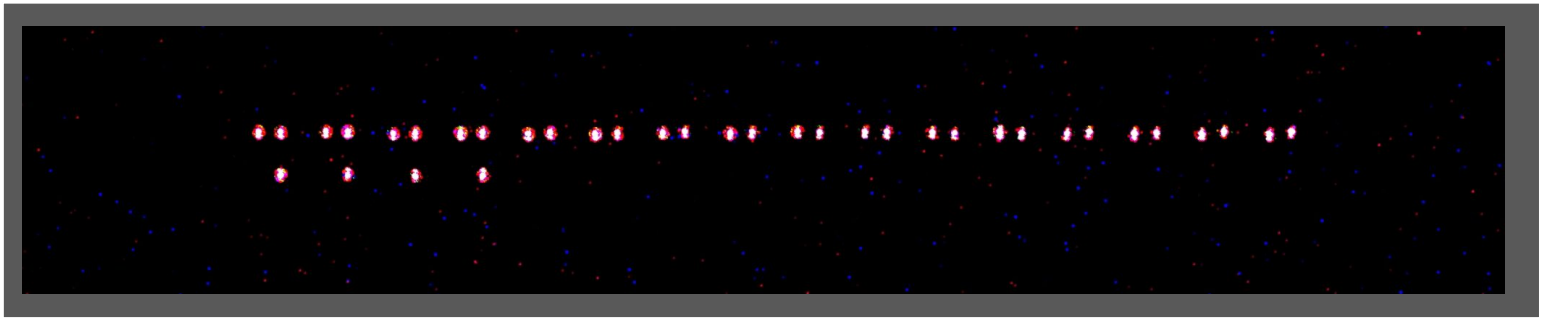
hardwear.io

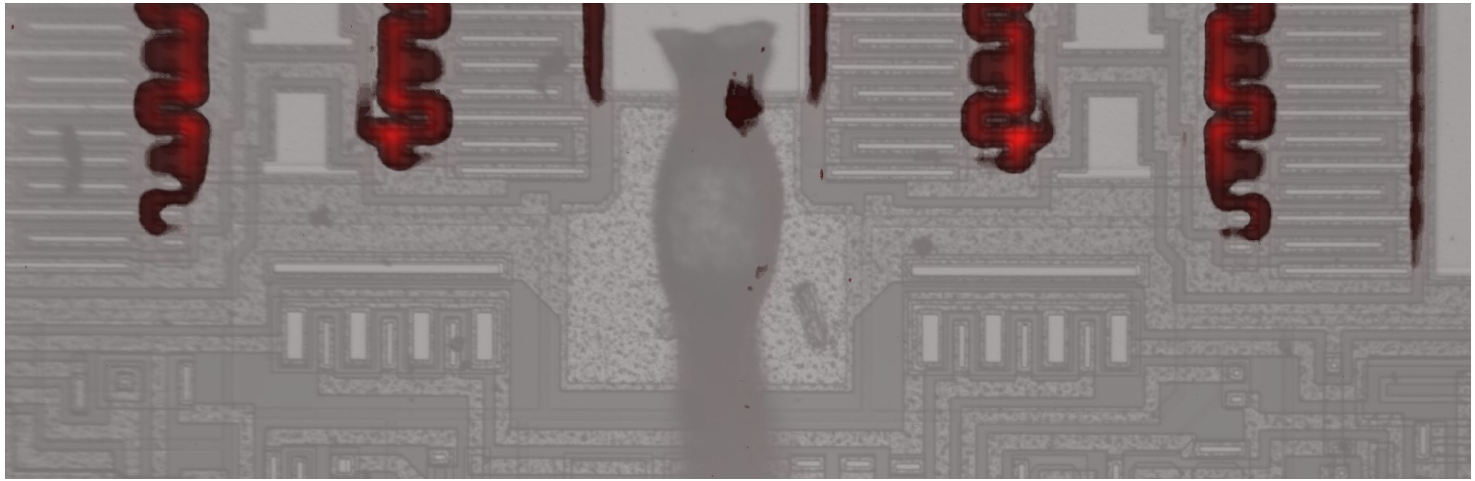
Extracting programmable logic with photons



John McMaster
McMaster Consulting

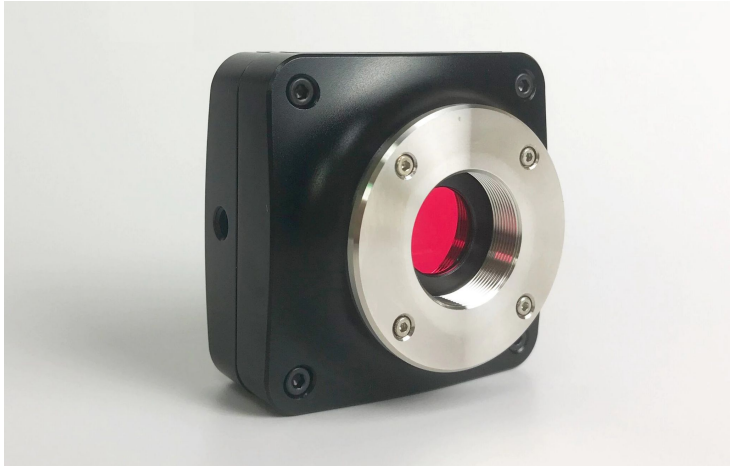
Infrared (IR) emissions

- Emitted during electron recombination
- Emitted at the bandgap of silicon: ~1.14 eV => 1088 nm
- Can correlate emissions to circuit activity



Previous talk: use a sensitive $500 CMOS camera

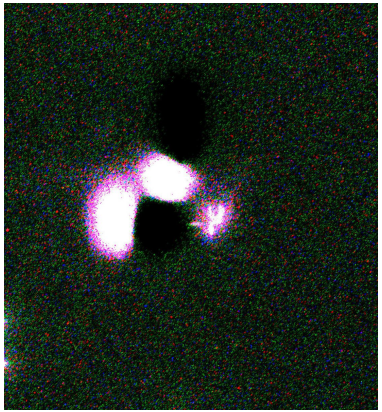
- Back illuminated CMOS
- Remove IR block filter, install IR pass filter



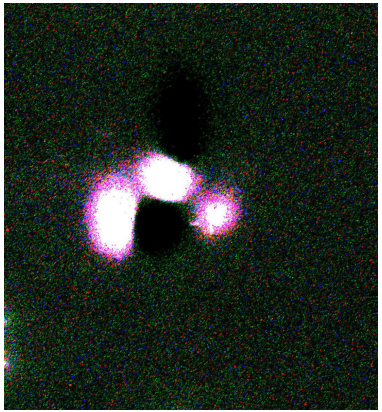
E3ISPM20000KPA

Previous talk: RE bipolar logic based on emissions

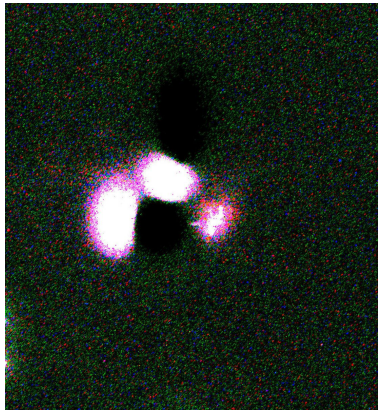
- 2 input NAND gate



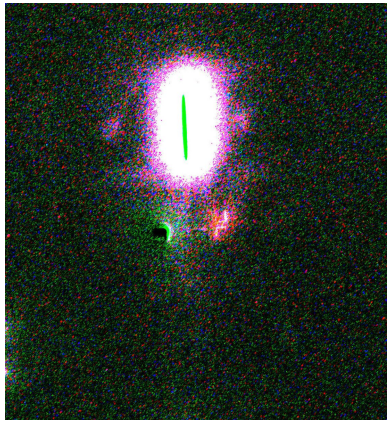
I1: L, I2: L
O: H



I1: H, I2: L
O: H



I1: L, I2: H
O: H



I1: H, I2: H
O: L

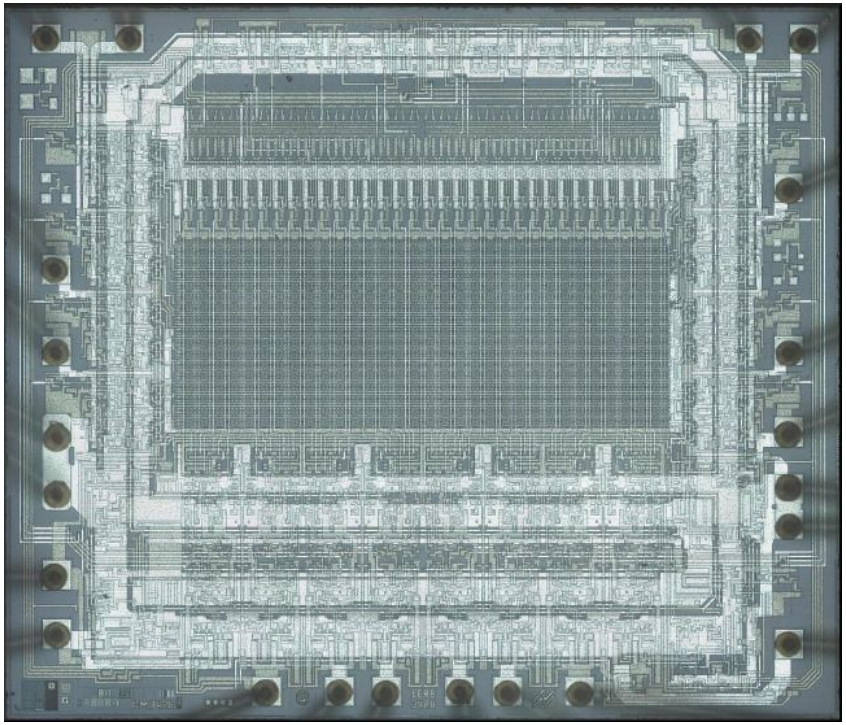
This is all good but...

- Research / theoretical on simple devices
- How can we apply this in the real world?
- Get creative: what can we do outside of cryptography?

TrackStar documentation project

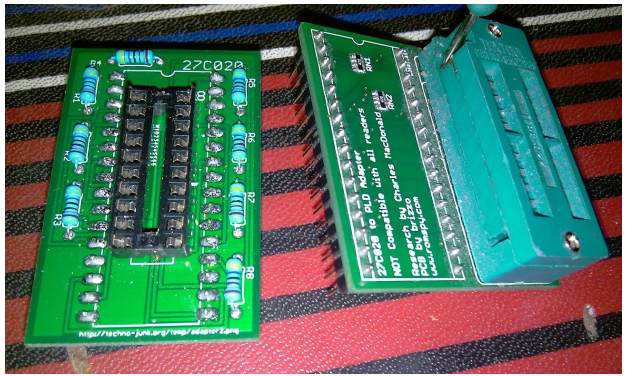
- TrackStar 128 allows running Apple][programs on a PC
- Crazy! How does it work?
- Problem: logic devices are protected. What next?

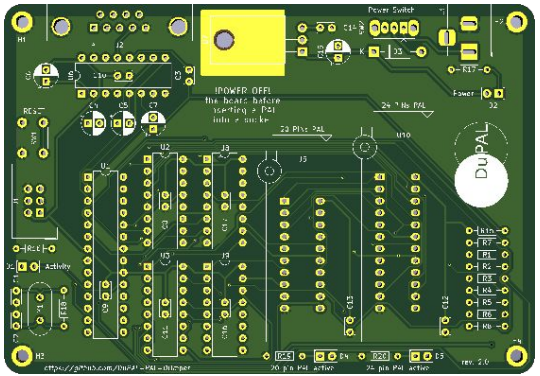
PAL16R8 (DM3428)



Brute forcing PAL devices

- PAL's are relatively simple. Why not just apply all the input permutations?
- Existing projects: readpal, DuPAL, etc
- Played with pal866 (open-tl866) to avoid custom hardware
- May produce equivalent firmware, not the original

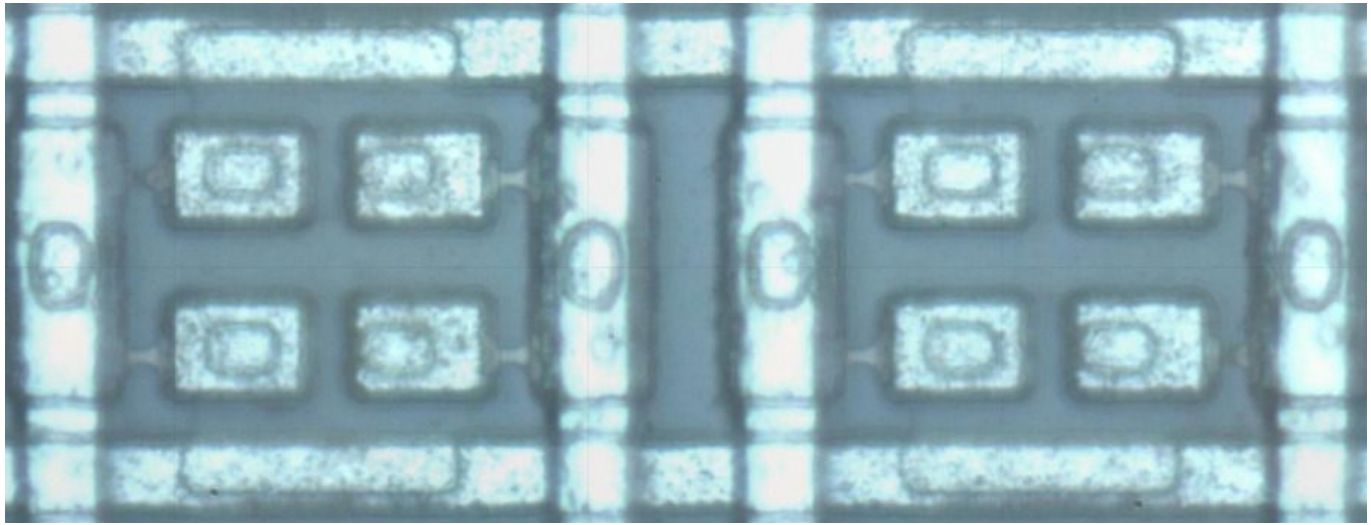




https://github.com/DuPAL-PAL-DUmper
https://www.arcade-projects.com/threads/help-me-solve-an-ssv-mystery.12138/

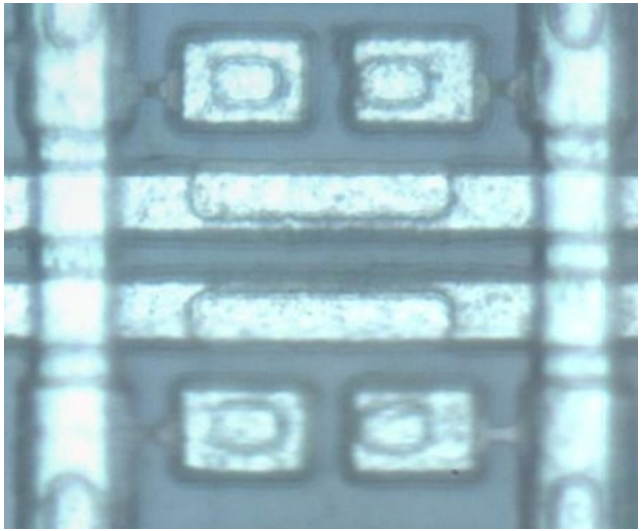
Can it be treated like a mask ROM?

- Yes! Fuses are visible under microscope
- Use existing rompar etc workflow
- But what is the memory layout?



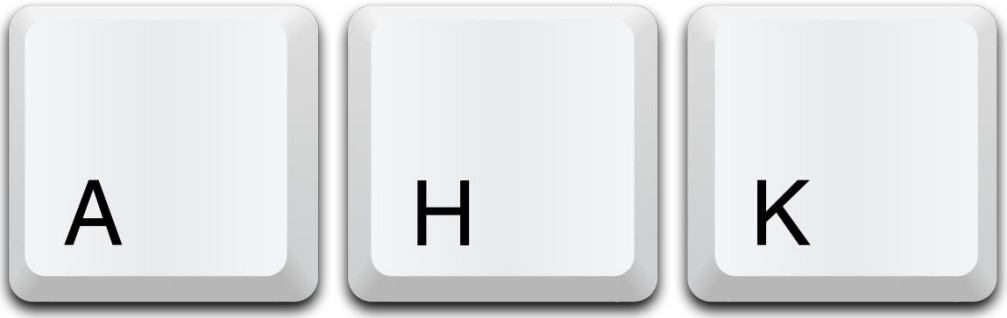
Emissions to the rescue

- Blowing fuse => massive light emission
- Blow one fuse at a time to map memory layout



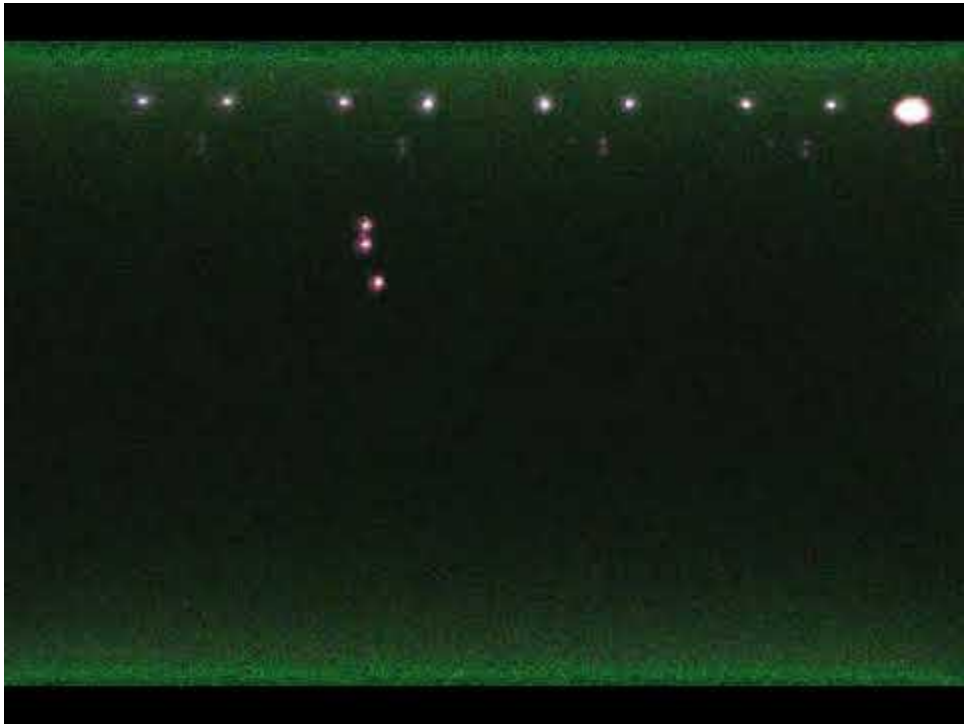
Automation

- AutoHotkey script drives BP Microsystems GUI
- AutoHotkey TCP server takes JSON from Linux Python program
- Camera in time lapse mode
- Python script synchronized to frame saving

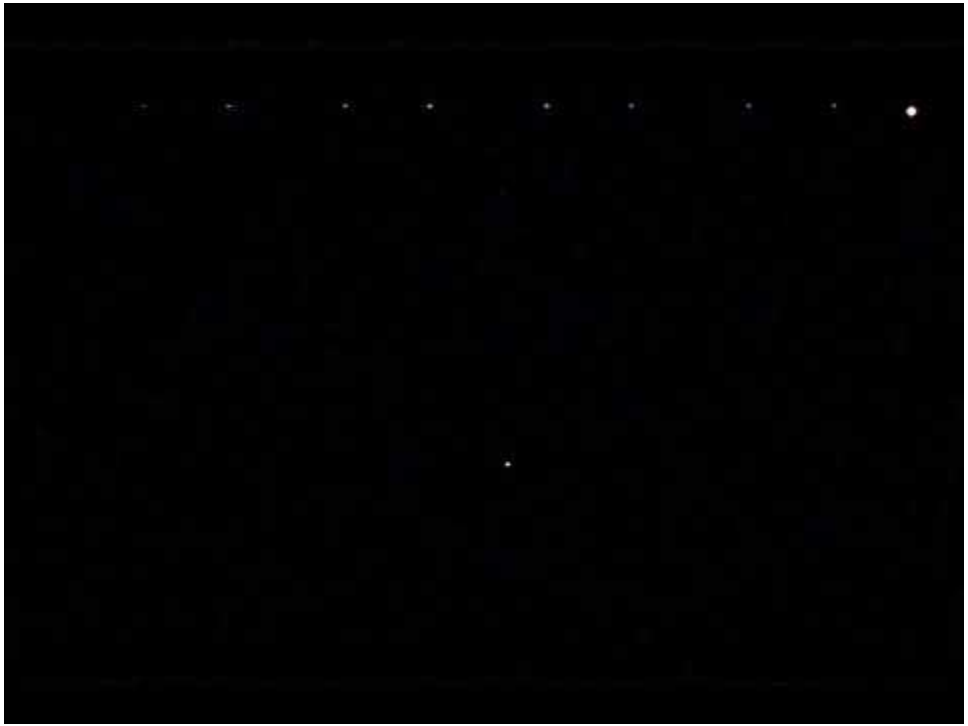


Automation

Abracadabra! inter-word layout

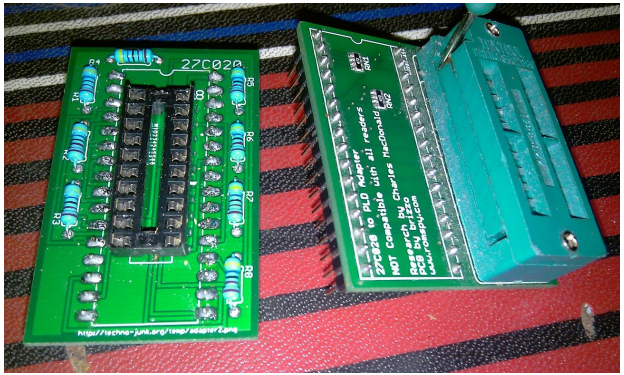


Abracadabra! intra-word layout (word 30)



Verification

- MCU mask ROM => disassemble
- Logic => valid circuits?
- Convert .JED to verilog, compare against test waveforms



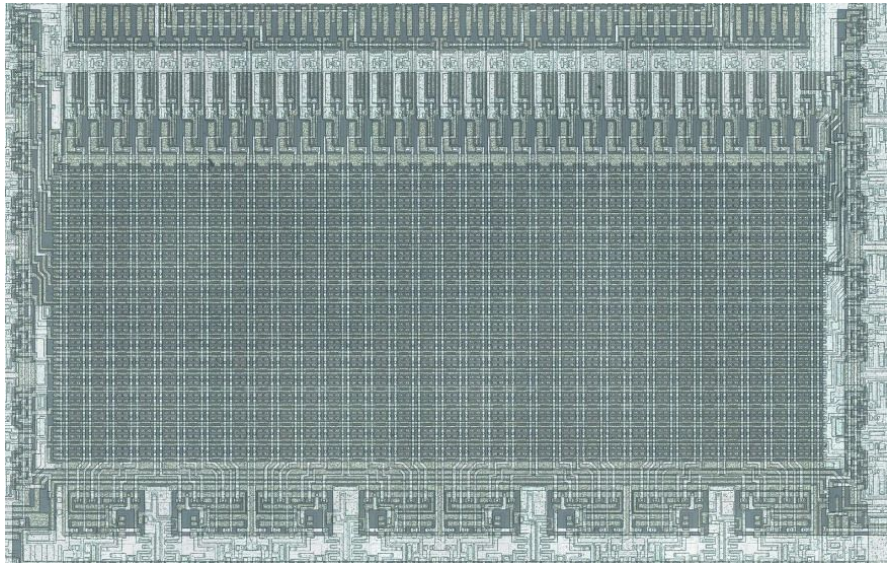
https://www.arcade-projects.com/threads/help-me-solve-an-ssv-mystery.12138/

Putting it together

1. Collect test vectors from PAL device
2. Decap PAL device
3. Photograph PAL
4. Convert photographs into bit matrix (manually, rompar, etc)
5. Convert bit matrix to JED using custom Python program
6. Convert JED into Verilog (MAME?)
7. Verify test vectors against Verilog
8. Pretend it worked
9. Do something better

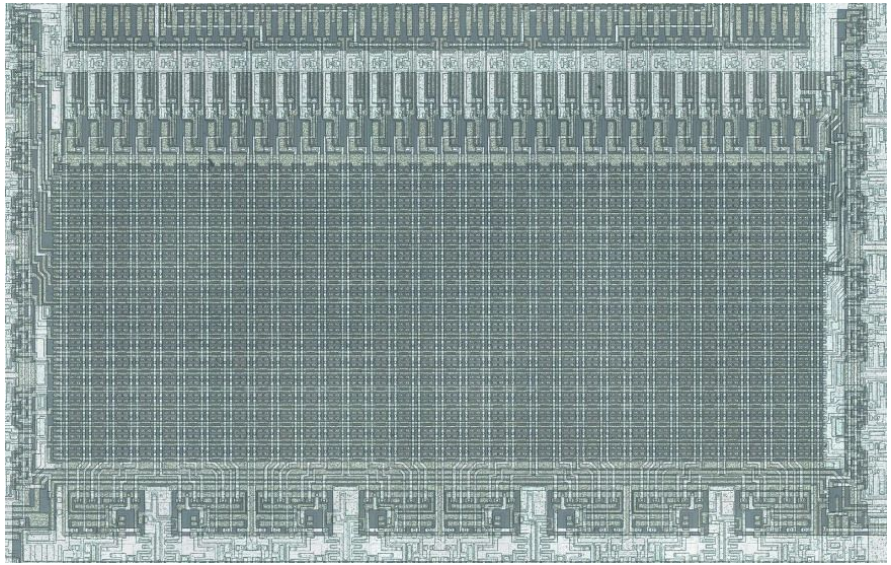
Can we do better?

- If we can instantly locate a fuse, why not just locate the security fuses (2)?
- Theory: they are related to the extra fuses along the edges



Disappointment

- No security fuses visible while burning :(
- Maybe hidden under metal?



Datasheet to the rescue

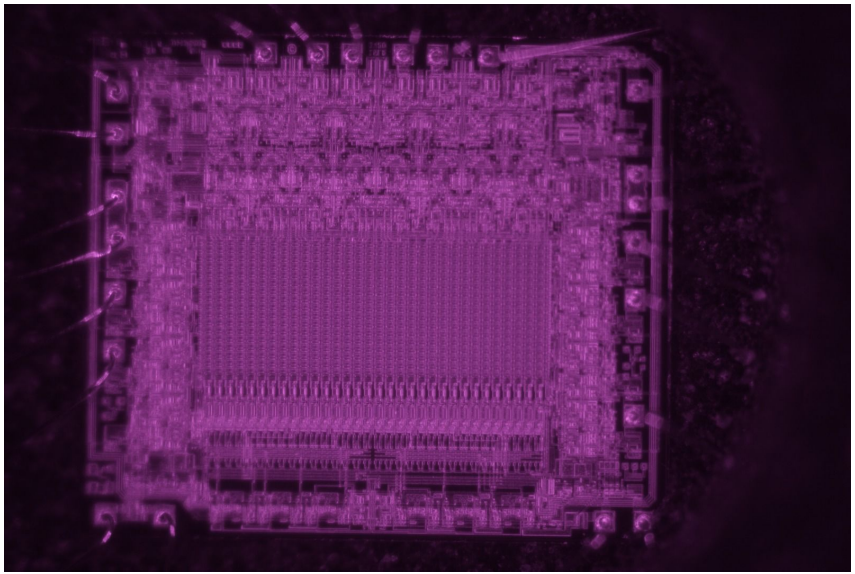
- 1986_National_Programmable_Logic_Design_Guide.pdf

  "**Note 2**: It is recommended that precautions be taken to minimize electrostatic discharge when handling and testing this product. **Pins 1 and 11 are connected directly to the security fuses...**"

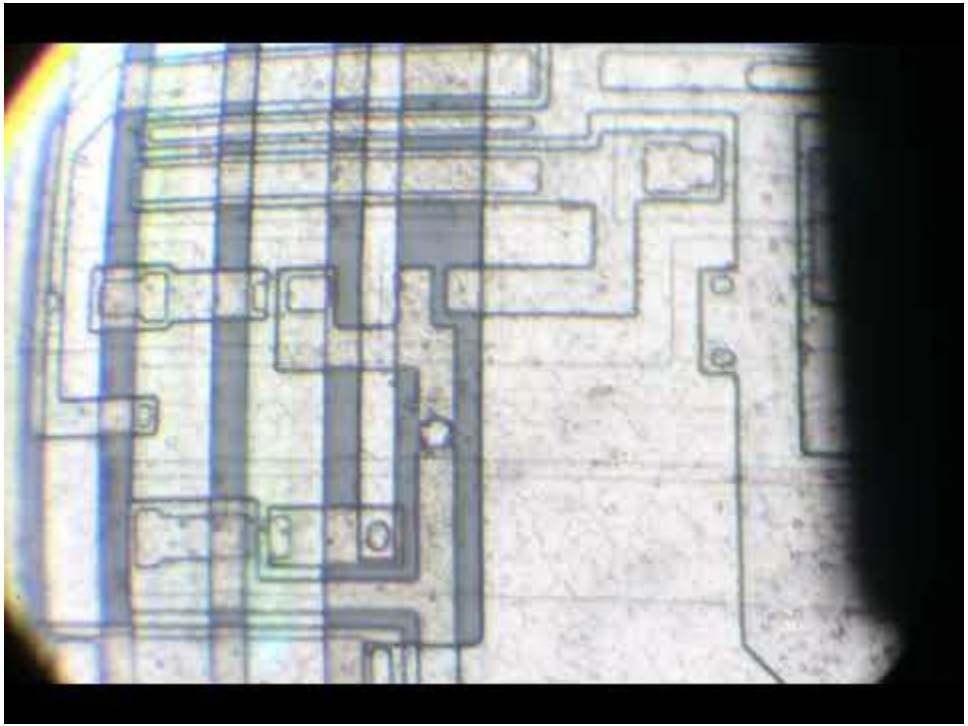
- Let's try again!

Viola!

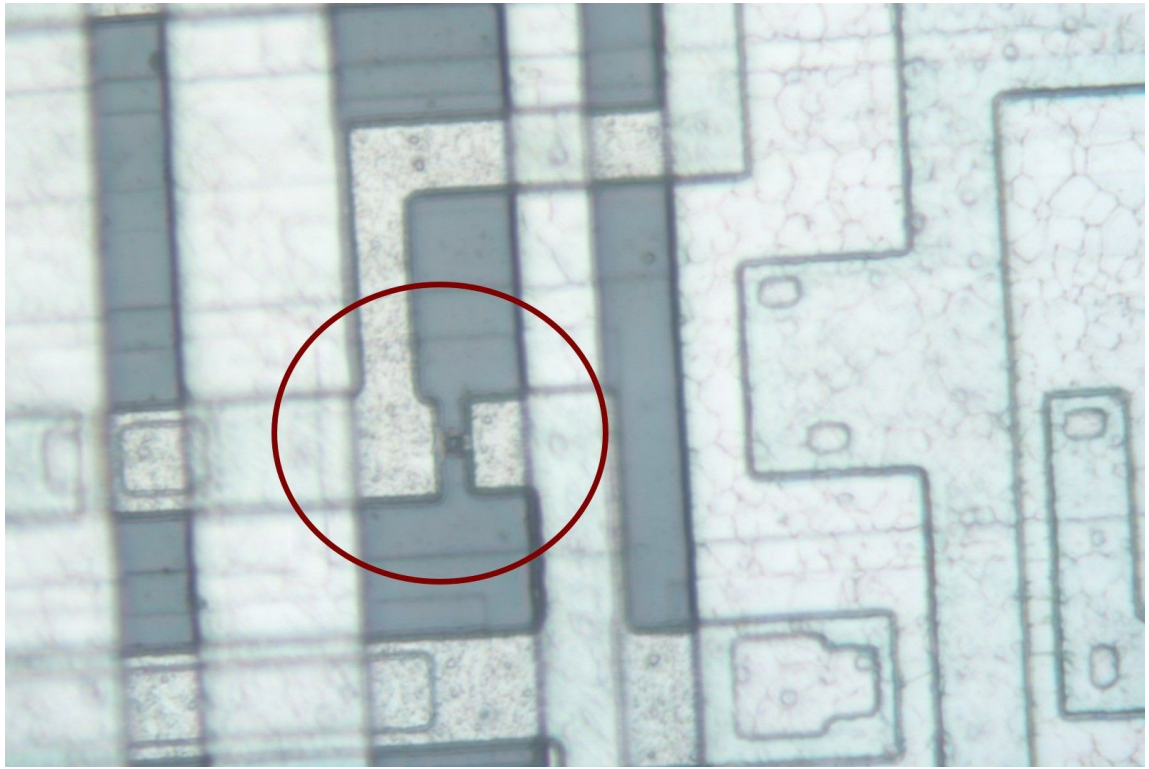
- Two bright spots indicating security fuses



Can we see it burn in real time?



Secured fuse close up



What next?

- Microprobe security fuses to remove protection
- How:

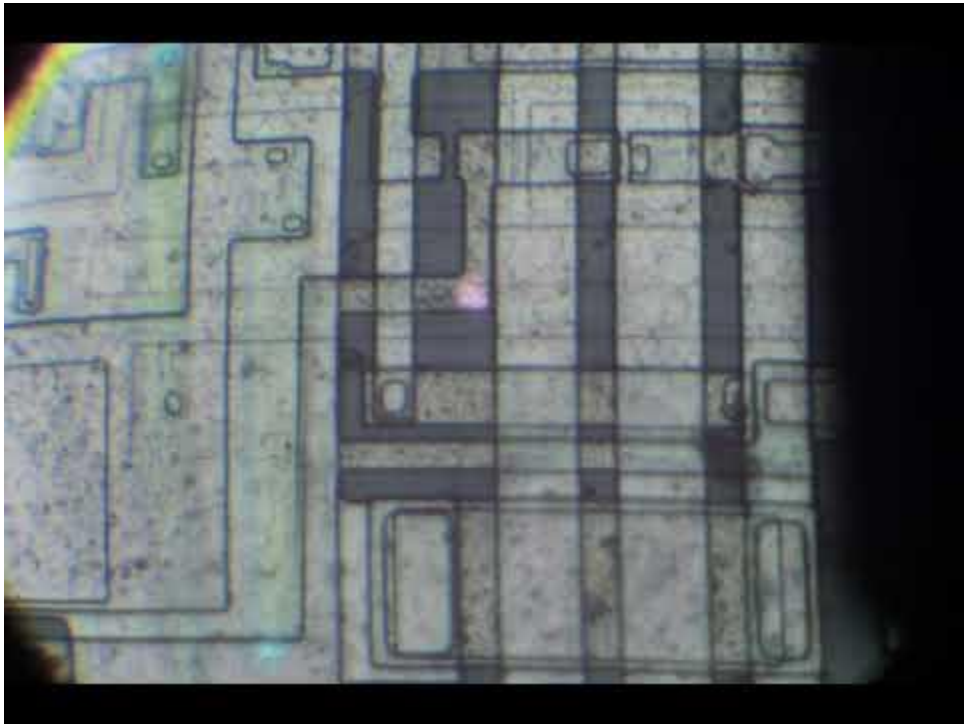
  1) Use laser probe station to cut holes in overglass to expose metal

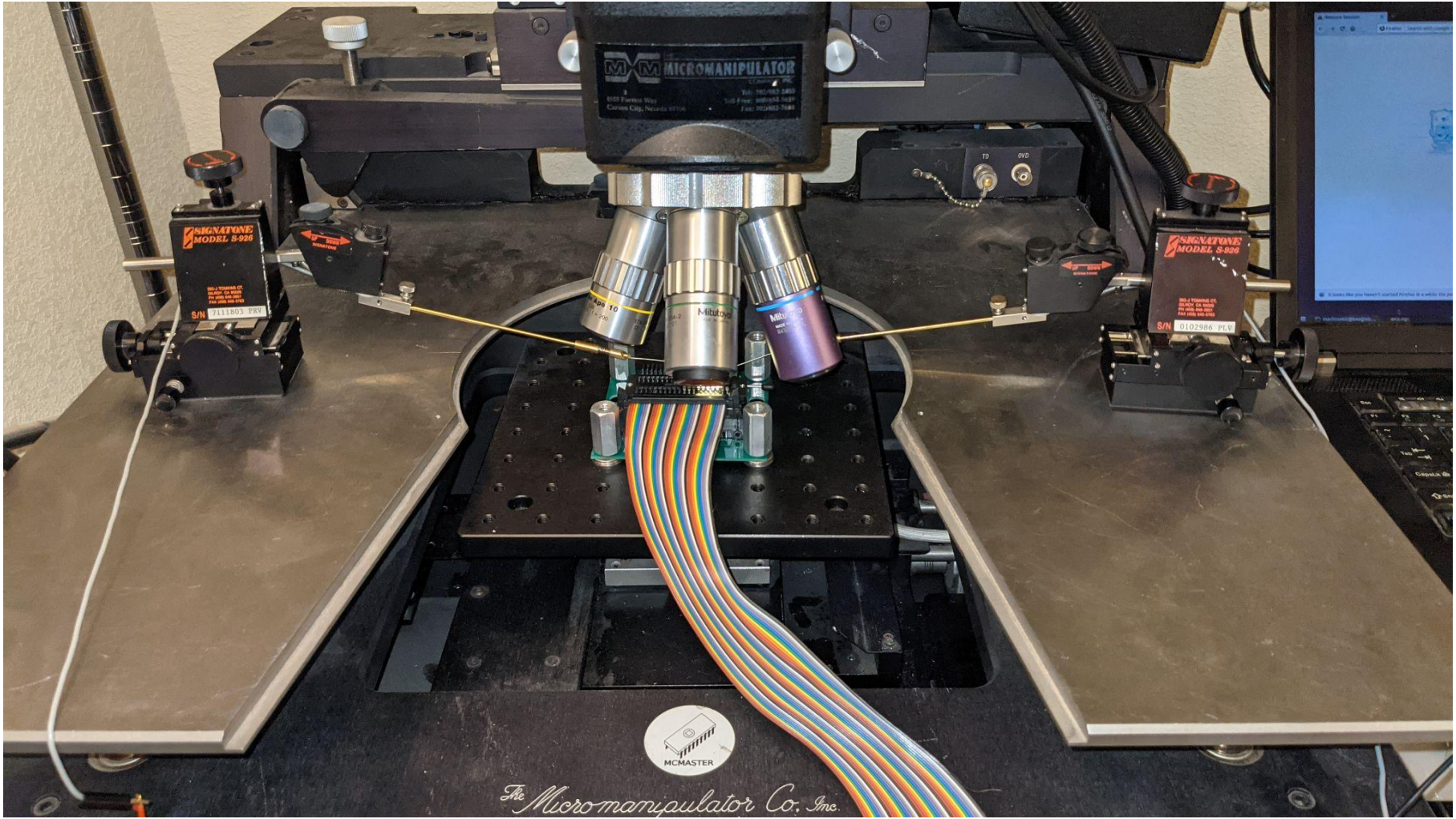
  2) Microprobe exposed circuit traces to force logic value

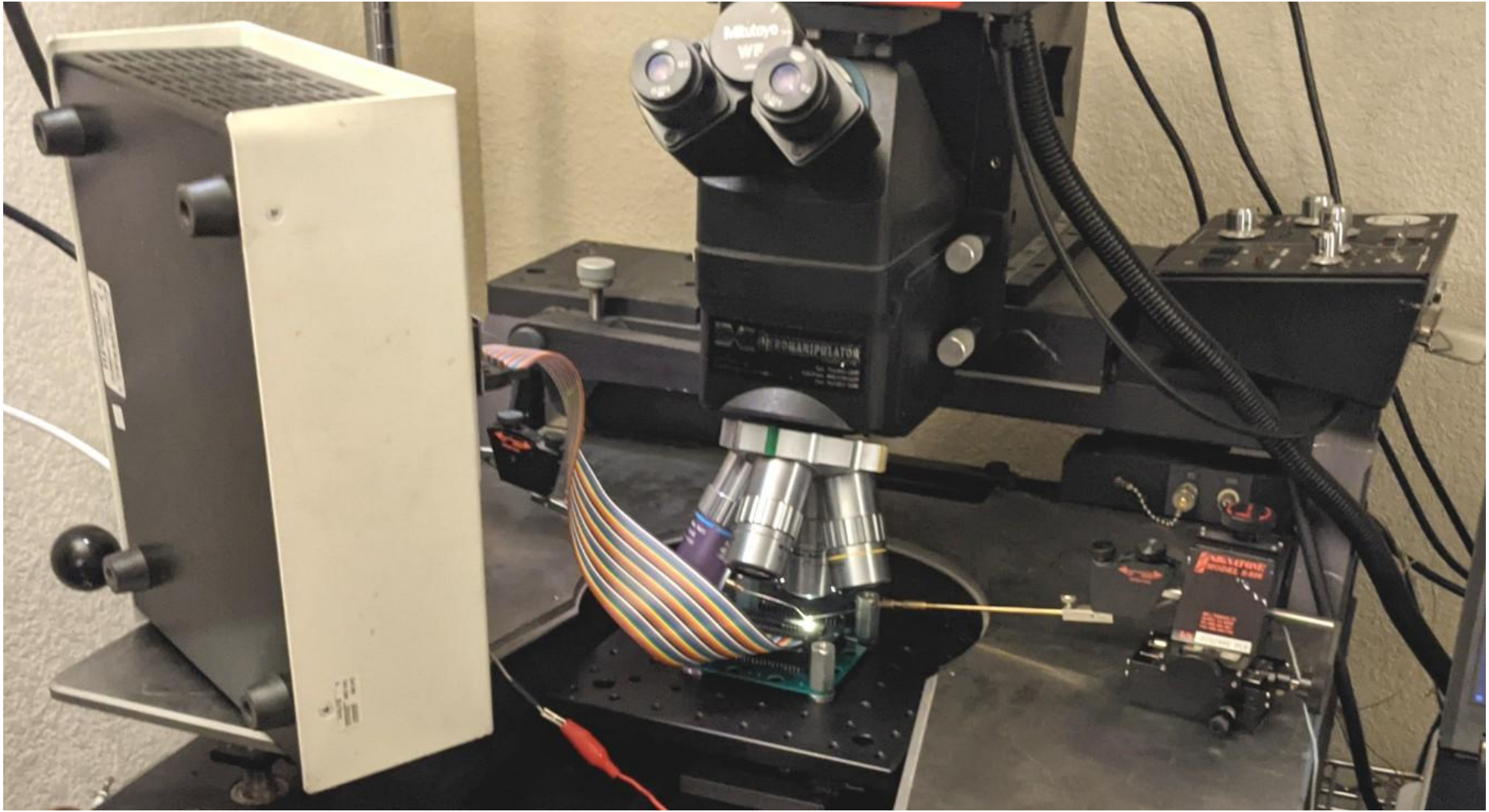
How to probe?

- Take scope traces before / after security to understand logic states
- Ideally constant value (VCC/GND) => don't need to actually bridge fuse
- Old/big logic => not using active probes

Time to put some photons in

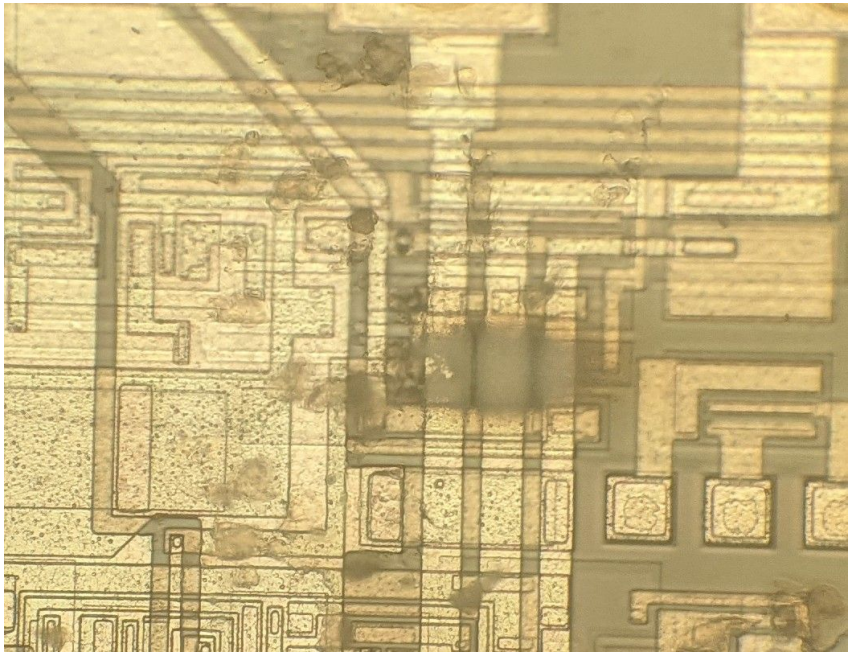



MICROMANIPULATOR
SIGNATONE
MODEL S-926
S/N 7111803 PRV
SIGNATONE
MODEL S-926
S/N 0102986 PLV
MCMASTER
The Micromanipulator Co., Inc.

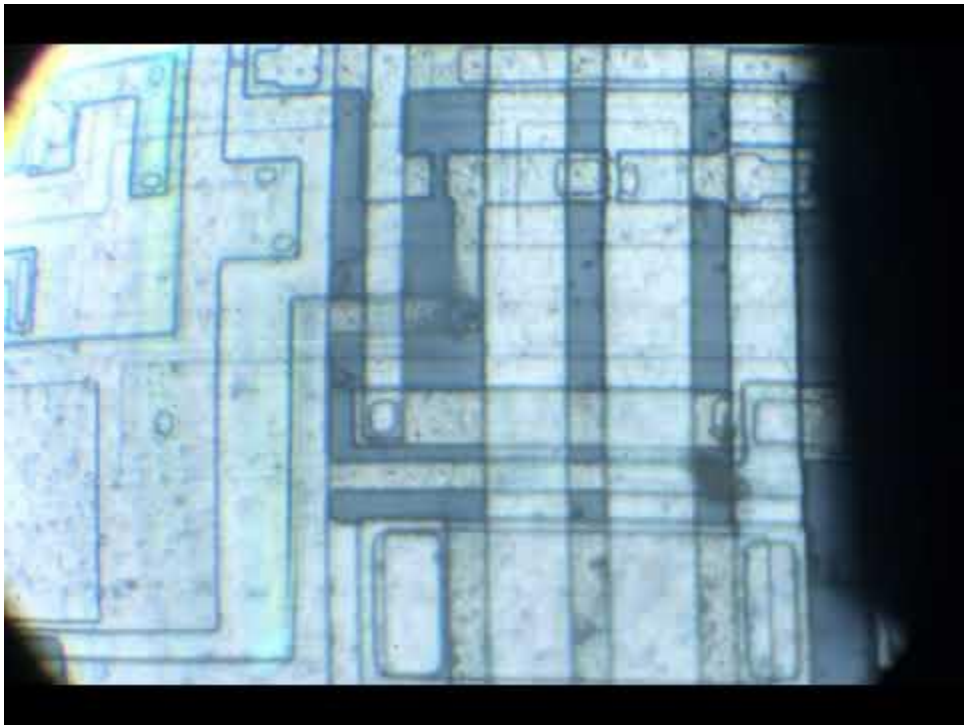

Mitutoyo
MANIPULATOR

Oopsie daisy...short circuit

- Tungsten probe somehow melted before the IC trace!

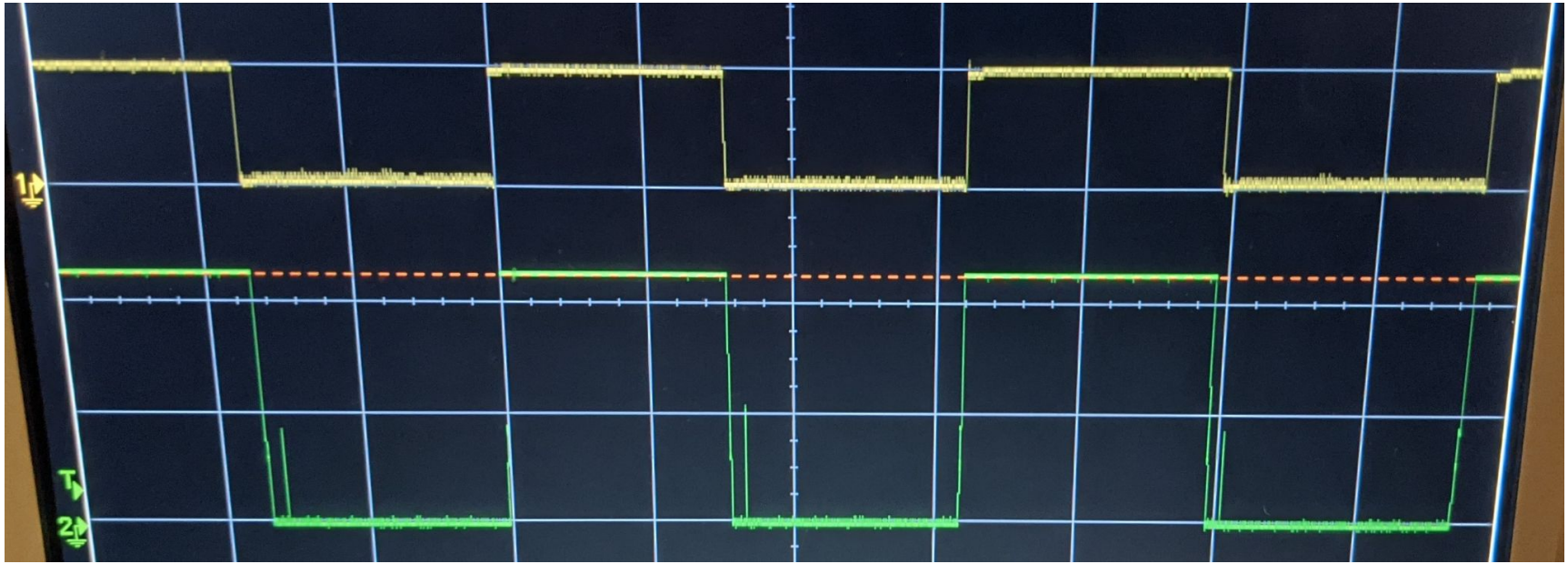


Try probing again...



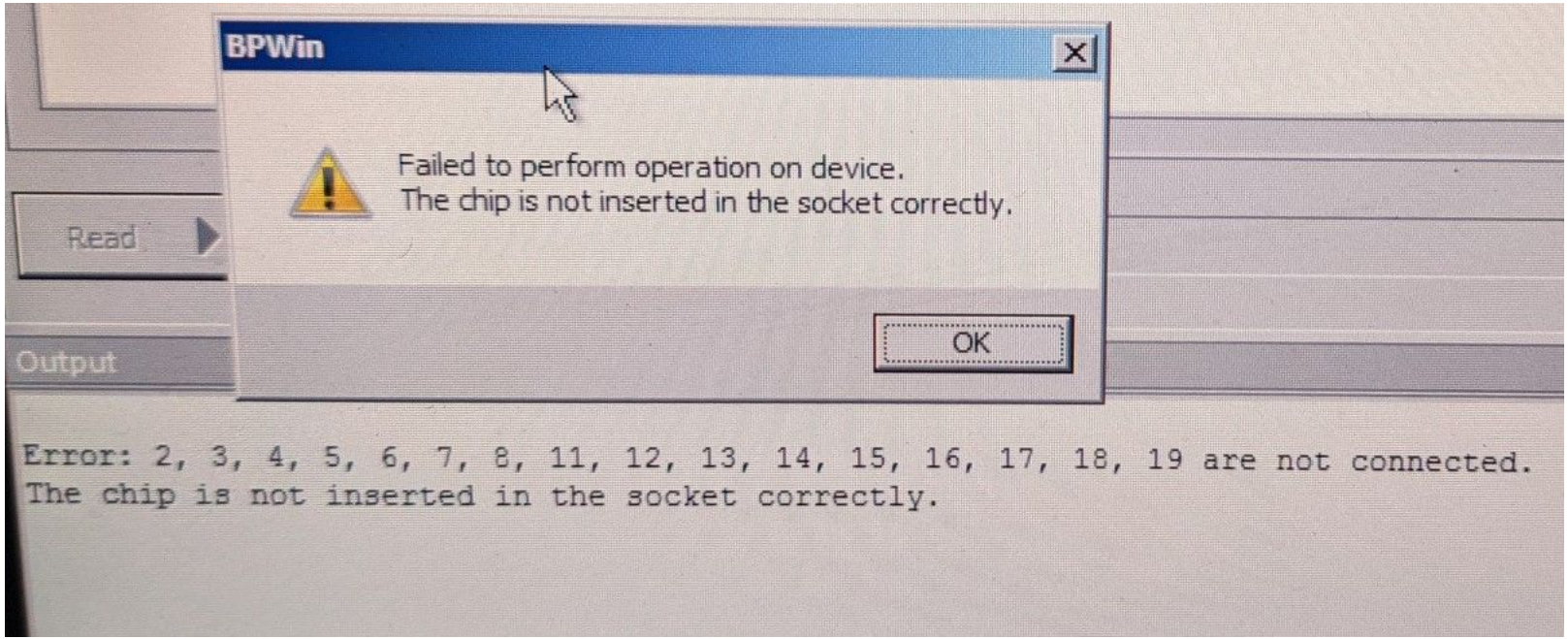
An unexpected result

- Not constant VL or VH => analog / specialized circuit?
- 5V (inner die) and ~12V (outer die) observed



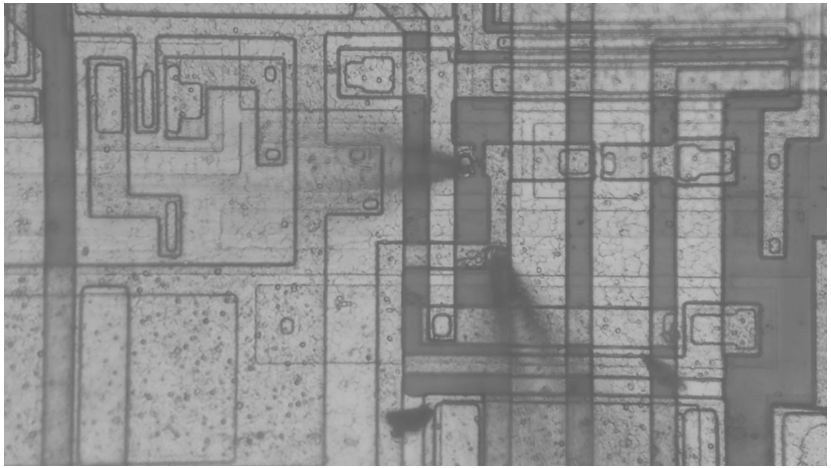
External power made it sad

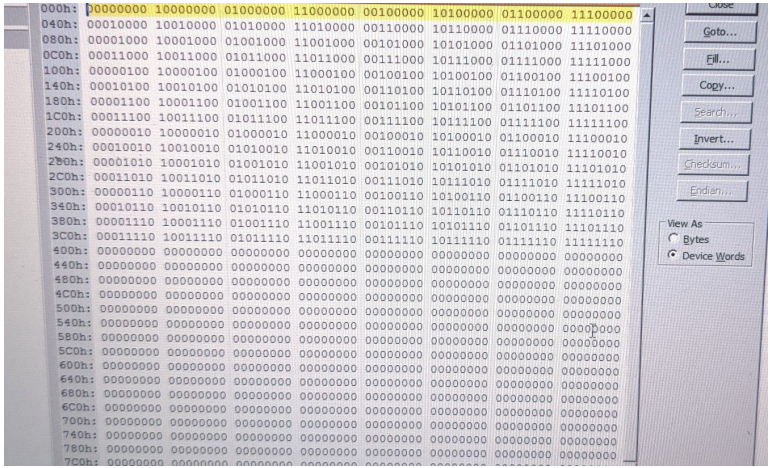
- Tried both sides



...so just bridge the contacts w/ two microprobes

- Success! But half of the data => do twice
- PoC here but also worked on Trackstar PAL16R8

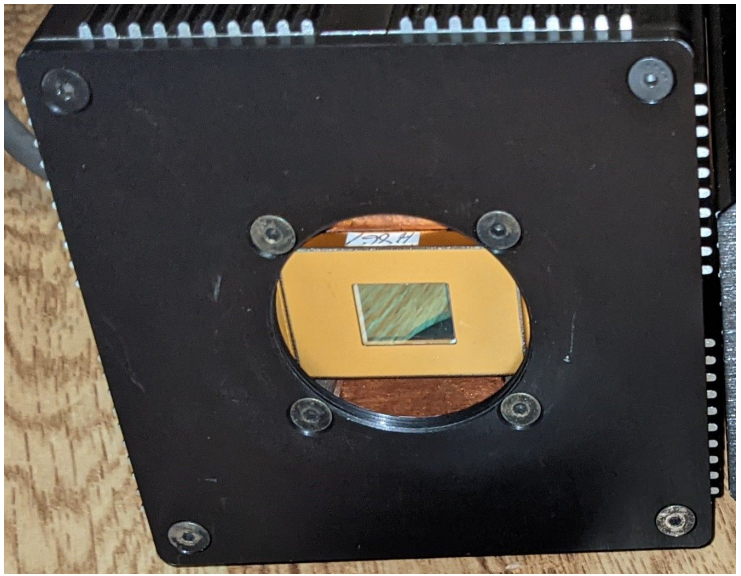




Epilogue (if time)

Need better camera for more modern applications?

- InGaAs cameras! Non-silicon IC
- More sensitive but short exposure time :(
- Neither useful in practice

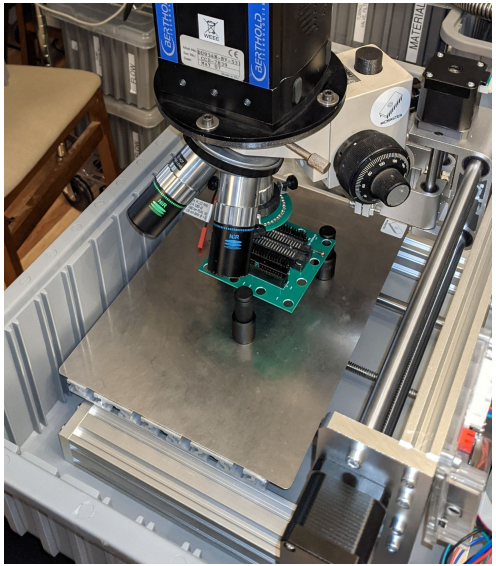


What else is out there?

- Last time: cooled camera had low noise, but CMOS had best sensitivity
- Could we get a cooled CMOS detector? Yes!
- $$$ but scrapped out of surplus medical device
- Special thanks to Bayley of oneTesla for loaning me this camera!

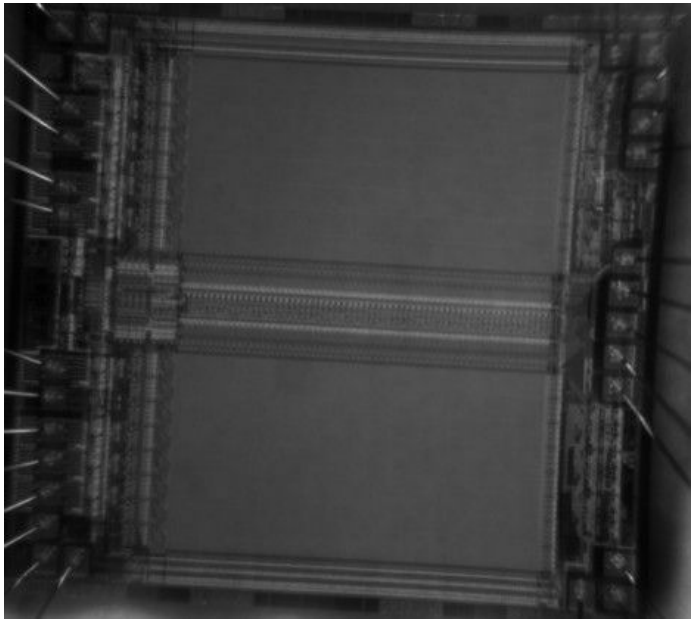
Improved enclosure

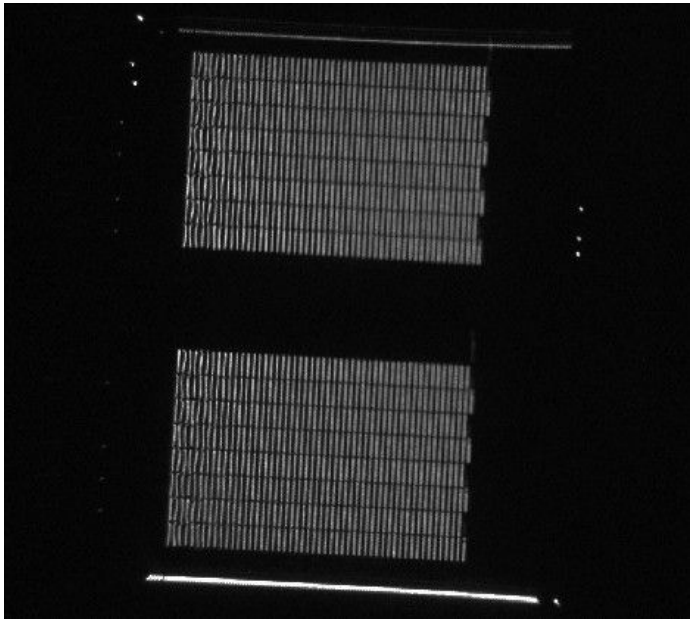
- Mount camera on low cost CNC to manipulate in the dark
- More properly sealed enclosure



EPROM / flash?

- EPROM plausible, flash no
- TODO: EPROM security fuse (ex: 8751, PIC)





Future work

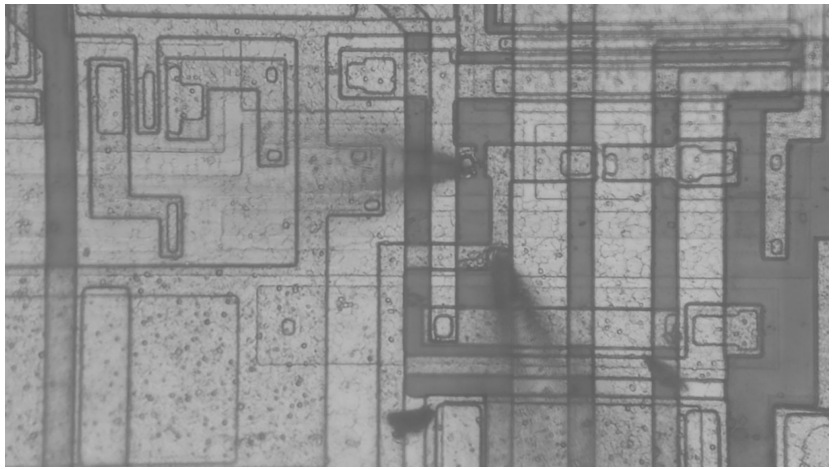
- Antifuse
- EPROM security fuses
- Bridge w/ silver epoxy?

Special thanks

- Ethan Wright: helping with lab work
- Bayley Wang: cooled Apogee camera
  - https://onetesla.com/
- Brandon Cobb: Trackstar support, partial funding
  - http://www.diskman.com/

Thanks for listening!

- @johndmcmaster
- BP Microsystems AHK: https://github.com/JohnDMcMaster/bpmicro
- pal16 scripts: https://github.com/JohnDMcMaster/icfuzz
- pal866: https://github.com/JohnDMcMaster/pal866



Backup

Summary

- Fuse burning readily visible under microscope
- EPROM was visible in some of my tests
- Flash was not visible in my tests

Demo?

- Questions first?
- Step 1: find fuses
- Step 2: microprobe security

Bit matrix to Verilog workflow

- python3 txt2jed.py capture.txt capture.jed
- jedutil -convert capture.jed capture_mame.bin
- python3 pal16r8_to_verilog.py capture_mame.bin >capture.v

Laser settings

- Green, High-650
- Marker around 2.25 / 2.25

An unexpected result

- Not VL or VH => analog / specialized circuit
- Options:

  1) Emulate voltage externally

  2) Bridge fuse(s)

	Blank	Secure (outer)	Secure (inner)
Left	1.62V	1.70V	Oops
Right	1.55V	1.47V	0.01V