NEW YORK UNIVERSITY

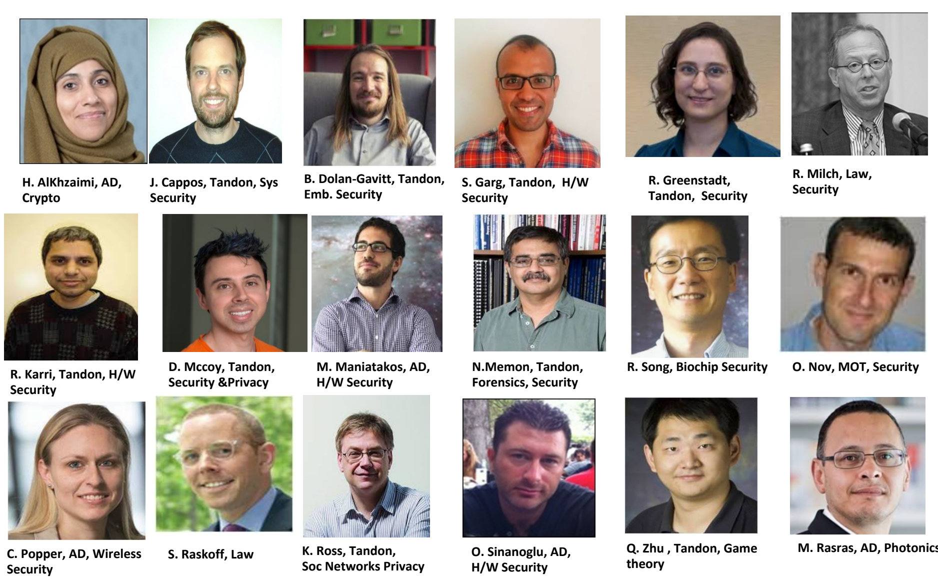
High-Level Approaches to Hardware Security

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

1.High-Level is a promising level to Design Security Accelerators
K. Basu, D. Soni, N. Mohammed, R. Karri, *NIST Post Quantum Cryptography: A Hardware Evaluation Study*, Jan 2019; iacr eprint
2.High-Level is a promising level to Design 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
3.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
J. Rajendran, H. Zhang, O. Sinanoglu and R. Karri, *High-level synthesis for security and trust,* IEEE Intl On-Line Testing Symposium, pp. 232-233. July 2013, doi: 10.1109/IOLTS.2013.6604087
4.HLS can be used to Watermark Designs
C. Pilato and K. Basu and M. Shayan and F. Regazzoni and R. Karri, *High-Level Synthesis of Benevolent Trojans*, Design Automation Test in Europe Conference, pp. 1118—1123, March, 2019.
5.HLS can be used for Seamless and Meaningful Design Obfuscation
C. Pilato, F. Reggazoni, S. Garg and R. Karri, *TAO: Techniques for Algorithm-Level Obfuscation During High-Level Synthesis*, IEEE/ACM Design Automation Conference, June 2018, DOI: 10.1109/...
6.HLS can be used for Seamless and Meaningful Taint Propagation
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
7.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.
8.A Black-Hat can use High-Level Synthesis to undermine Designs (weaken crypto, drain battery, etc)
C Pilato, K Basu, F Regazzoni, R Karri, *Black-Hat High-Level Synthesis: Myth or Reality?* IEEE Trans. VLSI, DOI: 10.1109/TVLSI.2018.2884742

http://cyber.nyu.edu/

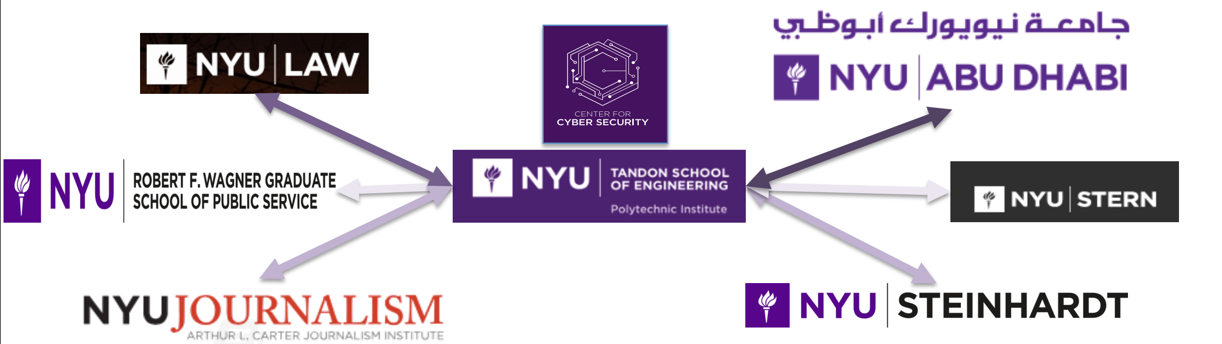
NEW YORK UNIVERSITY



Mission

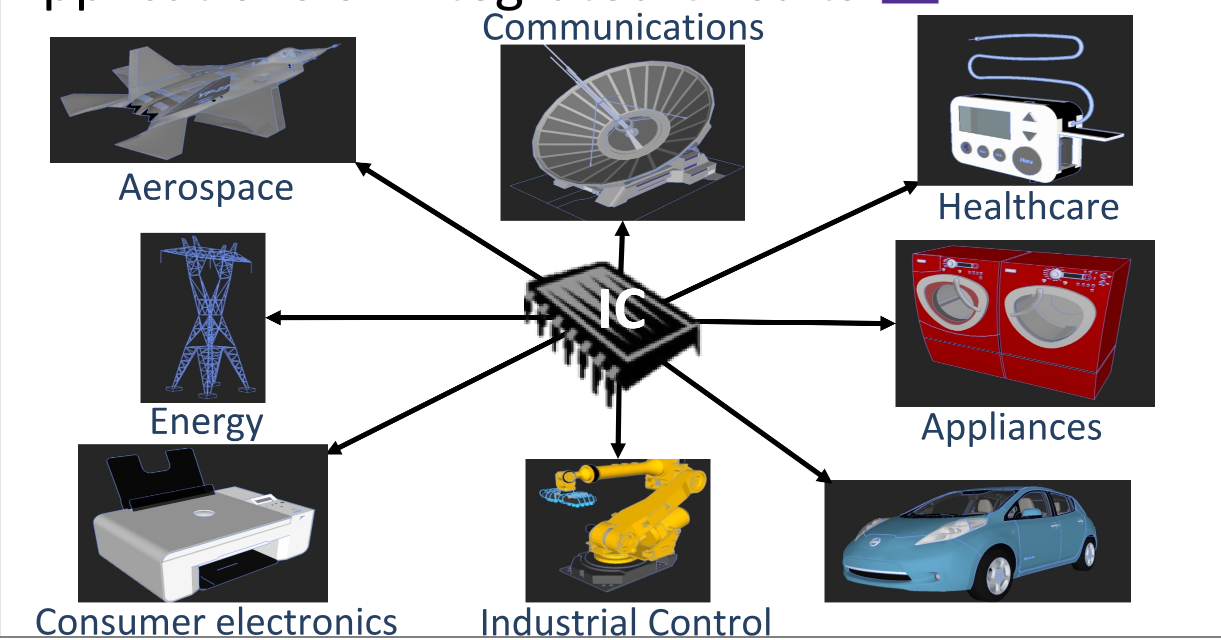
NEW YORK UNIVERSITY

NYU CCS is an interdisciplinary center dedicated to
- Research technical and other means to secure cyber infrastructure.
- Educate the next generation of cybersecurity professionals.
- Shape public discourse on policy and legal aspects of cybersecurity.



Applications of Integrated Circuits

NEW YORK UNIVERSITY

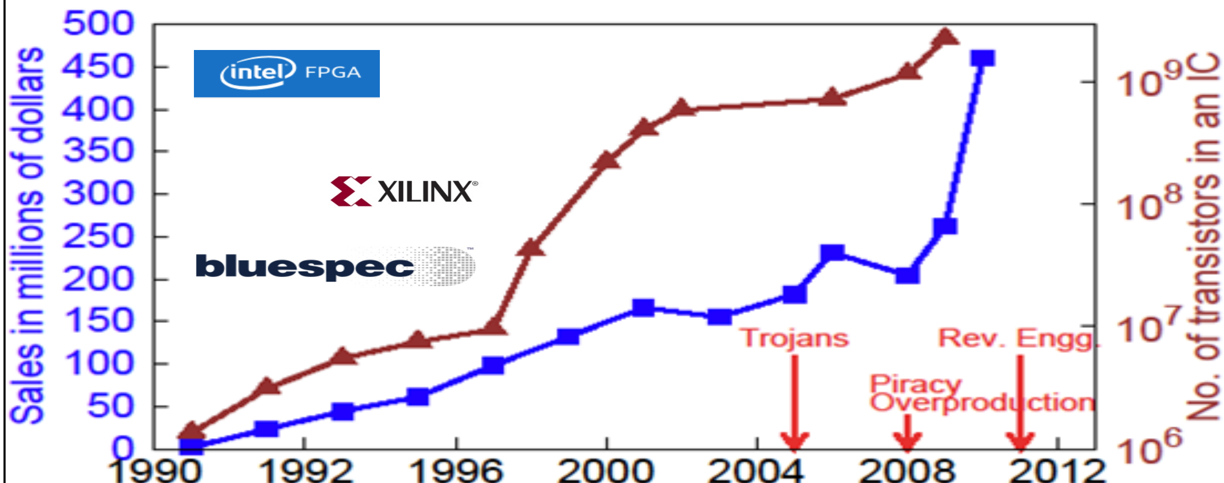


Contributions to H/W Security

NEW YORK UNIVERSITY

High-Level Synthesis

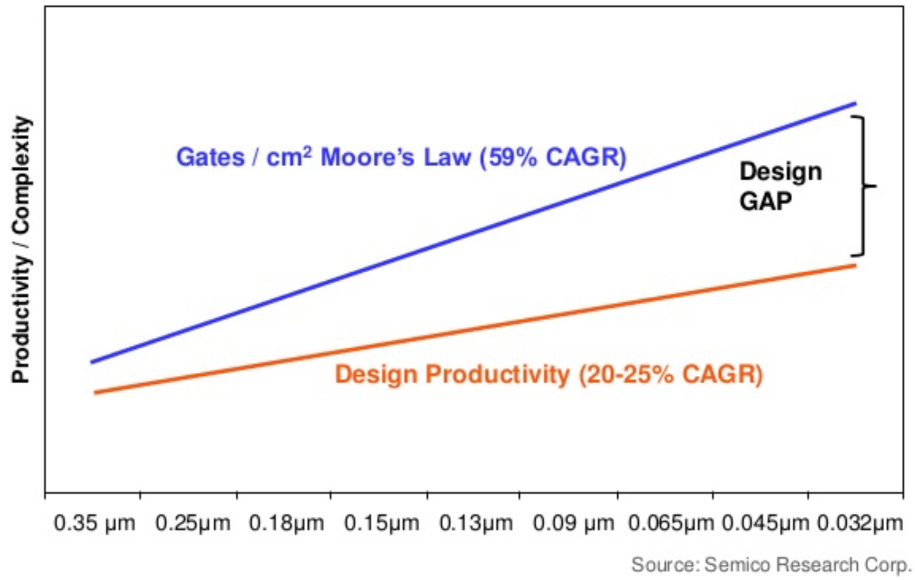
NEW YORK UNIVERSITY



CALYPTO ⟷ Mentor Graphics ⟷ Oasys FORTE DESIGN SYSTEMS ⟷ cādence

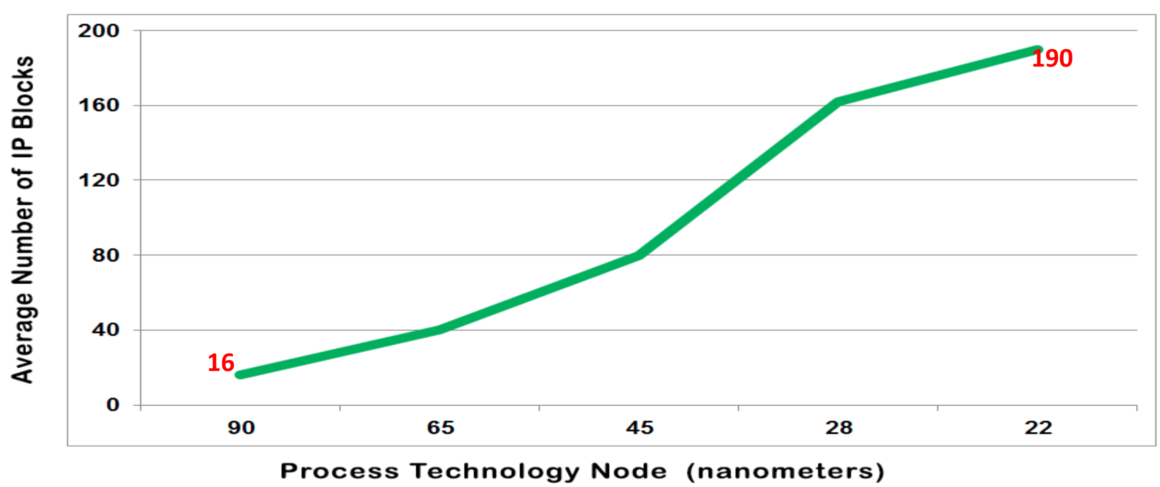
HLS is a Productivity Tool

NEW YORK UNIVERSITY



More 3rd Party IPs in a Design

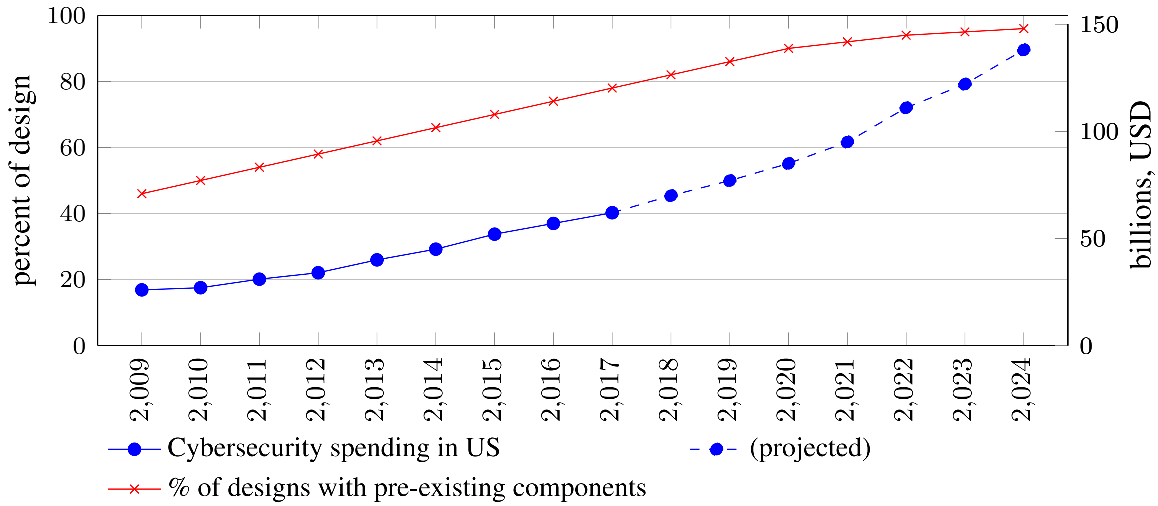
NEW YORK UNIVERSITY



(International Business Strategies, 2012)

Accelerator-based Design (ASAP)!

NEW YORK UNIVERSITY

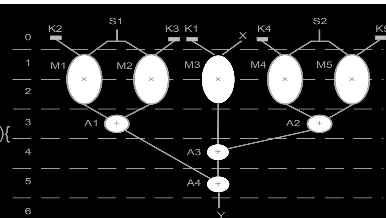


HLS Design Flow

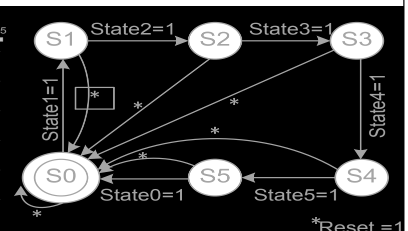
NEW YORK UNIVERSITY

```
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);
}
```

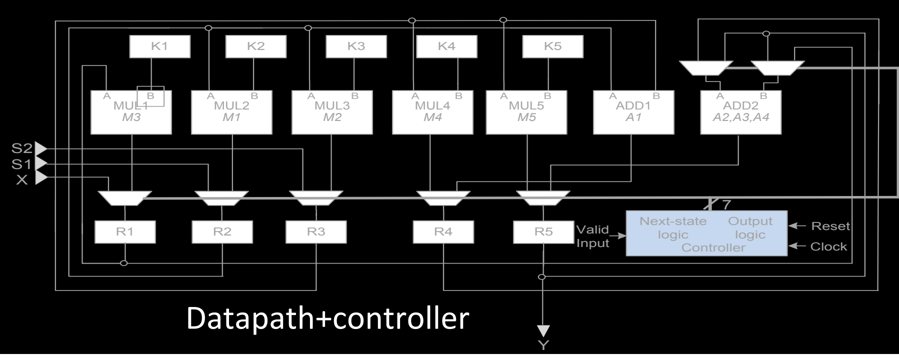
c-specification of biquad filter



Scheduling and binding



Finite state machine



Datapath+controller

Security-Aware (HLS for) ASAPs

NEW YORK UNIVERSITY

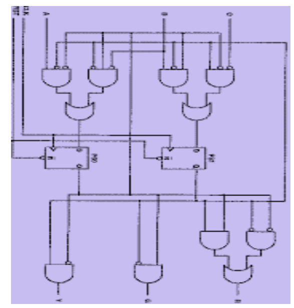
Hard to secure

Semantic Information

C/C++ → RTL → Netlist

```
for (i=0; i<N; i++)
      c[i] = a[i]+b[i];
      .....
```

```
always @ posedge clk
      a[i] <= b[i] + c[i];
      .....
```



Security-Aware HLS for ASAPs

NEW YORK UNIVERSITY

- Promising to add security constraints
- HLS in Hardware vs Programming Lang/Compilers in Software
- Semantics: sensitive constants, critical operations, protected control flow, run-time dependencies (sensitive IP)

Hard to secure

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

Semantic info

Takeaways

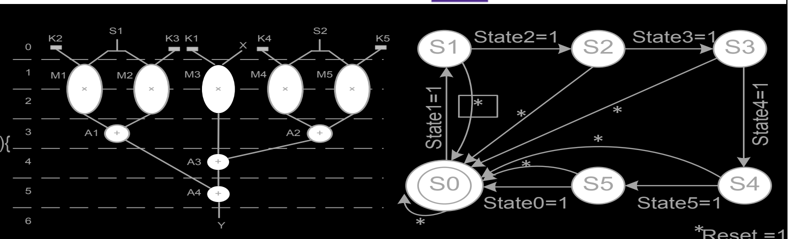
NEW YORK UNIVERSITY

1. High-Level is a promising level to Design Security Accelerators
   K. Basu, D. Soni, N. Mohammed, R. Karri, NIST Post Quantum Cryptography: A Hardware Evaluation Study; iacr eprint
2. High-Level 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
3. 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
   J. Rajendran, H. Zhang, O. Sinanoglu and R. Karri, *High-level synthesis for security and trust,* IEEE On-Line Testing Symposium, pp. 232-233. July 2013, doi: 10.1109/IOLTS.2013.6604087
4. HLS can be used to Watermark Designs
   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, pp. 1118—1123, March, 2019.
5. HLS can be used for Seamless and Meaningful Design Obfuscation
   C. Pilato, F. Reggazoni, S. Garg and R. Karri, *TAO: Techniques for Algorithm Level Obfuscation During High-Level Synthesis*, IEEE/ACM Design Automation Conference, June 2018, DOI: 10.1109/DAC.2018.8465830.
6. HLS can be used for Seamless and Meaningful Taint Propagation
   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
7. 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.
8. A Black-Hat HLS can Undermine Designs (weaken crypto, drain battery, …) !!
   C Pilato, K Basu, F Regazzoni, R Karri, *Black-Hat High-Level Synthesis: Myth or Reality?* IEEE Trans. VLSI, DOI: 10.1109/TVLSI.2018.2884742

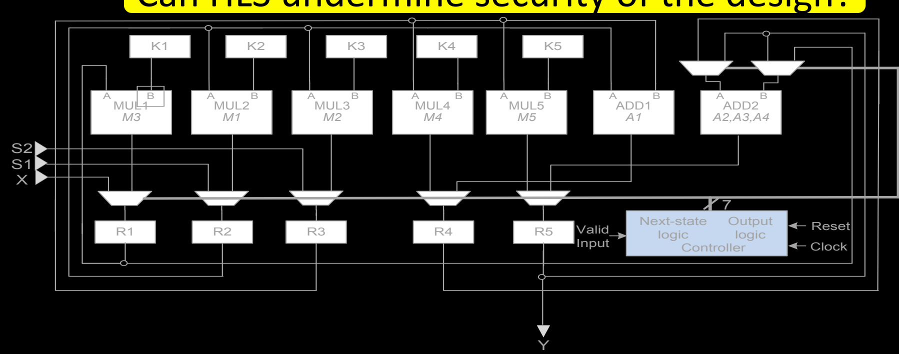
HLS Design Flow

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);
}
```

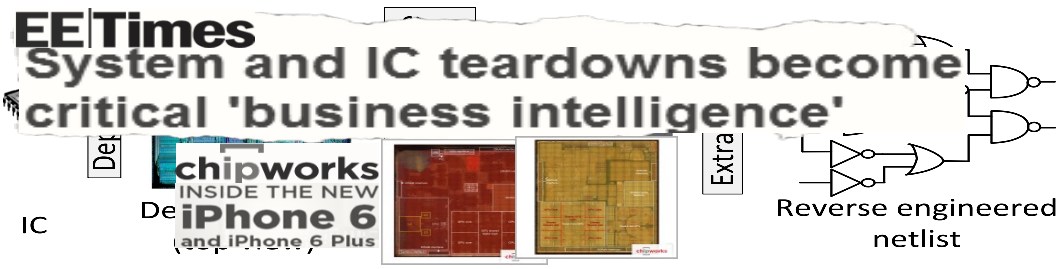


Can HLS undermine security of the design?



Threat: Reverse Engineering

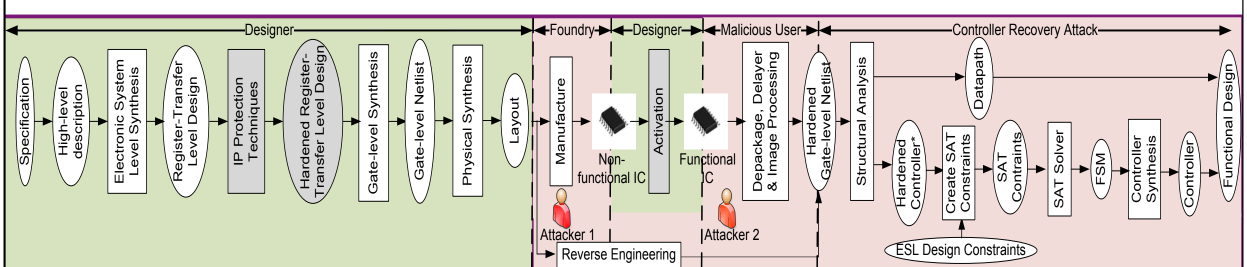
NEW YORK UNIVERSITY



- 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

HLS-based Reverse Engg an ASAP

NEW YORK UNIVERSITY

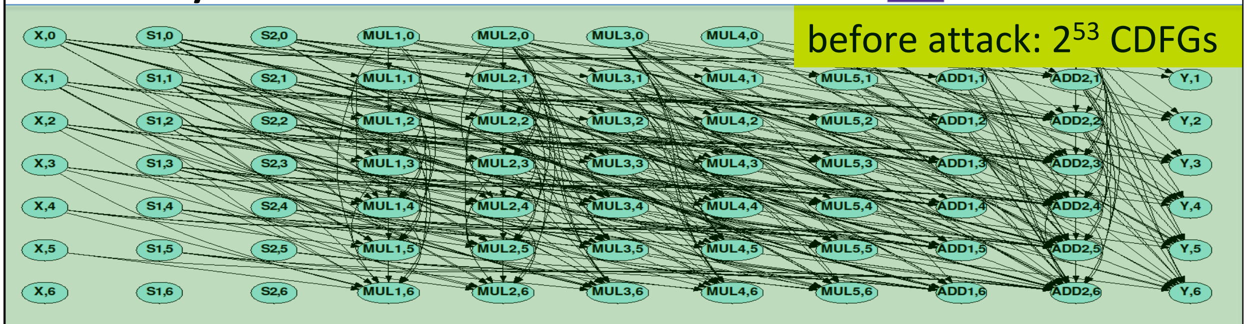


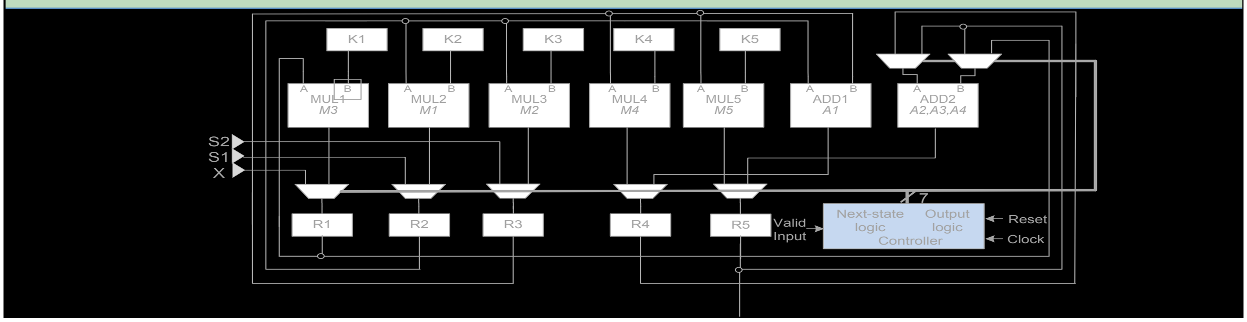
J. Rajendran, A. Ali, O. Sinanoglu and R. Karri, Belling the CAD: Toward Security-Centric Electronic System Design, IEEE Transactions on CAD, Vol 34, No. 11, pp. 1756-1769, November, 2015.

Security Metric: # of CDFGs

NEW YORK UNIVERSITY

before attack: 2^{53} CDFGs

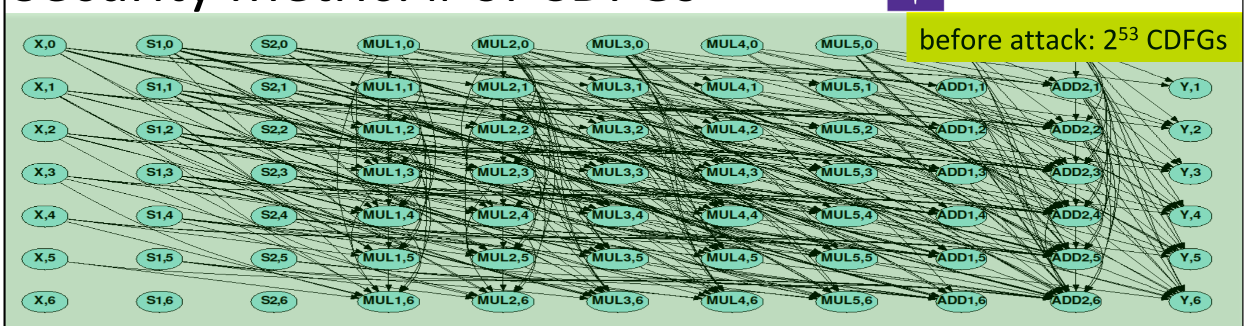


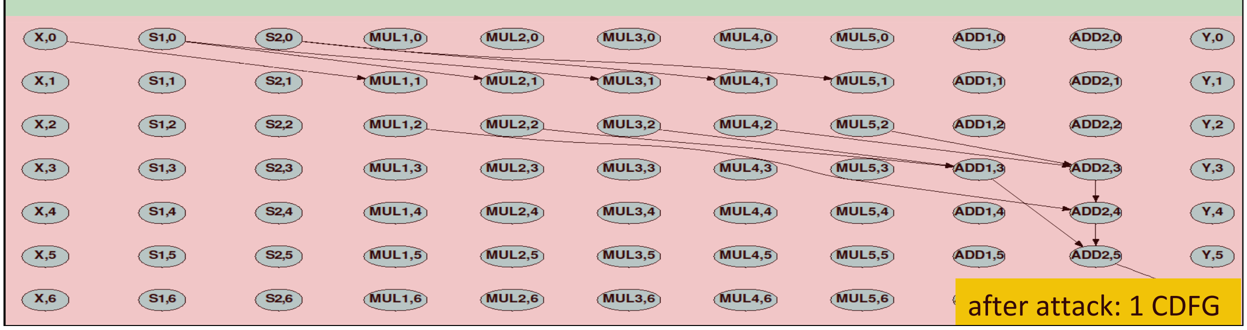


Security Metric: # of CDFGs

NEW YORK UNIVERSITY

before attack: 2^{53} CDFGs

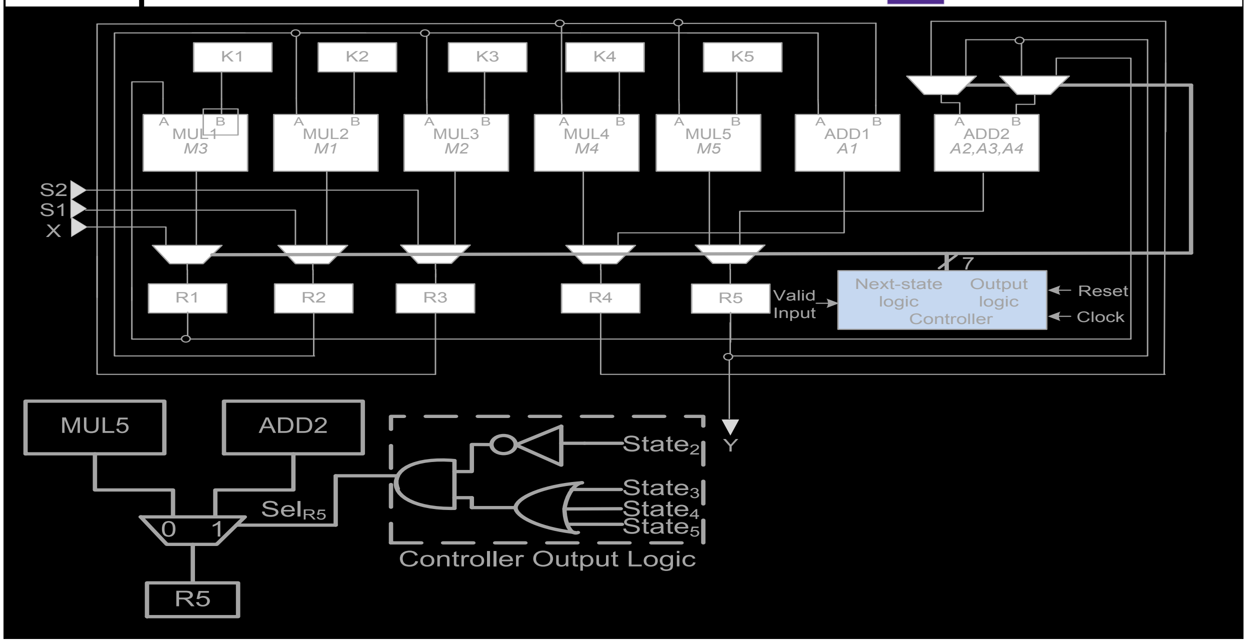




after attack: 1 CDFG

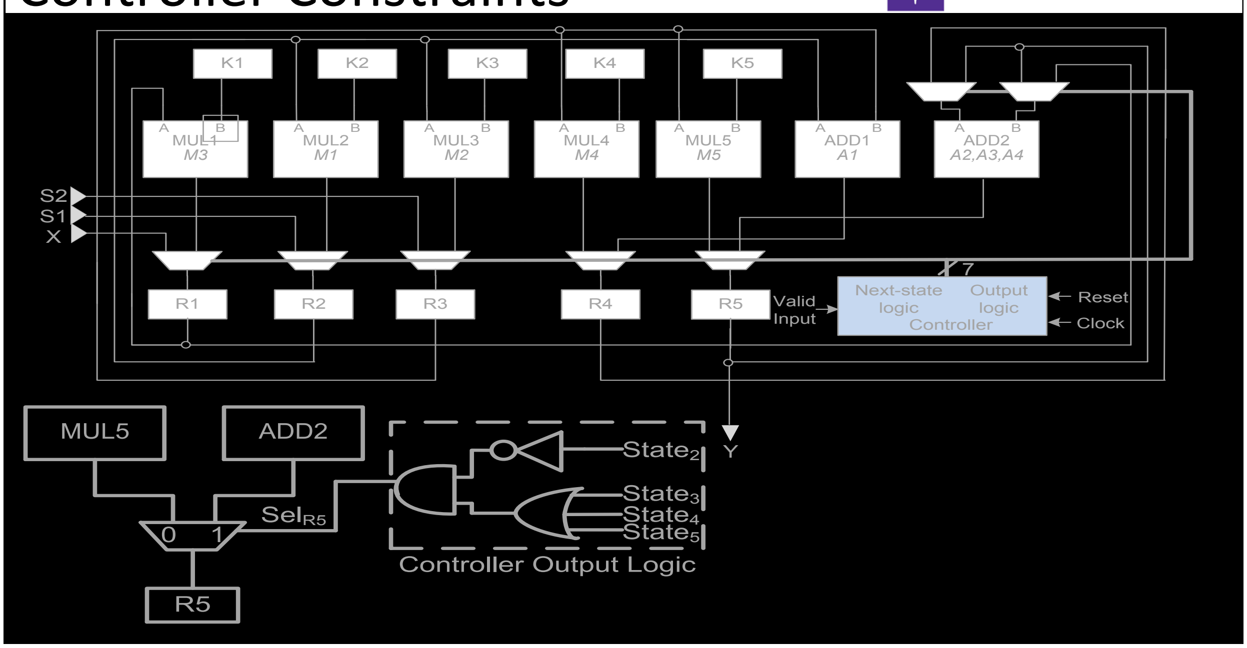
Datapath Constraints

NEW YORK UNIVERSITY



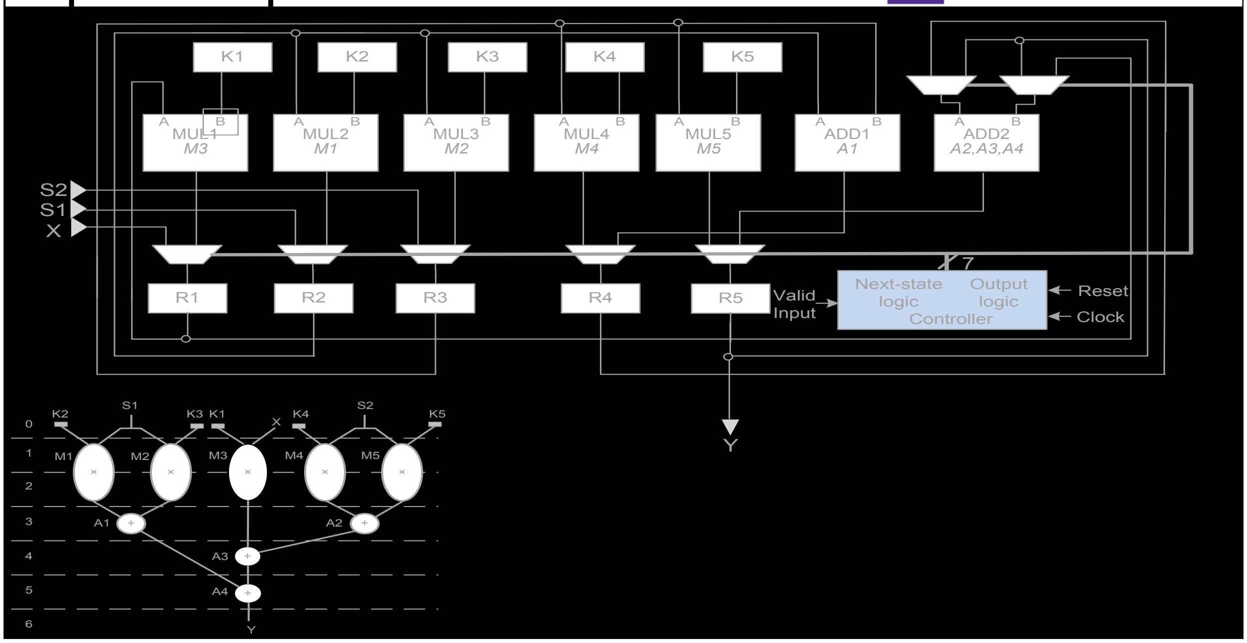
Controller Constraints

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}

# of CDFGs reduce drastically using HLS constraints

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.

Belled the CAD!

NEW YORK UNIVERSITY

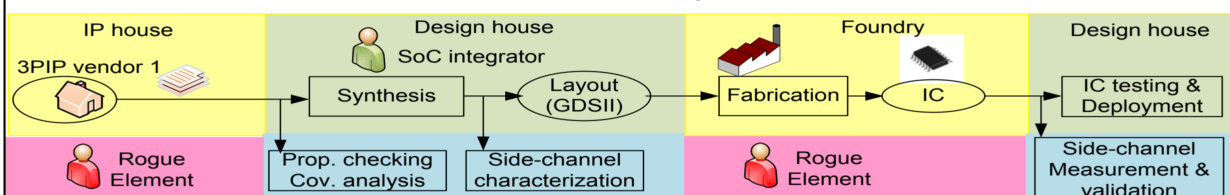
Design	Tools A,B, C, D & E: Non-pipelined and Resource-Constrained				
	Attack Success			Attack Cost	
	No. of compare points	% compare points matched	Equivalence checking	# of SAT literals	Time for solving SAT (s)
BQF	16	100	Pass	1050	0.01
Arai	128	100	Pass	5166	0.02
Chem	240	100	Pass	2415264	43
Dir	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Feig_dct	1024	100	Pass	517545	5.17
Honda	144	100	Pass	[illegible]	1.10
Lee	128	100	Pass	10374	0.05
Mcm	128	100	Pass	56160	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
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

All benchmarks reverse engineered in <30 minutes
Functionally equivalent and structurally identical!

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.

Threat: Malicious 3PIP (Trojans)

NEW YORK UNIVERSITY



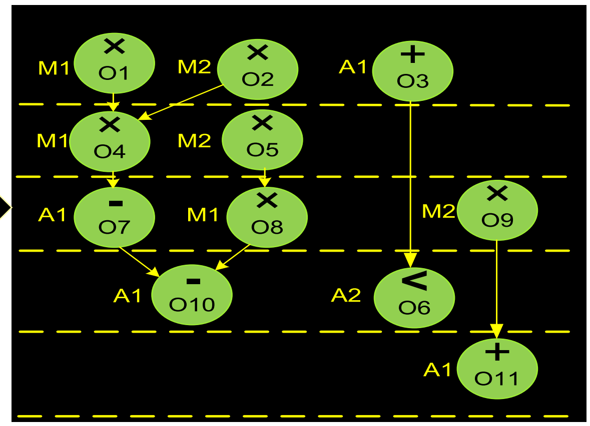
- 3PIP vendors are not trusted; may insert trojans
  - Trojans cause wrong outputs
  - Distributed: in different modules from same vendor may collude
- SoC integrator is trusted
  - SoC integrator uses components from 3PIP vendors
  - 3PIPs are integrated into a system and synthesized
- SoC is manufactured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

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

HLS for Trojan Detection

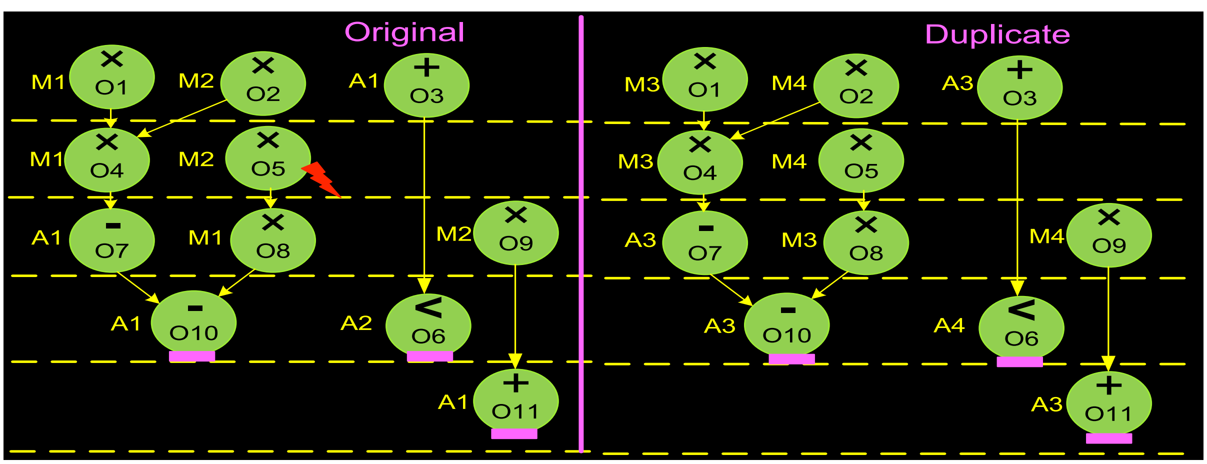
NEW YORK UNIVERSITY

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



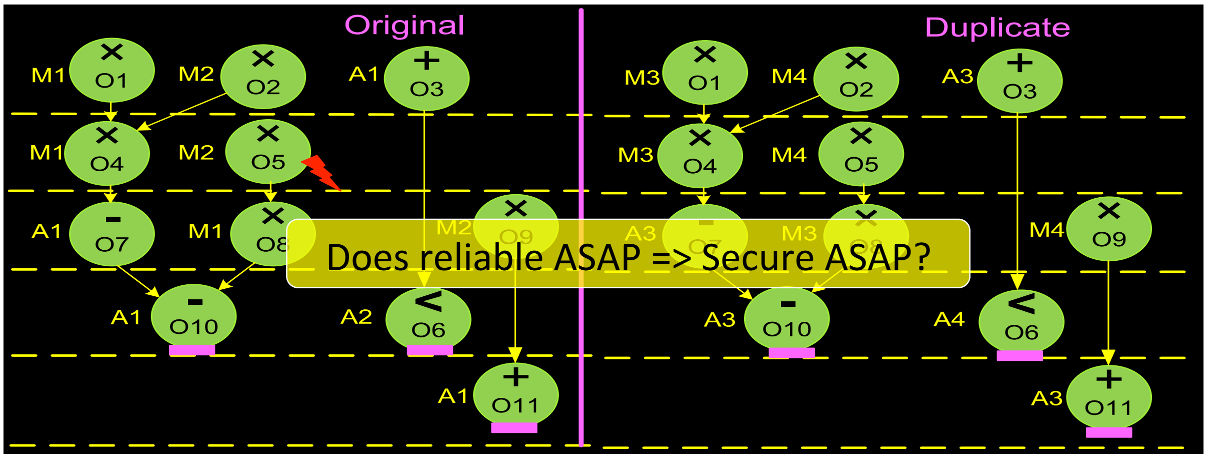
Detect “Natural” Faults

NEW YORK UNIVERSITY



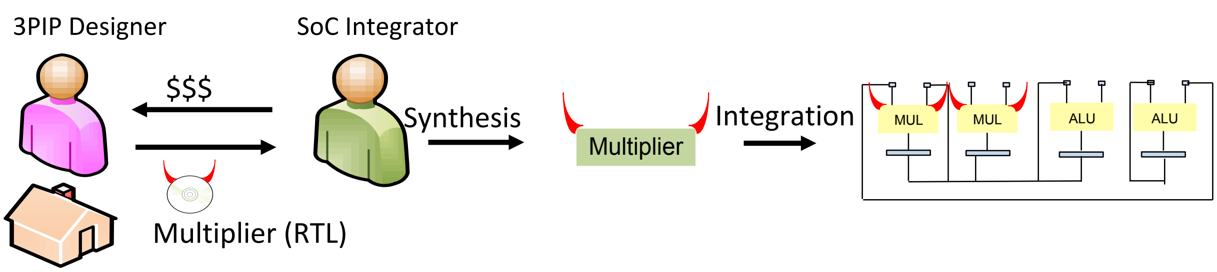
Detect "Natural" Faults

NEW YORK UNIVERSITY



Malicious 3PIPs

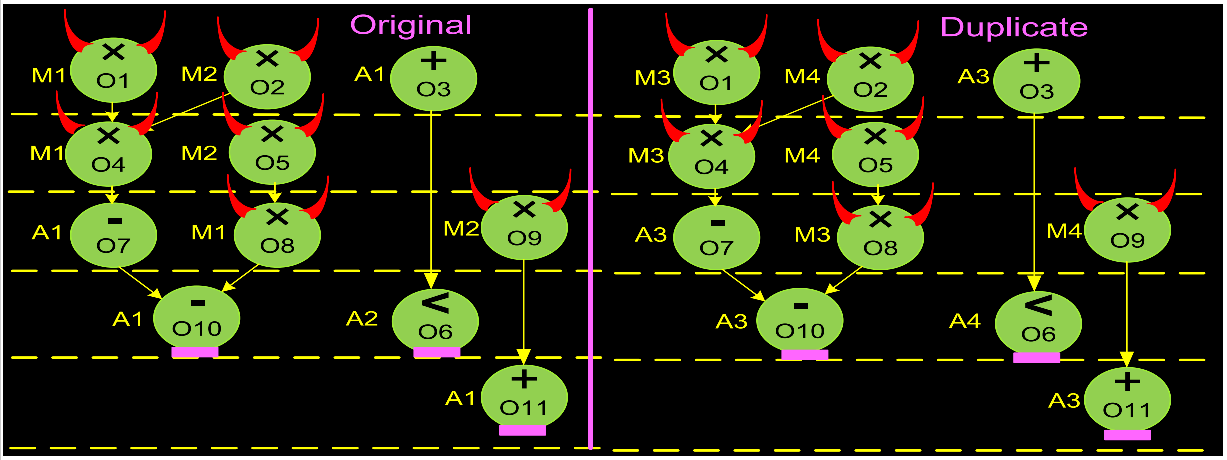
NEW YORK UNIVERSITY



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

Detect Trojan: Duplicate+Check

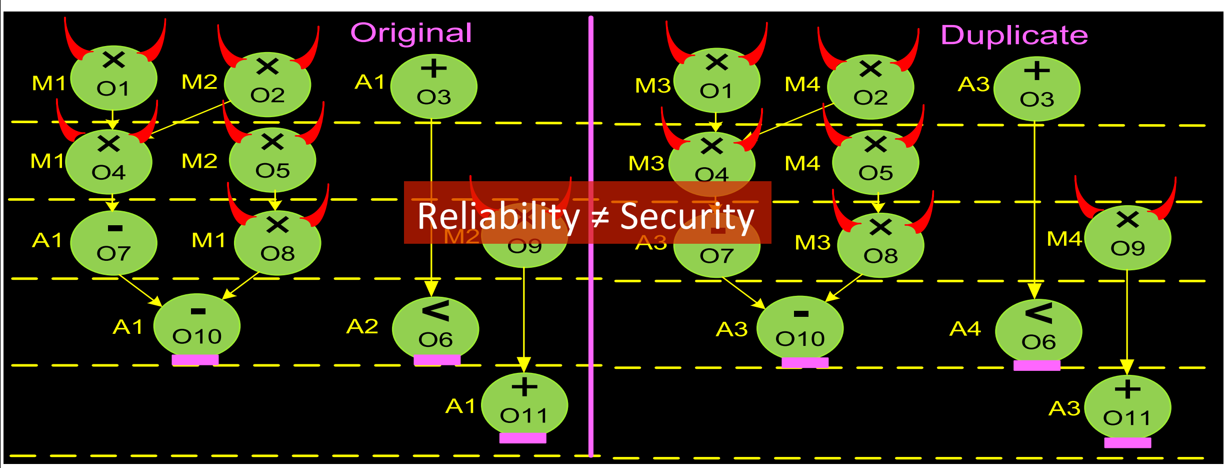
NEW YORK UNIVERSITY



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

Detect Trojan: Duplicate+Check

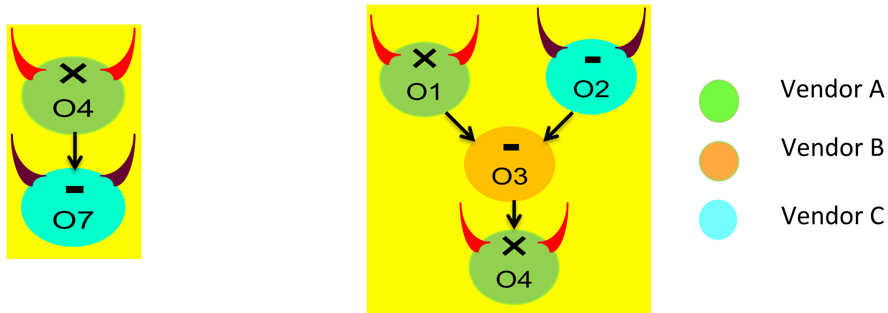
NEW YORK UNIVERSITY



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

Trojans May Collude

NEW YORK UNIVERSITY



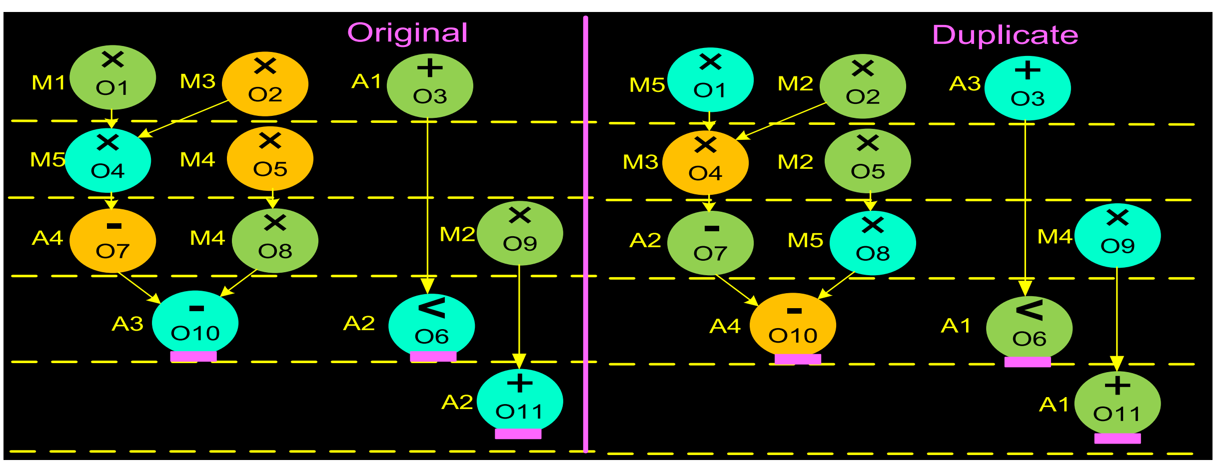
Parent-Child Parent-Parent

- 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

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

Detect Trojan: Duplicate+Diversify

NEW YORK UNIVERSITY

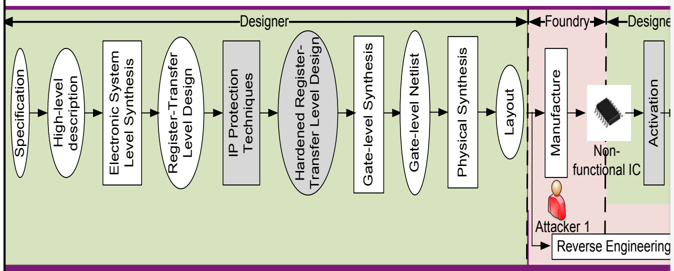


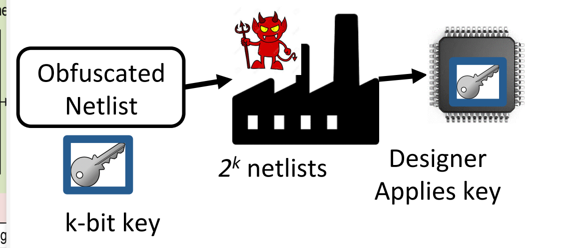
Duplicate + Diversify: 3 vendors; 3 multiplier 4 adder/compare/subtracts
Prevent Parent-Child Collusion and Parent-Parent Collusion

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

Threat: Untrusted Foundry

NEW YORK UNIVERSITY





- 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

NEW YORK UNIVERSITY

```
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

Control flow

Constants

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

Dependencies

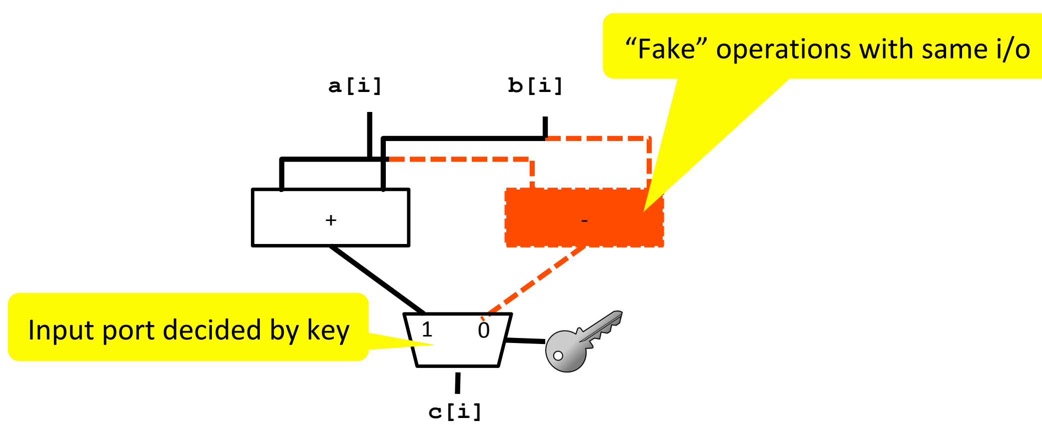
Operations

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.

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

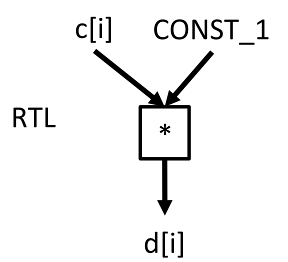


Obfuscate Constants

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;



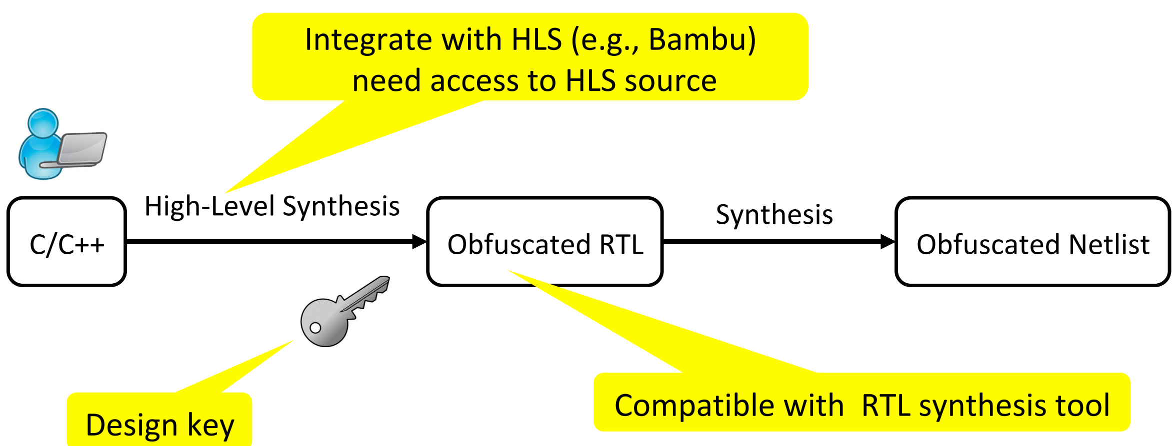
Obfuscated	Not obfuscated
Data co-efficients	Reset values
Signal extensions	Signal polarity
Mask values	

No impact on security

No impact on semantics

HLS Obfuscation

NEW YORK UNIVERSITY



Semantic Obfuscation: Branches, Dependencies, Operations, Constants

Results

NEW YORK UNIVERSITY

Design name	Algorithm Obfuscation			# of key bits
	Constant	Branch	DFG Variant	
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 constants/ key bits

Obfuscated branches

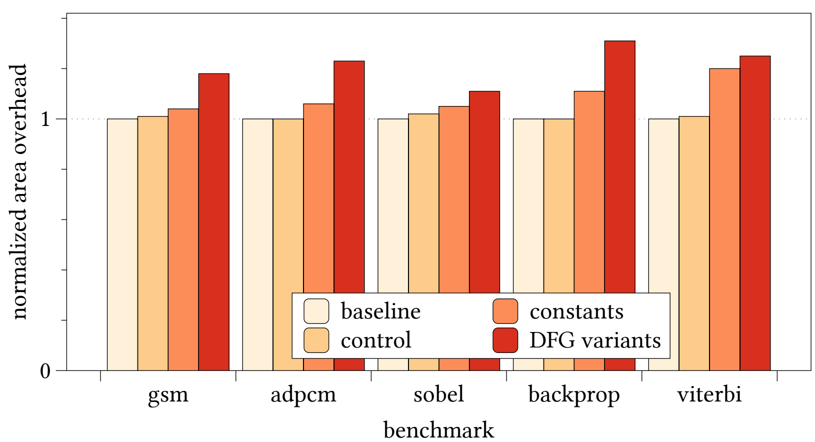
# of Basic Blocks / key bits

# of key bits

Bambu Open Source HLS (C-to-RTL HDL)

Overhead

NEW YORK UNIVERSITY

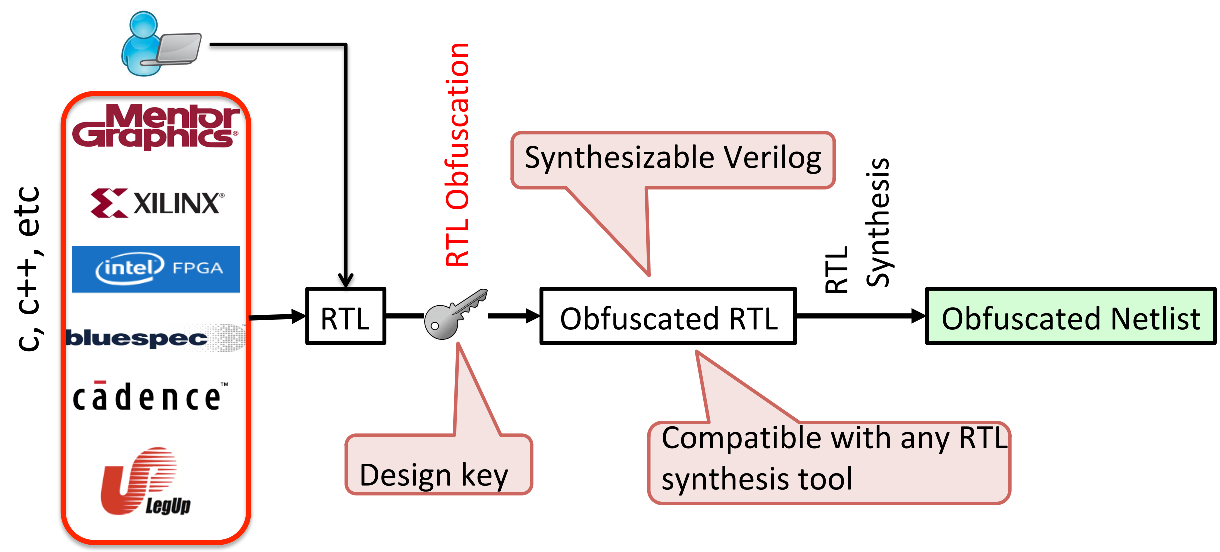


- Area overhead of each technique wrt the **baseline** version
  - Synopsys 32nm @ 500 MHz; Operation+Dependence obfuscation

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.

RTL Transformations for Security

NEW YORK UNIVERSITY



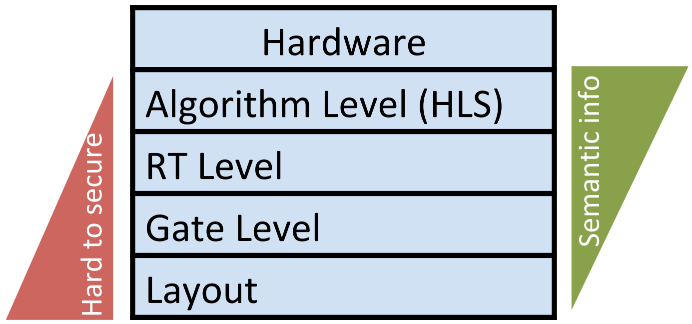
Conclusions

NEW YORK UNIVERSITY

1. High-Level is a promising level to Design Security Accelerators
   K. Basu, D. Soni, N. Mohammed, R. Karri, *NIST Post Quantum Cryptography: A Hardware Evaluation Study, Jan 2019*; iacr eprint
2. High-Level 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
3. 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
   J. Rajendran, H. Zhang, O. Sinanoglu and R. Karri, *High-level synthesis for security and trust,* IEEE Intl On-Line Testing Symposium, pp. 232-233. July 2013, doi: 10.1109/IOLTS.2013.6604087
4. HLS can be used to Watermark Designs
   C. Pilato and K. Basu and M. Shayan and F. Regazzoni and R. Karri, *High-Level Synthesis of Benevolent Trojans*, Design Automation Test in Europe Conference, pp. 1118—1123, March, 2019.
5. HLS can be used for Seamless and Meaningful Design Obfuscation
   C. Pilato, F. Reggazoni, S. Garg and R. Karri*, TAO: Techniques for Algorithm Level Obfuscation During High-Level Synthesis*, IEEE/ACM Design Automation Conference, June 2018, DOI: 10.1109/DAC.2018.8465830.
6. HLS can be used for Seamless and Meaningful Taint Propagation
   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
7. 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.
8. A Black-Hat can use High-Level Synthesis to undermine Designs (weaken crypto, drain battery, etc)
   C Pilato, K Basu, F Regazzoni, R Karri, *Black-Hat High-Level Synthesis: Myth or Reality?* IEEE Trans. VLSI, DOI: 10.1109/TVLSI.2018.2884742

Security: A Summary

NEW YORK UNIVERSITY



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

NEW YORK UNIVERSITY

?

Cell: 917 363 9703
rkarri@nyu.edu
http://cyber.nyu.edu