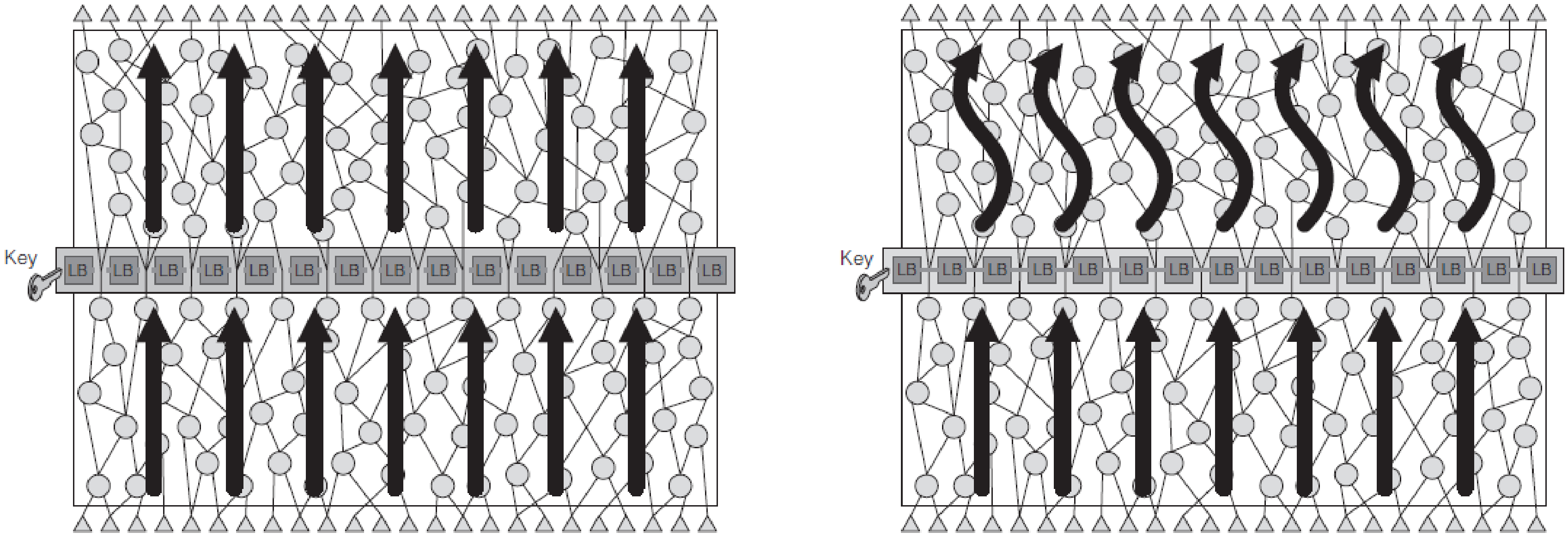
Reconfigurable Logic Barriers (LB

- Reconfigurable Logic Barriers attempts to split a circuit in two parts.
- This is done by inserting logic barriers so that every path coming from input must go through a barrier to go to the output.
- These logic barriers could be different types of gates, in this case they use lookup tables.
- These barriers need a specific key in order to configure the correct paths and make the circuit functional
- If the key is incorrect, the logic barrier will have the wrong path configuration and therefore the wrong functionality

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.


Key
LB LB LB LB LB LB LB LB LB LB LB LB LB LB LB
Key
LB LB LB LB LB LB LB LB LB LB LB LB LB LB LB

Reconfigurable Logic Barriers (LB)

- Reconfigurable Logic Barriers splits the circuit’s functionality into F_{fixed} and $F_{reconfig}$.
- F_{fixed} is the physical circuit with reconfigurable logic (logic barriers) inserted and it is sent to the foundry to be fabricated.
- $F_{reconfig}$ refers to the locations of the inserted reconfigurable logic which only the IP owner knows. With knowledge of the location of logic barriers, the IP owner can generate the key to activate chips.

Reconfigurable Logic Barriers (LB)

- The circuit is NOT physically split in two parts, its functionality is, meaning part of the circuit's functionality is fixed (F_{fixed}) and the other part will change according to the key given to the reconfigurable logic.
- $F_{reconfig}$ is the location of the reconfigurable logic which will program the correct or incorrect functionality of the circuit according to the given key.

Reconfigurable Logic Barriers

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

LB: Public Key Cryptography

- All chips are identical and contain logic barriers on the same location, therefore they all require the same key to be activated .
- In order to prevent foundry from using the same key to activate all chips, LB uses a PUF or TRNG to generate random public and private keys which are unique to each circuit.

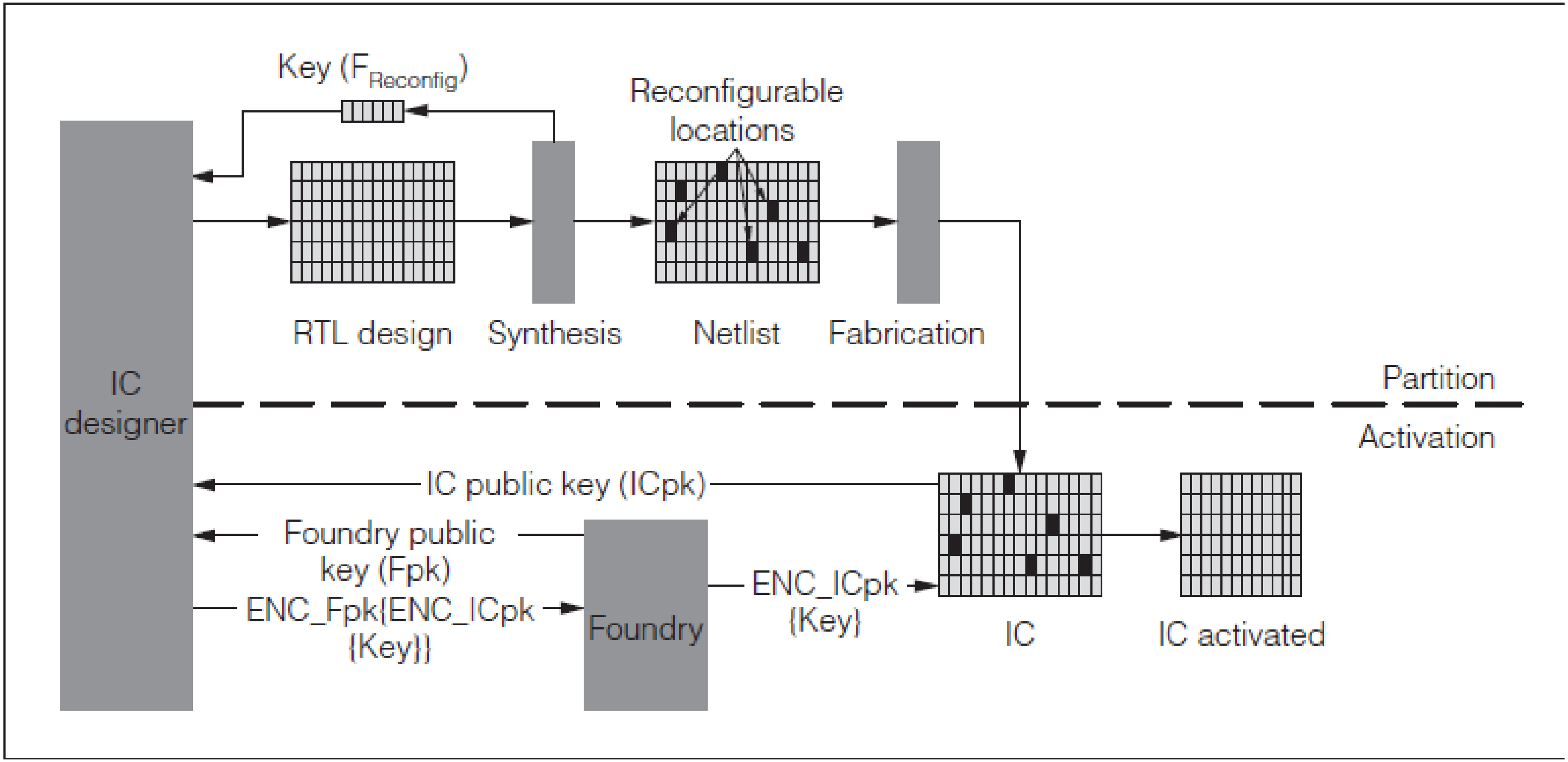
- ICs use PUFs or TRNGs to generate a private and public random keys.
  - Public key from chip is sent to IP Owner

- 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: Public Key Cryptography

- Circuit is designed and synthesized. Reconfigurable logic, i.e. lookup tables, are inserted in specific locations in the circuit. The locations of these lookup tables and the key needed to program them correctly ($F_{reconfig}$) is kept by the IP designer. The circuit is sent to be fabricated.
- After fabrication, in order to test, the chip is turned on and the public key obtained from the TRNG (ICpk). The public key (ICpk) is sent to the designer who uses it to encrypt the common Key.
- (Note: An additional step is shown where the foundry also has a public key (Fpk) which is used by the designer to add another layer of encryption. This step is used to safeguard the communication between the IC designer and the foundry, it does not help protect the chip if the foundry is the untrusted party.)
- The key encrypted using ICpk is given to the chips decryption logic which will decrypt it to the correct Key.

LB: Partitioning of Design

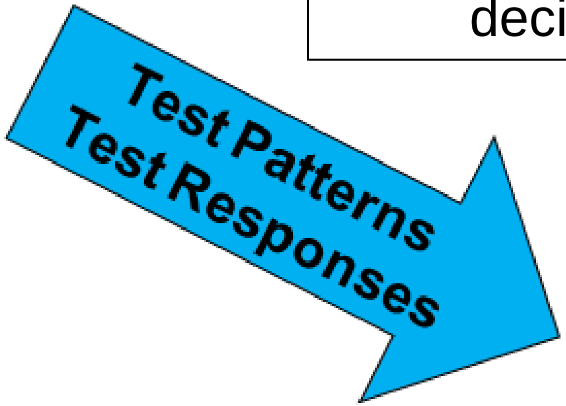


Logic Barriers Analysis

- **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



Test Patterns

Test Responses

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

Foundry & Assembly

Test Seems to be a Challenge!

- Currently, the chip designer provides the foundry and assembly with their chip design and all the test patterns and responses needed to test the chip.
- After this, it is up to the foundry and assembly to fabricate and test the chips.
- An untrusted foundry can easily overproduce chips without knowledge of the designer, it can also mark defective and out-of-spec chips as good chips in order to increase yield, or can send those defective and out-of-spec chips to the market in order to make up for the loss due to low yield.

Secure Split-Test (SST)

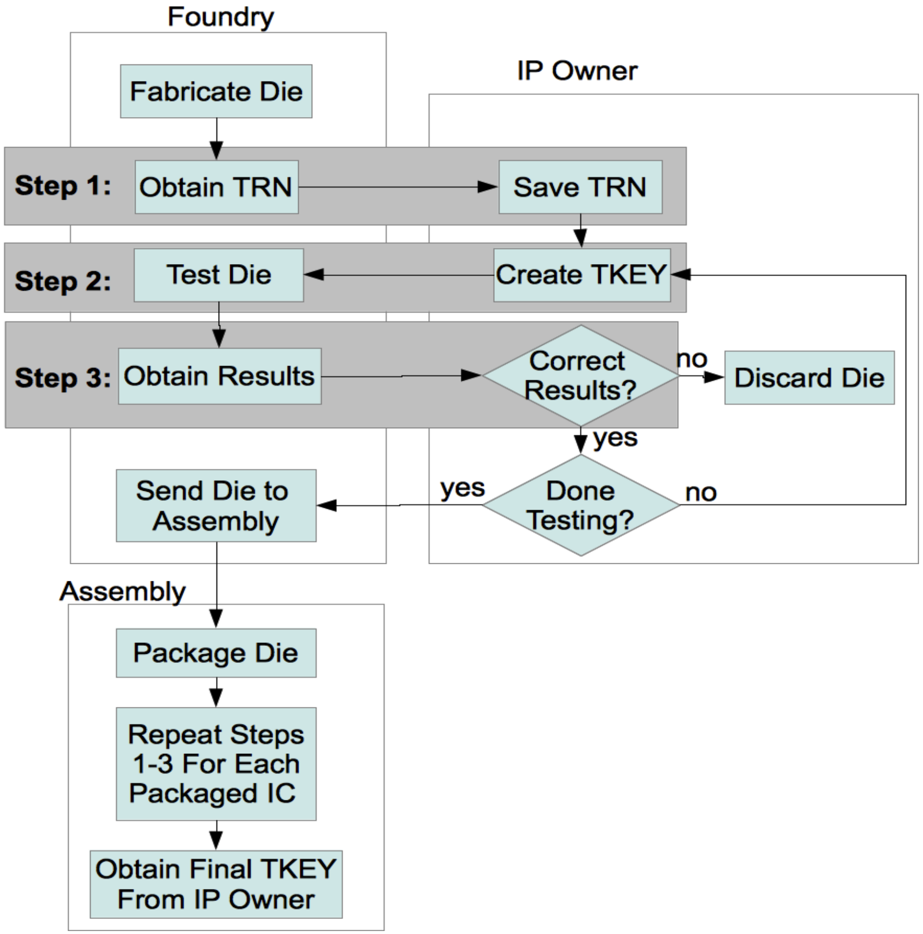
- Secure split test aims to add more layer of communication between foundry and IP Owner.
- It actively involves the IP owner in the testing and verification process of fabricated chips so that the IP owner will know which chips have passed and which have failed.
- This method also has the advantage of being able to calculate the yield from foundry and therefore prevents foundry from claiming a lower yield and what it actually is preventing overproduced ICs.
- The IP owner has control over how many chips it activates to be fully functional. Unlike other methods which activate the chips before testing in order for them to be tested correctly,
- Secure Split test activates the chips after testing, only after IP owner has verified their correct functionality, thus preventing defective and out-of-spec ICs.

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 (SST)

- Secure Split Test makes sure that only the exact number of good chips are sent to the market by only activating a specific number of chips that have passed functional tests.
- Usually with defective or out-of-spec ICs, the chips will seem functional and will only fail when a specific input or pattern in applied. Depending on the application, this pattern could be so specific that the chip could go years working correctly and only fail when used a certain way.
- This method is designed to make all overproduce, defective, and out-of-spec chips not functional. Thus, even if these chips are sent to the market, they will fail as soon as they are used the first time because they will not function correctly at all.

Secure Split-Test

Designer

Secure Spilt 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.

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

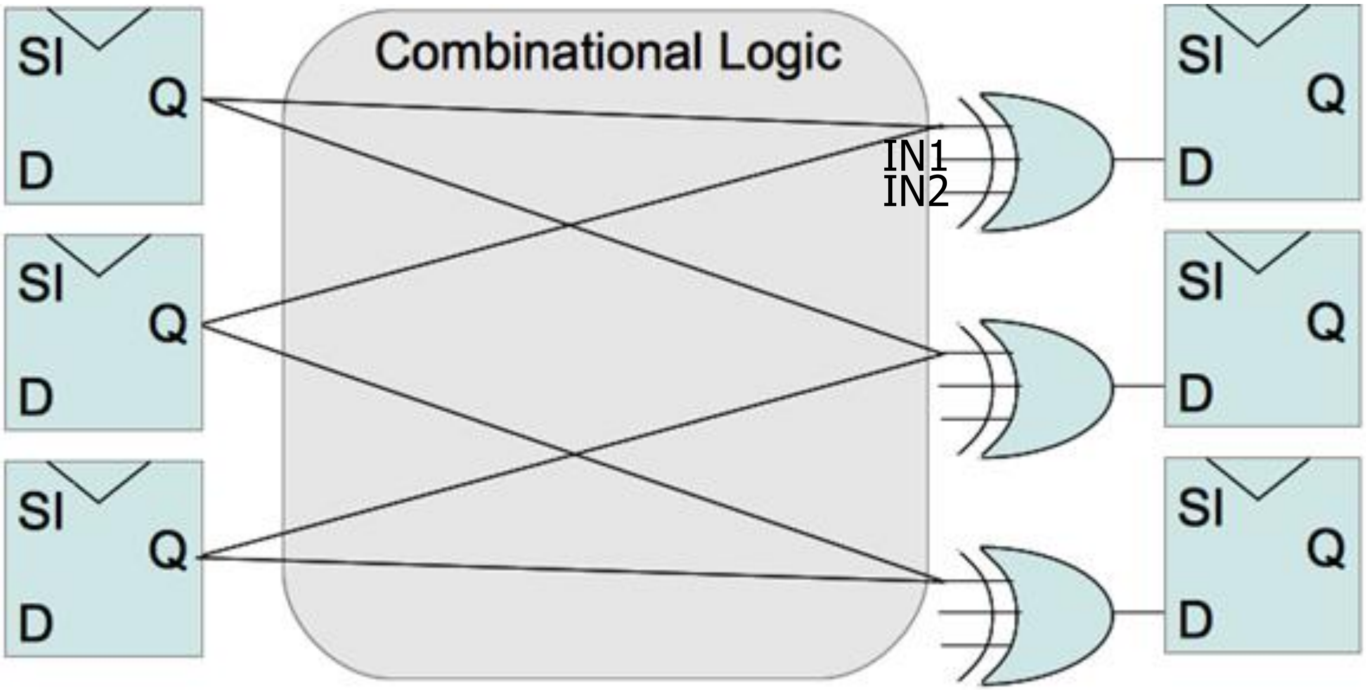
Foundry & Assembly

Secure Split-Test

- The first step of SST is the insertion of 3-input XOR gates at the input of flip-flops on non-critical paths. Inserting them at the flip-flop inputs means that these XORs are receiving the responses coming from the combinational logic before it. Depending on the value of the other XOR inputs, either the correct response will be stored into the flip flops or an inverted version of the response will be stored.
- The first input of the XOR gate is connected to the path.
- The other two inputs are called IN1 and IN2. Only when IN1 and IN2 are the same, the XOR will be transparent, if they are different, then the path will be inverted.

XOR Mask

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



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

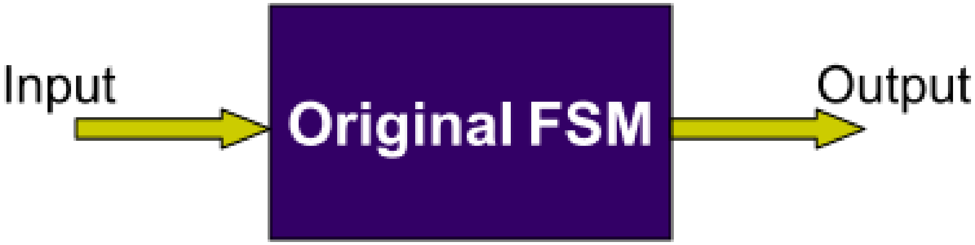
- **Prevents out-of-spec ICs**
  - Some specifications cannot be determined from patterns testing alone. If a chip does not meet these specifications, it could be considered as a passing chip.
  - With the addition of a few sensors on the chip, these specifications can be tested and checked by IP Owner 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

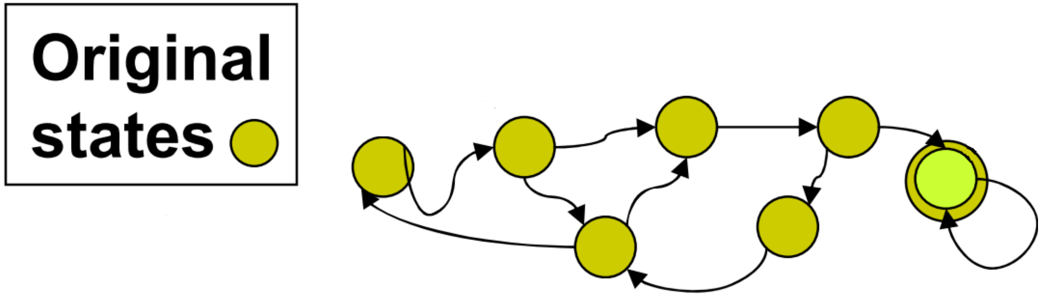
- These methods make use of PUFs or TRNGs, the main difference between these and the previous methods is that no encryption logic is required for these methods.
- FSM: Finite State Machine
- Sequence of inputs drive machine through different functional states
- Correct transitions give functional output

Input
Original FSM
Output

FSM

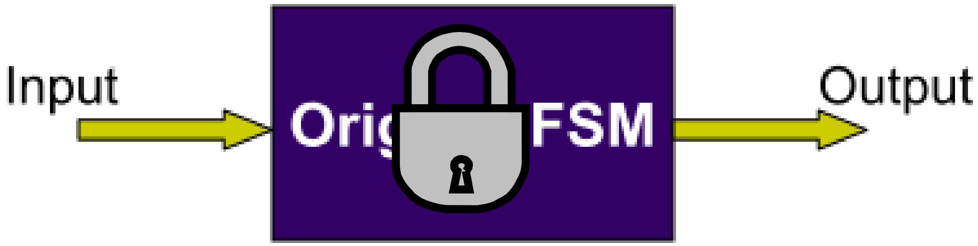


- Correct transitions give functional output

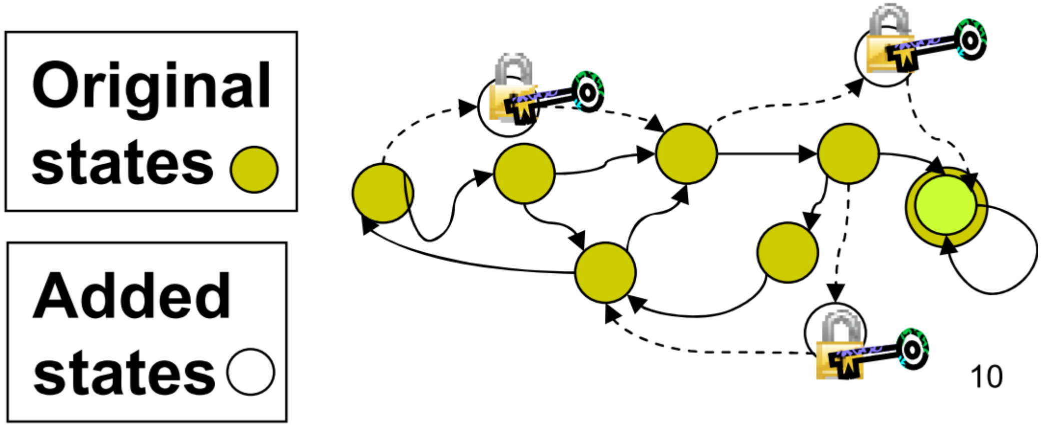


- Adding states to FSM gives IP owner controllability over sequence to reach functional states.

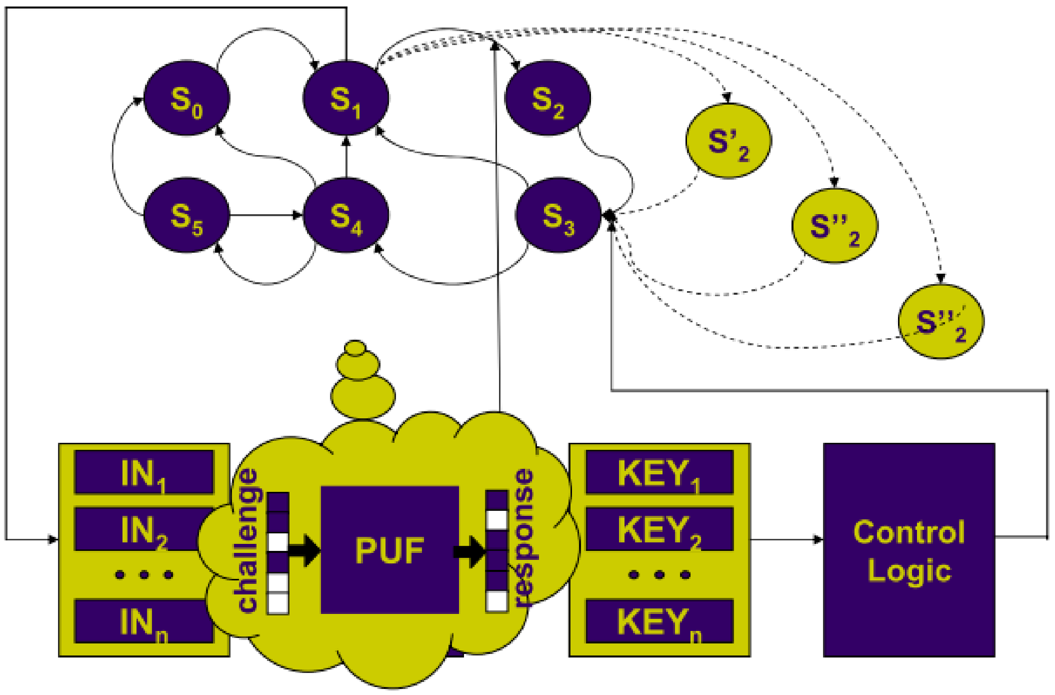
Boosted FSM (BFSM)



- On startup, inputs cause chip to go to one of 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.



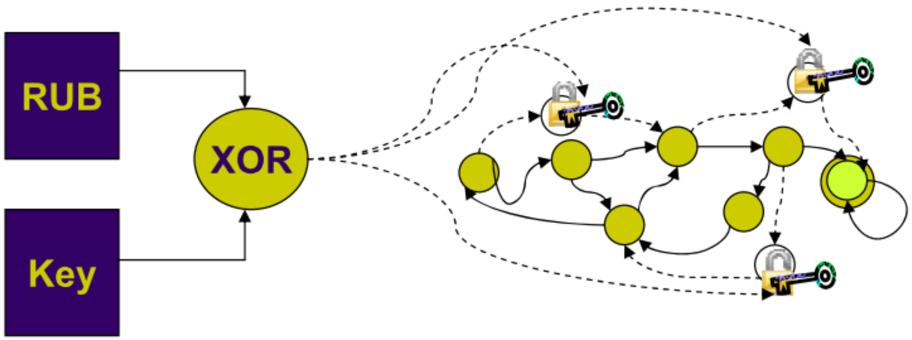
Remote Activation of ICs

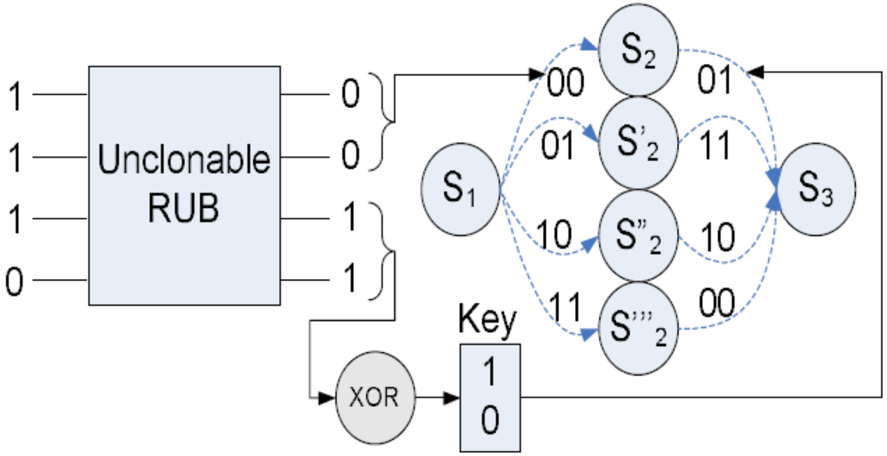


- Redundant states are added.
- Far less states needed than BFSM
- PUF response will send FSM to one of redundant states.

- **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.

Differences

Feature	Boosted FSM	Remote Activation FSM
Activation Mechanism	Internal key or sequence	External remote key or signal
Focus	Enhancing FSM complexity	Enabling remote control
Dependence	Depends on an internal FSM key	Depends on external communication
Security Application	Prevents reverse engineering	Ensures supply chain control
Common Use	Cloning prevention	Supply chain security

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.