Can we build a Trustworthy Billion Transistor Chip?

Samuel Pagliarini

Senior Research Scientist

Carnegie Mellon University
Acknowledgment

Can we build a Trustworthy Billion Transistor Chip?
The DoD saga in microelectronics

- **Semiconductor** industry began in the United States
- The US government procured **37% of all the ICs** (1970)
  - Space and defence were the big drivers of the innovation
- Japan and Korea took over the memory market (Fujitsu and Samsung, 1980)
- Taiwan enters the scene with the first **fab-for-hire** (TSMC, 1990)
- Early 2000s, reports already point out that the US will **lose access** to cutting edge silicon
The DoD saga in microelectronics

1970: Foundries were plenty, everyone had their own foundry

Today: Only one trusted foundry in the US, obsolete and expensive to maintain

EXCELLENT RESEARCH OPPORTUNITY!
Hardware Security

- Different meanings to different folks

- Threat-based view of the problem
  - Counterfeit chips
  - IP piracy / IC overbuilding
  - Reverse engineering
  - Hardware Trojans
  - Side channel attacks

USB-to-UART chip from FTDI
CMU: Pioneers in Split-Fabrication

- Hybrid manufacturing solution
  - Trusted fab
  - Untrusted fab

- Leverages the high-performance of untrusted fabrication (fast and power efficient transistors)

- Mitigates trojan insertion
- Prevents IP theft
- Successfully demonstrated on different foundries/techs
Drawbacks of Split-Fabrication

- Hybrid PDK needed
- Yield assignment?
- Alignment concerns?
- Finding foundries willing to play along 😐

ALTERNATIVE TO HIDE DESIGN INTENT FROM FOUNDRY?
Split-Chip Methodology

- Core concept: one design, two chips
- May have ‘zero’ performance loss if split thoughtfully
Split-Chip Methodology for ASICs

- Domestic & trusted foundry
- Legacy technology node
- Control oriented

- Offshore & untrusted foundry
- High performance, high density
- Data oriented, efficient processing

- Three demonstration vehicles in 16nm FinFET
Silicon Demonstration #1

- 1000 GPS correlators @ 2GHz
- Master/slave architecture
- Additional hardware security techniques
  - Dummy logic, keyed logic, layout obfuscation

![Diagram showing 1000 correlators at 2GHz, master/slave architecture, and additional hardware security techniques.]
Keyed logic

- Insertion of XOR/XNOR gates and key inputs
- Circuit output is corrupted if incorrect key is applied
- Effective against IP theft
  - Relies on a (post) programmed memory
Lightweight Layout Obfuscation

- Grouping cells into blocks makes physical synthesis tractable, but clustering cells possibly exposes intent.

  - Seeding can guarantee each correlator has a unique layout.
    - The aggregate of scrambled correlators looks like random logic.
    - Automated approach to achieve lightweight obfuscation.

Hierarchical fplan looks-flat fplan
Silicon Demonstration #1

- Manual decisions
  - Where to split the hierarchy
  - Communication between chips (encryption?)

- Lessons learned
  - Complex trade-offs: bandwidth & latency vs security
  - Very design specific. Automation possible?
Silicon Validation

- Split-Chip GPS correlators verified to be functional
  - Zero performance loss

![Image of a green circuit board with labeled components: test equipment, acts as trusted IC; Correlator chip, acts as untrusted IC; key bits from trusted]
Silicon Demonstration #2

- Design characteristics
  - 300k correlators in a 5mm x 5mm die
  - Corresponds to approximately 20M standard cells (high density)
  - 5 clock domains (2.4GHz to 1MHz)
  - Floorplan organized in a 3x3 matrix
GPS correlator chip - floorplan

- Roughly 300,000 correlators on the same die (~20M std cells)

Power distribution concerns
GPS correlator chip - floorplan

- Roughly 300,000 correlators on the same die (~20M std cells)

[config 1]
GPS correlator chip - floorplan

- **channel_0**: 1.2 GHz
- **channel_1**: 1.2 GHz
- **channel_2**: 1.2 GHz
- **channel_3**: 5mm
- **channel_4**: 5mm
- **channel_5**: 5mm
- **channel_6**: 5mm
- **channel_7**: 5mm
- **channel_8**: 5mm

- Roughly 300,000 correlators on the same die (~20M std cells)

[config 2]
GPS correlator chip - floorplan

- channel_0 (600 MHz)
- channel_1 (600 MHz)
- channel_2 (600 MHz)
- channel_3 (600 MHz)
- channel_4
- channel_5
- channel_6
- channel_7
- channel_8

Roughly 300,000 correlators on the same die (~20M std cells)

[config 3]
GPS correlator chip - floorplan

- Channel 0 (300 MHz)
- Channel 1 (300 MHz)
- Channel 2 (300 MHz)
- Channel 3 (300 MHz)
- Channel 4 (300 MHz)
- Channel 5 (300 MHz)
- Channel 6 (300 MHz)
- Channel 7 (300 MHz)
- Channel 8 (300 MHz)

Roughly 300,000 correlators on the same die (~20M std cells)

[config 4]
GPS correlator chip - floorplan

- Roughly 300,000 correlators on the same die (~20M std cells)

[config 5 – test/bringup only]
Split-Chip instead of Split-Fab

- Need for a CAD tool to assess the partitioning trade-offs
  - Technologies can be very different in nature
- Obfuscation schemes like keyed logic can be sought

How to achieve obfuscation?
Obfuscation on Untrusted Chip

- Modelled some existing logic locking techniques within the tool
- Vast literature available, several variants proposed

Split-Chip Design

Advanced Node Untrusted Foundry

Legacy Node Trusted Foundry

Original SoC

Untrusted Chip

Trusted Chip

PPA Critical IP

Add Secondary obfuscation

Security + PPA Critical IP

Security Critical IP

AES

RSA

MD5

DES3

SHA256
Vulnerability Optimization

\[ X = \{0, 1, 2, 1, 0, 3, \ldots\} \]

0: trusted
1: untrusted
2: untrusted, key logic
3: untrusted, FSM obf.

\[ \min f(x) = \text{Vulnerability}(x) = \text{Exposure}(C) \times \text{Criticality}(M) \]

s.t.
constraints met

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Exposure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trusted</td>
<td>0.1</td>
</tr>
<tr>
<td>Untrusted</td>
<td>1.0</td>
</tr>
<tr>
<td>Untrusted w/ FSM Obf.</td>
<td>0.9</td>
</tr>
<tr>
<td>Untrusted w/ Keyed Logic</td>
<td>0.8</td>
</tr>
</tbody>
</table>

**Technique exposure**
- Ranks technique-related risk
- Score starts from 0 for maximal security on trusted chip

**Module criticality**
- User-defined scores
- Quantifies module risk/importance
Case study: Common Evaluation Platform

- No constraints relaxed from original design
- Fully on Untrusted IC w/ high vulnerability

- Several constraints relaxed
- Significant decrease in vulnerability
Case study: Common Evaluation Platform

- Added keyed logic as secondary obfuscation to further reduce vulnerability

Manual observation: small tweak allows obfuscation on every system module
Pairing and encryption framework

- Uses SRAM as **signature** and asymmetric encryption (RSA)
- 1-to-1 pairing of Trusted and Untrusted ICs
  - Threats that are **unique to Split-Chip**
- Only **public key** is exchanged in the clear
Can we build a Trustworthy Billion Transistor Chip?
We can build a Trustworthy-ish Billion Transistor Chip

How to measure trust?
Lack of metrics

- Widespread problem for the hardware security community
- Self-critical: my interpretation of the state of the art

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Exposure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trusted</td>
<td>0.1</td>
</tr>
<tr>
<td>Untrusted</td>
<td>1.0</td>
</tr>
<tr>
<td>Untrusted w/ FSM Obf.</td>
<td>0.9</td>
</tr>
<tr>
<td>Untrusted w/ Keyed Logic</td>
<td>0.8</td>
</tr>
</tbody>
</table>
Thanks!