DAFCA

Hardware security

Integrated Circuit Security – New Threats and Solutions

Miron Abramovici and Paul Bradley

DAFCA

Cyber Security Workshop

April 2009

Outline

- **The IC security problem**
- Possible solutions?
- An interesting technology
- A new approach
- Conclusions

The IC Security Problem

- **Insecure global sources of:**
  - IC manufacturing
    - Silicon foundries
    - Packaging houses
  - IP cores
  - Design services and tools
  - Test services and tools
- **Attack objectives**
  - Sabotage of the system mission
  - Extract sensitive/secret information
- **Compared with attacks on software, attacks on ICs are much more difficult to detect and defend against**

Attacks We Defend Against

- **Pre-silicon intrusions: design is modified to create Trojans to be activated in the field**
  - Attacks occur post-silicon
  - Possible targets
    - Functional logic
    - Infrastructure logic
    - Layout (deterioration attacks)
- **Post-silicon tampering**
  - Make the device work in “illegal” or “unauthorized” modes to extract protected data or to reverse engineer
  - Modify the silicon (using FIB)

Possible Pre-Silicon Intrusions

- **Modifications of functional logic**
  - Corrupt IP cores
  - Clock logic
  - Voltage control logic
- **Modifications of infrastructure logic** (DFX)
  - Testability
  - Manufacturability
  - Reliability
  - Debug
- **Modifications of layout**
  - Thinner conductors
  - Weaker transistors

Possible Post-Silicon Tampering

- **Modify operating conditions**
  - Temperature
  - Power
  - Frequency
  - Radiation
- **Use FIB to cut or create connections**
- **Force “illegal” operations**
  - Invalid or undocumented instructions
  - Create protocol errors
  - Denial of service
  - Access functional data via debug or test operation

Complex Attack Scenarios

1.

- Modify netlist or layout to insert an unconnected Trojan (using spare gates)
- FIB tampering to connect the Trojan to functional logic
- Trojan activated post-deployment
  - at some future time (time bomb)
  - triggered by an event (booby trap)

2.

- Modify clock logic and debug/test logic
- Stop clock during normal operation and
  - Use scan chains to extract critical data from registers
  - Use RAM BIST to extract critical data from memory

Outline

- The IC security problem
- **Possible solutions?**
- An interesting technology
- A new approach
- Conclusions

Can These Provide Solutions?

- **Off-line hardware manufacturing test**
  - NO: Trojans activated after deployment
  - NO: Hidden logic inactive in test mode
  - NO: Hardware tests are based on the known model
- **Pre-silicon design verification**
  - NO: tests are not exhaustive
  - NO: Trojans not activated in the verification models
  - NO: Trojans may be hidden in infrastructure or analog parts
- **Reverse engineering of a suspect IC**
  - NO: only certain ICs may be attacked
  - NO: not scalable (may take too long)
  - NO: may not have a golden reference

Can These Provide Solutions?

- **Formal correctness proofs**
  - NO: may not have the real netlist or RTL
  - NO: silicon may be modified by FIB
- **Compare behavior of suspected chip with golden model**
  - NO: most mismatches do not matter
  - NO: cannot do exhaustive testing
  - NO: may not have a golden reference
- **Thermal analysis and other non-destructive techniques**
  - NO: may not have a golden reference

Golden Model = Illusion

- **The highest level RTL model may be corrupted by IP cores with hidden logic**
- **No golden models for infrastructure logic**
- **Fabricating the same IC in a secure environment may still be affected by untrusted tools**
  - Even if we have a golden chip, behavior comparisons are not practical or reliable

The Bottom Line

- **Cannot guarantee that deployed chip does not carry unintended logic**
- **Unacceptable risk for ICs used in critical missions or infrastructures**
- **Must do on-line checks** (to complement necessary, but not sufficient, pre-deployment checks)
- **Use same on-line checks to detect tampering**
- **After detection, must also provide countermeasures**

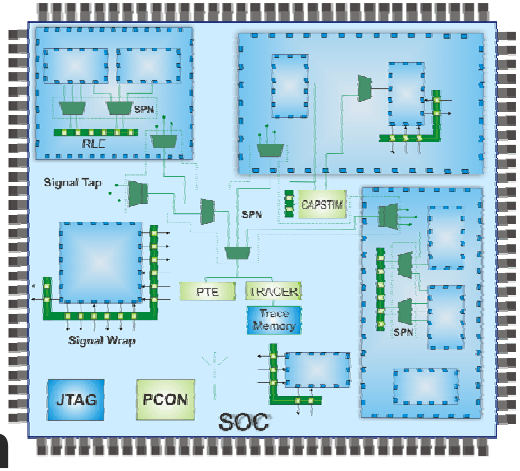
- **What to check?**
- **How many checks are practically feasible?**

Outline

- The IC security problem
- Possible solutions?
- **An interesting technology**
- A new approach
- Conclusions

DAFCA Technology

- **Distributed reconfigurable logic**
  - Soft macros inserted at RTL
  - Synthesized together with functional logic
- **Provides a reconfigurable infrastructure platform**
- **Configured and controlled**
  - via JTAG
  - from an embedded processor
  - configuration can be done at any time
- **Does not interfere with normal operation**
- **Invisible to the application software**
- **Reusable for many applications**



DAFCA
Hardware Security

Applications

- **In-system at-speed silicon validation and debug** (silicon-proven)
  - Logic analysis
  - On-chip functional test
  - Assertions in silicon
  - In-system scan-based debug
  - Performance monitoring
  - Fault and error injection
  - Hardware-software co-debug
- **...**
- **Extensions for IC Security**

Outline

- The IC security problem
- Possible solutions?
- An interesting technology
- **A new approach**
- Conclusions

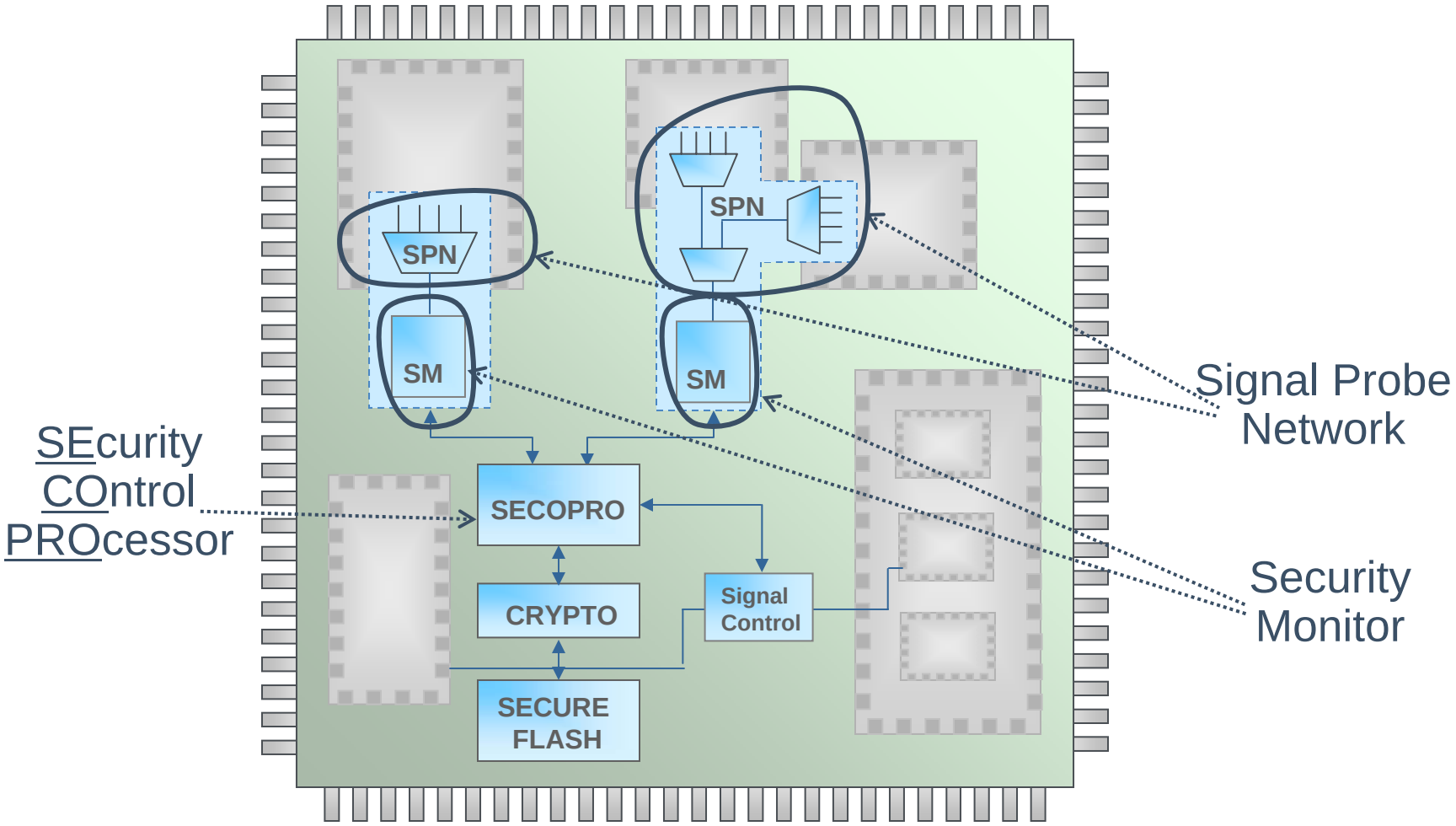
A New Approach to IC Security

- **Must defend against attacks in mission mode**

- **Add Design-For-Enabling-Security (DEFENSE) logic for**
  - on-line Security Monitors
  - counter-measures to detected attacks
- **DEFENSE logic should be**
  - invisible to the functional logic
  - invisible to the application software
  - impossible to understand by analyzing the netlist

Current DAFCA technology provides the basis to satisfy these requirements

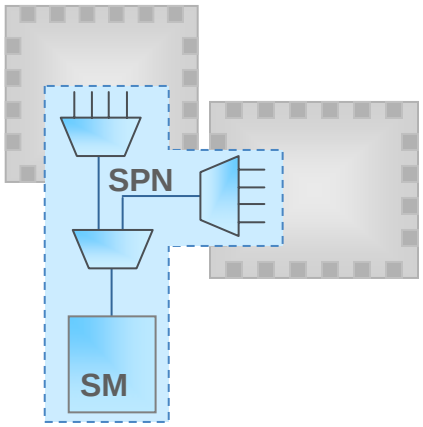
SoC with DEFENSE Logic



DAFCA Hardware Security

Signal Probe Network

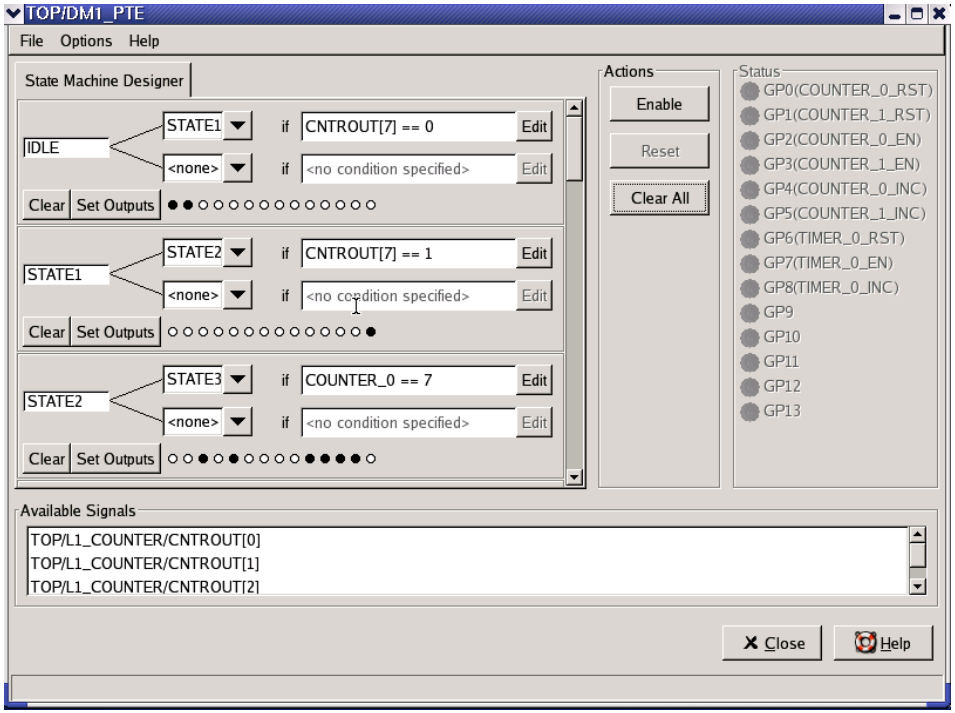
- **Distributed pipelined MUX network**
- **Configured on-line to**
  - select signals to be monitored
  - connect selected signals to Security Monitors
- **Repeatedly configured to analyze different groups of signals**
- **Connections signals → SPNs → SMs provide redundancy to increase probability of surviving attacks**

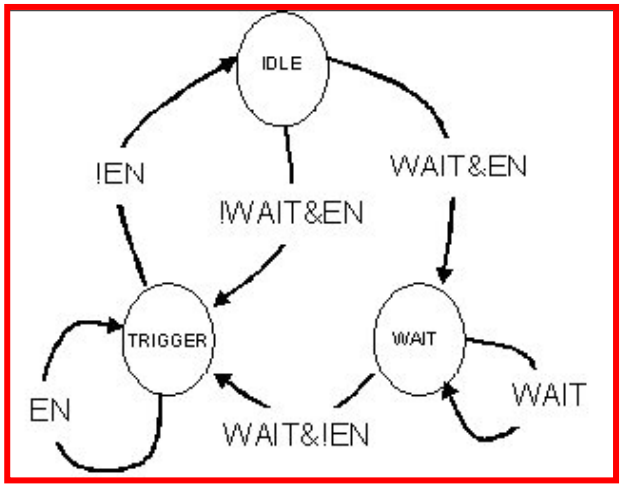


DAFCA
Hardware Security

Security Monitor

- Contains reconfigurable logic resources
- Configured to implement a finite state machine (FSM) to check relations among its input signals





DAFCA
Hardware Security

What Do Monitors Check?

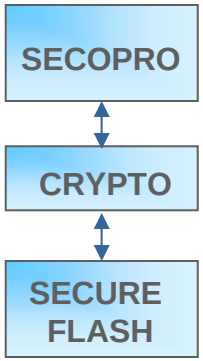
- **Security violations**
  - Access to a restricted address space
  - A control signal supposed to be inactive is activated
  - A core responds to a request addressed to another core
  - A core whose clock is turned off has output changes
  - Denial of service
  - Test mode asserted in normal operation
- **General correctness properties**
  - Standard communication protocols (AMBA, PCI, etc)
  - Block-specific
  - Signal-specific

How Many Checks?

- SPNs can be repeatedly configured to bring different groups of signals to be analyzed
- **Security Monitors can be repeatedly configured to implement different checks**
- → A group of checks can be run concurrently for a limited interval
- → SECOPRO continuously runs one group of checks at a time
- → Reconfigurability allows time-sharing of hardware for large number of security checks

Security Control Processor

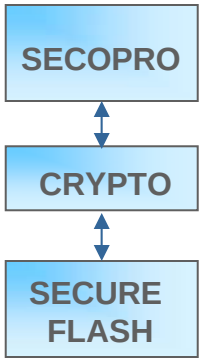
- **Separate from application processors**
- **Its control logic is configured on power-on** (obfuscation)
- **Configures and controls SPNs, Monitors, and Signal Controllers**
- **Performs periodic self-checks of the DEFENSE platform**
- **Designed with duplicated units to increase probability of surviving attacks**



DAFCA
Hardware Security

Secure Flash Memory

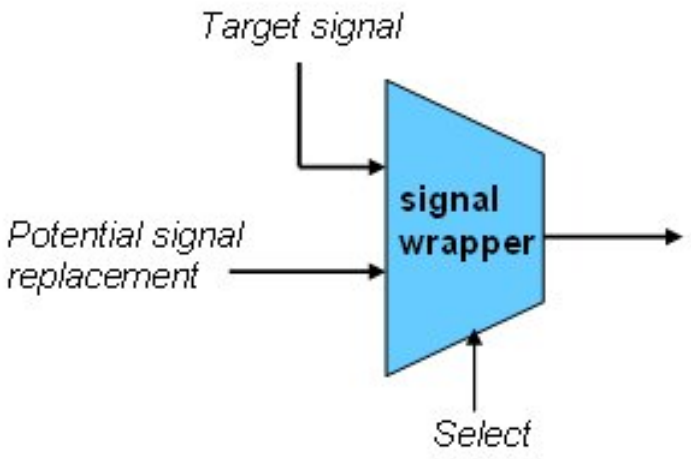
- **Stores encrypted configurations for**
  - SECOPRO
  - Security Monitors
  - SPNs
- **Key is locally generated by Physically Unclonable Function (PUF)**
- **Key is not known outside the chip**
- **Non-volatile memory loaded only in secure environment**



DAFCA
Hardware Security

Countermeasures

- **Need to override signals**
- **Wrapping signals adds controllability**
- **Examples of countermeasures against an offending core**
  - Disable clocking
  - Power off
  - Continuously reset
- **Also need system-level countermeasures**
  - Replace with a spare
  - Put the chip in a recovery state
  - Wipe out confidential data
  - Stop operation



DEFENSE Logic Requirements

- **Invisible to the functional logic** ✓
- **Invisible to the application software** ✓
- **Impossible to understand by analyzing the netlist** ✓
  - It's gates and flip-flops (no hard macros)
  - Its function is "not there" without configuration bits

DEFENSE logic and functional logic are interspersed
➔ the functional logic is also more difficult to understand

Outline

- The IC security problem
- Possible solutions?
- An interesting technology
- A new approach
- **Conclusions**

A New Approach to Hardware Security (1)

We propose the configurable DEFENSE platform

- **Natural extension of our commercial solution**
- **Effectively invisible**
- **Performs a large numbers of complex on-line security checks**
- **Detects a large spectrum of security attacks** (Trojans, tampering, time bombs, booby traps, deterioration)
- **Complements pre-deployment solutions**

A New Approach to Hardware Security (2)

Reconfigurable DEFENSE platform for SoCs

- **Application-independent and technology-independent**
- **Supports user-defined countermeasures**
- **Can accommodate new checkers for new threats** (remote reconfiguration)
- **Equally applicable to ASICs, ASSPs, and FPGAs**