IOActive®
Research-fueled Security Services

\ WHITE PAPER \

RP2350 Hacking Challenge

Andrew Zonenberg
Principal Security Consultant

Antony Moor
Senior Director of Silicon Lab Services

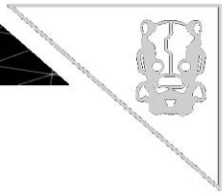
Daniel Slone
Silicon Lab Technician

Lain Agan
Senior Security Consultant

Mario Cop
Semiconductor Engineer

January 2025

Contents



Executive Summary

IOActive, Inc. (IOActive) has demonstrated the ability to extract the contents of antifuse bit cells in the Raspberry Pi RP2350 microcontroller via an invasive physical attack. These antifuse bit cells are intended to store firmware encryption keys and other sensitive data.

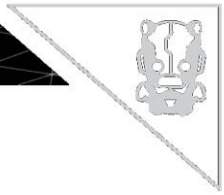
An attacker in possession of an RP2350 device, as well as access to semiconductor deprocessing equipment and a focused ion beam (FIB) system, could extract the contents of the antifuse bit cells as plaintext in a matter of days. While a FIB system is a very expensive scientific instrument (costing several hundred thousand USD, plus ongoing operating expenses in the tens of thousands per year), it is possible to rent time on one at a university lab for around $200/hour for machine time[1] or around two to three times this for machine time plus a trained operator to run it. This is low enough to be well within the realm of feasibility in many scenarios given the potential value of the keys in the device.

The attack can theoretically be performed with only a single device and would take a skilled attacker approximately 1-2 weeks of work to perform the initial reverse engineering and process development on blank or attacker-programmed test chips. Actual target devices would take 1-2 days per chip to prepare the sample and extract a small amount of data such as a key; a full fuse dump might require an additional day of machine time for imaging of the entire array.

As with any invasive attack, there is a small chance of the device being damaged during handling and deprocessing, so a real-world attacker would likely procure several samples to ensure a successful extraction.

The attack is based on established semiconductor failure analysis techniques and is not specific to the RP2350; it is likely that similar techniques can be used to extract data from other antifuse-based devices.

[1] For example, the University of Washington currently charges $200/hr for FIB time plus $165/hr for staff time https://www.moles.washington.edu/facilities/molecular-analysis-facility/rates/



RP2350 Device Overview

The RP2350 is a 32-bit dual core[2] microcontroller developed internally by Raspberry Pi and manufactured on a 40 nm 8-metal CMOS process by TSMC. The die measures 2.218 x 2.477 mm (5.493 mm^2).

The RP2350 has no internal flash and relies on an external QSPI flash chip (although variants exist with stacked logic and flash dice in a single package); firmware is either copied to the 520 kB of internal SRAM and executed from there, or run execute-in-place directly from the flash.

To protect sensitive data stored in firmware, the RP2350 allows firmware to be encrypted at rest. This flow (see section 10.1.2 of the datasheet) involves loading a signed, unencrypted stub and an encrypted firmware image from external flash into SRAM, verifying it against a key burned into the OTP fuses, and executing the stub after the signature check completes. The stub reads the firmware decryption key from the fuses, decrypts the remainder of the image into SRAM, and executes it. Thus, in order to decrypt a dump of the external flash memory, an attacker must extract this key from the fuses.

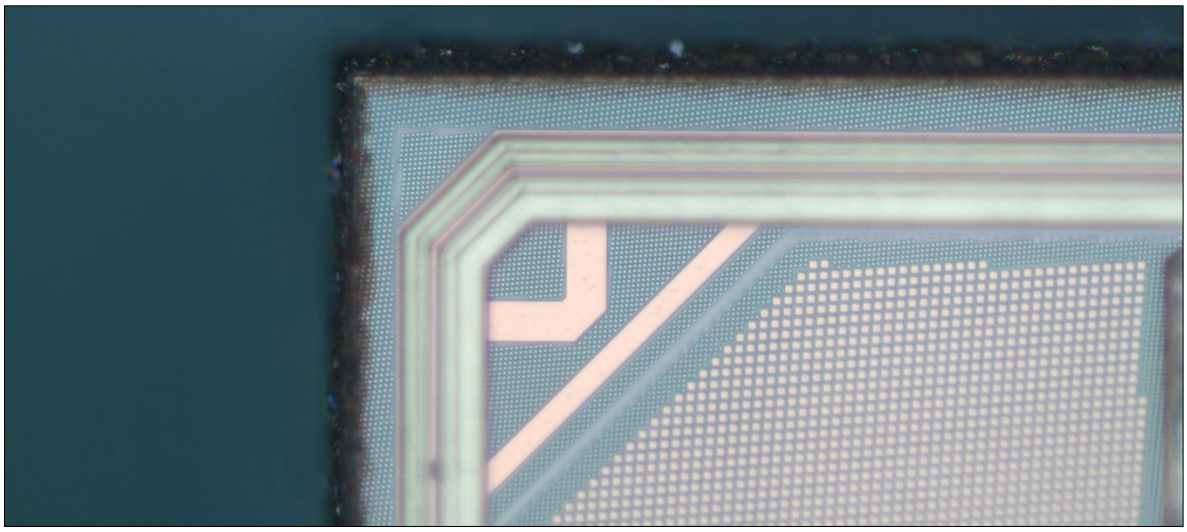


Figure 1. Northwest corner of die showing TSMC fiducial

[2] Four physical CPUs (two RISC-V and two ARM), of which only two can be used at any one time due to capacity limitations of the AMBA bus fabric

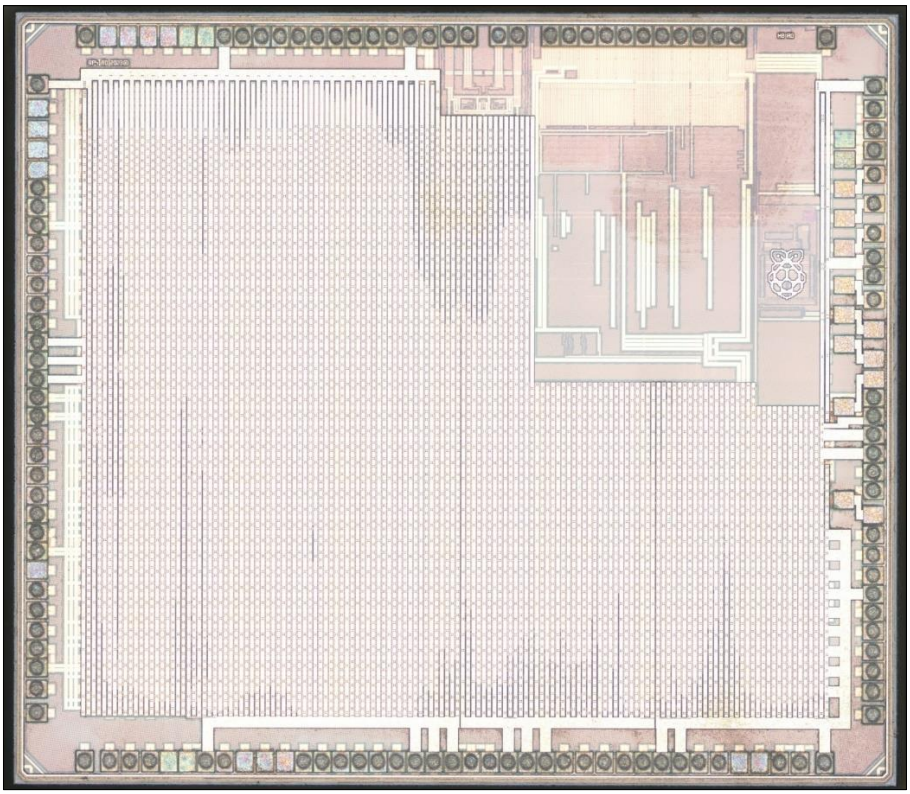


Figure 2. Top metal (M8) die overview

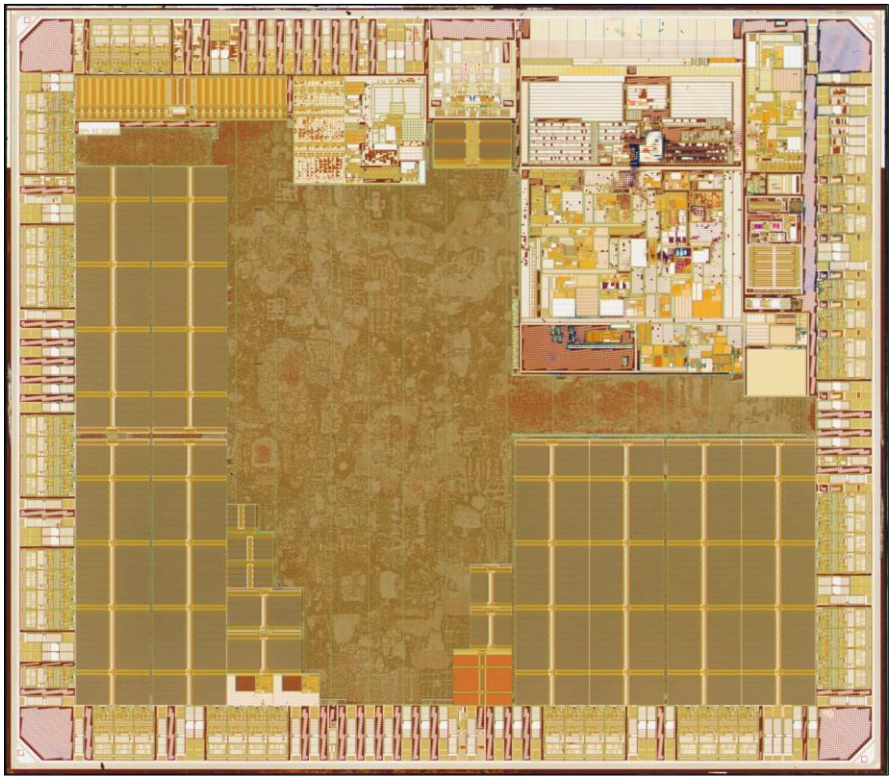


Figure 3. Substrate die overview after removal of metal and poly

Several different types of memory were observed on the die: single port 6T SRAM, dual port 8T SRAM, mask ROM, and OTP fuse.

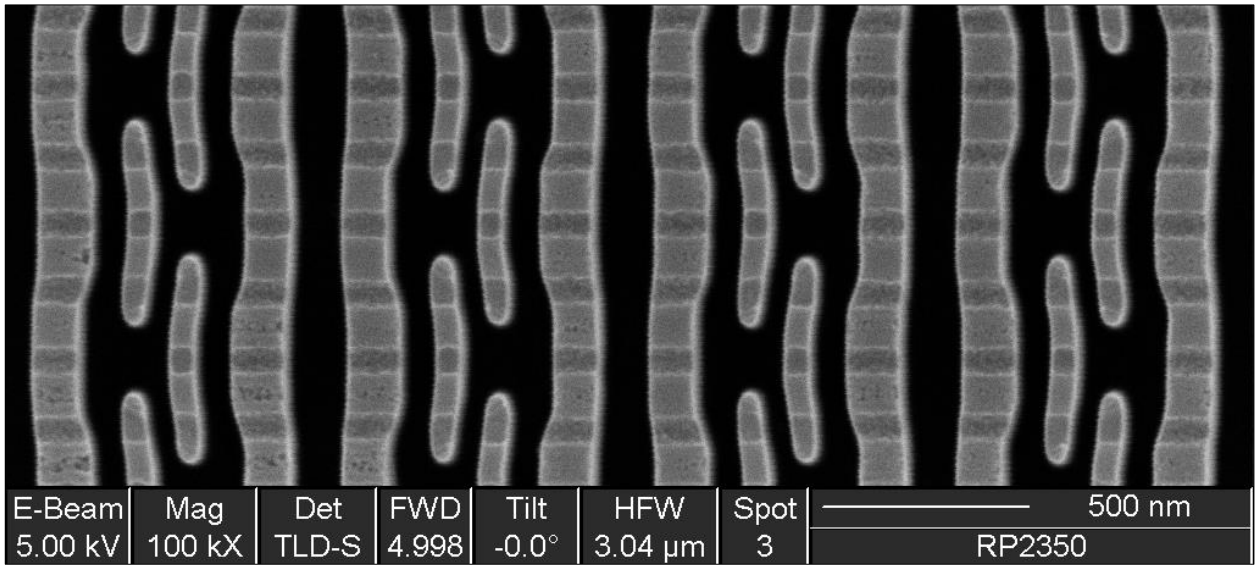


Figure 4. 6T single port SRAM cells, substrate view

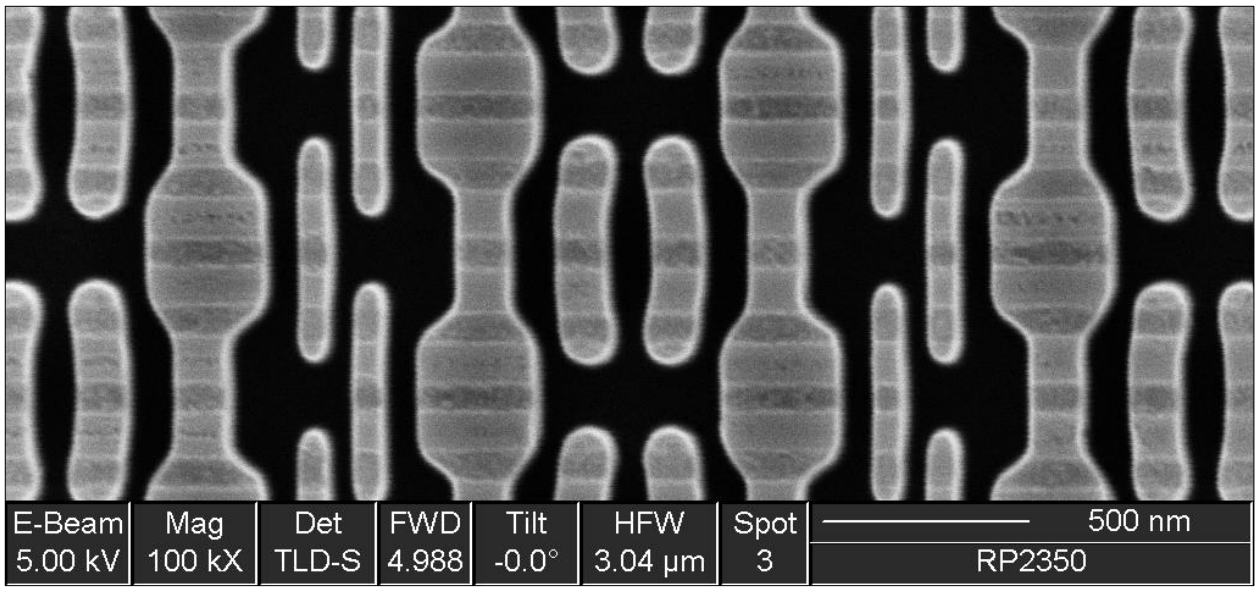


Figure 5. 8T dual port SRAM cells, substrate view

NOTE: When discussing images and IC layout in this document, we use the terms north/south/east/west to refer to physical relationships between objects. These terms refer to the die in the canonical orientation (with USB PHY at the 12 o'clock position and text/logos upright); however, some images are rotated for clarity and are noted as such.

RP2350 Fuses

The RP2350 fuses use the Synopsys “XHC & XBC OTP NVM IP: SHF Architecture” IP, likely dwc_nvm_ts40np5sxxxxh0nopxxxi[3] or one of its close cousins.[4]

Synopsys heavily promotes the security of their fuse architecture, claiming “State of memory (even for a few bits) is virtually impossible to detect using physical attack or reverse engineering techniques.”[5] This is consistent with claims made by many other manufacturers of antifuse-based IPs, FPGAs, and various high-security devices and appears to reflect the currently prevailing—though ultimately incorrect—belief in the semiconductor industry that antifuses offer greatly increased security against data extraction over other forms of memory (e.g. mask ROM or flash).

The fuse array is logically structured as 64 pages, each containing 64 rows of 24 bits, accessible as either 24-bit raw words or as 16-bit words with ECC. The ECC data is stored in bits 23:16 of the fuse data word, and, in non-ECC mode, this range can be used for an additional byte of user data.

The fuse array is located in the northwest corner of the die, just south of the I/O pad ring and west of the USB PHY.

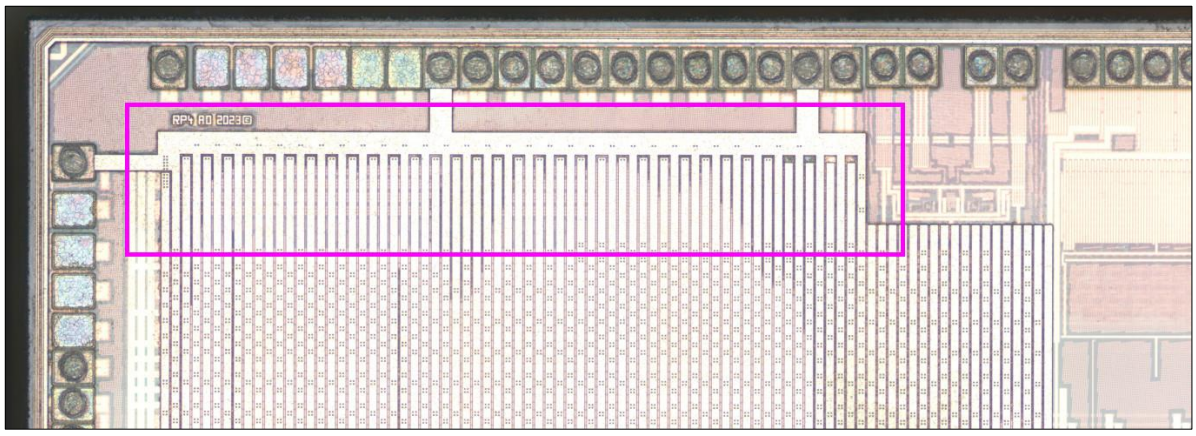


Figure 6. Top metal overview showing the approximate location of the fuse bank. The M8 power grid in this area has no vias (since the fuse bank presumably does not use as much power as the logic core), providing an easy way to identify the approximate region of interest during deprocessing.

The entire memory array, including row/column logic but not the high voltage generation block, is approximately 617 x 126 µm (0.078 mm^2) and consists of 24 vertical tiles, 12 on

[3] https://www.synopsys.com/dw/ipdir.php?c=dwc_nvm_ts40np5sxxxxh0nopxxxi

[4] The fuse IP is offered in variants for each of the many flavors of TSMC 40 nm processes, with subtly different power/performance tradeoffs (GP, LP, ULP, etc.). These process variants are difficult to distinguish without extensive measurements of transistor characteristics, doping strength, layer thicknesses, etc.; however, the differences are of no importance to an attacker who only wishes to extract fuse data.

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

either side of a central spine. Each of these tiles is one bit plane (i.e. each tile is logically 4096 single-bit rows and contains the Nth bit of every word in the memory).

The eight western-most bit planes store the ECC data; the remaining 16 store the user fuse values. Bit ordering is sequential with bit planes 0-3 west of the center line and 4-15 east of center; note that this does not match the order in which the 24-bit values are printed by picotool (ECC bits at positions 23:16 followed by data bits at 15:0). In the physical layout, the ECC bits are adjacent to data bit 0 rather than 15. Bit planes in the east half of the array are mirrored relative to those in the west (i.e. column addressing within the plane is inverted).

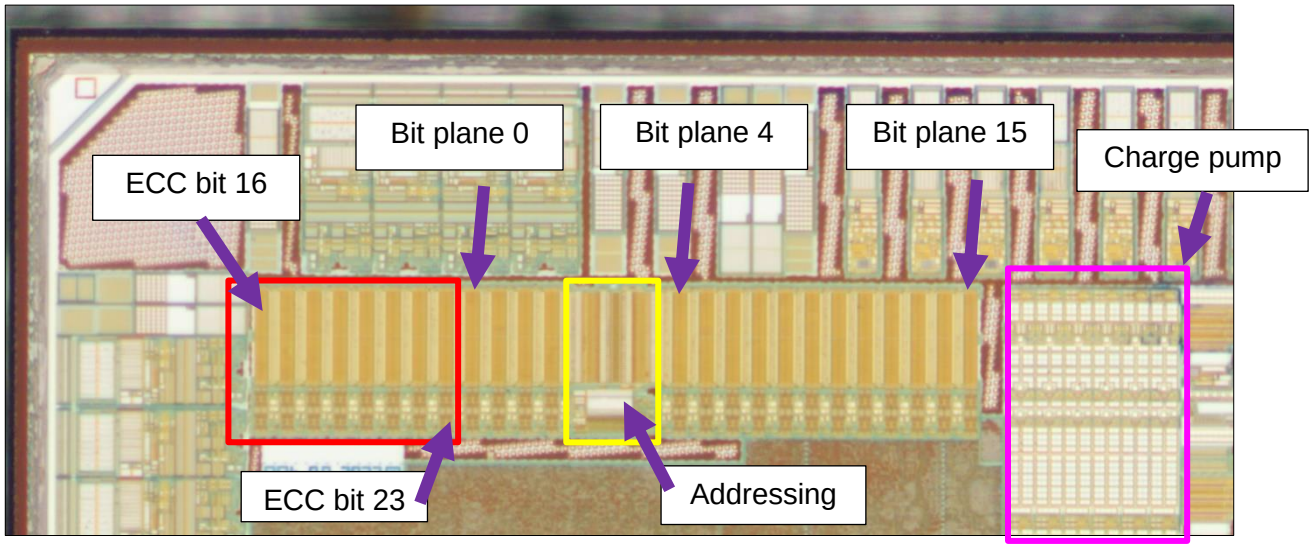


Figure 7. Overview of the fuse array at substrate level, showing the 24-column structure

Zooming in closer, we can see each bit plane is symmetric about the center line and has some column logic duplicated at the north and south side, while some is present only at the south. Row logic runs vertically between each pair of bit planes, with additional row logic present at the center of the array.

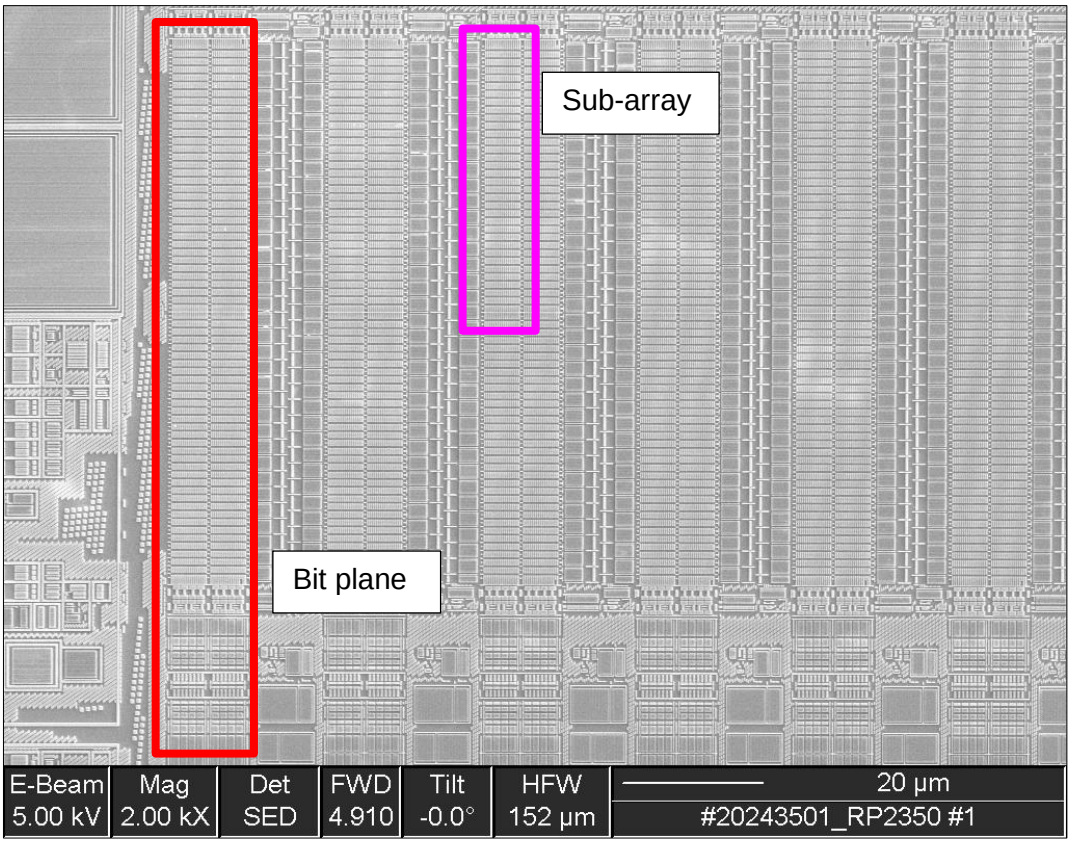


Figure 8. A single bit plane of the fuse array at the substrate layer after removing all metal and poly

Each bit plane consists of a 2x2 tile of identical sub-arrays, with what appears to be a timing or calibration dummy bit cell at the center of the array.

Each sub-array consists of 18 columns of bit cells (16 active bit cells that are electrically functional and one extra column of dummy cells on either side for lithography purposes). The far north and south ends of the array have dummy bit cell rows as well, similar to those found at the center of the array; however, the dummy bit is missing in the second eastmost column of each bit plane. This row appears to be some sort of timing or calibration feature that has similar drive strength and loading behavior to a fuse bit cell but always reads as a constant value.

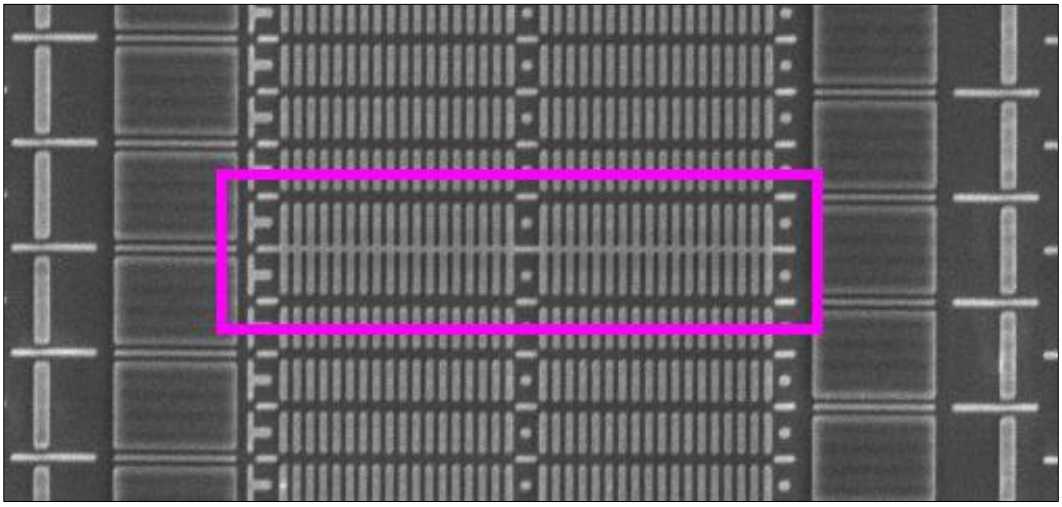


Figure 9. Substrate view of the midline of a single bit plane with dummy bit cell rows marked

Each sub-array consists of 32 rows of cells, for a total of 32 columns x 64 rows in the entire bit plane. This is 2048 cells, suggesting that the actual unit cell is one-half of the visible feature (since the bit plane must have a total capacity of 4096 bits). This was later confirmed by circuit analysis.

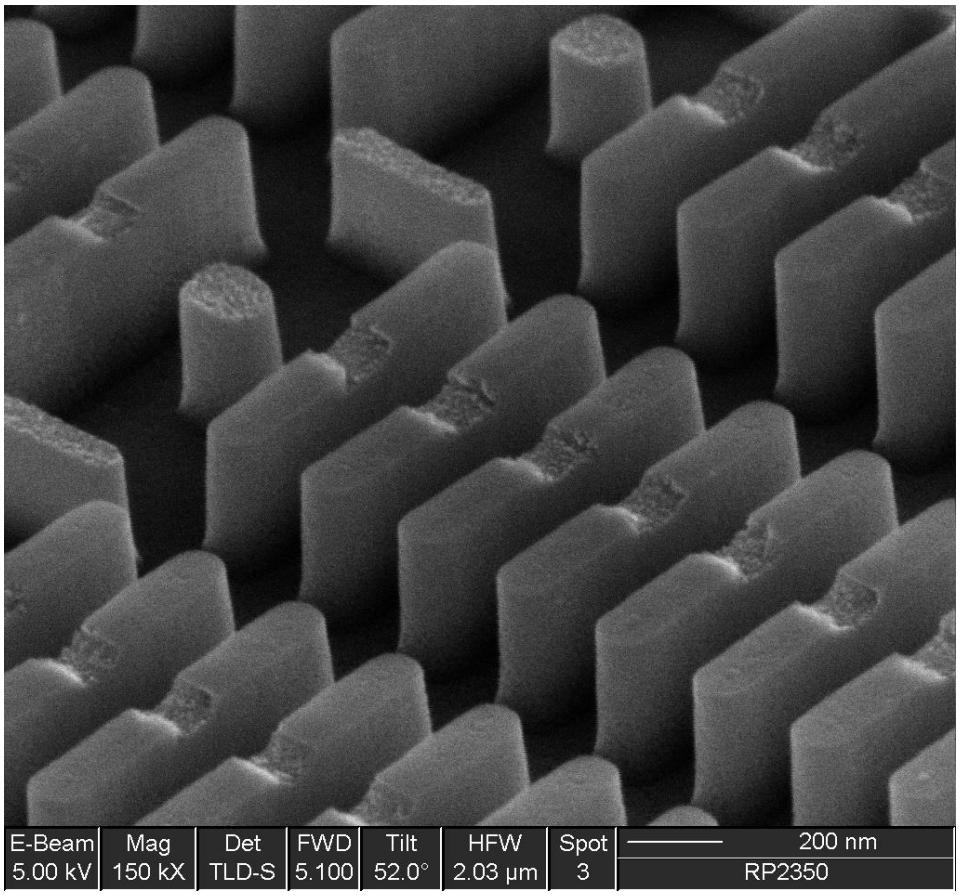


Figure 10. Angled substrate view of fuse bit cells

The unit cell is approximately 571 x 438 nm (0.25 μm^2). It is based on dielectric breakdown for programming and can be irreversibly set from a logic 0 to 1 state.

Synopsys marketing material[6] refers to the cell as a 1T (single transistor) bit cell; however, this is perhaps an oversimplification: it consists of an N-channel MOSFET using a thick (same as used in the I/O pads) oxide layer between the gate and channel. The source contact is the same as an ordinary transistor. The drain end, however, is very different: rather than a contact down to a N+ implant in the silicon, the N+ doped polysilicon gate overhangs the end of the channel with thin (same as used in core logic) oxide layer between the gate and the channel. When the cell is programmed, this overhanging poly becomes the drain of the transistor.

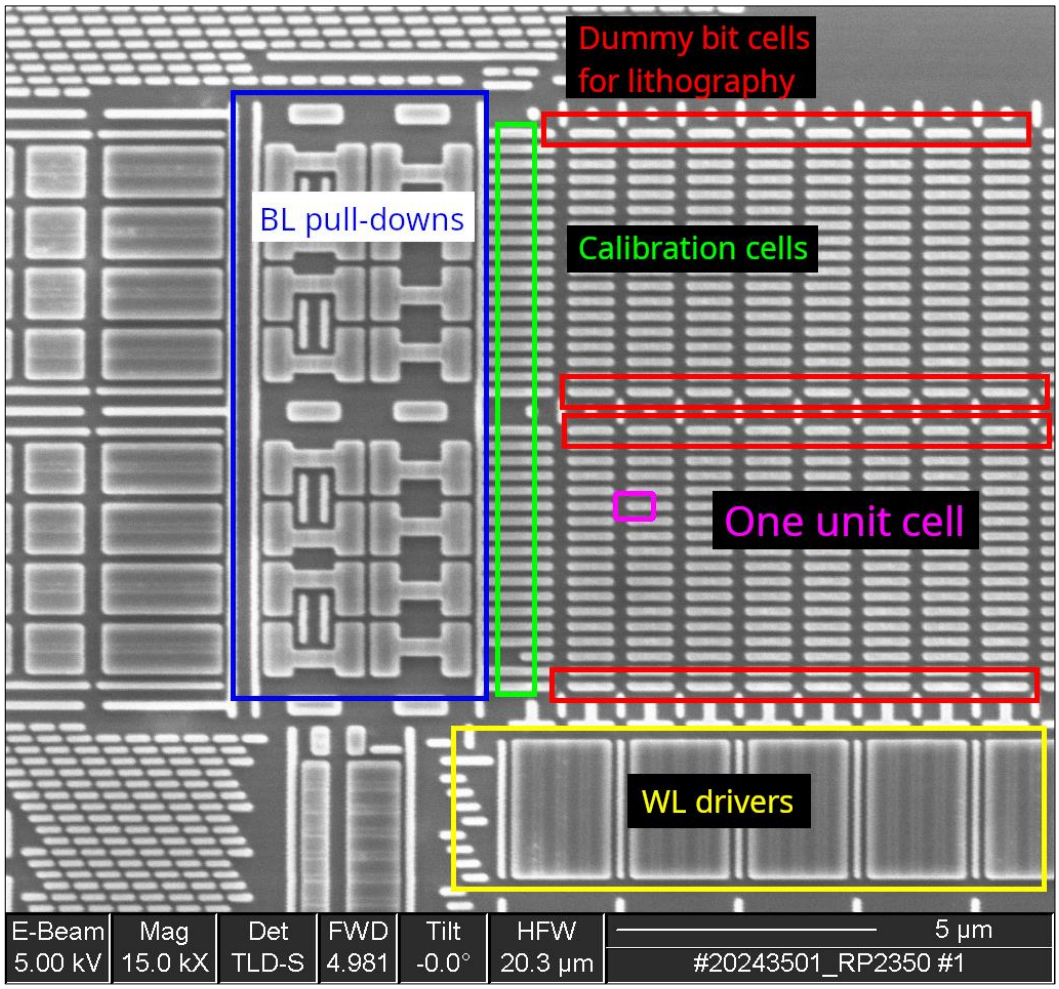


Figure 11. SEM image of die substrate (all metal and poly removed) showing south end of the bit cell column and wordline drivers. North is to the right. Note missing bit cell in the calibration row.

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

The polysilicon word line (WL) connects to a parallel metal track on metal 2 (M2) running east-west, while the bit line (BL, connected to the source of the transistor) connects to a metal track on metal 1 (M1) running north-south, as seen in Figure 12 and Figure 13.

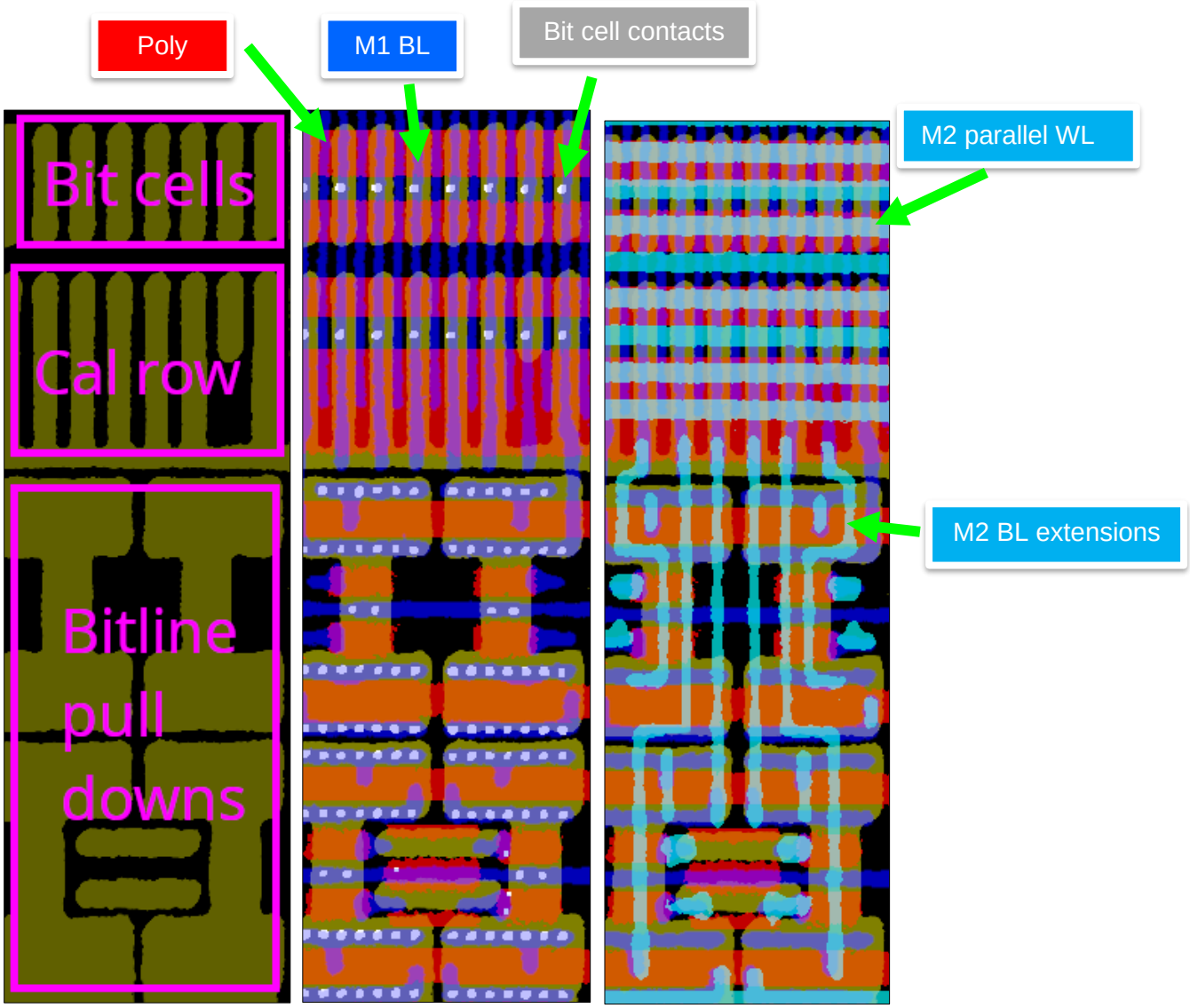


Figure 12. Top-down layout extraction of southeast corner of a bit plane showing substrate (olive), poly (red), contact (white), M1 (blue), M2 (cyan): Note 4-way symmetry about the vertical axis except for the calibration row.

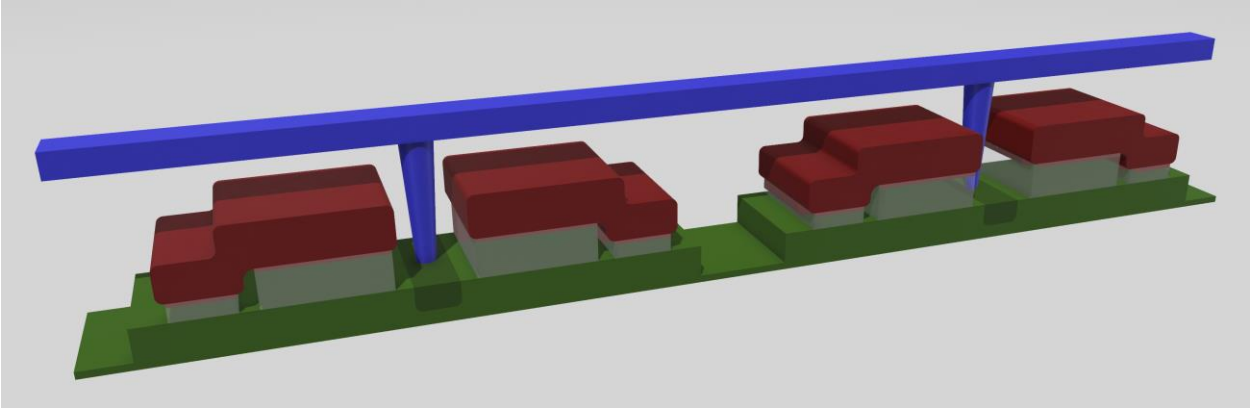


Figure 13. 3D rendering (not to scale) of two rows of cells (four data bits) sharing a common BL. M2 not shown.

To read the cell, the WL is driven weakly high (strong enough to turn on the transistor but not enough to trigger breakdown of the drain-channel dielectric), while the BL is pulled weakly low. If the cell is unprogrammed, or logic 0 (Figure 14), the BL remains low since the transistor is on but the drain is floating.

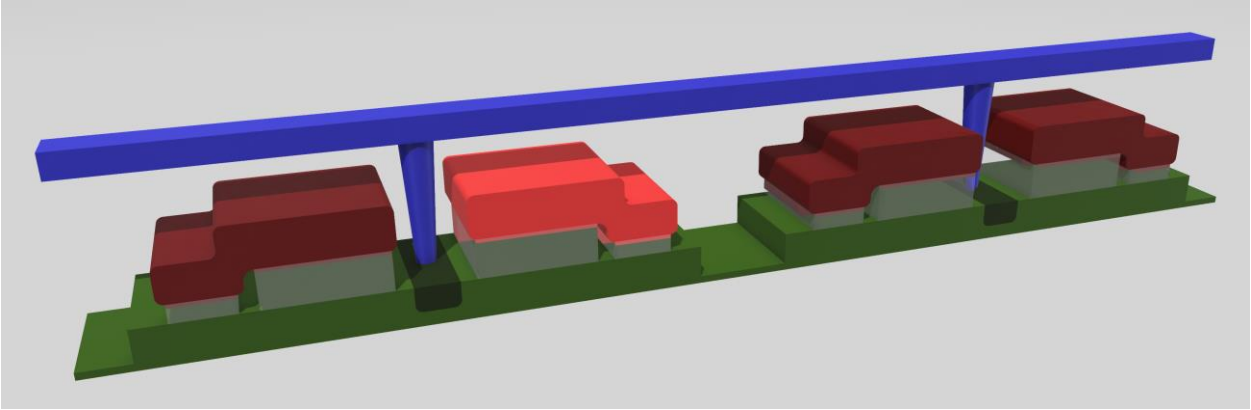


Figure 14. Reading an unprogrammed bit cell. The WL is energized but no current flows because the fuse dielectric is intact.

If the cell is programmed, or logic 1, current flows from the WL through the broken-down oxide, the transistor channel, and into the BL pulling it high (Figure 15).

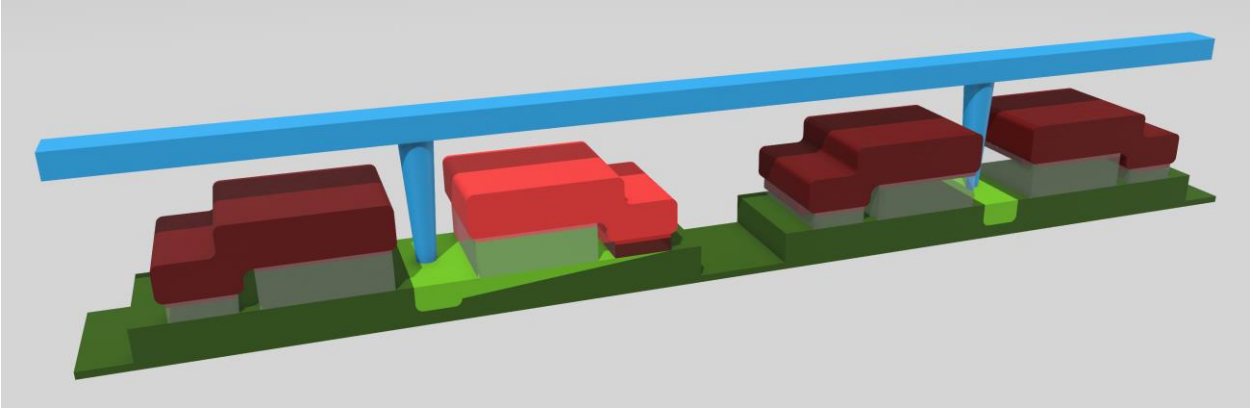


Figure 15. Reading a programmed bit cell. The WL is energized and current flows through the ruptured dielectric, the selected transistor, and into the BL

During programming, the WL is driven to a high positive voltage by an on-die charge pump. The voltage is chosen to be high enough to reliably induce dielectric breakdown in the thin oxide between the drain and channel, while being low enough to not damage the thick oxide over the channel. Since the WL connects to the gate of the transistor in the bit cell, the transistor is in the on state.

The BL is then driven strongly low by a large transistor at the north or south end of the bit cell array. This results in the drain of the transistor (and thus the lower side of the thin dielectric) being strongly grounded, while the poly on the top side of the thin dielectric is at a high voltage. This causes dielectric breakdown, punching through the thin oxide and turning the overhanging poly into the drain of the transistor (Figure 16).

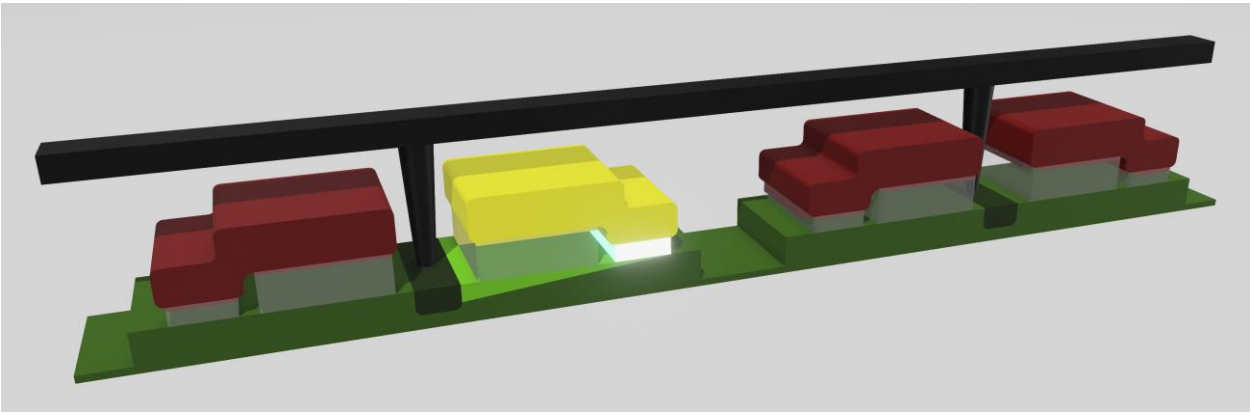
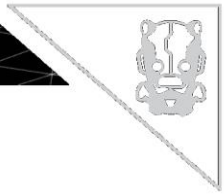


Figure 16. Programming a bit cell. The WL is driven to a high voltage while the BL is strongly grounded. Dielectric breakdown ruptures the insulator between the gate and drain.

Within a bit plane, column addressing broadly increments out from the centerline (i.e. east to west in the west half of the array, and west to east in the east half). Odd-numbered nibbles reverse the bit ordering, since the column address logic consists of mirrored pairs of 4-bit tiles.



Row addressing increments from south to north for the low half of the array (pages 0 to 1f) and north to south for the high half of the array (pages 20 to 3f).

IOActive's Attack

Our attack extracts the contents of the fuses via passive voltage contrast (PVC) in a FIB instrument. FIB systems use a positively charged beam of ions (usually Ga^+ although other elements such as helium and xenon are used in some specialized applications) to scan across the sample and generate an image in a similar manner to that of a SEM. While a FIB can also be used at higher beam current levels to cut wires or deposit metals/dielectrics to perform circuit edits, this capability is not used for the attack. PVC runs at very low beam currents to minimize the amount of undesired sputtering.

By scanning the sample with positively charged particles, positive charges are induced in the regions of the sample that are struck by the beam. These charges gradually decay to ground through various leakage paths; however, while the sample is charged it influences the behavior of the beam and creates visible brightness differences. (While the magnitude of the brightness change can give a qualitative idea of the intensity of the charge on a given circuit node, quantitative voltage measurements are not possible.)

In a SEM, charging of the sample often results in significant deflection of the incident beam, causing the image to "shimmer" and move around. FIB images are largely immune to this effect, since a gallium ion is over 10^5 times heavier than an electron. The secondary electrons released by the beam striking the sample, however, are strongly deflected by nearby charges. This results in positively charged areas appearing dark (since the negatively charged electrons are attracted back to the sample and do not reach the detector), while negatively charged or neutral areas appear brighter.

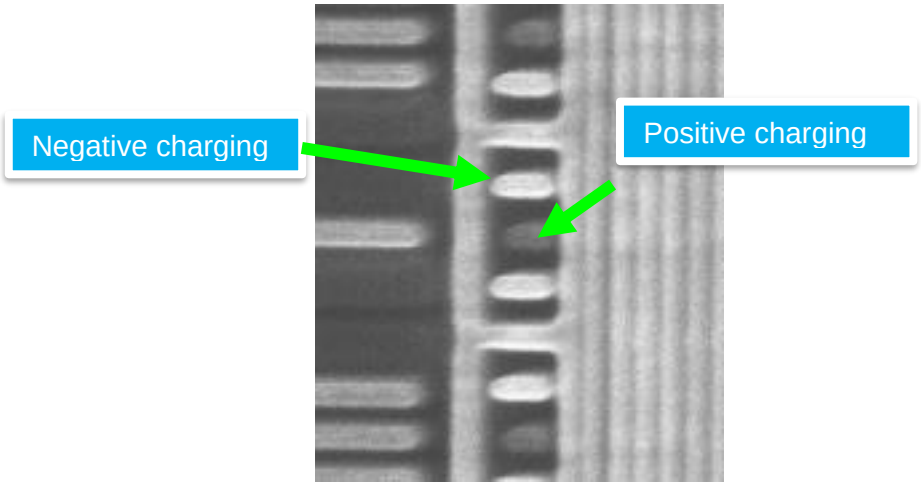


Figure 17: Example of passive voltage contrast in a FIB secondary electron image

While a single-beam FIB can be used for PVC with success, it is typically better to use a dual-beam instrument combining a SEM and FIB. The SEM can be used for nondestructive imaging (e.g. to confirm that the sample has been deprocessed fully to the contact layer without any M1 residue), as well as to inject negative charges to oppose the positive charging of the ion beam.

The technique is called “passive” as the beam is used both for imaging and stimulation, as opposed to active voltage contrast which requires using external hardware (microprobes or similar) to inject voltages into nodes and then observing the resulting image changes with an electron or ion beam. Active voltage contrast is a more powerful technique, although it requires additional hardware and is slower (since physical probes must be aligned to each target on the device) so PVC is typically preferred given the choice.

In order to dump fuses using PVC, the target device is first depackaged, then deprocessed to the contact layer (removing M1 and above) across the entire fuse area. The remainder of the device is not needed, so the attacker does not need to be concerned about scratches, overetching, or other damage to the die outside of the fuses.

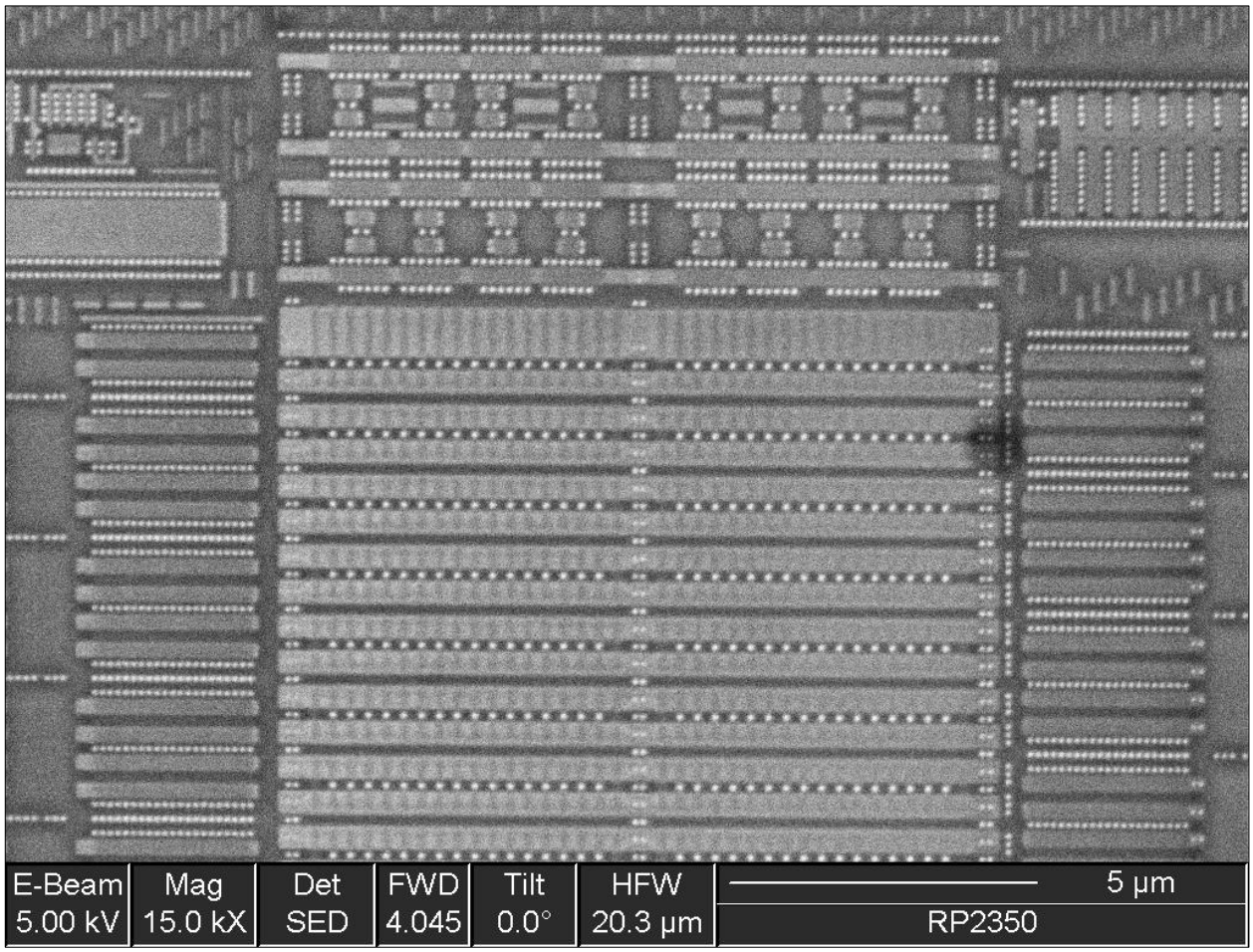


Figure 18. North end of fuse bit plane imaged in SEM after deprocessing to contact layer. Polysilicon can be seen through the interlayer dielectric. No fuse data can be seen in this view due to negative WL bias from the electron beam (all bitcell transistors are off).

While PVC can be performed with an electron beam in an SEM in some applications, e-beam PVC is not suitable for attacking this particular fuse IP because the WLs must be

positively biased in order to turn on the bit cell transistors. Thus, the ion beam is used instead.[7]

Once the sample is deprocessed to the correct layer and placed in the FIB, the only remaining task is to image the array with the ion beam and observe the pattern of light and dark spots, then map this back to actual data bits by reverse engineering the addressing from test chips programmed with known fuse values.

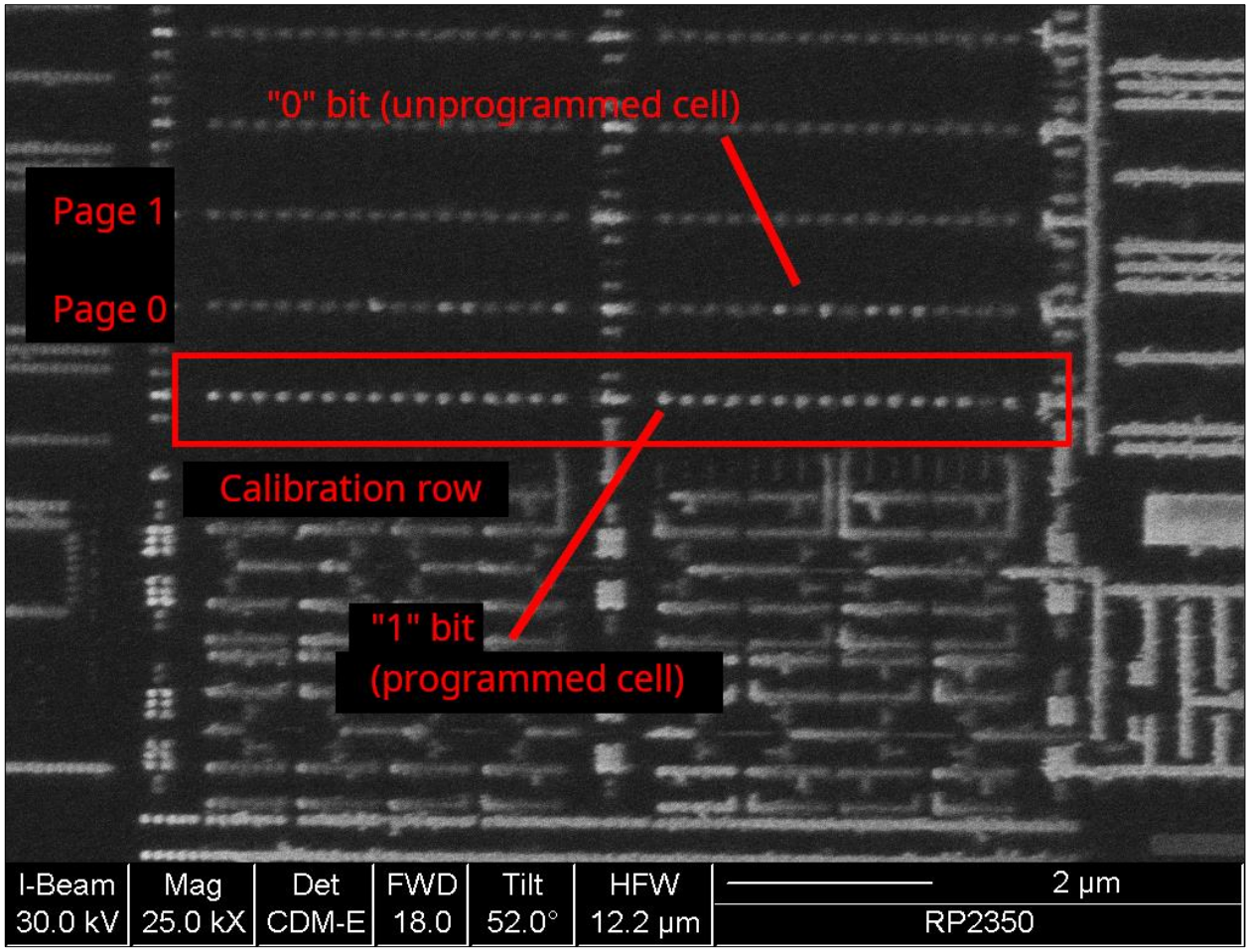


Figure 19. Early PVC test image of a chip in factory-programmed state (no user fuse values). Note trim/serial data values in page 0 while page 1 and up are blank.

After some testing, it was determined that each row of 32 BL vias maps to one page of 64 memory words (one row of 32 tiles, each containing two back-to-back bit cells sharing a

[7] It is possible that SEM active voltage contrast (injecting a positive WL bias with a probe and then using the e-beam to image the resulting charge distribution) would be successful; however, we have not tested this.

single via). In the FIB PVC images, logic 1 bits show up as bright while logic 0 bits show up as dim[8].

Each page contains two WLs, one to select the upper row of bits and one to select the lower. If both WLs are floating, they will both charge up when the beam contacts them and the observed pattern of light and dark vias will be the bitwise OR of the high and low 32 bits in the page (i.e. bits 63:32 OR bits 31:0). While this reveals less information than a full memory dump, it is likely to be sufficient for dumping keys (either because the adjacent memory rows are blank or because it reduces the search space to be small enough to tractably brute-force). This also enables a rapid scan of the entire fuse address space to locate programmed (thus interesting) regions.

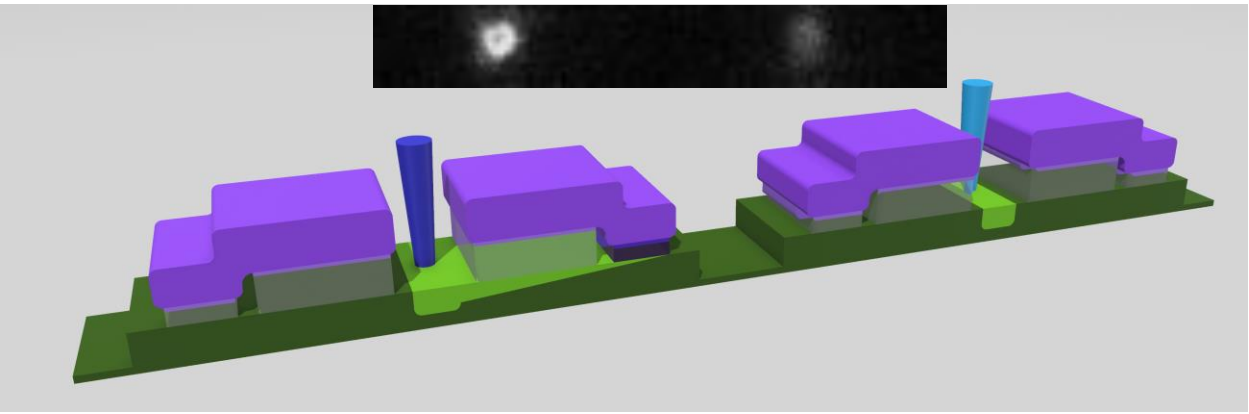


Figure 20. Mass readout of bit cells as ORed-pairs by FIB PVC. All WLs are energized simultaneously by the beam, but only to around V_t. The left pair of bit cells has one programmed and one unprogrammed bit, so the via is pulled to ground (shows light in PVC). The right pair has no programmed bits, so the via floats high (shows dark in PVC)

[8] We found this counterintuitive, as during normal operation of the memory array logic 1 bits drive a *high* voltage onto the bit line while logic 0 bits allow the bit line to float *low*. This would result in logic 1 bits showing up as dark in the PVC image, the opposite of what is actually observed.

Our working hypothesis is that the ion beam charges the wordline up to roughly V_t of the bitcell transistors, before reaching an equilibrium in which leakage current to ground through the bitcells, wordline drivers, etc. is equal to the 10 pA injected current from the beam. In this state, the wordline voltage is high enough to turn the bitcell transistors partially on, but not so high that it appears dark in the PVC image.

When the beam strikes the bitline contact for an unprogrammed bit, the injected charge has no escape path and the via becomes strongly positive, showing up dark in the PVC image. If the bit is programmed, however, the injected charge flows out through the bitcell transistor, into the wordline, and ultimately to ground. This keeps the via's voltage fairly low, showing bright in the PVC image.

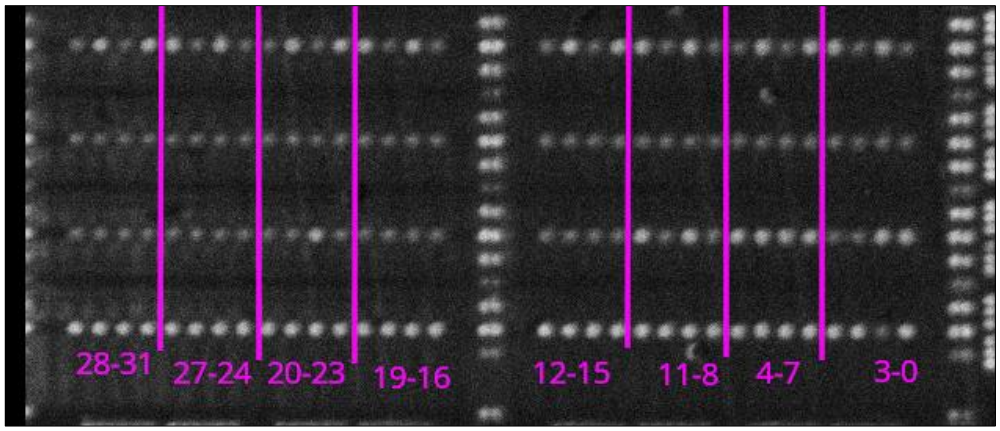


Figure 21: PVC image of factory trim values from early readout test before optimizing the process. Contrast is poor but data is still recoverable.

Figure 21 shows one of the readout tests conducted while reverse engineering the address map.

Page 0 (factory trim data) values for this chip were (row 0 to 63):

```
1af81d, 262f1b, 0adde8, 130b3a, 31fdf3, 0a07a7, 1813b3, 04c47d,
338784, 211c75, 2d9768, 3e0ee1, 000000, 000000, 000000, 000000,
175dae, 09ddb4, 000000, 000000, 000000, 000000, 000000, 000000,
2d001e, 000000, 000000, 000000, 000000, 000000, 000000, 000000,
000000, 000000, 000000, 000000, 000000, 000000, 000000, 000000,
000000, 000000, 000000, 000000, 000000, 000000, 000000, 000000,
000000, 000000, 000000, 000000, 000000, 000000, 1ae31d, 13fb60,
000000, 000000, 000000, 000000, 000000, 000000, 000000, 000000.
```

Extracting the LSB from the high and low 32 words and ORing them together, since both WLs are energized in this test, gives (MSB to LSB) 00000000 01000000 00001010 11110011.

Reading the pattern of light and dark vias for page 0, and flipping the bit ordering of odd numbered nibbles, gives the same bit sequence: 00000000 01000000 00001010 11110011. This confirms successful data extraction.

Repeating the same descrambling, the test vectors written to pages 2 and higher (repeating patterns of 01010101, 00110011, 00001111 every three pages) can be seen and decoded as well.

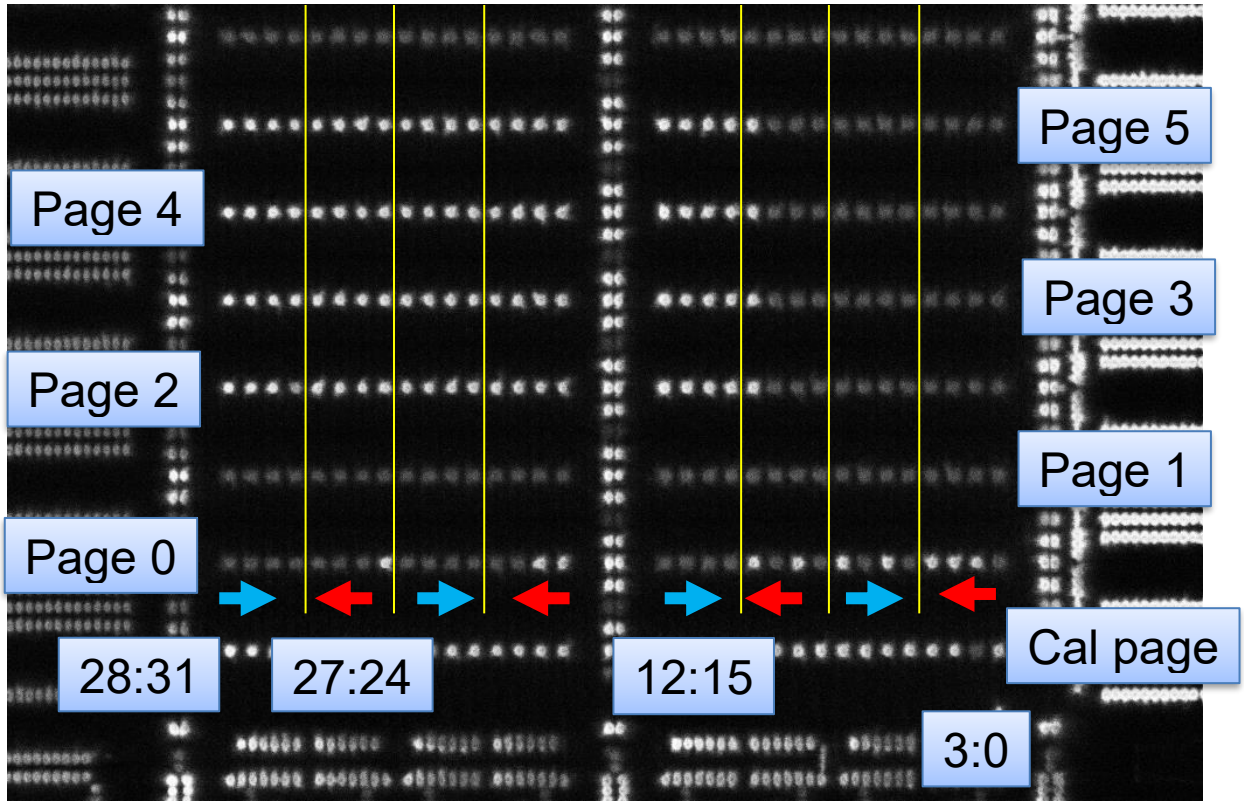


Figure 22. Bit 21 plane from a test device annotated with bit/page ordering for the west side of the array. Bit planes east of the center line (bits 4-15) are mirrored so that row 0 is at the west side and addresses increment to the east; bit planes west of center (0-3 and ECC 23-16) use the bit ordering shown here.

This sample has all WLs energized by the beam so the bitwise OR of low and high halves is displayed.

This same process was used to determine the ordering of all of the other bit planes in the device and a Python script was developed which allows linear fuse data dumps to be converted into ASCII art renderings of the expected PVC image in physical address order. A second script, allowing a physically addressed bitmap to be turned into a series of fuse values, was developed to simplify the generation of test patterns.

If one WL is tied to ground (by FIB platinum, a probe needle, etc.) while its counterpart floats, we expect that the contents of one half of the row can be read out without interference from the other, followed by repeating the process to read the opposite half, thus obtaining a dump of every bit cell at the cost of additional attacker effort.

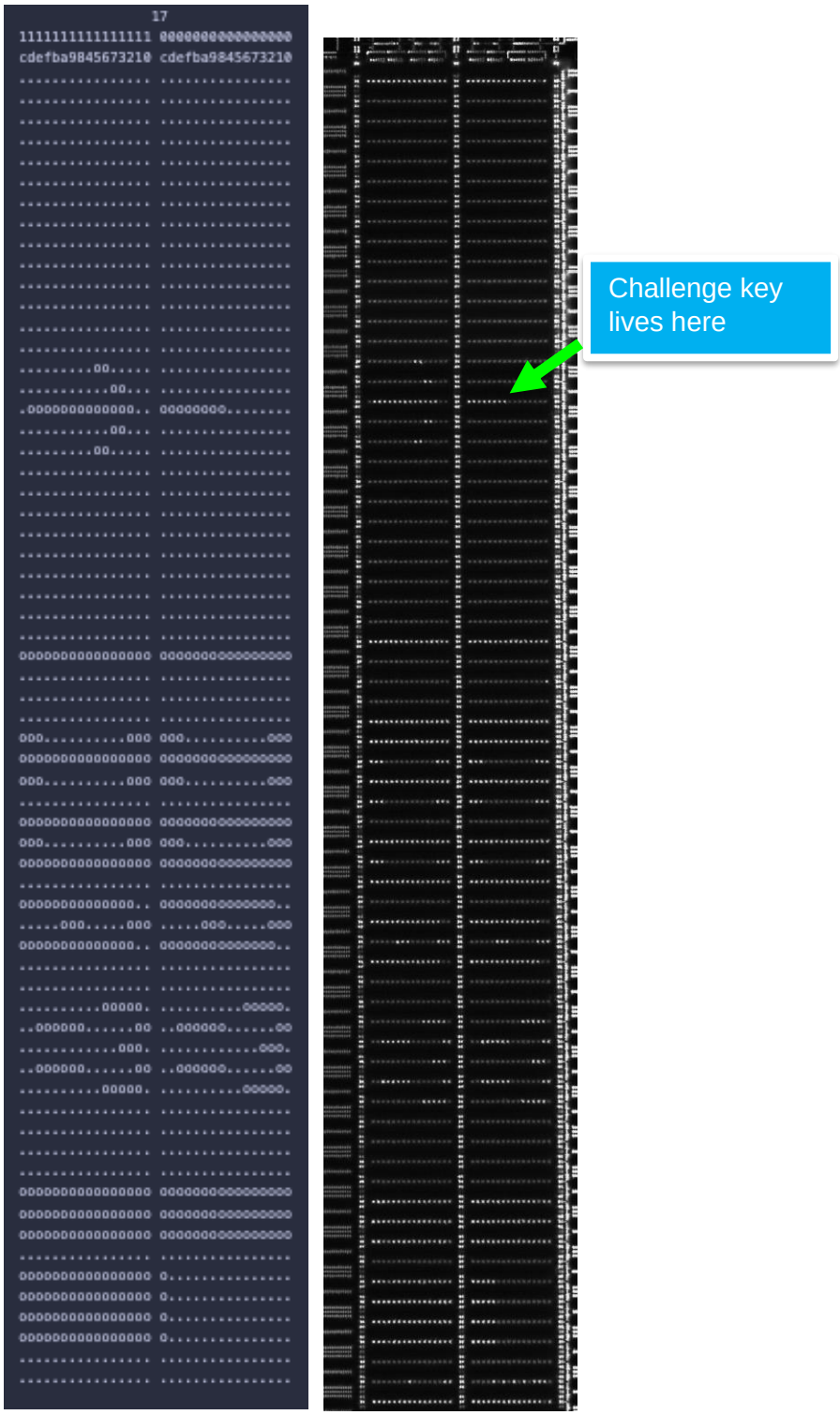
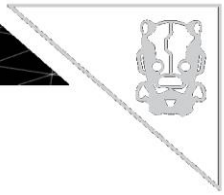


Figure 23. ASCII art rendering of bit plane 17 (left) and FIB image of bit plane 21 (right) from test device.

The patterns shown here include "IOA" logo text, an ASCII art "=3" cat face, several unique bit patterns for reverse engineering the bit plane ordering and testing various readout techniques, and an arrow pointing at the physical location of the 0xc08-c0f fuse rows storing the challenge key.



Recommendations

Voltage contrast is a powerful failure analysis technique that is difficult to entirely prevent. Active shielding and similar techniques pose little impediment: since PVC is used to directly extract individual memory rows rather interacting with than a functioning chip, all long-range interconnect is gone.

Silicon Designers

Our general recommendation to silicon designers is that long-lived secrets not be stored in cleartext in physical memory cells (fuses, mask ROM, or flash). If the fuse content were transparently encrypted at rest by logic outside the memory, a fuse dump would be of little value to an attacker unless they also had extracted and reverse engineered the circuitry performing the encryption—a significantly more difficult and expensive prospect.

RP2350 Users

Users of the RP2350 can mitigate the basic form of the attack by using a “chaffing” technique taking advantage of the paired nature of the bit cells. By using only the low half (words 0-31) or high half (words 32-63) of a page to store keys, and storing the complement of the key[9] in the opposite half of the page, each pair of bit cells will have exactly one programmed and one unprogrammed bit. Since the basic PVC technique cannot distinguish between the two bits sharing the common via, the attacker will see the entire page as being all 1s.

This mitigation does not provide complete protection, however: by taking advantage of the circuit-edit capabilities of a FIB, an attacker could likely cut or ground the word lines being used for chaffing and then use PVC to read only the key bits. We intend to explore this extended attack in the future but have not yet tested it.

[9] It is important to store the inverse of the full 24-bit pattern including ECC bits in the bit cells used as chaff (i.e. the ECC bits of the chaff pattern must be the inverse of the ECC bits of the key words). This may produce an invalid ECC code value, thus the 24-bit non ECC address range must be used when programming.

