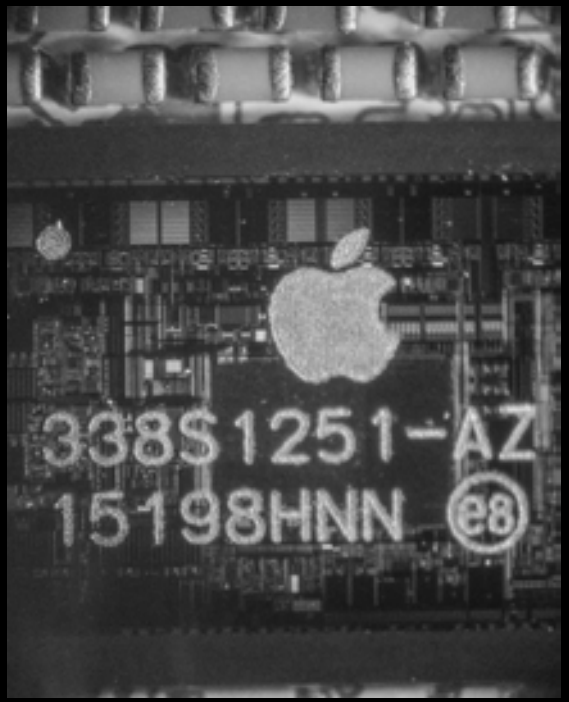
IRIS: Infra-Red, in situ

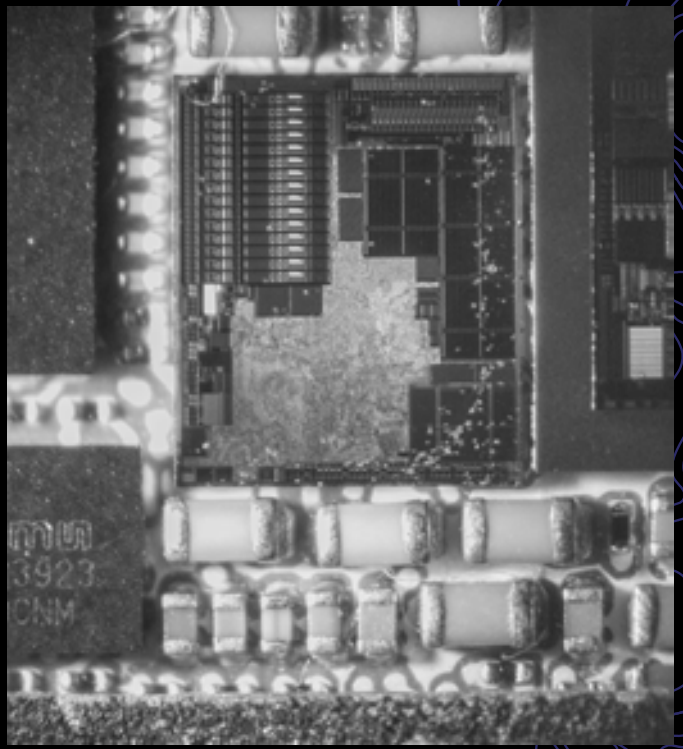
Non-destructive, in-circuit verification of silicon

bunnie | masto: @bunnie@treehouse.systems | bsky: @bunnie.org

38C3

"Non-destructive In-circuit Verification of Silicon"





3803

Problem Statement:

I want to control my data.

My data is in my hardware.

Therefore, I need to trust my hardware.

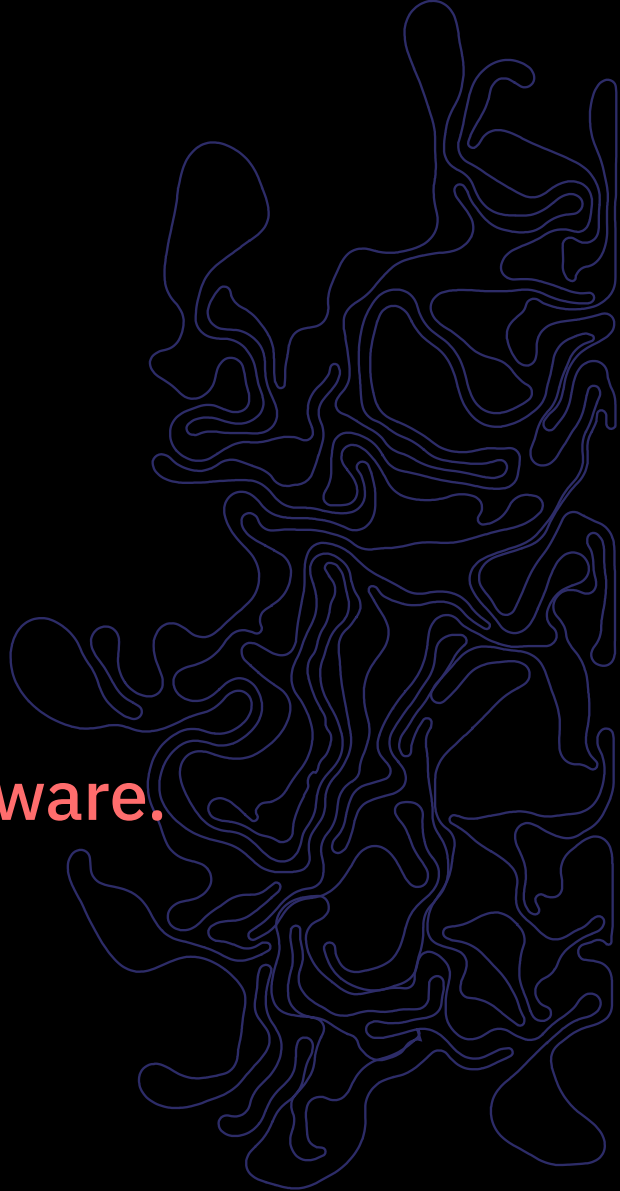
Problem Statement:

I want to control my data.

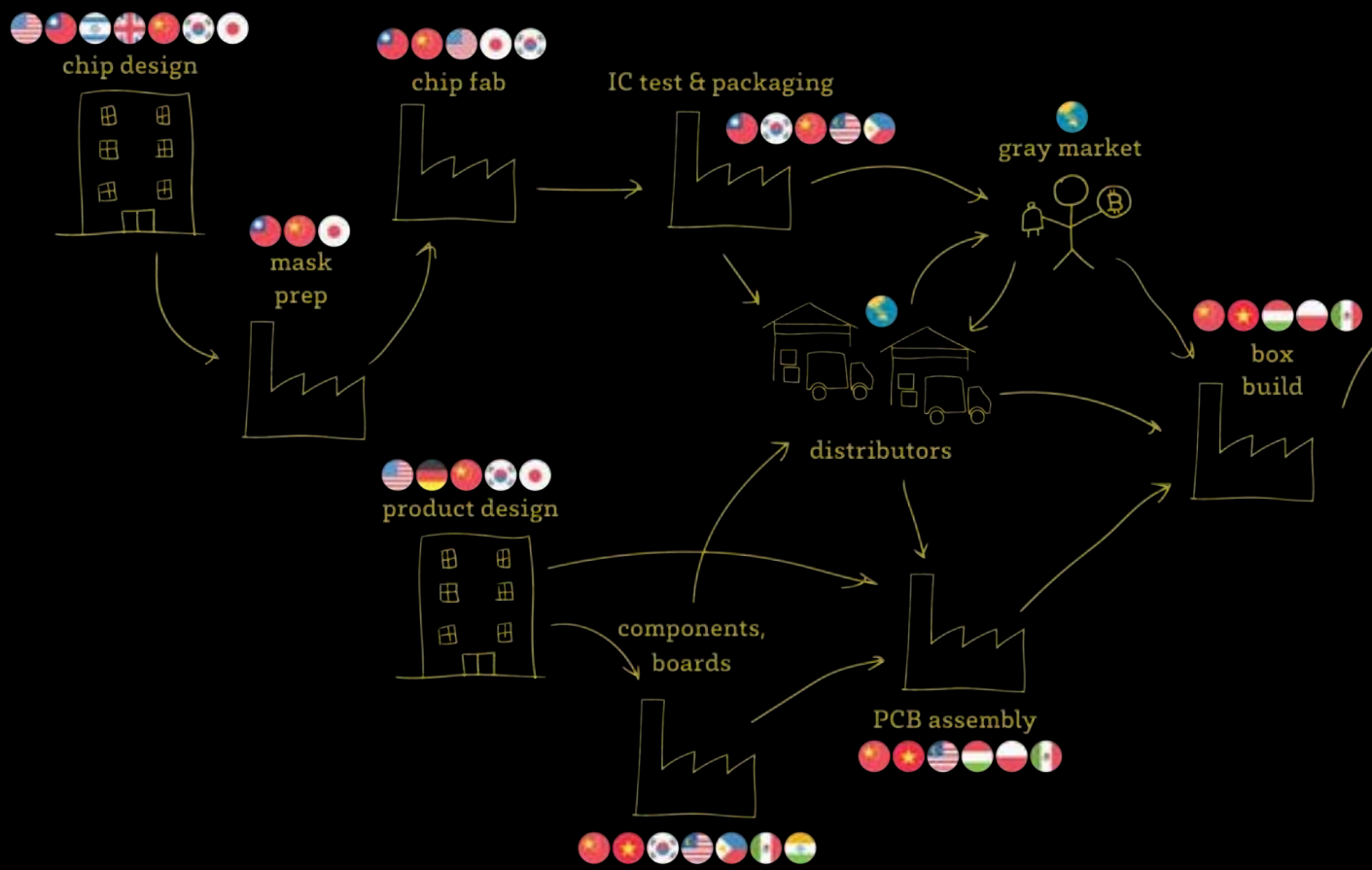
My data is in my hardware.

Therefore, I need to trust my hardware.

...how can I trust my hardware?

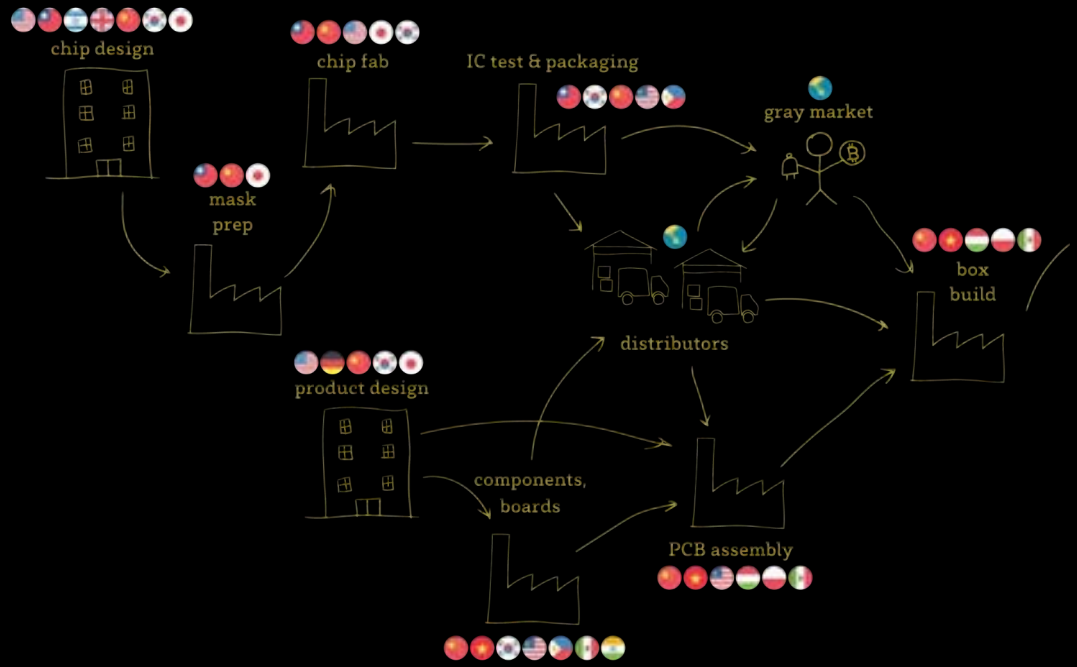


Trust Issues: Concerns About "The Supply Chain"



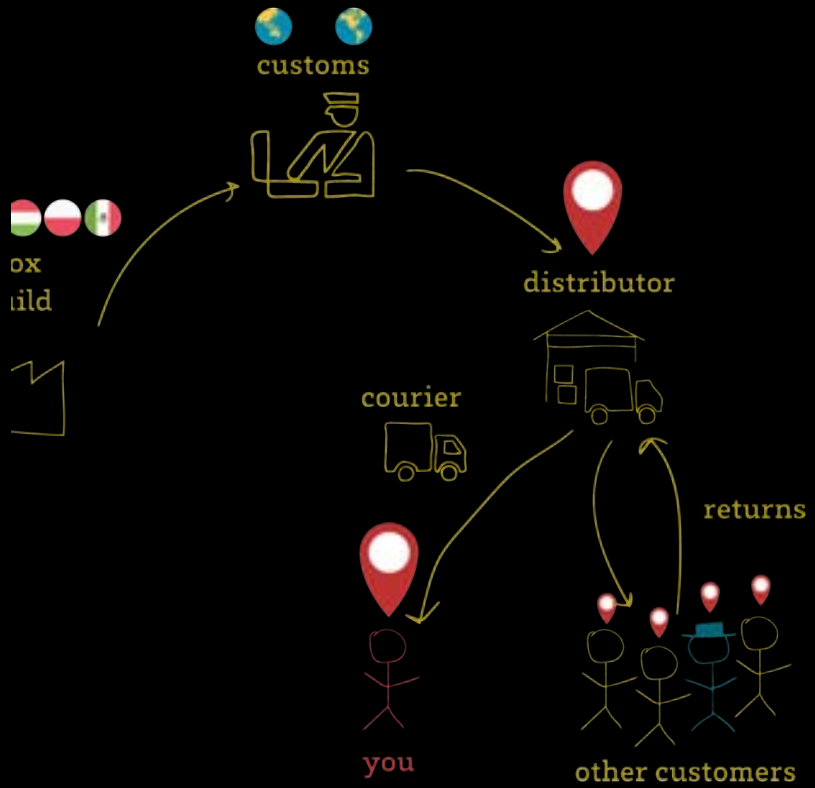
3803

The supply side gets all the political attention...



- Look at all those flags we don't recognize/trust!
- OMG did you say *gray market*!
- Wait ... so many jobs not in our country!
- [COUNTRY NAME] FIRST!!!
  - Cue billions in subsidies and political shenanigans

...but the distribution side is just as problematic!

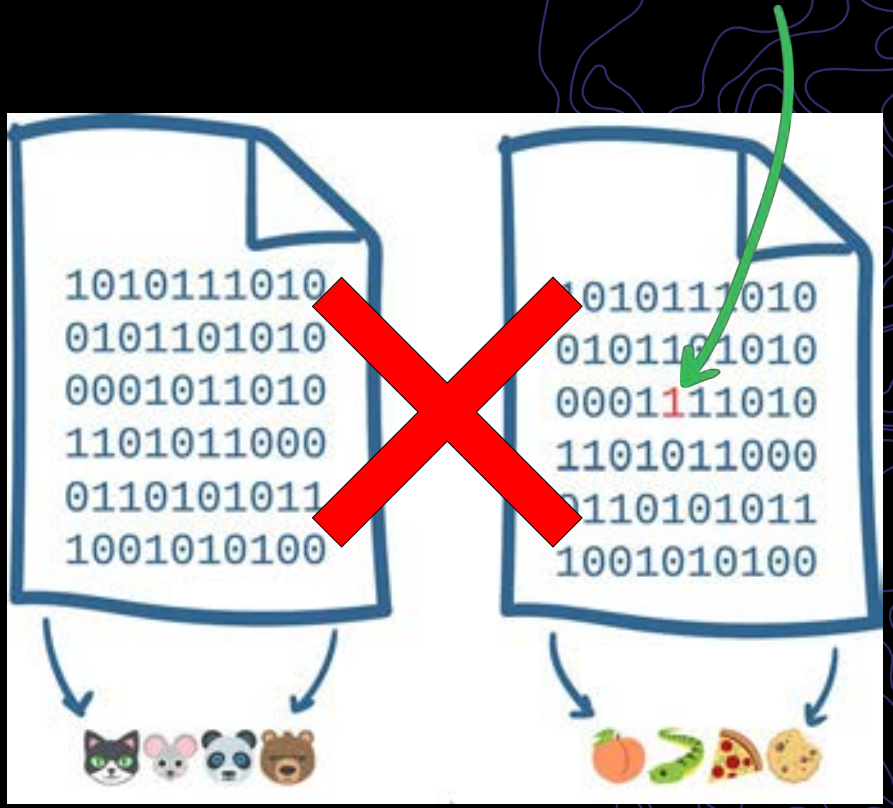


- In software, we never trust the CDN
  - Would you download software over unencrypted http?
  - ...yet we instinctively trust unauthenticated couriers?
- "Any tourist" can buy, modify, return products
- Distributors aren't security experts

The Big Problem: You Can't "Hash" Hardware

- There is no convenient, easy-to-use method to confirm the correctness of hardware immediately before its use

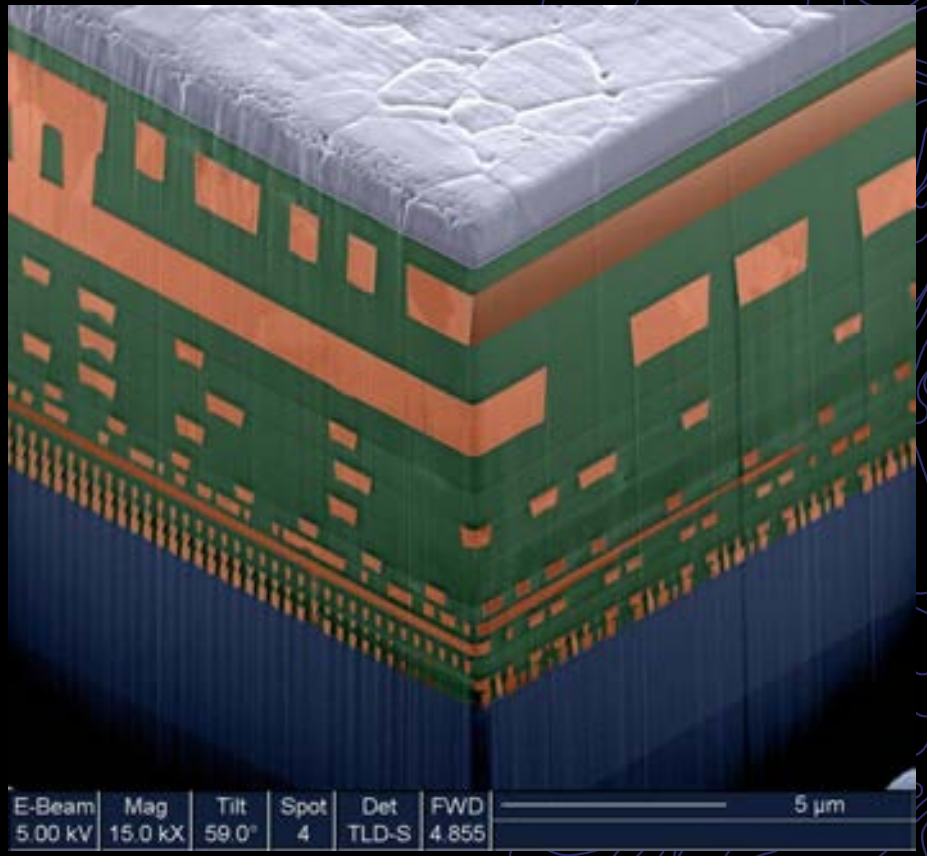
- Hardware is one big "Time of Check versus Time of Use" (TOCTOU) problem!



* This does not exist for hardware

SEM Analysis is Destructive

- SEM can analyze a chip to the transistor level
- Requires cross-sectioning the chip for the beam to reach internal layers
- Can't check and use a specific chip



https://www-03.ibm.com/press/us/en/photo/19014.wss

Alternatives Exist, but...

- "Ptychographic X-Ray Imaging" to the rescue?
  - Non-destructive
  - 3D imaging of complex chips
  - Great for reverse engineering and design verification

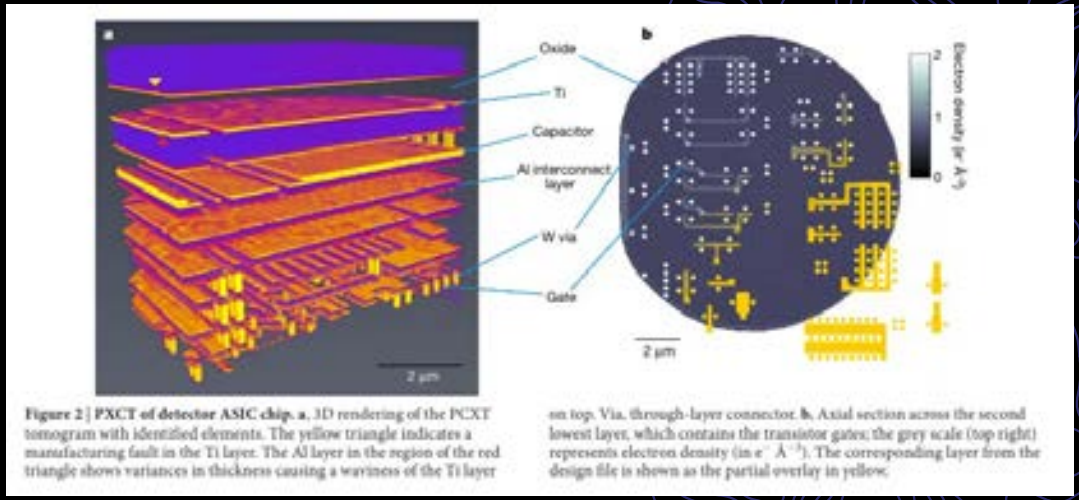


Figure 2 | PXCT of detector ASIC chip. a, 3D rendering of the PCXT tomogram with identified elements. The yellow triangle indicates a manufacturing fault in the Ti layer. The Al layer in the region of the red triangle shows variances in thickness causing a waviness of the Ti layer on top. Via, through-layer connector. **b**, Axial section across the second lowest layer, which contains the transistor gates; the grey scale (top right) represents electron density (in $e^- Å^{-3}$). The corresponding layer from the design file is shown as the partial overlay in yellow.

https://www.nature.com/articles/nature21698

...They Require a Building-Sized Microscope

https://www.psi.ch/en/sls/about-sls

Problem Statement:

I want to control my data.

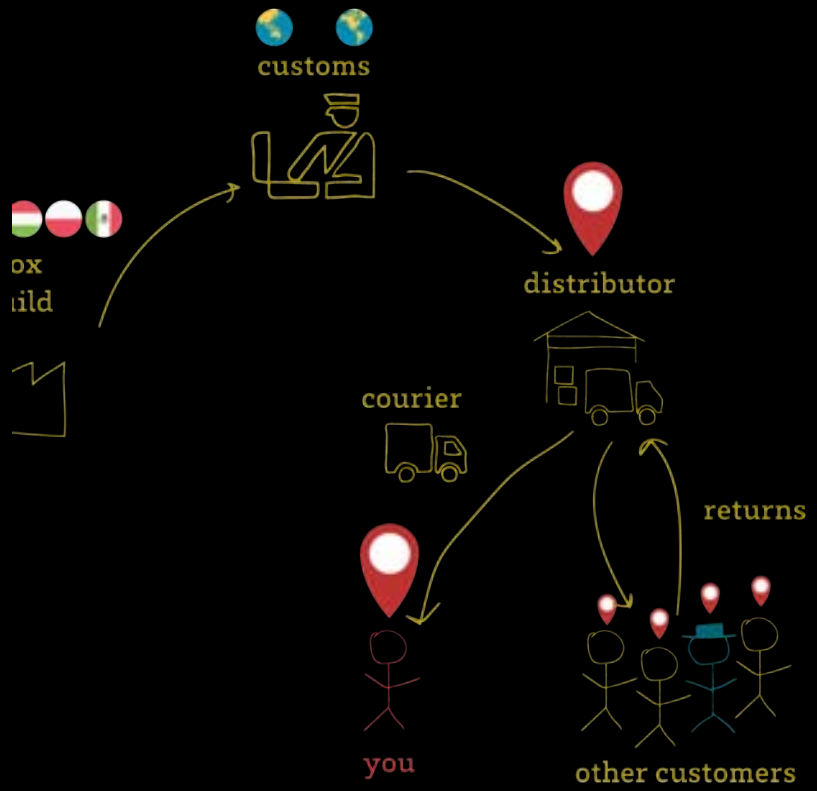
My data is in my hardware.

Therefore, I need to trust my hardware.

Step 1:

Define the Threat Model

Assumption: The Threat Is the Supply Chain

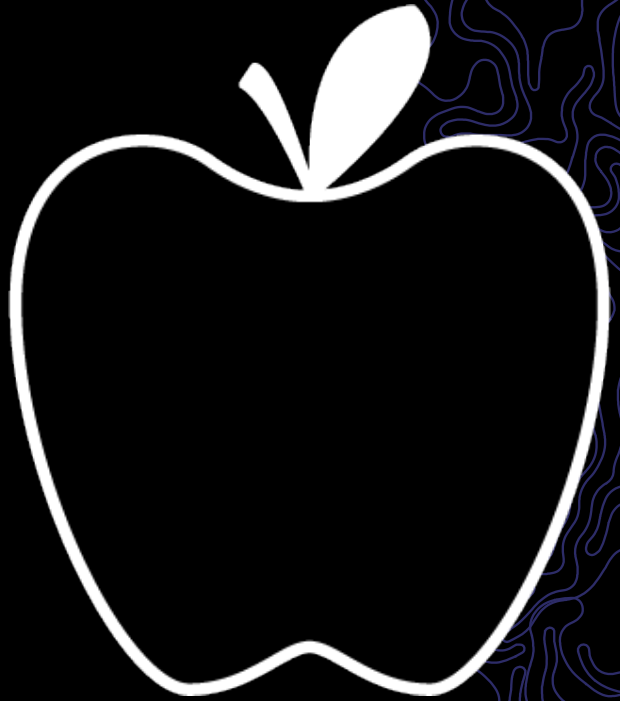


- "Is it Hackable?" is different from "Has it been modified?"
  - "Can I trust this piece of hardware" as-delivered is *the* question for this talk
- Thus:
  - "Can this piece of hardware resist arbitrary probing" after theft/seizure is *not* in-scope
  - In my opinion, you can't win that game anyways...

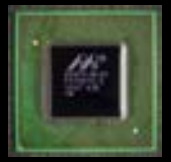
Unpacking the Supply Chain Threat Model by Analogy

Can I trust this chip?

Is this safe to eat?



Limitations of the Analogy



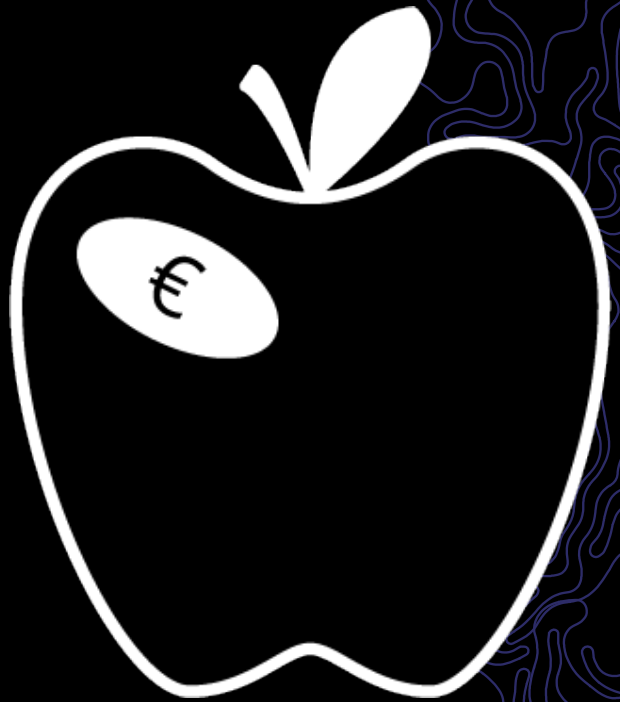
- Stakes:
  - A modified chip in a server could impact millions of users
- Remedies:
  - Chips are made in billion-dollar fabs

- Stakes:
  - A poisoned fruit might make the person who ate it sick
- Remedies:
  - Fruit grows on trees

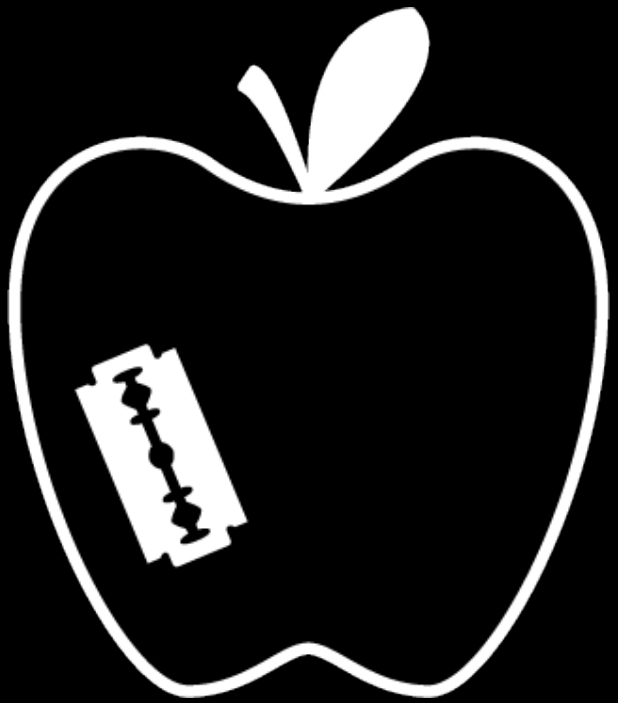
However, both require global supply chains...

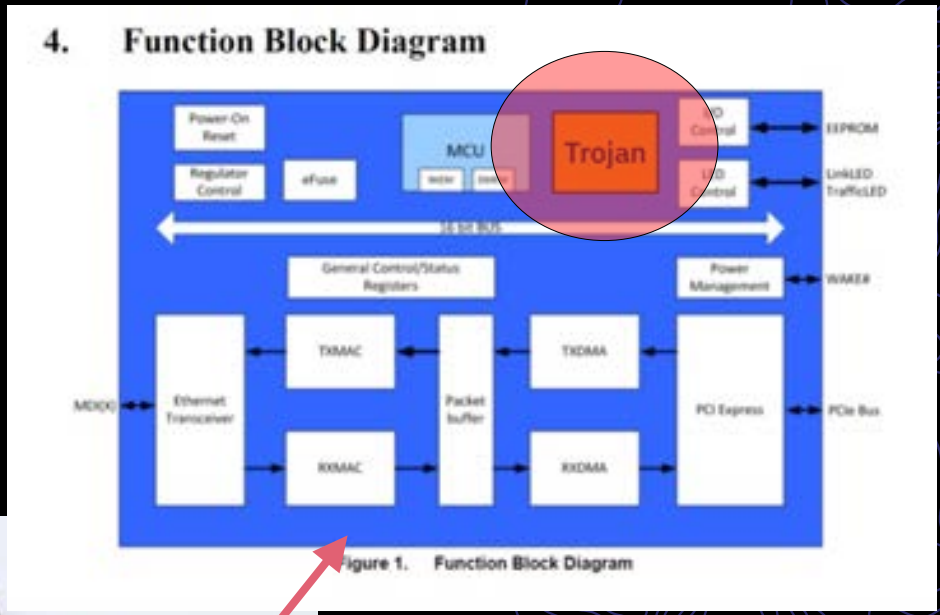
...and we verify our chips about as much as we verify our fruit.

Level 0: Detectable at Home (Point of Use)
Exemplar: Misrepresentation of Goods



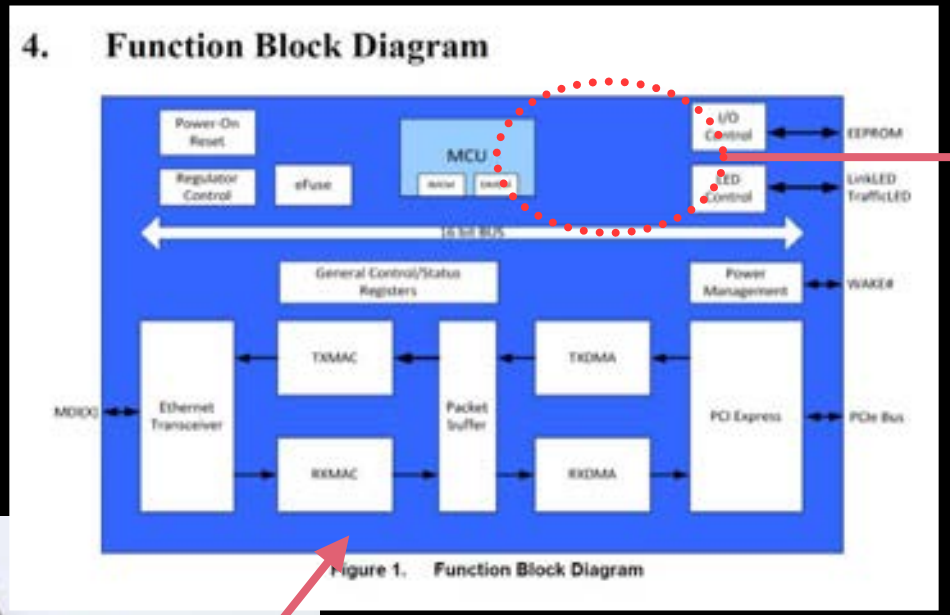
Level 1: Easily Detected With $1k–$10k Tools
"Block-Level Modifications"

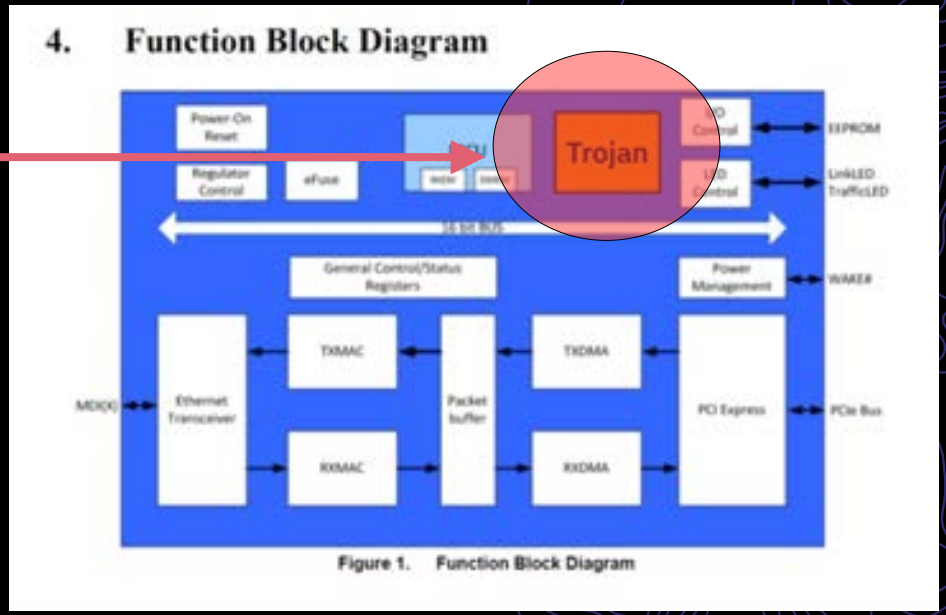




3803

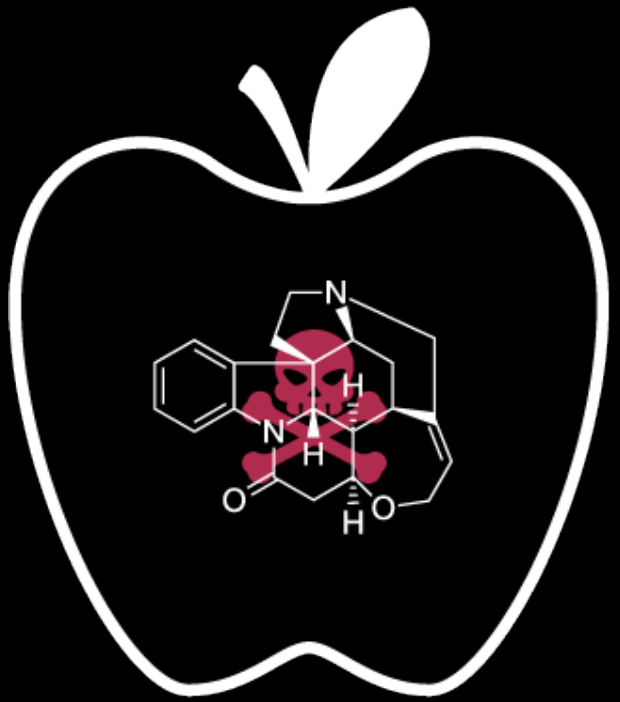
Examplar: Modified NIC Chip

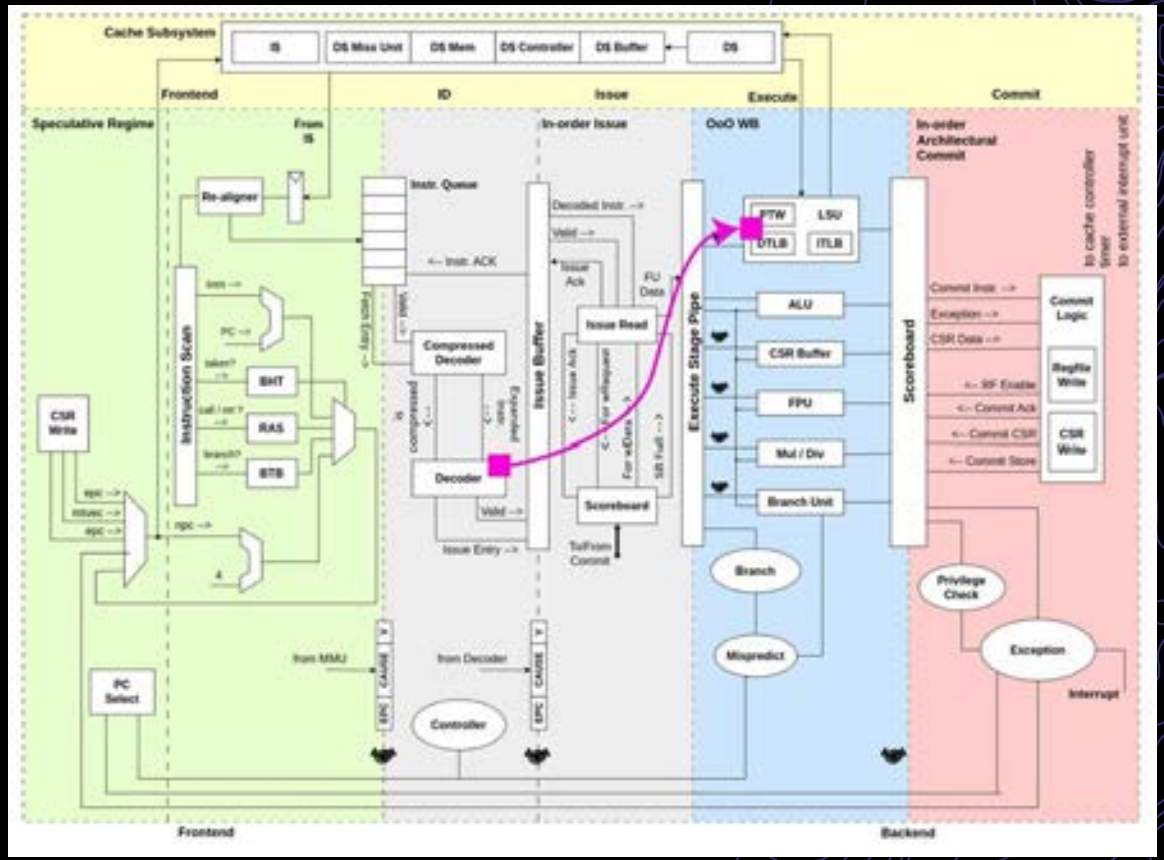




- NIC blocks available now as F/OSS or low-cost IP
- Uses older process (~65nm)
- Estimate <$300k up-front cost to mount attack
- Unit cost is possibly even profitable

Level 2: Detected With $10k–$100k tools
Sub-block RTL-Level Modifications

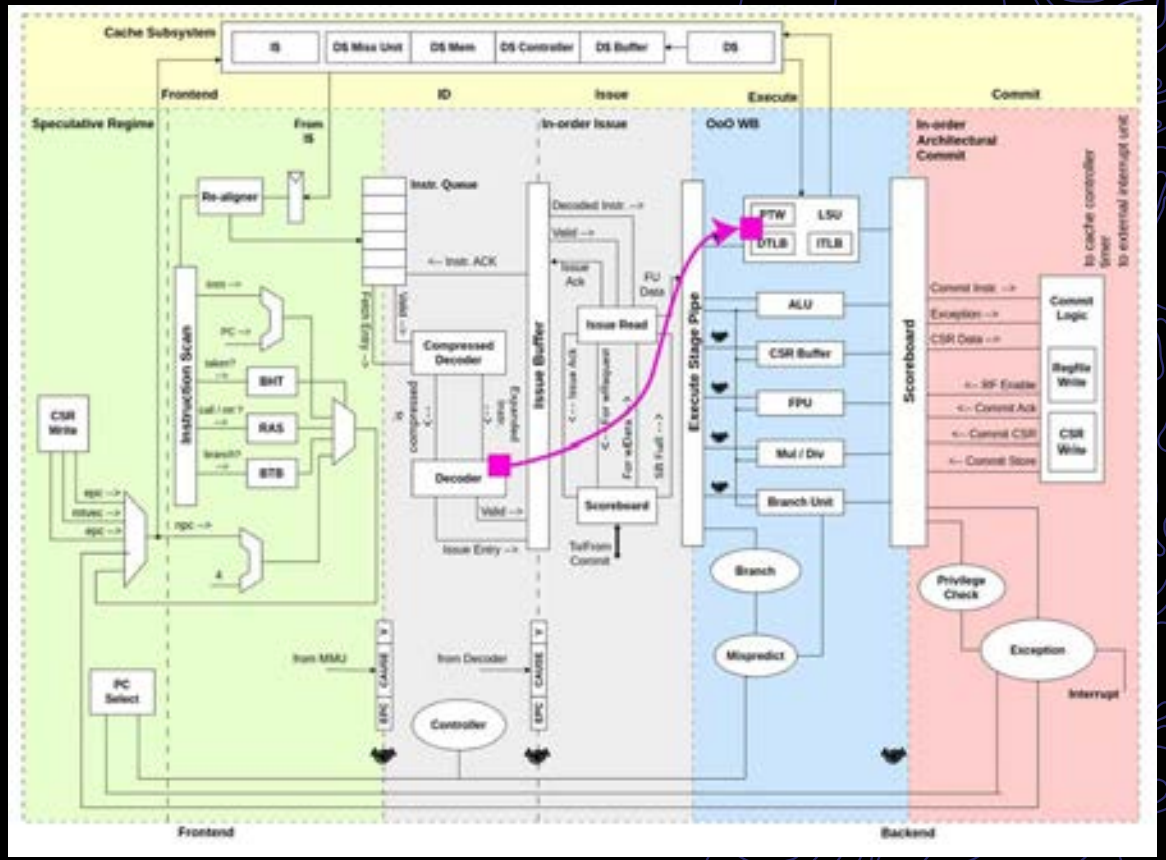




https://github.com/openhwgroup/cva6?tab=readme-ov-file

Key Assumptions

- Assumption: there are two versions of the chip in the supply chain, one with the modification, and one without
  - "Bad by design" is a different question
  - https://ghostwriteattack.com/riscvuzz.pdf



https://github.com/openhwgroup/cva6?tab=readme-ov-file

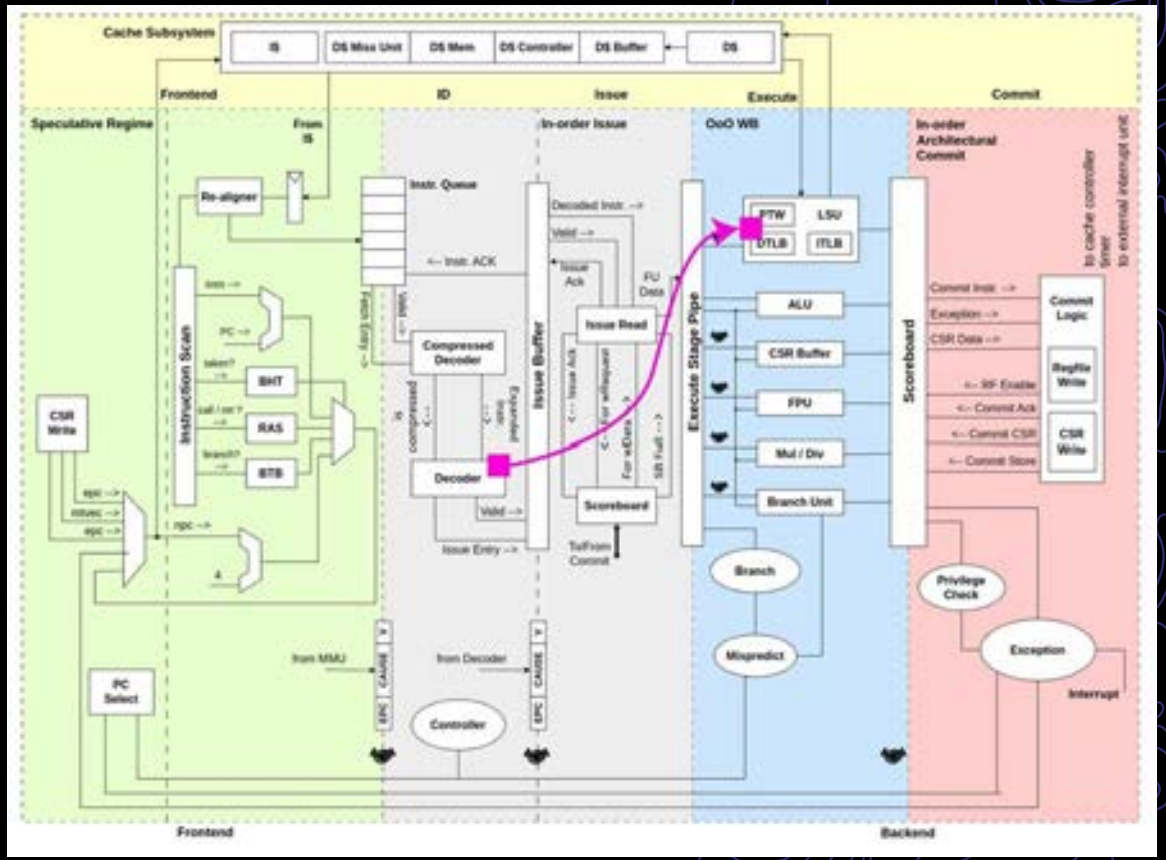
Exemplar: Modifying a CPU Pipeline

- Observation:
  - ra (x1) on RISC-V is the link register
  - Compiled code only uses it in limited contexts, e.g.: "jalr, ra target"
- Create a memory protection bypass with trigger using this primitive

```
ffd0381e <xous_kernel::arch::riscv::current_pid>:
ffd0381e:   1141                    addi    sp,sp,-16
ffd03820:   c606                    sw  ra,12(sp)
ffd03822:   0000f097            auipc   ra,0xf
ffd03826:   db4080e7            jalr    -588(ra) # ffd125d6 <__read_satp>
ffd0382a:   8159                    srli    a0,a0,0x16
ffd0382c:   0ff57593            zext.b  a1,a0
ffd03830:   c581                    beqz    a1,ffd03838 <xous_kernel::arch::riscv::curre
ffd03832:   40b2                    lw  ra,12(sp)
ffd03834:   0141                    addi    sp,sp,16
ffd03836:   8082                    ret
ffd03838:   ffd15537            lui a0,0xffd15
ffd0383c:   a2450513            addi    a0,a0,-1500 # ffd14a24 <_ebss+0xfff91d64>
ffd03840:   0000d097            auipc   ra,0xd
ffd03844:   01a080e7            jalr    26(ra) # ffd1085a <core::option::unwrap_failed>
```

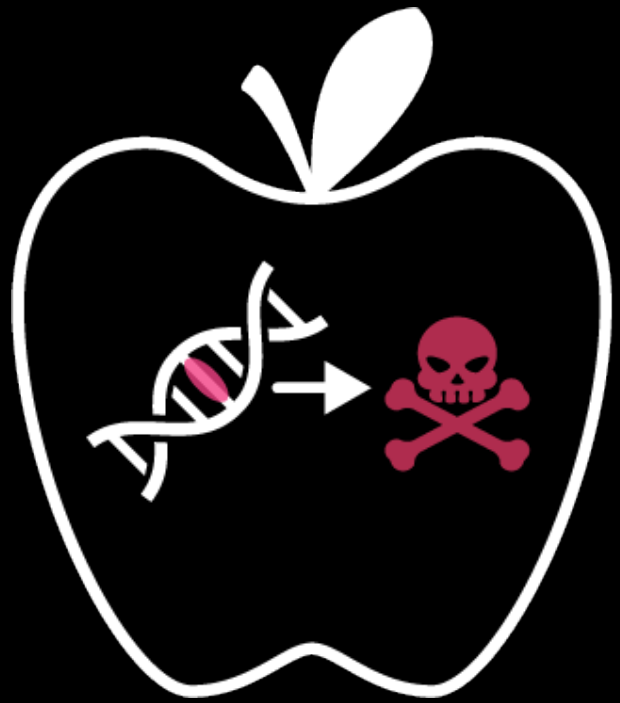
Exemplar: Modifying a CPU Pipeline

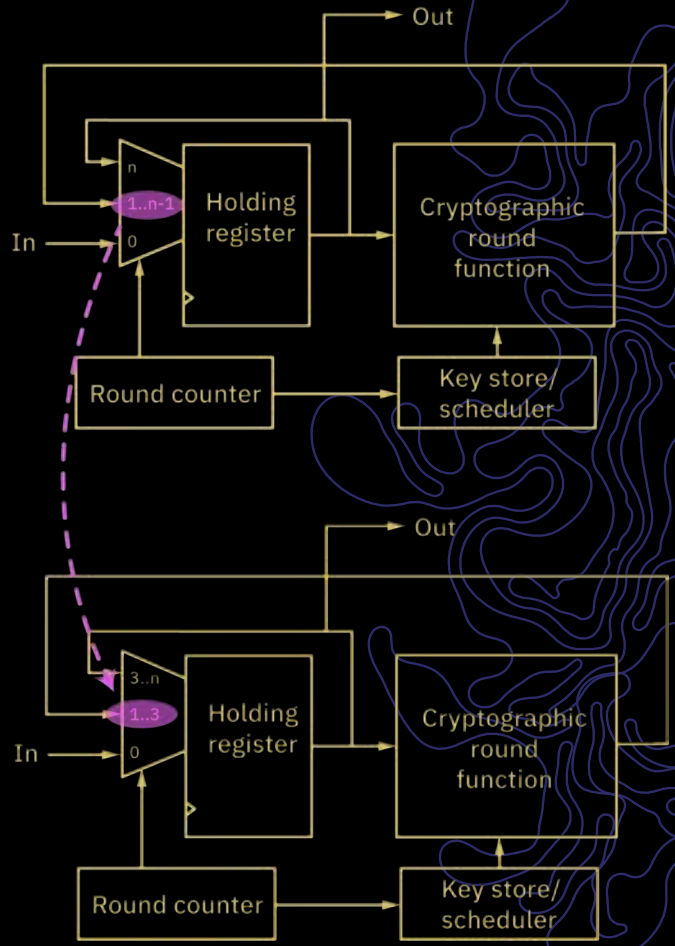
- Hypothetical Trojan:
  - Decoding a "load" using ra as the address base...
  - ...causes ra contents to be treated as if a physical address
  - Thus bypassing virtual memory protection
  - Optional:
    - Use unlock "knock" sequence to frustrate discovery by fuzzing
    - i.e. sequence is armed by a preceding "dummy" instruction like "addi x0, x0, 0x666"
  - Requires O(10)-O(100) logic cells to implement



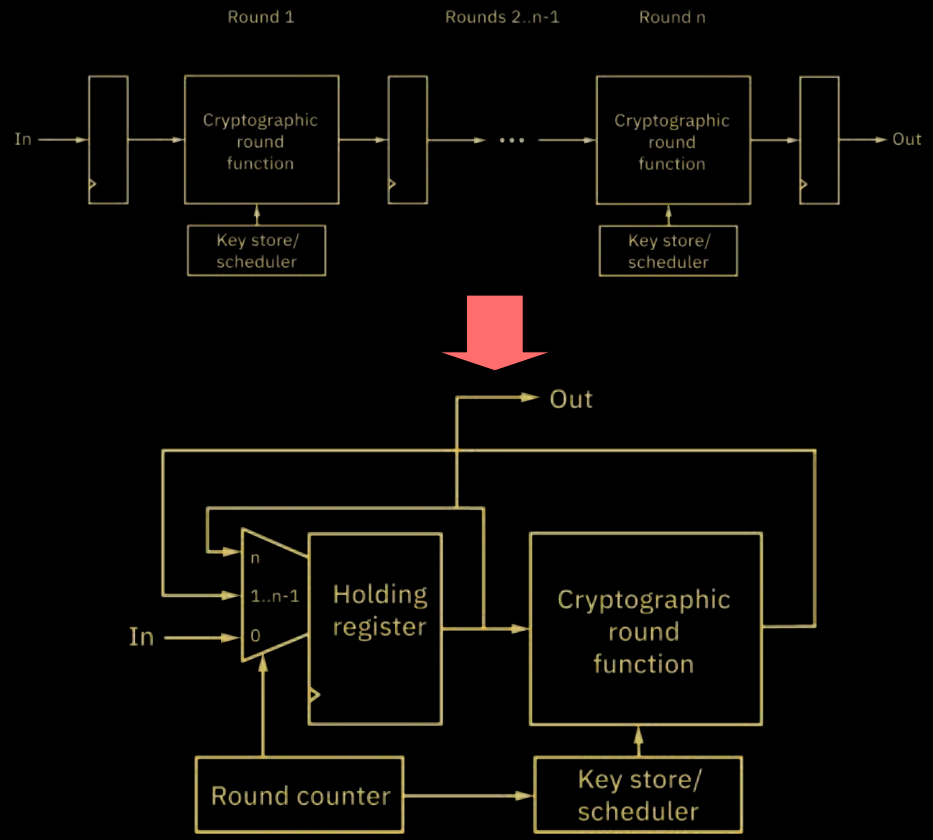
3803

Level 3: Requires $1mm+ Tools/Novel Techniques
Exemplar: Tailored Mask Edits



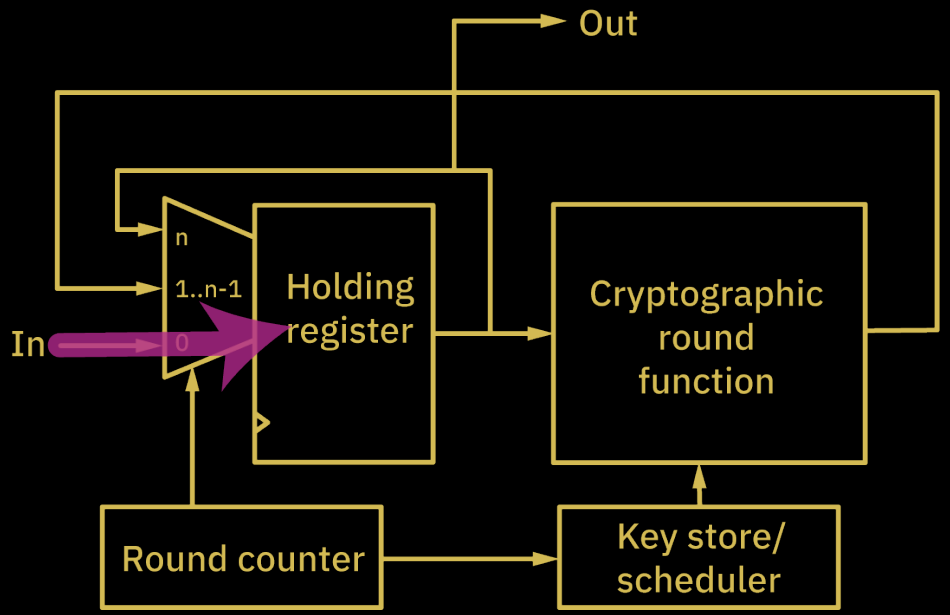


Exemplar: Reduced Round Cryptography Using a Small Mask Edit



- Some ciphers use repeated round of computation for security
  - Instead of implementing N copies of the hardware...
  - ...a single round is implemented in a loop

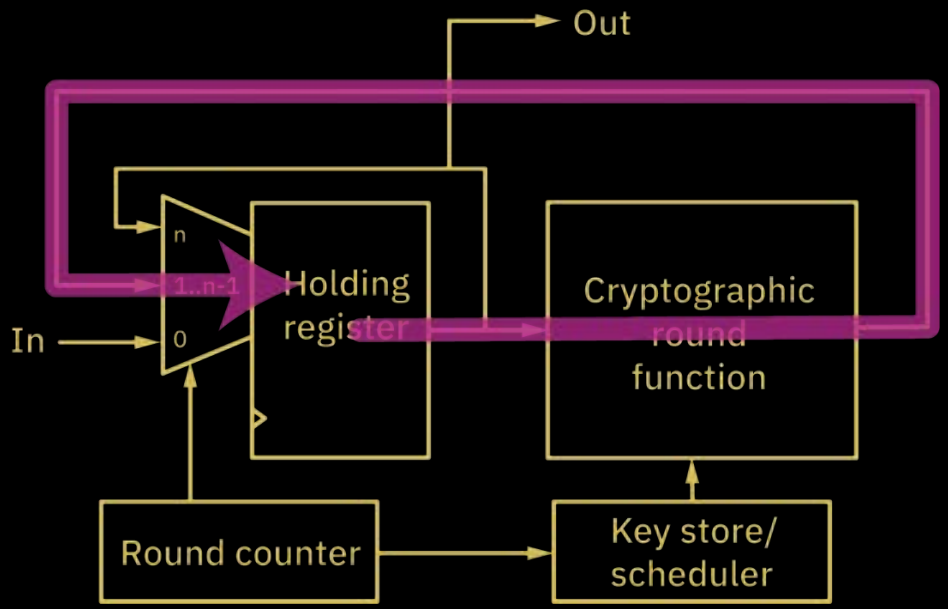
Background: Multi-Round Cipher



- Round "0"
  - Load in fresh data

```
0000 - load
0001 - round
0010 - round
0011 - round
0100 - round
0101 - round
0110 - round
0111 - round
1000 - round
1001 - round
1010 - round
1011 - round
1100 - round
1101 - round
1110 - round
1111 - hold
```

Background: Multi-Round Cipher

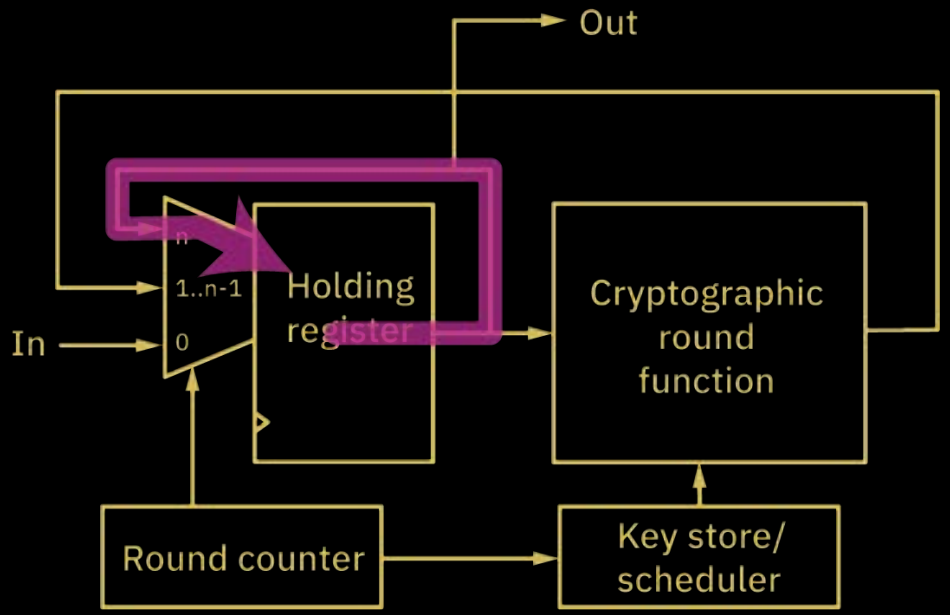


- Rounds "1..(n-1)"
  - Repeatedly apply the round function to the data

```
0000 - load
0001 - round
0010 - round
0011 - round
0100 - round
0101 - round
0110 - round
0111 - round
1000 - round
1001 - round
1010 - round
1011 - round
1100 - round
1101 - round
1110 - round
1111 - hold
```

e.g. 14 rounds for AES-256

Background: Multi-Round Cipher

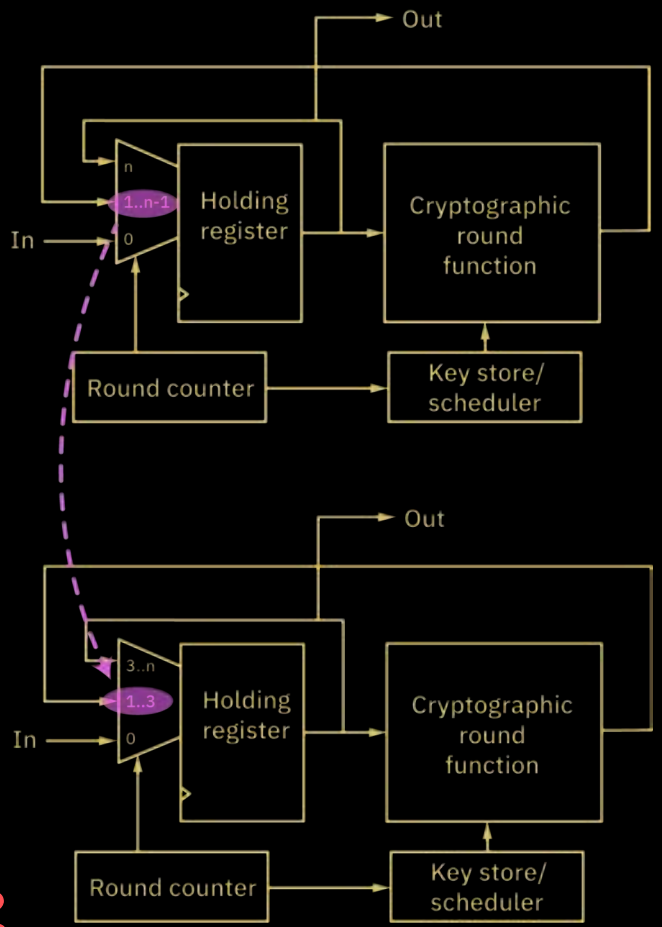


- Round "n"
  - Hold the result for read-out

```
0000 - load
0001 - round
0010 - round
0011 - round
0100 - round
0101 - round
0110 - round
0111 - round
1000 - round
1001 - round
1010 - round
1011 - round
1100 - round
1101 - round
1110 - round
1111 - hold
```

3803

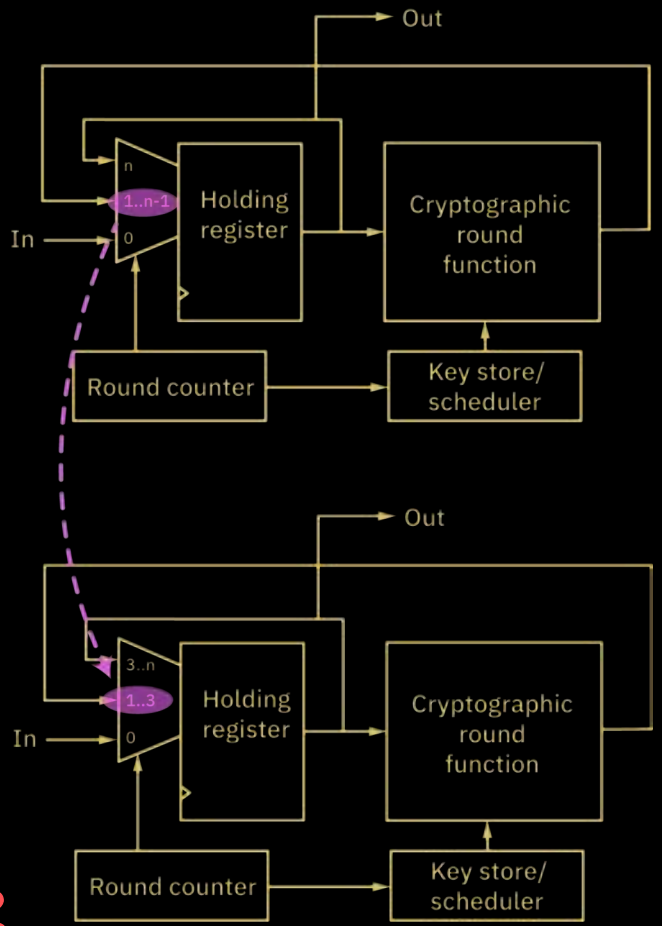
The Attack



- What if you tied the upper bits of the "holding register" selection input together?

```
0000 - load
0001 - round
0010 - round
0011 - round
0100 - round
0101 - round
0110 - round
0111 - round
1000 - round
1001 - round
1010 - round
1011 - round
1100 - round
1101 - round
1110 - round
1111 - hold
```

The Attack

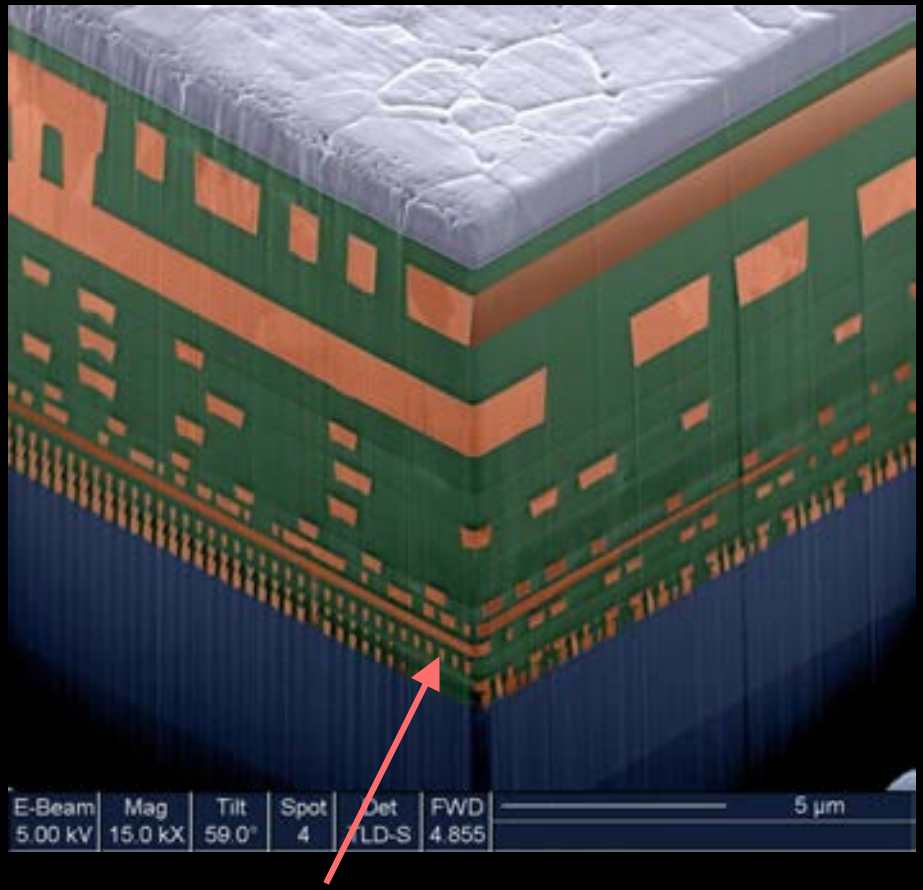


- What if you tied the upper bits of the "holding register" selection input together?

```
0000 - load
0001 - round
1110 - round
1111 - hold
0000 - load
0001 - round
1110 - round
1111 - hold
0000 - load
0001 - round
1110 - round
1111 - hold
0000 - load
0001 - round
1110 - round
1111 - hold
```

- Only 2 rounds matter!
  - But! Timing side channel and power side channel looks "as if" the full rounds happened

The Attack



- Why it's sneaky:
  - Symmetric reduction of rounds -> decryption/encryption works "fine"
  - Sidechannels same or very similar
  - Reduced-round variants still have reasonable bulk statistics
  - If secret key is truly kept secret inside the chip...
    - ...Detection requires cryptanalysis of ciphertext
- Why it's hard to detect:
  - Maybe just a via-only change!

Threat Model Recap

Level 3: Detected only with $1mm+ tools and/or requires new techniques

Level 2: Detected with $10k-$100k tools

Level 1: Detected with $1k-$10k tools

Level 0: Detected with <$1k tools

- Current state of practice:
  - Level 3: maybe destructive analysis required???
  - Level 2: academic papers
  - Level 1: practiced by targeted industries
  - Level 0: routinely practiced

3803

In Practice, Nobody is Checking

Nobody is checking

A few people are checking

- The general public does not check chips beyond Level 0
  - Public companies that do check also do not disclose problems
  - Disclosing supply chain issues is bad for business
- Threat actors have broad latitude to operate without consequence

This Work: Infra Red, *in situ* (IRIS)

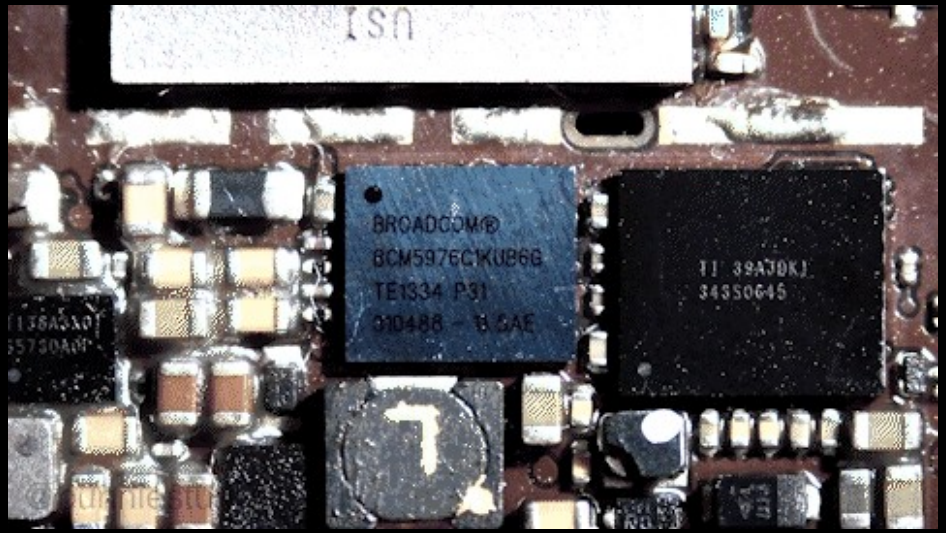
Academics & agencies

Targeted Industries

Point of Use (at-home)

- Reduce detection barrier by an order of magnitude
  - Increase the capability of at-home detection by at least one level
  - *Improve* trust in hardware for everyday people

Introducing IRIS: Infra-Red, *in situ* Verification of Silicon

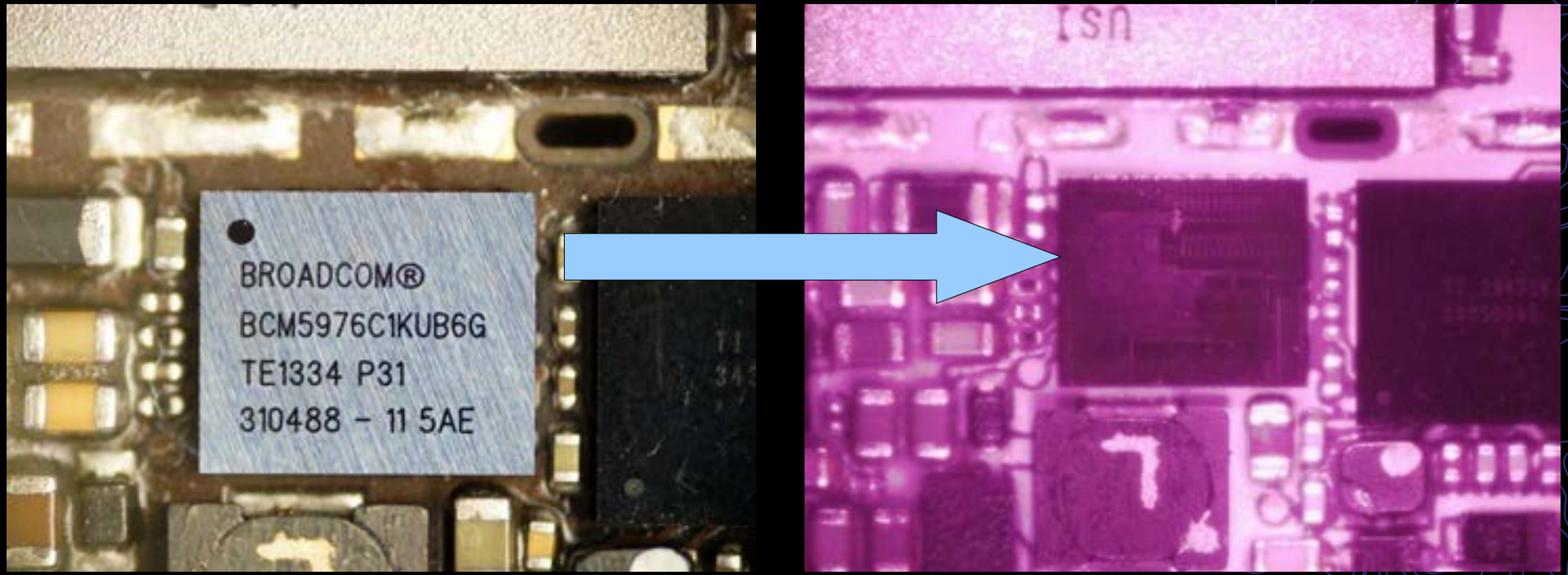


- A method for inspecting certain types of chips
- After they are attached to a circuit board
- Without damage

What Type of Chips?

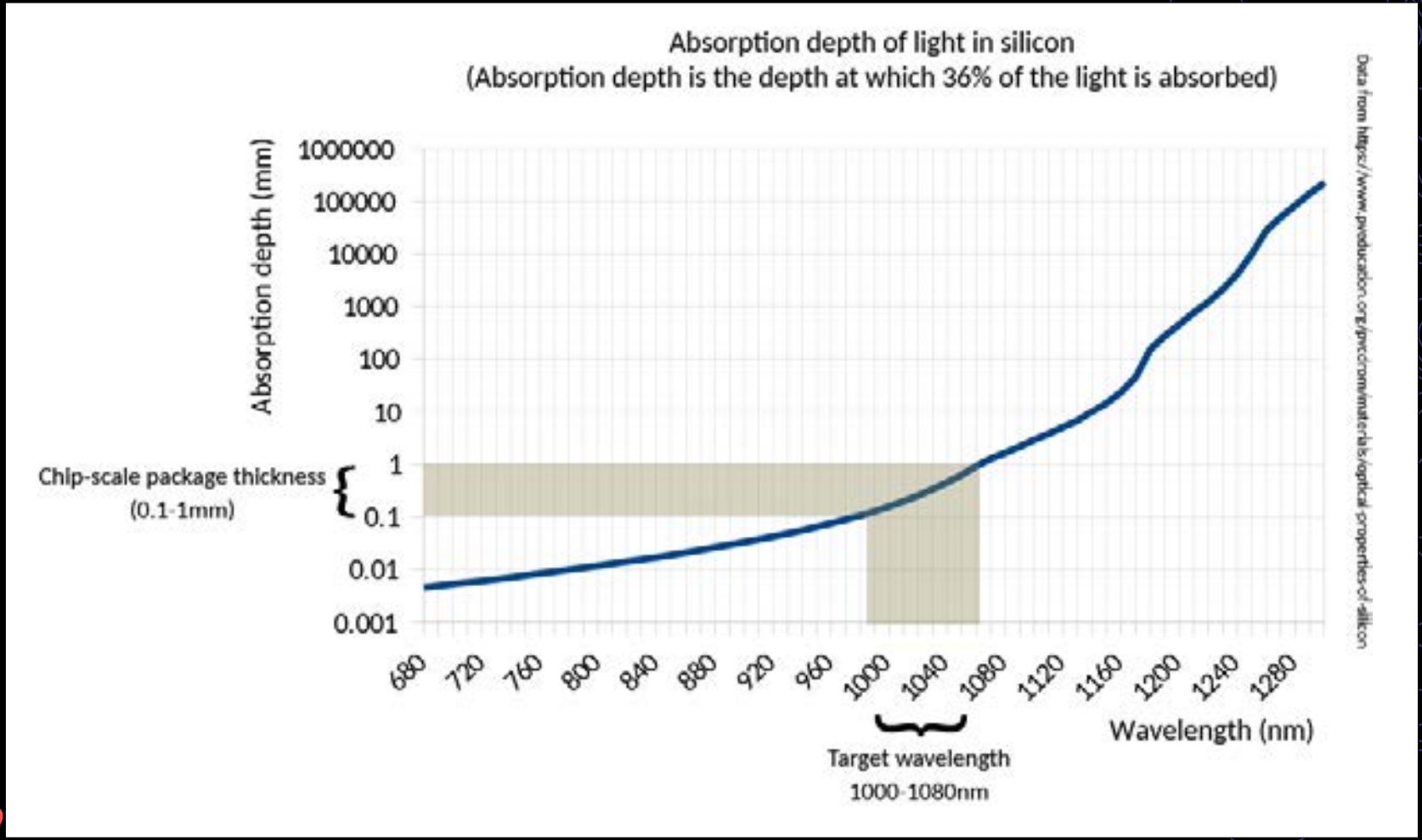
- Short answer: "The shiny ones"
  - WLCSP or FCBGA types of packages
  - Exposed silicon back with no film or paint applied
  - Ideally polished and/or thinned
  - P- (lightly) doped substrate
    - TSMC-like foundry
    - P+ doped substrate (Intel, SMIC?) scatters light too much
- Does not work for chips in plastic packages
  - Manufacturer must "design for inspectability"

How it Works:
Silicon is Transparent to Infrared Light

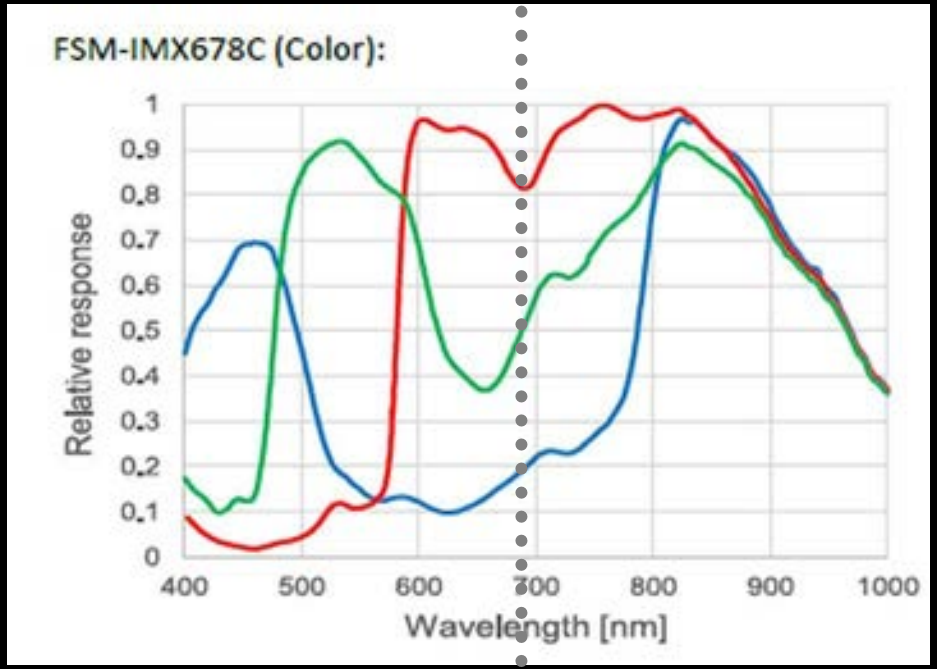


3803

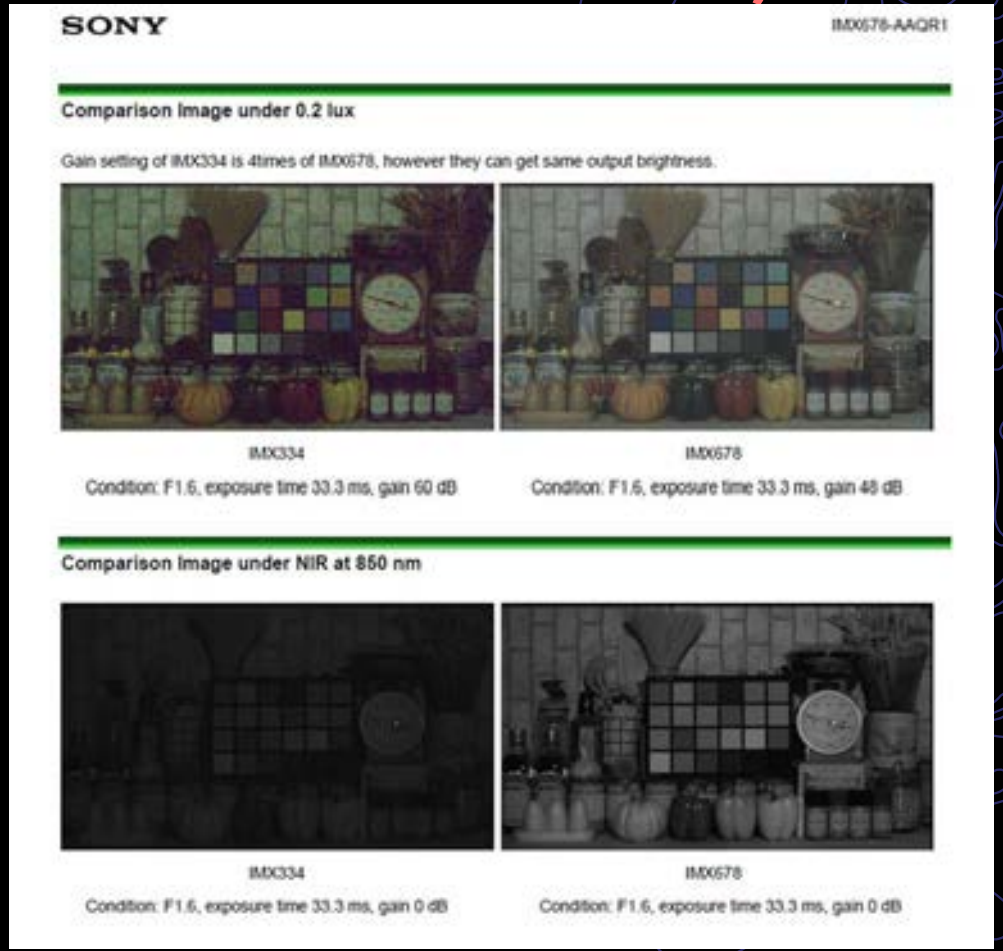
Silicon is Transparent to Infrared Light



Some CMOS Cameras are Sensitive to IR (e.g.: Sony Starvis2 → Surveillance Market)

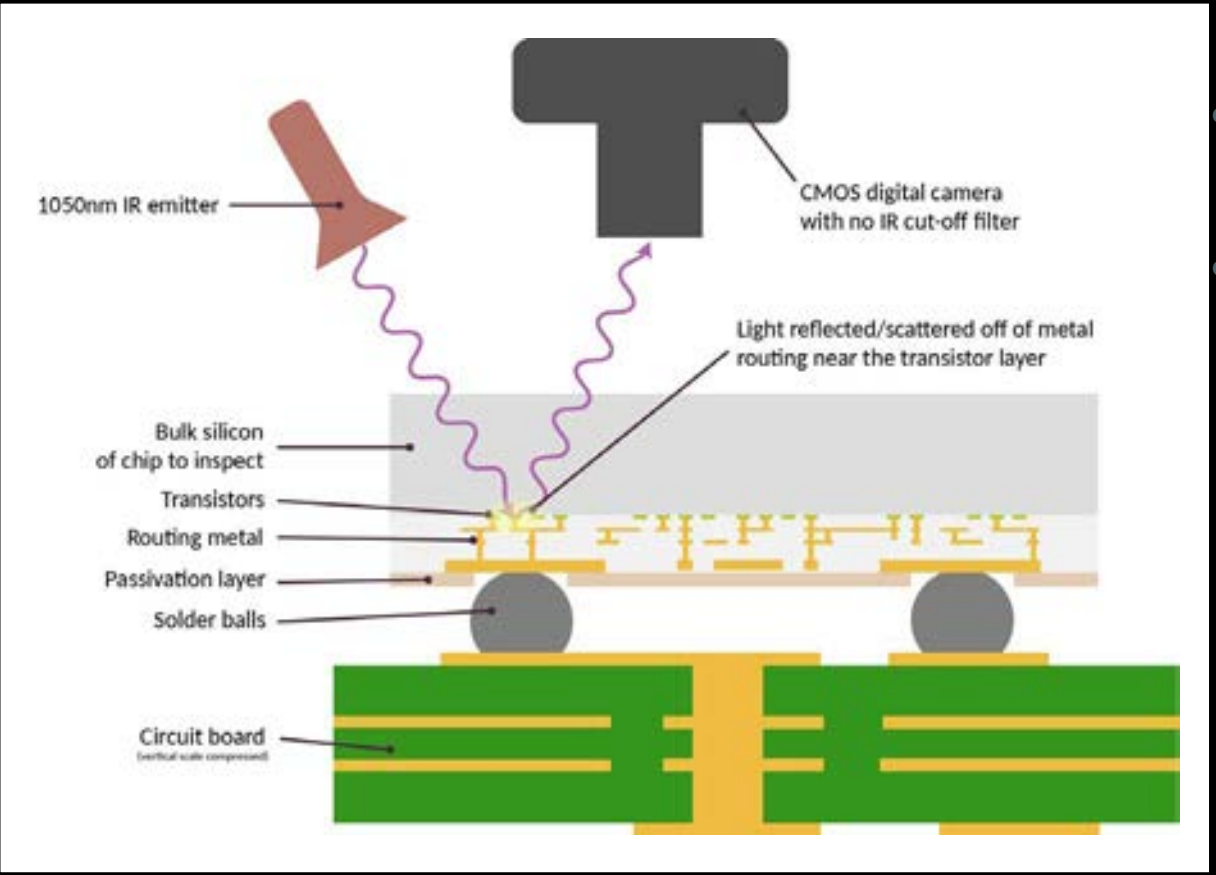


visible infrared



3803

Putting it All Together: IRIS



- Inspection of chips from the back side
- After they have been assembled into a product

Prior Work

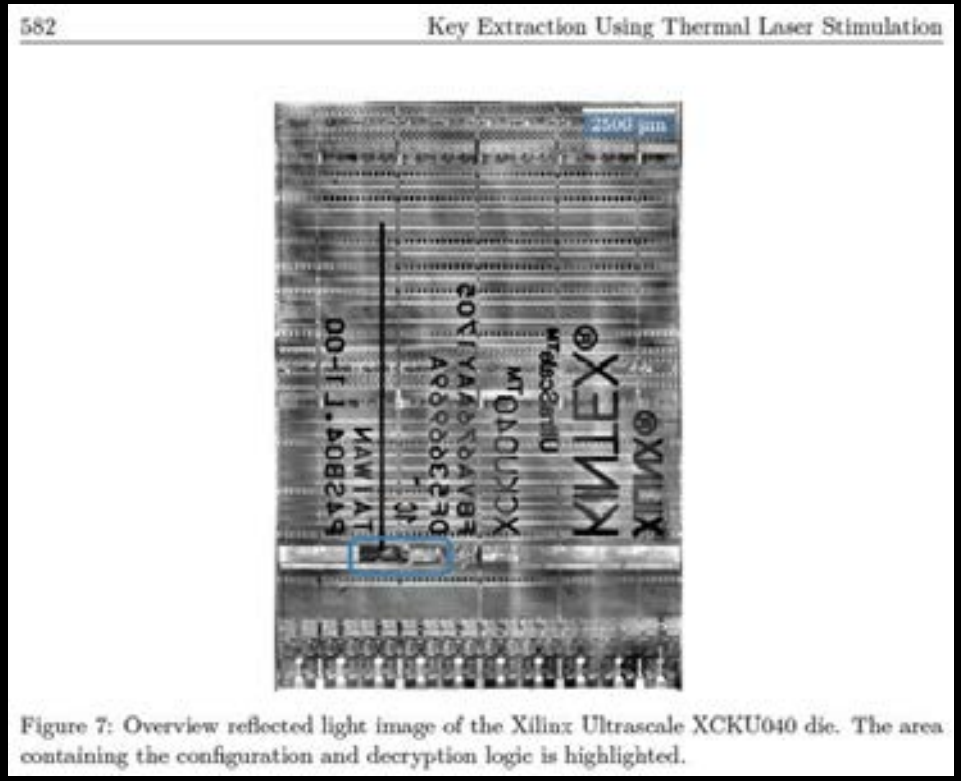
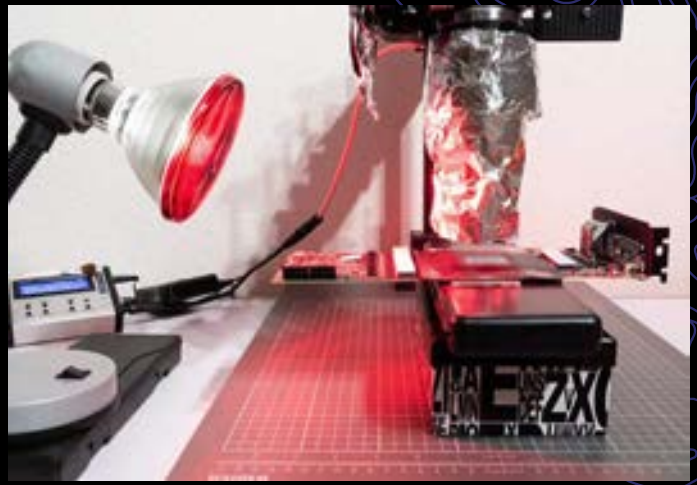
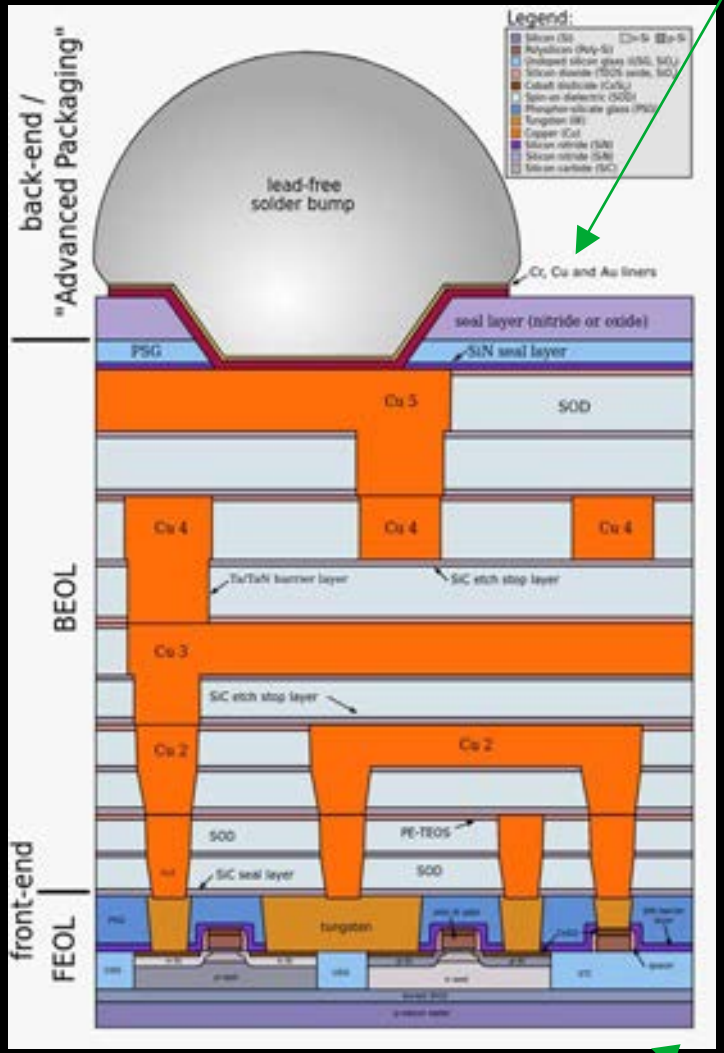

582 Key Extraction Using Thermal Laser Stimulation

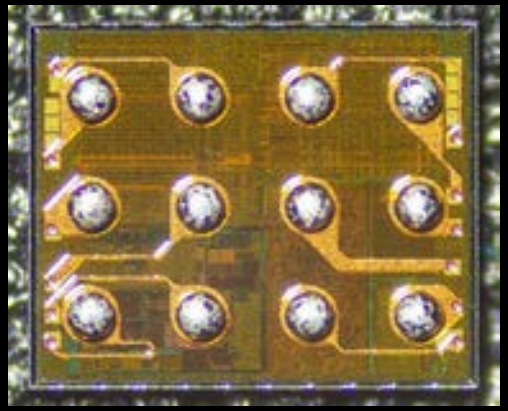
Figure 7: Overview reflected light image of the Xilinx Ultrascale XCKU040 die. The area containing the configuration and decryption logic is highlighted.

- "Key Extraction Using Thermal Laser Stimulation"
  - Lohrke et al CHES 2018 (via Dimitry Nedospasov)
  - Hamamatsu Phemos-1000
- Fritzchens Fritz flickr feed
  - Backside IR imaging with CMOS camera



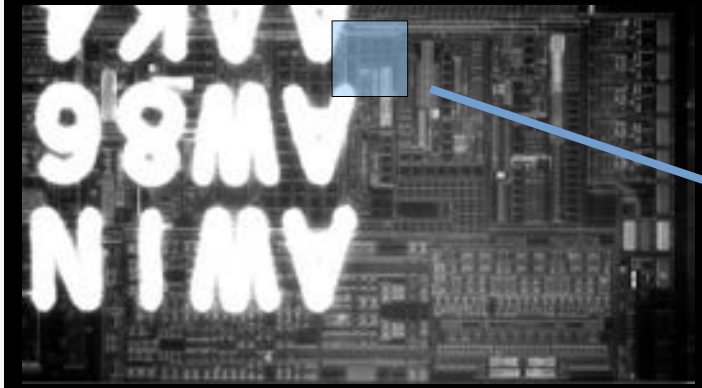


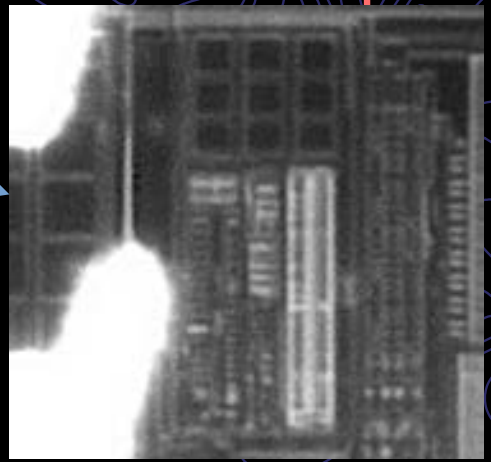
FRONTSIDE



Why Backside?

- The backside metal is closest to the transistors
- Topside metal tends to be just regular arrays for power distribution + pads





BACKSIDE

IRIS Implementations

<EUR300
fully manual adjustments

~EUR5000, fully automatic adjustments

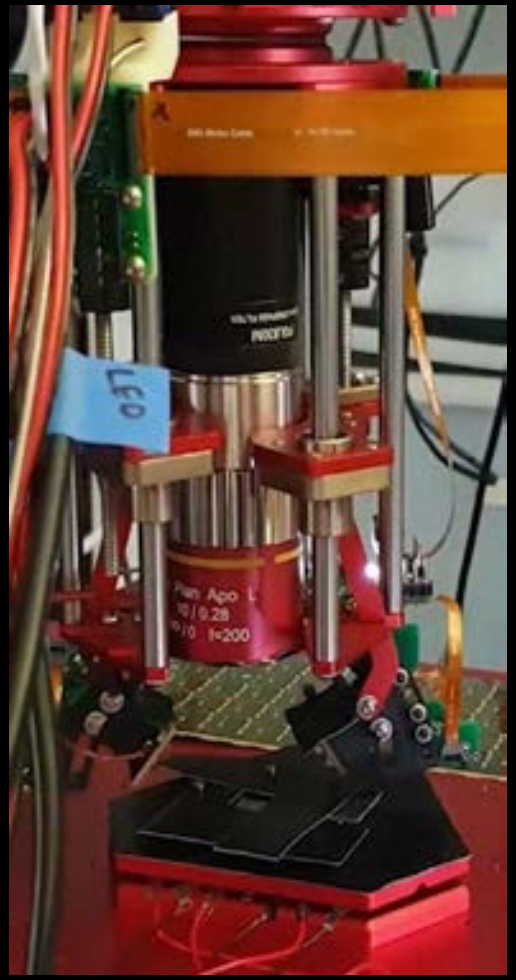
3803

Manual Adjustment

- Possible to generate high quality images
- Fussy to set up
- Repeatability issues

- Useful for end-user verification setups
  - Lower cost
  - More effort, but used rarely - only when new chips are acquired

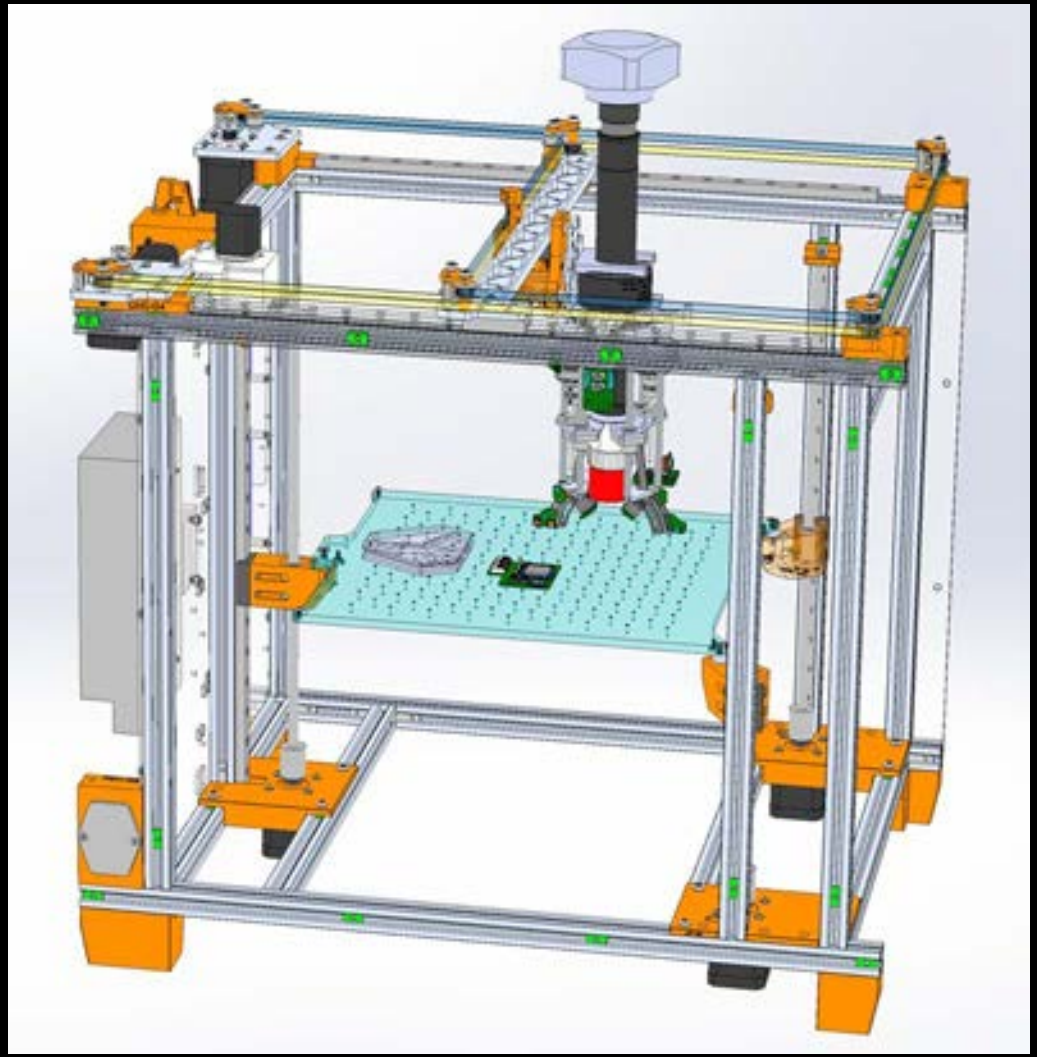
3803



Automated Adjustment

- <10 micron precision repeatability
- Fully automated X/Y/Z positioning
- Fully automated light positioning
- Good repeatability
- Useful for
  - Generating reference images
    - Higher quality images used as comparison point for end users
  - Higher throughput screening
  - Higher confidence measurements

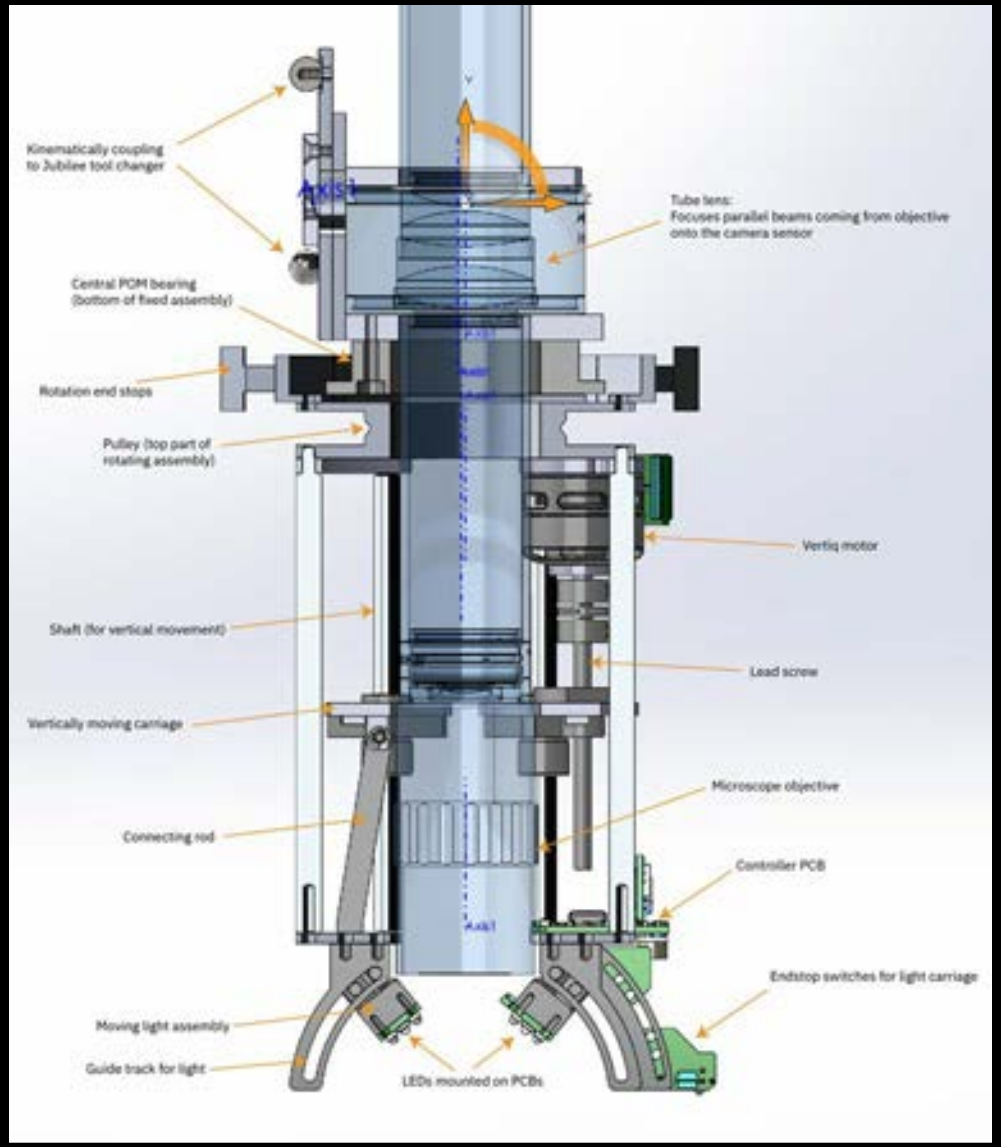
3803

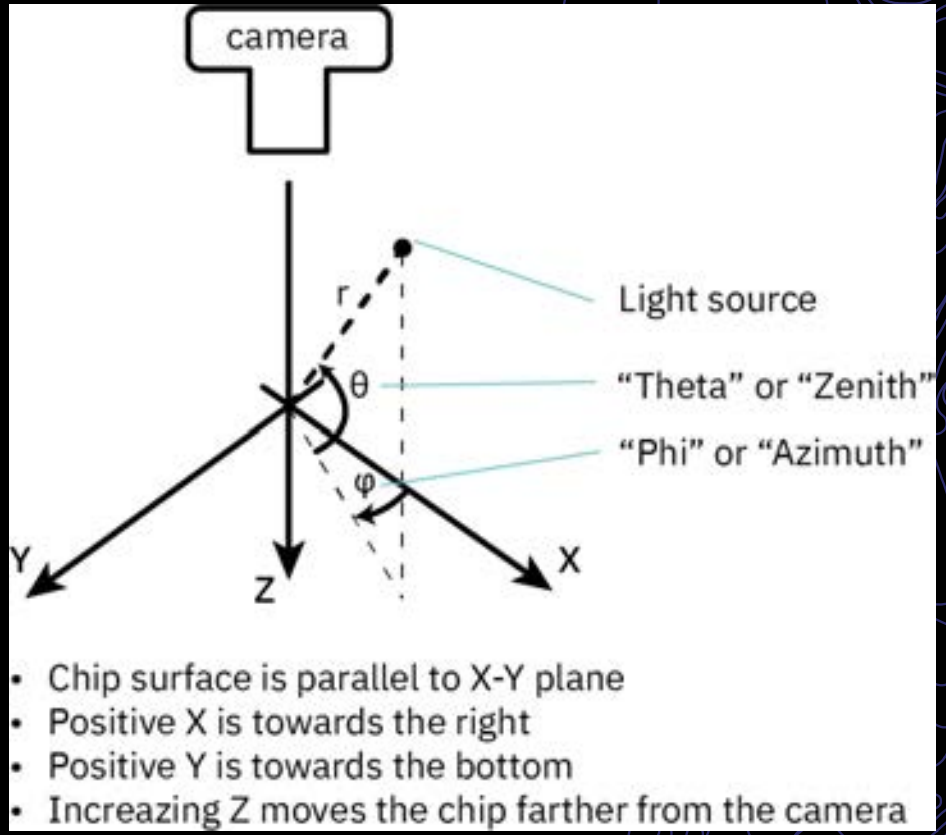


Automated Platform: Jubilee

- Developed by Prof Nadya Peek's laboratory
- Open source, 3D motion platform
- Kinematically coupled Z-stage
- https://machineagency.github.io/science_jubilee/

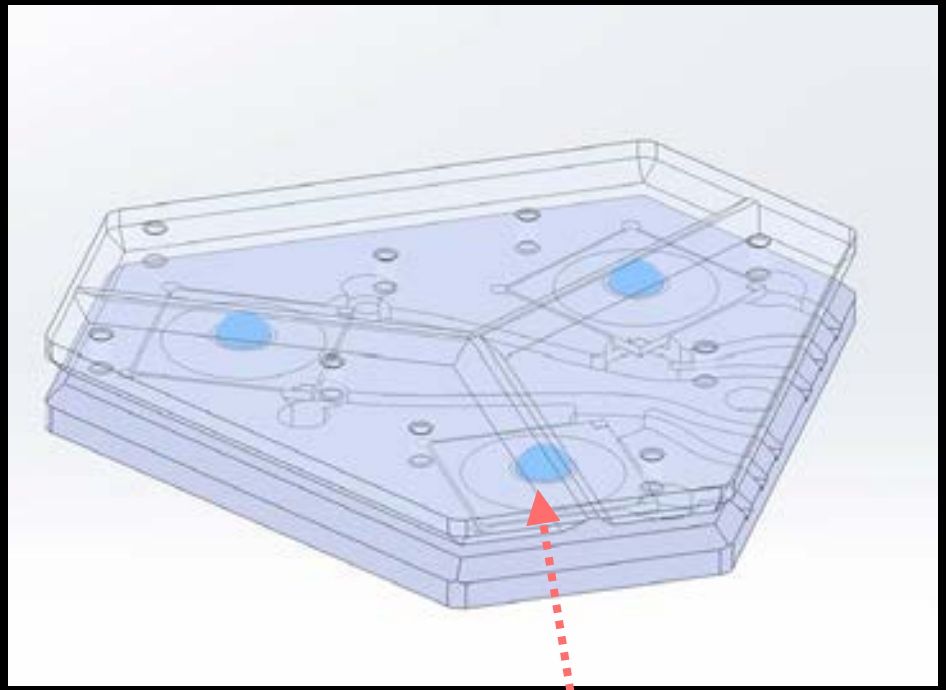
Microscope Core





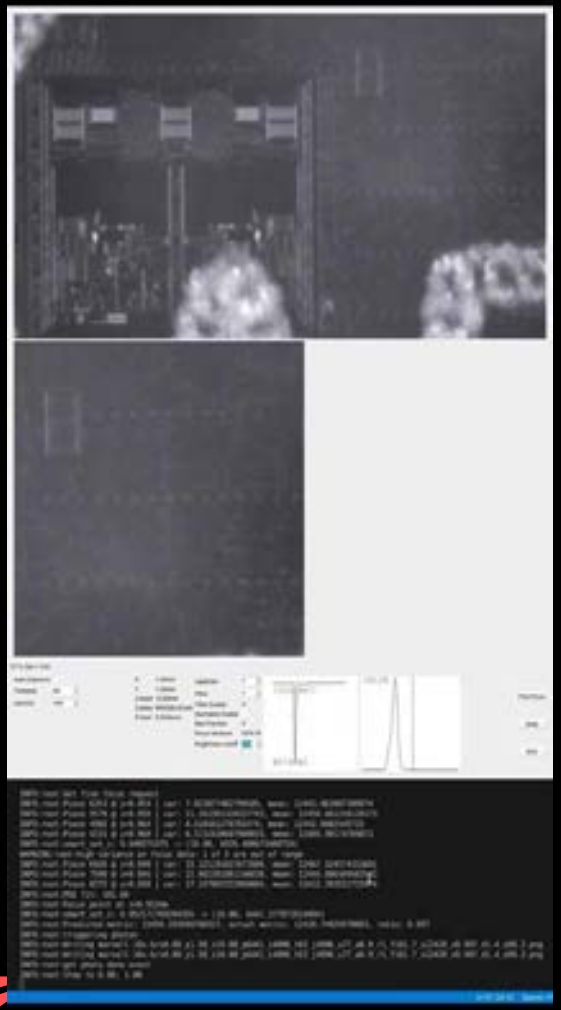
Chip Features vs. Angle of Incident Light

Nanometer-Precision Fine Focus Stage For <$200



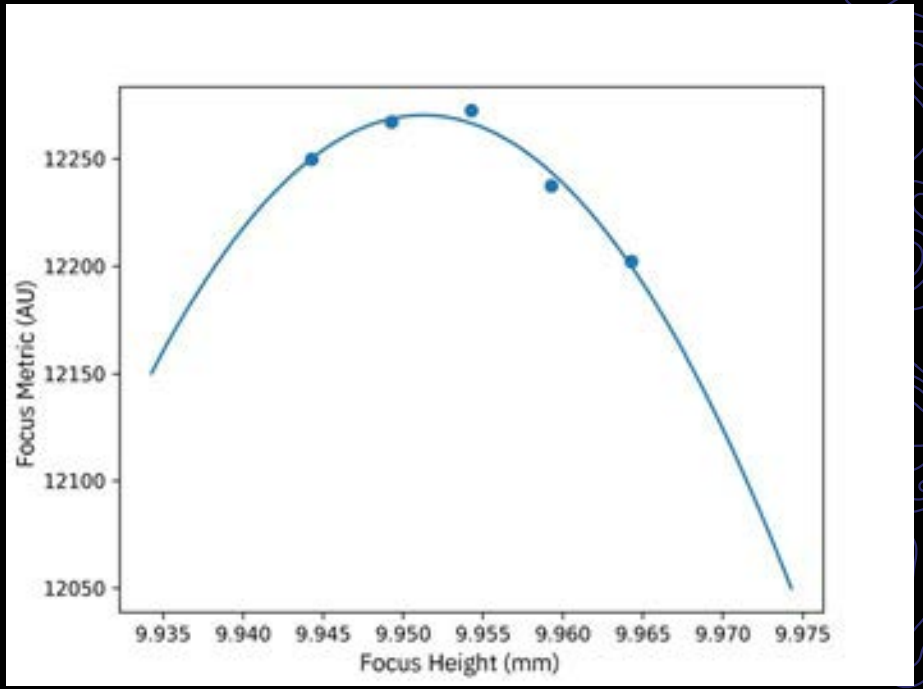
- 3x piezo actuators
  - Originally used for automotive haptics
- Kinematic coupling
  - Glass hemisphere into V-groove
  - "Exactly constrained" design
  - Sub-micron repeatability of stage removal

3803

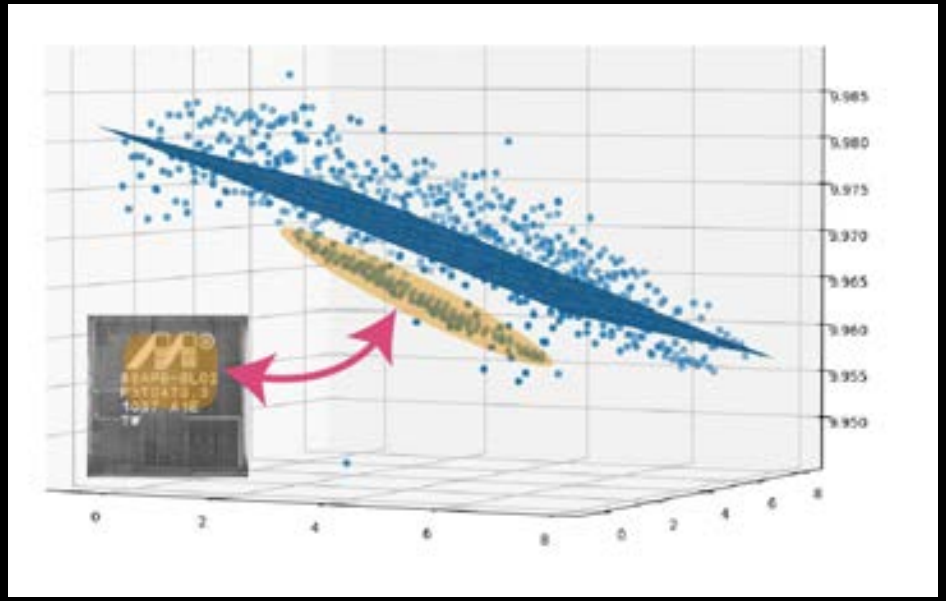


Imaging Software

- Autofocus
- Auto step and repeat
- https://github.com/bunnie/jubiris

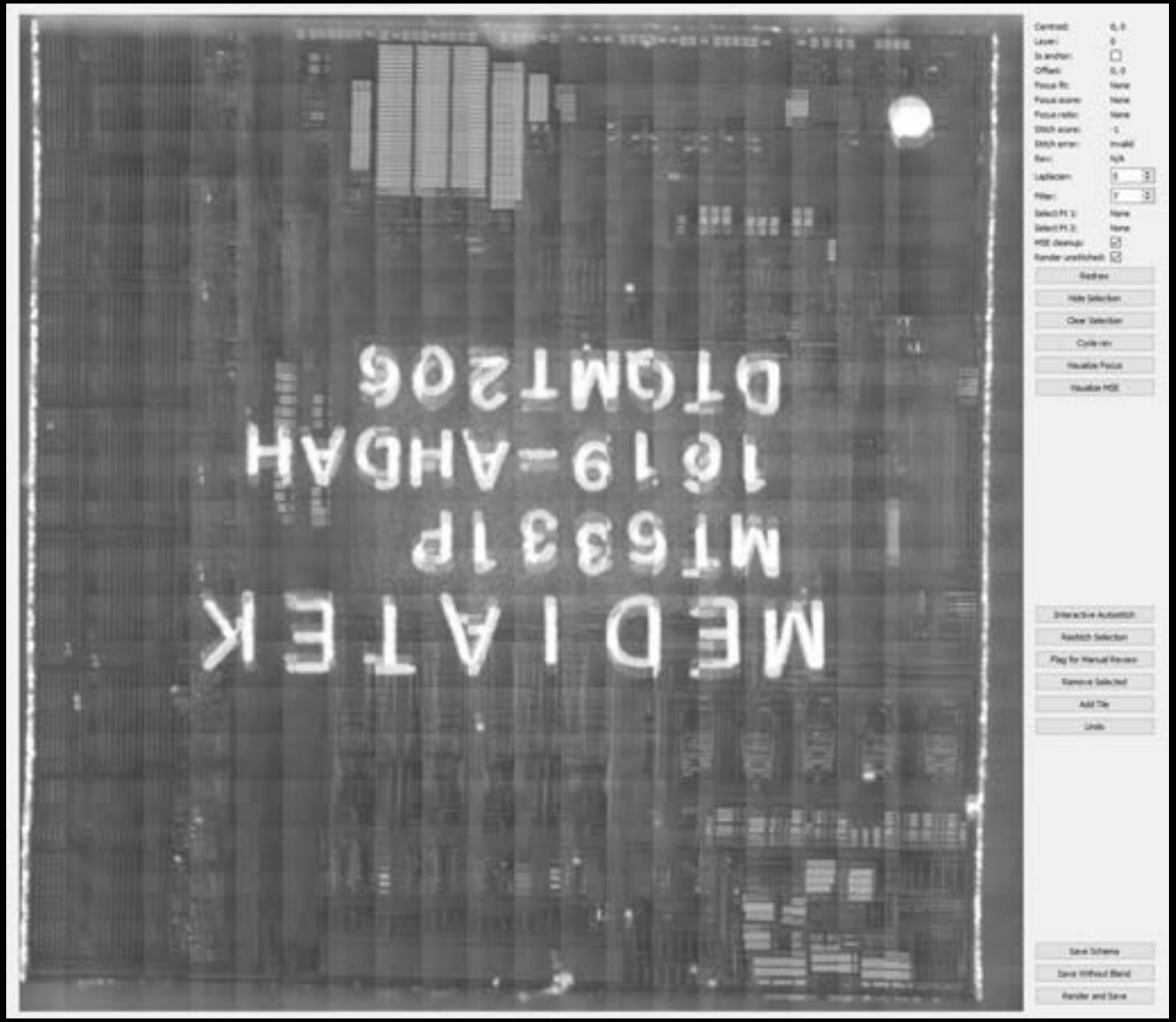


Focus Convergence



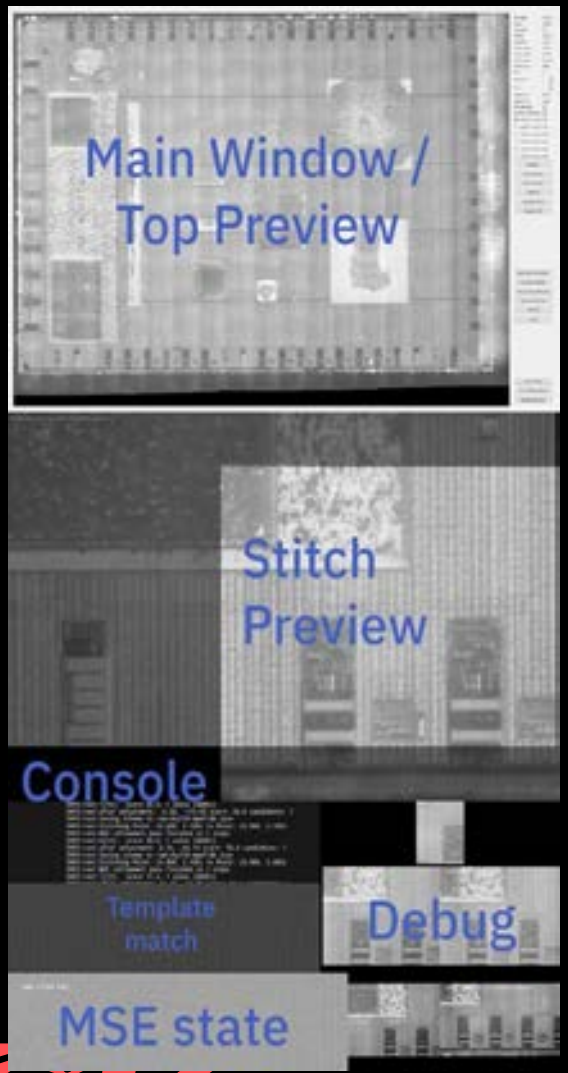
- Better than +/-5um on average versus ideal plane
  - Within depth of field limit for 10x objective
- <10 seconds per image step
  - Depends heavily on environmental vibrations
- Some divergence due to top markings

3803



Semi-Automated Image Stitching

- Sampled images do not perfectly align due to machine tolerances

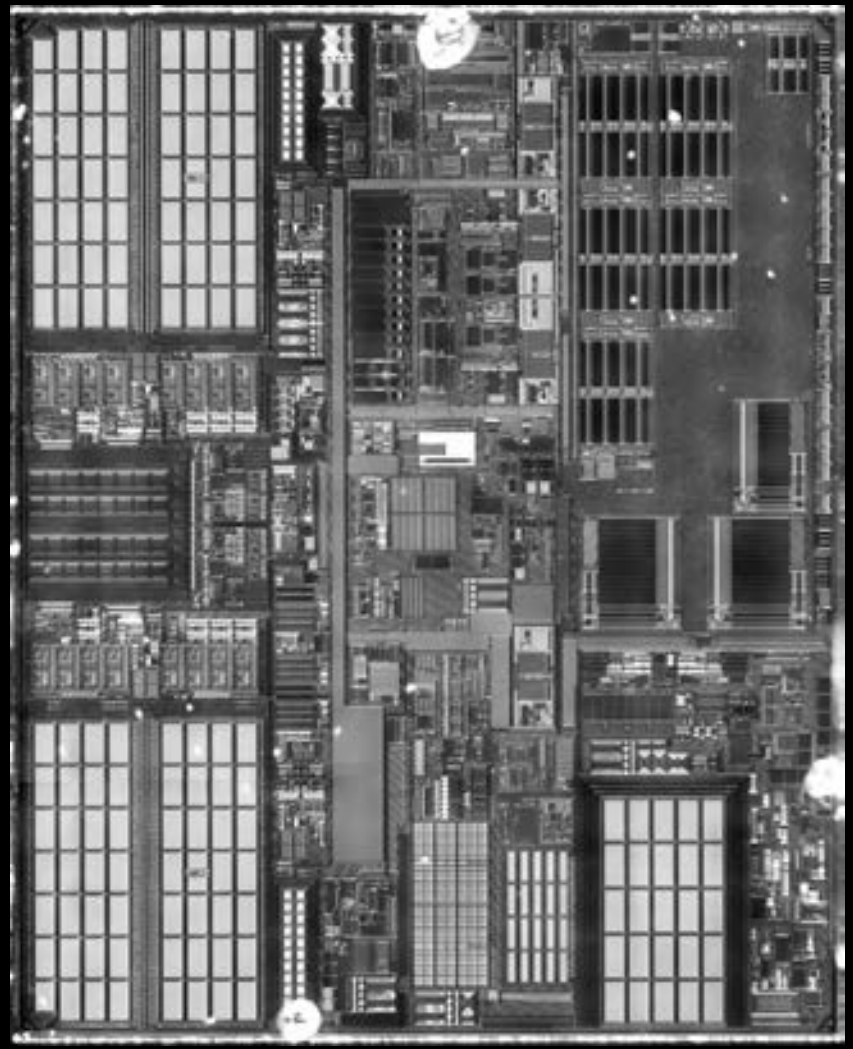


Template Stitch + MSE Cleanup

- Stitch ~200 images in about 10-15 minutes
- Some manual cleanup needed

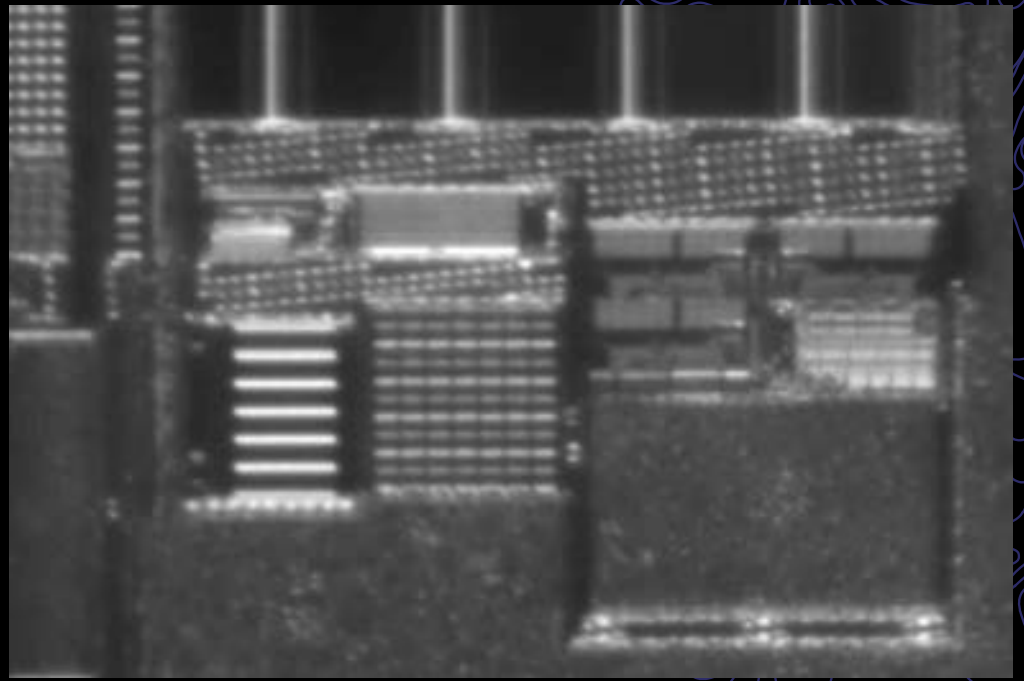
- Everything in IRIS in open source and documented with blog posts

https://bunnie.org/iris



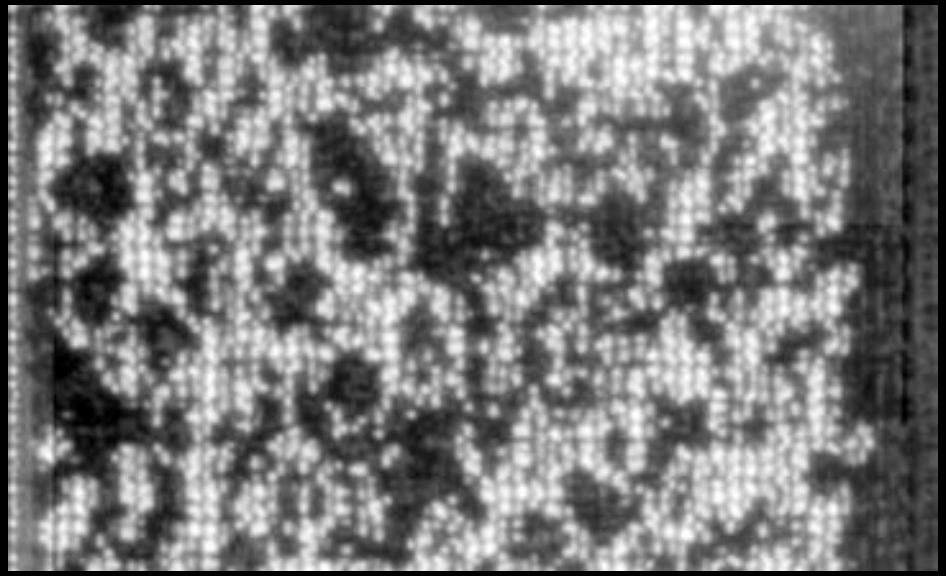
IRIS Examples

- https://siliconpr0n.org/archive/doku.php?id=tag:collection_bunnie&do=showtag&tag=collection_bunnie

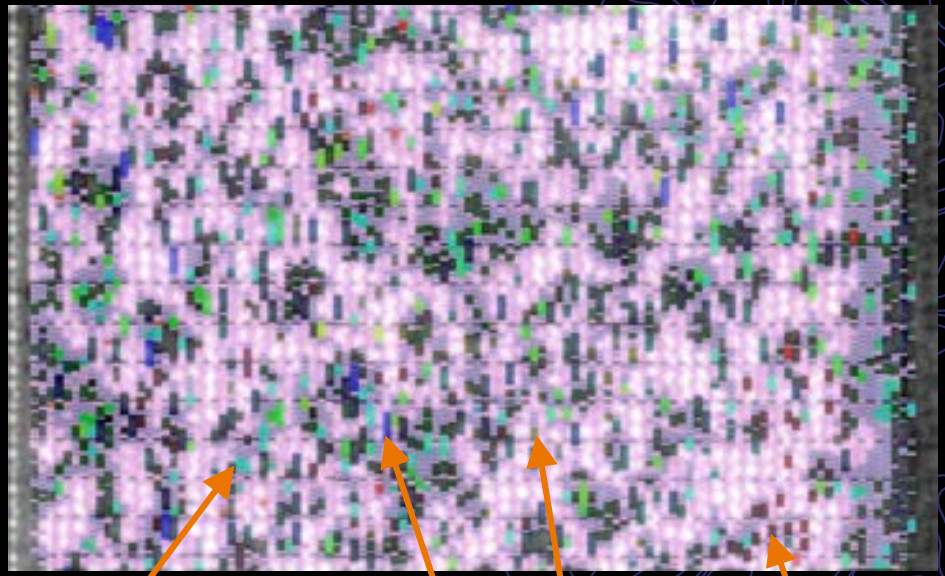


3803

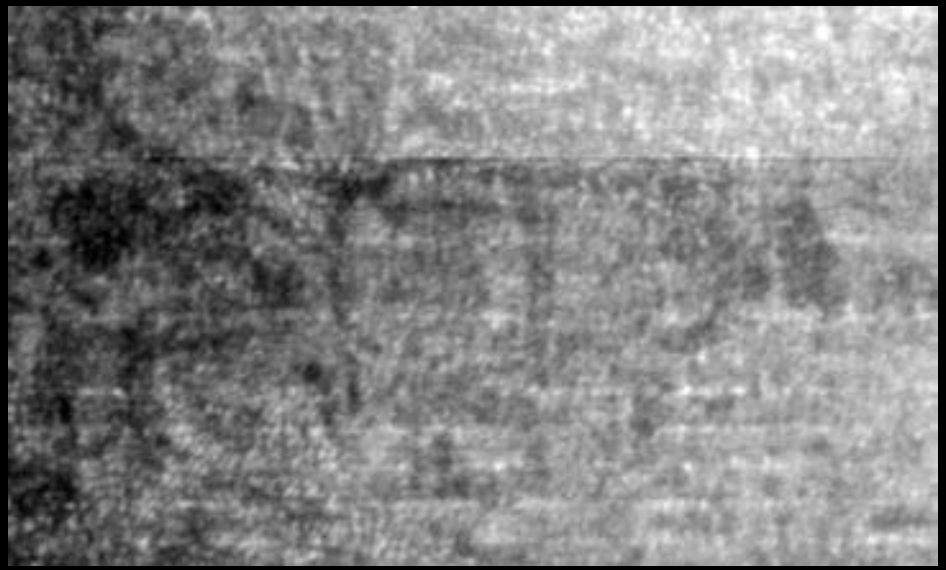
IRIS Examples: Seeing Standard Cells



SKY130 process



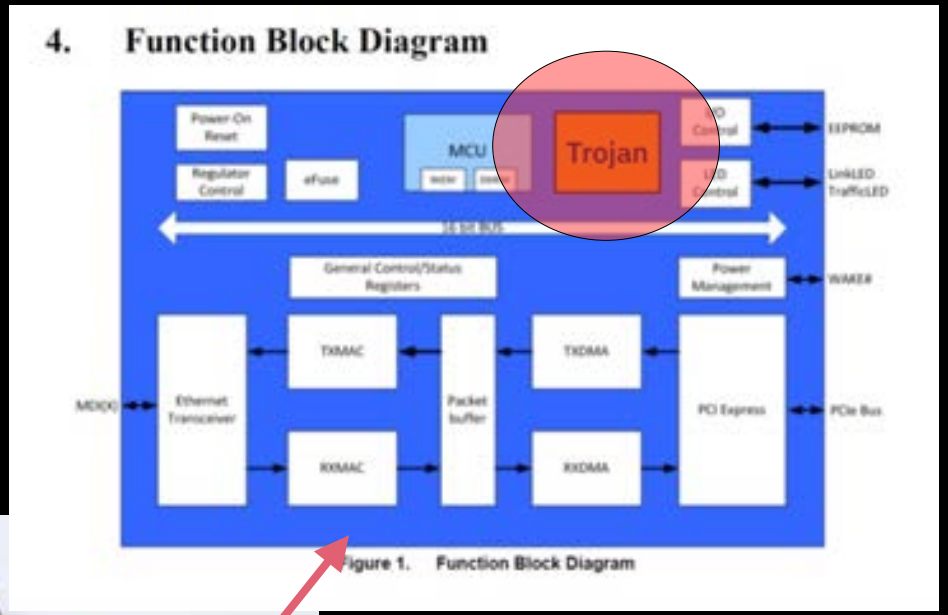
More Standard Cells



TSMC 22nm process, same scale as SKY130 on previous slide

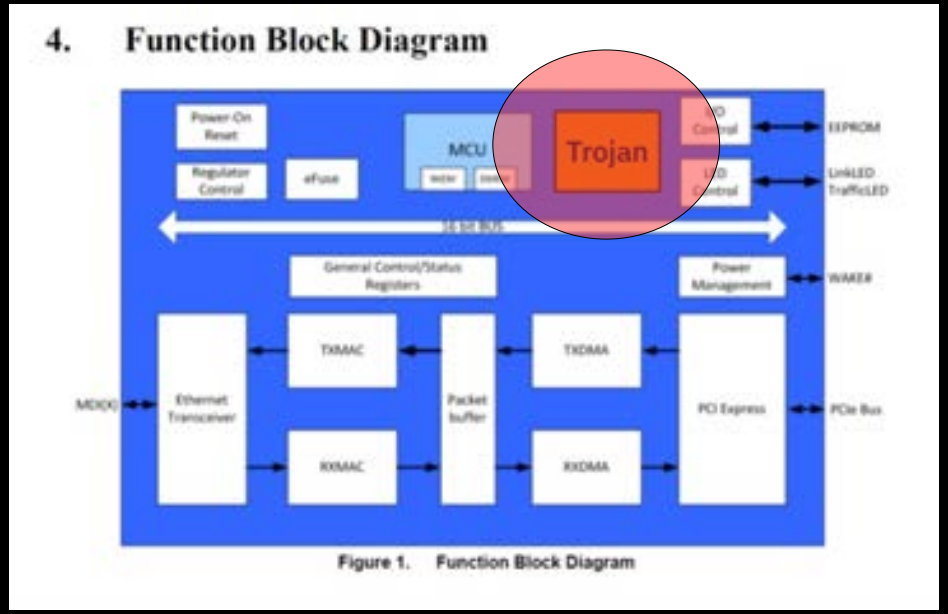
So, What Does IRIS Get Us?

Level 1: Block-Level Modification



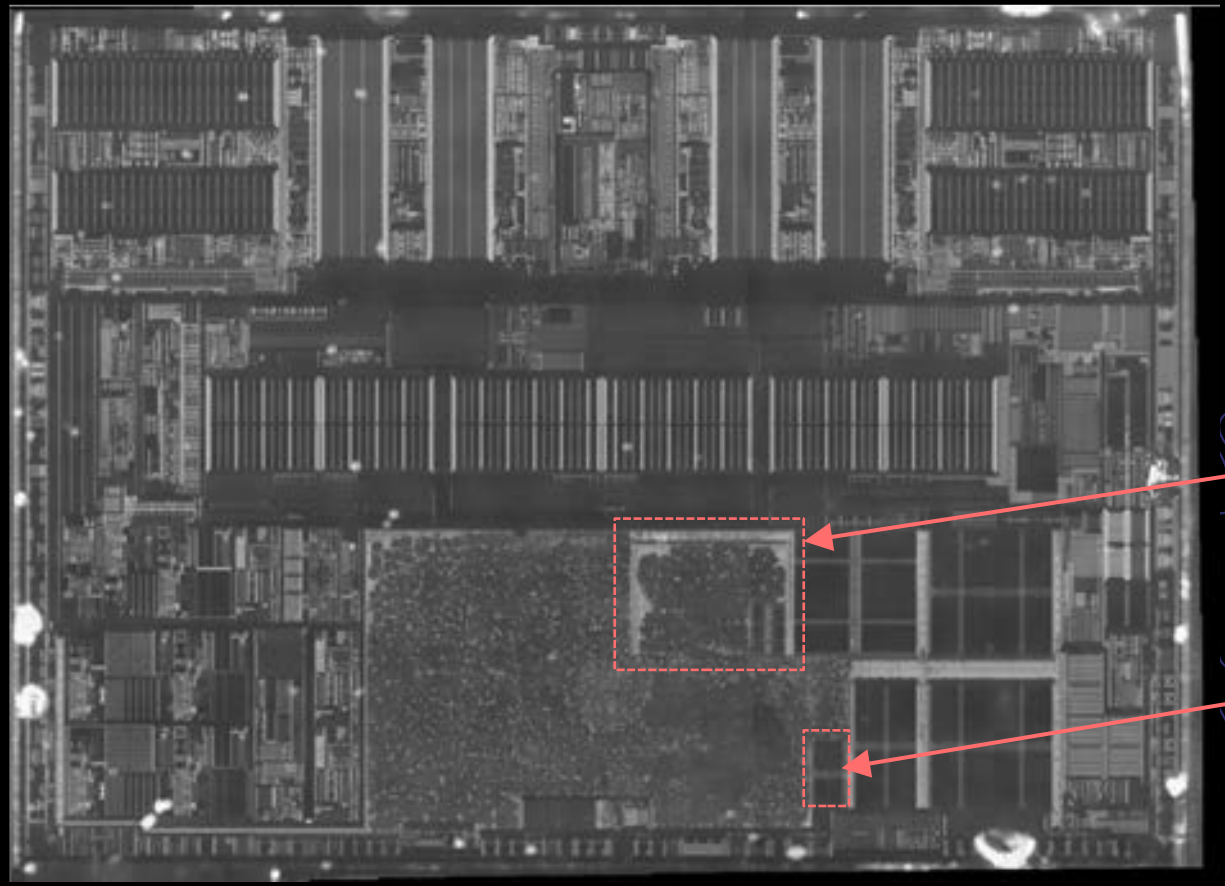
- If chip in WLCSP package:
  - Easy to "diff out" block-level modifications
  - Would need reference images, possibly crowd-sourced

Grounding a Hypothetical Trojan



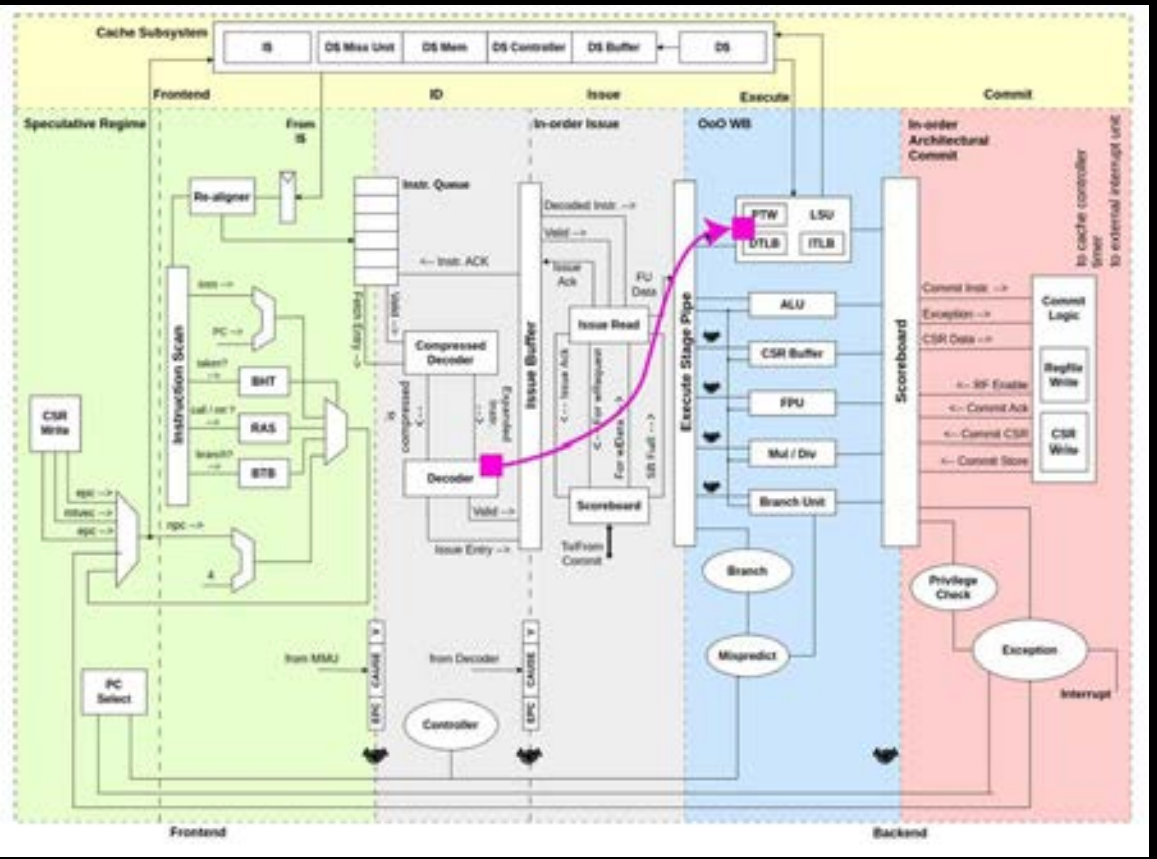
- Hypothetical "Trojan":
  - Records ~few kiB of network traffic
  - Has a trigger
    - Say, respond to ICMP secret knock to exfiltrate data

Example of Block Sizes



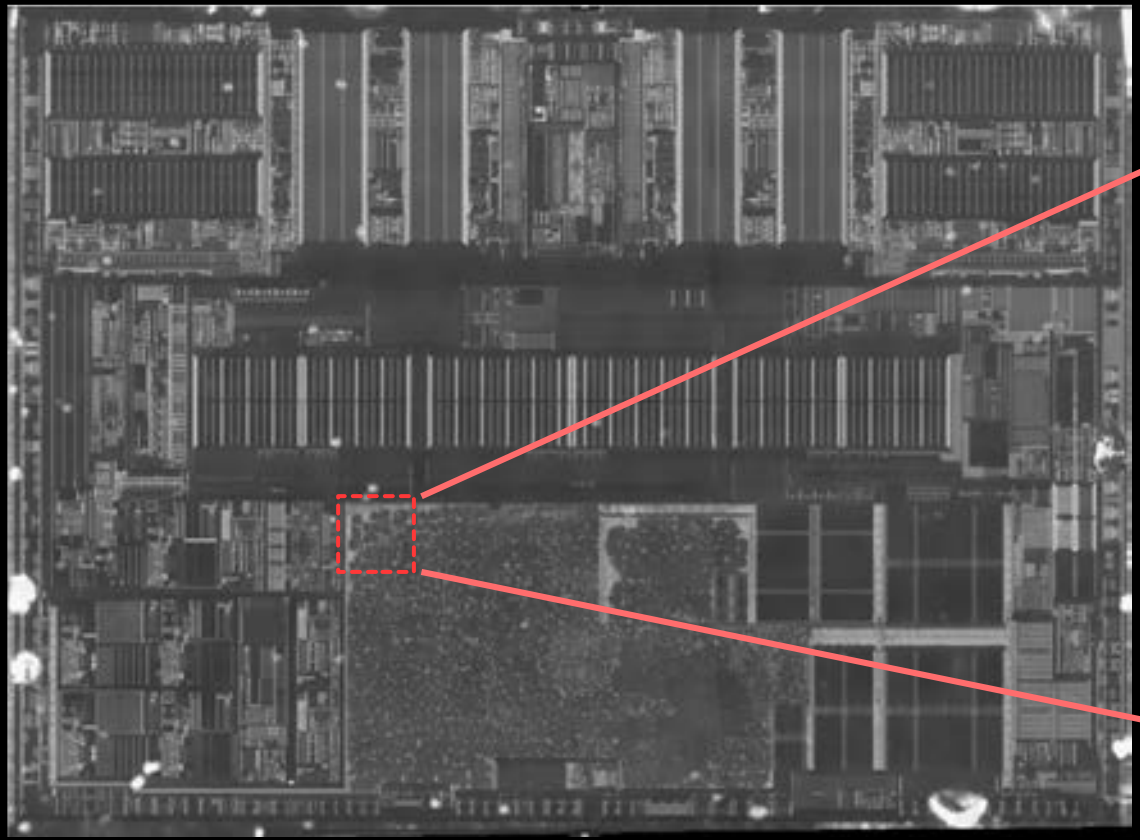
Estimated @ 65nm node

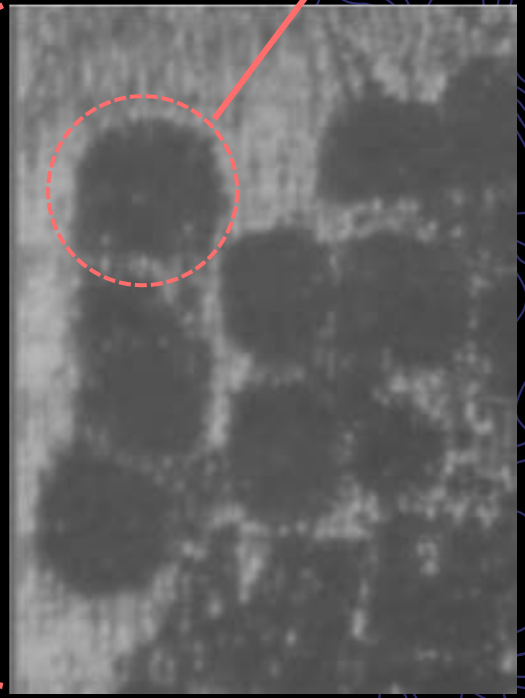
Level 2: Small RTL Modifications



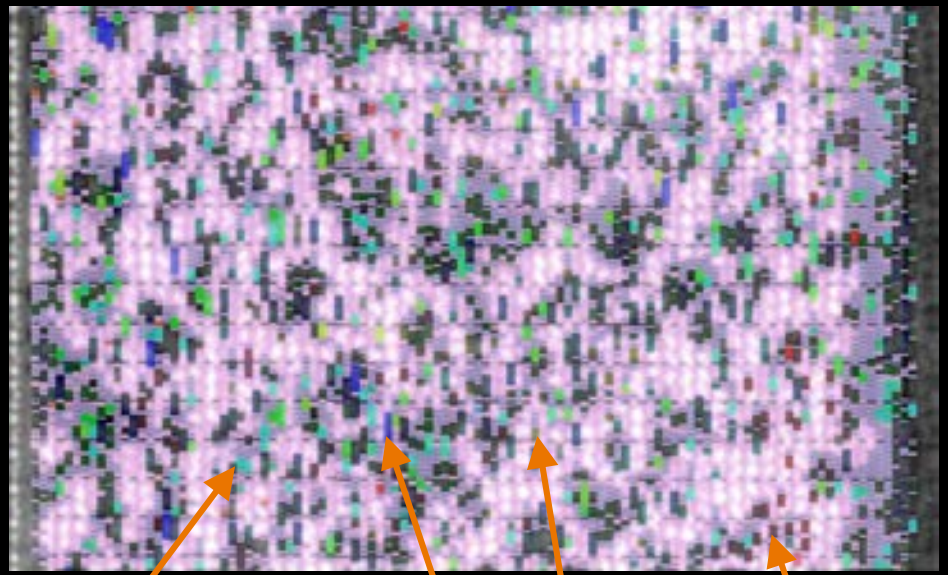
- "Probably detectable"
  - Naive RTL insertion would have place/route deviations
  - Recall from earlier discussion:
    - O(10)-O(100) cells added

Example of Place & Route Logic Patterns



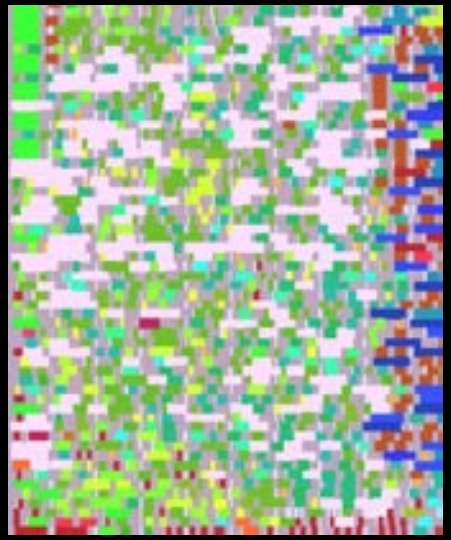


Limitations of Comparing IRIS Images



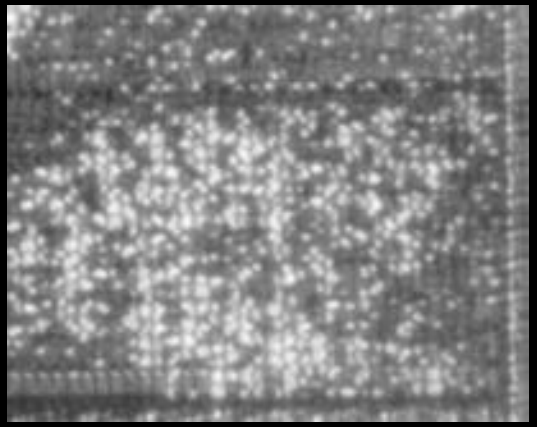
- Logic gates show up as fuzzy blobs "by type of gate"
  - In reality we can only know "how many gates"
  - "Exactly what gates" may be spoofable
- An omnipotent adversary could "lock down" place/route paths to maintain net shape, logic cell types
  - Would leave some trace, e.g. reduced timing margin, power consumption changes

Related Work in Progress: Automated Gate Count Census

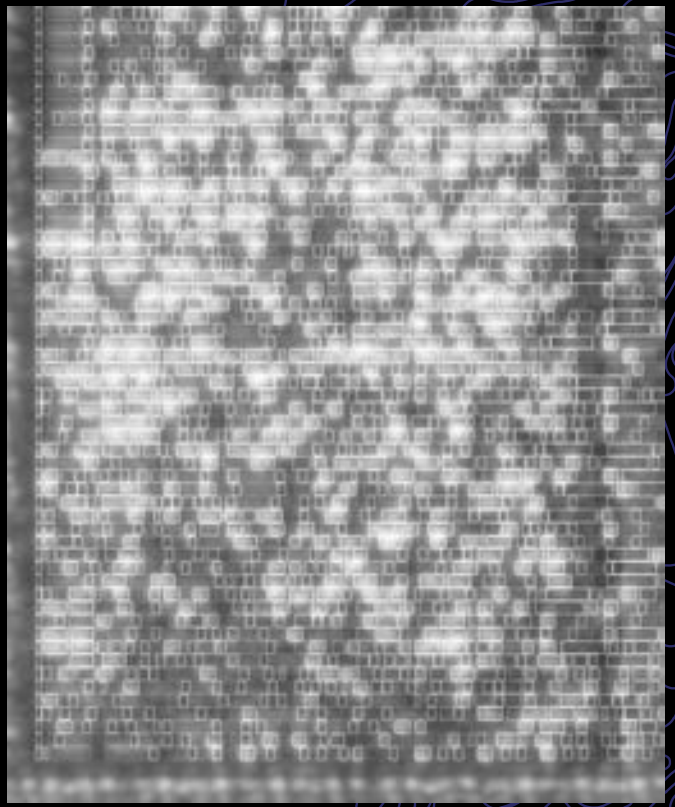


Design data
(standard cell map)

+



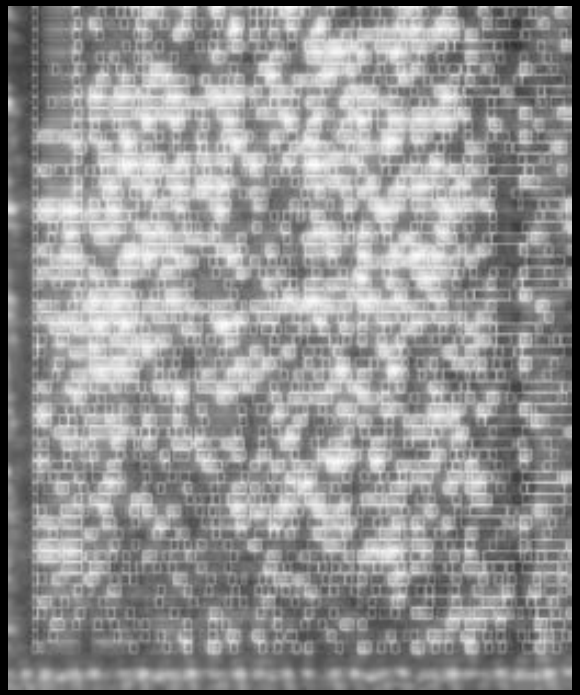
Imaging data
(arbitrary rotation & translation)

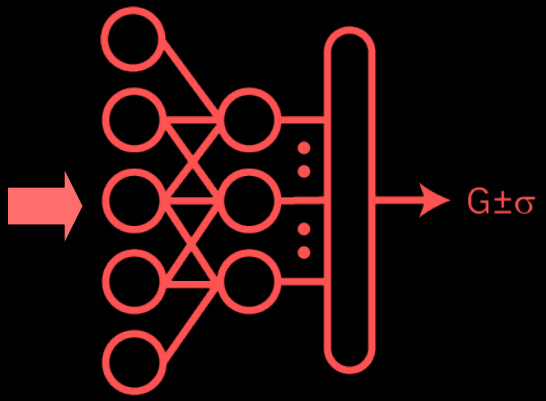


Aligned cell-to-image map

3803

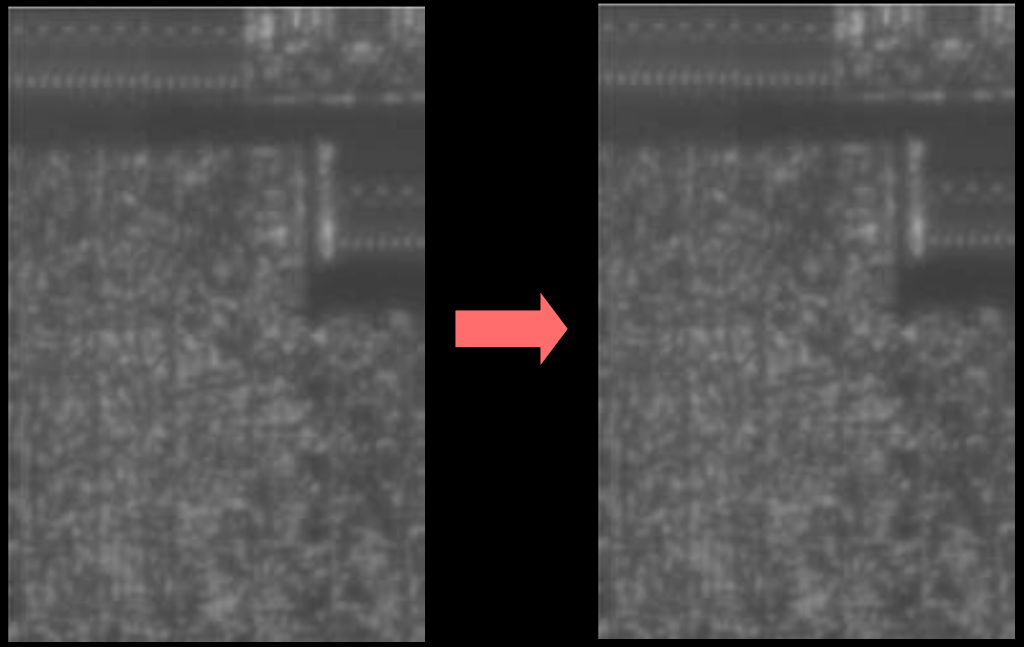
Quantifying Gate Counts





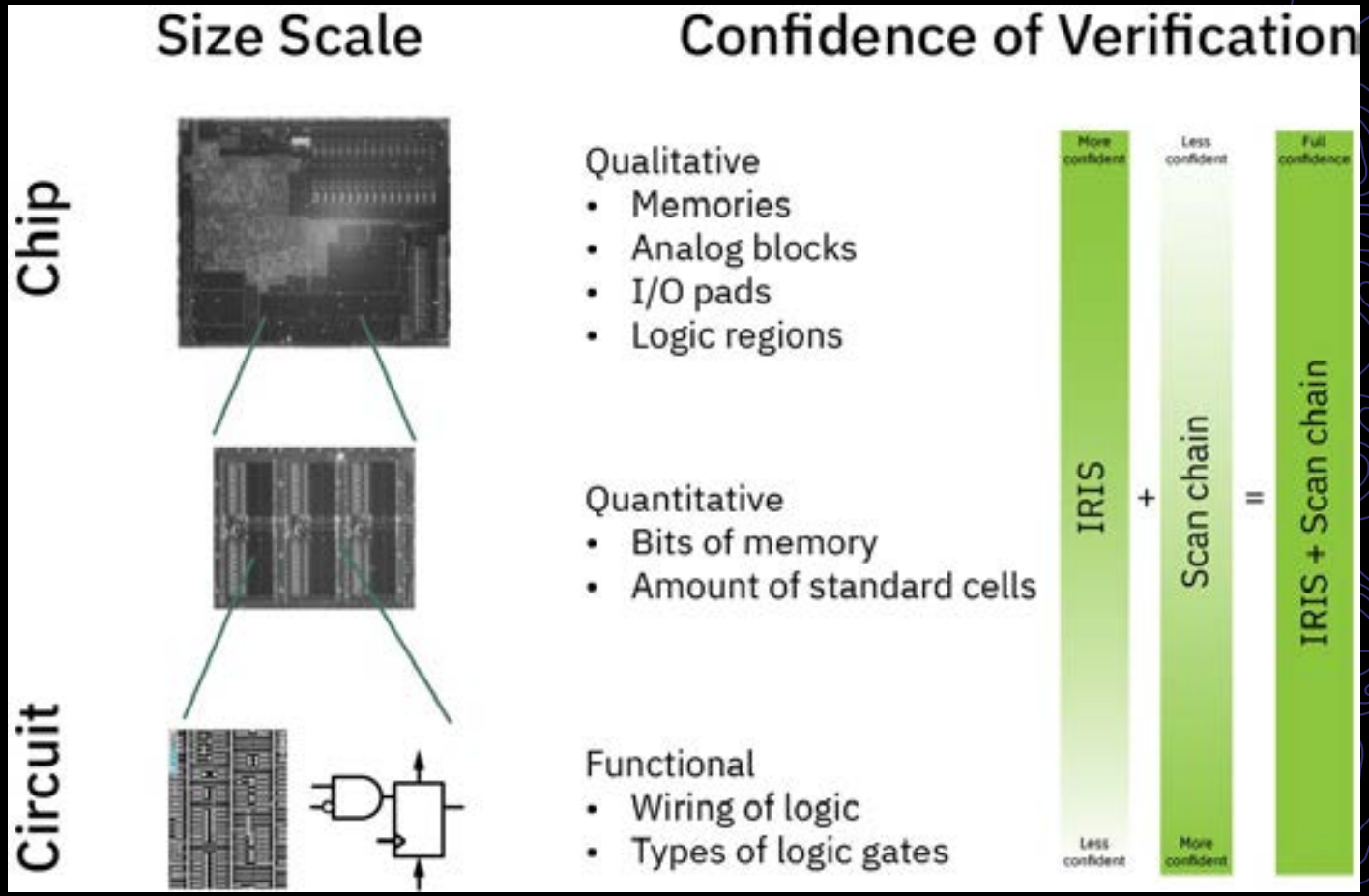
- Trying to train a CNN classifier to estimate gate count
  - "G" plus/minus an uncertainty of "sigma"
    - Uncertainty due to noise, dirt, scratches, process variations...
  - Bonus if it can classify types of logic cells

Level 3: Targeted Mask Modifications



- No difference in images, by attacker's intention
  - Modifications solely on mid-level metal layers
  - No extra logic gates, but functionality is changed
    - "Spare cells" possibly used for malicious purposes

Next Steps: Hybrid Verification



Even If We Can't Get to 100% Confidence: IRIS is Better than Just Trusting The Label

>>

~100MiB image of chip

~64 bytes of text labeling

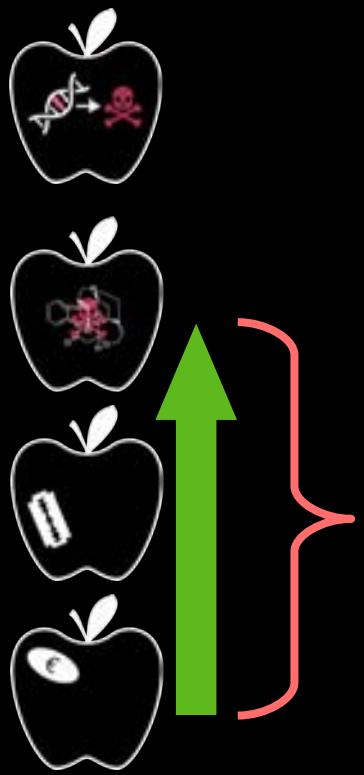
Status Quo: "Just Trust the Label"

Nobody is checking

A few people are checking

- Threat actors operate in a zero-risk, zero-consequences scenario

Infra Red, *in situ* (IRIS): Improving Trust For Everyday People



Still some things we can't catch, but...

...IRIS could raise the bar

- Even modest IRIS adoption may deter threat actors
  - "Easier" hardware Trojans are no longer a zero-risk proposition
- Ideally, products are designed to facilitate inspection
  - This only happens if there is demand for inspectable products
- Also, it's just fun to look inside chips!

Demo / Q&A

@bunnie@treehouse.systems
@bunniestudios.bsky.social

With thanks to:

nlnet FOUNDATION

NGI ZERO ENTRUST

Github sponsors:

3803

INFRARED IN SITU

https://bunnie.org/iris

Self organized session
Day 2 14:00-16:00
“Microscope Nerds and IC Die Photography Meetup”
fail0verflow (I6/H3Foyer) for a pointer