Trojan Scanner

Navid Asadi

Physical Inspection and Attacks on Electronics (PHIKS)
Hardware Trojan as a Threat

- **Hardware Trojan:**
  - Malicious addition, deletion or modification to existing circuit elements.

- **What Hardware Trojans can do?**
  - Reduce the reliability to cause early failure
  - Hijack to control or change the functionality
  - Leak sensitive information (Encryption keys)

- **Targeted Applications**
  - IoT devices (Home automation – Google Home, Alexa, Security cams, locks)
  - Aerospace & Military applications
  - Civilian applications like Aviation, Security, Healthcare, Financial…and many more
The economics of the semiconductor industry today have created a ‘horizontal’ business model.

Cost of maintaining top-end fab prohibitively expensive ~ $$ Billions.

Foundries at advanced nodes are almost exclusively off-shore today.
Trust Issues in Design, Fabrication, etc.

- **Trust Issues**
  - Foundry receives (almost) everything from design house
    - GDSII layout ⇒ Netlist, Test Vectors
  - A design house has little to no control over an off-shore foundry.

- **Threats**
  - IC Threats
    - Overproduction
    - Trojan Insertion
  - IP Threats
  - Out-of-Spec/Defective Products
• **Trust Issues**
  - Foundry receives (almost) everything from design house
    - GDSII layout ⇒ Netlist, Test Vectors
  - A design house has little to no control over an off-shore foundry.

• **Threats**
  - IC Threats
    - Overproduction
    - Trojan Insertion
  - IP Threats
  - Out-of-Spec/Defective Products
Taxonomy of Hardware Trojans

Trojan Classification

- Physical Characteristics
- Activation Characteristics
- Action Characteristics

Type
- Functional
- Parametric

Size
- Big
- Small

Distribution
- Tight
- Loose

Characteristics
- Internally Triggered
- Externally Triggered
- Modify Function
- Modify Specification
- Transmit Info

All Rights Reserved
Trojan Detection Techniques

Trojan Detection Approaches

Destructive

- Full chip Reverse Engineering

Run -Time Monitoring

- Resource Utilization
- On Chip Sensors
- Reconfigurable Computing

Non - Destructive

Test Time

- Logic Test
- Side Channel Analysis

Timing & Delay

- Quiescent Current
- Transient Current
- EM Radiation
- Multiple Parameters
Trojan Detection Techniques

Trojan Detection Approaches

Destructive

- Full chip Reverse Engineering
  - Time Consuming
  - Most Effective

- Run –Time Monitoring
  - Resource Utilization
  - On Chip Sensors
  - Reconfigurable Computing

Non - Destructive

- Test Time
  - Logic Test
  - Side Channel Analysis

- Timing & Delay
  - Quiescent Current
  - Transient Current
  - EM Radiation
  - Multiple Parameters

Non - Destructive

- Reconfigurable Computing
- Logic Test
- Side Channel Analysis
- Resource Utilization
- On Chip Sensors
- Timing & Delay
- Quiescent Current
- Transient Current
- EM Radiation
- Multiple Parameters

All Rights Reserved
Trojan Detection Approaches

Destructive
- Full chip Reverse Engineering

Non-Destructive
- Timing & Delay
  - Quiescent Current
  - Transient Current
  - EM Radiation
  - Multiple Parameters

Run-Time Monitoring
- Resource Utilization
- On Chip Sensors
- Reconfigurable Computing

Test Time
- Logic Test
- Side Channel Analysis

Consumes extra resources
- CPU
- Power
- Memory
- Silicon area
Trojan Detection Techniques

Trojan Detection Approaches

Destructive
- Full chip Reverse Engineering

Non-Destructive
- Test Time
  - Logic Test
  - Side Channel Analysis
- Run-Time Monitoring
  - Resource Utilization
  - On-Chip Sensors
  - Reconfigurable Computing
- Timing & Delay
  - Quiescent Current
  - Transient Current
  - EM Radiation
  - Multiple Parameters

- Highly sensitive to noise and process variation
- Detects only small Trojans
Next Generation Trojans - Challenges

# of Trojan gates vs. IC gates

Testing 64 input adder at 100 MHz needs $2^{65} \sim 10^{18}$ input combination and it may take $10^{10}$ s ~ 317 years.

Long detection time

Difficulty of Triggering HT

We need a super fast and reliable HT detection technique

Stealthy Trojans
## Hardware Trojans and their Footprints

<table>
<thead>
<tr>
<th>Trojan Type</th>
<th>Implemented by</th>
<th>Payload</th>
</tr>
</thead>
<tbody>
<tr>
<td>AES-T100</td>
<td>Flip Flops &amp; XOR</td>
<td>Leaking LC circuit</td>
</tr>
<tr>
<td>AES-T400</td>
<td>Modulating unused pin on chip</td>
<td>Transmitting key bits</td>
</tr>
<tr>
<td>AES-T1800</td>
<td>Shift Register</td>
<td>Increased power</td>
</tr>
<tr>
<td>AES-T600</td>
<td>Shift Register &amp; Two Inverters</td>
<td>Leakage current</td>
</tr>
<tr>
<td>B15-T100</td>
<td>6 Logic Cells Inserted</td>
<td>Reducing clock frequency</td>
</tr>
<tr>
<td>B19-T300</td>
<td>Counter Circuit</td>
<td>Manipulation of address bus</td>
</tr>
<tr>
<td>Basic RSA-T200</td>
<td>Disable encoding on RTL level</td>
<td>Denial of service</td>
</tr>
<tr>
<td>RS232-T1800</td>
<td>Chain of Invertors</td>
<td>No Info in benchmark</td>
</tr>
<tr>
<td>EthernetMAC10GE-T100</td>
<td>Critical path is widened / narrowed</td>
<td>Reliability Impact</td>
</tr>
<tr>
<td>EthernetMAC10GE-T200</td>
<td>Part of clock tree is widened</td>
<td>Reliability Impact</td>
</tr>
<tr>
<td>EthernetMAC10GE-T300</td>
<td>Part of clock tree is narrowed</td>
<td>Reliability Impact</td>
</tr>
<tr>
<td>EthernetMAC10GE-T400</td>
<td>Narrowing power lines</td>
<td>Reliability Impact</td>
</tr>
<tr>
<td>EthernetMAC10GE-T500</td>
<td>Narrowing ground lines</td>
<td>Reliability Impact</td>
</tr>
<tr>
<td>EthernetMAC10GE-T600</td>
<td>Making design susceptible to crosstalk</td>
<td>Denial of service</td>
</tr>
</tbody>
</table>

In an IC, all Hardware Trojans leave their footprints on either Active or Metal Layer!!

Source: Unique hardware Trojans from the list of 94 Trojans reported at TrustHub
# SEM Imaging Time Table

Technology node: 130nm; Chip size: 1.5mm x 1.5 mm

<table>
<thead>
<tr>
<th>Scanning Speed &amp; Resolution</th>
<th>1500um x 1500um</th>
<th>500um x 500um</th>
<th>100um x 100um</th>
<th>20um x 20um</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 (1.0 µs/Pixel)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(512 x 512)</td>
<td>1 sec</td>
<td>9 sec</td>
<td>3 min 45 sec</td>
<td>1 hr 33 min 45 sec</td>
</tr>
<tr>
<td>(1024 x 1024)</td>
<td>2 sec</td>
<td>18 sec</td>
<td>7 min 30 sec</td>
<td>3 hr 7 min 30 sec</td>
</tr>
<tr>
<td>(2048 x 2048)</td>
<td>6 sec</td>
<td>54 sec</td>
<td>22 min 30 sec</td>
<td>9 hr 22 min 30 sec</td>
</tr>
<tr>
<td>4 (3.2 µs/Pixel)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(512 x 512)</td>
<td>1 sec</td>
<td>9 sec</td>
<td>3 min 45 sec</td>
<td>1 hr 33 min 45 sec</td>
</tr>
<tr>
<td>(1024 x 1024)</td>
<td>4 sec</td>
<td>36 sec</td>
<td>15 min 10 sec</td>
<td>6 hr 13 min 10 sec</td>
</tr>
<tr>
<td>(2048 x 2048)</td>
<td>14 sec</td>
<td>2 min 5 sec</td>
<td>52 min 5 sec</td>
<td>21 hr 42 min 5 sec</td>
</tr>
<tr>
<td>5 (10.0 µs/Pixel)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(512 x 512)</td>
<td>5 sec</td>
<td>45 sec</td>
<td>18 min 45 sec</td>
<td>7 hr 48 min 45 sec</td>
</tr>
<tr>
<td>(1024 x 1024)</td>
<td>22 sec</td>
<td>3 min 18 sec</td>
<td>1 hr 22 min 30 sec</td>
<td>1 d 10 hr 22 min 30 sec</td>
</tr>
<tr>
<td>(2048 x 2048)</td>
<td>1 min 25 sec</td>
<td>6 min 25 sec</td>
<td>5 hr 18 min 45 sec</td>
<td>5 d 12 hr 48 min 45 sec</td>
</tr>
<tr>
<td>6 (32 µs/Pixel)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(512 x 512)</td>
<td>11 sec</td>
<td>1 min 30 sec</td>
<td>36 min</td>
<td>15 hr</td>
</tr>
<tr>
<td>(1024 x 1024)</td>
<td>43 sec</td>
<td>6 min 30 sec</td>
<td>2 hr 45 min</td>
<td>1 d 21 hr 5 min</td>
</tr>
<tr>
<td>(2048 x 2048)</td>
<td>2 min 52 sec</td>
<td>24 min</td>
<td>10 hr 45 min 10 sec</td>
<td>11 day 1 hr 30 min</td>
</tr>
<tr>
<td>7 (100.0 µs/Pixel)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(512 x 512)</td>
<td>32 sec</td>
<td>4 min 48 sec</td>
<td>2 hr</td>
<td>2 days 2 hours</td>
</tr>
<tr>
<td>(1024 x 1024)</td>
<td>2 min 6 sec</td>
<td>18 min 54 sec</td>
<td>7 hr 52 min 30 sec</td>
<td>8 d 4 hr 25 min 6 sec</td>
</tr>
<tr>
<td>(2048 x 2048)</td>
<td>7 min 54 sec</td>
<td>1 hr 11 min 6 sec</td>
<td>1 d 5 hr 37 min 30 sec</td>
<td>30 d 20 hr 37 min 30 sec</td>
</tr>
</tbody>
</table>
High Voltage

Field of View (Magnification)

Dwelling Time

Resolution

5 kV 10 kV 15 kV

500 µm 100 µm 20 µm

3.2 µs /pixel 10 µs /pixel 32 µs /pixel

512 x 512 1024 x 1024 2048 x 2048
Trojan Scanner: Golden Chip

Back Side Thinned ICs
- Golden IC
- IC under Authentication (IUA)

SEM Imagining
- SEM Parameters
- Image Stitching (Golden IC & IUA)

Image Processing
- Image Enhancement
- Image Comparison

Trojan Detection
- Suspected Areas Detected on Trojan Map

Images:
- (a) Back Side Thinned ICs
- (b) SEM Imagining
- (c) Image Processing
- (d) Trojan Detection
- (e) Match vs. Mismatch
## Case Study of Trojan & Footprints on ICs

<table>
<thead>
<tr>
<th>Size of Change</th>
<th>Change Type</th>
<th>Footprint</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Smallest</strong></td>
<td><strong>Modification</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NAND $\leftrightarrow$ NOR</td>
<td>Active Region</td>
</tr>
<tr>
<td></td>
<td>NAND $\rightarrow$ A.B+C (or any custom logic)</td>
<td>Active Region</td>
</tr>
<tr>
<td></td>
<td>Splitting active P well $\rightarrow$ P + N well</td>
<td>Active Region</td>
</tr>
<tr>
<td></td>
<td>Changing number of inputs</td>
<td>Active Region</td>
</tr>
<tr>
<td></td>
<td>Resizing 1x $\rightarrow$ 2x</td>
<td>Active Region</td>
</tr>
<tr>
<td></td>
<td>Interconnects / Power / GND - Thinning</td>
<td>Metal Layer 1</td>
</tr>
<tr>
<td><strong>Camouflage Cells</strong></td>
<td>NOR $\leftrightarrow$ NAND</td>
<td>Metal layer 1</td>
</tr>
<tr>
<td><strong>Insertion / Deletion</strong></td>
<td>Invertor NOT</td>
<td>Active Region</td>
</tr>
<tr>
<td></td>
<td>NAND / NOR</td>
<td>Active Region</td>
</tr>
<tr>
<td><strong>Biggest</strong></td>
<td>Capacitor</td>
<td>Active Region</td>
</tr>
</tbody>
</table>
Modification of Logic Gates

NAND ↔ NOR

NAND ↔ A.B + C
Insertion Based Trojans

Inverter* NOT Gate

MOS Capacitor#

*Capacitor as a Trojan Implemented: A2: Analog Malicious Hardware by Yang et. Al
#Trojan RS232-T1800 Implemented using two Inverters – Trust Hub
Image Pre-Processing

a. Original SEM Image
Scan the whole die as fast as possible while capturing sufficient feature details to compare with the layout.

b. Histogram Equalization
Increase contrast of doping regions in SEM image for better feature detection.

c. Gaussian Filtering
A 5x5 Gaussian filter is applied to remove the Gaussian noise in SEM image.

d. Median Filtering
A 3x3 median filter is applied to effectively remove noise and preserve the edge information to detect every unique footprint of a logic cell.

e. Thresholding
Segmenting SEM image into a binary image to separate the dark background and the foreground active region shape.
Trojan Detection

Filtering Denoising

Thresholding

Golden IC

SSIM

$$SSIM(x, y) = l(x, y) \cdot c(x, y) \cdot s(x, y)$$

Slowest

Speed: 32 µs /pixel

Fastest

Speed: 3.2 µs /pixel

Research
1. **Descriptor Assignment**
Assigning Fourier descriptor (FD) to every unique logic cell from SEM Image and Layout.

2. **Classifier Training**
Training machine learning model using different variations of a logic cell to account for imaging and manufacturing.

3. **Predictor Matching**
A machine learning based predictor matches the SEM and layout descriptors to detect a change.
Assigning descriptor to every unique logic cell from SEM Image and Layout.

**Classifier Training**

- $FD_1$, $FD_2$, ..., $FD_n$
- Label 1, Label 2, ..., Label n

Train SVM Classifier

**Descriptor Assignment**

- Pre-processed SEM
- Layout

**Gate1-2**
- Segmentation

**Gate1-5**
- Segmentation

**Predictor - Matching**

- Predictor: FD match?
- Yes
- Unchanged Logic cell
- No
- Logic Cell
- Flagged suspicious

All Rights Reserved
Descriptor Assignment --- Fourier Descriptor

a) Obtain cell’s mask by binary thresholding.

b) Obtain contour of the mask based on the pixel difference of the shape edge.

c) Obtain shape signature: The distance between contour centroid and contour coordinates.

d) Calculate Fourier transform of shape signature:

\[ f[k] = DFT(C[n]) = \frac{1}{N} \sum_{n=0}^{N-1} C[n] e^{(-\frac{j2\pi kn}{N})}, \ k = 0,1, ..., N - 1 \quad (1) \]

where \( f[k] \) is the Fourier transform of the \( k^{th} \) coordinate and \( C[n] \) is the contour.

e) Combine upper and lower Fourier descriptors for the whole gate:

\[ FD_g = [f_{upper}[k], f_{lower}[l]], k = 0,1, ..., N - 1 \ and \ l = 1,2 \quad (2) \]

where \( f_{upper} \) and \( f_{lower} \) is upper and lower half of logic cell respectively.
Training machine learning model using different variations of a logic cell to account for imaging and manufacturing.

Classifier Training

FD₁, FD₂, ..., FDₙ
Label 1, Label 2, ..., Label n

Train SVM Classifier

Descriptor Assignment

Pre-processed SEM

Layout

Segmentation

Gate1-2

FD₁-2

FD₁-5

Gate1-5

Segmentation

Gate1-2

Predictor - Matching

FD₁-2

FD₁-5

Predictor: FD match?

Yes

Unchanged Logic cell

No

Logic Cell Flagged suspicious

All Rights Reserved
A machine learning based predictor matches the SEM and layout descriptors to detect a change.
Layout vs. SEM Image Comparison

Location 1: Row1, Cell 2
- Detected: Modified Gate
- Not as Gate 2
- Modification 1

Location 2: Row1, Cell 4
- Detected: Modified Gate
- Not as Gate 3
- Modification 2

Location 3: Row2, Cell 4
- Existing: Gate (2)
- Insertion

Location 4: Row2, Cell 7
- Detected: Modified Gate
- Not as Gate 9
- Modification
1. Modification of logic cell
Logic cells encircled Location at 1, 2 and 4 are modified to emulate Trojan and successfully detected as change.

2. Insertion of logic cell
Logic cell insertion at empty space location 3 is detected as an insertion.
Trojan Scanner Challenges
Camouflage Cells Detection

Standard cell layout of regular 2-input (a) NAND and (b) NOR gate. Camouflaged standard cell layouts of 2-input (c) NAND and (d) NOR gate.

M1 – High kV required 15 kV

M2 – FIB Delayering & Low kV Imagining

OR

Very high kV 25-30 kV but little blurred
## RE vs Trojan Scanner

### Reverse Engineering vs Trojan Scanner

<table>
<thead>
<tr>
<th></th>
<th>Full Reverse Engineering</th>
<th>Trojan Scanner</th>
</tr>
</thead>
<tbody>
<tr>
<td># of samples required</td>
<td>50-100</td>
<td>1</td>
</tr>
<tr>
<td>Detected Trojans</td>
<td>All Types</td>
<td>All types except reliability Trojans</td>
</tr>
<tr>
<td>Processing time</td>
<td>Months</td>
<td>hours</td>
</tr>
<tr>
<td>Functionality extraction</td>
<td>Required</td>
<td>Not required</td>
</tr>
<tr>
<td>Gate Identification</td>
<td>Required</td>
<td>Not required</td>
</tr>
</tbody>
</table>

### Summary of Detection Methods

<table>
<thead>
<tr>
<th>Hardware Trojans</th>
<th>Logic Test</th>
<th>Power SCA</th>
<th>Delay SCA</th>
<th>Run Time</th>
<th>Trojan Scanner</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functional</td>
<td>Maybe</td>
<td>Maybe</td>
<td>Maybe</td>
<td>Maybe</td>
<td>✓</td>
</tr>
<tr>
<td>Parametric</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Big</td>
<td>Maybe</td>
<td>✓</td>
<td>Maybe</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Small</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>Maybe</td>
<td>✓</td>
</tr>
<tr>
<td>Tight</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>Maybe</td>
<td>✓</td>
</tr>
<tr>
<td>Loose</td>
<td>✓</td>
<td>Maybe</td>
<td>✓</td>
<td>Maybe</td>
<td>✓</td>
</tr>
</tbody>
</table>
Readings

- IEEE transaction on image processing: Image Quality Assessment: From Error Visibility to Structural Similarity
- ACM TODAES: Hardware Trojans: lessons learned after one decade of research
- IEEE design & test of computers: A survey of hardware Trojan taxonomy and detection