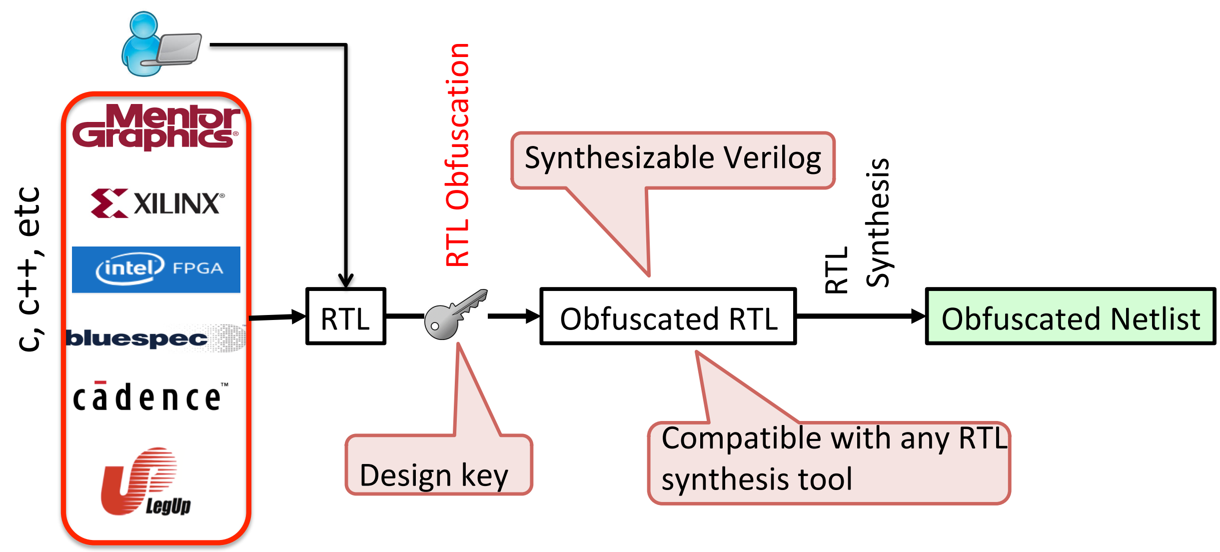
NEW YORK UNIVERSITY




- Area overhead of each technique wrt the **baseline** version
  - Synopsys SAED 32nm @ 500 MHz
- Operation+Dependence obfuscation

# RTL Transformations for Security

NEW YORK UNIVERSITY




# RTL Obfuscation: Results

NEW YORK UNIVERSITY

| DSP module | Const. Obf. | Branch Obf. | Ops. Obf. | Total key bits |
|---|---|---|---|---|
| add_only_decimator_par32x | 80 / 240 | 24 | 32 | 296 |

Module name

Obfuscated constants / Number of used key bits

Obfuscated branches

Obfuscated operations

Total number of used key bits

# RTL Obfuscation: Results

NEW YORK UNIVERSITY

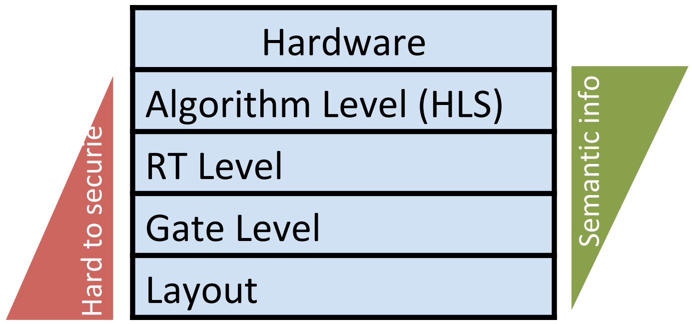
| DSP module | Const. Obf. | Branch Obf. | Ops. Obf. | Total key bits |
|---|---|---|---|---|
| add_only_decimator_par32x | 80 / 240 | 24 | 32 | 296 |
| aod_par32x_section0 | 64 / 208 | 100 | 128 | 436 |
| aod_par32x_section1 | 32 / 120 | 69 | 80 | 269 |
| aod_par32x_section2 | 16 / 68 | 55 | 56 | 179 |
| coarse_time_delay_par2x | 0 / 0 | 4 | 0 | 4 |
| convert_to_cos_sin_32x_elem0 | 34 / 173 | 64 | 97 | 334 |
| data_select_and_decimate_par4x | 0 / 0 | 4 | 0 | 4 |
| delay_i_par{2x0\|2x1\|4x0} | 0 / 0 | 1 | 0 | 1 |
| dotprod_par4x_16taps0 | 0 / 0 | 17 | 31 | 48 |

# Conclusions

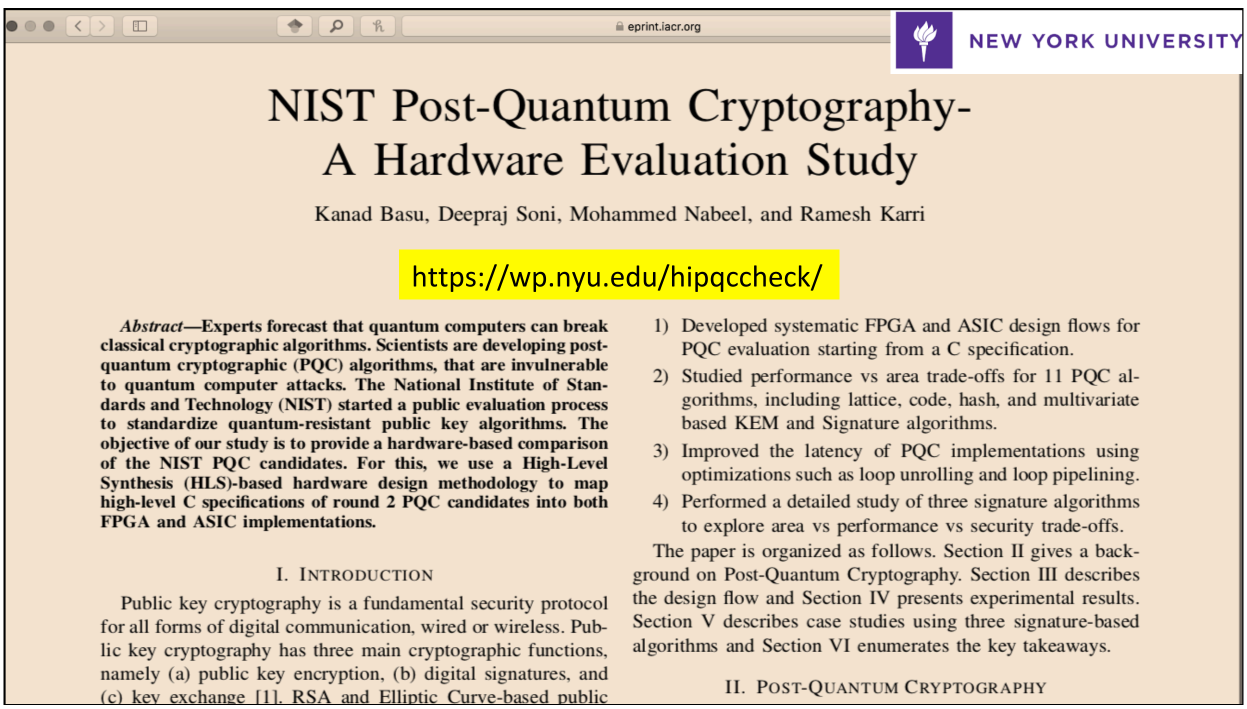
NEW YORK UNIVERSITY

1. RTL is a promising level to Design-in Security
   - C Pilato, S Garg, K Wu, R Karri, F Regazzoni, *Securing Hardware Accelerators: a New Challenge for High-Level Synthesis,* (a Perspective Paper), IEEE Embedded Systems Letters, DOI: 10.1109/LES.2017.2774800
2. HLS can be used for Trojan Detection and Isolation
   - J. Rajendran, O Sinanoglu, and R Karri, *Building Trustworthy Systems Using Untrusted Components: A High-Level Synthesis Approach*, IEEE Trans VLSI, 24(9): 2946-2959, Sep 2016, DOI: 10.1109/TVLSI.2016.2530092
3. Watermark designs during High-Level Synthesis
   - C. Pilato and K. Basu and M. Shayan and F. Regazzoni and R. Karri, High-Level Synthesis of Benevolent Trojans, Design Automation and Test in Europe Conference (DATE), pp. 1118—1123, March, 2019.
4. Design obfuscation benefits from High-Level semantic information
   - C. Pilato, F. Reggazoni, S. Garg and R. Karri, TAO: Techniques for Algorithm Level Obfuscation During High-Level Synthesis, Proc IEEE/ACM Design Automation Conf, June 2018, DOI: 10.1109/DAC.2018.8465830.
5. Taint Propagation is seamless during HLS
   - C. Pilato, F. Reggazoni, S. Garg and R. Karri, TaintHLS: High-Level Synthesis For Dynamic Information Flow Tracking, IEEE Trans. CAD, DOI: 10.1109/TCAD.2018.2834421
6. HLS-generated designs can be reverse engineered !
   - J. Rajendran, A. Ali, O. Sinanoglu and R. Karri, *Belling the CAD: Toward Security-Centric Electronic System Design,* IEEE Trans. CAD, Vol 34, No. 11, pp. 1756-1769, Nov 2015, DOI: 10.1109/TCAD.2015.2428707.
7. One can use High-Level Synthesis for Black-Hat purposes
   - C Pilato, K Basu, F Regazzoni, R Karri, Black-Hat High-Level Synthesis: Myth or Reality? IEEE Transactions on Very Large Scale Integration (VLSI) System, DOI: 10.1109/TVLSI.2018.2884742

# Security: A Summary




Sensitive IP: Constants, control flow, dependencies, operations, CDFGs



eprint.iacr.org

## NIST Post-Quantum Cryptography- A Hardware Evaluation Study

Kanad Basu, Deepraj Soni, Mohammed Nabeel, and Ramesh Karri

https://wp.nyu.edu/hipqccheck/

***Abstract*—Experts forecast that quantum computers can break classical cryptographic algorithms. Scientists are developing post-quantum cryptographic (PQC) algorithms, that are invulnerable to quantum computer attacks. The National Institute of Standards and Technology (NIST) started a public evaluation process to standardize quantum-resistant public key algorithms. The objective of our study is to provide a hardware-based comparison of the NIST PQC candidates. For this, we use a High-Level Synthesis (HLS)-based hardware design methodology to map high-level C specifications of round 2 PQC candidates into both FPGA and ASIC implementations.**

### I. Introduction

Public key cryptography is a fundamental security protocol for all forms of digital communication, wired or wireless. Public key cryptography has three main cryptographic functions, namely (a) public key encryption, (b) digital signatures, and (c) key exchange [1]. RSA and Elliptic Curve-based public

1) Developed systematic FPGA and ASIC design flows for PQC evaluation starting from a C specification.
2) Studied performance vs area trade-offs for 11 PQC algorithms, including lattice, code, hash, and multivariate based KEM and Signature algorithms.
3) Improved the latency of PQC implementations using optimizations such as loop unrolling and loop pipelining.
4) Performed a detailed study of three signature algorithms to explore area vs performance vs security trade-offs.

The paper is organized as follows. Section II gives a background on Post-Quantum Cryptography. Section III describes the design flow and Section IV presents experimental results. Section V describes case studies using three signature-based algorithms and Section VI enumerates the key takeaways.

### II. Post-Quantum Cryptography

# Security-Aware HLS

NEW YORK UNIVERSITY

NEW YORK UNIVERSITY

?

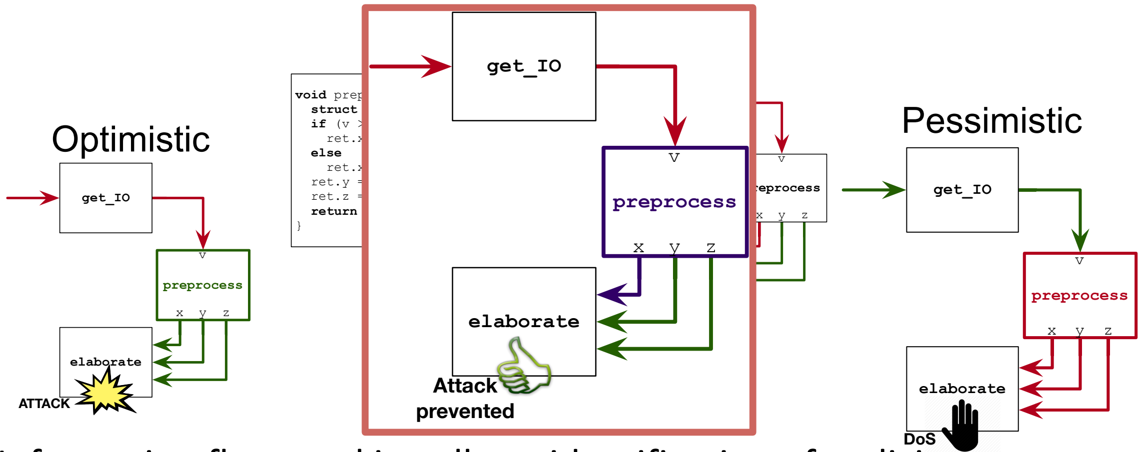
Cell: 917 363 9703

rkarri@nyu.edu

http://cyber.nyu.edu

# Monitoring Information Flow

NEW YORK UNIVERSITY

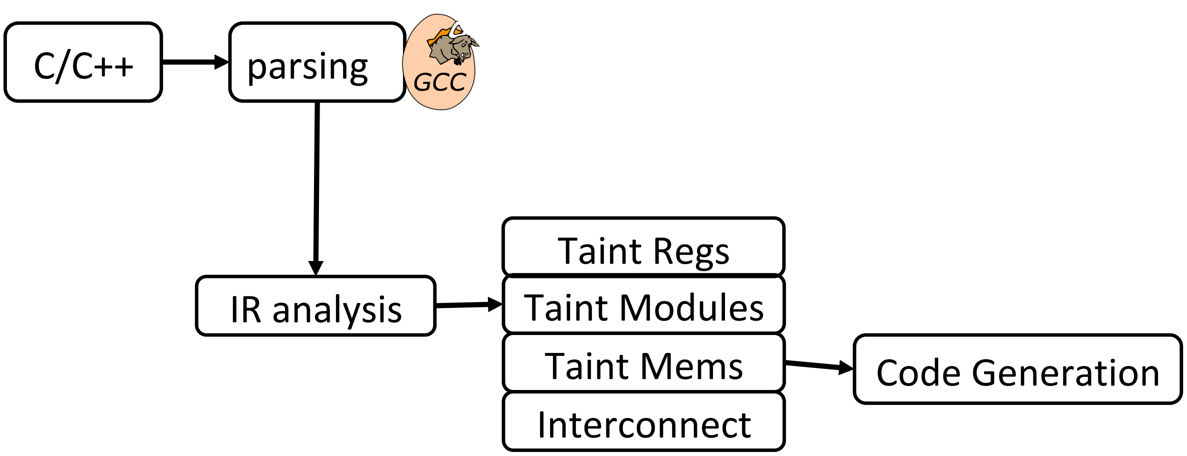



- Information flow tracking allows identification of malicious uses
- No existing support for hardware accelerators for intrinsic DIFT

C. Pilato, F. Reggazoni, S. Garg and R. Karri, TaintHLS: High-Level Synthesis For Dynamic Information Flow Tracking, IEEE Trans. CAD, DOI: 10.1109/TCAD.2018.2834421

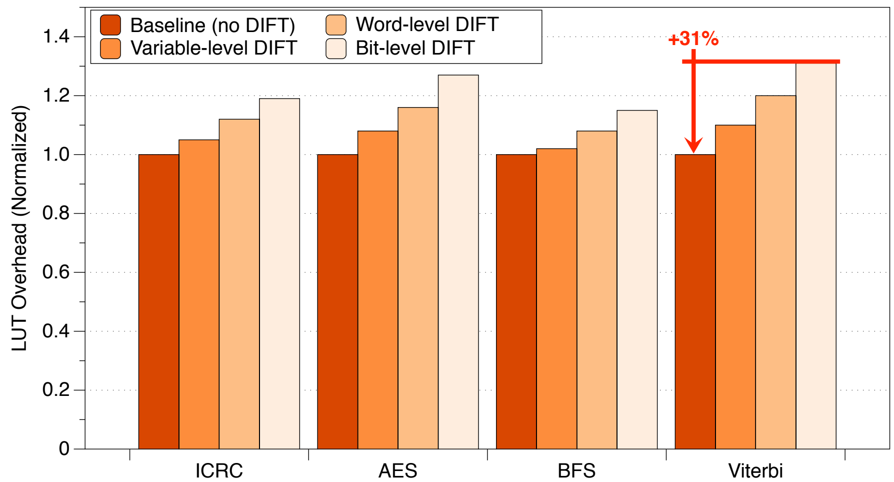
# HLS for Information Flow Tracking

NEW YORK UNIVERSITY



# Taint-HLS: Area Overhead

NEW YORK UNIVERSITY




Xilinx Virtex-7 FPGA @ 100 MHz