# Protecting Critical Integrated Circuit Intellectual Property from Theft and Counterfeiting

Ioannis Savidis

Department of Electrical and Computer Engineering
Drexel University

September 15, 2016

# Integrated Circuits and Electronics (ICE) Design and Analysis Laboratory



**Degrees:** B.S.E., Duke University
M.S., University of Rochester
Ph.D., University of Rochester (2013)

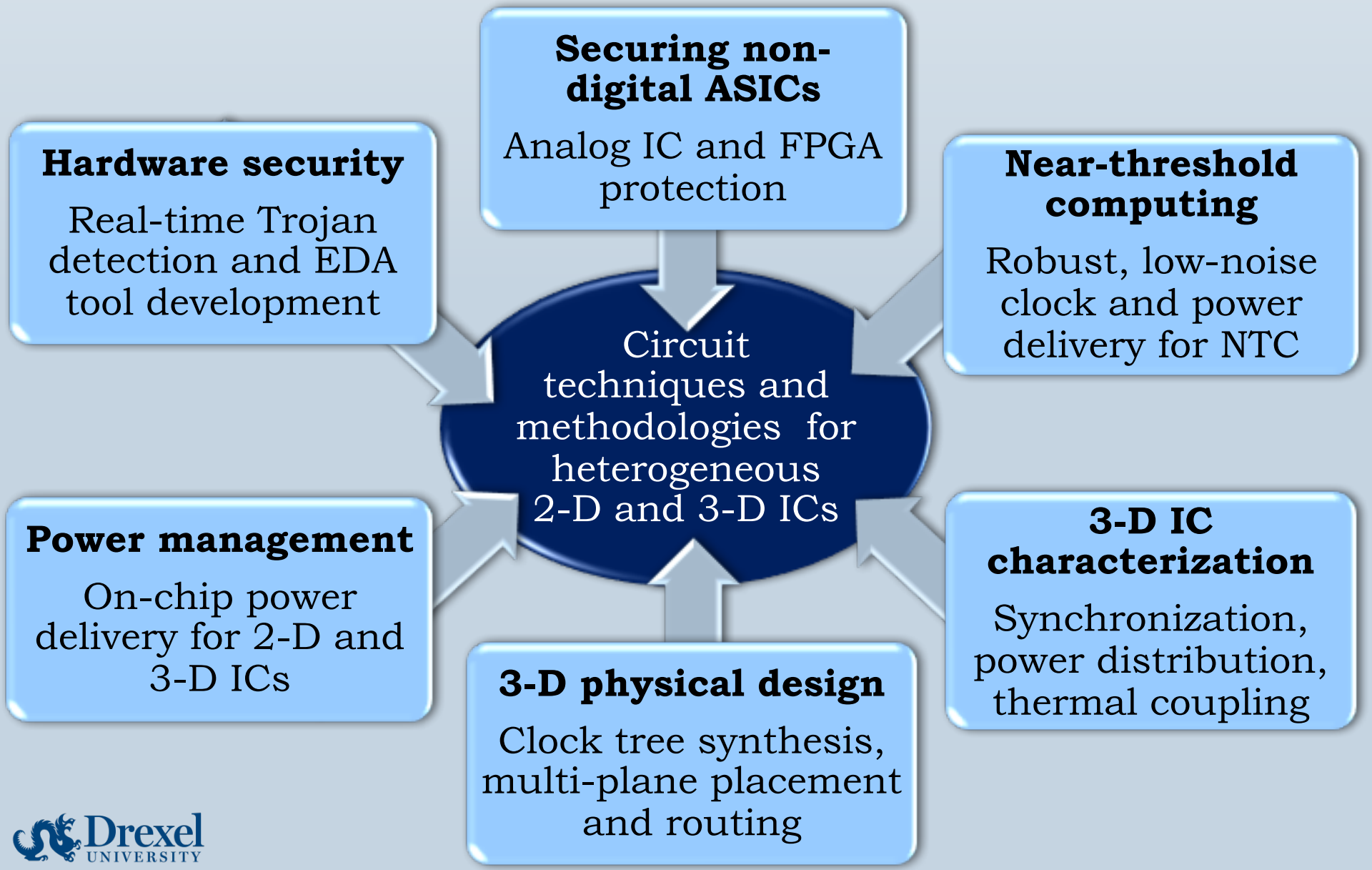
**Research Interests**
Analysis, modeling, and design methodologies for high performance digital and mixed-signal integrated circuits; Emerging integrated circuit technologies; Electrical and thermal modeling and characterization, signal and power integrity, and power and clock delivery for 3-D IC technologies; On-chip power management; Low-power circuit techniques; Algorithms and methodologies for secure IC design

**LABORATORY & TEAM**

- **Three Ph.D. students**
  - Kyle Juretus – Secure IC design
  - Divya Pathak – On-chip power management
  - Shazzad Hossain – Near-threshold computing for low power applications
- **Two M.S. students**
  - Isuru Daulagala – Clock tree synthesis for 3-D integrated circuits
  - Vaibhav Venugopal Rao – Analog IC IP protection
- **One B.S. student**
  - Akash Rai Sinha
- **2,000 square feet of dedicated research space**
- **Access to leading CAD software packages:** Cadence (Virtuoso, Encounter, assure), Synopsys (Primetime, Hspice, Taurus), and Mentor Graphics (Calibre)

# Research Foci





# Presentation Outline



- Introduction to IP Protection
- Logic Encryption
- Reduced Overhead Logic Obfuscation
- Design for Trust
- Conclusions

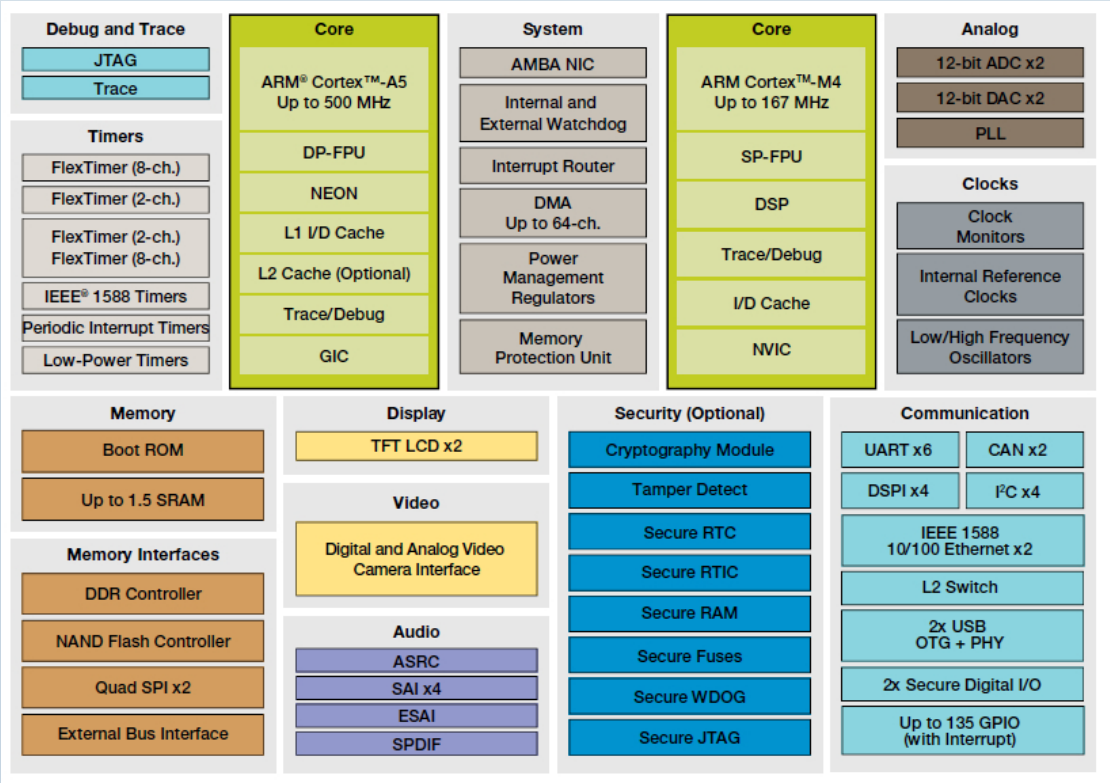
# Presentation Outline



- Introduction to IP Protection
  - ***Security Threats of Horizontal IC Design Flow***
  - Countermeasures to Security Threats
- Logic Encryption
- Reduced Overhead Logic Obfuscation
- Design for Trust
- Conclusions

# Increased SoC Complexity



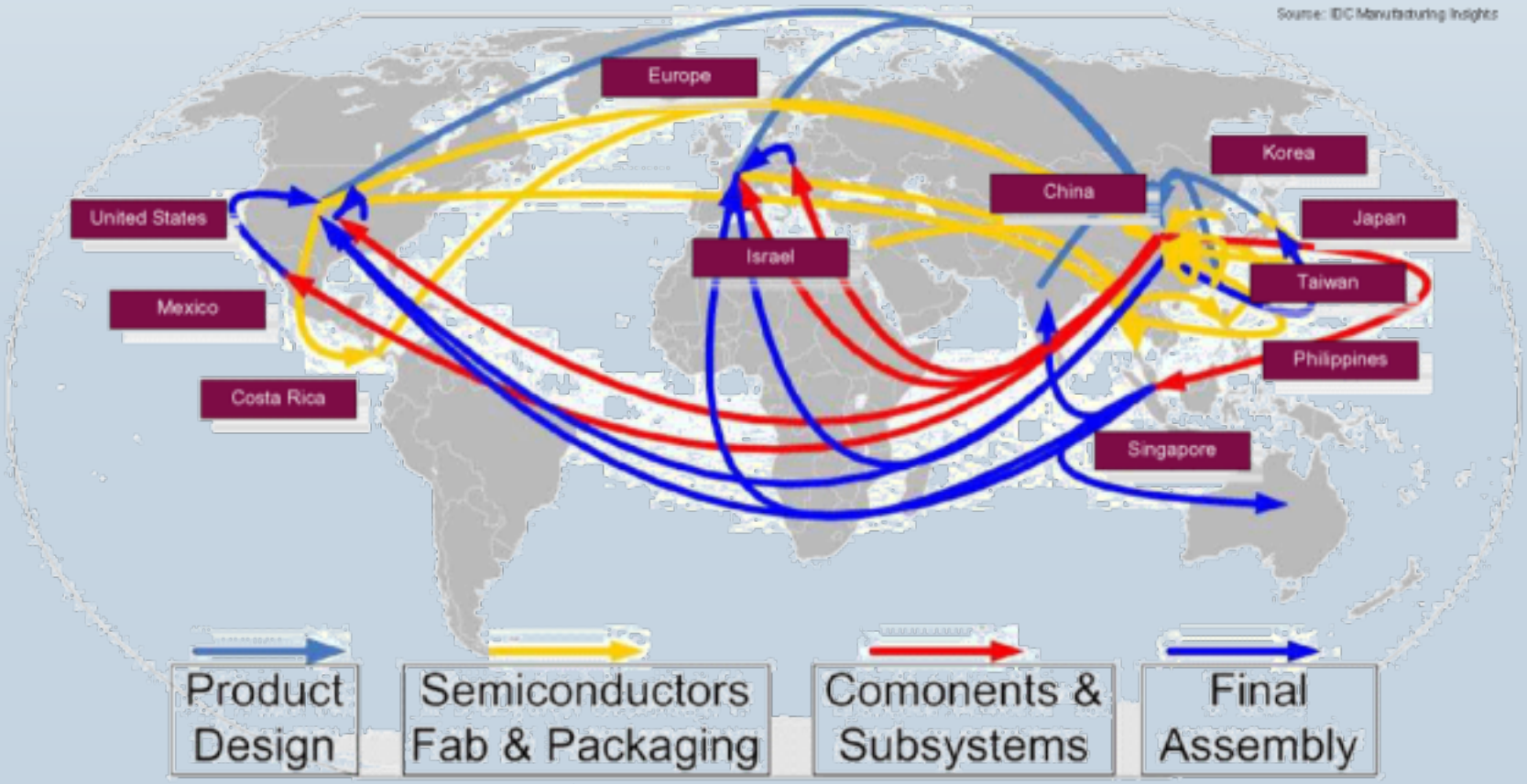



* Courtesy of Freescale (i.MX6 SoC)

- Increase in the number and type of circuit blocks integrated into a single system-on-chip
- Multi-vendor IP integration
  - ARM, OpenCores, Altera, Synopsys, Cadence, S3 Group, ….

# Globalization of the IC Supply Chain






* Courtesy of VentureOutsource.com

- Globalization of the design, fabrication, packaging, and test processes

# Threats at Different Stages of IC Production

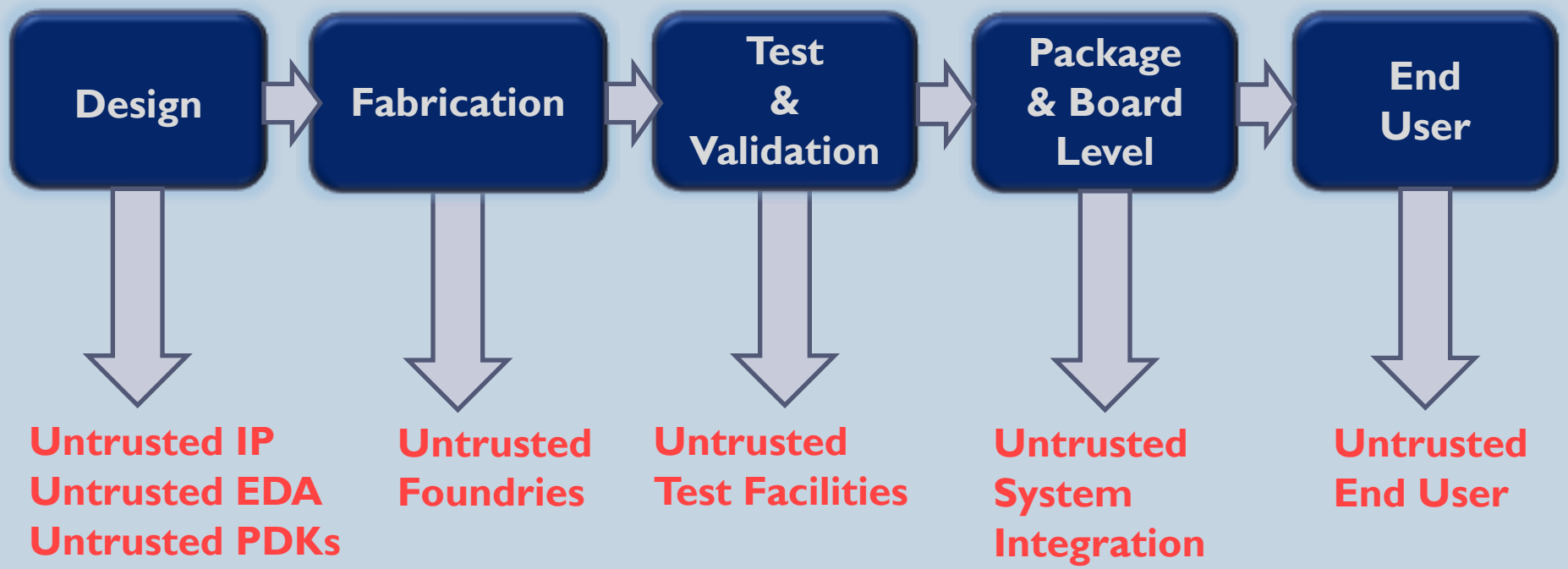


**EXPANSION OF 3RD PARTY USAGE IN IC DESGIN FLOW:**

- Rapid technology change
- Expanded capabilities
- Design time constraints

**RESULTING IC HARDWARE SECURITY THREATS**

- IC Counterfeiting
- IC overproduction
- IP theft
- Insertion of HW Trojans (malicious ctry



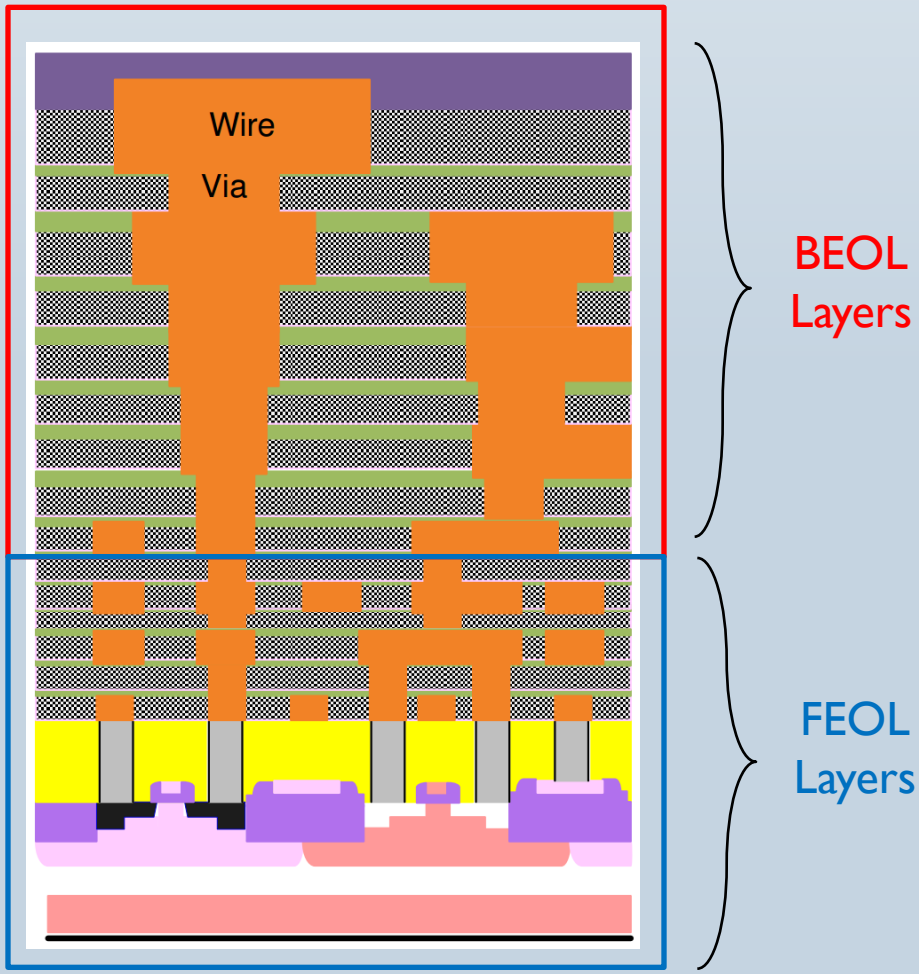
# Presentation Outline



- Introduction to IP Protection
  - Security Threats of Horizontal IC Design Flow
  - ***Countermeasures to IP Theft and Counterfeiting***
- Logic Encryption
- Reduced Overhead Logic Obfuscation
- Design for Trust
- Conclusions

# Split Manufacturing of Integrated Circuits






- Layout split into front end of line layers (FEOL) and back end of line layers (BEOL)
- Prevents a single foundry from gaining full access to the IC
- Limitations
  - End-user able to reverse engineer IC
  - Does not prevent black-box usage of a design

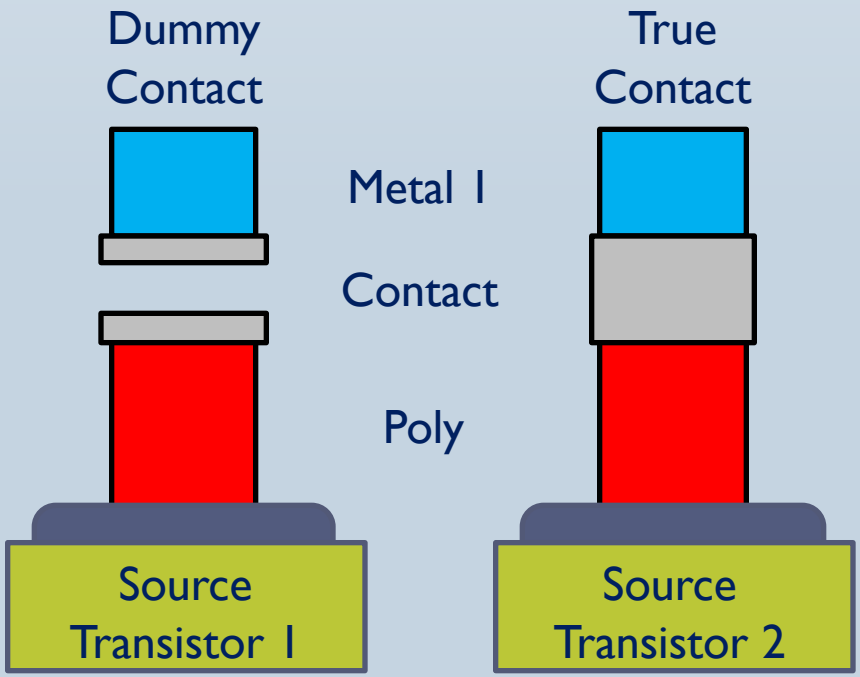


J. Rajendran, O. Sinanoglu, and R. Karri, "Is Split Manufacturing Secure?," *in the Proceedings of the Design, Automation & Test in Europe Conference & Exhibition*, pp. 1259-1264, 2013.

# IC Camouflaging



- Disguises layer connections with dummy contacts
  - Increases difficulty of reverse engineering the design
- Limitations
  - Foundry has access to information required to produce IC
    - Reverse engineer IC
    - Overproduce ICs
  - Design can be used as black-box




J. Rajendran, M. Sam, O. Sinanoglu, and R. Karri "Security Analysis of Integrated Circuit Camouflaging," *in the proceedings of the ACM SIGSAC conference on Computing & communications security*, pp. 709-720, 2013.

# Presentation Outline

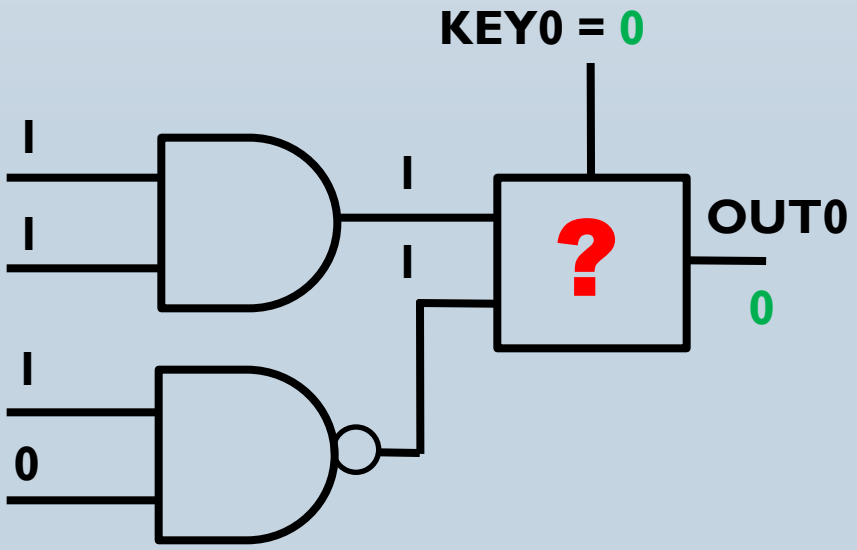


- Introduction to Hardware Security
- ***Logic Encryption***
  - Fault Analysis-Based Logic Encryption
  - Reconfigurable Logic Barriers (LUTs)
  - Evaluating the Security of Logic Encryption
- Reduced Overhead Logic Obfuscation
- Design for Trust
- Conclusions

# Logic Encryption



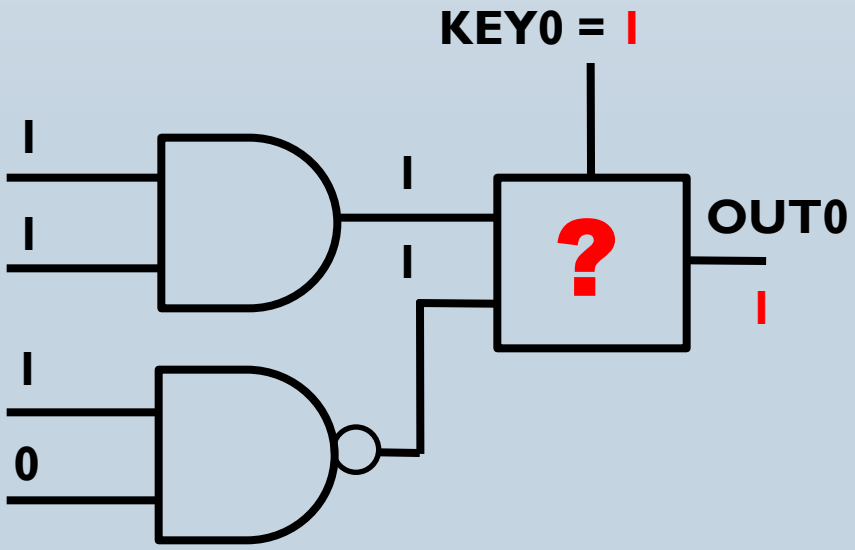
- Inserts key gates into a design
  - Require key for correct operation
- Conceals design information from foundry and untrusted end-users
- Prevents black-box usage
- Increases difficulty to insert a hardware Trojan



# Logic Encryption



- Inserts key gates into a design
  - Require key for correct operation
- Conceals design information from foundry and untrusted end-users
- Prevents black-box usage
- Increases difficulty to insert a hardware Trojan



# Presentation Outline

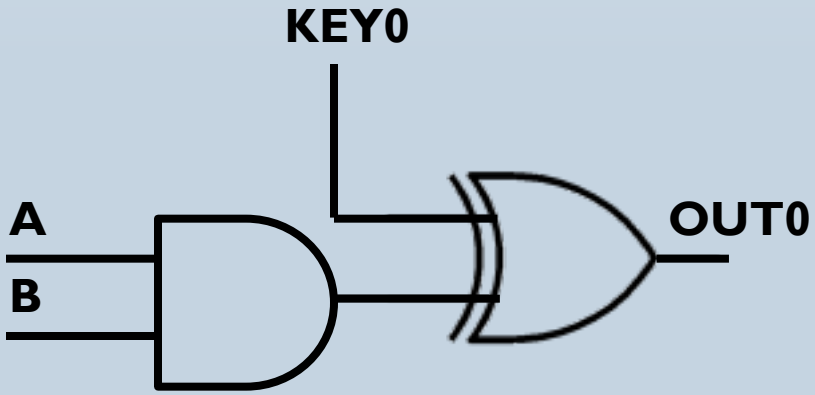


- Introduction to Hardware Security
- *Logic Encryption*
  - ***Fault Analysis-Based Logic Encryption***
  - Reconfigurable Logic Barriers (LUTs)
  - Evaluating the Security of Logic Encryption
- Reduced Overhead Logic Obfuscation
- Design for Trust
- Conclusions

# XOR Based Logic Encryption



- Add XOR to output of gate
- Requires application of key (KEY0) to function correctly
- XOR acts as buffer when KEY0 = 0
- XOR acts as inverter when KEY0 = 1
- Inverters are added before/after the XOR gate to increase the difficulty of determining KEY0

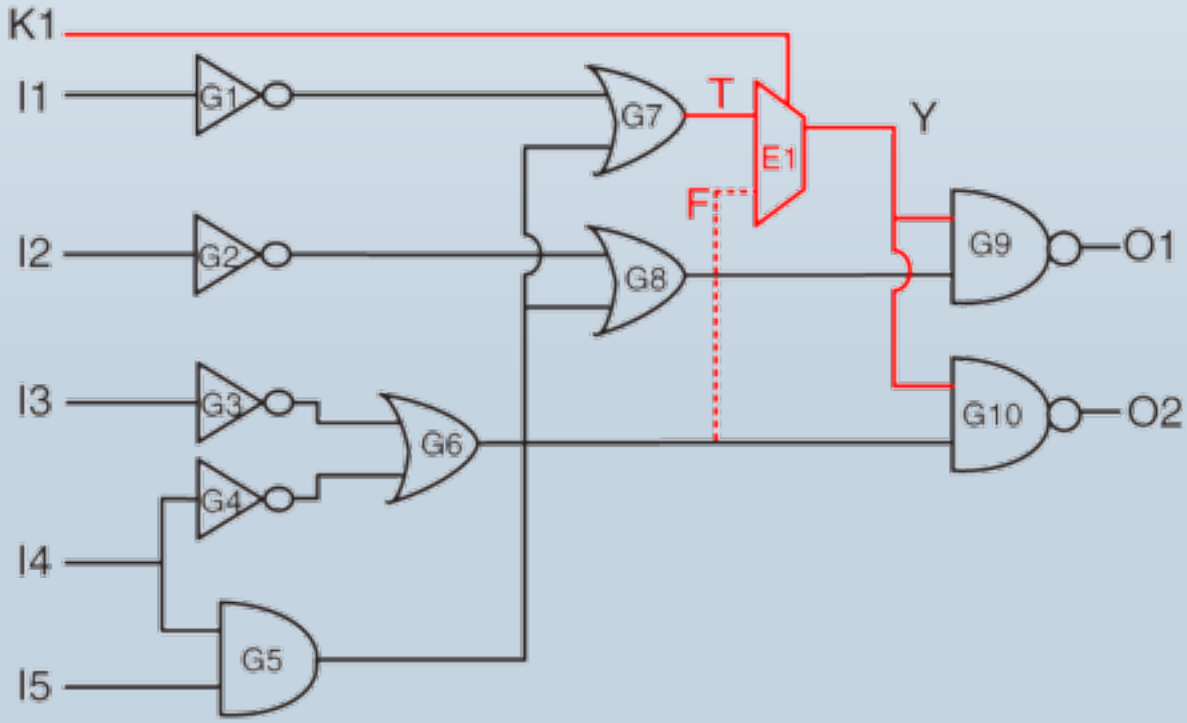





J. Rajendran, H. Zhang, C. Zhang, G. Rose, Y. Pino, O. Sinanoglu, and R. Karri, "Fault Analysis-Based Logic Encryption," IEEE Transactions on Computers, Vol. 64, No. 2, pp. 410–424, February 2015.

# 2x1 MUX Based Encryption





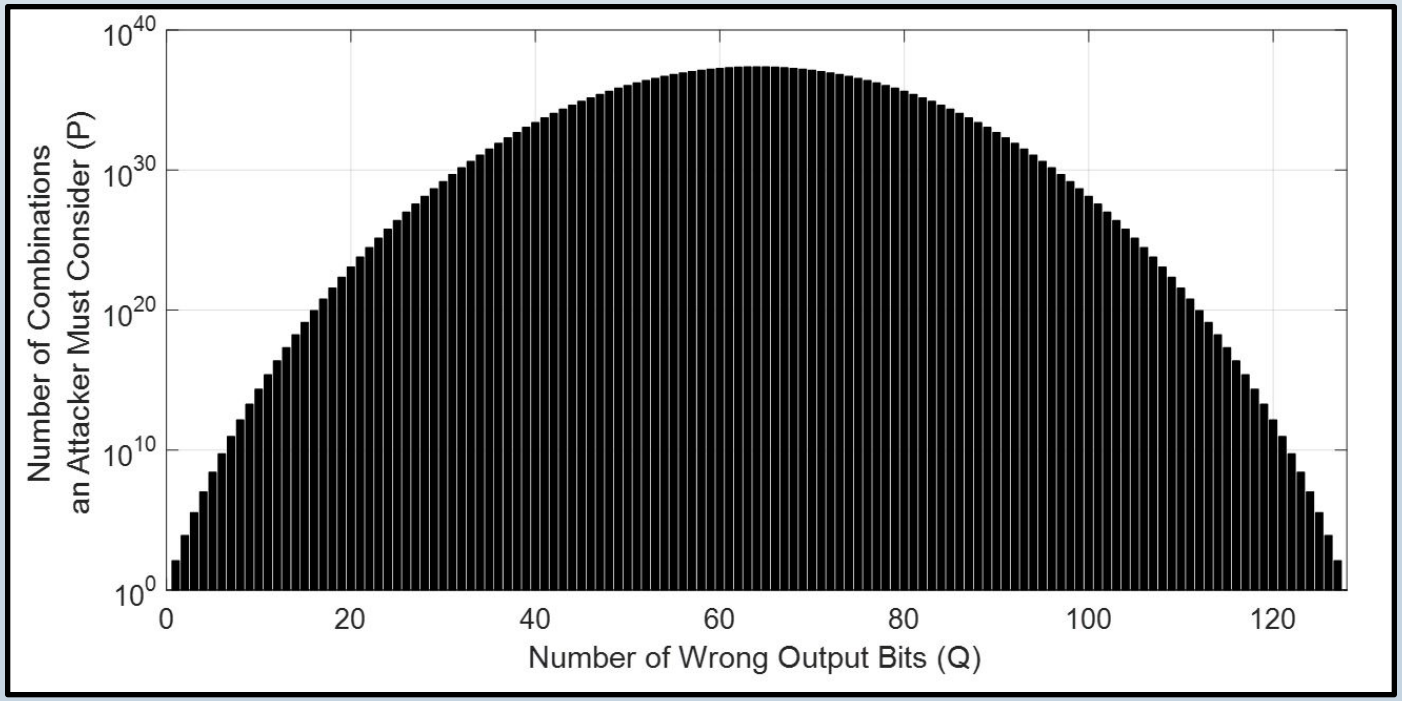

- Insert 2x1 MUX after gate
- Connect one input to original gate
- Other input is the “false” input from another net in the design



J. Rajendran, H. Zhang, C. Zhang, G. Rose, Y. Pino, O. Sinanoglu, and R. Karri, “Fault Analysis-Based Logic Encryption,” IEEE Transactions on Computers, Vol. 64, No. 2, pp. 410–424, February 2015.

# 50% Hamming Distance Metric






- Maximize output ambiguity seen by adversary
  - Incorrect key produces correct output
  - Incorrect key produces complete corruption
- Modeled as $N$ (number of outputs) choose $Q$ (number of incorrect outputs) probability
  - Maximized when $Q = N/2$

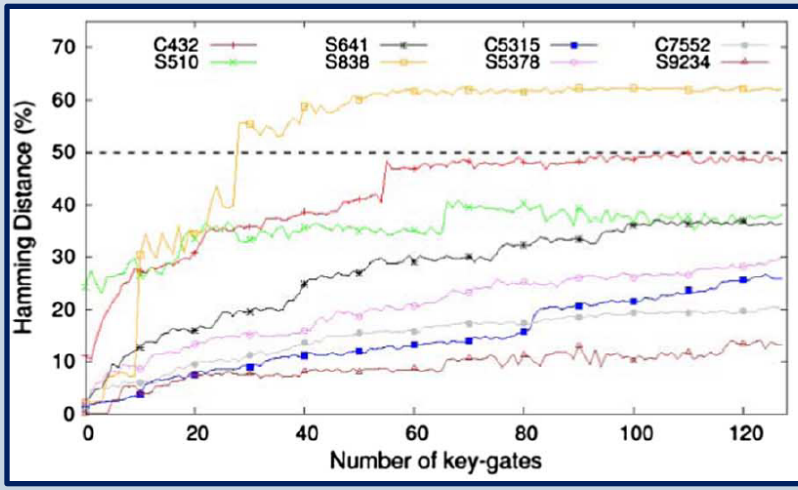


J. Rajendran, H. Zhang, C. Zhang, G. Rose, Y. Pino, O. Sinanoglu, and R. Karri, "Fault Analysis-Based Logic Encryption," IEEE Transactions on Computers, Vol. 64, No. 2, pp. 410–424, February 2015.

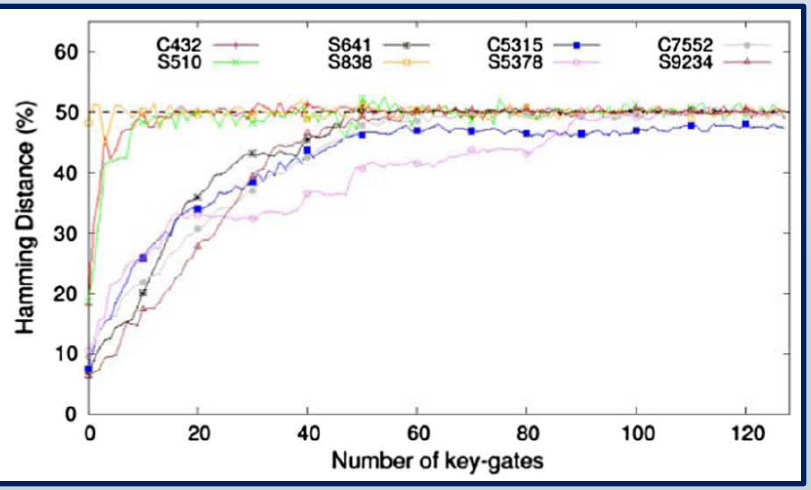
# Evaluation of Gate Selection Algorithm



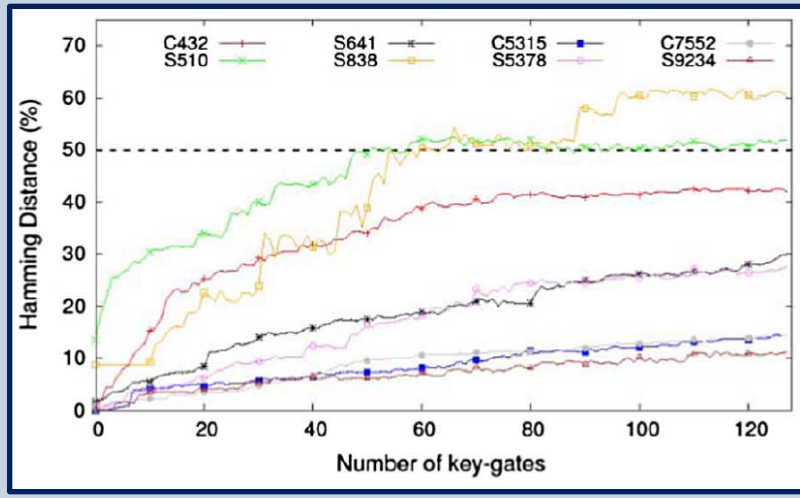
Random XOR Selection



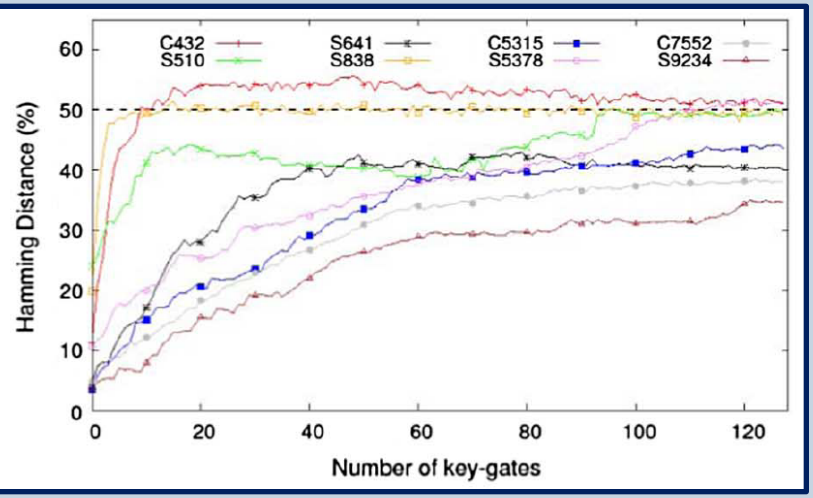

Fault-Analysis Based XOR Selection




Random 2x1 MUX Selection




Fault-Analysis Based 2x1 MUX Selection






J. Rajendran, H. Zhang, C. Zhang, G. Rose, Y. Pino, O. Sinanoglu, and R. Karri, "Fault Analysis-Based Logic Encryption," IEEE Transactions on Computers, Vol. 64, No. 2, pp. 410–424, February 2015.

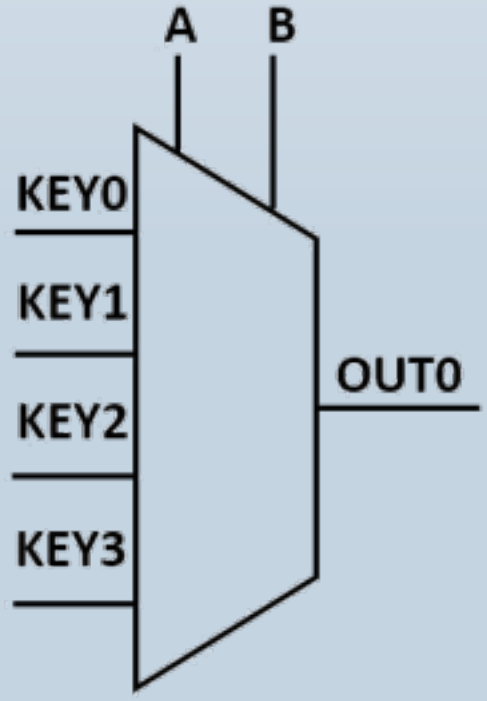
# Presentation Outline



- Introduction to Hardware Security
- *Logic Encryption*
  - Fault Analysis-Based Logic Encryption
  - ***Reconfigurable Logic Barriers (LUTs)***
  - Evaluating the Security of Logic Encryption
- Reduced Overhead Logic Obfuscation
- Design for Trust
- Conclusions

# LUT Based Logic Encryption






- Replace gate with LUT
  - Removes the information of the original gate
- Key values determine the functionality of the gate
- Able to implement any 2-input function (more key combinations)

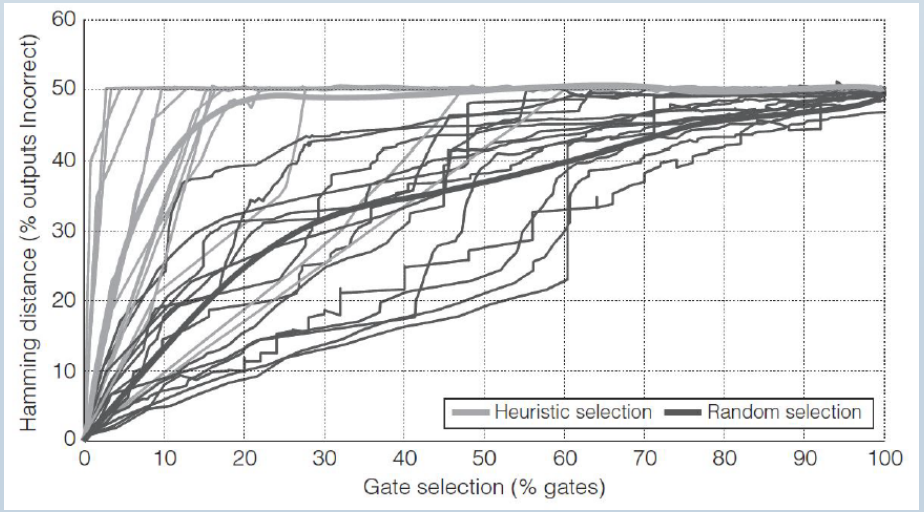


A. Baumgarten, A. Tyagi, and J. Zambreno, "Preventing IC Piracy Using Reconfigurable Logic Barriers," IEEE Design and Test of Computers, Vol. 27, No. 1, pp. 66–75, February 2010.

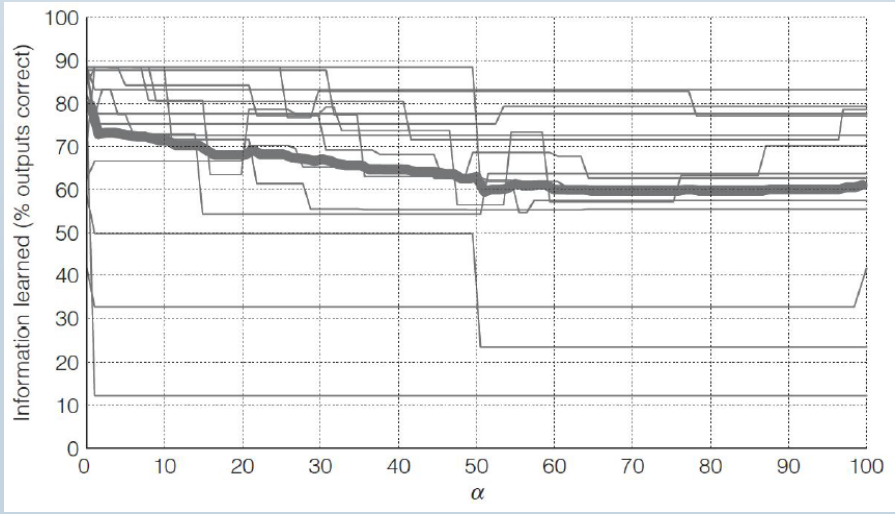
# Evaluation of ODC and Cut Height Selection



## Analysis of Hamming Distance




## Analysis of α and Security




- ODC cut based selection achieves 50% Hamming distance much faster than random selection
- The $\alpha$ value of 50 reveals the least amount of information when attacking a single LUT



A. Baumgarten, A. Tyagi, and J. Zambreno, "Preventing IC Piracy Using Reconfigurable Logic Barriers," IEEE Design and Test of Computers, Vol. 27, No. 1, pp. 66–75, February 2010.

# Presentation Outline



- Introduction to Hardware Security
- *Logic Encryption*
  - Fault Analysis-Based Logic Encryption
  - Preventing IC Theft Using Reconfigurable Logic Barriers
  - ***Evaluating the Security of Logic Encryption***
- Reduced Overhead Logic Obfuscation
- Design for Trust
- Conclusions

# Attack Model for Logic Encryption



- Adversary has access to activated IC
  - Use as black-box to determine correct input/output combinations
- Utilize SAT solver to determine valid solutions
  - Based on correct input/output constraints
- Divide keys in equivalence classes
  - For all input/output constraints that match thus far
- Determine distinguishing input patterns to eliminate key values



P. Subramanyan, S. Ray, and S. Malik, "Evaluating the Security of Logic Encryption Algorithms," Proceedings of the IEEE International Symposium on Hardware Oriented Security and Trust, pp. 137 – 143, May 2015.

# Overview of Attack Algorithm



**Algorithm 4:** Logic decryption algorithm

**Input:** C and eval

**Output:** $\overrightarrow{K_C}$

$i := 1;$

$F_1 = C(\overrightarrow{X}, \overrightarrow{K_1}, \overrightarrow{Y_1}) \wedge C(\overrightarrow{X}, \overrightarrow{K_2}, \overrightarrow{Y_2});$

**while** $sat[F_i \wedge (\overrightarrow{Y_1} \neq \overrightarrow{Y_2})]$ **do**

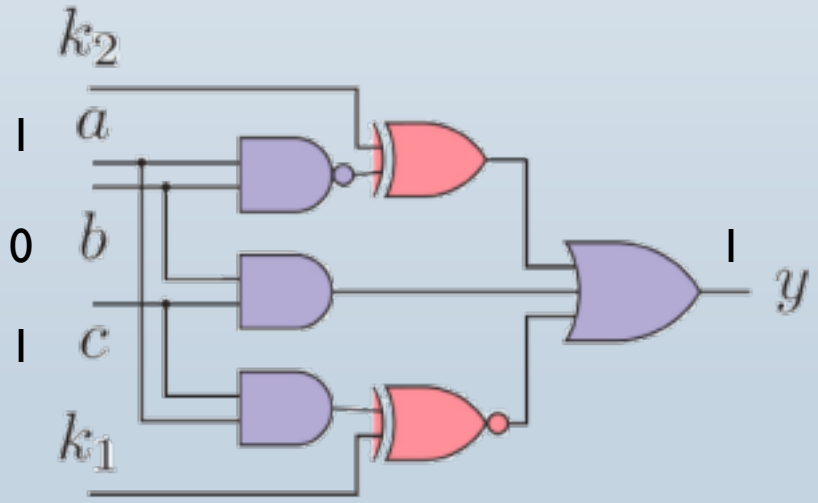
**1)** $\overrightarrow{X_i^d} := sat\_assignment_{\overrightarrow{X}}[F_i \wedge (\overrightarrow{Y_1} \neq \overrightarrow{Y_2})];$

**2)** $\overrightarrow{Y_i^d} := eval(\overrightarrow{X_i^d});$

**3)** $F_{i+1} := F_i \wedge C(\overrightarrow{X_i^d}, \overrightarrow{K_1}, \overrightarrow{Y_i^d}) \wedge C(\overrightarrow{X_i^d}, \overrightarrow{K_2}, \overrightarrow{Y_i^d});$
$i := i + 1;$

**end**

$\overrightarrow{K_C} := sat\_assignment_{\overrightarrow{K_1}}(F_i)$




1) $\overrightarrow{X_1^d}$ *generated as* (1,0,1)

2) $\overrightarrow{Y_1^d}$ *evaluated as* 1



P. Subramanyan, S. Ray, and S. Malik, "Evaluating the Security of Logic Encryption Algorithms," Proceedings of the IEEE International Symposium on Hardware Oriented Security and Trust, pp. 137 – 143, May 2015.

# Overview of Attack Algorithm



**Algorithm 4:** Logic decryption algorithm

**Input:** C and eval

**Output:** $\overrightarrow{K_C}$

$i := 1;$

$F_1 = C(\overrightarrow{X}, \overrightarrow{K_1}, \overrightarrow{Y_1}) \wedge C(\overrightarrow{X}, \overrightarrow{K_2}, \overrightarrow{Y_2});$

**while** $sat[F_i \wedge (\overrightarrow{Y_1} \neq \overrightarrow{Y_2})]$ **do**

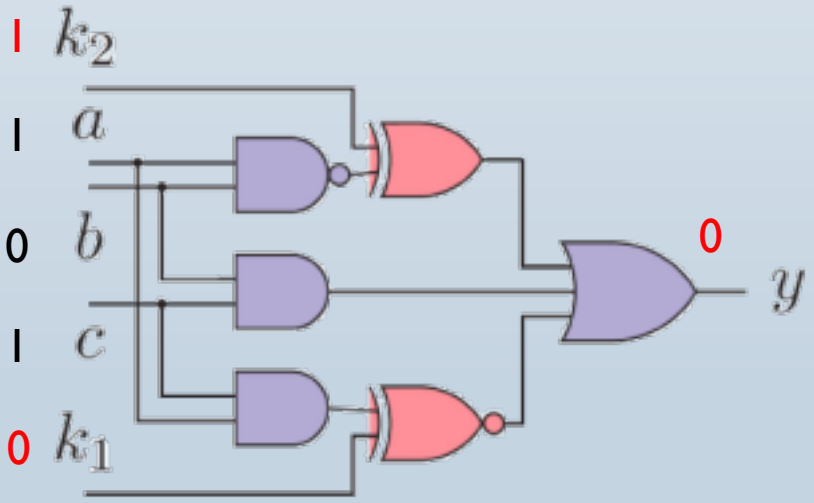
**1)** $\overrightarrow{X_i^d} := sat\_assignment_{\overrightarrow{X}}[F_i \wedge (\overrightarrow{Y_1} \neq \overrightarrow{Y_2})];$

**2)** $\overrightarrow{Y_i^d} := eval(\overrightarrow{X_i^d});$

**3)** $F_{i+1} := F_i \wedge C(\overrightarrow{X_i^d}, \overrightarrow{K_1}, \overrightarrow{Y_i^d}) \wedge C(\overrightarrow{X_i^d}, \overrightarrow{K_2}, \overrightarrow{Y_i^d});$
$i := i + 1;$

**end**

$\overrightarrow{K_C} := sat\_assignment_{\overrightarrow{K_1}}(F_i)$




1) $\overrightarrow{X_1^d}$ *generated as* (1,0,1)

2) $\overrightarrow{Y_1^d}$ *evaluated as* 1

3) $(k1, k2) = (0,1)$ *eliminated as valid key*



P. Subramanyan, S. Ray, and S. Malik, "Evaluating the Security of Logic Encryption Algorithms," Proceedings of the IEEE International Symposium on Hardware Oriented Security and Trust, pp. 137 – 143, May 2015.

# Overview of Attack Algorithm



**Algorithm 4:** Logic decryption algorithm

**Input:** C and eval

**Output:** $\overrightarrow{K_C}$

$i := 1;$

$F_1 = C(\overrightarrow{X}, \overrightarrow{K_1}, \overrightarrow{Y_1}) \wedge C(\overrightarrow{X}, \overrightarrow{K_2}, \overrightarrow{Y_2});$

**while** $sat[F_i \wedge (\overrightarrow{Y_1} \neq \overrightarrow{Y_2})]$ **do**

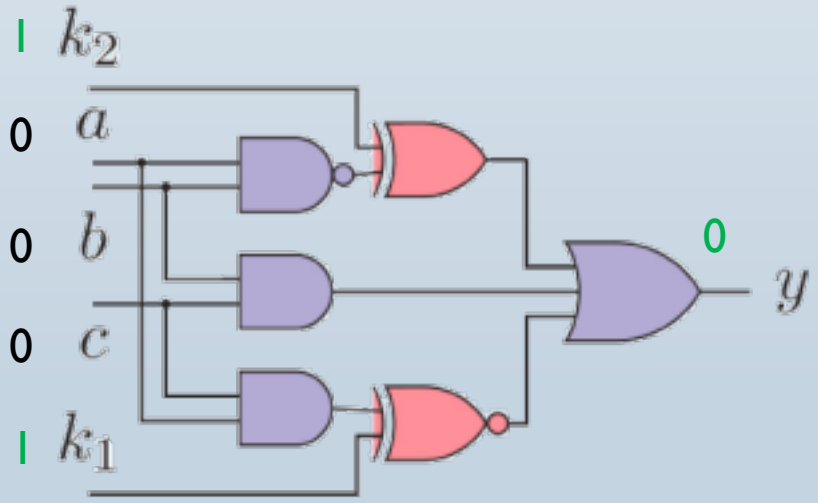
1) $\overrightarrow{X_i^d} := sat\_assignment_{\overrightarrow{X}}[F_i \wedge (\overrightarrow{Y_1} \neq \overrightarrow{Y_2})];$

2) $\overrightarrow{Y_i^d} := eval(\overrightarrow{X_i^d});$

3) $F_{i+1} := F_i \wedge C(\overrightarrow{X_i^d}, \overrightarrow{K_1}, \overrightarrow{Y_i^d}) \wedge C(\overrightarrow{X_i^d}, \overrightarrow{K_2}, \overrightarrow{Y_i^d});$
$i := i + 1;$

**end**

$\overrightarrow{K_C} := sat\_assignment_{\overrightarrow{K_1}}(F_i)$




## Second Iteration

1) $\overrightarrow{X_2^d}$ *generated as* (0,0,0)

2) $\overrightarrow{Y_1^d}$ *evaluated as* 0

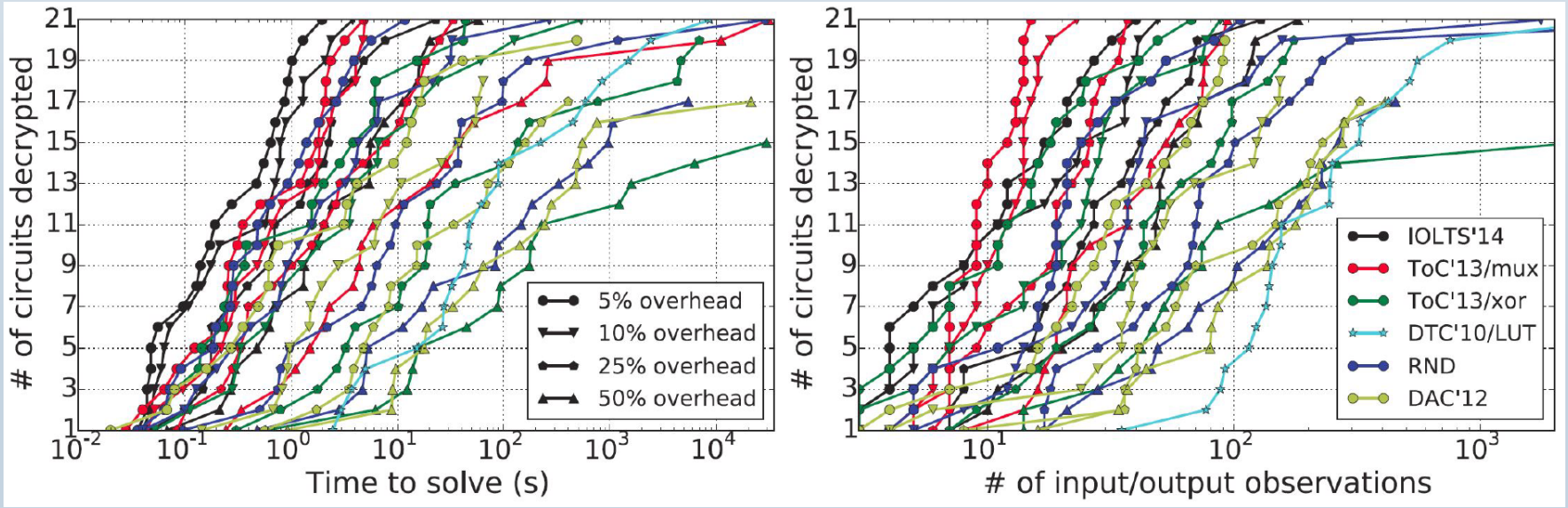
3) $(k1, k2) = (1,1)$
*only valid key*



P. Subramanyan, S. Ray, and S. Malik, "Evaluating the Security of Logic Encryption Algorithms," Proceedings of the IEEE International Symposium on Hardware Oriented Security and Trust, pp. 137 – 143, May 2015.

# Susceptibility of Logic Encryption to SAT Attack





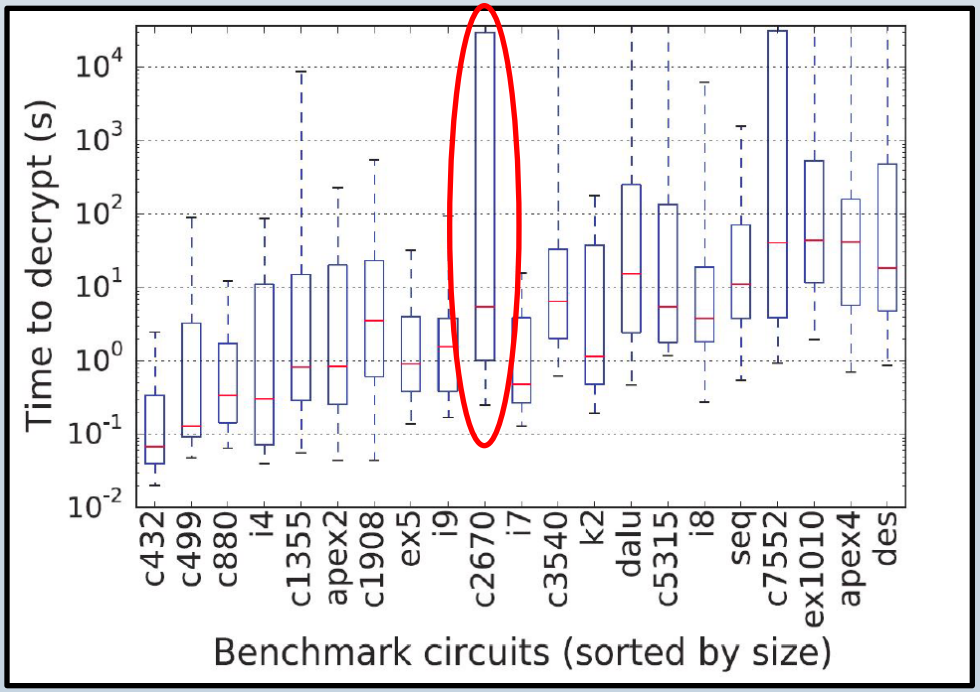

- XOR/XNOR encryption takes the longest time to solve and number of input/output combinations
- LUT is one the best performing in terms of number of input/output observations
  - Overhead is not constrained
- 2x1 MUX does not perform well



P. Subramanyan, S. Ray, and S. Malik, "Evaluating the Security of Logic Encryption Algorithms," Proceedings of the IEEE International Symposium on Hardware Oriented Security and Trust, pp. 137 – 143, May 2015.

# Strength of Circuit Security to SAT Attack






- Smaller benchmark C2670 has one of the highest decryption times
- AND gate tree found within design
  - Requires unique distinguishing input for each key value



P. Subramanyan, S. Ray, and S. Malik, "Evaluating the Security of Logic Encryption Algorithms," Proceedings of the IEEE International Symposium on Hardware Oriented Security and Trust, pp. 137 – 143, May 2015.

# Presentation Outline



- Introduction to Hardware Security
- Logic Encryption
- Reduced Overhead Logic Obfuscation
  - ***Overhead of Logic Encryption***
  - Gate Level Logic Encryption
- Design for Trust
- Conclusions

# Standard Cell Characterization



- 180 nm technology node characterized
- Standard cells designed for same drive strength
- 5 fF load driven by each cell

| Standard Cell | Prop. Delay (ps) | Avg. Power (nW) | Avg. Leakage Power (pW) | Area ($\mu m^2$) |
|---|---|---|---|---|
| AND | 69.79 | 70.40 | 192.5 | 30.73 |
| NAND | 36.71 | 72.67 | 149.7 | 23.25 |
| OR | 92.90 | 64.52 | 252.3 | 30.73 |
| NOR | 42.09 | 96.85 | 193.6 | 23.25 |
| XOR | 91.23 | 148.0 | 455.3 | 45.69 |
| XNOR | 106.7 | 204.6 | 527.6 | 41.95 |
| **Average** | **73.24** | **109.5** | **295.2** | **32.60** |

# XOR Overhead



- XOR-Based Logic Encryption
  - 140.2% Average Performance Overhead
  - 85.45% Average Power, 233.9% Average Leakage Power Overhead
  - 124.9% Average Area Overhead

| Standard Cell | Prop. Delay | Avg. Power | Avg. Leakage Power | Area |
|---|---|---|---|---|
| AND | 116.8% | 113.6% | 250.0% | 107.4% |
| NAND | 247.6% | 95.68% | 288.1% | 174.1% |
| OR | 69.54% | 170.6% | 162.2% | 107.4% |
| NOR | 219.1% | 73.98% | 235.2% | 174.1% |
| XOR | 99.28% | 48.18% | - | 84.95% |
| XNOR | 88.66% | 10.61% | - | 101.4% |
| **Average** | **140.2%** | **85.45%** | **233.9%** | **124.9%** |

# LUT Overhead



- LUT-Based Logic Encryption
  - 99.82% Average Performance Overhead
  - 73.95% Average Power, 368.2% Average Leakage Power Overhead
  - 197.2% Average Area Overhead

| Standard Cell | Prop. Delay | Avg. Power | Avg. Leakage Power | Area |
|---|---|---|---|---|
| AND | 75.53% | 107.4% | 365.3% | 194.8% |
| NAND | 239.4% | 116.0% | 462.3% | 289.6% |
| OR | 30.03% | 120.4% | 242.2% | 194.8% |
| NOR | 201.3% | 63.96% | 402.9% | 289.6% |
| XOR | 35.92% | 29.59% | - | 98.25% |
| XNOR | 16.78% | 6.305% | - | 115.9% |
| **Average** | **99.82%** | **73.95%** | **368.2%** | **197.2%** |

# Presentation Outline

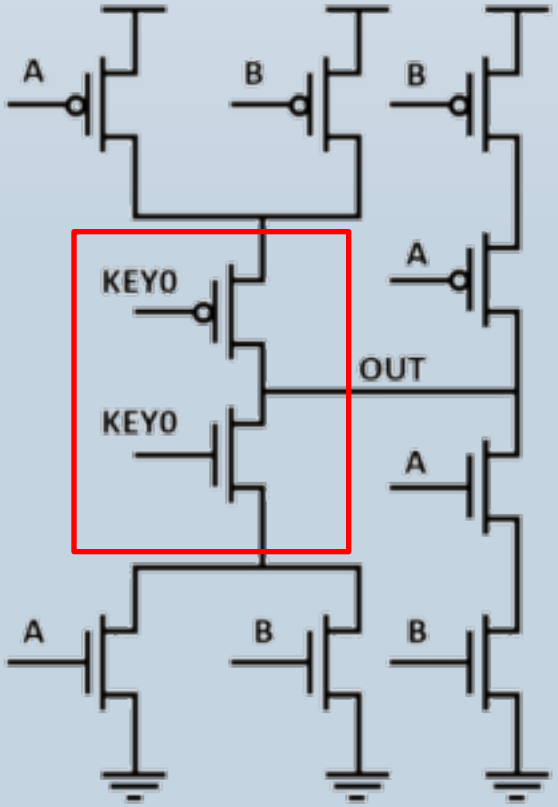


- Introduction to Hardware Security
- Logic Encryption
- Reduced Overhead Logic Obfuscation
  - Overhead of Logic Encryption
  - ***Gate Level Logic Encryption***
- Design for Trust
- Conclusions

# Gate Level Logic Encryption



- Design gates with inbuilt encryption
- Significantly lower performance, power, and area overheads of logic encryption
- Allow for more IC designs to implement security without hindering design constraints

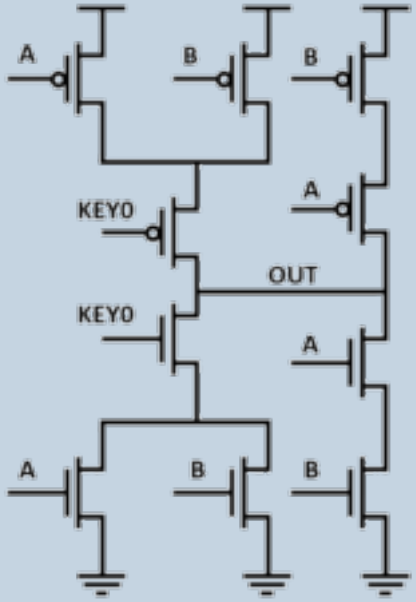


# Method 1: Stack-Based Topology



- Transistors connected to output node with key input in order to control functionality
- Benefits from functionality that does not require replicated high and low connections

**NAND/NOR Stack-Based Topology**



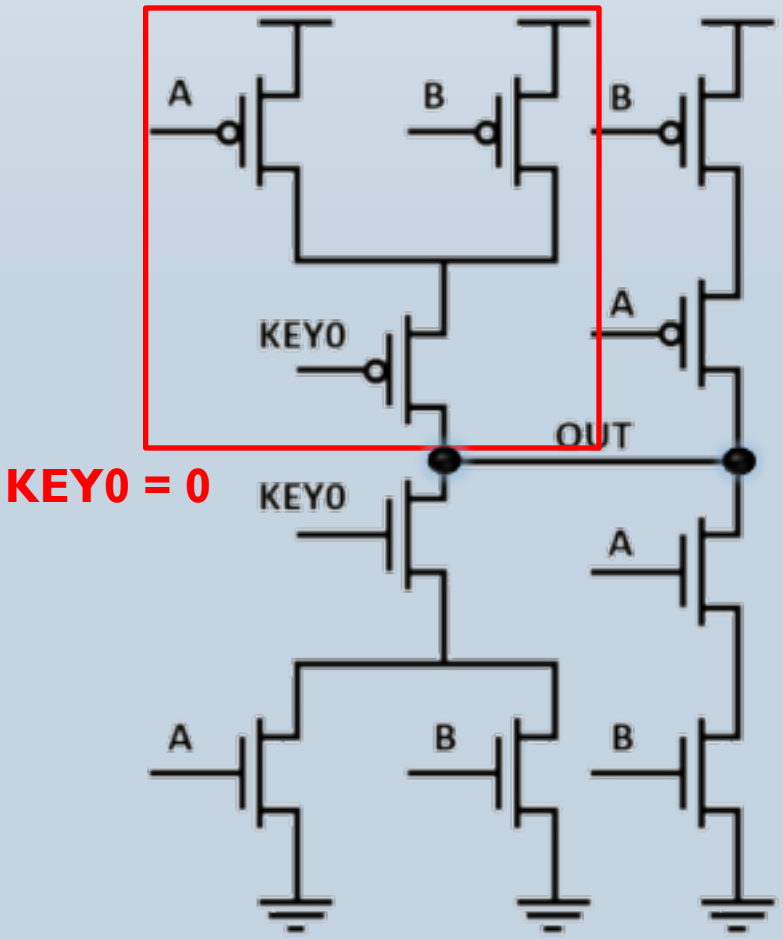

**Truth Table**

| KEY0 | A | B | OUT |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |

# Method 1: Stack-Based Topology



**AND/OR Stack-Based Topology**




**Truth Table**

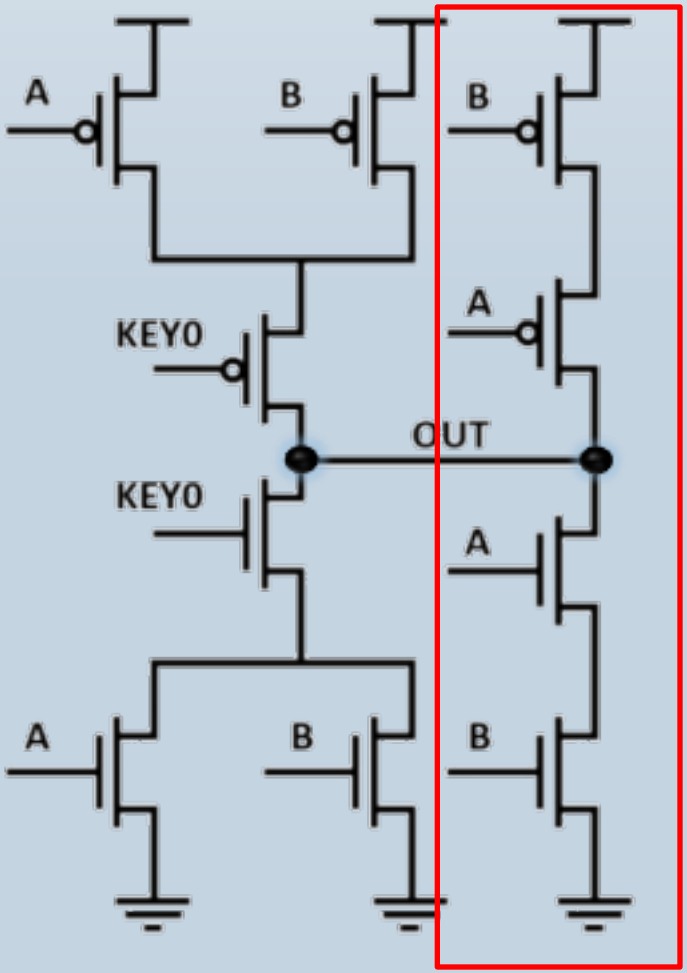
| KEY0 | A | B | OUT |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |

- Input to KEY0 determines the functionality of the gate
- Either the pull-up network (KEY0 = 0) or the pull-down network (KEY0 = 1) is connected to the output node OUT

# Stack-Based Topology Shared Functionality



**AND/OR Stack-Based Topology**




**Truth Table**

| KEY0 | A | B | OUT |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |

- Common functionality between both the AND and OR gates are shared within the encrypted gate
- Sharing common input/output values reduces the complexity of the encrypted gate
  - Reduces area and power consumption

# Method 2-Transmission Gate Topology

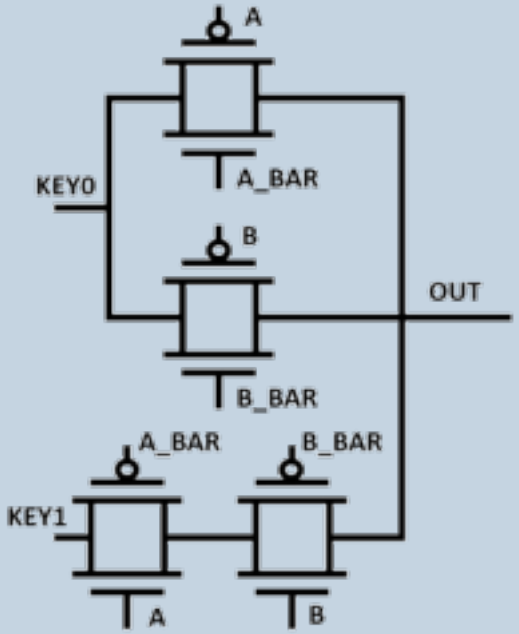


- Transmission gates used to pass keys on certain logic conditions
- Reduces cost of replicated logic
- Removes additional level of logic due to key input transistor

**Truth Table**

| KEY0 | KEY1 | A | B | OUT |
|---|---|---|---|---|
| 0 | 0 | X | X | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | X | X | 1 |

**AND/NAND Transmission Gate Topology**



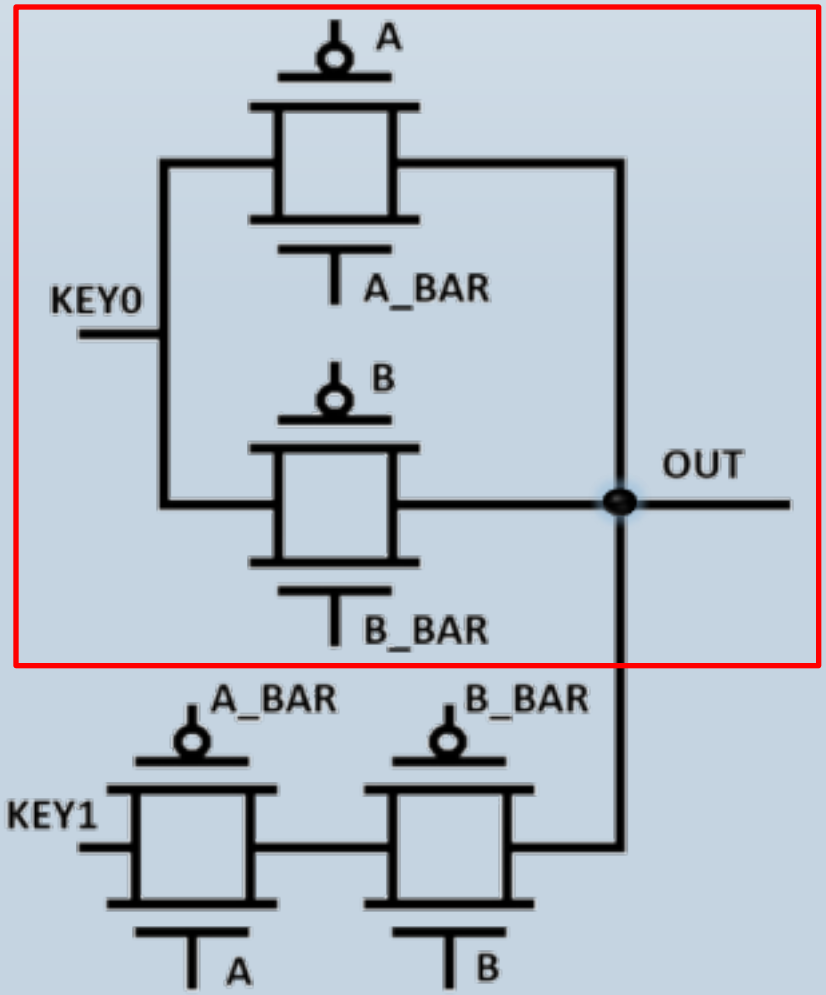
# Method 2-Transmission Gate Topology



**Truth Table**

| KEY0 | KEY1 | A | B | OUT |
|---|---|---|---|---|
| 0 | 0 | X | X | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | X | X | 1 |

**AND/NAND Transmission Gate Topology**




- Key values themselves determine the functionality of the encrypted gate
- Transmission gates pass the key value based on the inputs to the gate

# Transmission Gate Topology

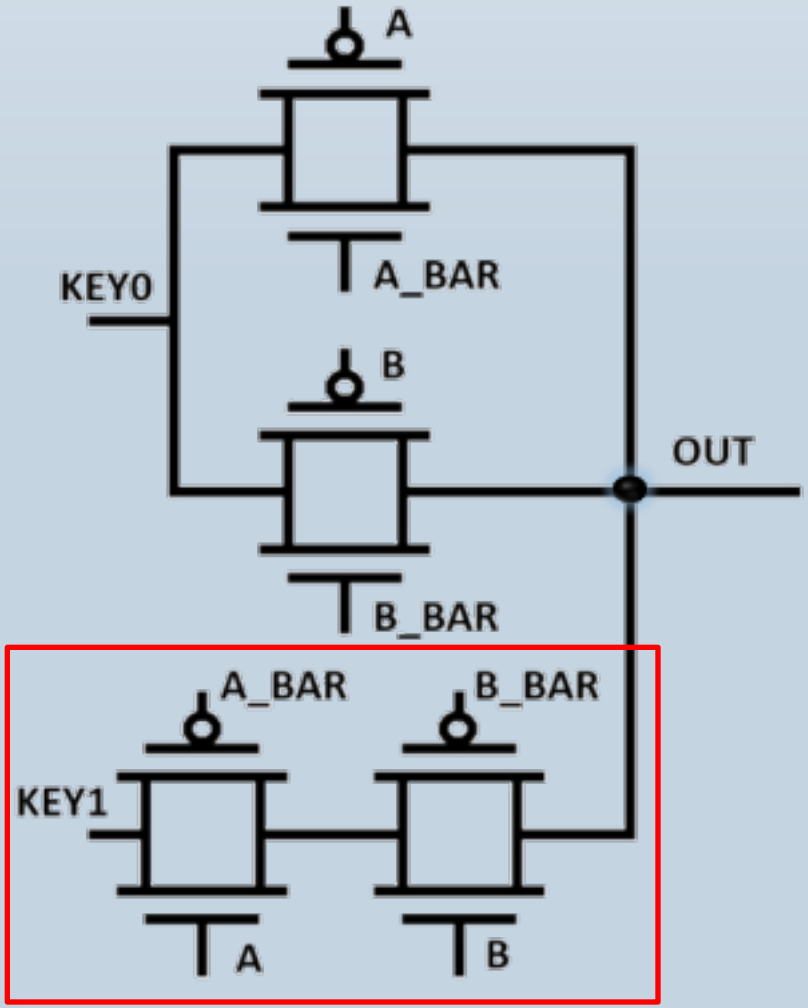


**Truth Table**

| KEY0 | KEY1 | A | B | OUT |
|---|---|---|---|---|
| 0 | 0 | X | X | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | X | X | 1 |

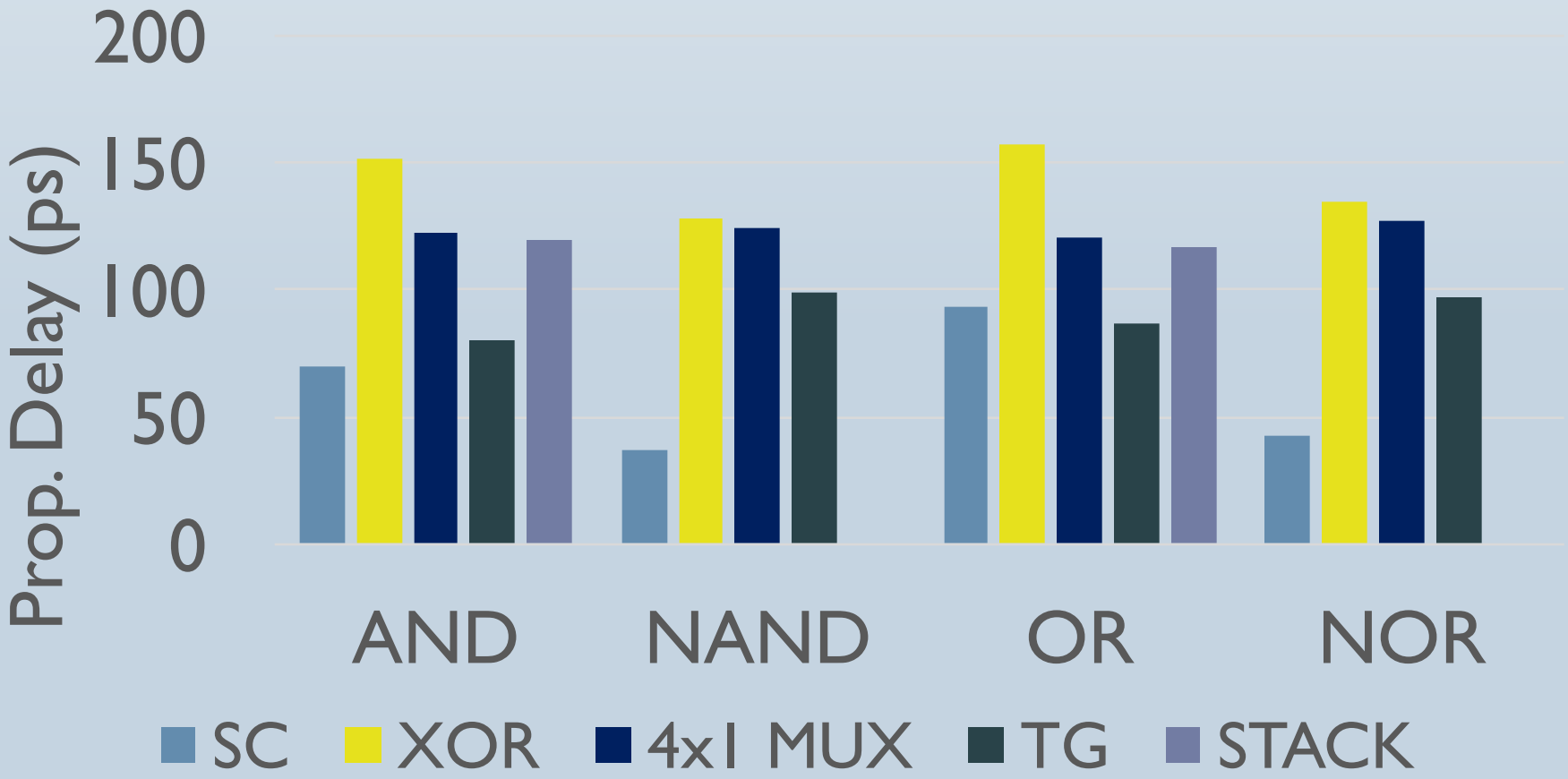
- Unique input combination that produces only a single 0 or 1 output is determined by the bottom path
- Both transmission gates must be on for KEY1 to pass to the output

**AND/NAND Transmission Gate Topology**



# Evaluation of Propagation Delay

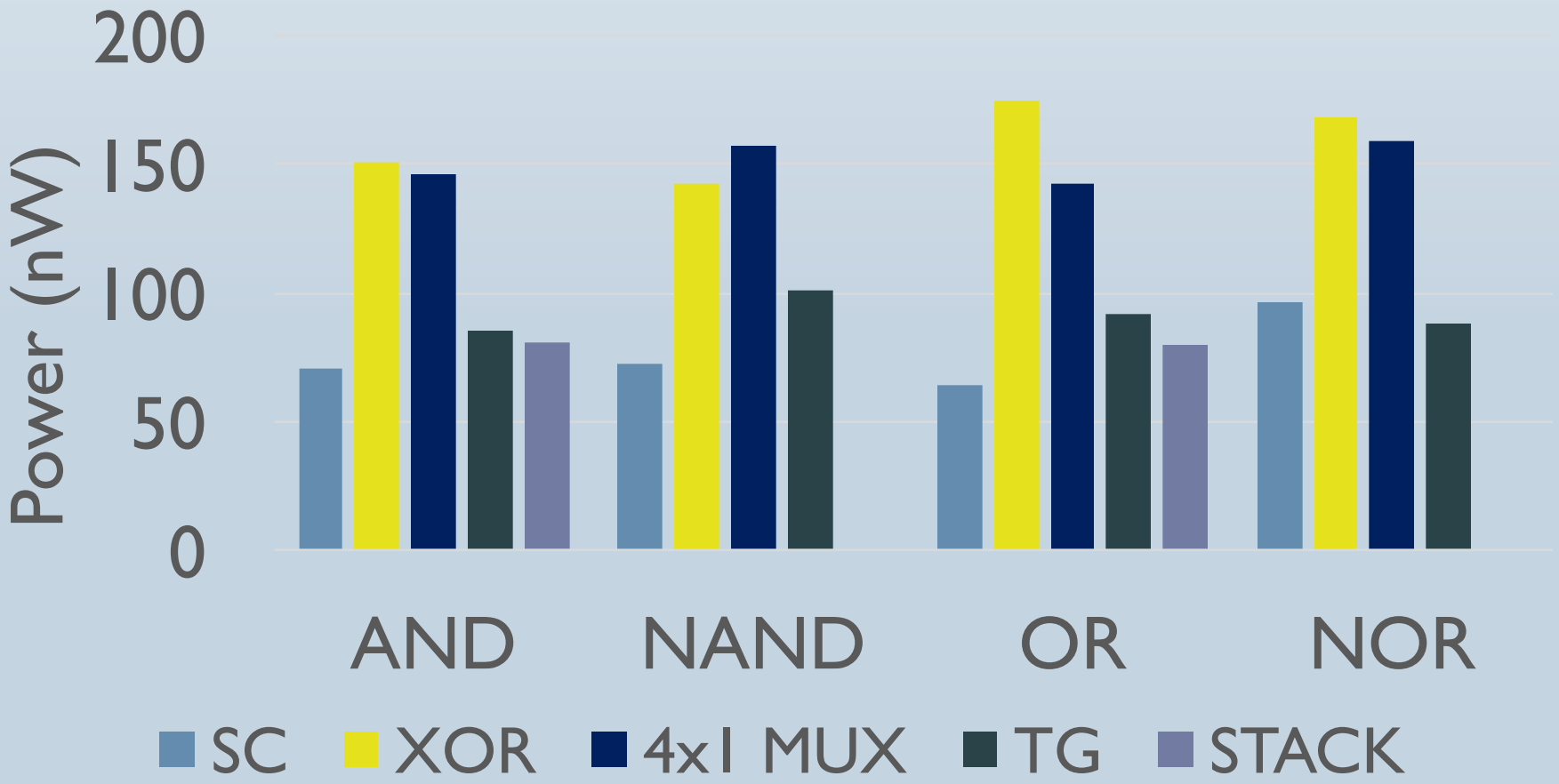





- Stack-based topology provides average increase of 13.1% in performance
- Transmission gate topology reduces performance by 31.1% on average

# Average Power Consumption to Encrypt Gates

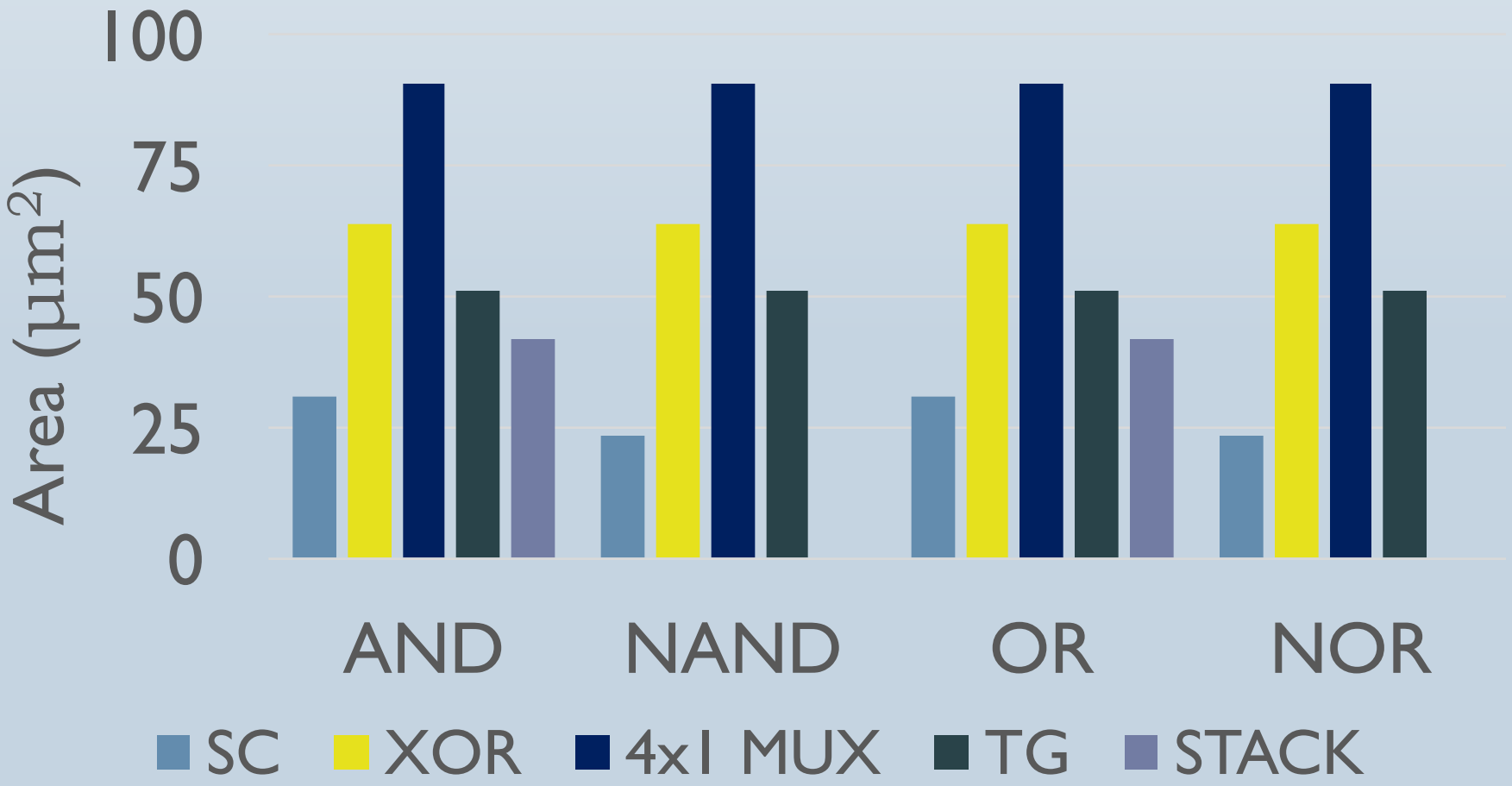





- Stack-based provides average reduction of 47.3% in power consumption
- Transmission gate reduces power consumption by 40.6% on average

# Area Occupied by Encrypted Gates






- Stack-based topology provides average reduction of 44.4% in used area
- Transmission gate topology reduces used area by 31.7% on average

# Presentation Outline

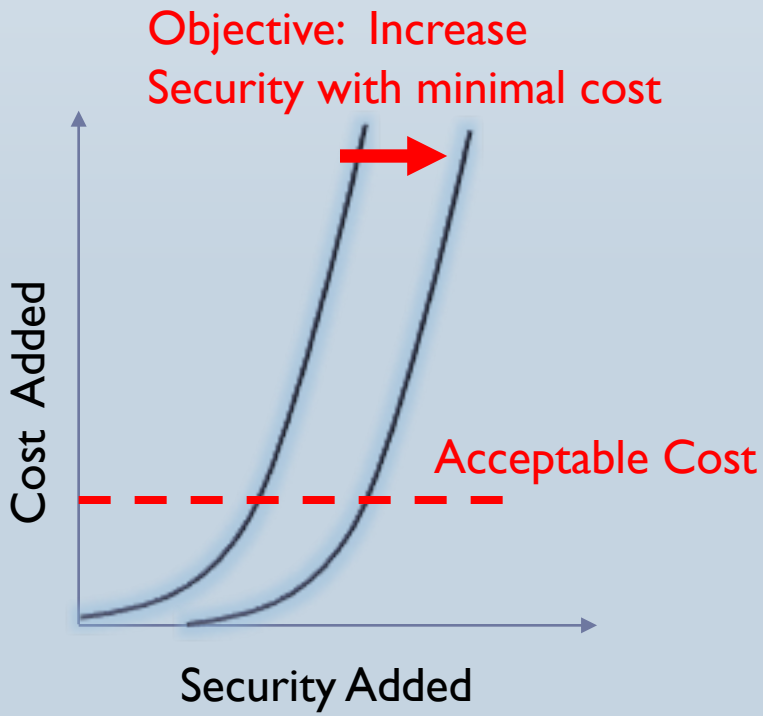


- Introduction to Hardware Security
- Logic Encryption
- Reduced Overhead Logic Obfuscation
- ***Design for Trust***
- Conclusions

# Integrating Security as Design Constraint



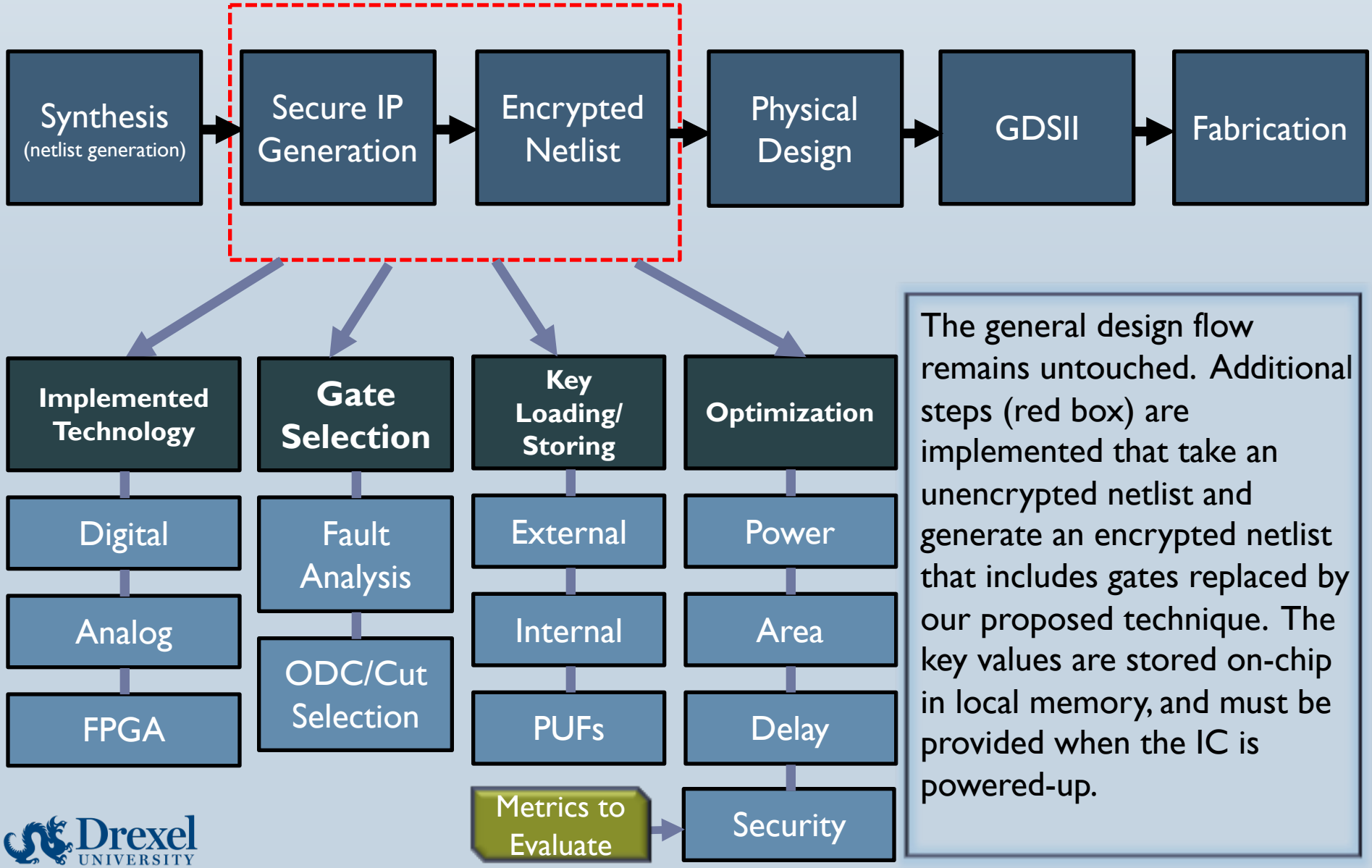
- Implementing hardware security creates tradeoffs with respect to other design constraints
  - Area
  - Power
  - Performance
  - Design time




- Minimize the cost of adding hardware security to IC design process

# Logic Encryption Design Flow






The general design flow remains untouched. Additional steps (red box) are implemented that take an unencrypted netlist and generate an encrypted netlist that includes gates replaced by our proposed technique. The key values are stored on-chip in local memory, and must be provided when the IC is powered-up.

# Presentation Outline



- Introduction to Hardware Security
- Logic Encryption
- Reduced Overhead Logic Obfuscation
- Design for Trust
- ***Conclusions***

# Conclusions



- ICs are increasingly vulnerable to third party tampering and threats that include: IC counterfeiting, intellectual property theft, IC overproduction, and insertion of hardware Trojans
- Logic encryption requires a secure and low-overhead solution for wide spread adaptation
- Gate level logic encryption significantly lowers the per-gate overheads of logic encryption
  - Transmission gate topology: 31.1% improvement in performance improvement, 40.6% reduction in power, 31.7% area reduction
  - Stack-based topology: 13.1% improvement in performance, 47.3% reduction in power consumption, 44.4% area reduction
- Our research is focused on improving gate selection and optimization algorithms to create low cost secure designs

# Drexel ICE Laboratory Research



## 3-D Integrated Circuits

- 3-D/VLSI design methodologies for power and clock network design
- Multi-plane power noise modeling
  - Methodologies to mitigate cross-plane noise coupling
- Power management for multi-domain, multi-plane delivery
- Clock tree synthesis for heterogeneous device planes

## Hardware Security and Trust

- Detection of hardware Trojans
  - Real time detection through side-channel monitoring
- Prevention of attack through design for trust
  - Metrics to quantify gains of implemented security measure
  - Algorithms and methodologies to obfuscate circuit design
- Techniques to protect analog circuits
- Securing FPGA functionality from theft

## Heterogeneous Systems Integration

## Low Power Circuit Design

-Near-threshold circuits (NTC) for low-power applications
  - Implement circuit families including CMOS and current mode logic in NTC

-Modify circuit techniques and methodologies for NTC
  - Gate and circuit modifications to assure timing closure while meeting area constraints

# Thank you