Introduction to the Hardware Trojan Problem
Globalization

- Companies worldwide develop ICs
- Designed, Fabricated, and Assembled separately
  - More companies, more vulnerabilities
  - Fab-less Designers
Globalization

- IP Cores
  - Reusable modules
  - Licensed to designers
  - Present at each abstraction level
- SoC Designs
- Too costly to reverse globalization
HW Threats

Any of these steps can be untrusted
HW Threats

IP Vendor

System Integrator

Manufacture

IP Trust

IC Trust

Untrusted
Issues with Third IP Design

Company X

System-on-chip (SoC)

Company Z

Company V

Company Y

Company W
Issues with Third IP Design

These companies are located across the world

There is no control on the design process

Company X

System-on-chip (SoC)

Company Y

Company Z

Company V

Company W
HW Threats

- IP Vendor
- System Integrator
- Manufacture
- IP Piracy
- IC Trust

Untrusted
HW Threats

- IP Vendor
- System Integrator
- Manufacture

Untrusted Foundry
- IC Trust
- IC Piracy (Counterfeiting)
- Secure Manufacturing Test
IC/IP Trust Problem

- Chip design and fabrication is becoming increasingly vulnerable to malicious activities and alterations with globalization

- Design and Foundry:
  - A designer/foundry can add functionality to the design

- An adversary can introduce:
  - A Trojan designed to disable and/or destroy a system at some future time
  - A Trojan that may serve to leak confidential information covertly to the adversary
IC/IP Trust Problem

Chip design and fabrication is becoming increasingly vulnerable with globalization.

Design and Foundry:
- A designer/foundry can add functionality to the design.

Adversary:
- An adversary can introduce:
  - A Trojan designed to disable and/or destroy a system at some future time.
  - A Trojan that may serve to leak confidential information covertly to the adversary.

U.S. Senate, 2003
Defense Science Board, 2005
Semiconductor Equipment and Materials Industry (SEMI), 2008
IEEE Spectrum, 2008
IEEE Symposium on Hardware-Oriented Security and Trust (HOST)

More articles have addressed this issue within the last few years.
Untrusted Designer and Foundry

Design Process:
- IP
- CAD Tools
- STD Cells
- Models
- Design Specification

Fabrication Process:
- Fab Interface
- Mask
- Fab

Manufacturing Test Process:
- Wafer Probe
- Dice & Package
- Package Test

IC Authentication:
- Trojan Detection and Isolation

Deploy and Monitor

Trusted
Either
Untrusted
Applications and Threats

Thousands of chips are being fabricated in untrusted foundries
Hardware Trojan – Back Door

- Adversary can place an Antenna on the fabricated chip.
- Such Trojan cannot be detected since it does not change the functionality of the circuit.
- Adversary can send and receive secret information.
- Adversary can disable the chip, blowup the chip, send wrong processing data, impact circuit information etc.
Time Bomb

Untrusted Hardware

Counter
Finite state machine (FSM)
Comparator to monitor key data
Wires/transistors that violate design rules

- Such Trojan cannot be detected since it does not change the functionality of the circuit.
- In some cases, adversary has little control on the exact time of Trojan action
- Cause reliability issue
Defining the Problem

Photo Credit: Meter Mulligan. 2007. Under the Creative Commons license.
Hardware vs. Software Trojans

Hardware Trojans
- A Trojan is inserted into an IC
- Once inserted, the Trojan behavior cannot change
- An IC is very much like a black box, a Trojan cannot be observed

Software Trojans
- A Trojan is part of the code in software
- A Trojan behavior can change
- A Trojan can be added to a software via network
- Once identified, it can be removed and added to a database to look for it in the future
Taxonomy

Taxonomy: Insertion Phase

- Functional blocks
- Protocols
- Mask
- Fab
- Package
- System assembly
- System deploy
- Specification
- Design
- Fabrication
  - Third-party IP blocks
  - Standard cells
  - Synthesis and simulation tests
  - Hardware Description Language
- Testing
  - Test vectors
- Assembly
  - Automatic test equipment
Taxonomy: Abstraction Level

- **System level**
  - **Register-transfer level**
    - **Gate level**
      - **Transistor level**
        - **Physical level**
          - **Modify parameters**
    - **Change logic**
  - **Modify constraints**
  - **Add/modify gates**
  - **Modify layout**
  - **Modify wiring**
  - **Change protocol**
Case Study: RTL Trojan

- Code segment of 8051 microprocessor in VHDL
- Trojan changes program counter behavior
  - Increment maps to accumulator jump
  - Behaves normally while inactive
- Cannot directly control number of gates used
Case Study: Gate Level Trojan

- Gate Level Trojan to attack cryptographic hardware
  - Trigger seeks "10100011"
  - On trigger, encryption is skipped
- Particular gates used can be controlled
  - Location cannot
- Practical GL Trojans are in netlist form
Taxonomy: Activation Mechanism

- Also called the "trigger"
- A rare trigger makes a Trojan stealthier
  - not always possible
- Adversary goal:
  - Adversary can predict or induce triggering
  - User / chip tester cannot
Internal vs. External

- Externally Triggered
  - Depends directly on external inputs
  - Can be both user and component driven
    - e.g. transmitter
- Internal
  - Can also include internal signals
Case Study: Physical Condition
Case Study: Time Bomb Trigger

- Subclass of time-based
  - Called "time bomb"
- Weaknesses
  - What if chip tester waits long enough?
  - Increasing time increases area
    - $O(\log_2(n))$

Example:
$1\text{GHz} \times 1\text{ day} = 8 \times 10^{13}$
$\log_2(8 \times 10^{13}) = 47 \text{ bits}$
Case Study: Time based trigger

detect: process(rst, counter1, counter2) 
begin 
if(rst='1') then 
  trigger <= '0';
  elsif((counter1 > counter2+8) or (counter2 > counter1+8)) then 
    trigger <= '1';
  end if;
end if;
end process;
Taxonomy: Effects

- For triggered Trojans also called the "payload"
- Functional Changes must be triggered
  - Otherwise they are not stealthy
- Information leakage associated with cryptography
- Is it possible to make a triggered performance altering Trojan?
Case Study: Triggered Performance Degradation

- RO activates frequently burning the chip.
- Requires long trigger pulsewidth
  - Activation probability should still be low
  - Can use latch
Case Study: Key Leaking Trojan

- MOVX_A_ATDPTR implies the key is being moved from the acc.
- Requires just two 2:1 multiplexiers to
- Is this the activation rare enough?
  - Opcodes are easily manipulated
  - $2^{32}=4.3 \times 10^9$
  - $x \ 100\text{MHz} = 50\text{s}$
  - Assume instructions are 1-9 cycles

In FSM Controller:

In Memory Controller:

```plaintext
178  isKey <= '1' when s_command = MOVX_A_ATDPTR else '0';
179
180
181
172  JB <= s_ramx_data_in when isKey = '1' else "ZZZZZZZZ" when isKey='0';
173
174```
Taxonomy: Location

- Location refers to the part of the system
  - It does not refer to physical placement
- Not all Trojans will have a single or any location
- Location likely implies implies either
  - Activation mechanism
  - Effect

Diagram:
- Location
  - Processor
  - Memory
  - I/O
  - Power supply
  - Clock grid
Taxonomy: Physical Characteristics

- **Distribution**: is the Trojan spread out?
  - distributed Trojans will impact uniformly
- **Structure**
  - If the layout changes, detection is trivial
    - Trojans have an area constraint
  - Detection schemes assume unchanged