

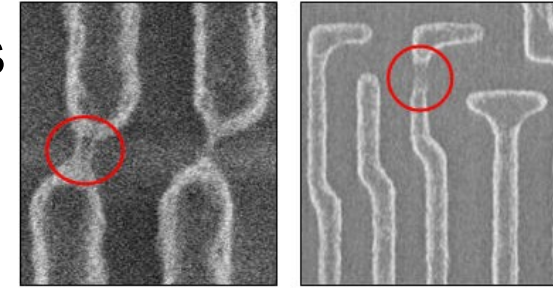
# Hardware Metering

**Mark Tehranipoor**

**Introduction to Hardware Security & Trust**  
**University of Florida**

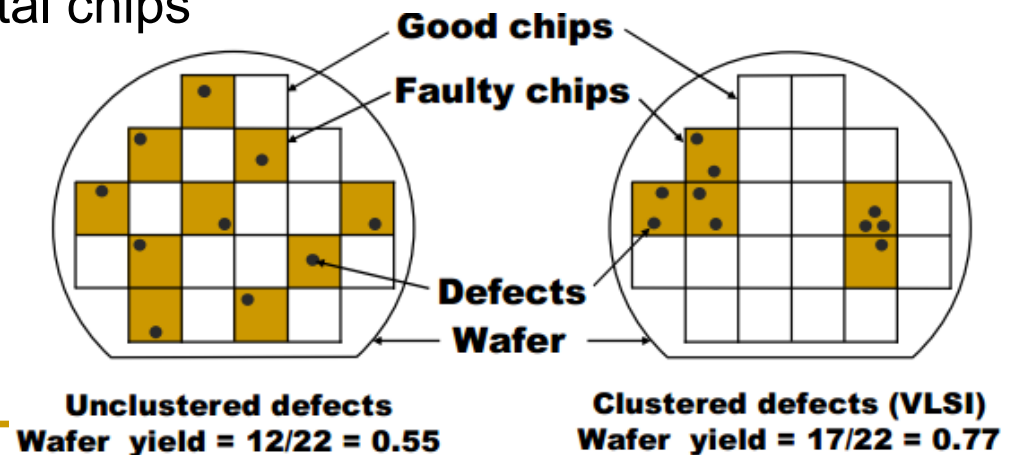
# Background: Test and Yield

- Errors in fabrication process cause defects on chip which causes chip to malfunction.
- Chips are tested in order to detect defects
- Failing chips are discarded
- Fraction(percentage) of remaining good chips is called the yield.



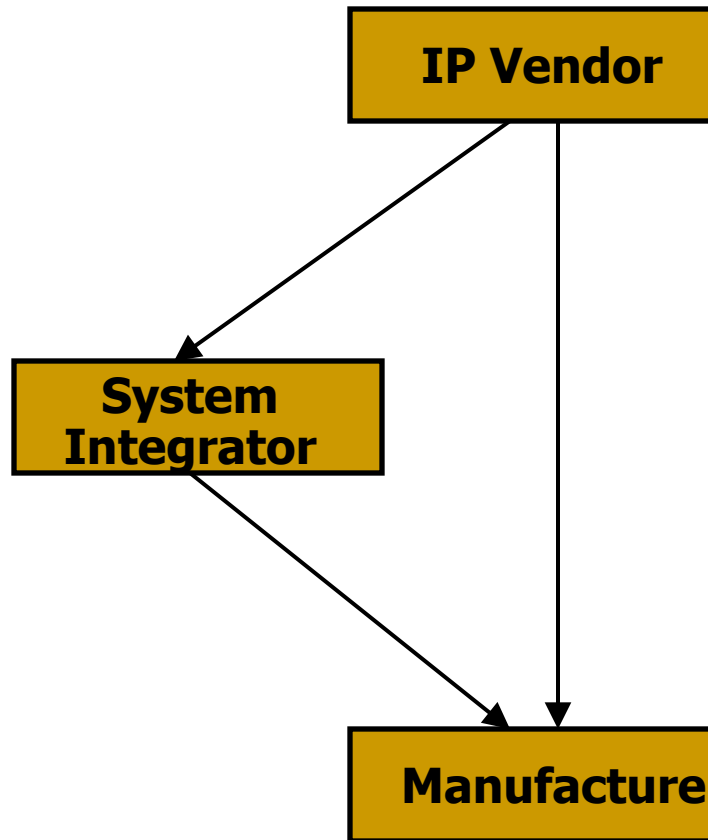
$$\text{Yield} = \frac{\text{total chips} - \text{discarded chips}}{\text{total chips}}$$

- Foundry decides/predicts yield



# HW Threats

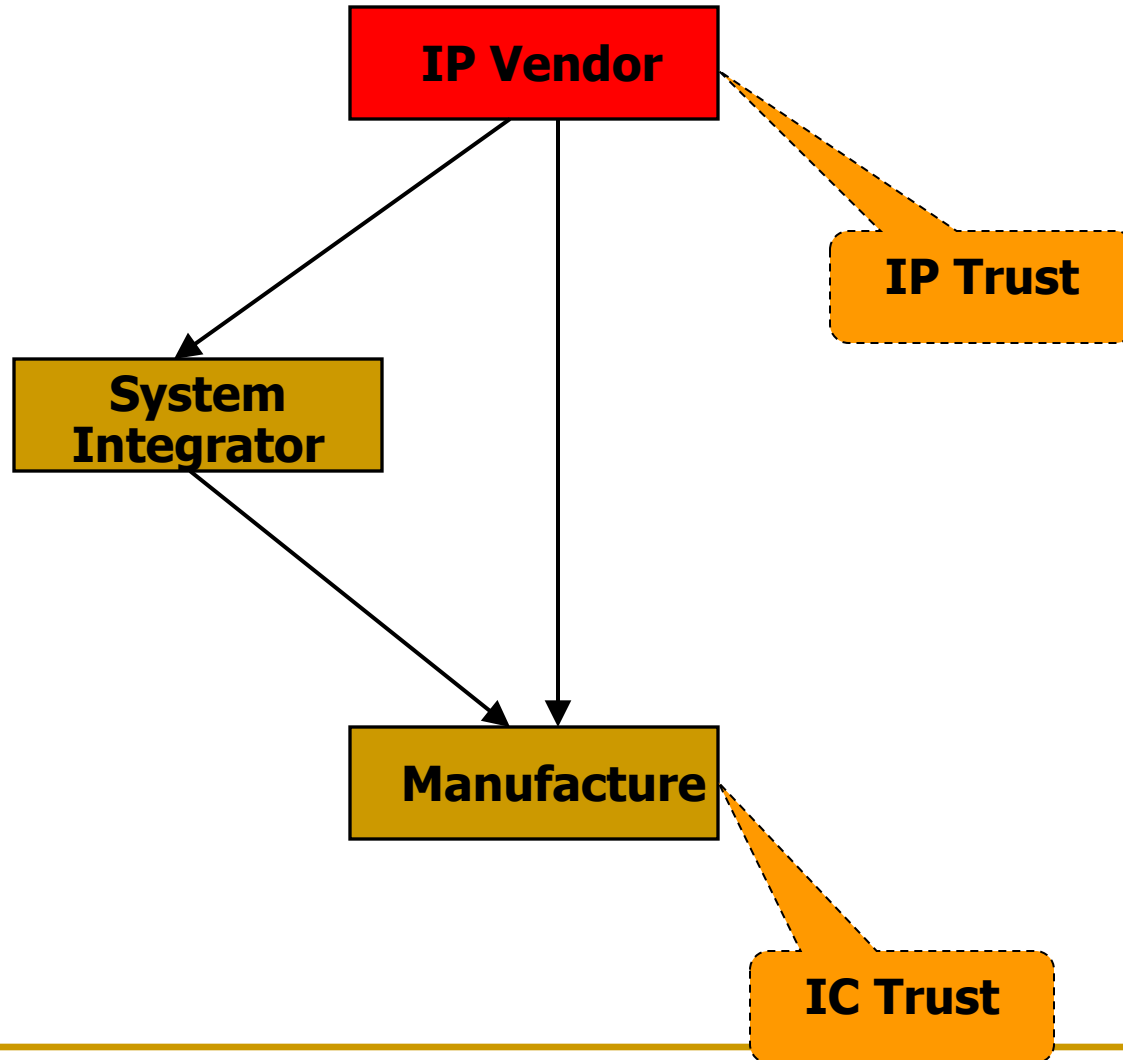
---



**Any of these steps can be untrusted**

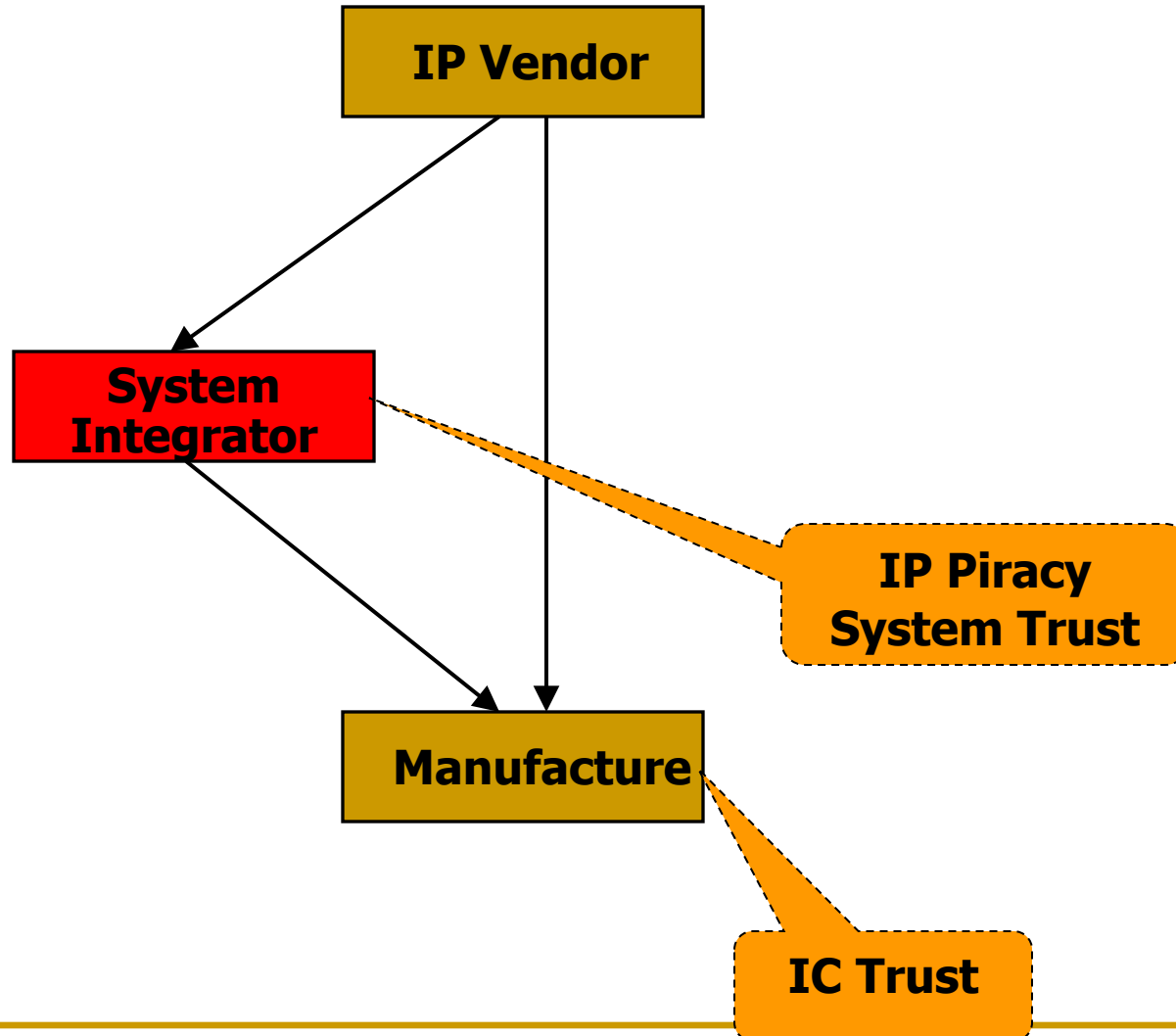
---

# HW Threats



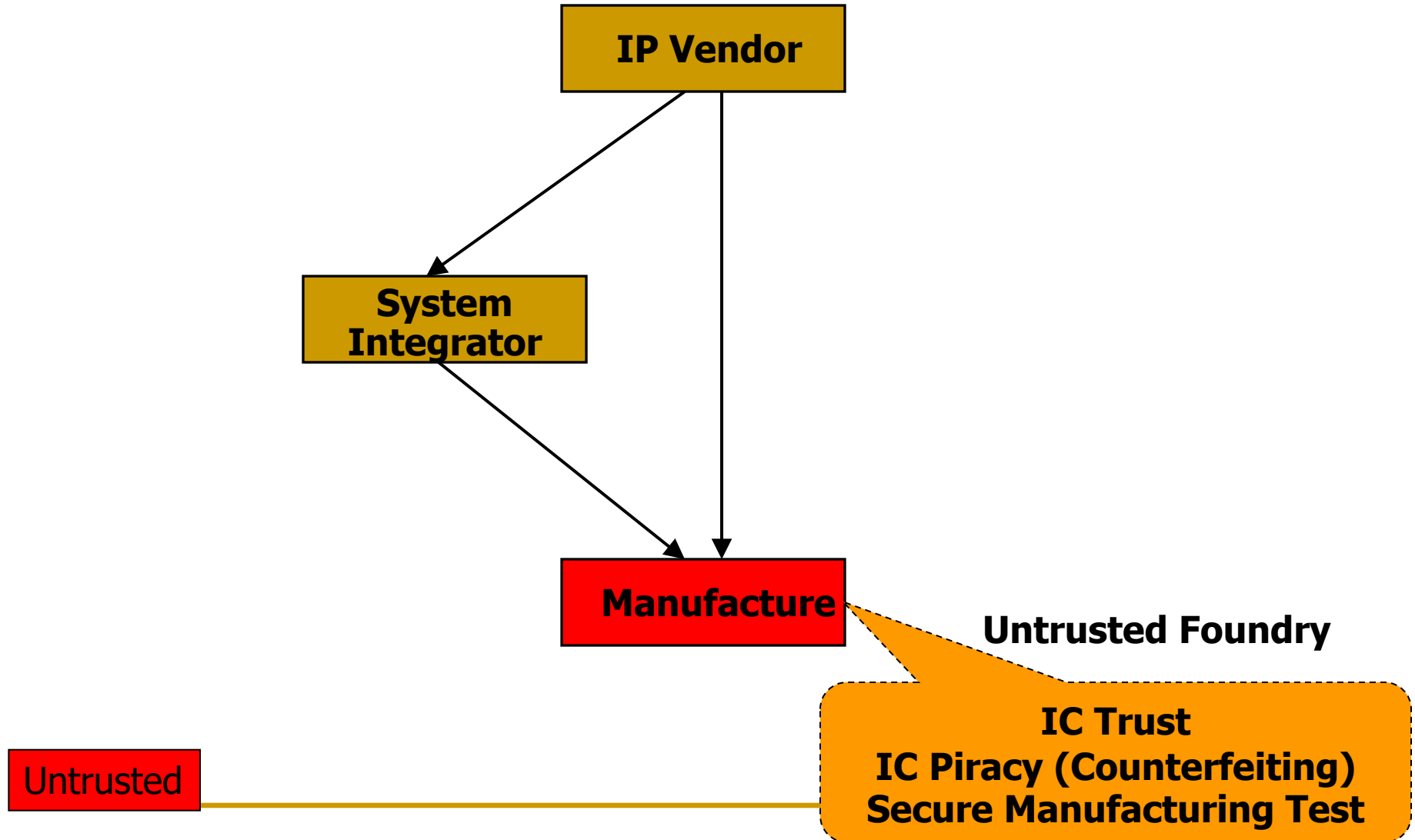
Untrusted

# HW Threats

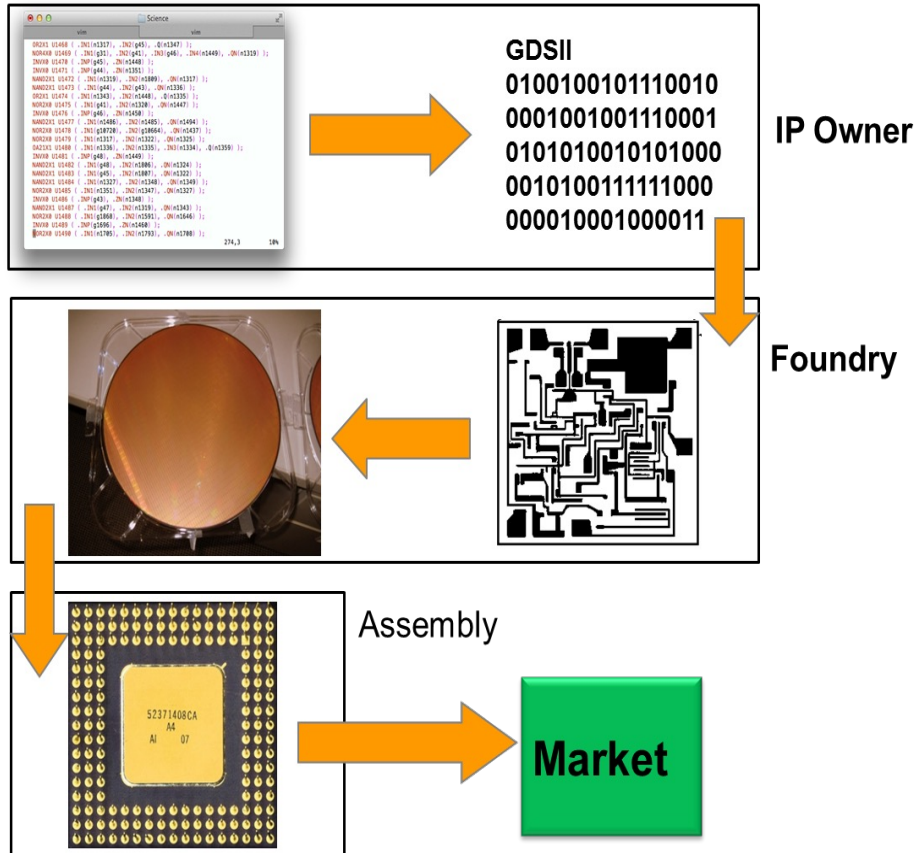


Untrusted

# HW Threats

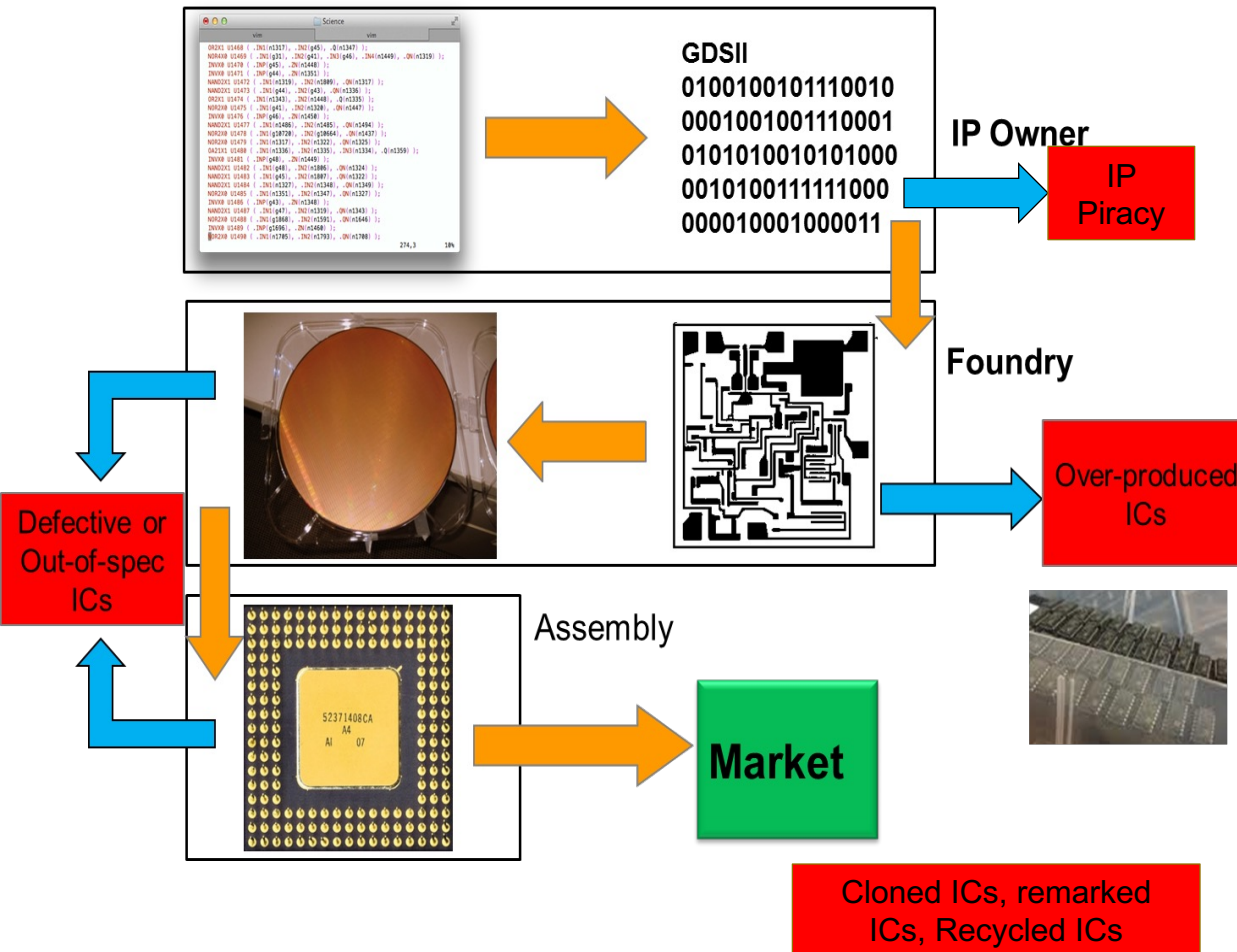


# Chip Production Flow



- Little communication between IP Owner and Foundry.
- Foundry is trusted with full design.
- Responsible for production of requested amount of chips.
- IP holder provides foundry/assembly with all **test patterns** and **responses**.

# Chip Production Flow



- Foundry looks for its own profit.
- Once mask is produced, producing IC's is simple and cheap.
- Lack of communication makes it difficult for owner to track produced chips.



# Stolen IP is a real thing...

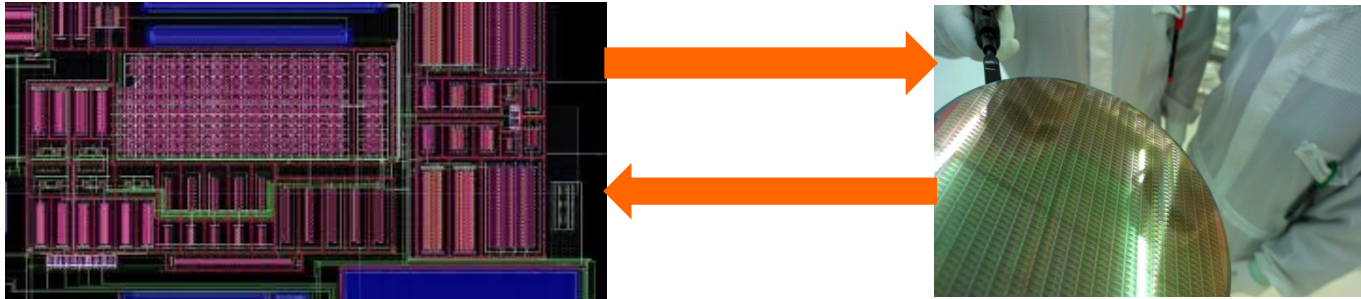
---

[wiki]

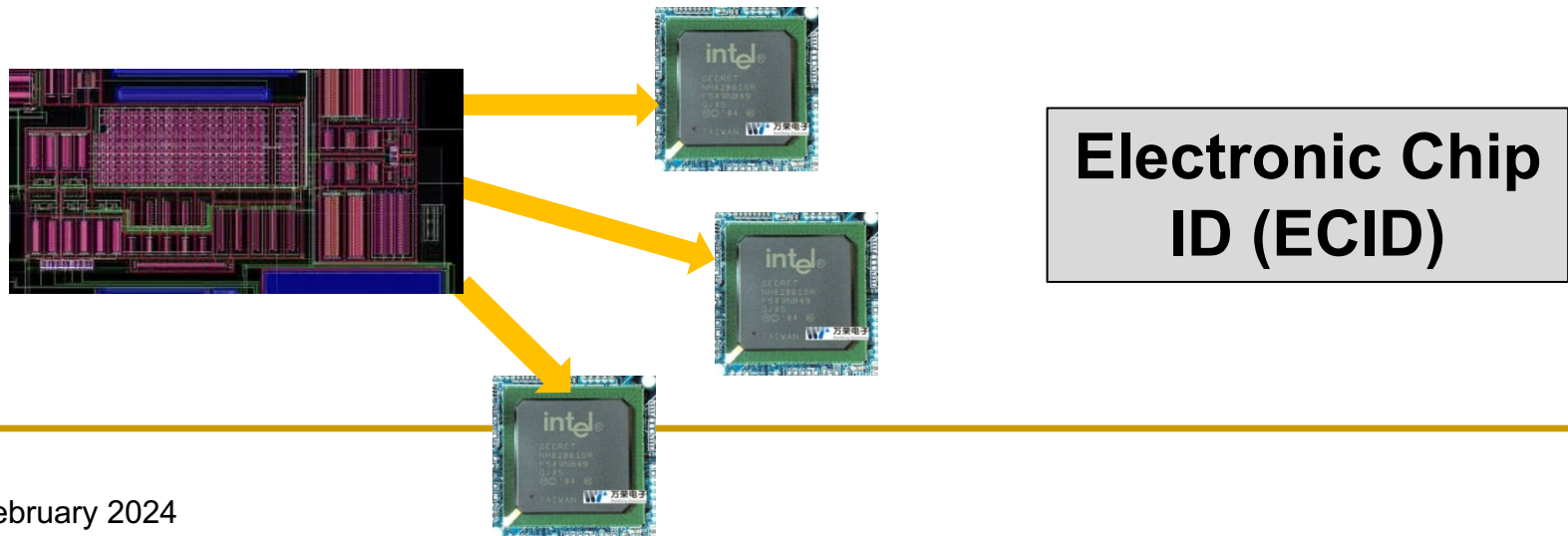
---

# Need for Hardware Metering

- Need for better communication between IP Owner and foundry/assembly.



- Need for IP Owner to be able to track produced chips.

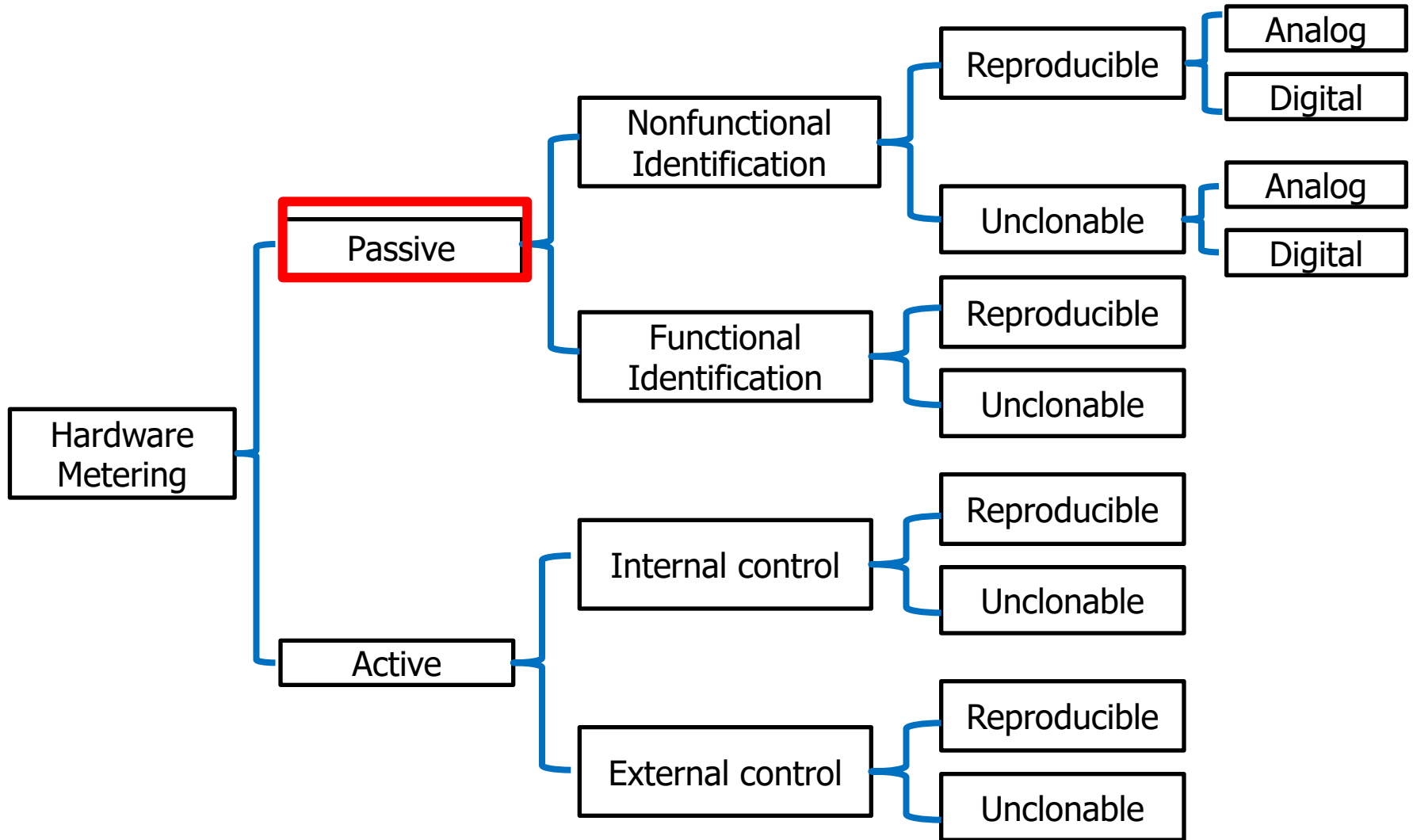


# Hardware Metering

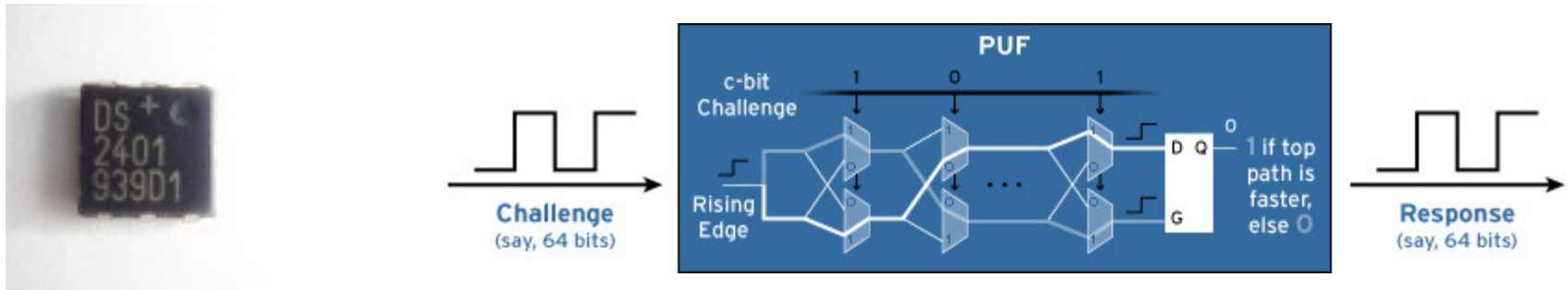
---

- **Hardware metering (IC metering):**
    - ❑ Set of security protocols that enable IP owners to achieve post-fabrication control over their ICs
    - ❑ Methods attempt to **uniquely tag each chip** to facilitate tracing them
    - ❑ Two main methods:
      - **Active metering**
      - **Passive metering**
  - **Could be applicable to PCBs, e.g., IoTs**
-

# Taxonomy of Metering Methods

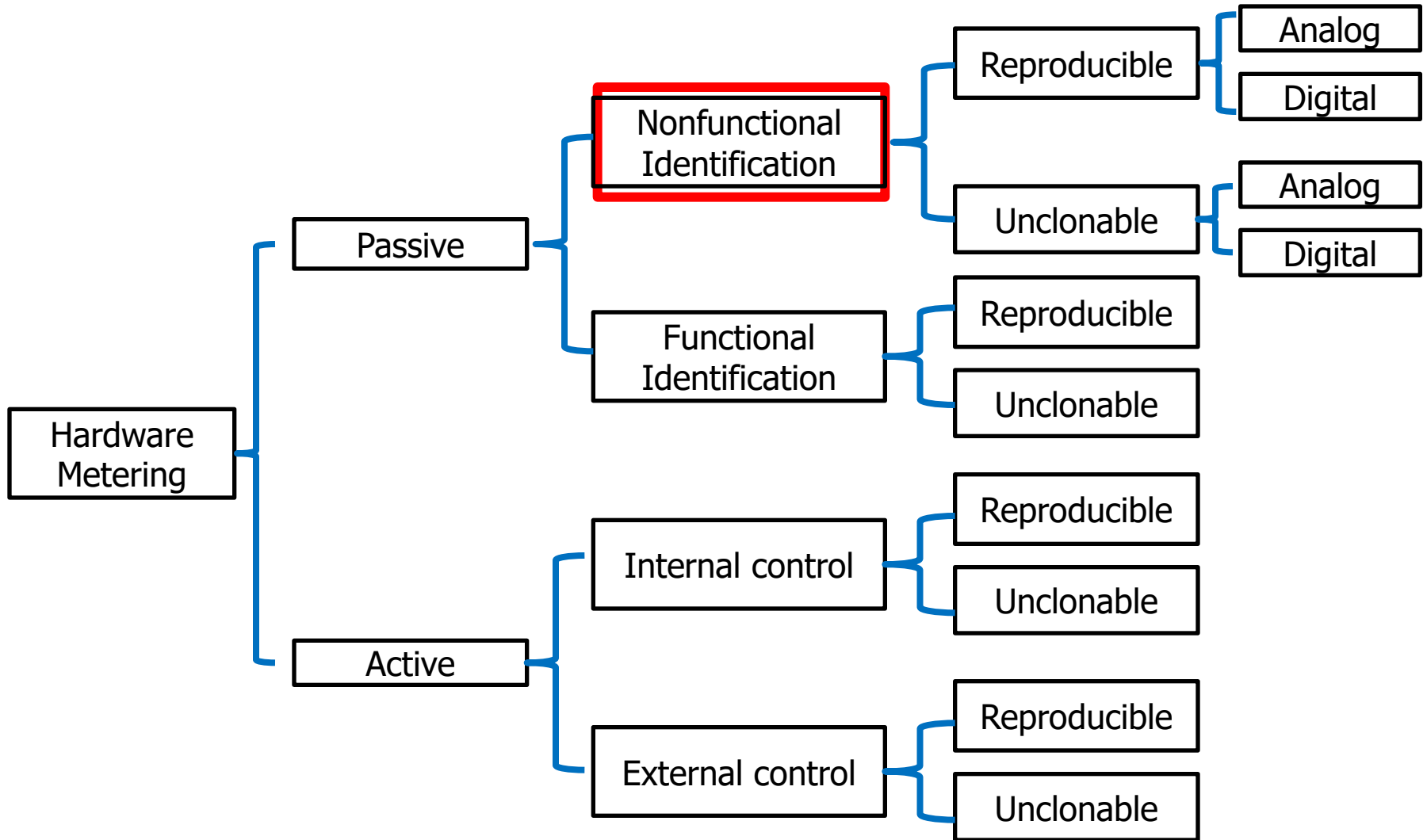


# Passive Metering



- ICs can be **passively monitored**.
- Can be achieved by physically identifying:
  - ❑ Serial numbers on chips
  - ❑ Storing unique identifiers in memory. These are called **Nonfunctional Identification**
    - E.g., Electronic Chip ID (ECID)
- Tagging an IC's functionality: **Functional Identification**

# Taxonomy of Metering Methods



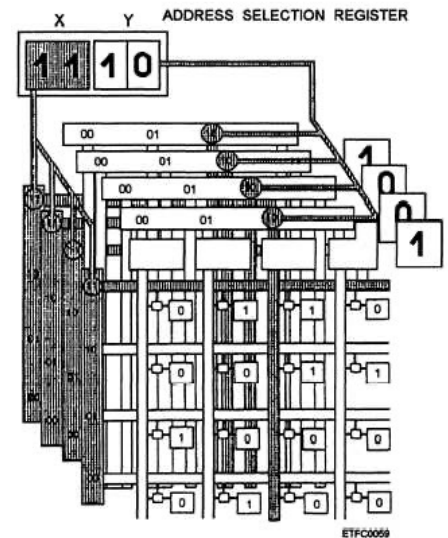
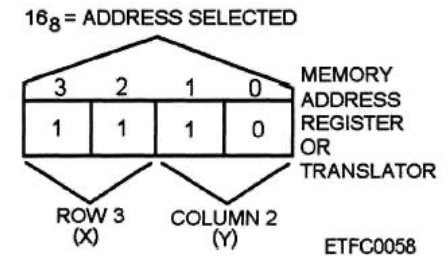
# Nonfunctional Identification

---

- Unique ID is separate from the chip's functionality.
- Vulnerable to cloning and/or removal.
  - ❑ Once chip is tagged, foundry can copy same tag on other chips or simply remove tag so chip cannot be traced.
- Possible to overproduce.
  - ❑ Foundry can produce multiple chips with same tag.
  - ❑ Out of millions of chips, probability of finding two matching tags is small.
- Two main types:
  - ❑ Reproducible
  - ❑ Unclonable

# Nonfunctional Identification: Reproducible Identifiers

- Unique ID's are stored on the chip package, on die, or in a memory on-chip.
- Examples:
  - ❑ Indented serial numbers
  - ❑ Digitally stored serial numbers
- Advantages:
  - ❑ Do not depend on randomness
  - ❑ Easy to track / identify.
- Disadvantages:
  - ❑ Easy to clone/modify
  - ❑ Easy to overproduce





# Nonfunctional Identification: Unclonable Identifiers

---

- Uses random process variations in silicon to generate random unique numbers called **fingerprints**.
- If additional logic is needed to generate these value, the method is said to be **extrinsic**.
- If no additional logic is needed, the method is called **intrinsic**.
- Advantages:
  - Values cannot be reproduced due to randomness in process variations
- Disadvantages:
  - Foundry could overproduce ICs without knowledge of IP owner
    - i.e., these methods do not prevent counterfeiting. The over-produced chip can be detected if IP owner gets his/her hands on those chips by comparing the identifier on the chip with his/her database

# Unclonable Identifiers

---

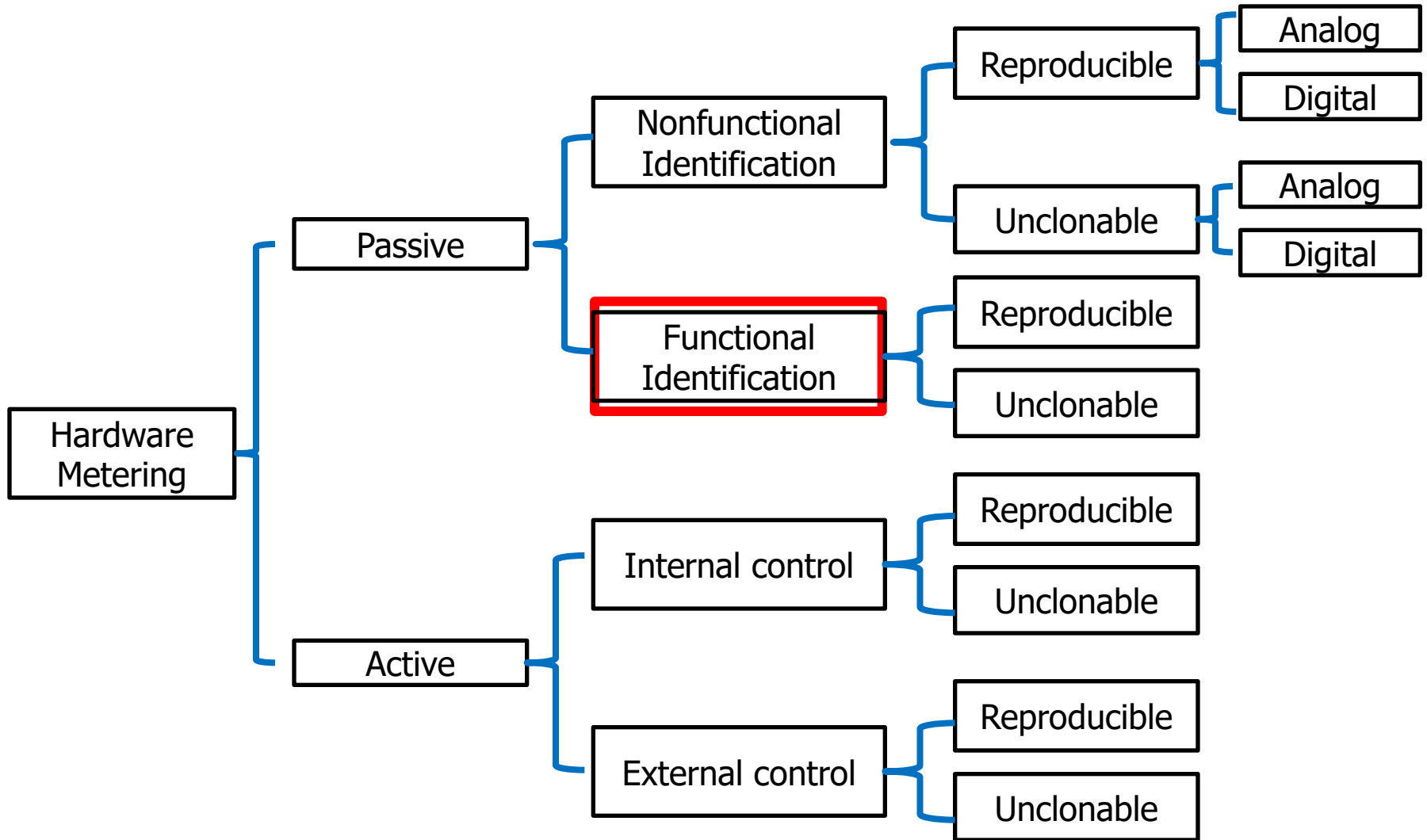
## ■ Extrinsic methods:

- ❑ Require additional logic such as PUF (Physical Unclonable Function) or ICID
- ❑ ICID
  - Threshold mismatches in array of transistors incurred different currents and therefore unique random numbers.
- ❑ PUFs
  - Series of ring oscillators (ROs) generate random value due to process variations.

## ■ Intrinsic methods:

- ❑ Unique identification if external test vectors can be applied.
- ❑ Uses IC **leakage**, **power**, **timing**, and **path signatures** (unique due to process variations).
- ❑ Does not need additional logic and can be readily used on existing designs

# Taxonomy of Metering Methods



# Functional Metering

---

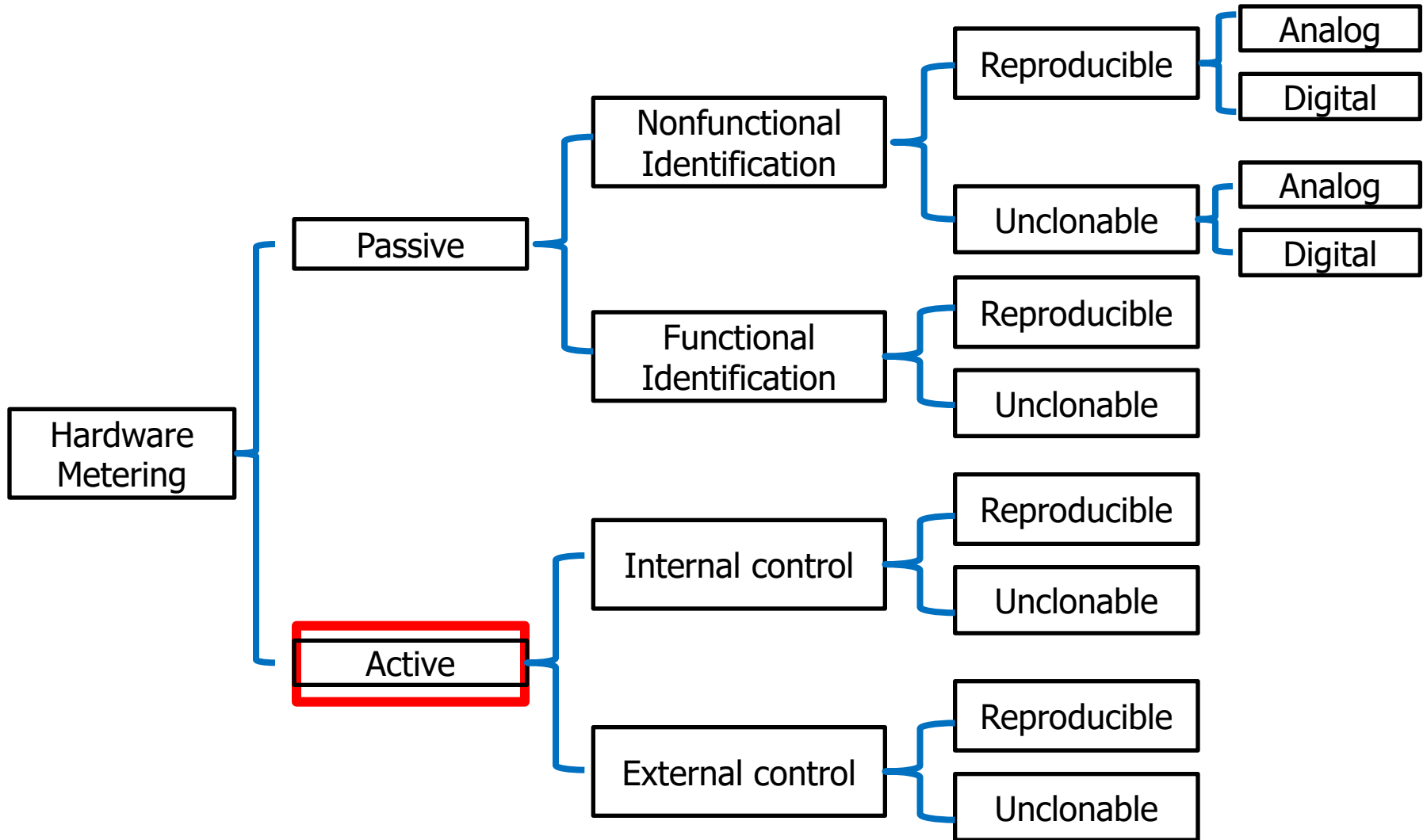
- Identifiers linked to chip's internal functional details during synthesis.
- Each chip's function gets a unique signature.
  - E.g., additional states added to generate same output
- Function unchanged from input to output
- Internal transactions unique to each chip
- Challenge in fabricating ICs with different paths from same mask.

# Functional Metering

---

- One method is fabricating chips from same mask and maintaining one programmable path.
  - ❑ E.g., Datapath could be programmed post-silicon.
  - ❑ IP Owner provides correct input/key combination to foundry to program chip post-silicon.
- Additional work proposes adding redundant states.
  - ❑ Programmable read logic enables selecting correct permutation for a control sequence.
- Drawbacks:
  - ❑ Testing such circuitry provides low coverage because the actual functionality of the chip is hidden during the test process by foundry and assembly
  - ❑ It requires the chip to go back to a trusted facility to be activated.

# Taxonomy of Metering Methods

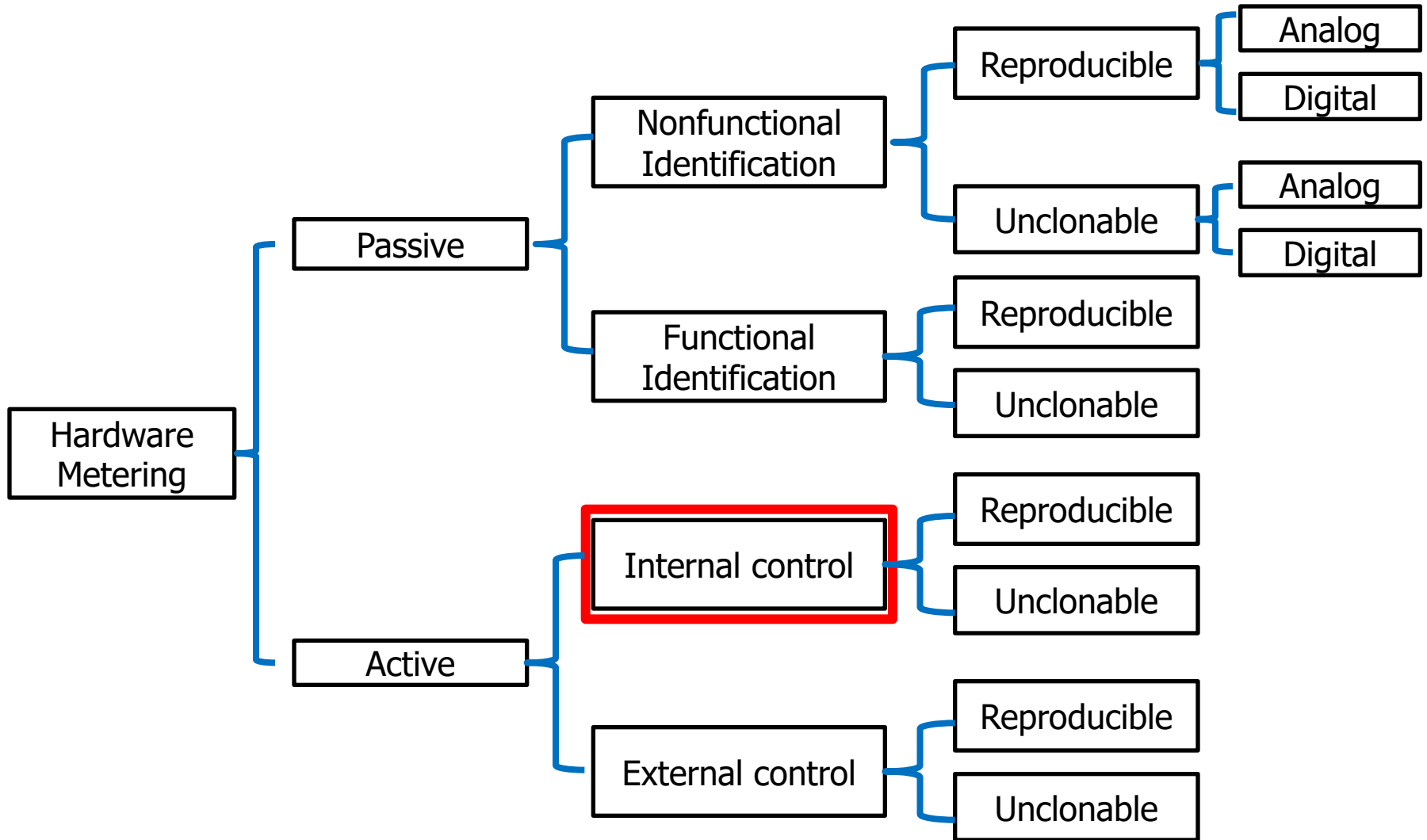


# Active Metering

---

- Provides active way for designer to enable, control, or disable IC.
- Unlike passive metering, active metering requires ***communication between design house (IP owner) and foundry.***
- Two types:
  - Internal
  - External

# Taxonomy of Metering Methods





# Internal (Integrated) Active Metering

---

- Hides states and transition in the design that can only be accessed by designer.
- Locks are embedded within structure of computation model in hardware design in form of FSM.
- Adding additional states or duplicating certain states in FSM adds ability for designer to decide which datapath (sequence of states) to use post-silicon.
  - Since states are added, specific combinations are needed to bring FSM to correct output. Only IP owner knows such combination.

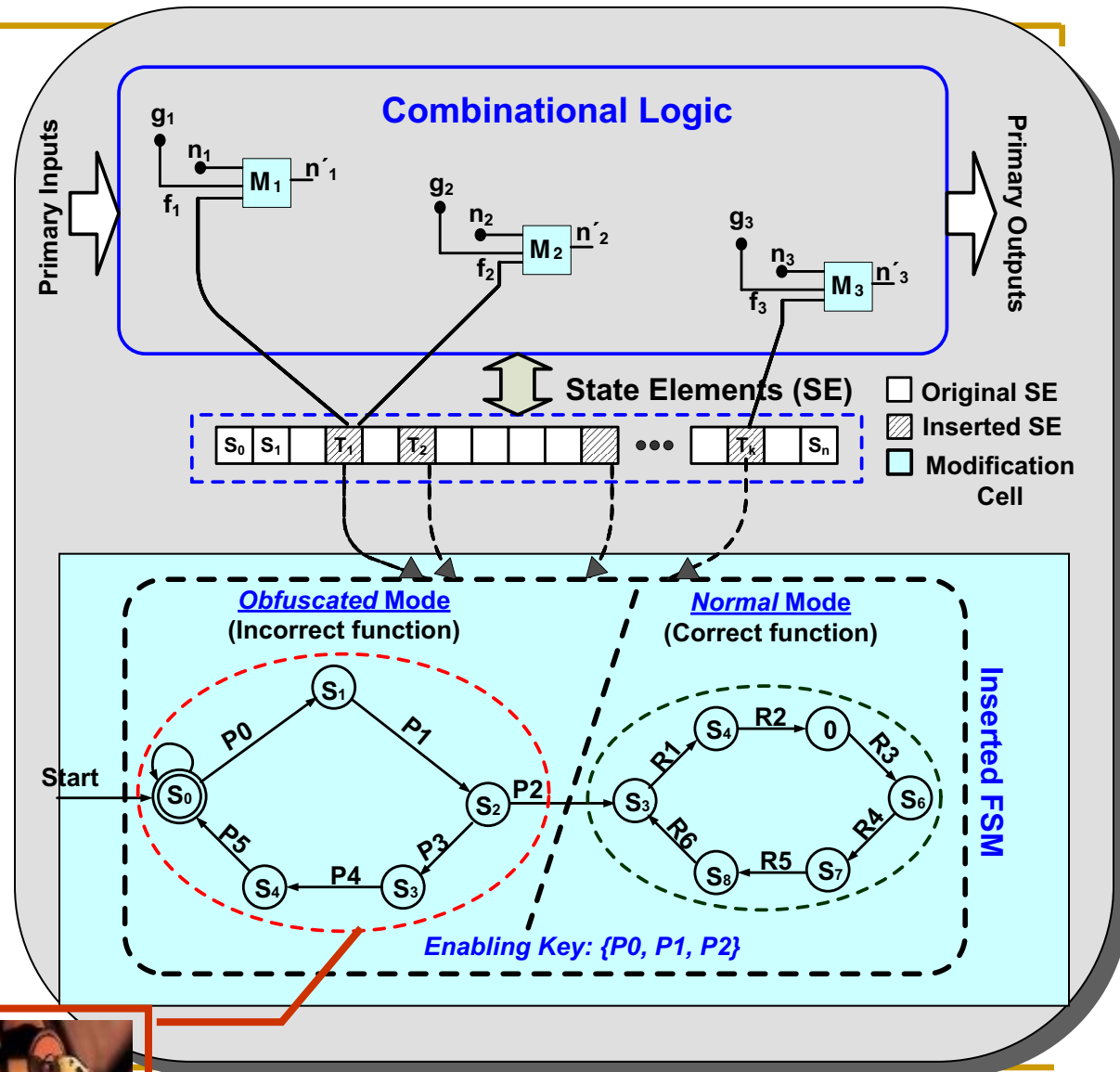
# State Space Obfuscation

## Basic Idea:

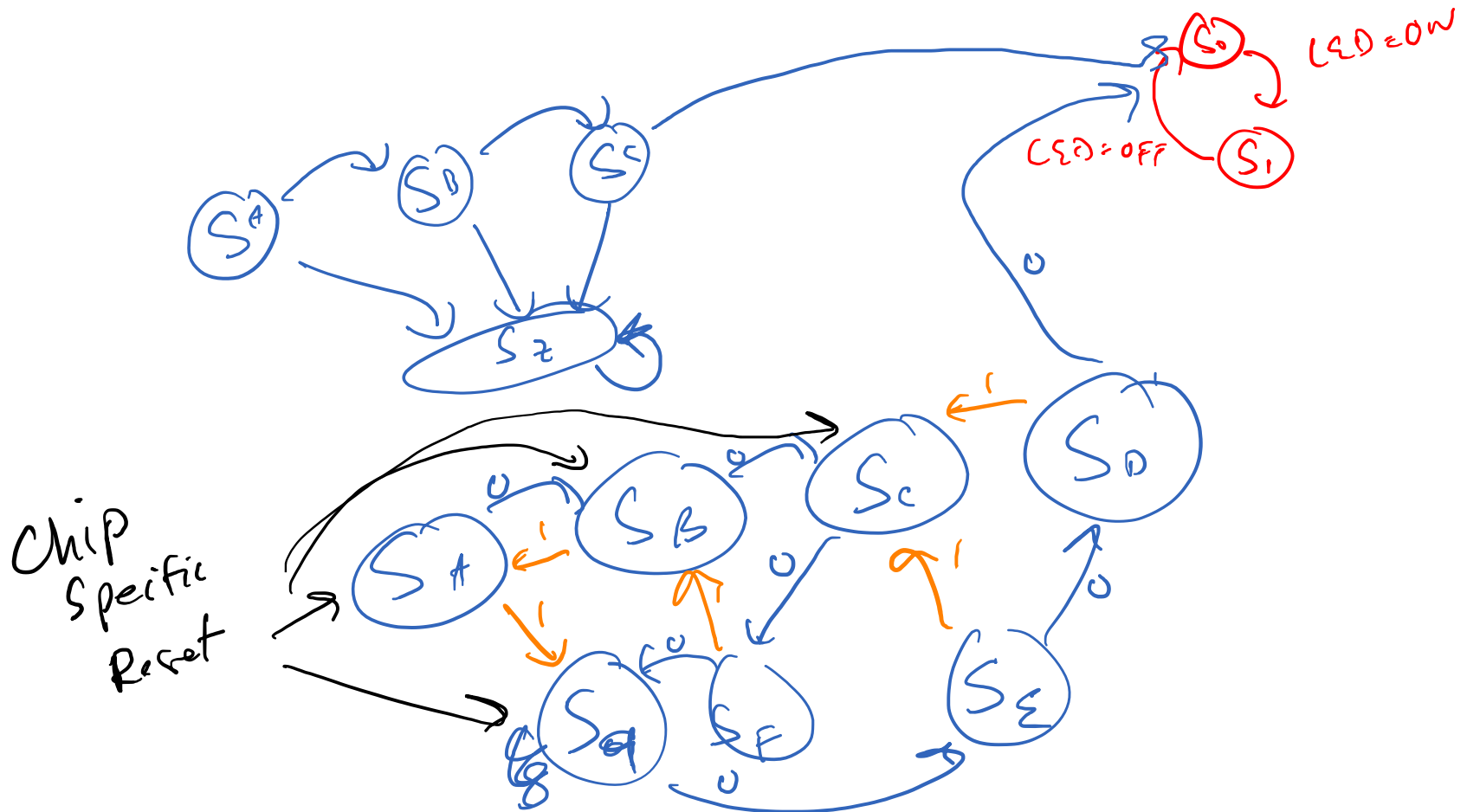
- A locking approach where normal behavior is enabled only upon appn. of a key
- Provable robustness

## Key Innovations:

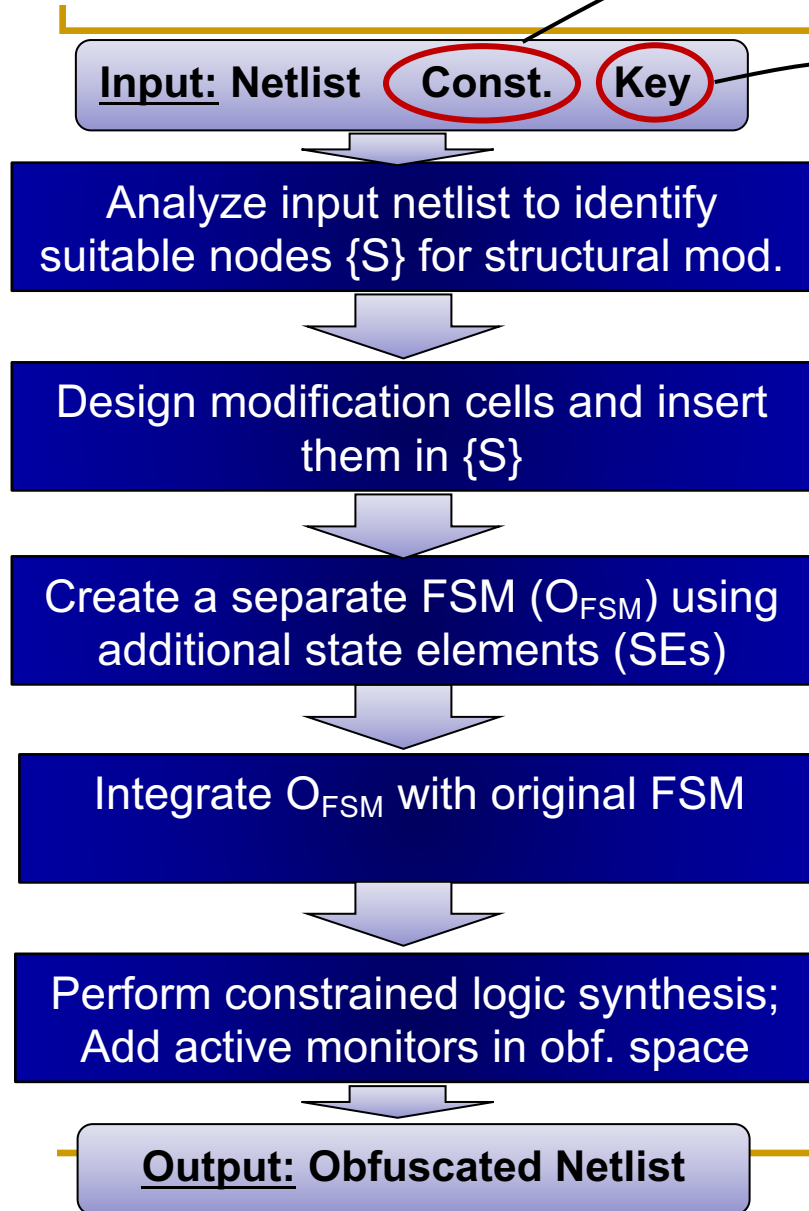
- It obfuscates the state space AND the comb. logic
- Uses rich theory of automata to transform the state space & associated logic



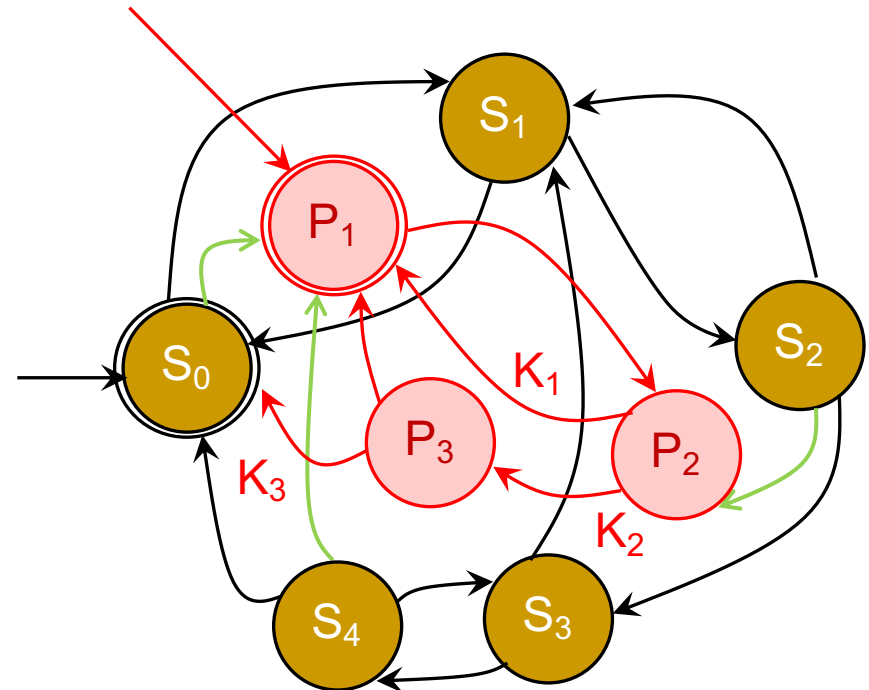
**Protects against Piracy, RE & Tampering**



# The Flow



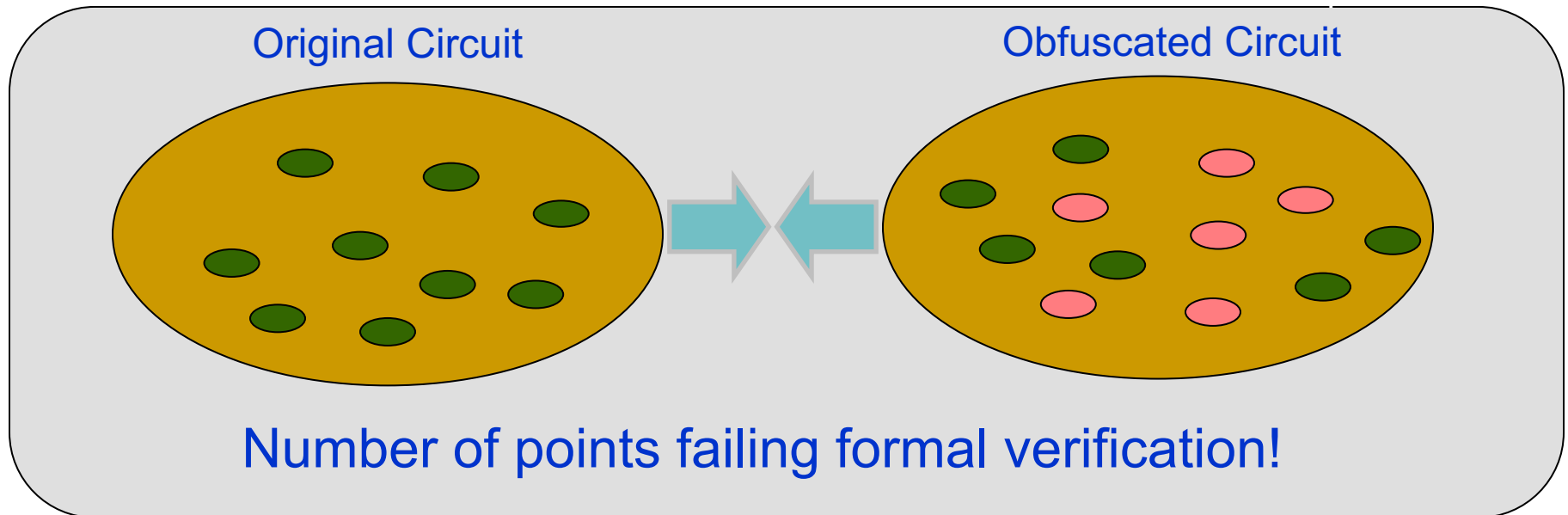
- Transforms underlying state machine



- Affects the dynamic behavior of the machine

# Challenges

1. How to measure level of obfuscation?
2. How to measure the corresponding security benefit?

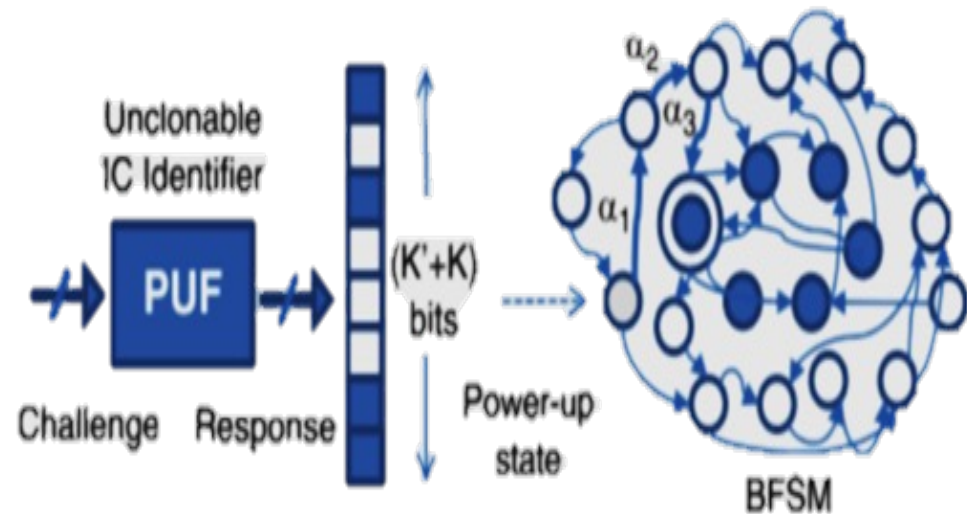


Improvement in Trojan coverage (w.r.t. defense against Trojan attacks)!



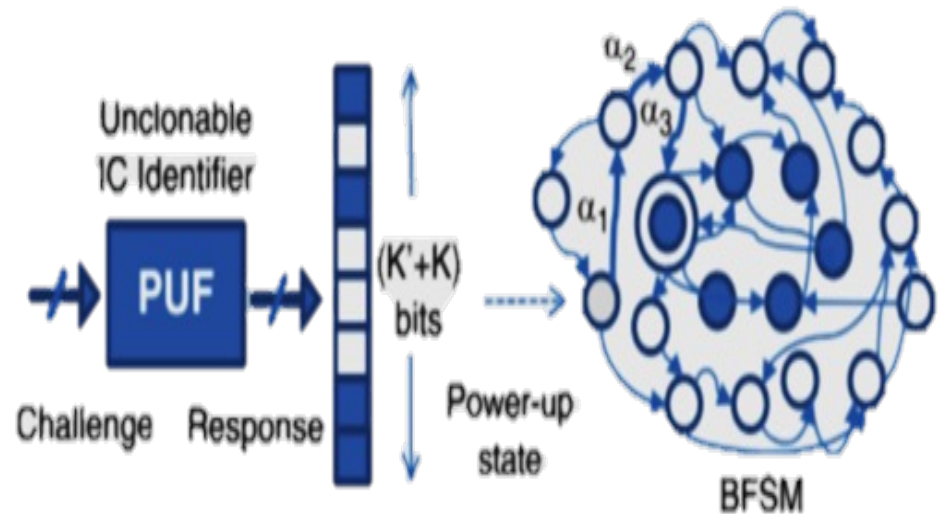
# Internal (Integrated) Active Metering

- States and transitions for controlling chips are integrated within functional specifications
- $K = \log_2(S)$  flip flops needed to implement  $S$  states
- Adding  $S_1$  states requires  $K_1 = \log_2(S_1 + S)$  flip flops
- Few additional flip flops can exponentially increase the number of states.



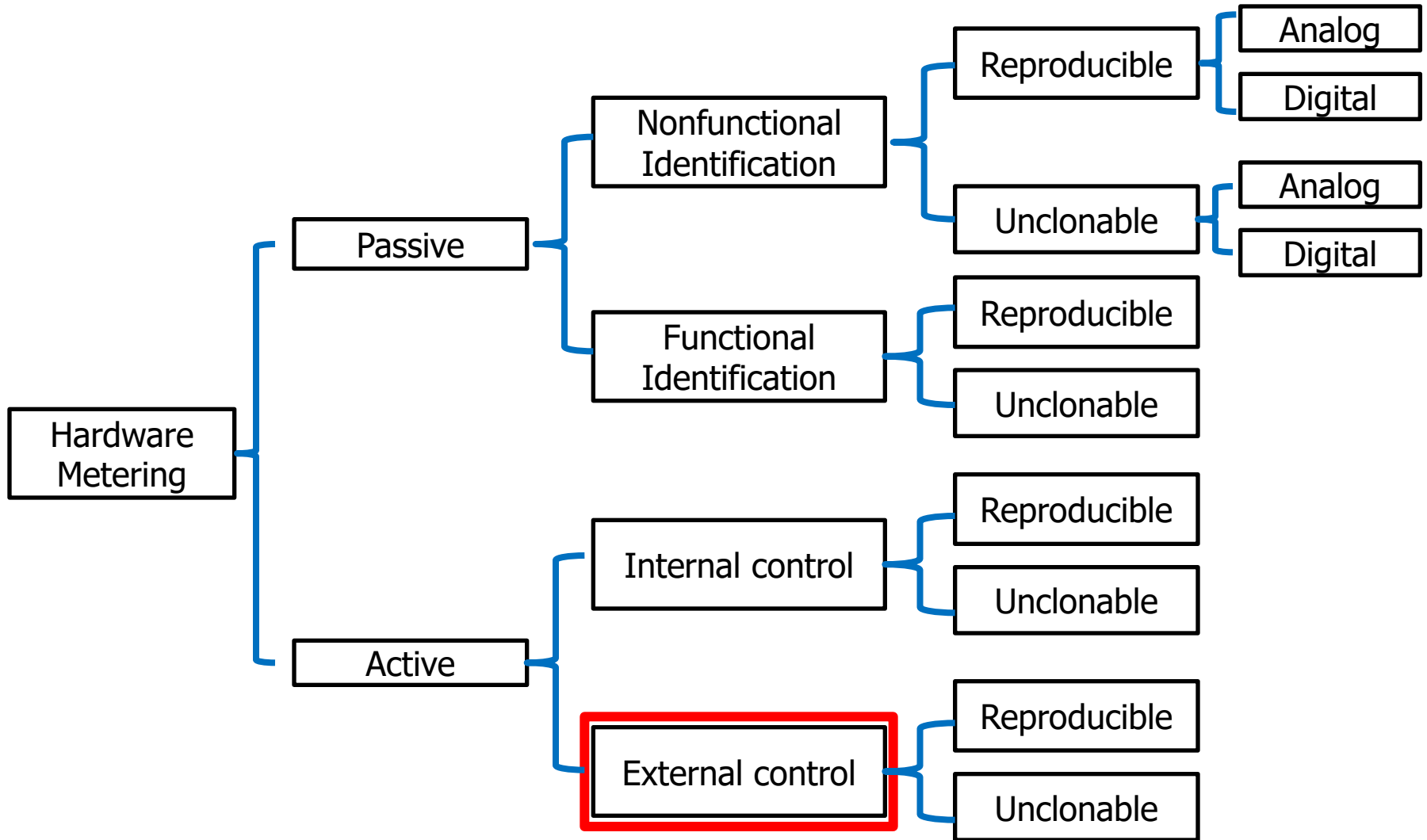
# Internal (Integrated) Active Metering

- PUF generates random values, it sends device to random FSM state.
- Only IP owner with knowledge of FSM can find correct sequence to set FSM to reset state.
- Storing a sequence on chip requires additional logic such as clocks and memory and also requires chip to wait until entire sequence has been shifted in.





# Taxonomy of Metering Methods



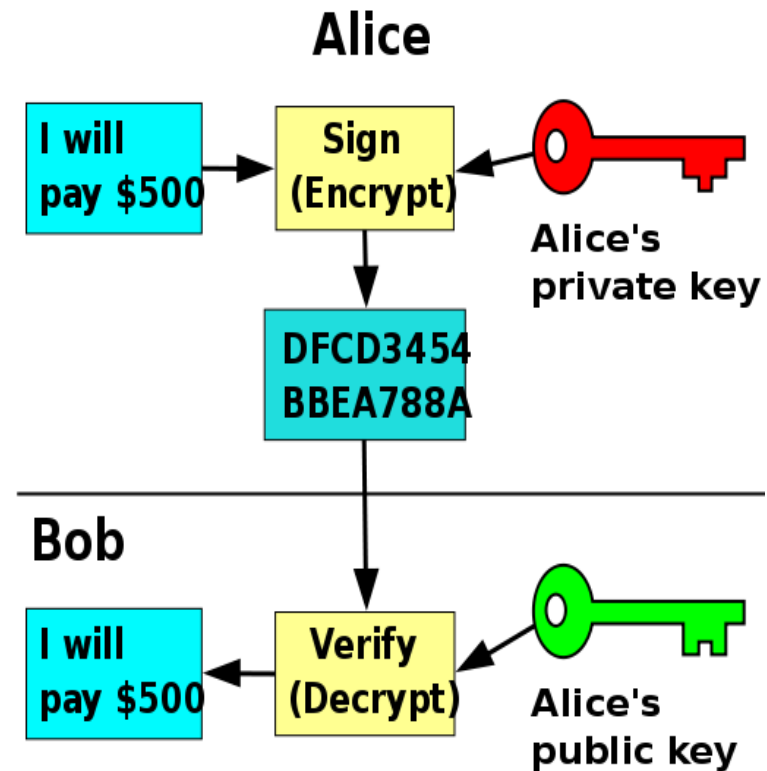
# External Active Metering

---

- Uses external asymmetric cryptographic techniques to lock IC.
- Cryptographic circuits rely on public and private keys to give IP owner control over activation/correct function of the circuit.
- Only IP owner knows private key to unlock IC's functionality or testability.

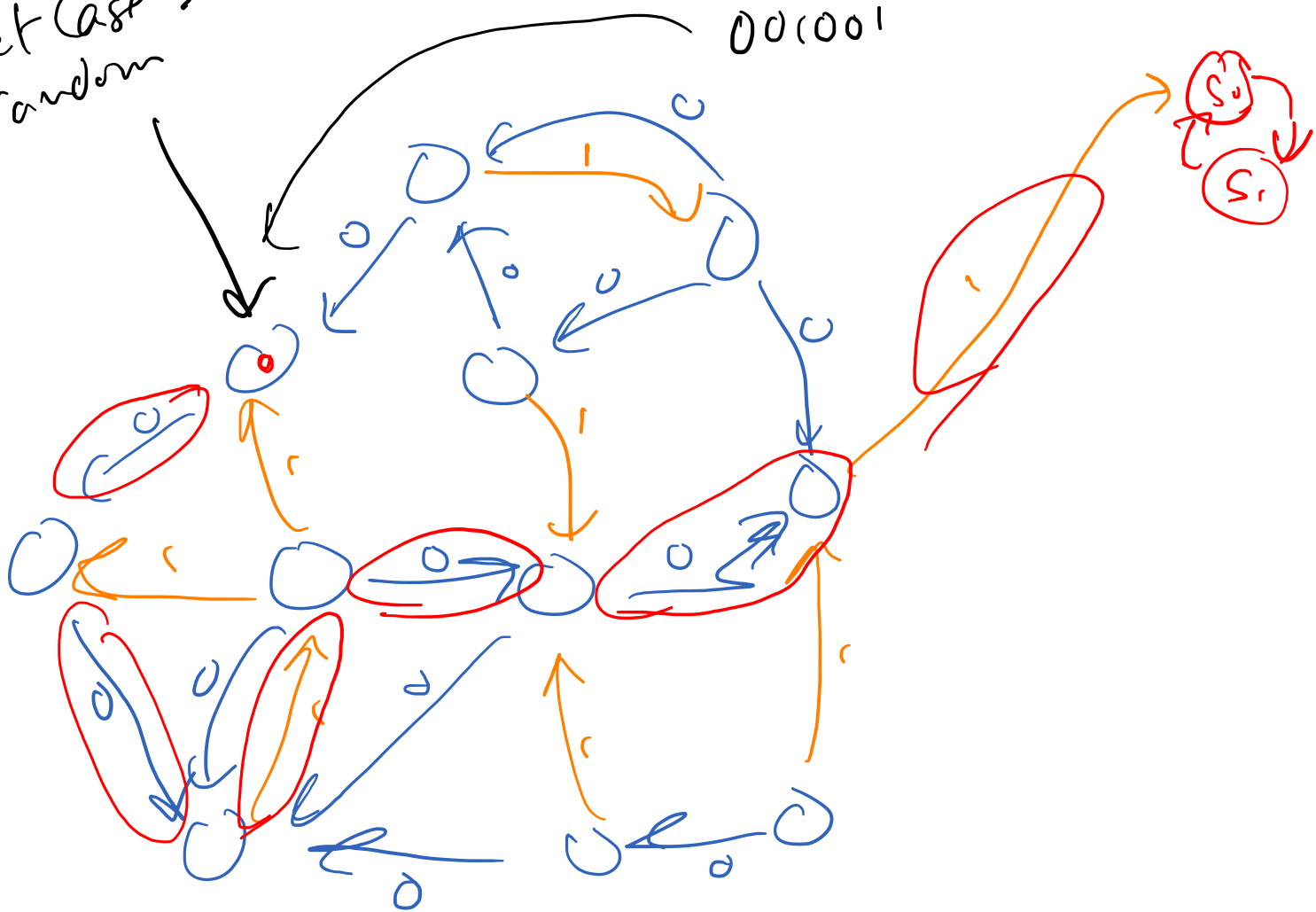
# Background: Public Key Cryptography

- Uses two large prime numbers  $p$  and  $q$  to generate co-prime  $n=pq$
- Private ( $d$ ) and public ( $e$ ) keys based on  $n$ ,  $p$ , and  $q$  are calculated
  - ( $e,n$ ) are shared, message is encrypted using ( $d,n$ )
  - Decryption can be done using ( $e,n$ )
- Security relies on magnitude of prime numbers  $p$  and  $q$



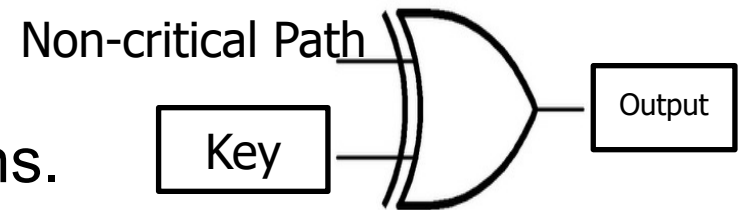
PUF  $\rightarrow$  001001  $\rightarrow$  RSA  $\rightarrow$  01010101  $\rightarrow$  RSA<sub>dec</sub>

Reset Case : PUF  
random

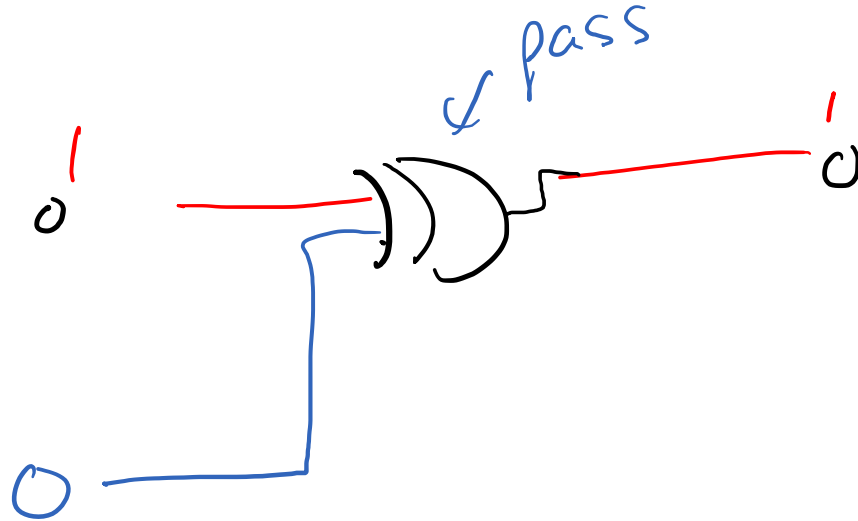


# EPIC: Ending Piracy of Integrated Circuits

- This technique tries to allow IP Owner to have control over number of chips activated.
- Uses public-key encryption to lock correct functionality of chip.
- At the gate level, XOR gates are placed on selected non-critical paths.
  - Only IP owner can generate key

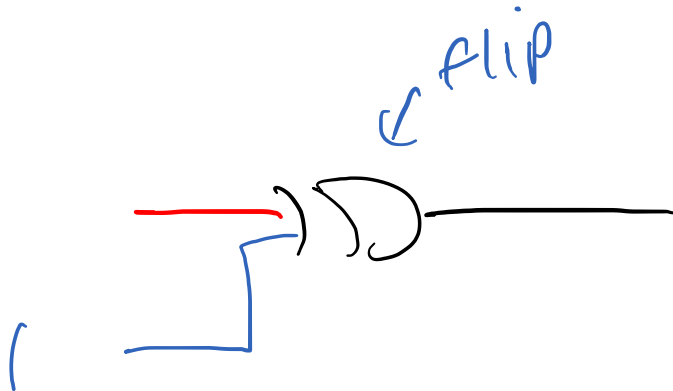


*Roy et al., DATE 2008*



$$1 \wedge 0 = 1$$

$$0 \wedge 0 = 0$$

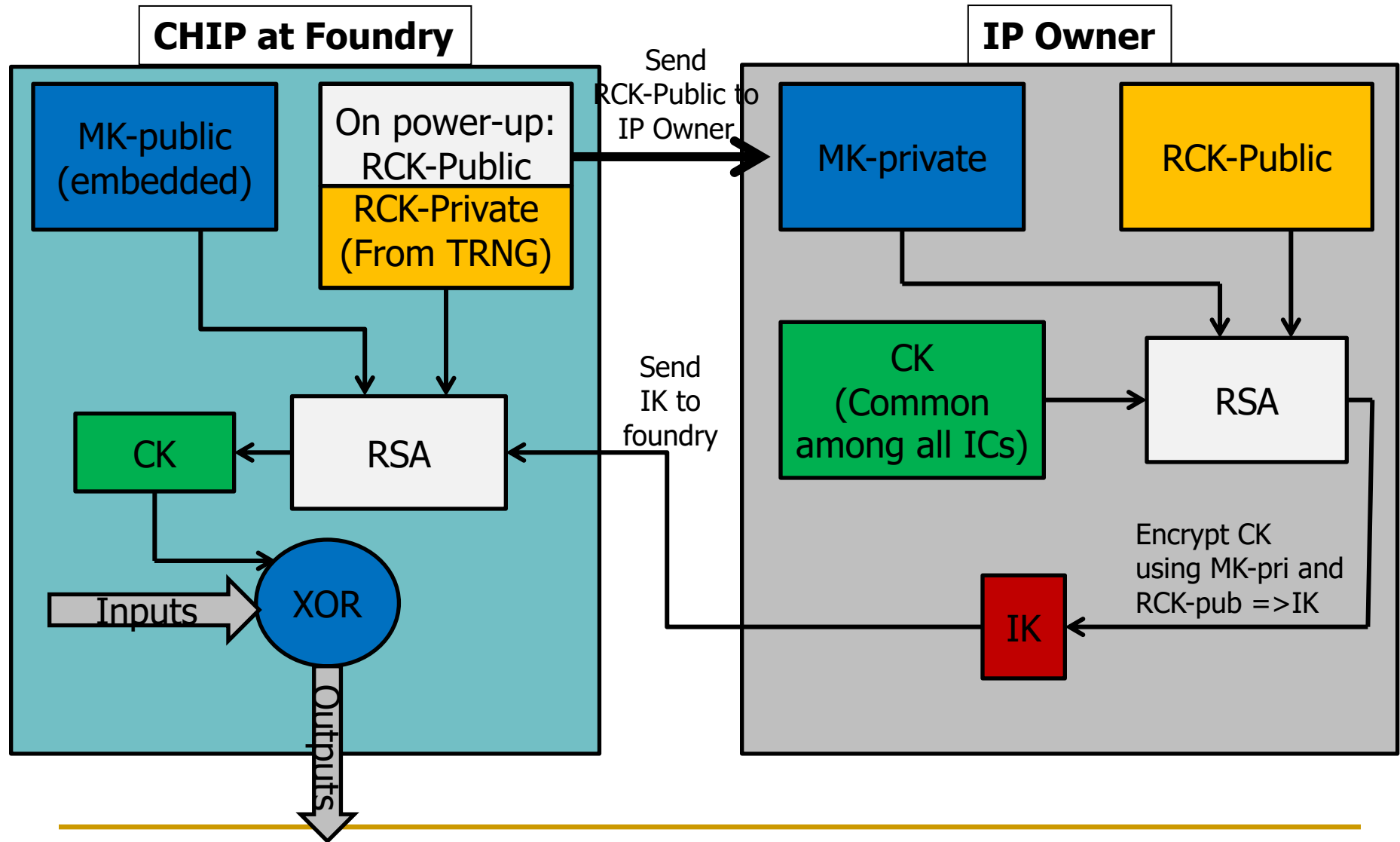


$$1 \wedge 1 = 0$$

$$0 \wedge 1 = 1$$



# EPIC High Level





# EPIC

---

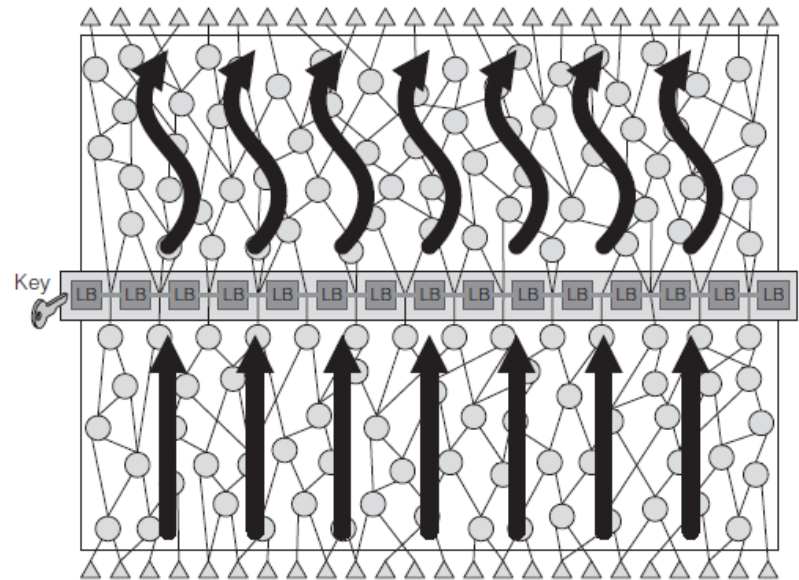
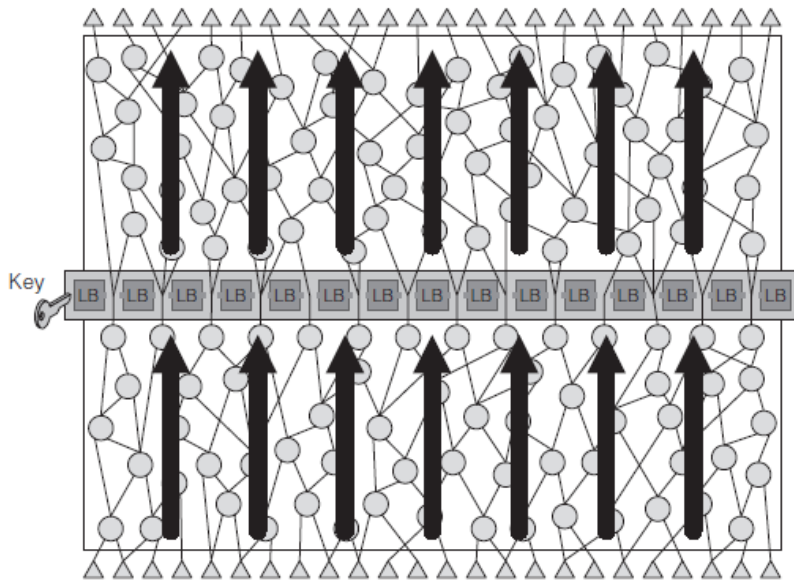
- Embedded in RTL is public Master Key (MK-Pub)
  - XOR gates are controlled by Common Key. Correct Common Key unlocks circuit's correct functionality.
    - k-XOR gates need a common key of length k
  - TRNG (True Random Number Generator) used to generate Random Chip Keys (RCK) on start up.
    - Upon power-up each chip generates a pair of private and public RCKs (RCK-private, RCK-public) which are **burned into programmable fuses**.
  - Fab sends RCK-public to IP owner.
-

# Analysis of EPIC

- Effective against cloned ICs.
  - Cloned ICs: Due to TRNG, each IC will have a unique random key, even cloned ICs. ICs need IK in order to be functional which only IP owner can generate.
- Not efficient against Over-produced ICs, Out-of-Spec ICs and defective ICs.
  - Over-produced ICs:
    - Fab could claim low yield and request more IKs than needed.
    - IP Owner has no way to verify yield or number of functional chips.
    - Foundry can still send keys to IP Owner. Keys are randomly generated and have no information on functionality of the IC.
  - Out-of-Spec ICs:
    - Foundry/assembly can send out the chip that are out of spec (their ID is a correct one)
  - Defective ICs:
    - Once IP owner sends Input Key, chip is activated. If chip is defective, IP Owner has no more communication with foundry and chip is already activated.

# Reconfigurable Logic Barriers (LB)

- Separates inputs from outputs such that every path from input to output passes through a barrier.
- Logic barrier (LB) is a group of logic that allows correct path only if correct key is applied.



# Reconfigurable Logic Barriers

---

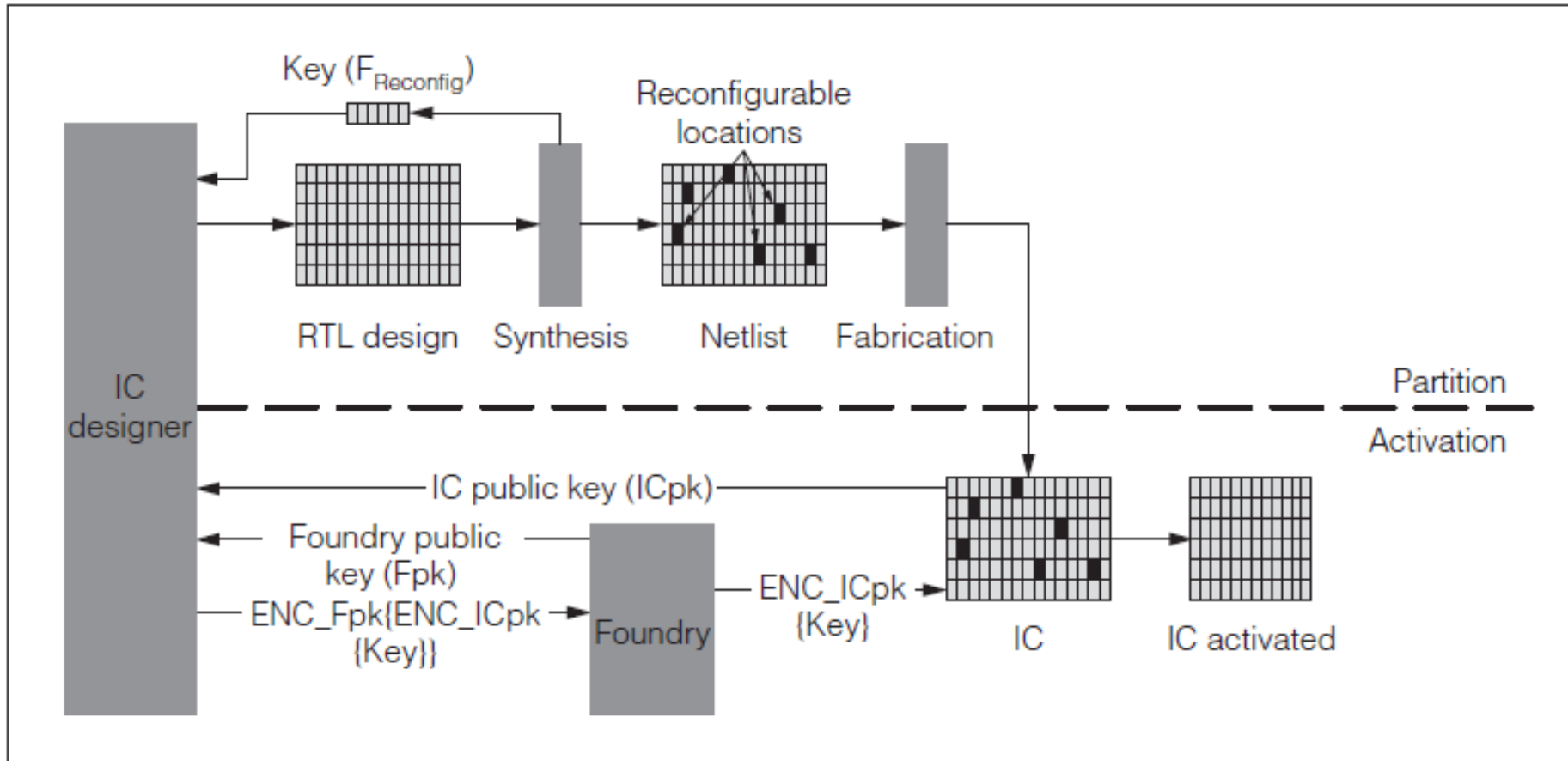
- IP owner decomposes IC functionality into  $F_{\text{fixed}}$  and  $F_{\text{reconfig}}$ .
- $F_{\text{fixed}}$  is given to foundry to fabricate.
- $F_{\text{reconfig}}$  is location of reconfigurable logic in combination with key needed to configure them correctly
- $F_{\text{reconfig}}$  can be programmed into reconfigurable locations using a secure key.

# LB: Public Key Cryptography

---

- ICs use PUFs or TRNGs to generate a private and public random keys.
  - Public key from chip is sent to IP Owner
- IP Owner uses public key and its own private key to encrypt unlocking key.
  - Encrypted key is decrypted on chip using IP Owner's public key and chip's private key.

# LB: Partitioning of Design



# Logic Barriers Analysis

---

## ■ **Effective against cloned ICs.**

- ❑ Chips are only functional if correct key is entered which only IP Owner can provide

## ■ **Ineffective against over-produced, defective, and out-of-spec ICs**

- ❑ Foundry can lower yield in order to receive additional keys to activate functionality.
- ❑ Key generated by chip does not have information about its functionality. Once key is applied, chip is functional.

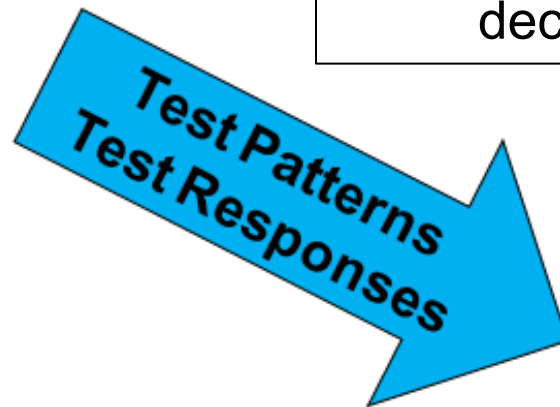
## ■ **Disadvantages:**

- ❑ Look up tables require significant area overhead – 5X more than using XOR gates, and timing overhead.

# Test Seems to be a Challenge!



Designer



Most techniques do not take into account the role “test” plays in the decision making process

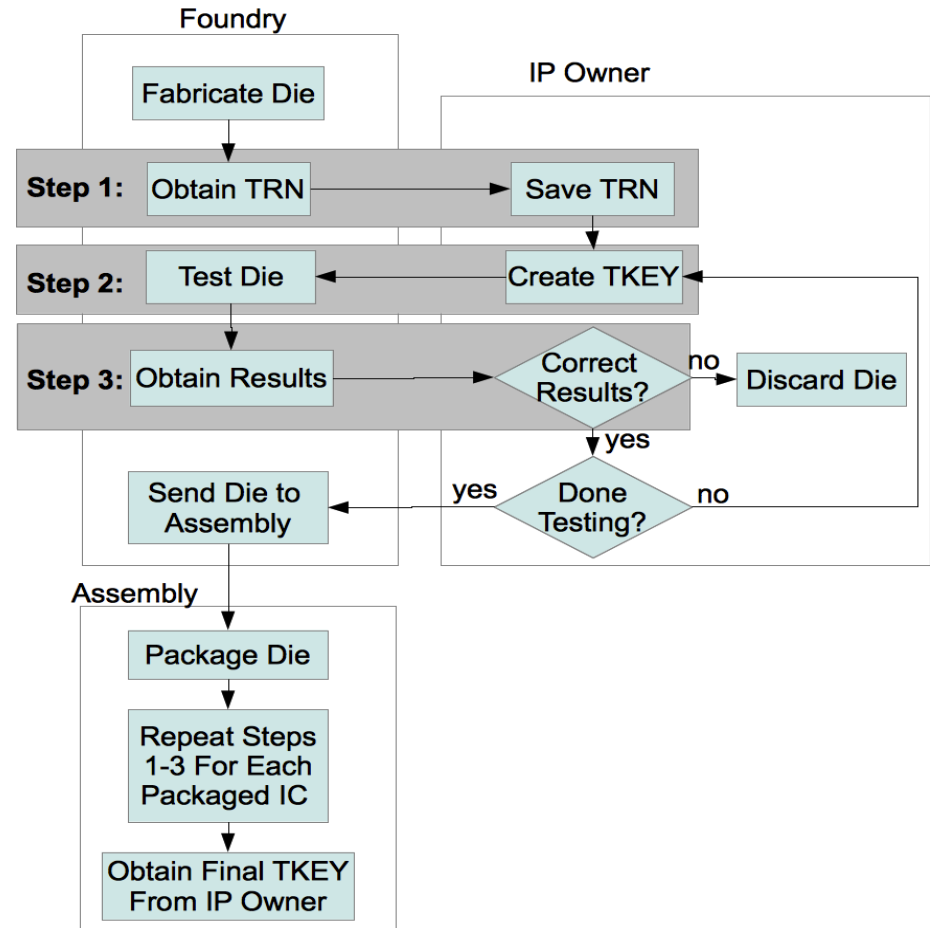


Foundry & Assembly



# Secure Split-Test (SST)

- Adds multiple layers of communication between IP owner, foundry, and assembly
- Ensures that IP owner will know exactly how many chips pass the test and how many have failed.
- Only chips that IP Owner has deemed functional will be given a functional key.



# Secure Split-Test



**Designer**

1. Designer has already put in hooks in the design that can ensure non-functional operation if the correct key is not included in the chip
2. Detecting a non-functional chip is significantly easier than using PUF and dealing with process variations

*Secure Split Test*

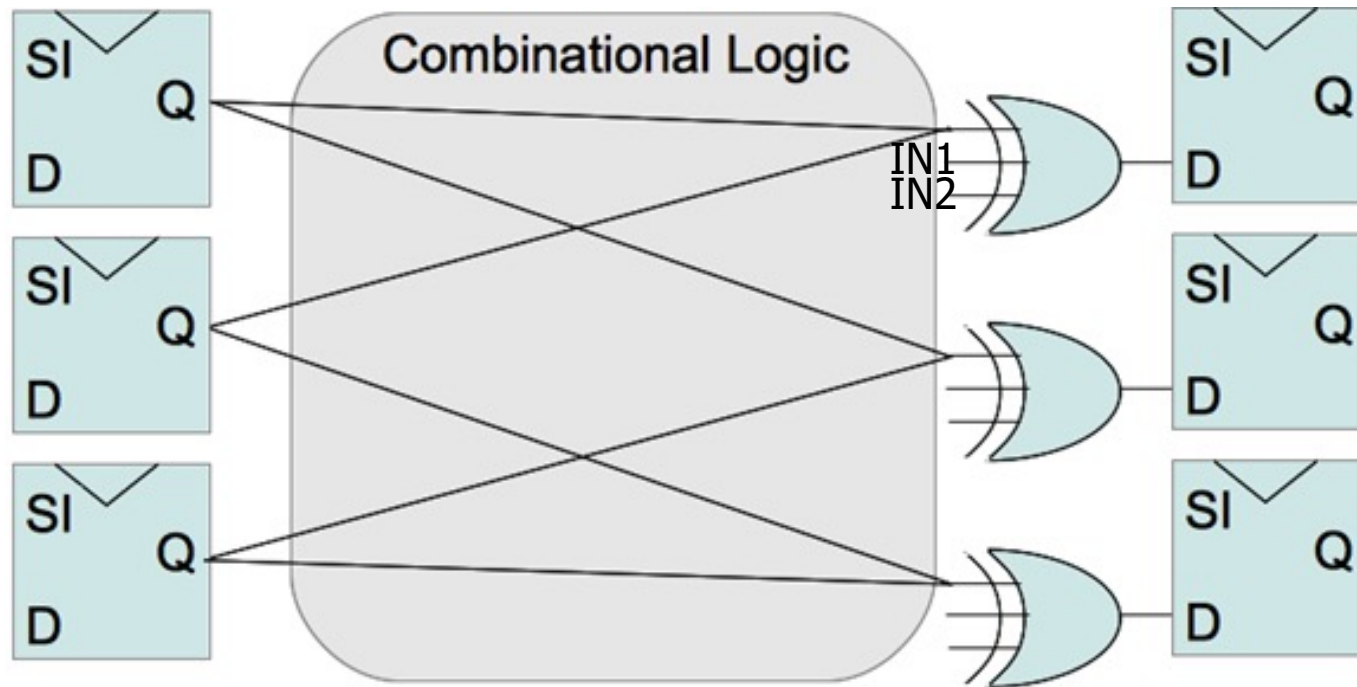
1. Foundry will not be able to ship any functional chips to the market
2. Same for defective chips and out-of-spec chips; the chips are simply non-functional.



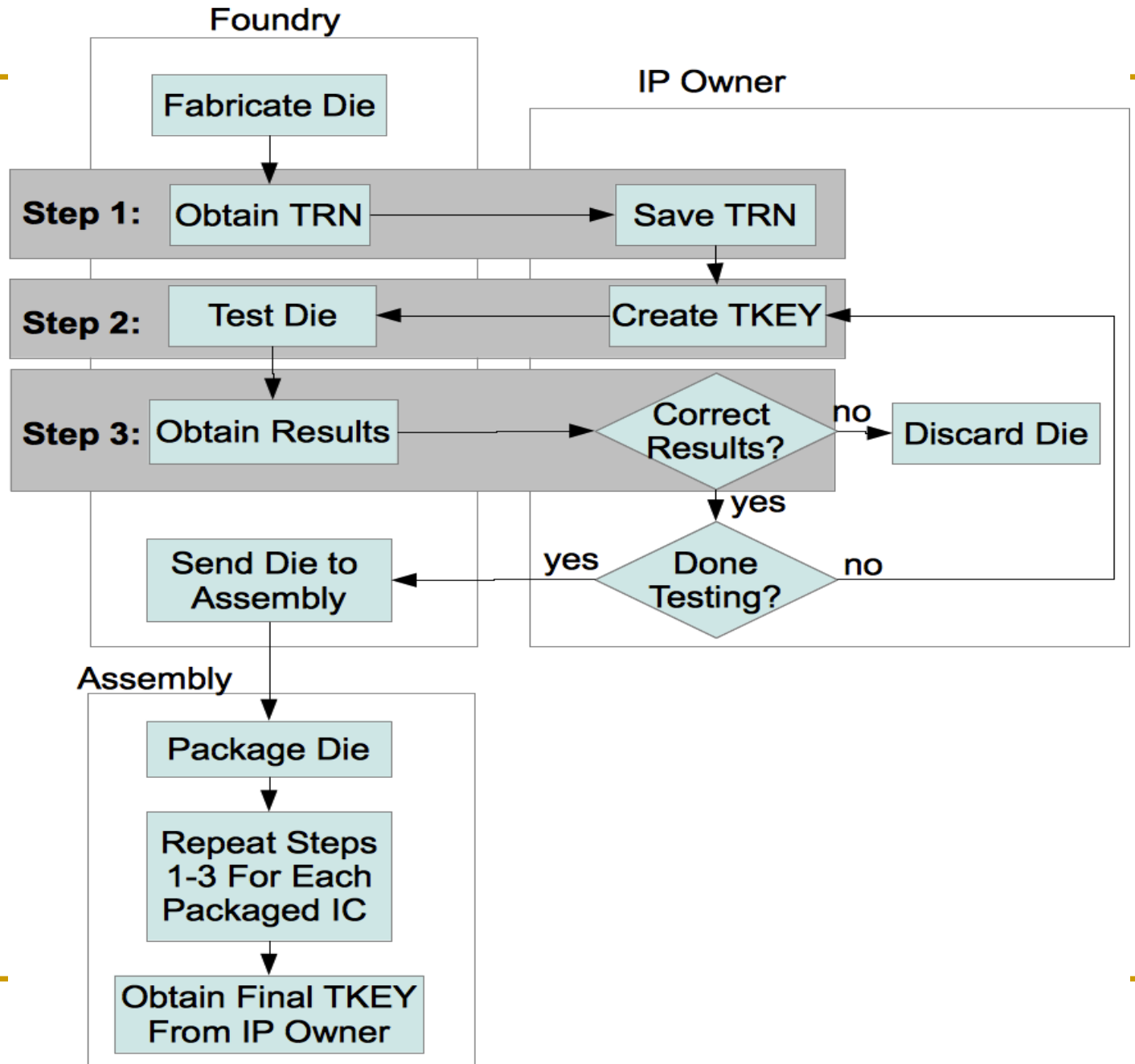
**Foundry & Assembly**

# XOR Mask

- Three-input XOR logic added to non-critical paths.
- XOR logic additional inputs are IN1 and IN2



# SST



# SST Analysis

---

- Effective against overproduced ICs, cloned ICs, and defective ICs
    - **Overproduced:**
      - IP Owner has control over number of TRNs received and TKEY/FKEYS sent to foundry/assembly
    - **Cloned:**
      - Chips are not functional unless FKEY has been produced by IP Owner
    - **Defective ICs:**
      - Foundry sends test results to foundry who checks results and decides if chip has correct test responses (chip is not yet functional at this stage)
-

# SST Analysis

---

## ■ Prevents out-of-spec ICs

- ❑ Some specifications cannot be determined from patterns testing alone. If a chip does not meet these specifications, it could be considered as a passing chip.
- ❑ With the addition of a few sensors on the chip, these specifications can be tested and checked by IP Owner during SST
- ❑ The IP owner will then be able to decide whether or not a chip passes the desired specifications in order to prevent out-of-spec ICs from going into market.

# Remote Activation of ICs Through FSM Modification

- FSM: Finite State Machine
- Sequence of inputs drive machine through different functional states
- Correct transitions give functional output

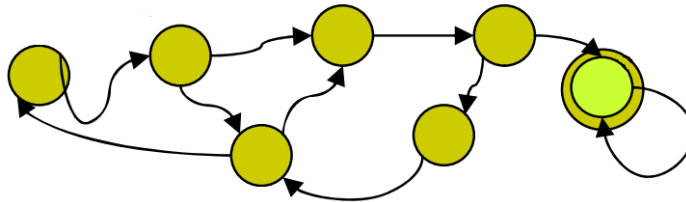


# FSM



- Correct transitions give functional output
- Adding states to FSM gives IP owner controllability over sequence to reach functional states.

**Original states** ●





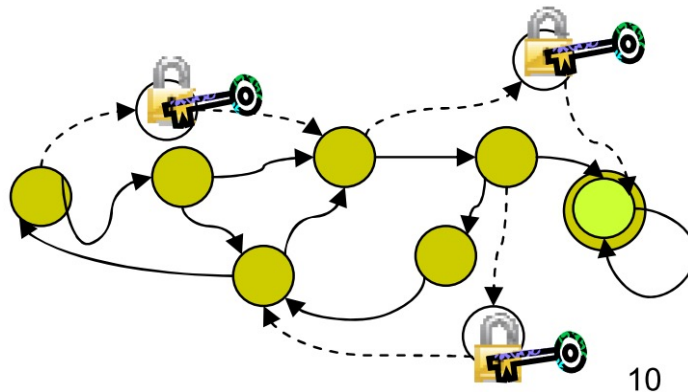
# Boosted FSM (BFSM)



- On startup, inputs cause chip to go to one of added states

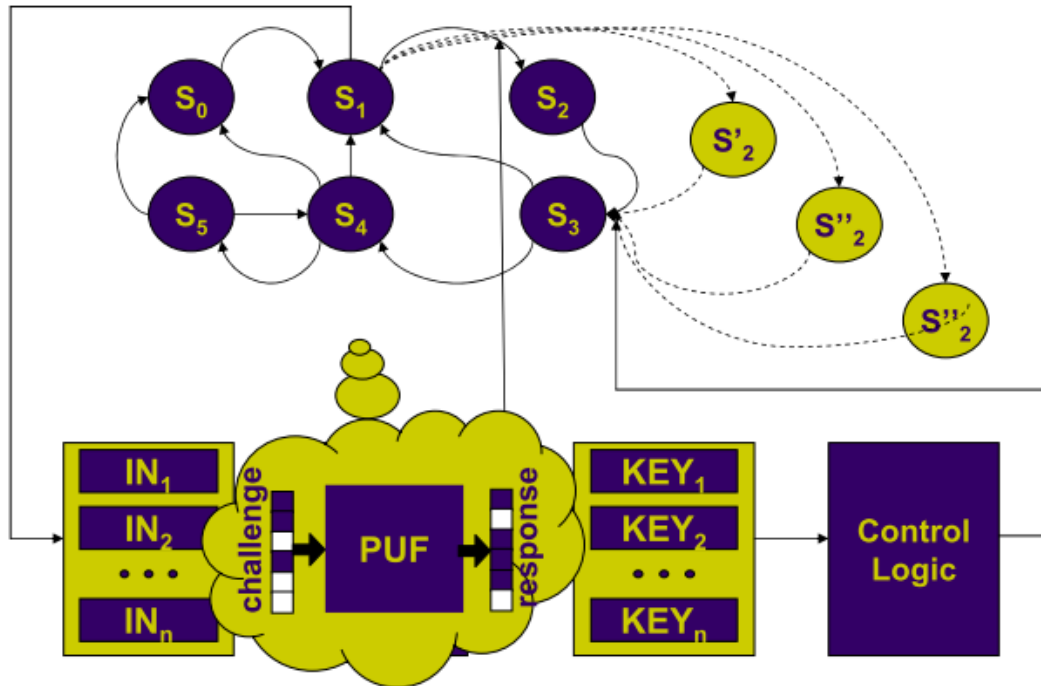
**Original states** ●

**Added states** ○



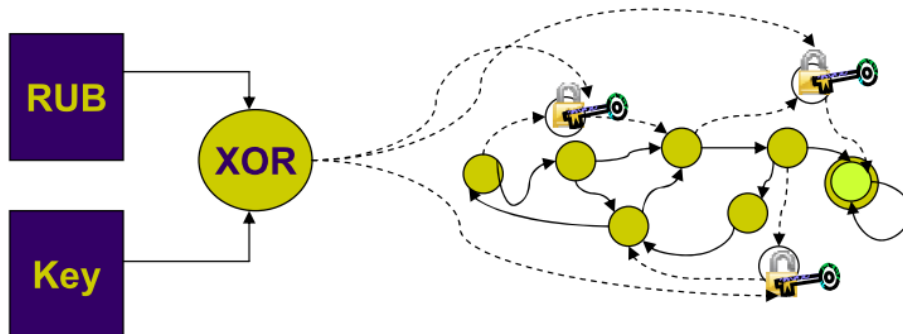
- IP Owner is only one with knowledge of FSM
- Only IP Owner knows right sequence (key) to bring FSM back to functional states.

# Remote Activation of ICs

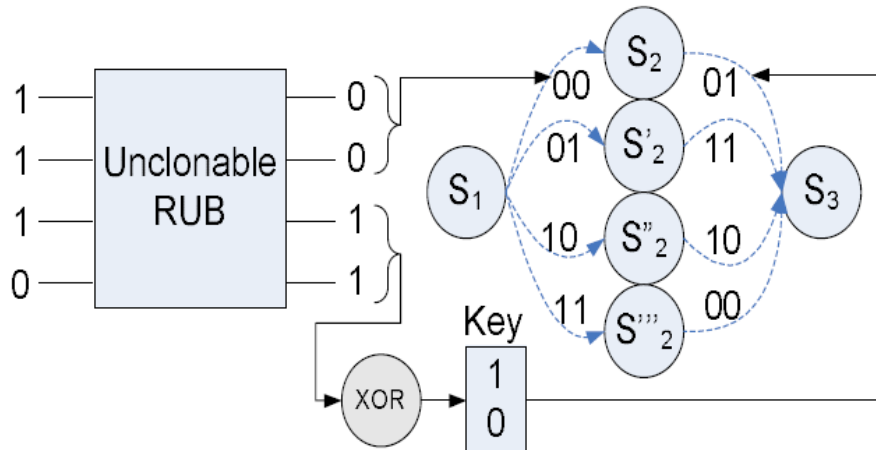


- Redundant states are added.
  - Far less states needed than BFSM
  - PUF response will send FSM to one of redundant states.
- 
- **Challenge:** PUF is yet to be reliable.

# Remote Activation of ICs



- RUB: Random Unique Block
- RUB must be stable – not change over time



- PUF (RUB) response is sent to IP Owner to generate key
- Key is then used to send FSM to correct state.

# Analysis of Boosted FSM and Remote Activation

---

- BFSM requires many additional FSM states.
- Remote activation only uses a few redundant states.
- Both use PUF which is affected by age, temperature, noise, etc.
- Both effective against cloned ICs but not effective against defective, over-produced, or out-of-spec ICs.

# References

- F. Koushanfar and G. Qu. Hardware Metering. In DAC 2001, 2001
- Y. Alkabani and F. Koushanfar. Active Hardware Metering for Intellectual Property Protection and Security. In USENIX Security, 2007
- F. Koushanfar, G. Qu, and M. Potkonjak. Intellectual Property Metering. In IH, pp.81-95, 2001.
- J.A. Roy, F. Koushanfar, and I.L. Markov. EPIC: Ending Piracy of Integrated Circuits. In EDAA, 2008.
- A. Boumgarten, A. Tyagi, and J. Zambreno. Preventing IC Piracy Using Reconfigurable Logic Barriers. In IEEE Design and Test of Computers, 2010
- M. Tehranipoor and C. Wang. Introduction to Hardware Security. Springer, pp. 103-120, 2012
- Y. Alkabani, F. Koushanfar, M Potkonjak. Remote Activation of ICs for Piracy Prevention and Digital Rights Management. In IEEE, 2007
- M. Tehranipoor. VLSI Design Verification and Testing: Test Economics.  
[http://www.engr.uconn.edu/~tehrani/teaching/test/03\\_Test%20Economics%20and%20Product%20Quality.pdf](http://www.engr.uconn.edu/~tehrani/teaching/test/03_Test%20Economics%20and%20Product%20Quality.pdf)
- R.L. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Communication of the ACM, 1978.
- G. Contreras, T. Rahman, and M. Tehranipoor, "Secure Split-Test for Preventing IC Piracy by Untrusted Foundry and Assembly," Int. Symposium on Defect and Fault Tolerance in VLSI Systems (DFT), 2013.
- T. Rahman, D. Forte, Q. Shi, G. Contreras, and M. Tehranipoor, "CSST: Preventing Distribution of Unlicensed and Rejected ICs by Untrusted Foundry and Assembly," IEEE Int. Symposium on Defect and Fault Tolerance Symposium (DFTS), Oct. 2014.
- 
-