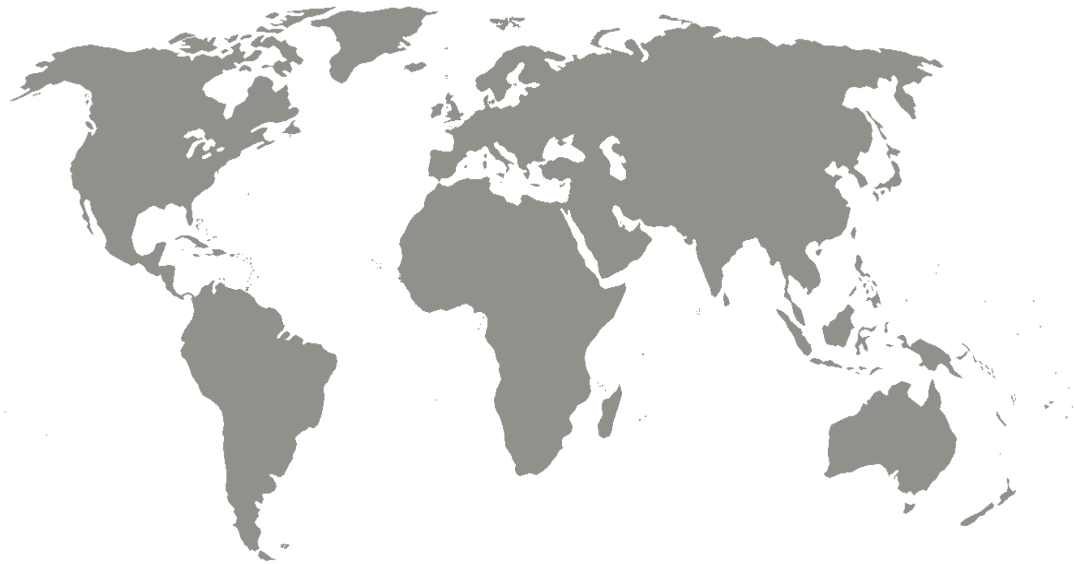
Introduction to the Hardware Trojan Problem

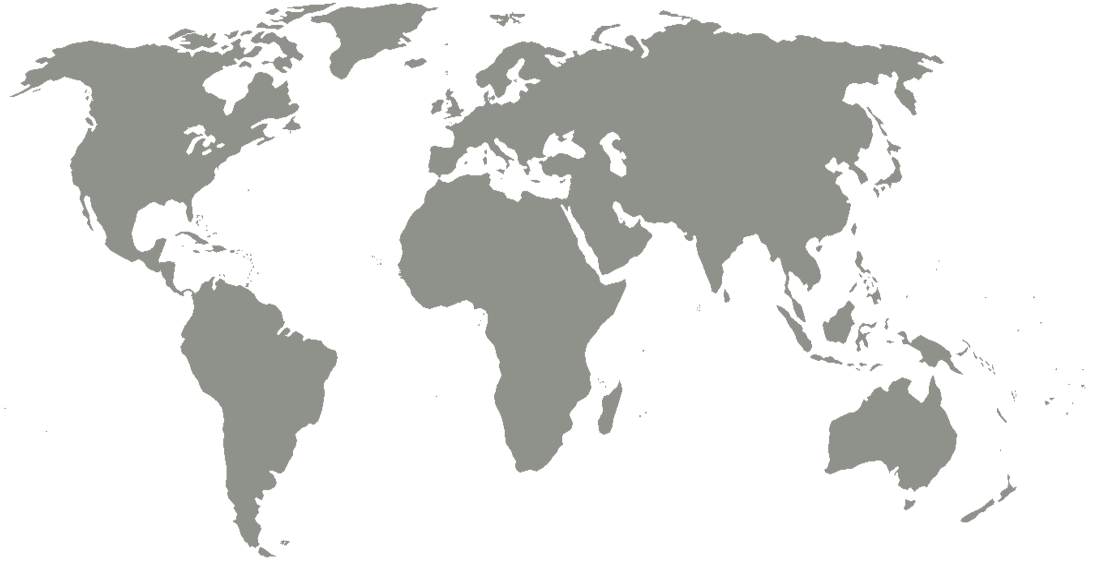
Globalization

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

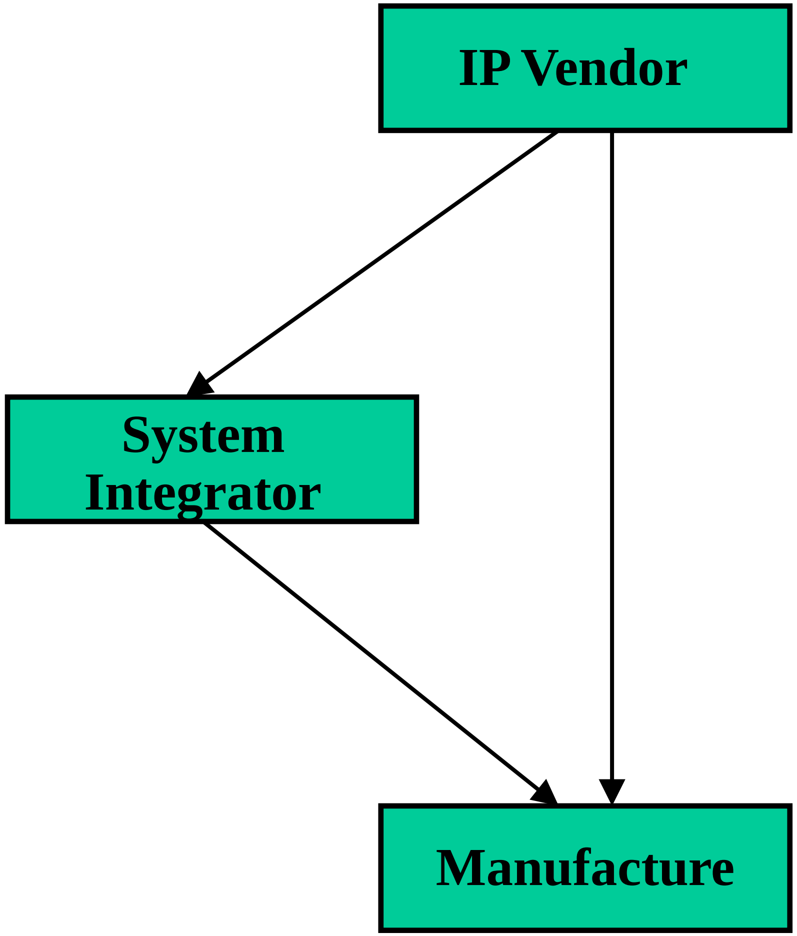


Globalization

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

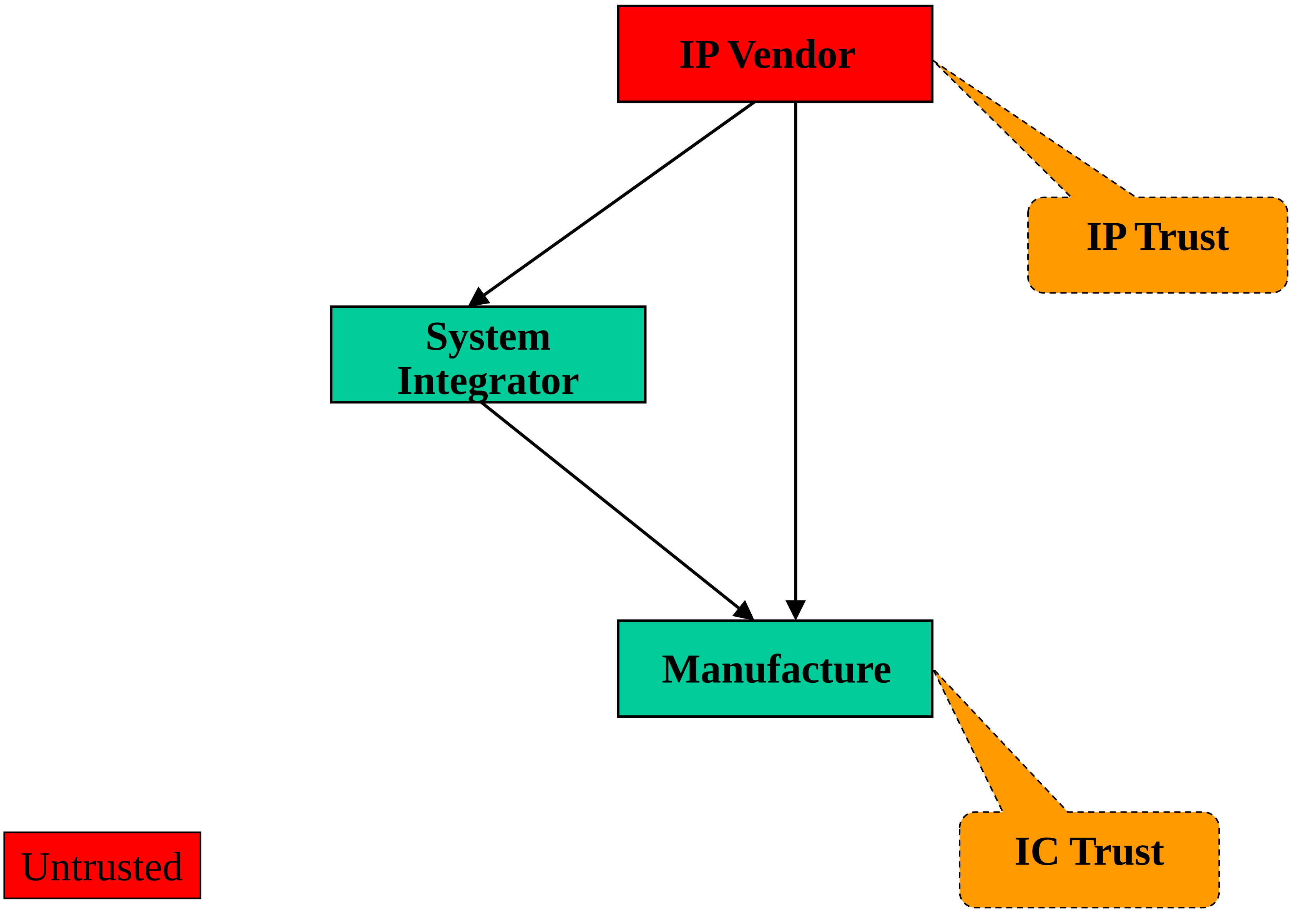


HW Threats



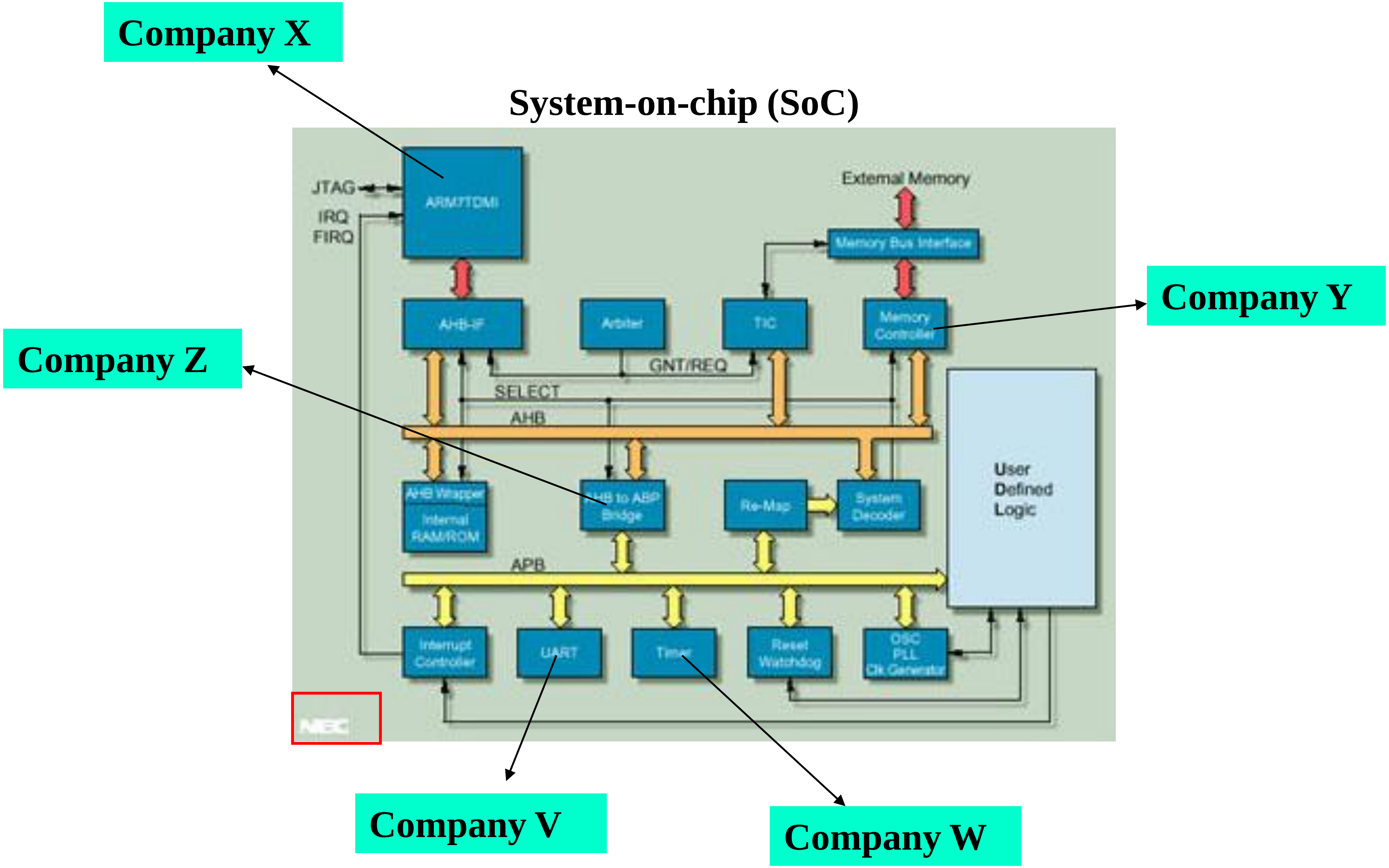
Any of these steps can be untrusted

HW Threats

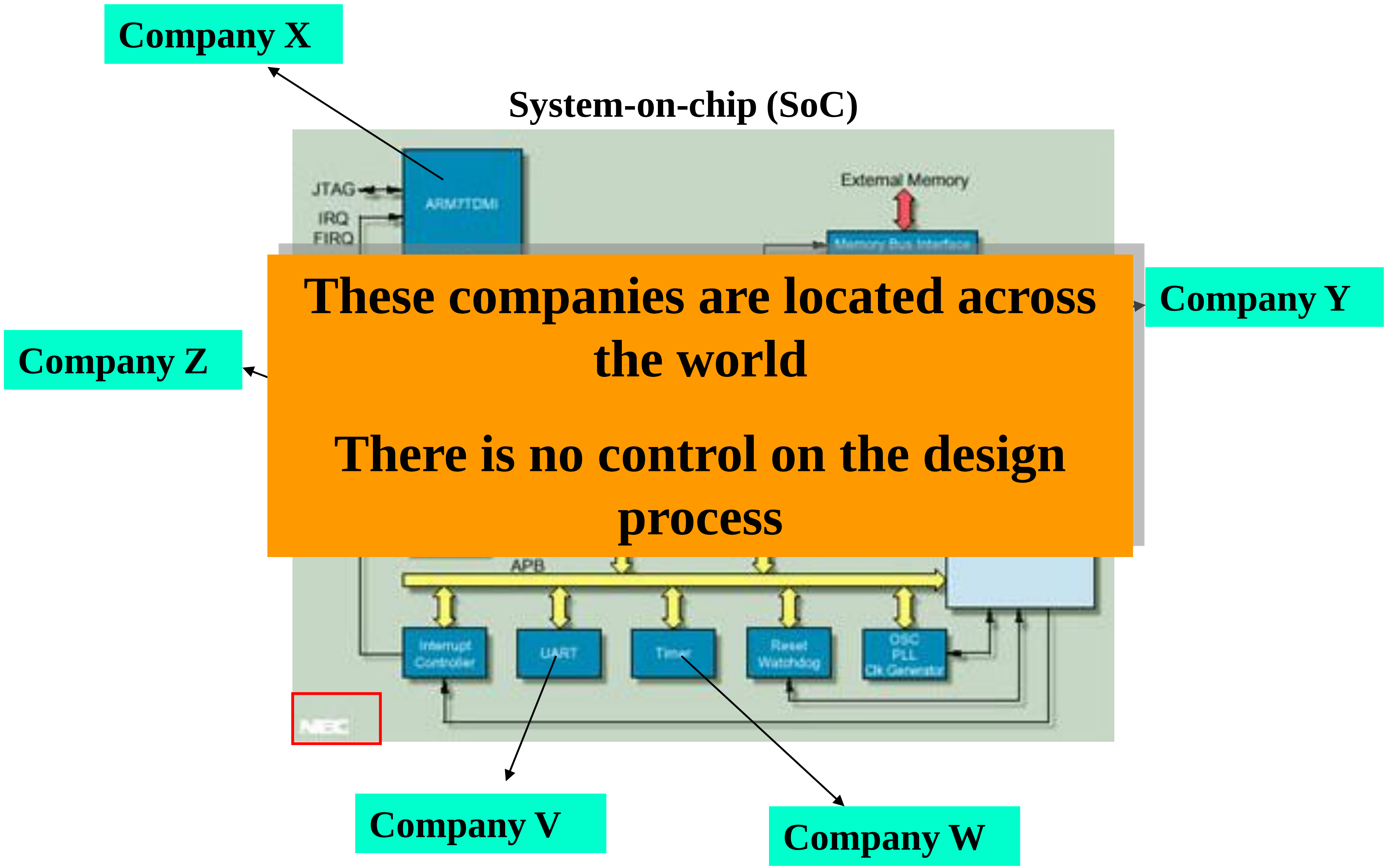


IP Vendor
IP Trust
System Integrator
Manufacture
IC Trust
Untrusted

Issues with Third IP Design



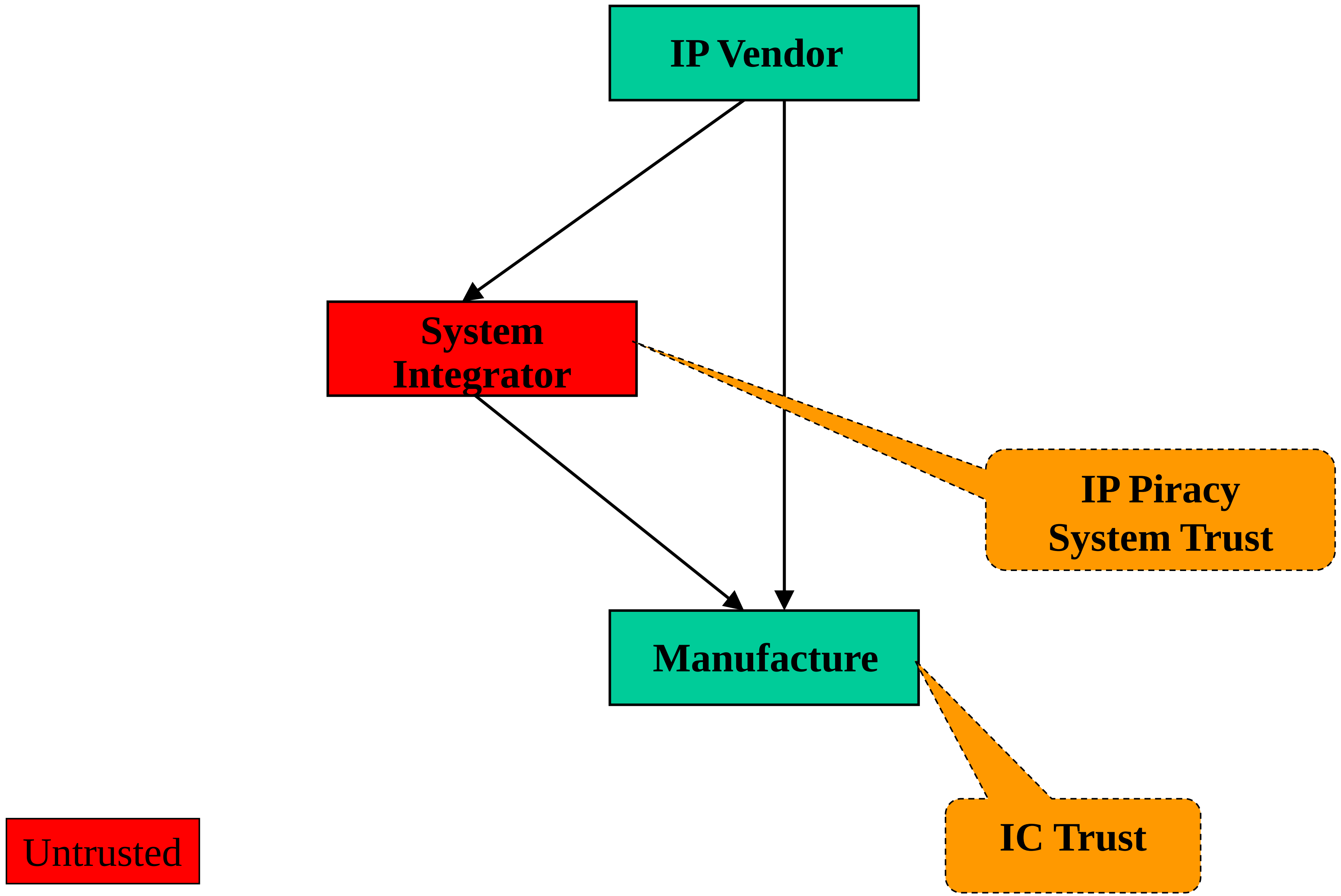
Issues with Third IP Design



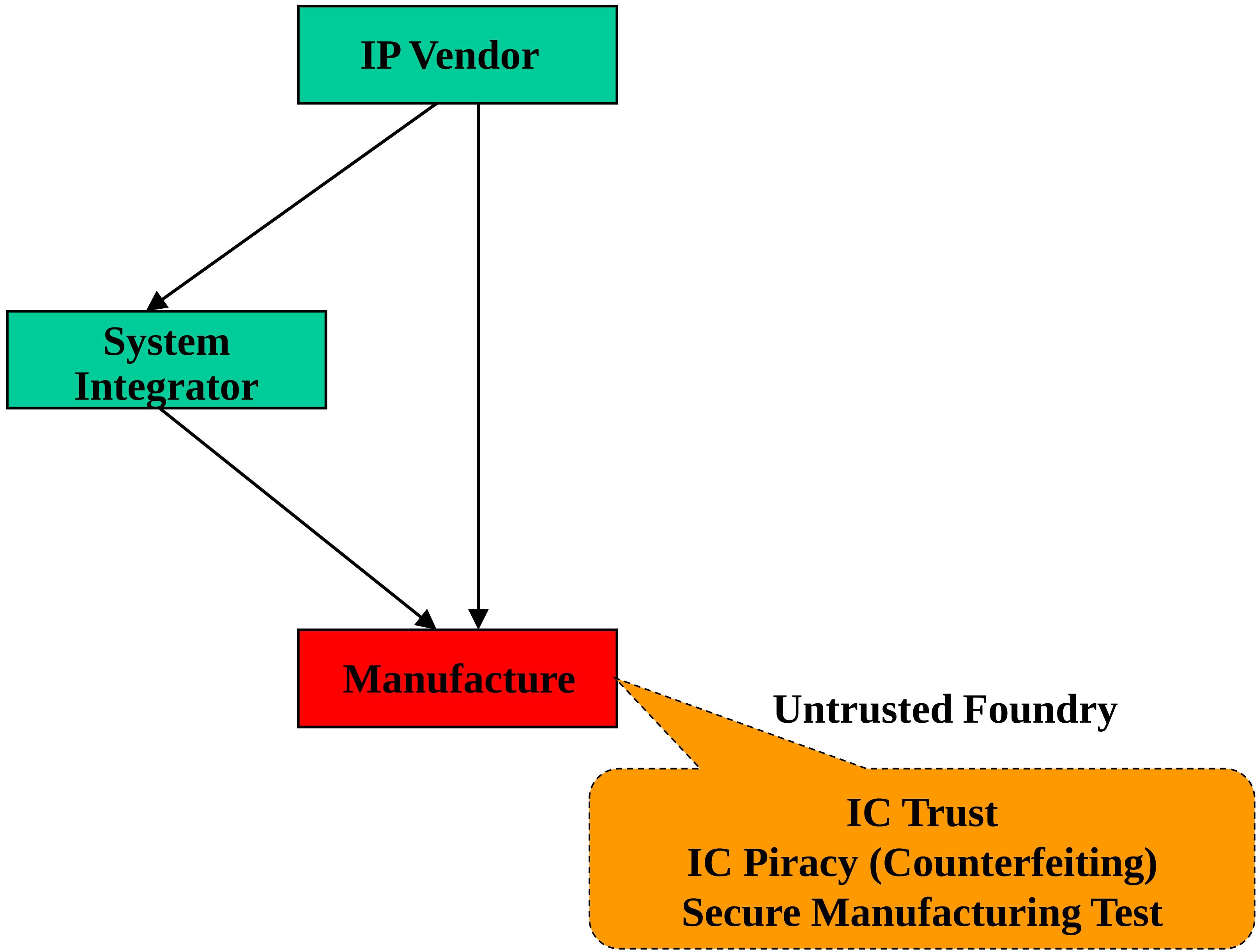
These companies are located across the world

There is no control on the design process

HW Threats


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

HW Threats



Untrusted

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 des ... easingly vulnerab ... ith globaliza**
- **Design a**
  - **A desig**
- **An adve**
  - **A Troj ... at some future**
  - **A Troj ... tion covertly to the**

U.S. Senate, 2003

Defense Science Board, 2005

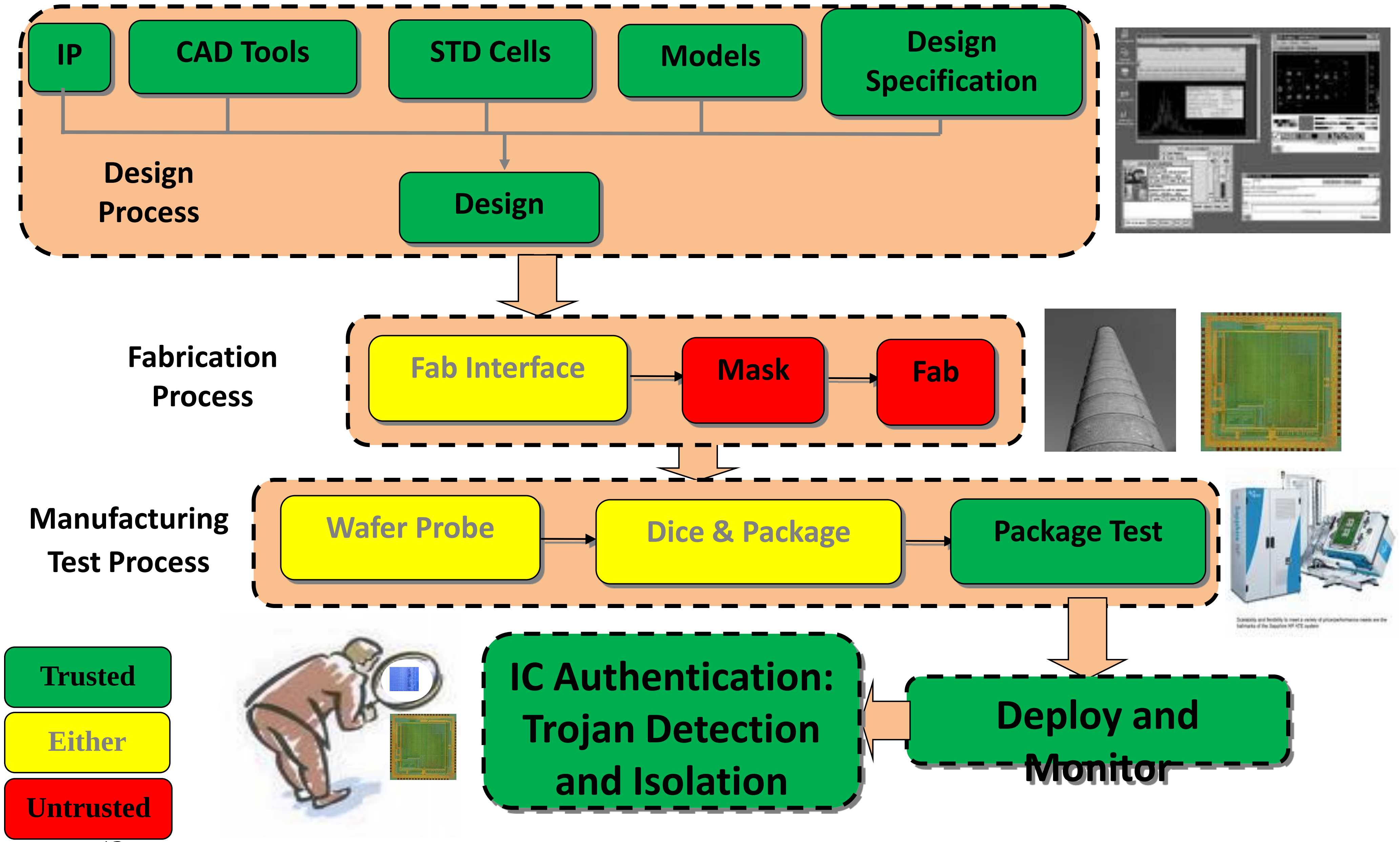
Semiconductor Equipment and Materials Industry (SEMI), 2008

IEEE Spectrum, 2008

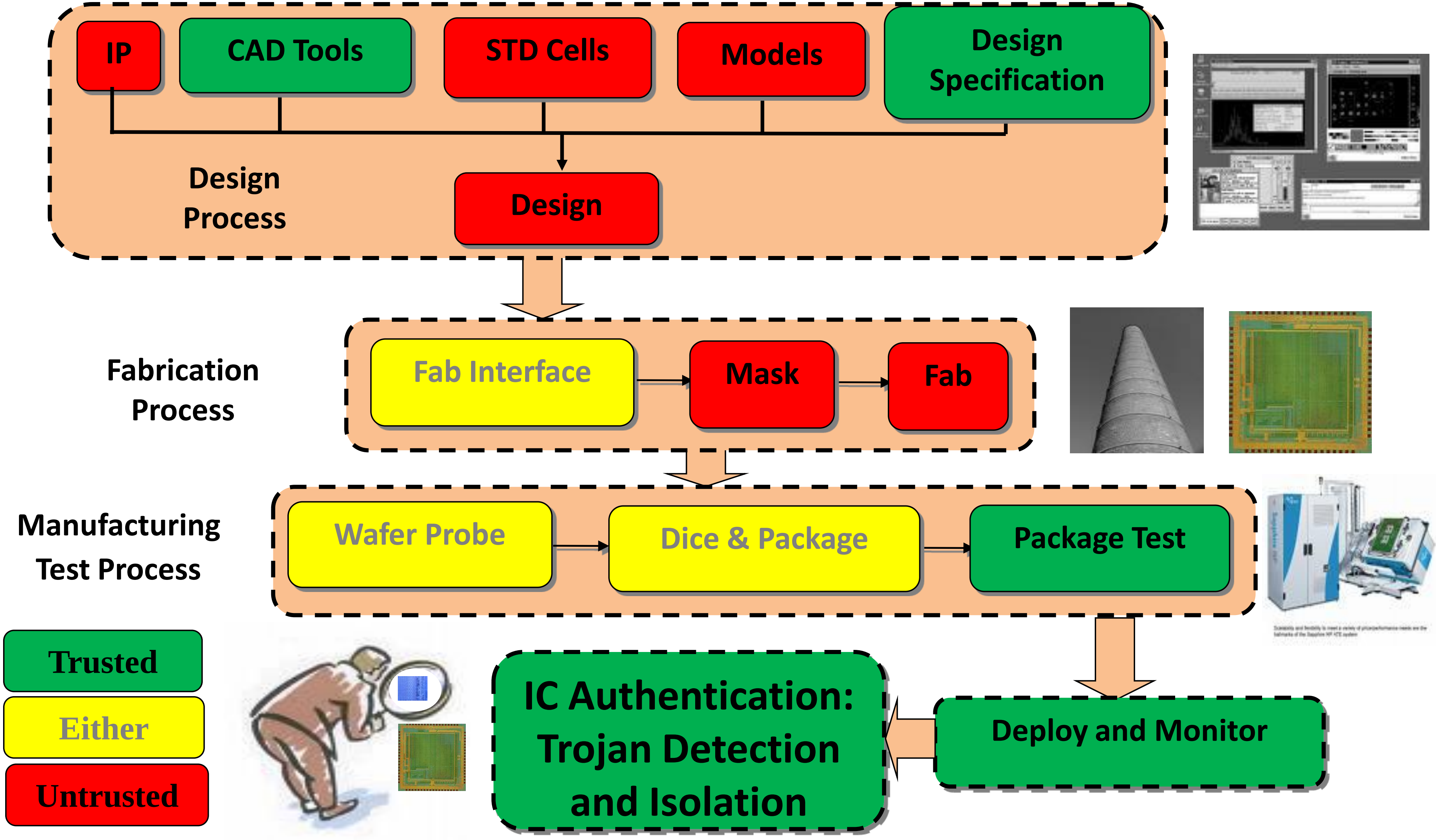
IEEE Symposium on Hardware-Oriented Security and Trust (HOST)

More articles have addressed this issue within the last few years

ASIC Design Process – Untrusted Foundry

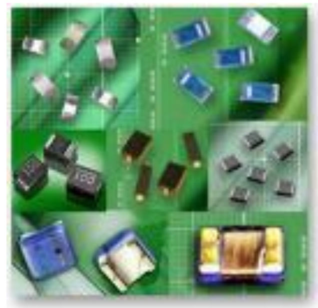


Untrusted Designer and Foundry

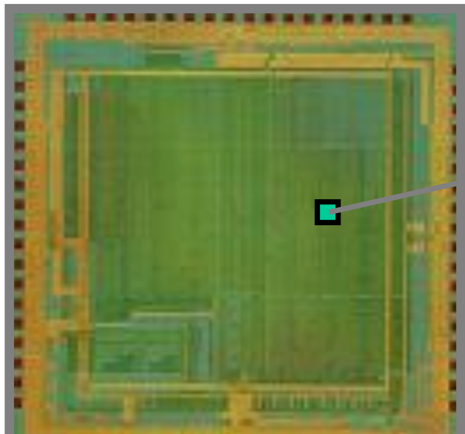


Applications and Threats

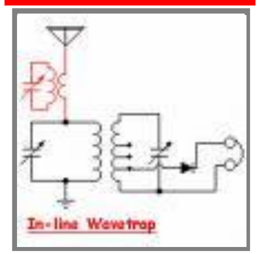
Thousands of chips are being fabricated in untrusted foundries



Hardware Trojan – Back Door



Antenna



Untrusted Hardware

- Adversary can send and receive secret information
- Adversary can disable the chip, blowup the chip, send wrong processing data, impact circuit information etc.

- Adversary can place an Antenna on the fabricated chip
- Such Trojan cannot be detected since it does not change the functionality of the circuit.

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

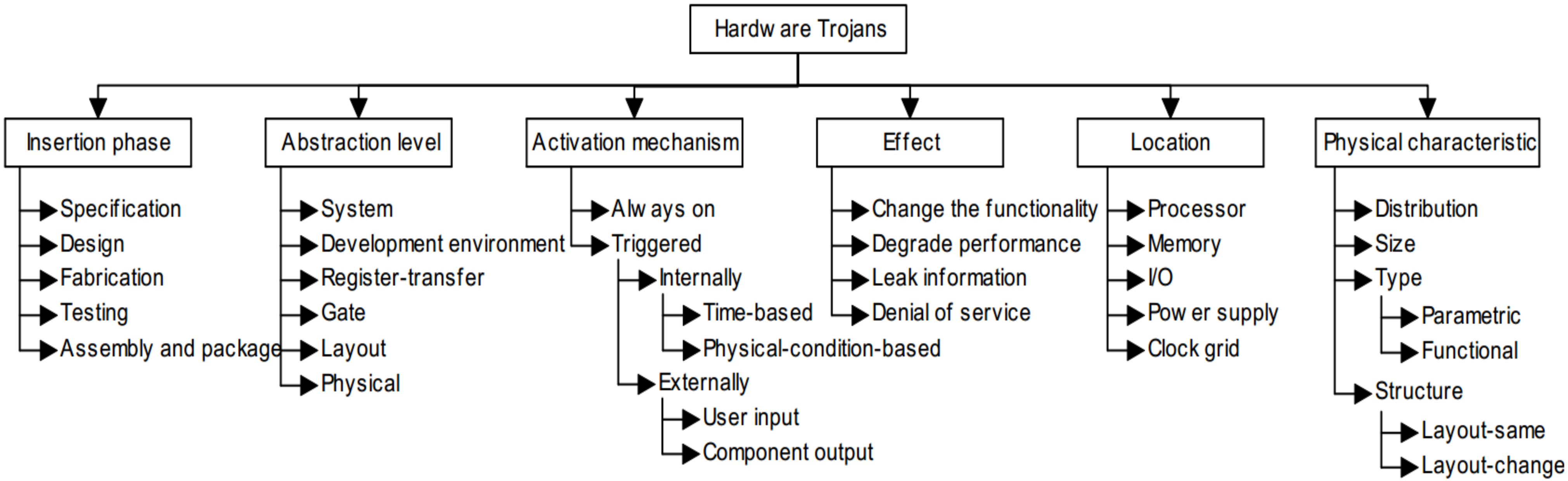
Defining the Problem

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**

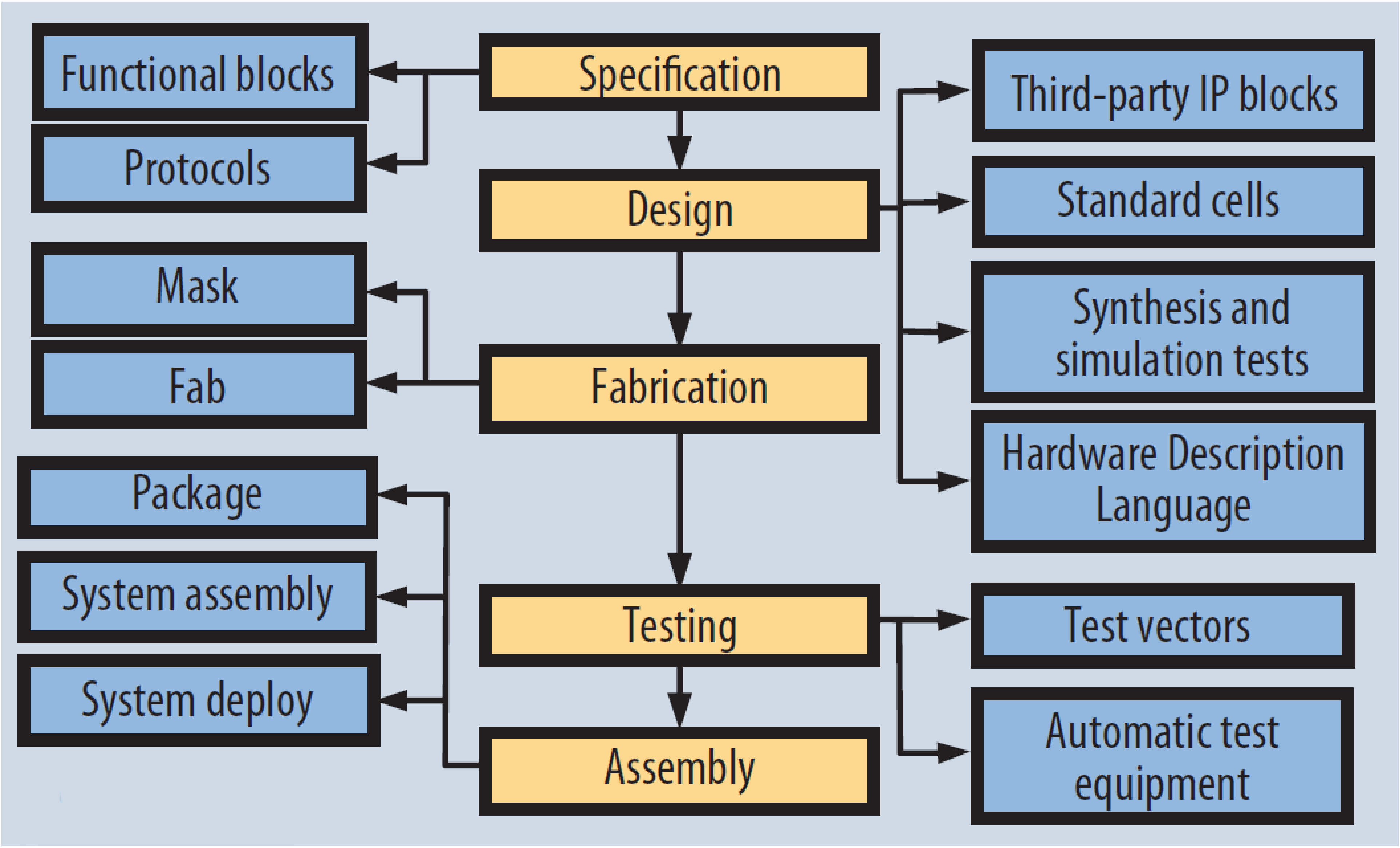
- **Software Trojans**
  - **A Trojan is part of the code in software**
  - **A Trojan behavior can change**
  - **A Trojan can be added to a software via network**
  - **Once identified, it can be removed and added to a database to look for it in the future**

Taxonomy

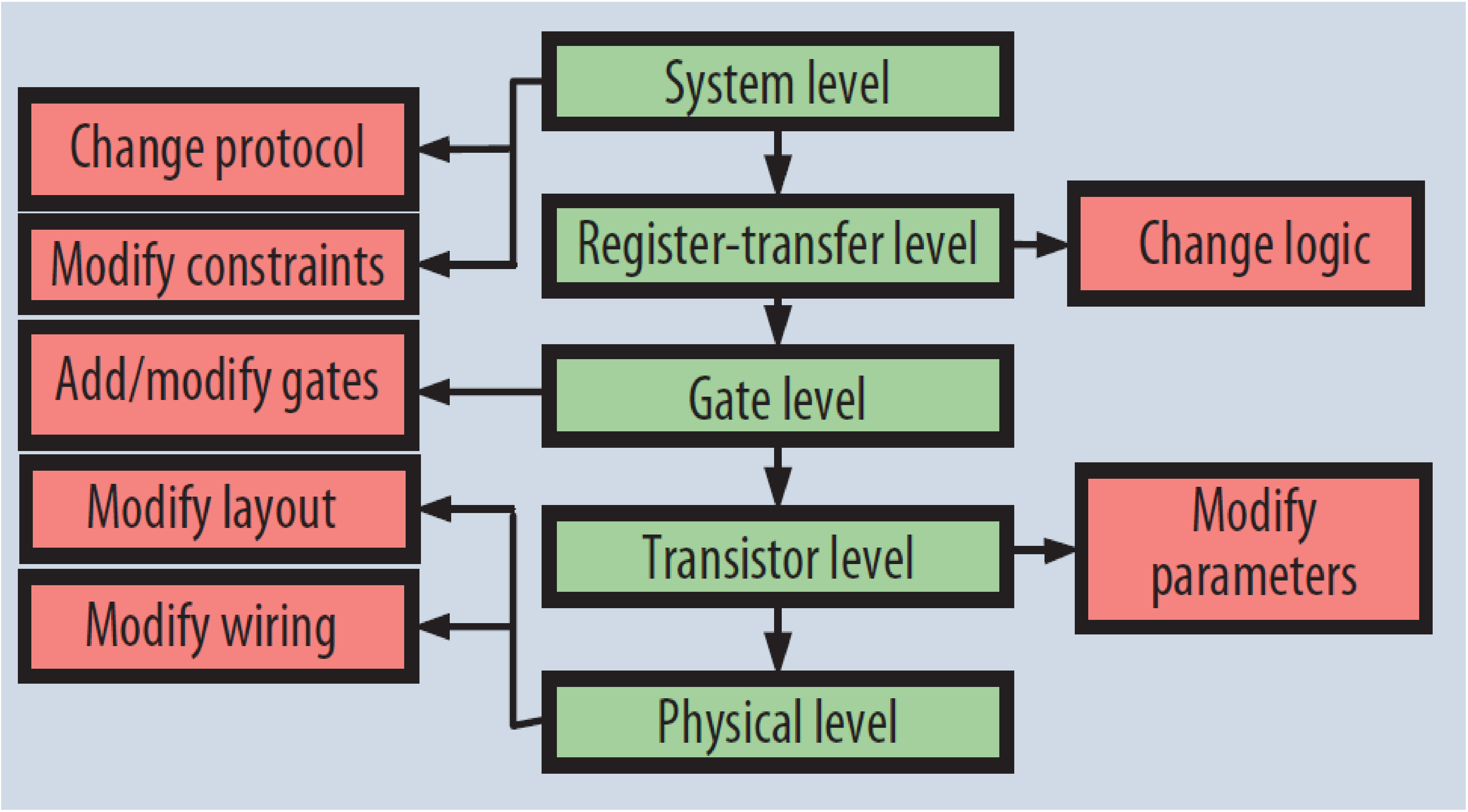


Karri, R.; Rajendran, J.; Rosenfeld, K.; Tehranipoor, M.; , "Trustworthy Hardware: Identifying and Classifying Hardware Trojans," *Computer* , vol.43, no.10, pp.39-46, Oct. 2010

Taxonomy: Insertion Phase



Taxonomy: Abstraction Level



Case Study: RTL Trojan

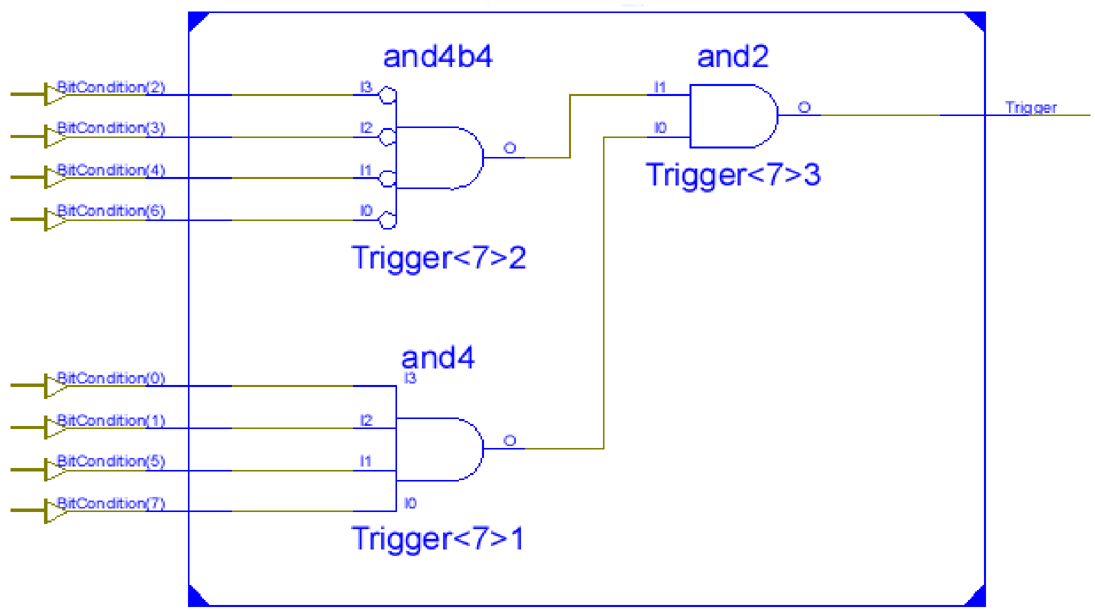
- Code segment of 8051 microprocessor in VHDL
- Trojan changes program counter behavior
  - Increment maps to accumulator jump
  - Behaves normally while inactive
- Cannot directly control number of gates used

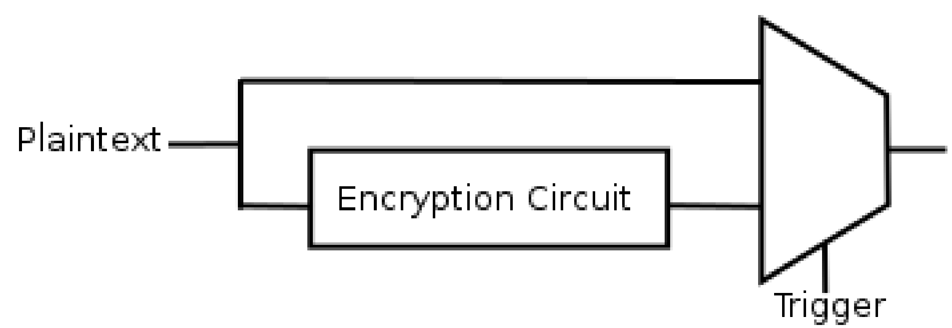
```
begin                          -- architecture structural

        -- This Trojan will perform a DoS attack with a single gate.
        -- Whenever the rare triggering condition is activated,
        --PC incrementations are maped to arbitrary jumps.
        troout <= s_pc_inc_en(3) &
                  (trigger or s_pc_inc_en(2)) &
                  s_pc_inc_en(1 downto 0);

```

```
    case s_pc_inc_en is
      when "0001" =>                       -- increment PC
        pc_comb <= pc_plus1;
      when "0010" =>                       -- for relativ jumps and calls
        pc_comb <= conv_unsigned(pc_plus1 + signed(rom_data_i),16);
      when "0011" =>                       -- load interrupt vectoradress
        pc_comb(15 downto 8) <= conv_unsigned(0,8);
        pc_comb(7 downto 0) <= s_help;
      when "0100" =>                       -- ACALL and AJMP
        pc_comb(15 downto 11) <= s_help16(15 downto 11);
        pc_comb(10 downto 8) <= s_ir(7 downto 5);
        pc_comb(7 downto 0) <= unsigned(rom_data_i);
      when "0101" =>                       -- JMP_A_DPTR, MOVC_A_ATDPTR
        pc_comb <= v_dptr + conv_unsigned(acc,8);
      when "0110" =>                       -- MOVC
        pc_comb <= s_help16;
      when "0111" =>                       -- LJMP, LCALL
        pc_comb(15 downto 8) <= s_help;
        pc_comb(7 downto 0) <= unsigned(rom_data_i);
      when "1000" =>                       -- RET, RETI
        pc_comb(15 downto 8) <= s_help;
        pc_comb(7 downto 0) <= s_reg_data;
      when "1001" =>                       -- MOVC_A_ATPC
        pc_comb <= pc_plus1 + conv_unsigned(acc,8);
      when others => pc_comb <= pc;
```

Case Study: Gate Level Trojan

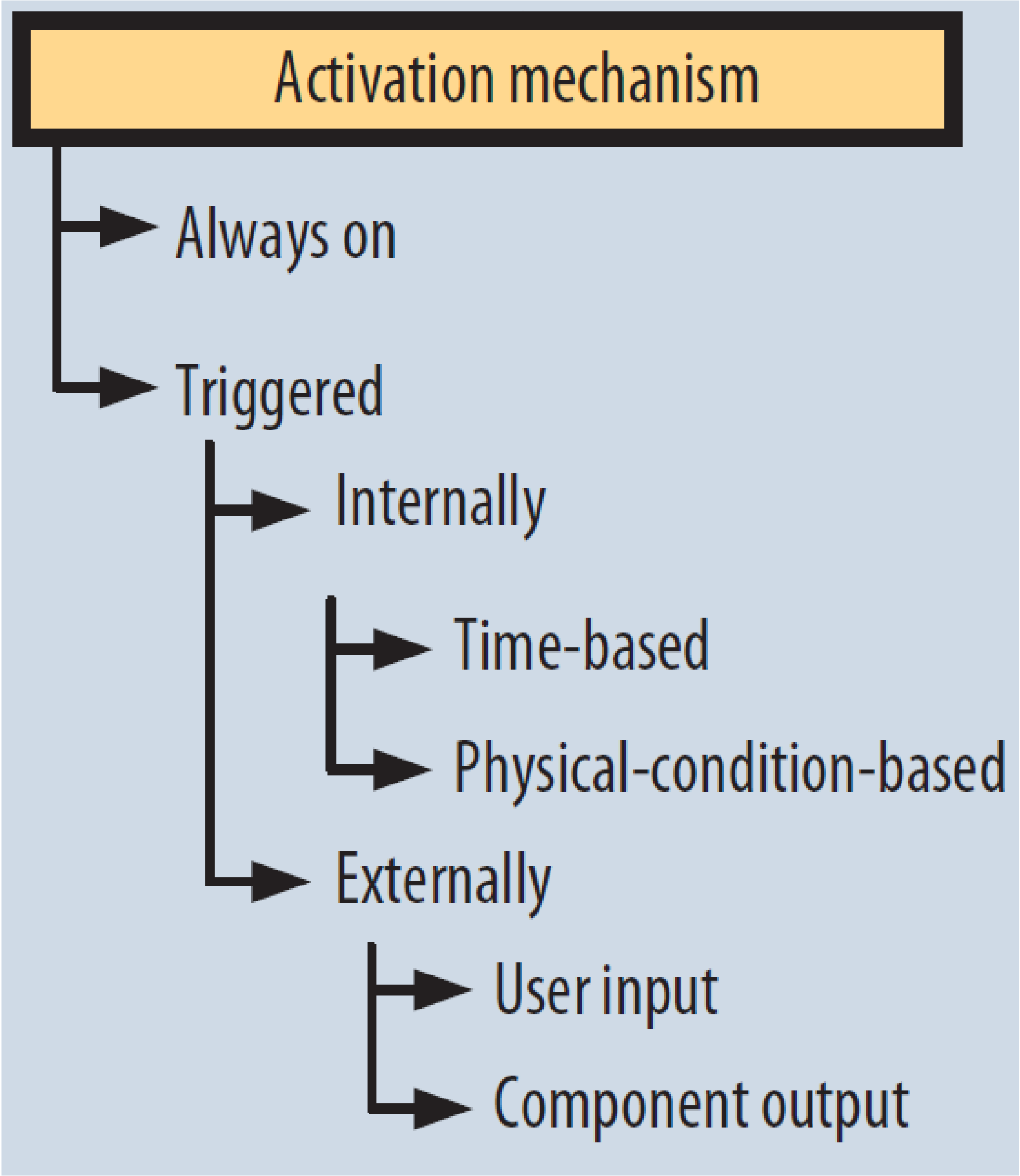
- Gate Level Trojan to attack cryptographic hardware
  - Trigger seeks "10100011"
  - On trigger, encryption is skipped
- Particular gates used can be controlled
  - Location cannot
- Practical GL Trojans are in netlist form





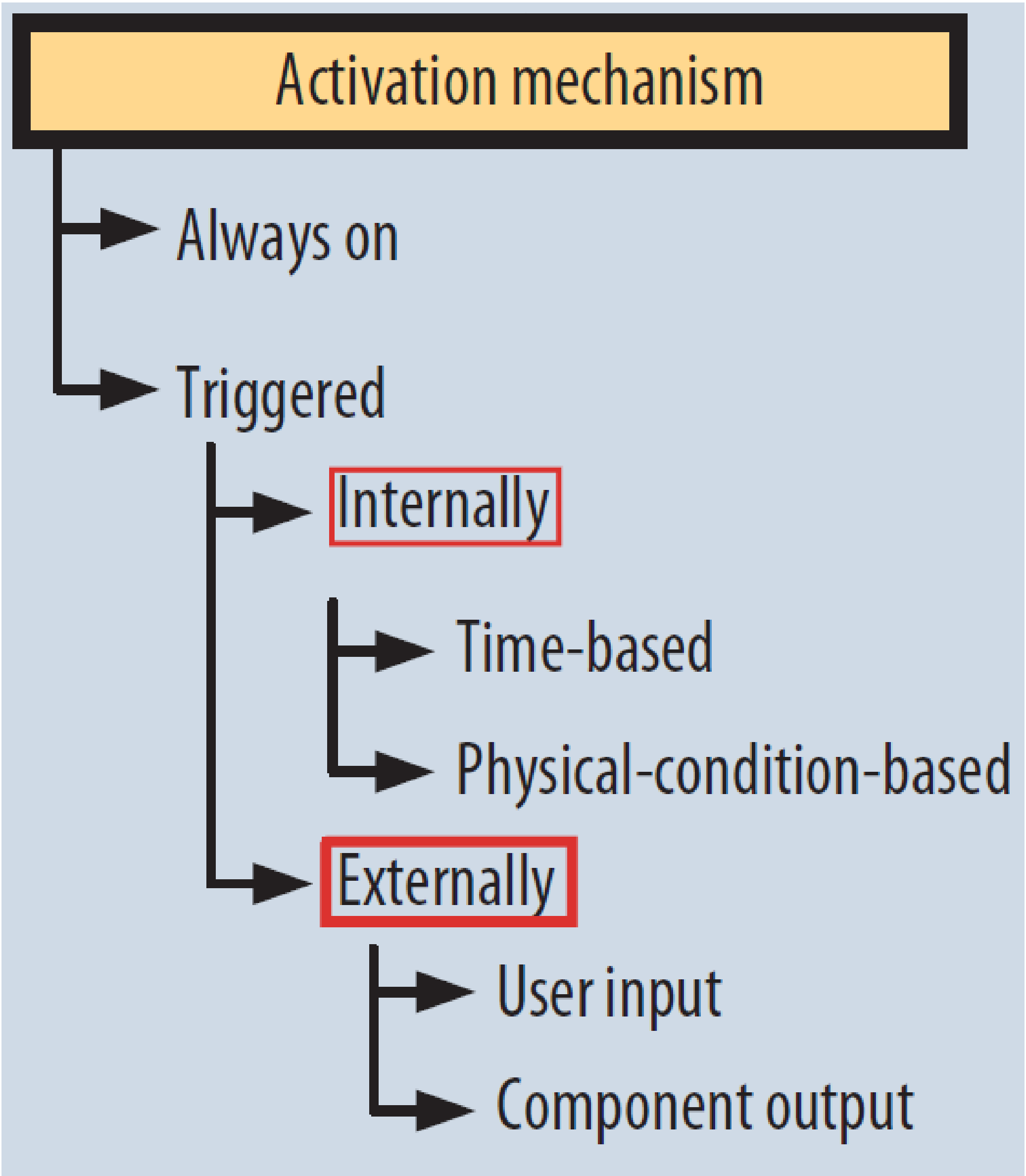
Taxonomy: Activation Mechanism

- 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

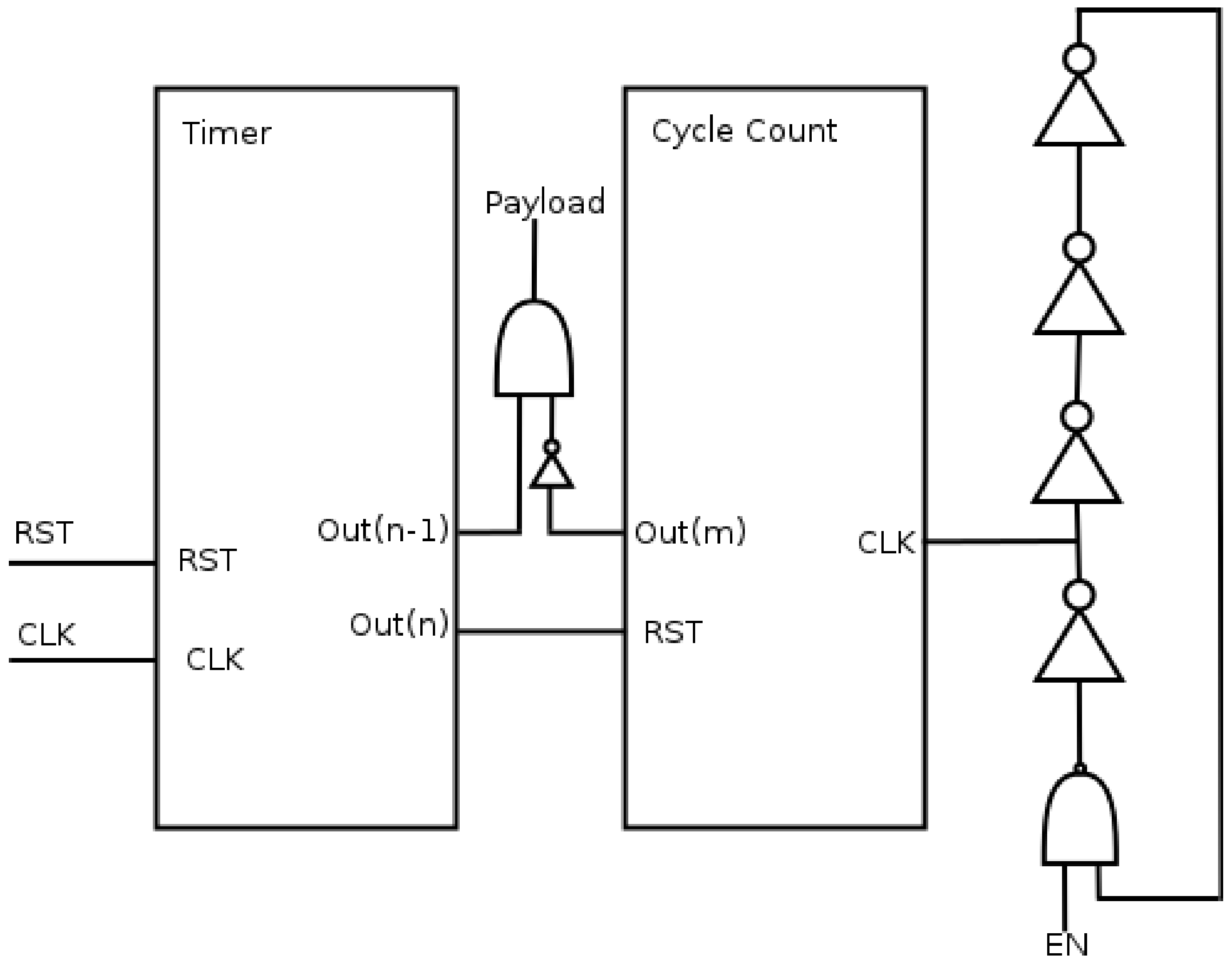


Internal vs. External

- Externally Triggered
  - Depends directly on external inputs
  - Can be both user and component driven
  - e.g. transmitter
- Internal
  - Can also include internal signals



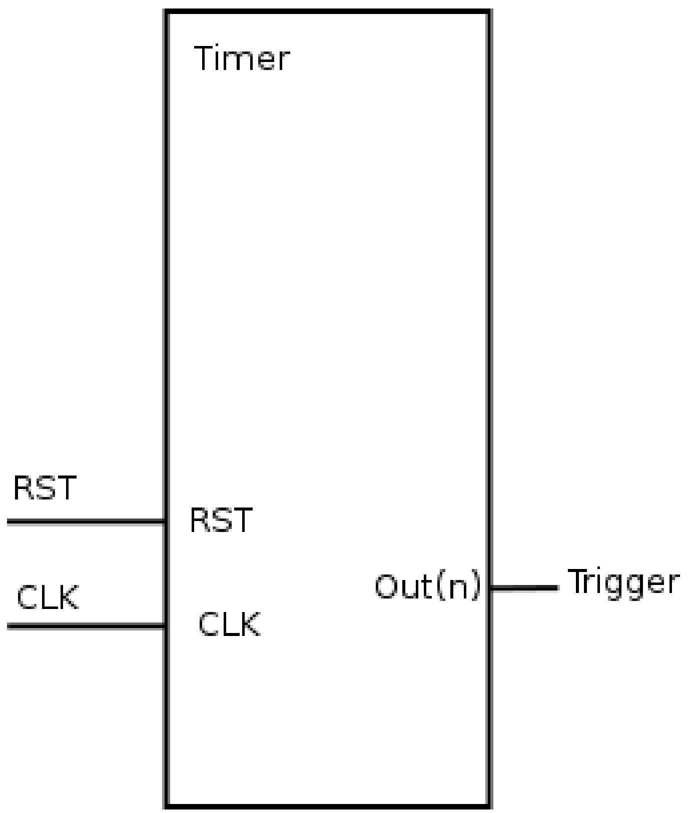
Case Study: Physical Condition



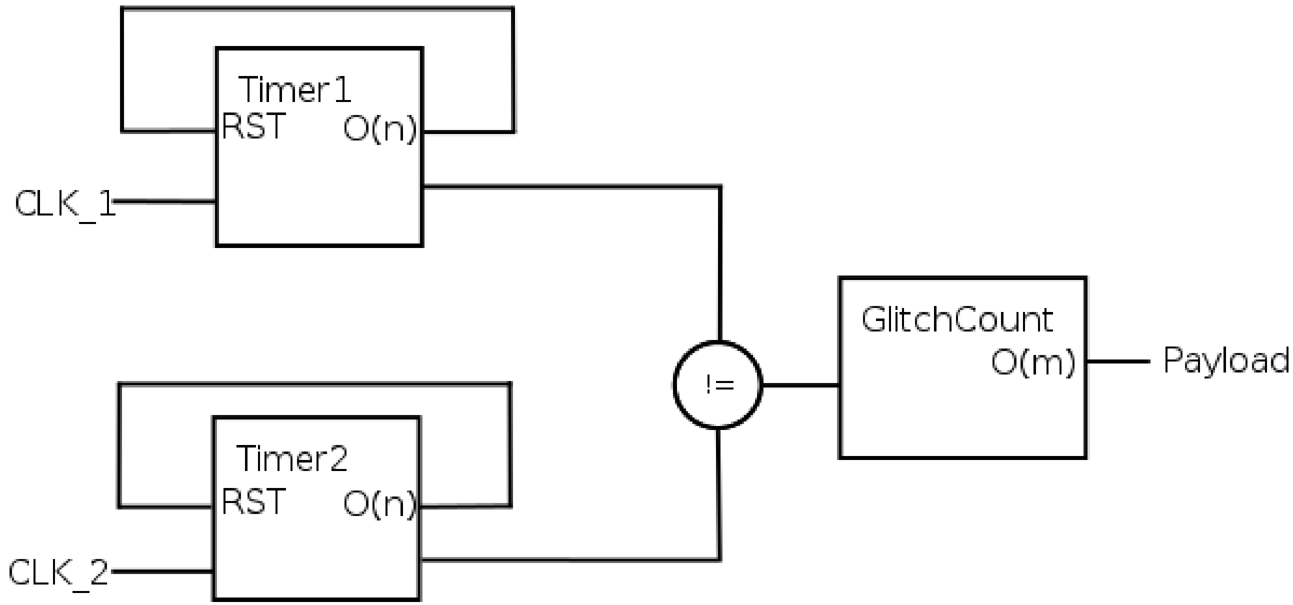
Case Study: Time Bomb Trigger

- Subclass of time-based
  - Called "time bomb"
- Weaknesses
  - What if chip tester waits long enough?
  - Increasing time increases area
    - O(log2(n))

Example:
1GHz * 1 day = 8×10^{13}
log2(8×10^{13}) = 47 bits



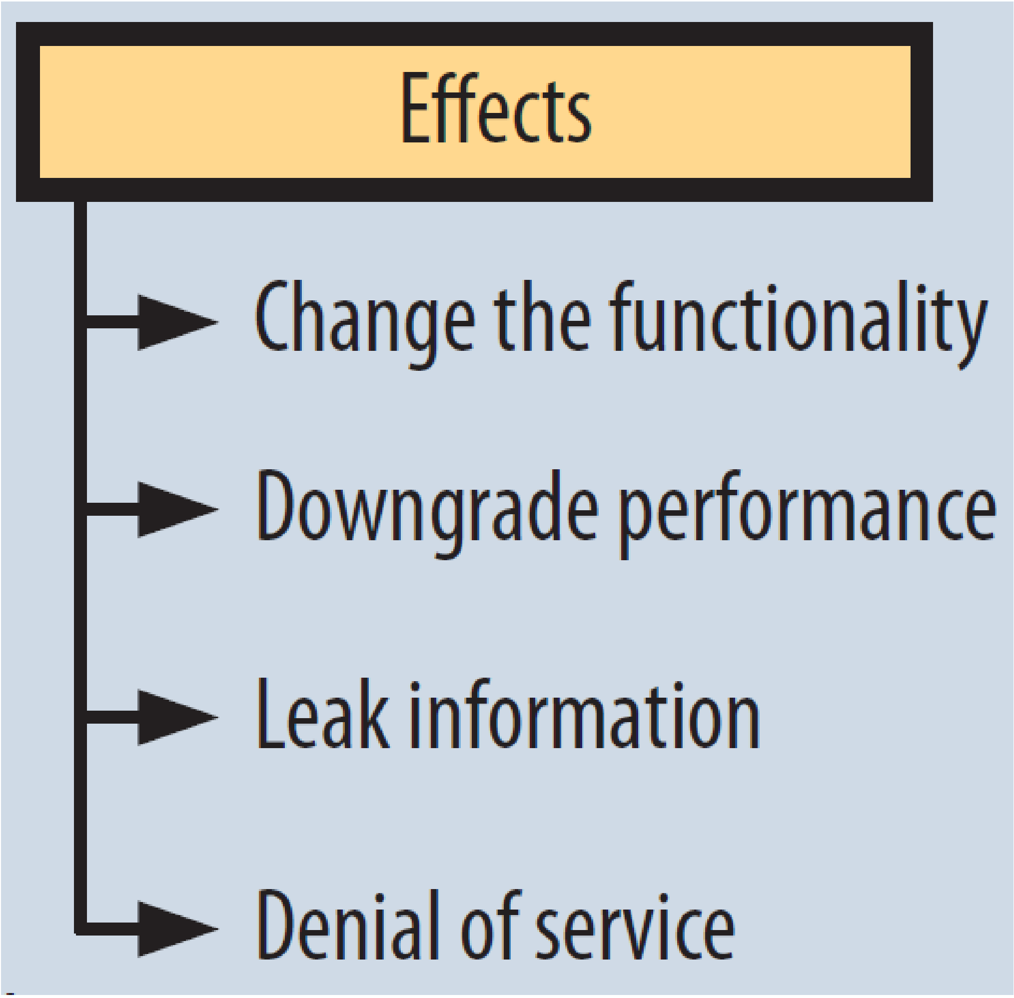
Case Study: Time based trigger




```
        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 process;
```

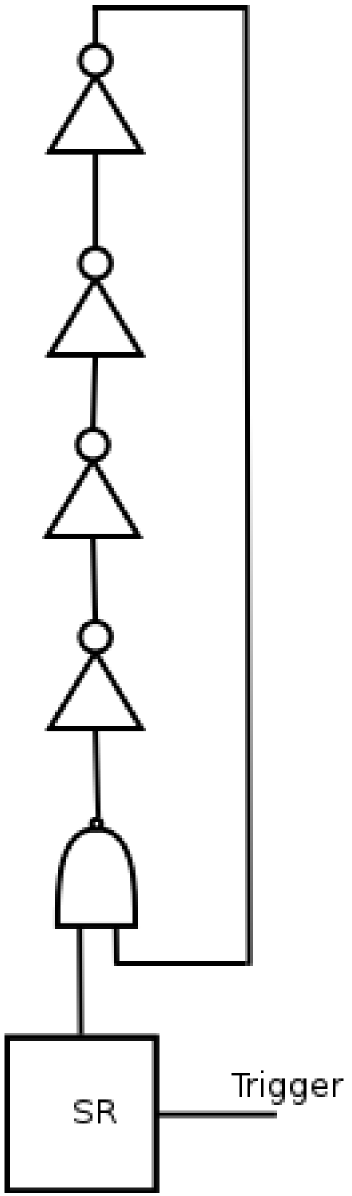
Taxonomy: Effects

- 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 x 10^9
  - x 100MHz = 50s
  - Assume instructions are 1-9 cycles

In FSM Controller:

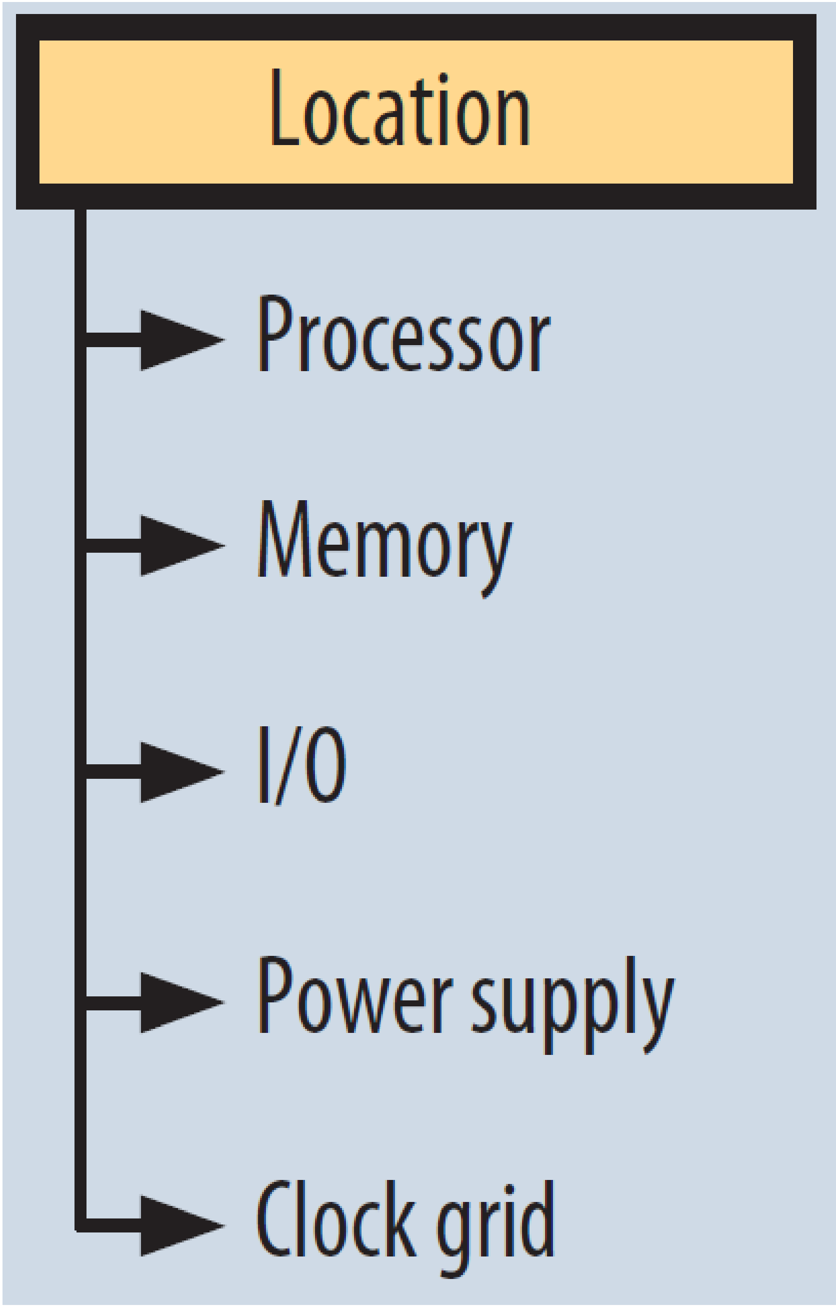
```
isKey <=
'1' when s_command = MOVX_A_ATDPTR else
'0';
```

In Memory Controller:

```
        JB <= s_ramx_data_in when isKey ='1' else
            "ZZZZZZZZ" when isKey='0;
```

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



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