# Extraction of Secrets from 40nm CMOS Gate Dielectric Breakdown Antifuses

by FIB Passive Voltage Contrast

Andrew D. Zonenberg

@azonenberg@ioc.exchange

Antony Moor, Daniel Slone, Lain Agan, Mario Cop

IOActive®

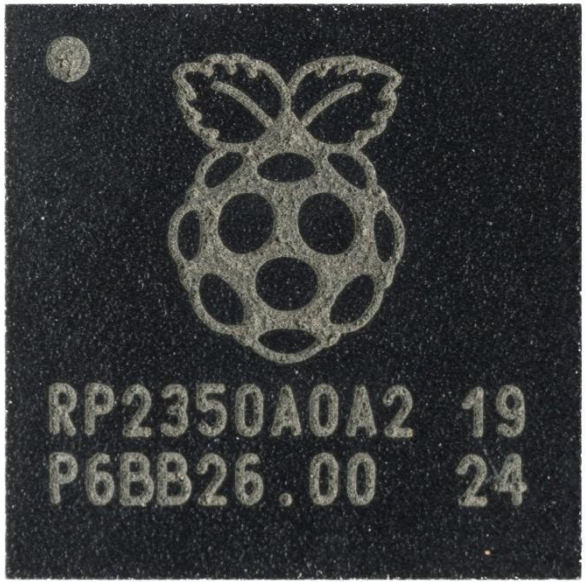
# IOActive Presentation Content

## Legal Notices

- **Disclaimer Notification**
  The views, opinions, findings, conclusions, positions, and/or recommendations expressed herein are those of the authors individually and do not necessarily reflect the views, opinions, or positions of IOActive, Inc.
- **No Warranties or Representations**
  The information presented herein is provided “AS IS” and IOActive disclaims all warranties whatsoever, whether express or implied. Further, IOActive does not endorse, guarantee, or approve, and assumes no responsibility for nor makes any representations regarding the content, accuracy, reliability, timeliness, or completeness of the information presented. Users of the information contained herein assume all liability from such use.
- **Publicly Available Material**
  All non-IOActive source material referenced in this presentation was obtained from the Internet without restriction on use.
- **Fair Use**
  This primary purpose of this presentation is to educate and inform. It may contain copyrighted material, the use of which has not always been specifically authorized by the copyright owner. We are making such material available in our efforts to advance understanding of cyber safety and security. This material is distributed without profit for the purposes of criticism, comment, news reporting, teaching, scholarship, education, and research, and constitutes fair use as provided for in section 107 of the Copyright Act of 1976.
- **Trademarks**
  IOActive, the IOActive logo and the hackBOT logo are trademarks and/or registered trademarks of IOActive, Inc. in the United States and other countries. All other trademarks, product names, logos, and brands are the property of their respective owners and are used for identification purposes only.
- **No Endorsement or Commercial Relationship**
  The use or mention of a company, product or brand herein does not imply any endorsement by IOActive of that company, product, or brand, nor does it imply any endorsement by such company, product manufacturer, or brand owner of IOActive. Further, the use or mention of a company, product, or brand herein does not imply that any commercial relationship has existed, currently exists, or will exist between IOActive and such company, product manufacturer, or brand owner.
- **Copyright**
  

# So… what's the RP2350?

- Dual core MCU
  - Switchable M33 or RV32!
- 32 kB boot ROM
- 520 kB SRAM
- 8 kB ECC OTP (or 12 kB non-ECC)
- No internal flash
  - Boots from external QSPI flash
  - Supports signed images or nonsecure XIP



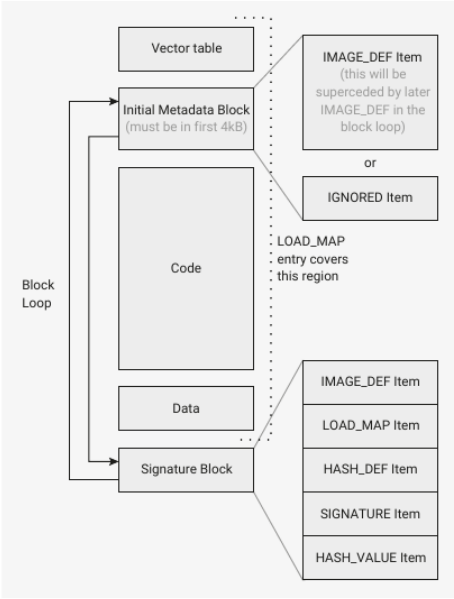





IOActive.

# Secure boot flow

- TL;dr: copy image from QSPI to SRAM
- Check signature
  - Pubkey + sig in image header
  - Hash of key burned into fuses
- If sig valid, run it
- Image is cleartext
  - Encrypted boot not *directly* supported








IOActive.

# Encrypted boot flow

- Secure-boot a trusted, open source “stub”
- Stub copies main firmware blob to SRAM
- Can then decrypt using a key stored in fuses
  - Fuses support read protection so can’t dump via JTAG etc
  - Attacker who can read fuses could thus decrypt FW

# Let's do some OSINT

SYNOPSYS®

Solutions Products Support News & Views Company

Home / Synopsys IP / Foundation IP / NVM / Antifuse-Based Split-Channel 1T-Fuse Bit Cell for OTP NVM I

## Antifuse-Based Split-Channel 1T-Fuse Bit Cell for OTP NVM

At the heart of all Synopsys One-Time Programmable (OTP) Non-Volatile Memory (NVM) IP is the memory array, using a patented, area-efficient, antifuse bit cell – 1T-Fuse – that employs gate-oxide breakdown as a robust, non-reversible programming mechanism. This is a proven, reliable and secure technology that has been widely adopted and used in a broad range of applications and markets.

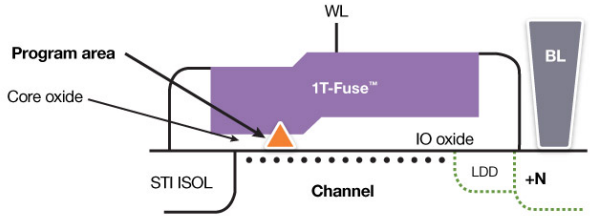



Figure 1: 1T-Fuse Bit Cell in Synopsys OTP NVM IP

The single-transistor OTP bit-cell (1T-Fuse) uses a unique split-channel architecture, where a single transistor gate covers both I/O (thick) and core (thin) oxides. The cell is programmed by a controlled, irreversible breakdown from the gate through the core (gate) oxide to the channel. The bit cell is programmed either with an embedded charge pump or through an external pin (at the tester, for example). The bit-cell is implemented using standard CMOS processes and requires no additional mask steps.

Gate oxide breakdown = antifuse

Cross section of memory bit cell

https://www.synopsys.com/dw/ipdir.php?ds=nvm_1t-bit-cell





IOActive.

# Let's do some OSINT

Arrays are available from a few 100 bits to around 1M bit and may be cascaded for even larger macros. Synopsys memory arrays are very flexible and can replace multi-time programmable memory in many applications. Conversion to mask ROM is also simple.

Available in standard CMOS processes, Synopsys OTP memory arrays don't require additional mask layers or process steps and provide a viable alternative to mask ROM, eFuses and Flash memory. The reduced size of the single-transistor bit cell results in better yield, higher security, improved reliability and lower overall cost.

**Security Features**

The inherent security features of the 1T-Fuse bit-cell include:

- Programming is by permanent structural change in few atomic layers (located far from diffusion)
- No physical attack can reveal programmed state in FinFET or HKMG technologies
- No leakage in non-programmed state
- State cannot be changed through exposure to high temperature, voltage or radiation
- No charge or voltage involved in state retention (unlike floating gate NVM, e.g. flash)
- State of memory (even for a few bits) is virtually impossible to detect using physical attack or reverse engineering techniques

That sounds like a challenge…

https://www.synopsys.com/dw/ipdir.php?ds=nvm_1t-bit-cell



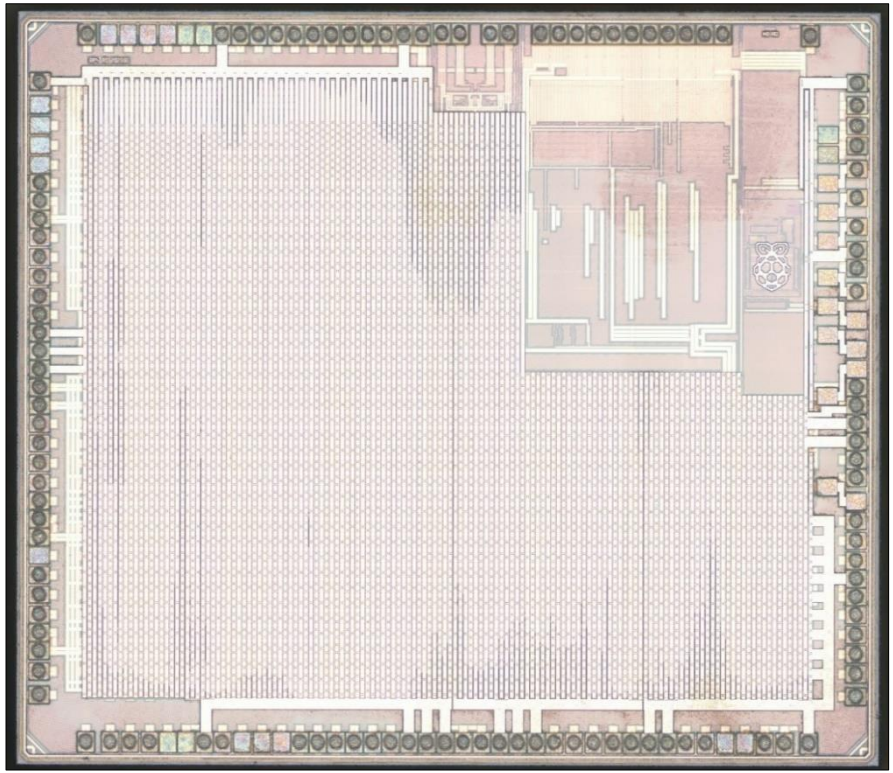


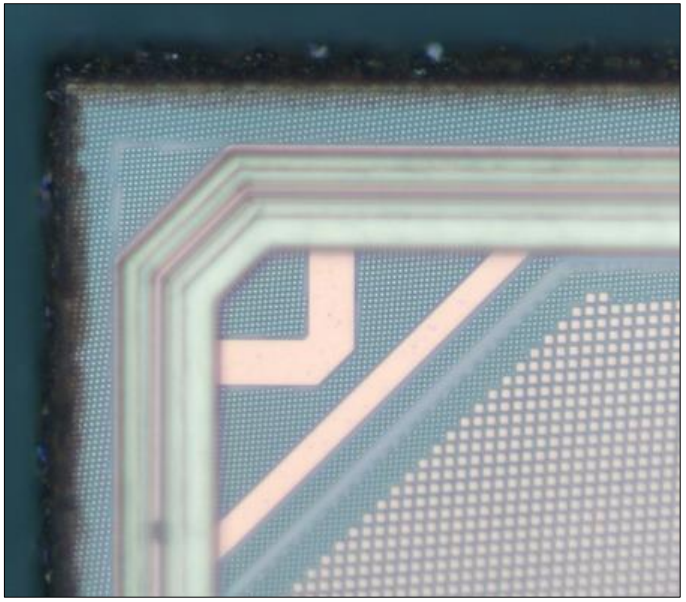
IOActive.

# How to get the key?

- Decap the chip
- Find the fuses
- Figure out a way to dump the bits
  - But it’s “virtually impossible”…

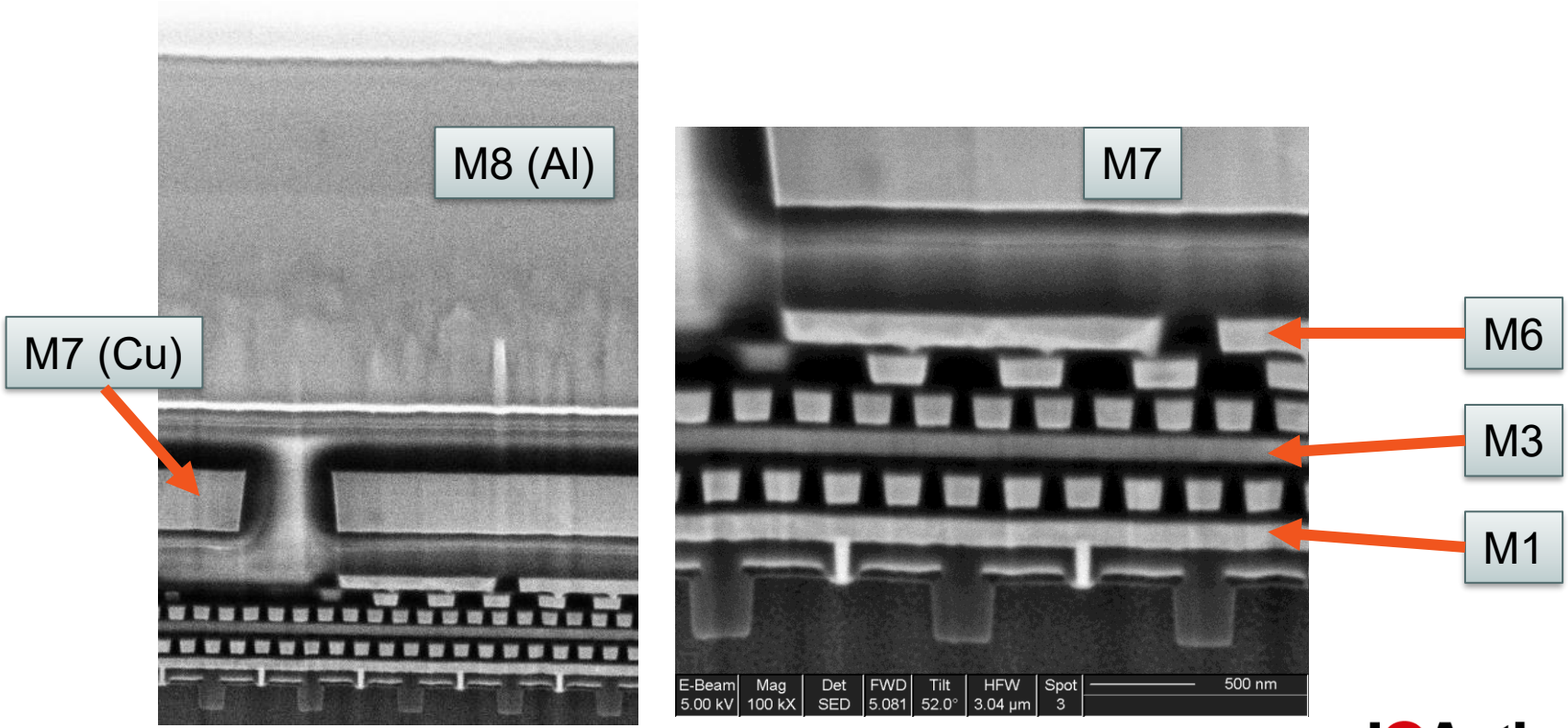
# Top metal (M8) overview





2.218 x 2.477 mm (5.493 $mm^2$)

# 40nm TSMC (7 Cu + 1 Al)

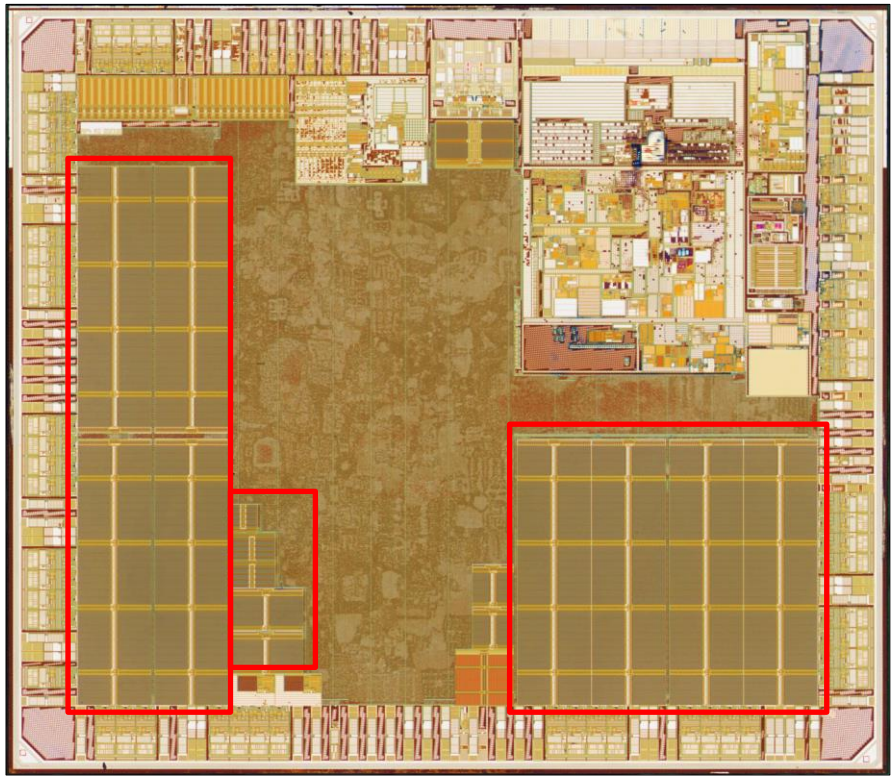


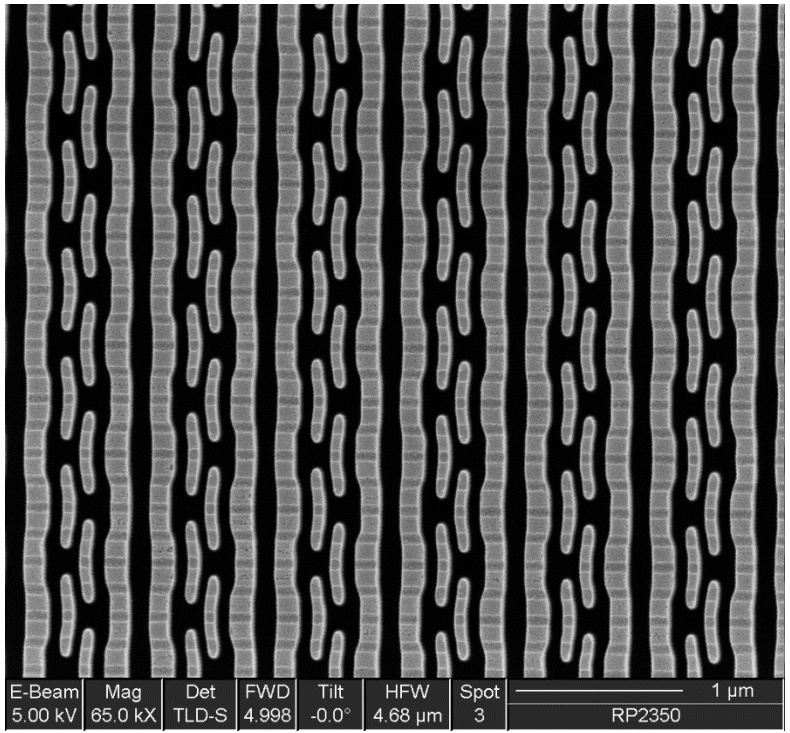





IOActive.

# 6T SRAM (single port)



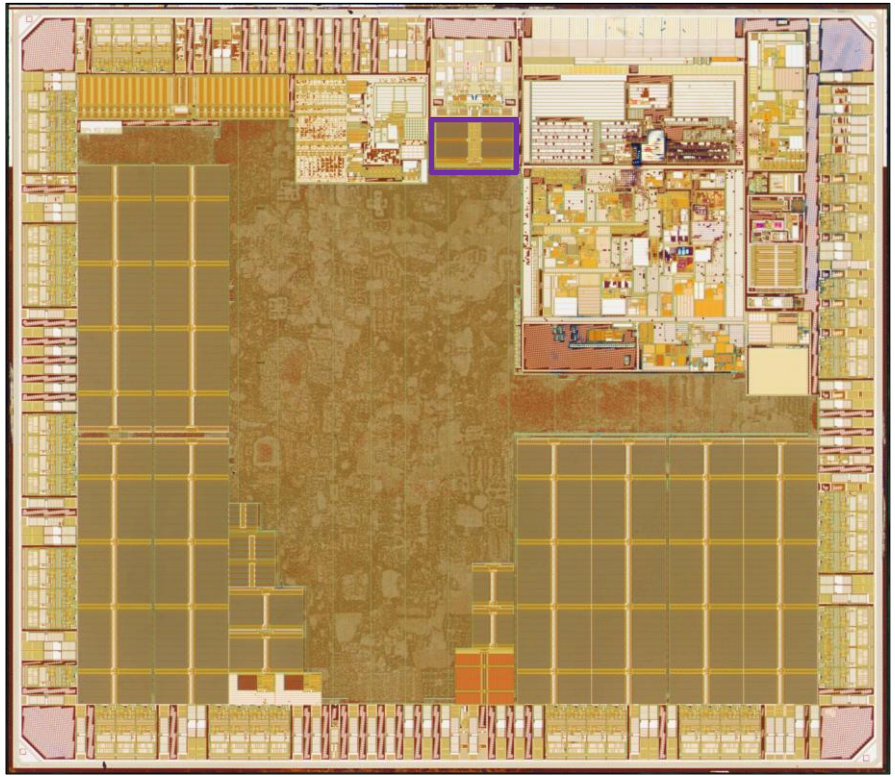


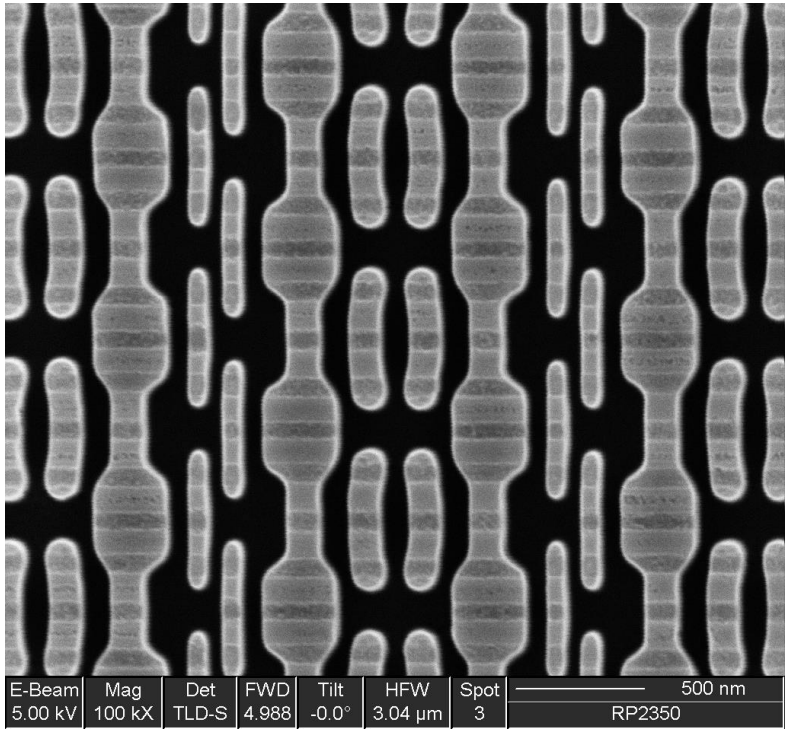





IOActive.

# 8T SRAM (dual port)



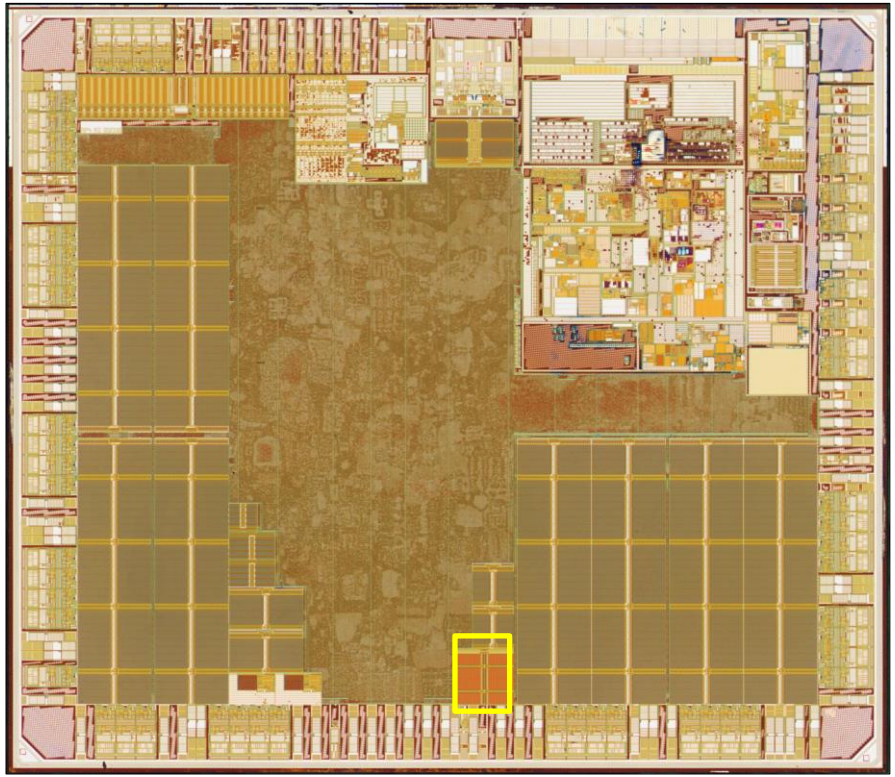


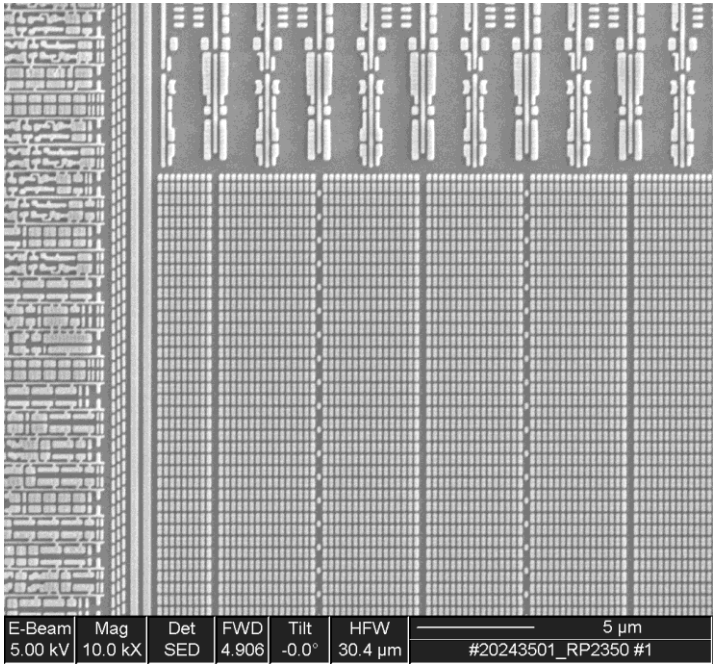





IOActive.

# Mask ROM (no bits visible, likely on M1)






Boot ROM is open source

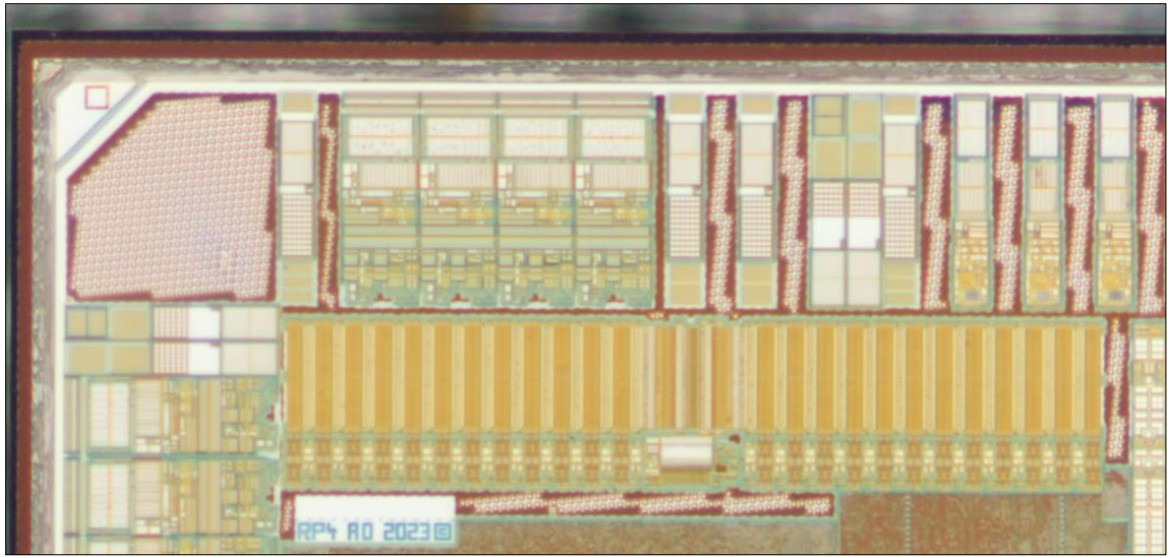
No point spending time dumping it



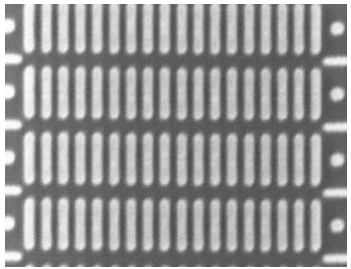


IOActive.

# 0w0 what's this?




- Not SRAM
- Not ROM
- Not flash
- Looks to be 24 cols
- Fuses are 24 bits
- Hmmm…

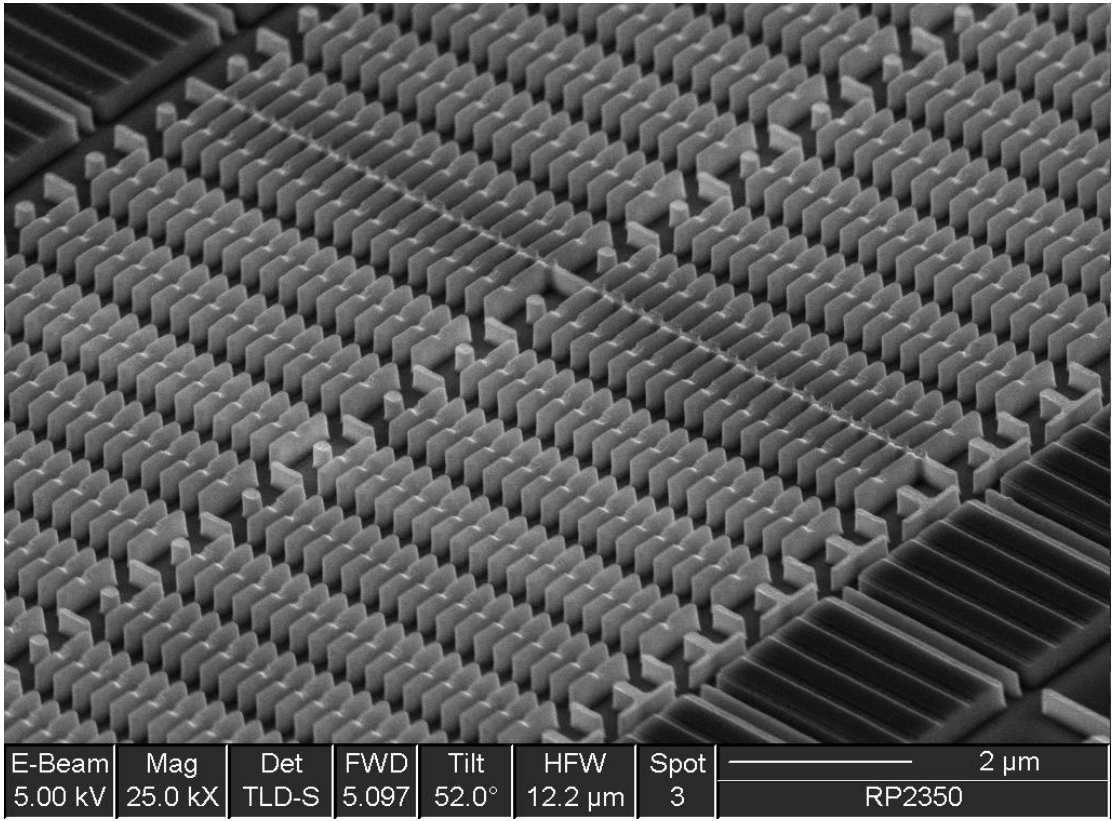






IOActive.

# Closer…

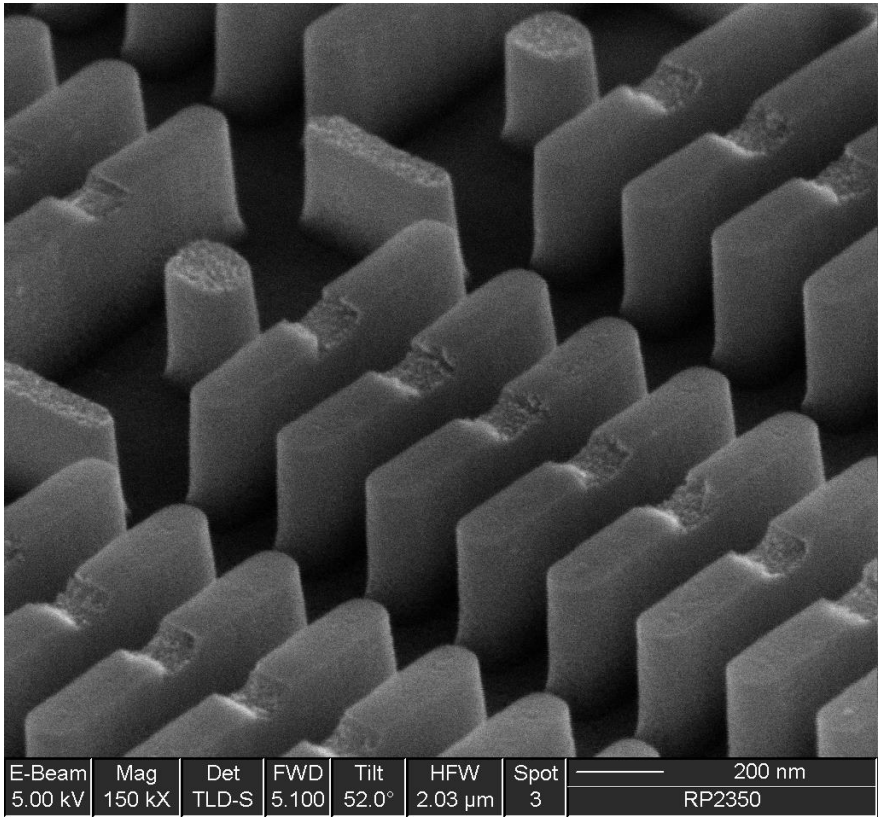







IOActive.

# Closer!

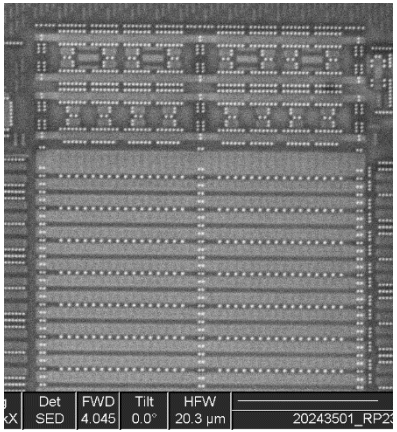




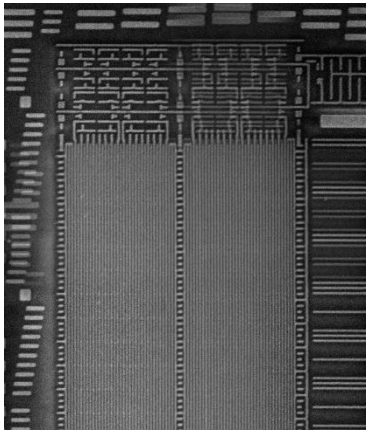



IOActive.

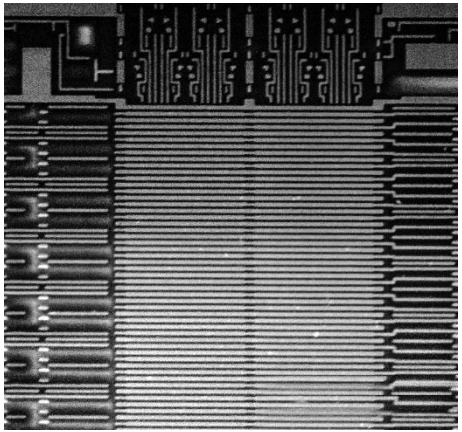
# Multi layer overview



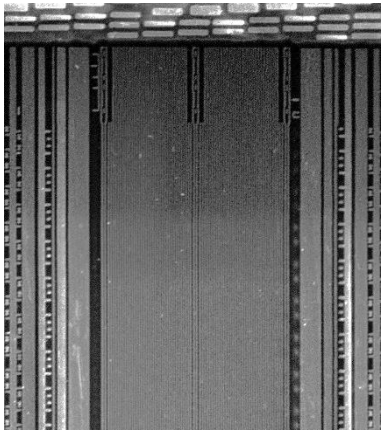

Poly
Contact



M1



M2



M3

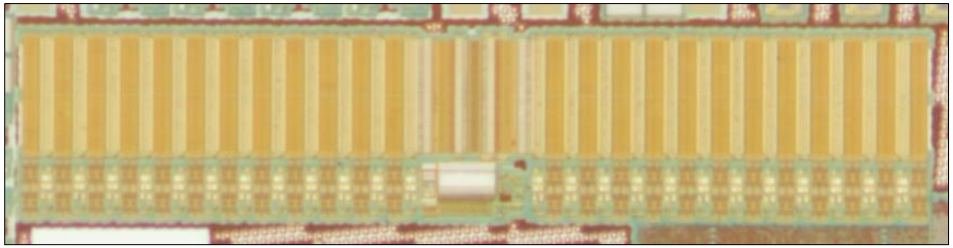


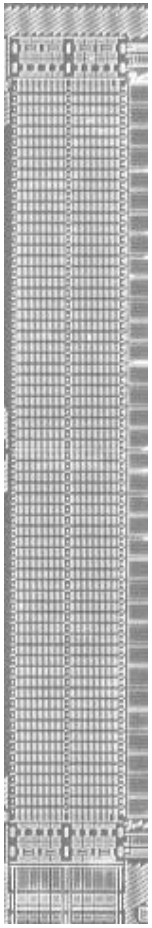


IOActive.

# High level address map

- 24 columns: matches the 24-bit fuse width
  - Each column is probably all rows of one bit
  - Don’t know which column is which bit plane yet
- 2 x 2 sub-arrays in each bit plane
  - 4096 words in the whole memory
  - Each sub-array must be 1024 words





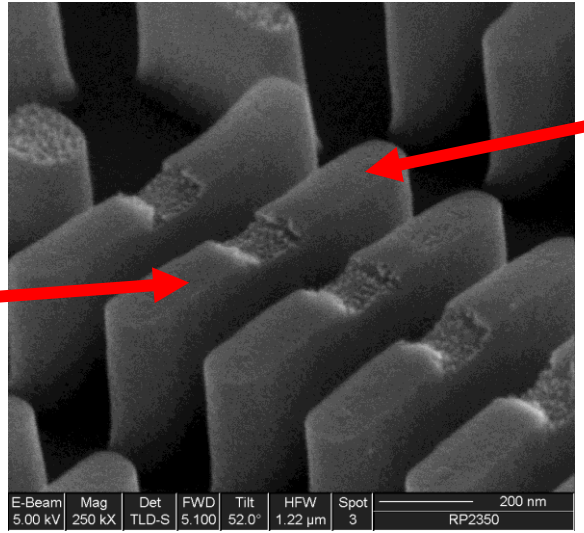




IOActive.

# Subtract the dummy features and…

- 16 columns x 32 rows per sub array
- 512 tiles for 1024 bits
- Each tile must store two bits

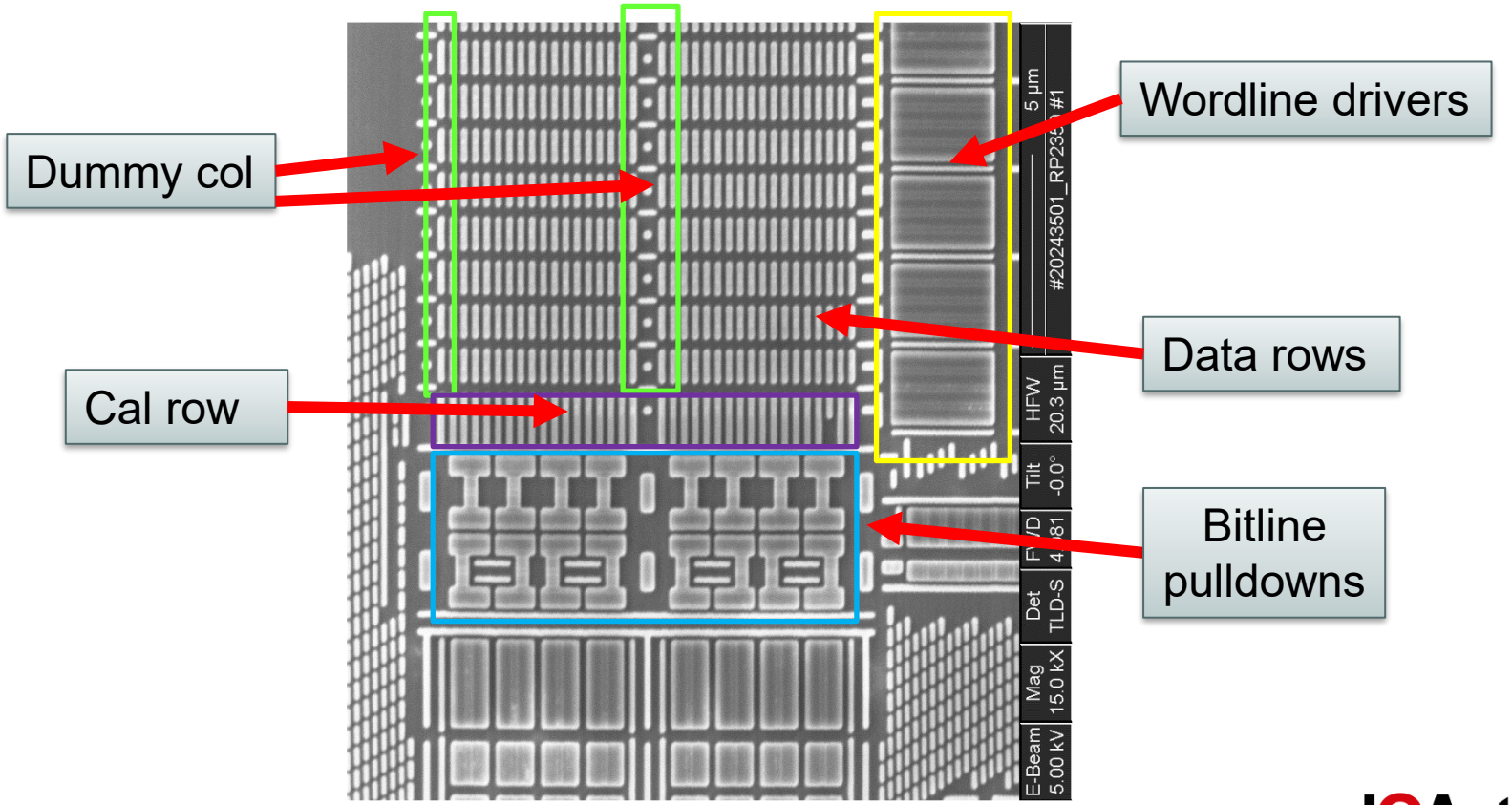







IOActive.

# Bottom of a single bit plane (substrate)

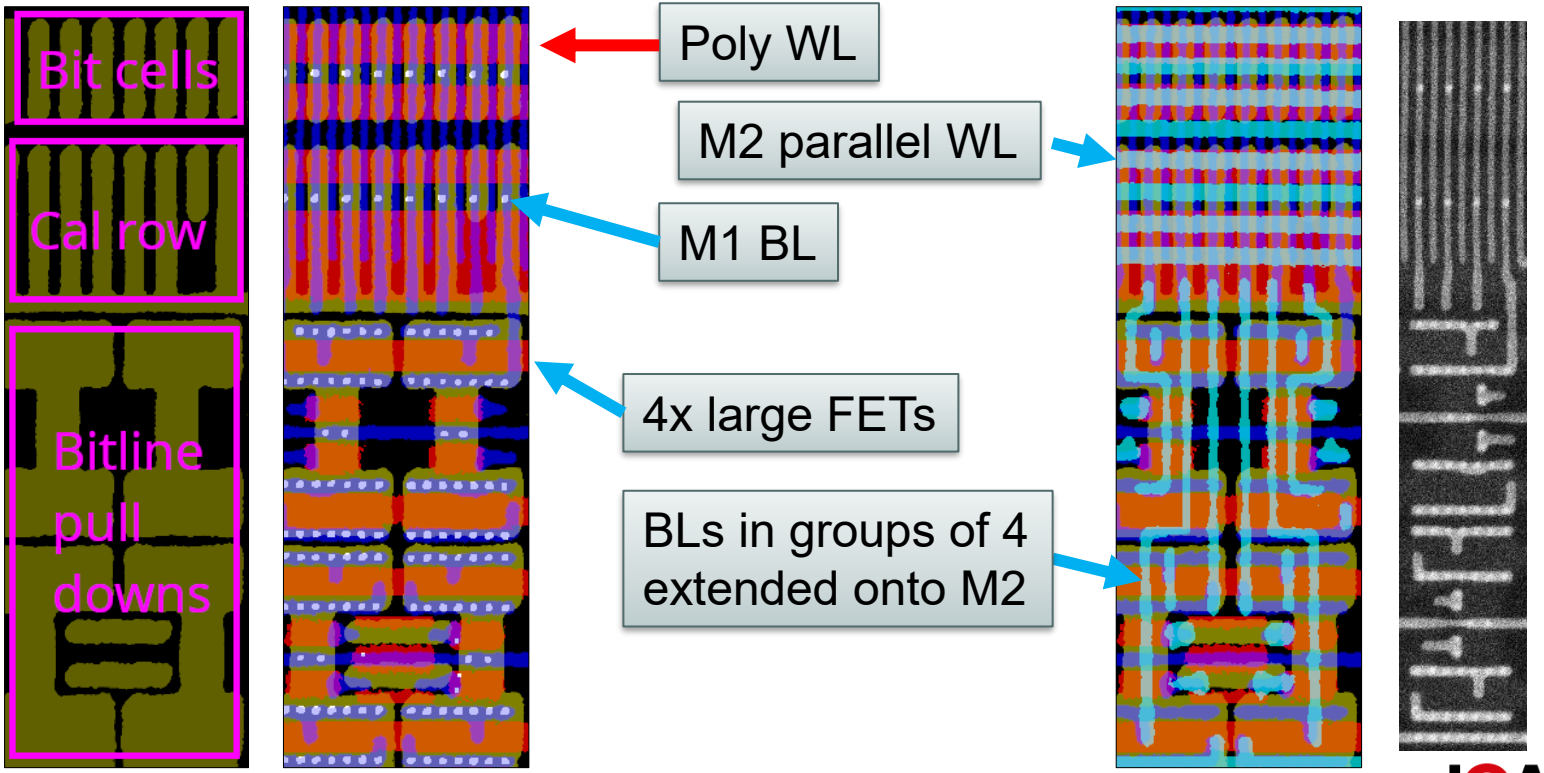







IOActive.

# Contacts, poly, M1, M2

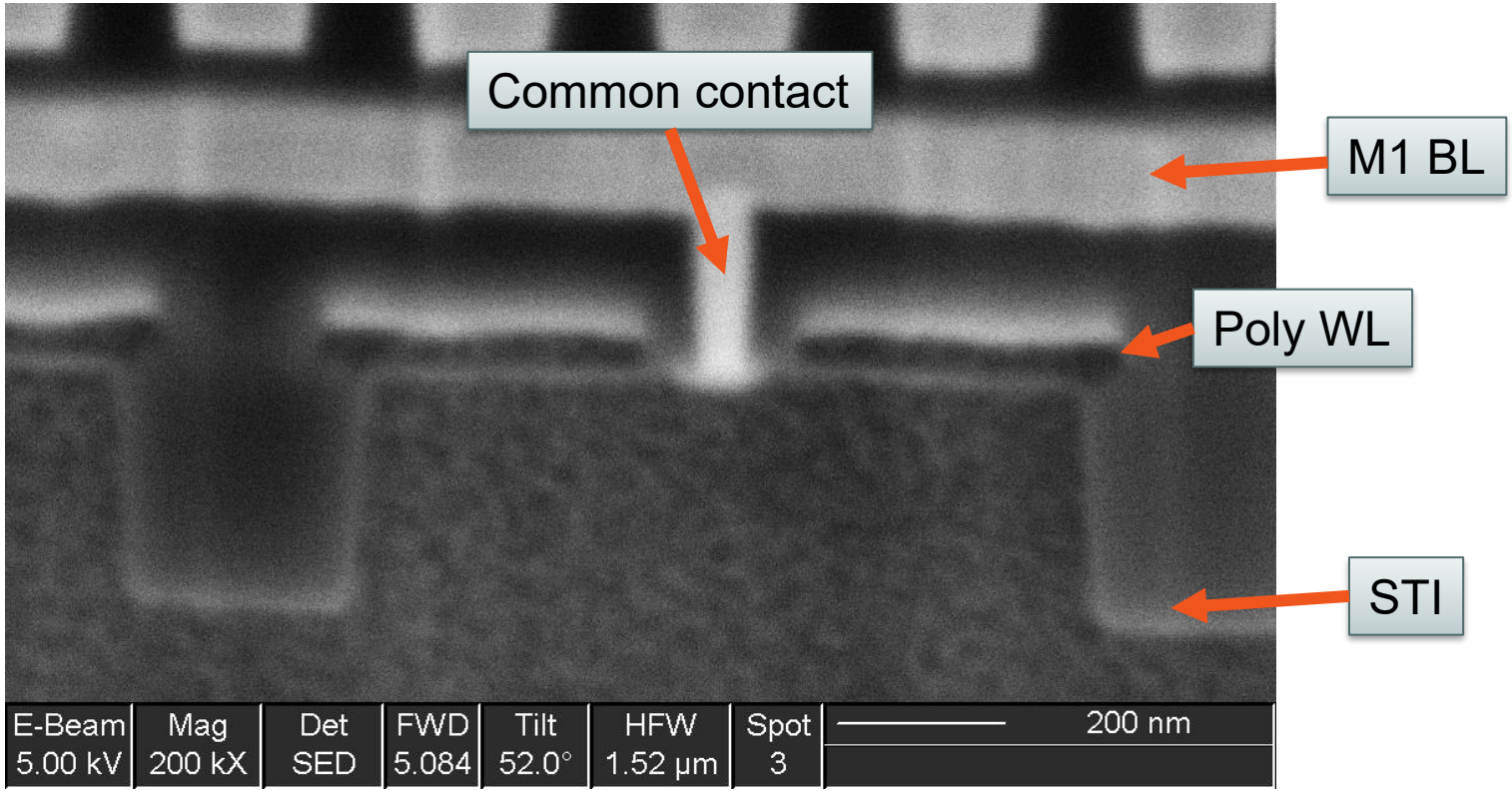







IOActive.

# Bitcell pair in cross section

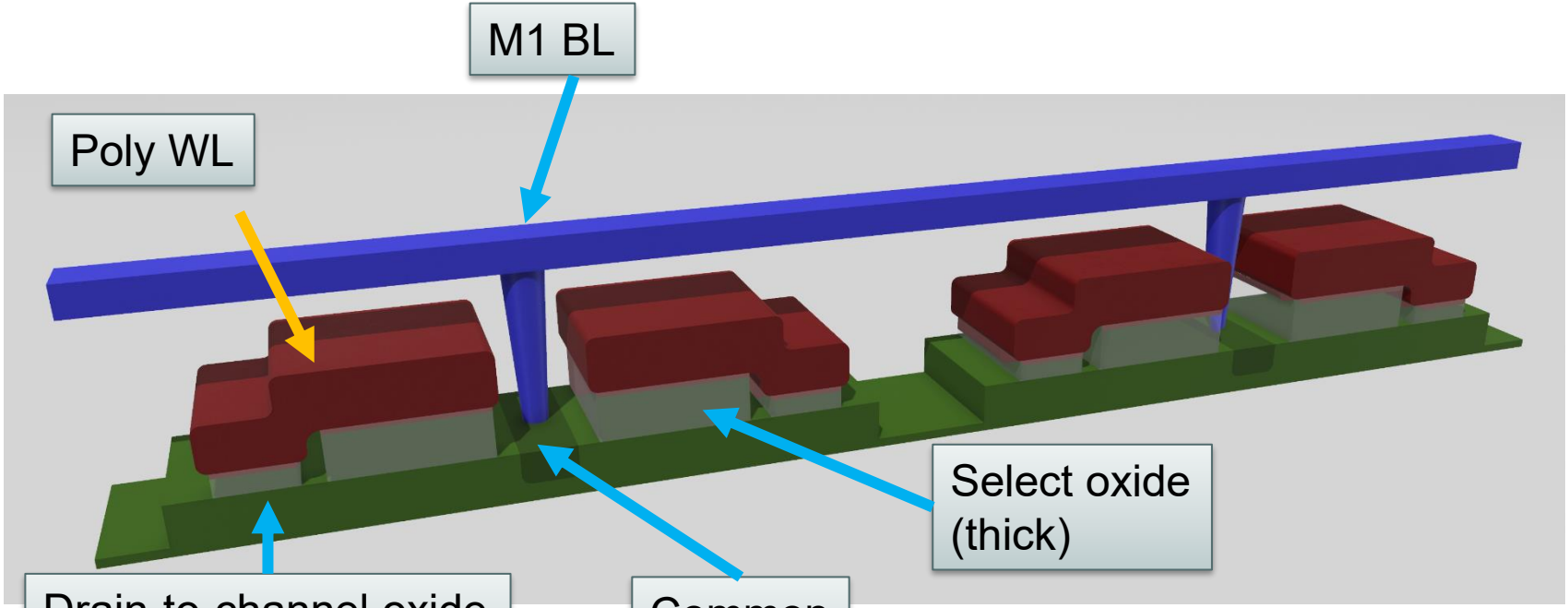







IOActive.

# Two rows of bit cells (not to scale)

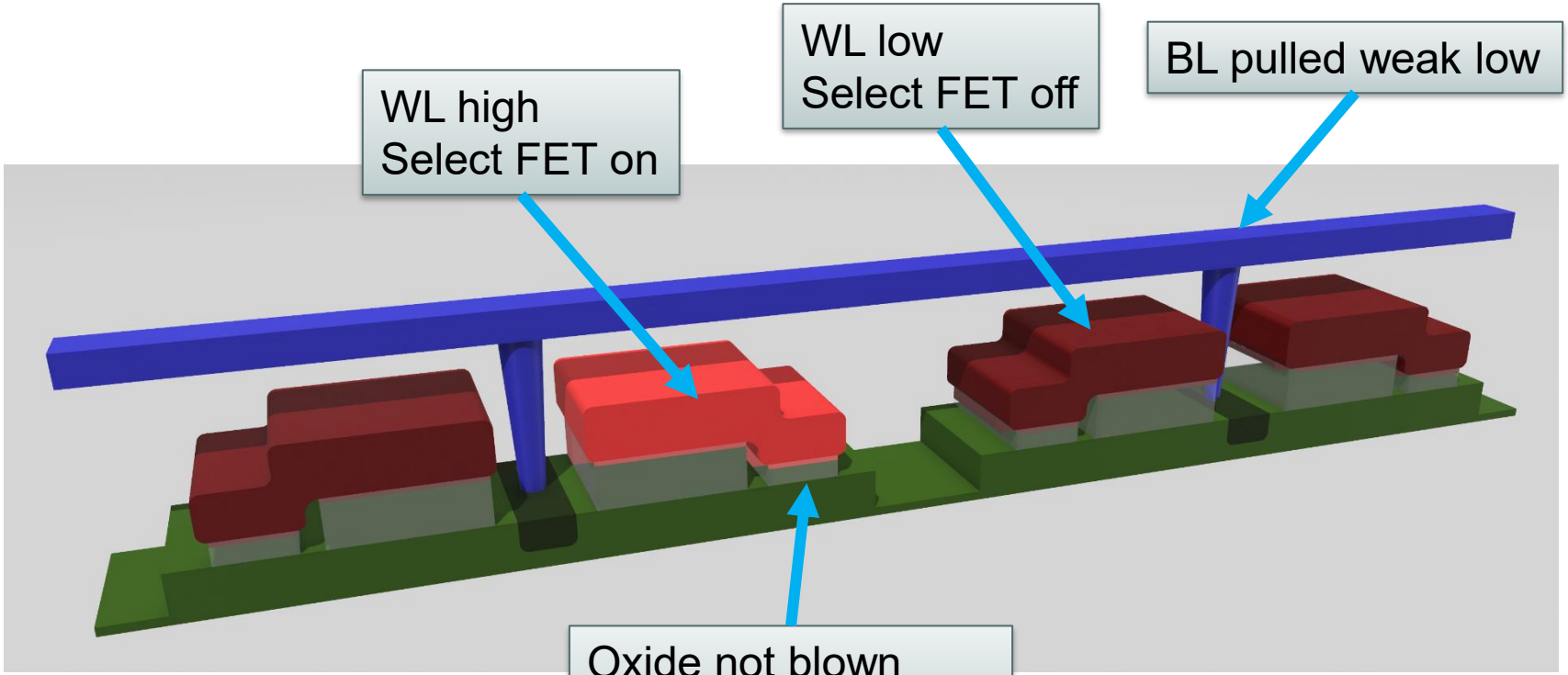







IOActive.

# Reading blank (0) bit



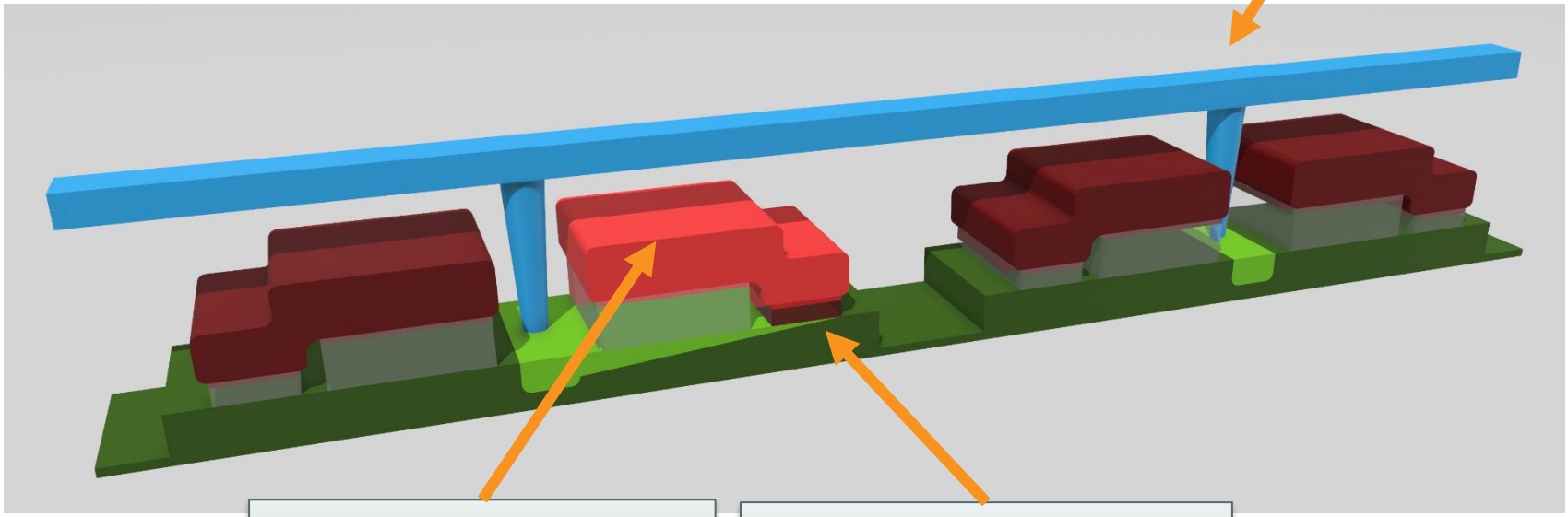





IOActive.

# Reading programmed (1) bit

BL driven high
Overrides pulldown



WL over channel high
Select FET on

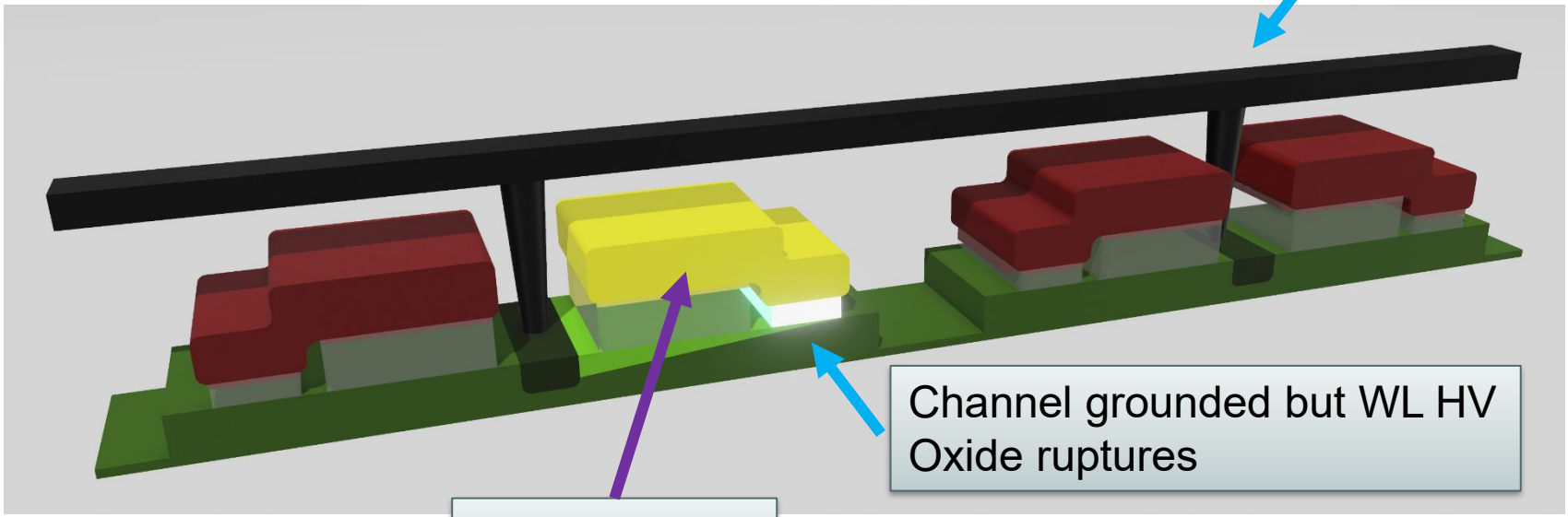
Oxide blown
N+ poly *is* drain of FET





IOActive.

# Programming 0 bit to 1



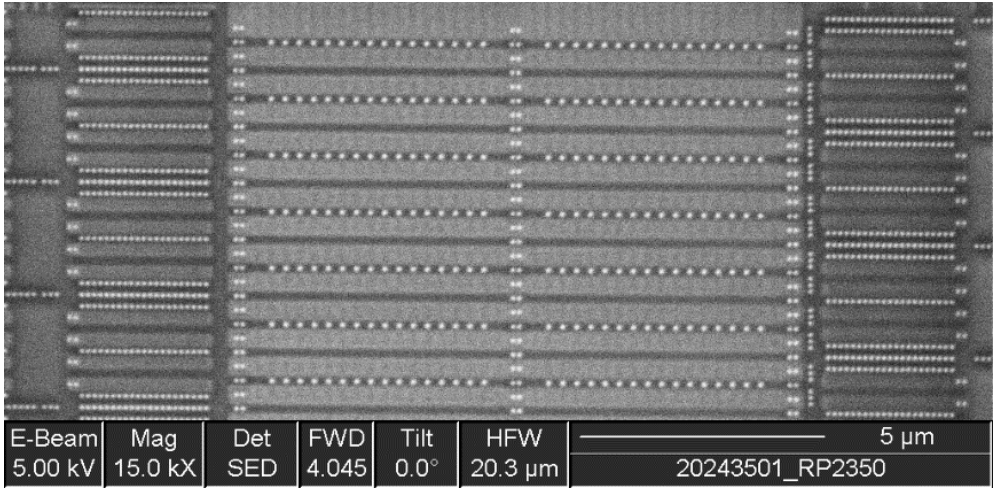





IOActive.

# What if we could just see the bits?

- But…
  - All bits look the same in SEM
- Maybe a different technique?








IOActive.

# What if we could **see voltages**?

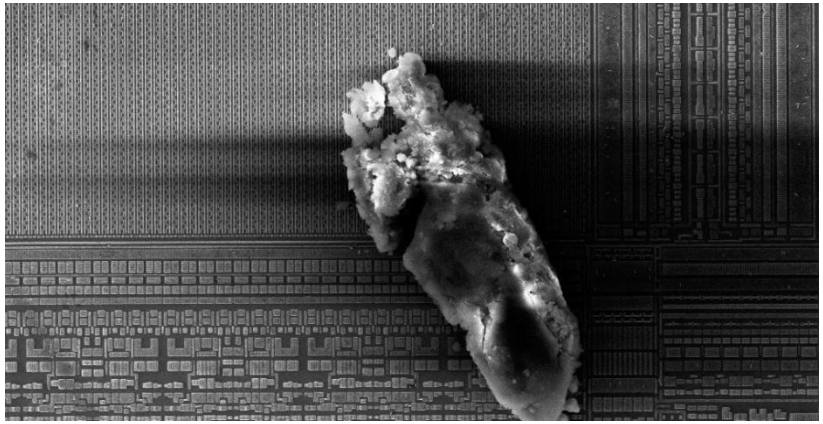
- Voltage contrast imaging!
  - Passive VC: beam used for imaging + biasing
  - Active VC: beam used for imaging, external bias

# Particle beam imaging recap

- Raster a beam of charged particles over the specimen
  - $e^-$ (SEM), $Ga^+$ (FIB) most common
- Beam interacts with sample atoms
- Detect these interactions somehow
  - X-rays
  - Secondary electrons
  - Backscattered electrons
  - Cathodoluminescence

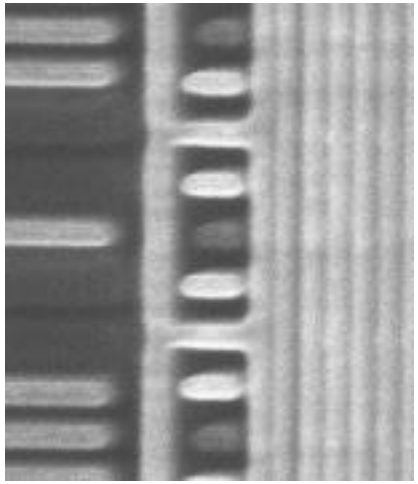
# Particle beam imaging w/ charged sample

- SEs are low energy / low mass – easily deflected
- Charged samples are normally bad for SE imaging
  - Causes image shifts, dark spots, difficulty imaging
  - This is why we coat
- But we can put this to use!



Positively charged dust speck
Attracts SEs from adjacent areas

# Voltage contrast imaging



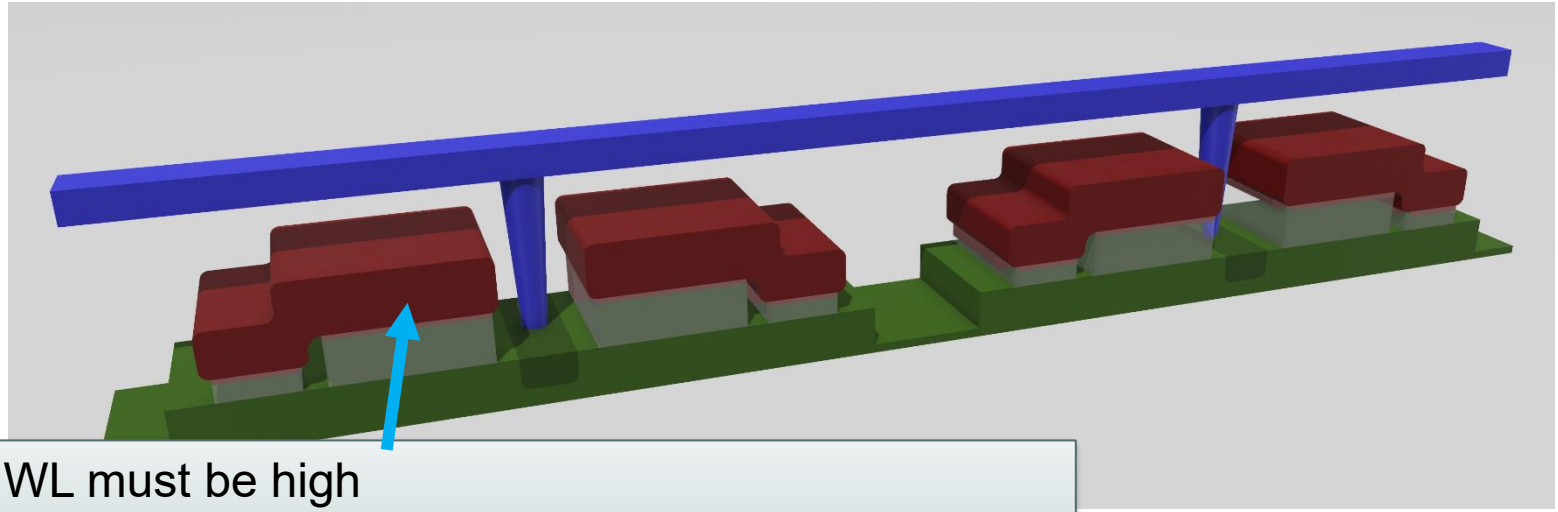
- **Deliberately** charge sample surface
- Charged polygons attract / repel SE's
  - Causes visible brightness shifts in the image
  - Positive charge: attract SE's, less detected, dark image
  - Negative charge: repel SE's, more detected, light image
- Infer memory state, connectivity, etc

# Passive voltage contrast (PVC)

- Imaging beam *also injects charge* into DUT
  - No probe or external power source needed
- Less precise control
  - Everything you can see is also having charge injected
  - Scan rate, coatings, accelerating voltage, etc. affect results
  - Balance charge deposition and bleed-off

# Can’t do SEM PVC on this bitcell structure

- Electron beam is negatively charged



WL must be high
If select FET is off, we see nothing
So injecting negative bias with e-beam is useless

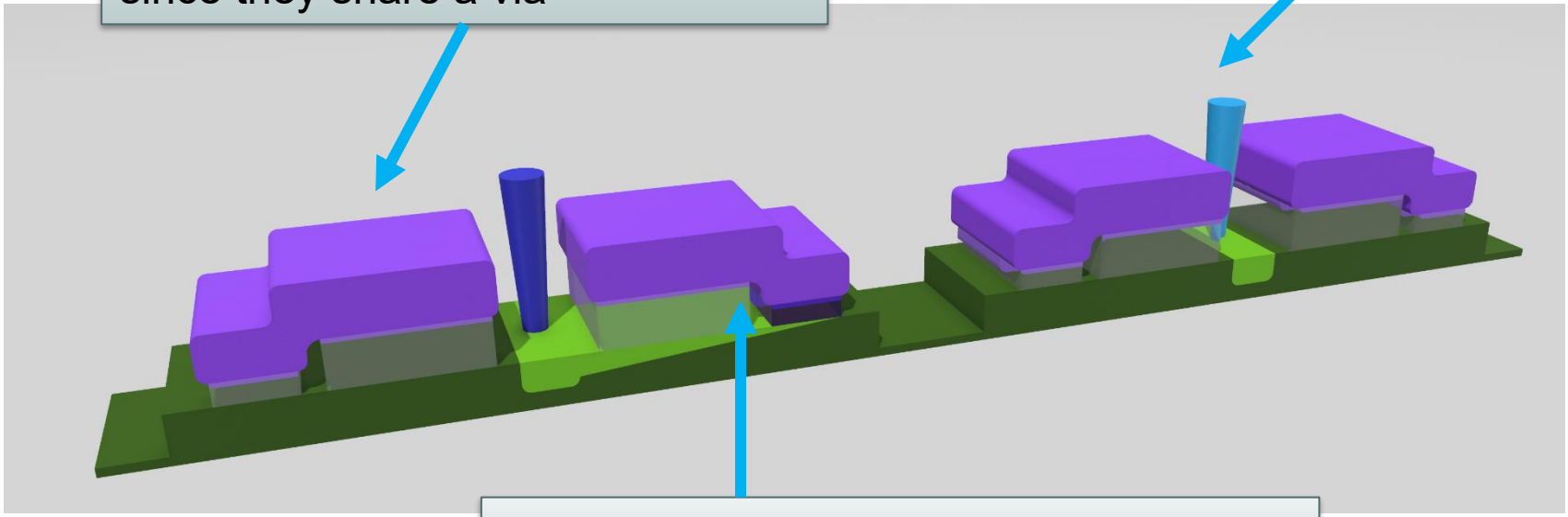
# FIB PVC

- Use positively charged $Ga^+$ beam
  - Injects positive charges into sample
- Destructive (sputters DUT surface)
  - Use low beam current, work fast to minimize damage

# Mass fuse readout via FIB PVC

Both halves on simultaneously
Cannot distinguish between halves since they share a via

Deprocess to contacts
No BL remaining



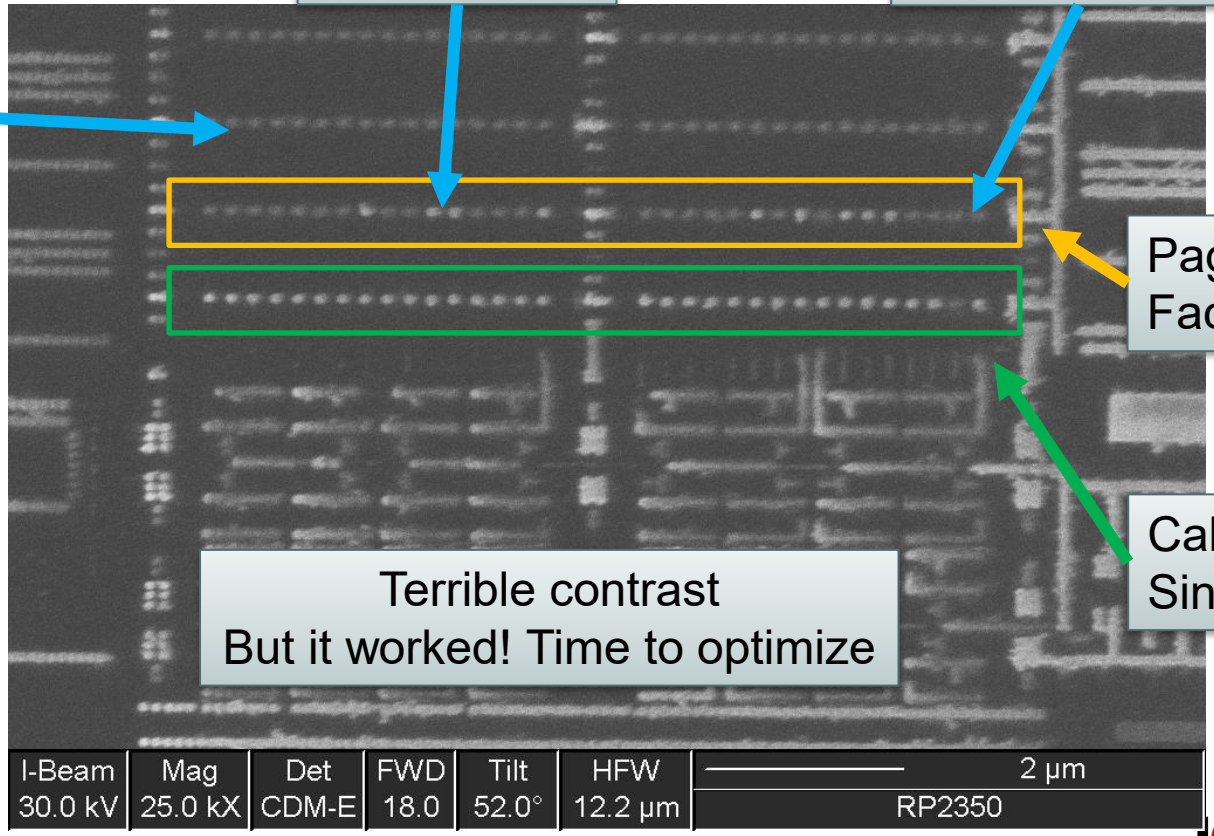
Ion beam injects positive charge on WLs
Positive $V_{gs}$ = bitcell transistor on





IOActive.

# First light



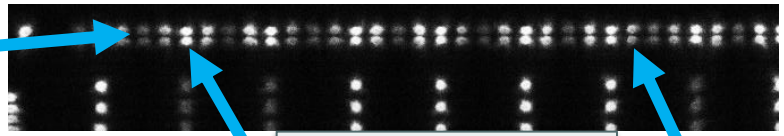





IOActive.

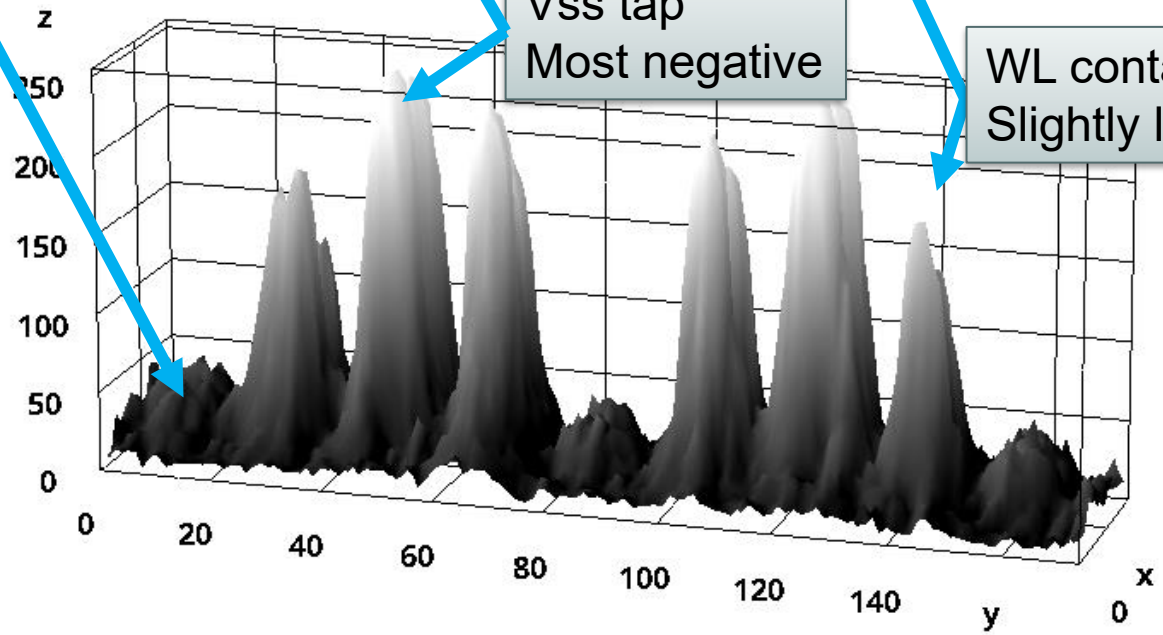
# Qualitative voltage measurements via PVC

Vdd tap
Least negative



Vss tap
Most negative

WL contact
Slightly less negative



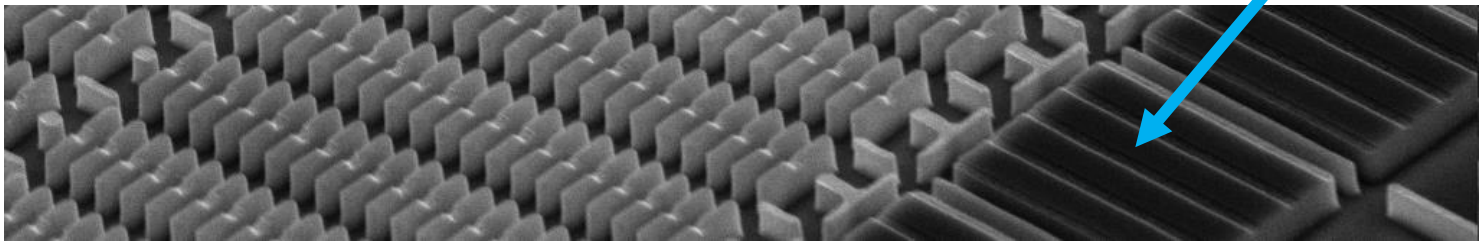





IOActive.

# Proposed Contrast Mechanism

- WL builds up *some* positive charge
- But most of it leaks to ground
- Likely sits at a few hundred mV
  - Around $V_t$, bitcell transistors starting to conduct
  - Can pass ~10 pA with reasonably low $V_{ds}$



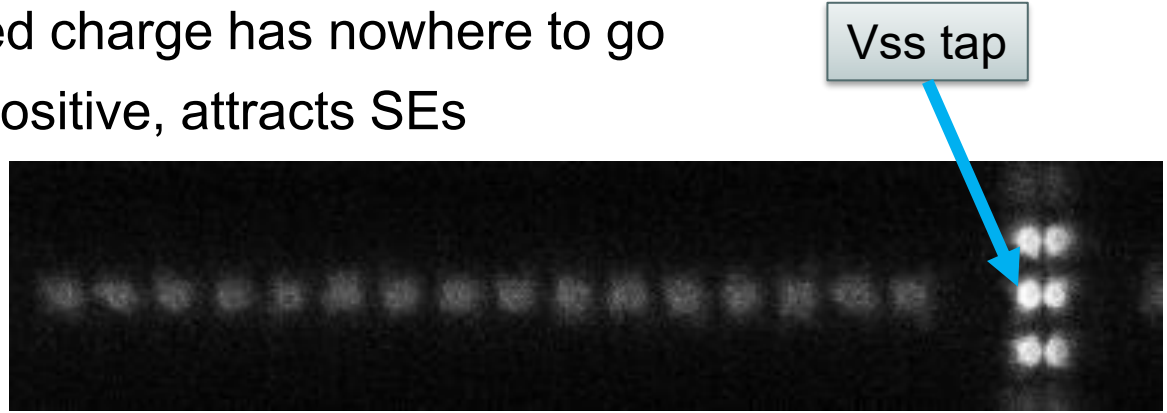





IOActive.

# Proposed Contrast Mechanism

- Unprogrammed bit
  - BL is floating even though select FET is on
  - Beam-induced charge has nowhere to go
  - BL strongly positive, attracts SEs
  - Dark image



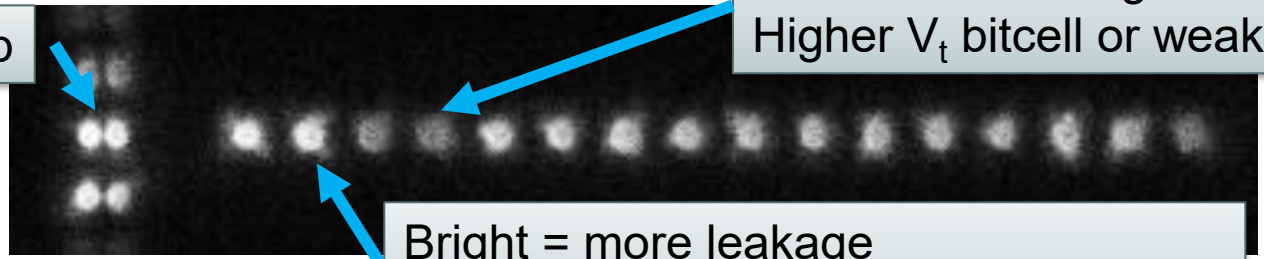





IOActive.

# Proposed Contrast Mechanism

- Programmed bit
  - BL is connected to WL by $R_{ds}$(on) of select FET
  - WL (thus BL) is at very low positive voltage
  - Light image, but a bit darker than grounded
  - Can see process variation!





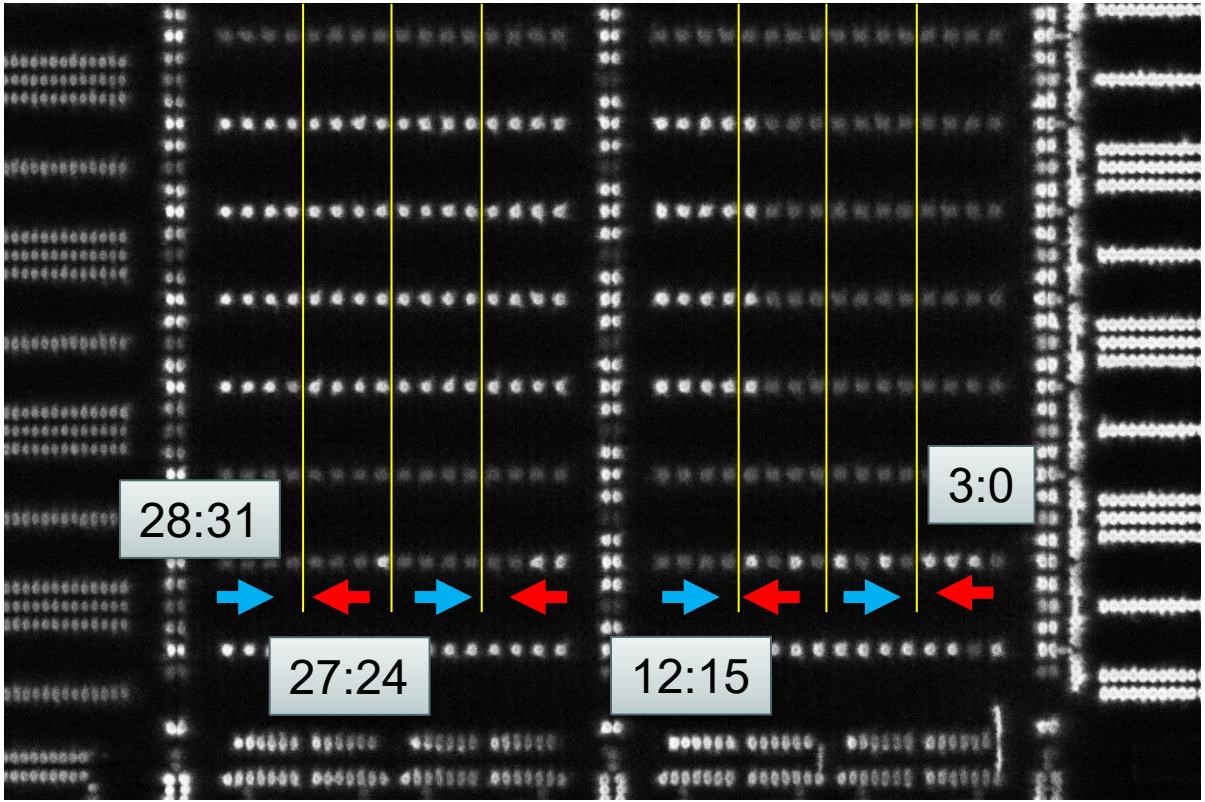

IOActive.

# Reversing the address map

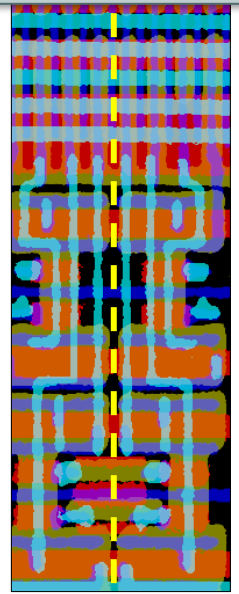
- Program our own test patterns
  - Different data in each bit plane to ID columns
  - Horizontally + vertically asymmetric to ID mirroring
  - Find the challenge key
  - And, of course, have some fun in the process

# Column ordering

West side of array shown
East side mirrored L-R




Even/odd nibbles mirrored



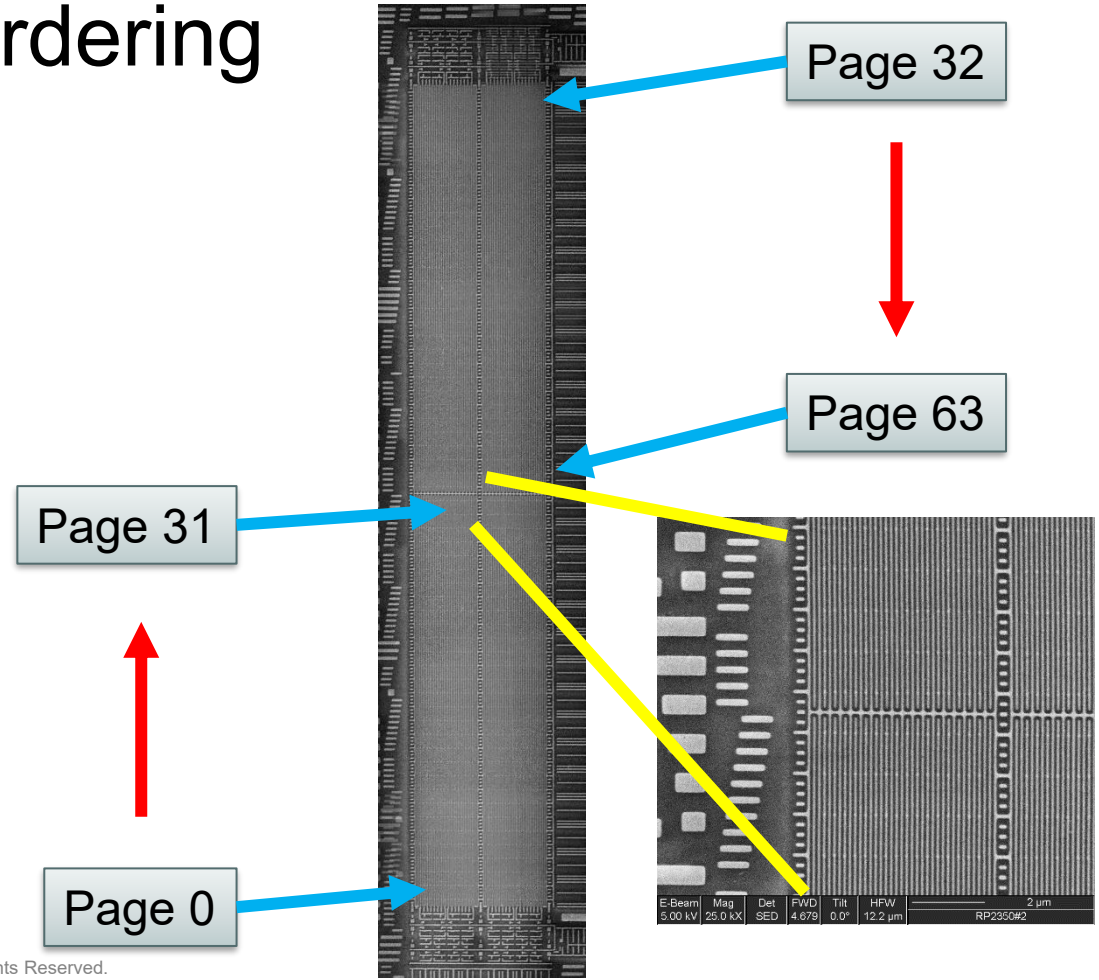




IOActive.

# Row ordering

M1 view

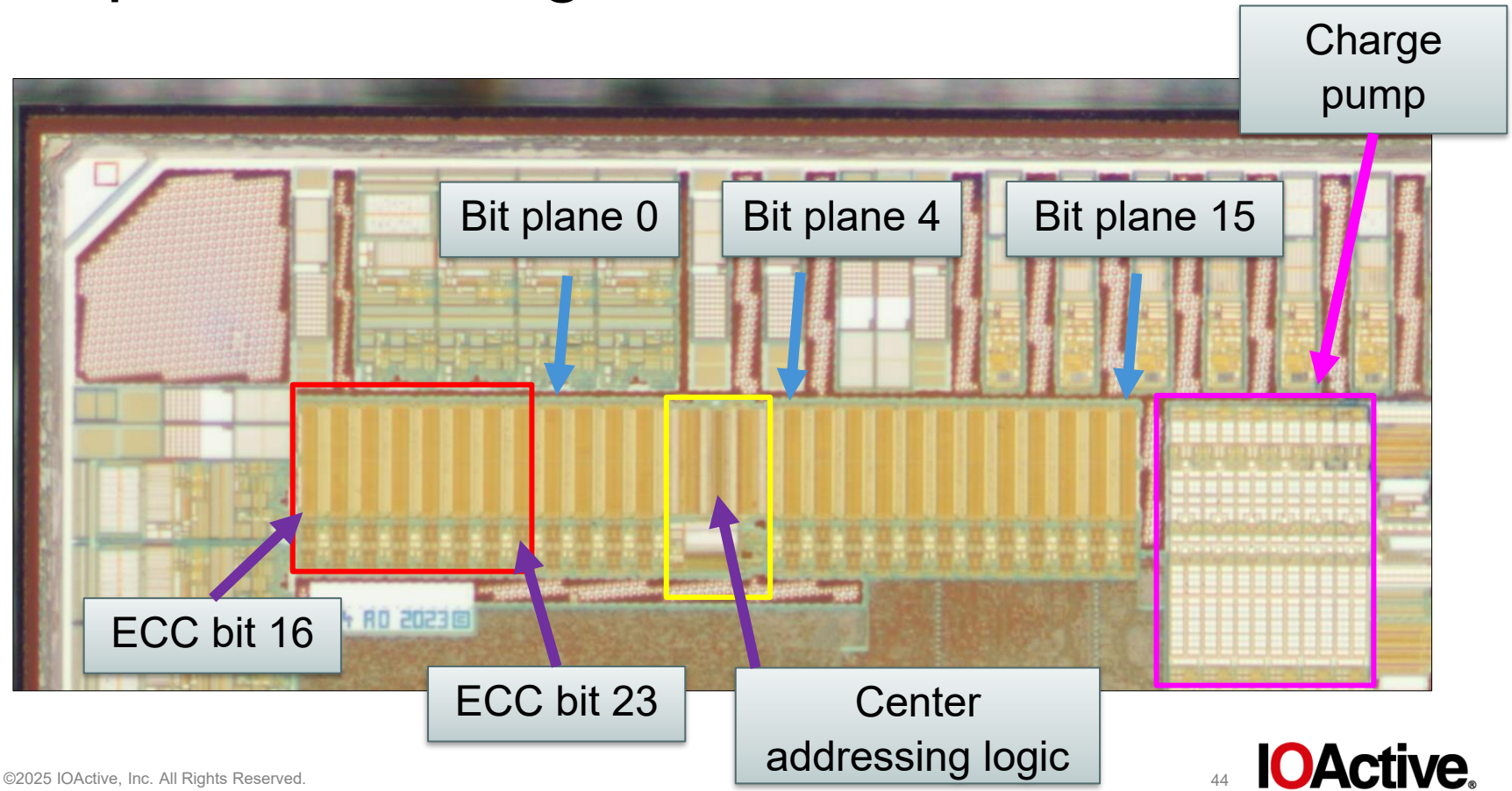







IOActive.

# Bit plane ordering

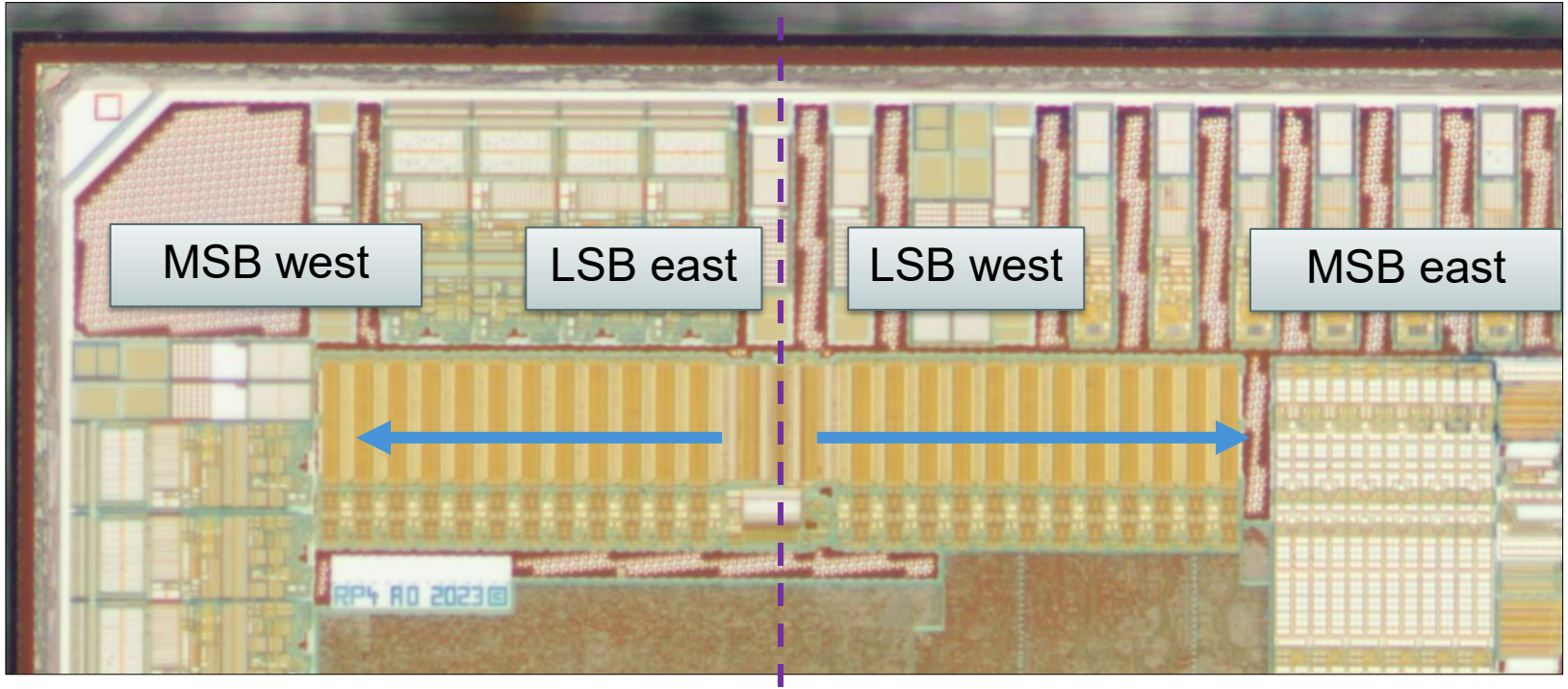







IOActive.

# Bit plane mirroring








IOActive.

# Fuse rendering script

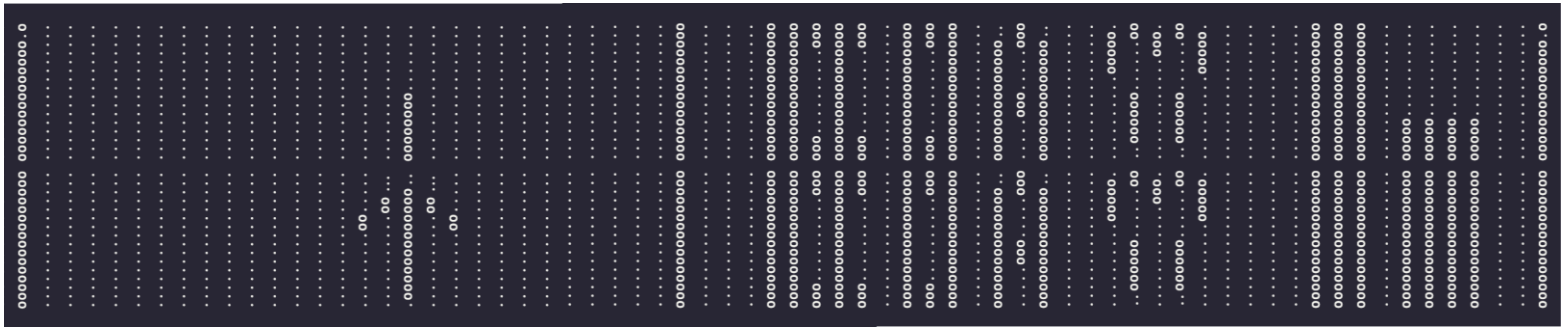
- Python script: fuse values in, ASCII art map out

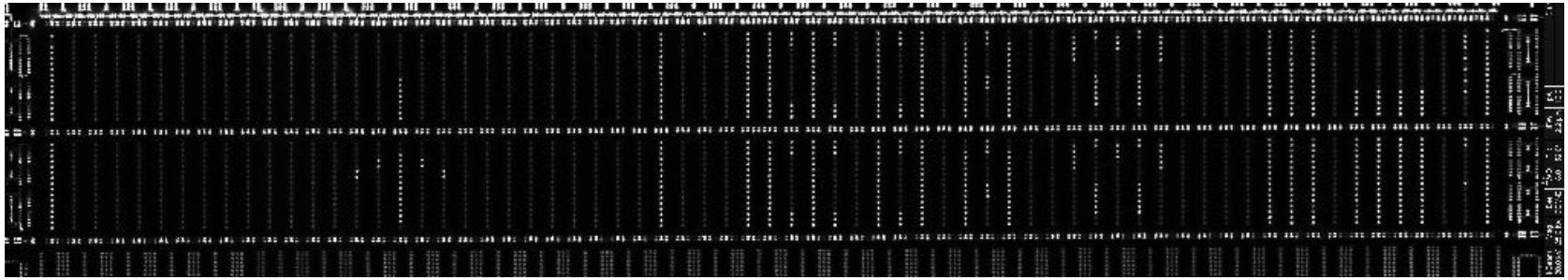




IOActive.

# Full bitplane extraction test



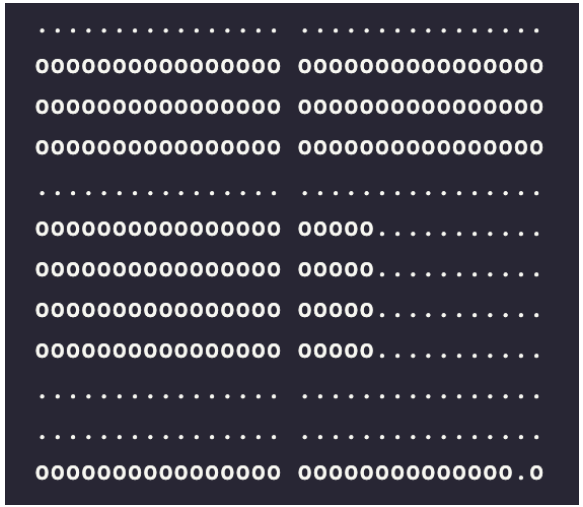


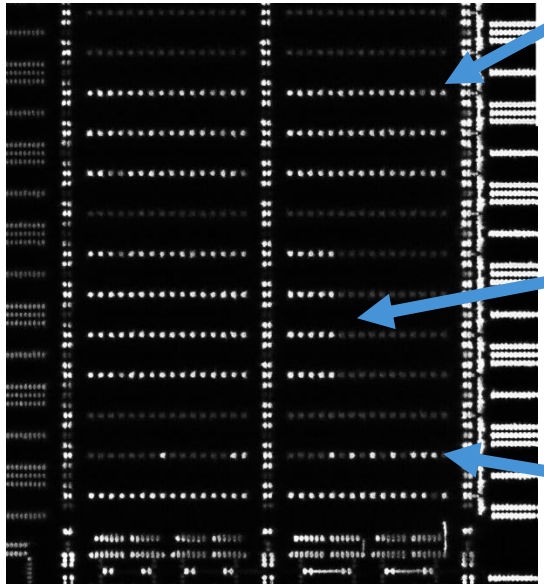




IOActive.

# Test pattern closeup



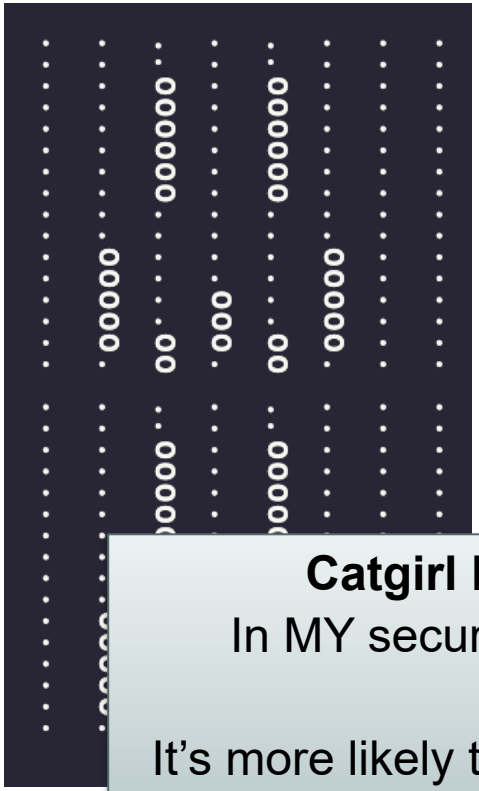



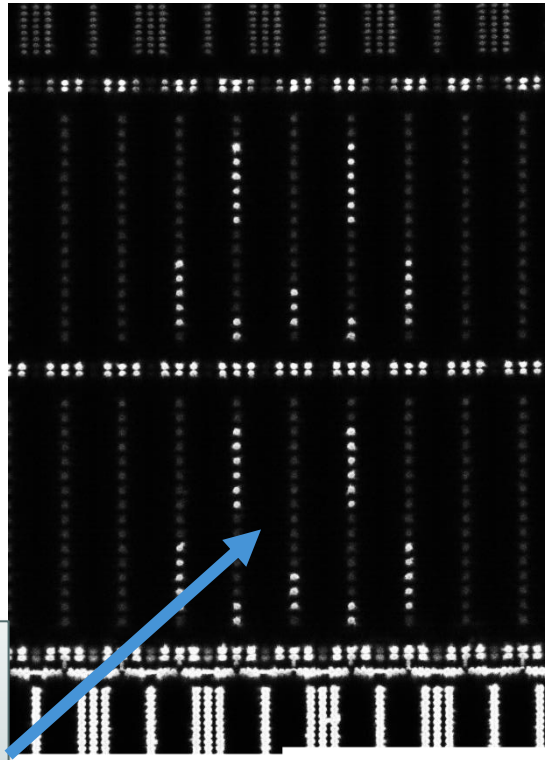





IOActive.

# Test pattern closeup



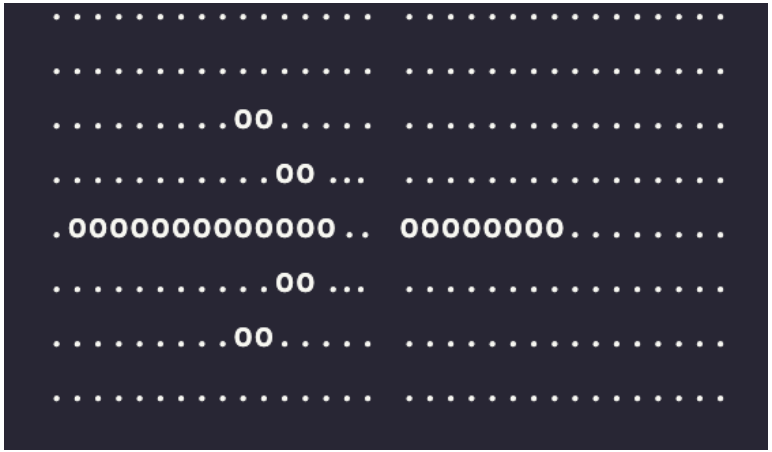


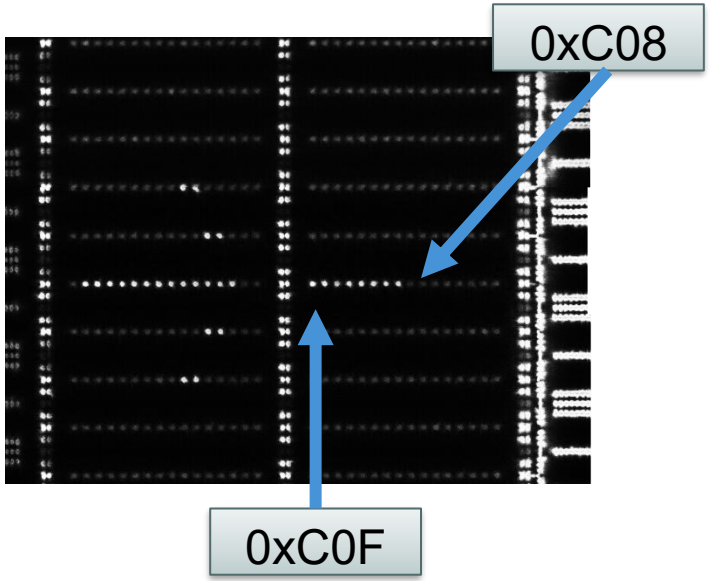
**Catgirl hackers?**

In MY secure boot keys?

It’s more likely than you think =3

# Test pattern closeup






The goal is easy: Find an attack that lets you dump a secret hidden in OTP ROW 0xc08 - the secret is 128-bit long, and protected by `OTP_DATA_PAGE48_LOCK1` and RP2350's secure boot!





IOActive.

# Possible improvements

- Use probe or FIB platinum to ground half the WLs
  - Should allow readout of even / odd halves separately
  - First experiment failed, haven't had time to try again
- Play w/ scan speed and reduced area scans
  - Bias everything –ve with e-beam
  - Then scan I-beam over WL (to charge) and BL (to image)
  - Make sure not to get *any* beam energy on the opposite WL
  - If perfectly focused, *might* work

# Mitigations

- PVC is physics, impossible to prevent
- Proper fix (requires silicon spin)
  - Encrypt data at rest so fuse dump is worthless to attacker
  - Force adversary to RE many Mgates to find hardwired key
- Near term: increase adversary workload
  - Store 0xFF or ~key in opposite half of paired fuses
  - Blocks the trivial mass-readout attack
  - Requires more work to dump one WL at a time

# Mitigations

- Spread key across many physical words
  - 1 word in N pages is Nx the work to dump vs data in same page
  - Store data all over the place and hash or XOR down
- Goal: force a FIB edit of ~all 2048 WLs in the fuse array
  - Can’t quite use 100% of rows because factory trim etc
  - But you can make the adversary hate you!

# Conclusions

- Antifuses aren’t as invisible as claimed
- A few k$ of FIB time goes a long way
- Any secret in on chip memory can and will be extracted
- Encrypting / obfuscating fuse data raises the bar

# Acknowledgements


- Daniel Slone, Mario Cop, Tony Moor (IOA)
  - Decap, deprocessing, some imaging
- Lain Agan (IOA)
  - Memory map RE, 3D renders, analysis script
- Entropic Engineering
  - RP2350s were hard to find right after launch, they hooked us up
- Raspberry Pi
  - For inviting the hacker community to pwn their product

# Questions?