Micro/Nanofabricated Implantable Neural Interfaces

BEAMeeting

2025-05-27

Travis Massey (massey21@llnl.gov)

Staff Scientist | Center for Bioengineering, Implantable Microsystems Group

Operations Manager | Center for Micro and Nano Technology

Jenny Zhou (zhou9@llnl.gov)

Senior Process Engineer| Center for Micro and Nano Technology

ENGINEERING
LAWRENCE LIVERMORE NATIONAL LABORATORY

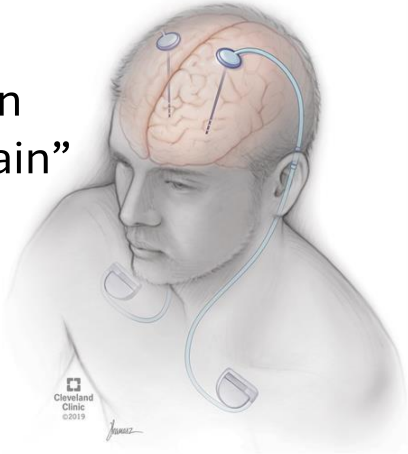
Prepared by LLNL under Contract DE-AC52-07NA27344.

Neural Interfaces

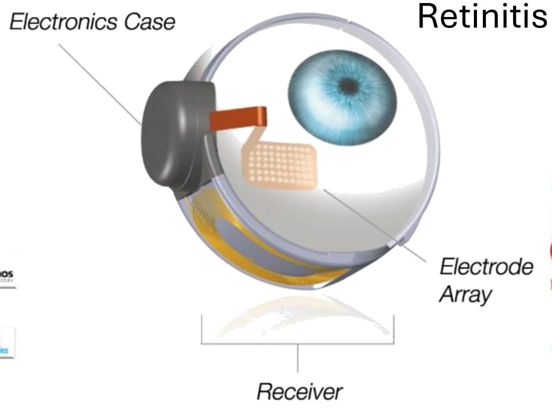
Engineered devices that read or alter neurons’ signaling

Deep-Brain Stimulation
“Pacemaker for the brain”

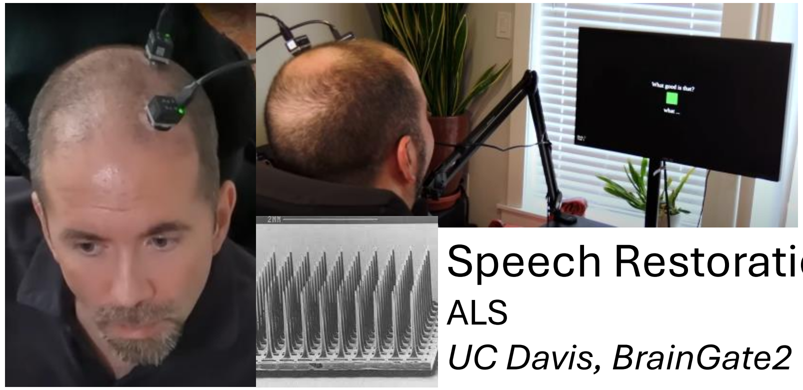
Parkinsons, Tremors,
Dystonia, Epilepsy, OCD



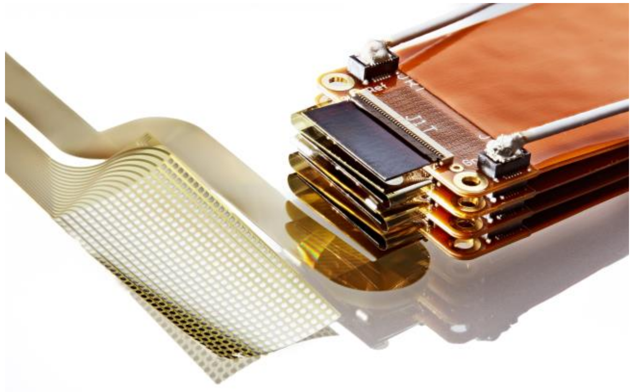
Artificial Retina
Retinitis Pigmentosa



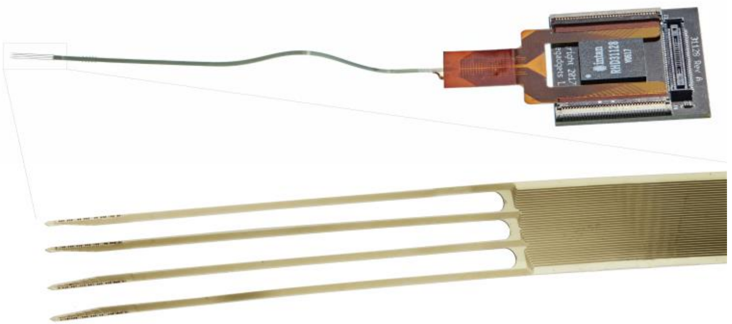
Invention of the Year
POPULAR SCIENCE
TIME
2013



Speech Restoration
ALS
UC Davis, BrainGate2



Speech/auditory cortical surface recording
LLNL/UCSF

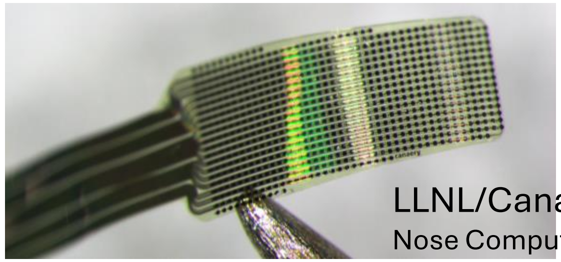


128 intracortical probe for rodents
LLNL/UCSF

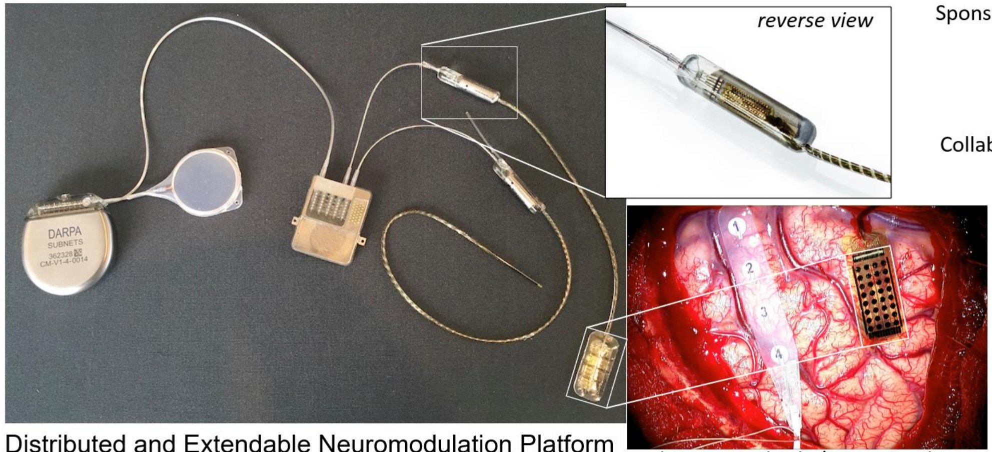
Anatomy of a Neural Interface

- Electrodes
- Shank/head
- Cable/traces
- Contacts
- Substrate
- Connector(s)
- Electronics
- Firmware
- Power

Micro/Nanofabricated Probe



LLNL/Canaery
Nose Computer Interface

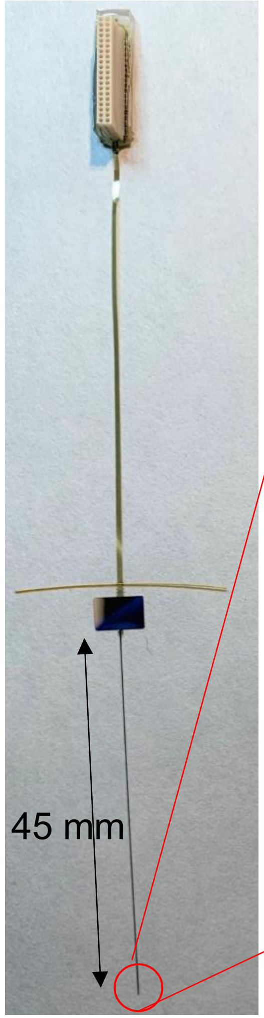


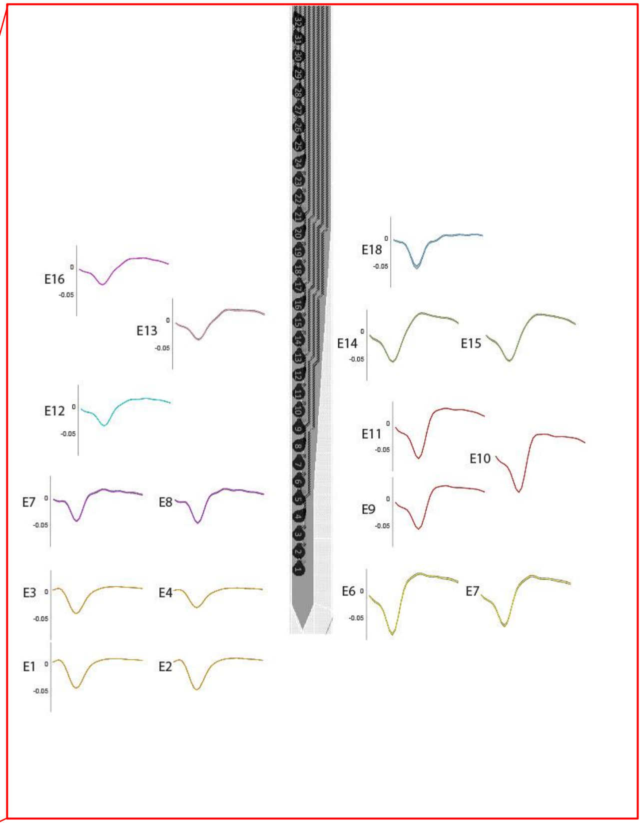
Distributed and Extendable Neuromodulation Platform
(LLNL, Haque, 2019)

Intraoperative in-human testing
(Chang, Leonard, Sellers - UCSF)

Sponsored by:
DARPA

Collaborators:
UCSF
UCLA
Boston Scientific

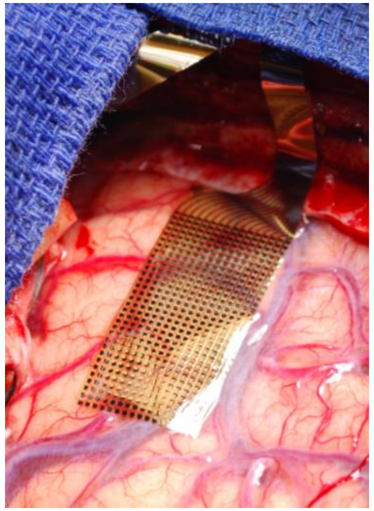




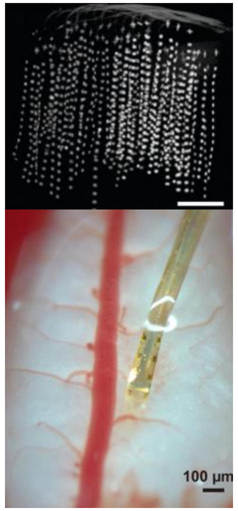
50-100 µV single units in NHP
Rueckemann, UW

Flexible Polyimide Probes at LLNL

Flexible
Conform to brain
Minimize tissue damage
Decouple from motion
Flex around blood vessels



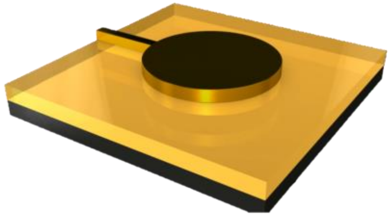
LLNL/UCSF (Massey/Chang)

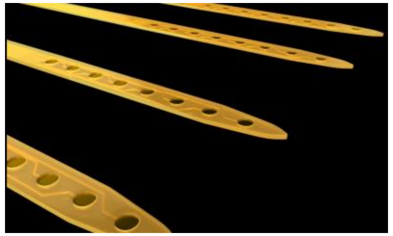


Xie (Rice)

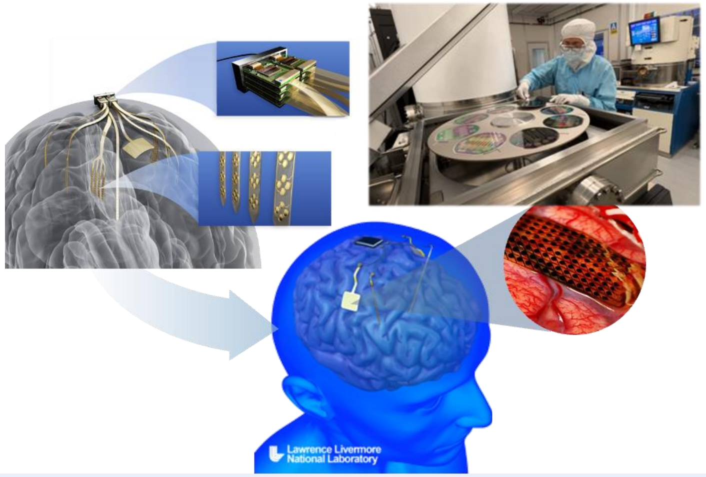
Polyimide Probes
Great engineering polymer
- Chemically stable
- Biocompatible
- Mechanically robust
- Low moisture uptake

Amenable to nanofab
- Spin-on, 0.5-25+ µm
- Holds fine features
- High-vac compatible
- Non-contaminating



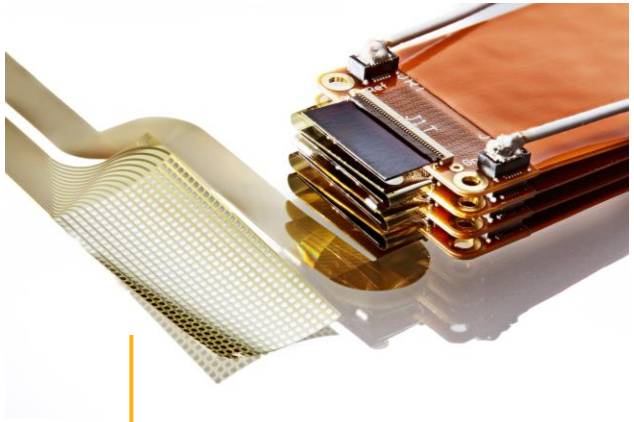


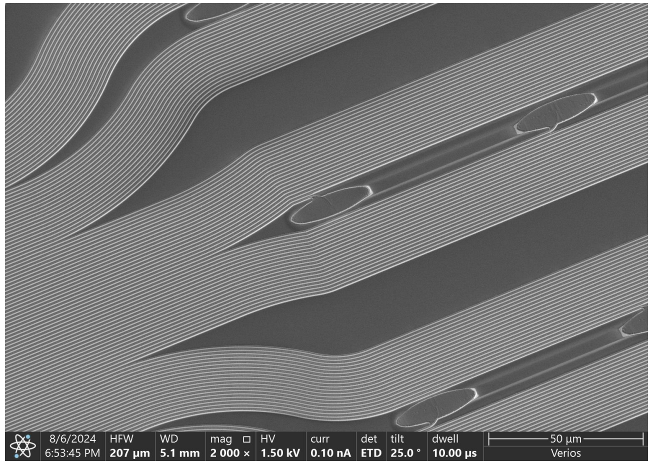
[Why] at LLNL
Center for Micro/Nano Technology
- 22k ft^2 → 33k ft^2
- High-throughput
- ISO13485 Biomedical Foundry



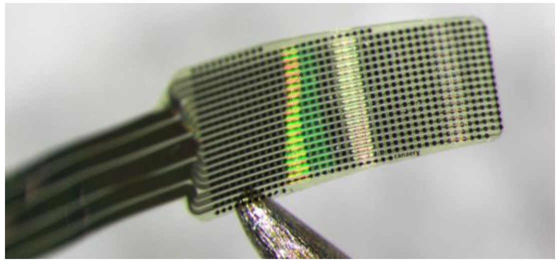
Micro → Nano

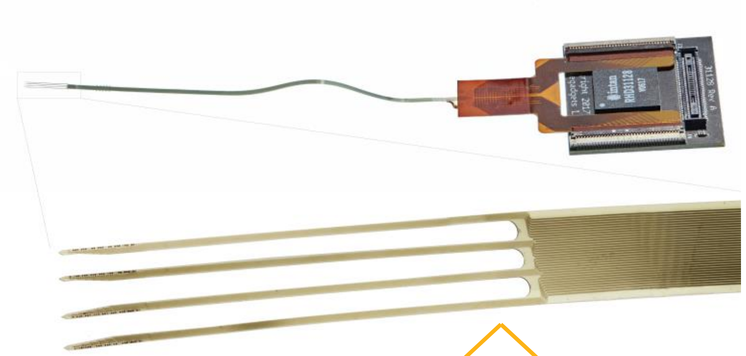
512ch
500 µm electrodes
5 µm traces





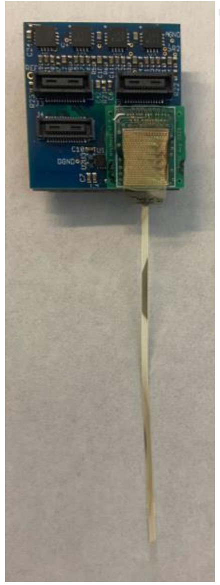
767ch
40 µm electrodes
0.42 µm traces

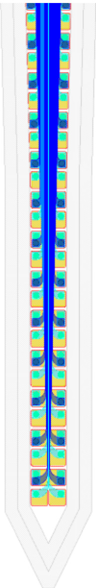


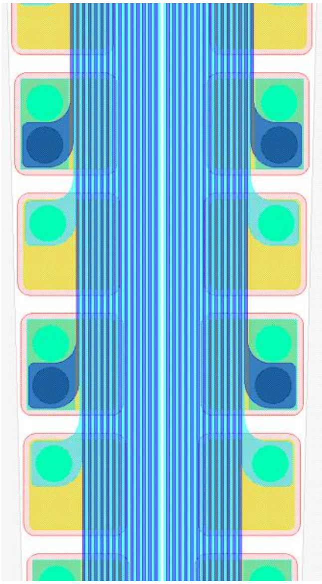


128 ch
20 µm electrodes
5 µm traces

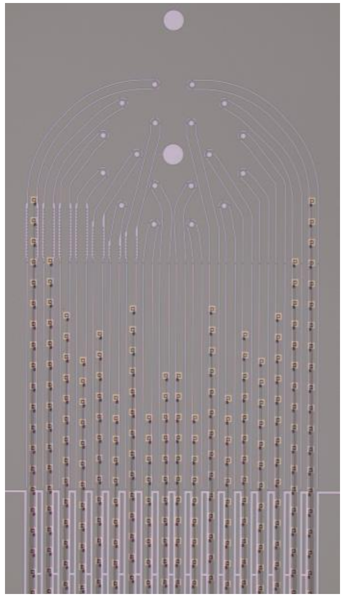
512 ch
15 µm electrodes
0.42 µm traces

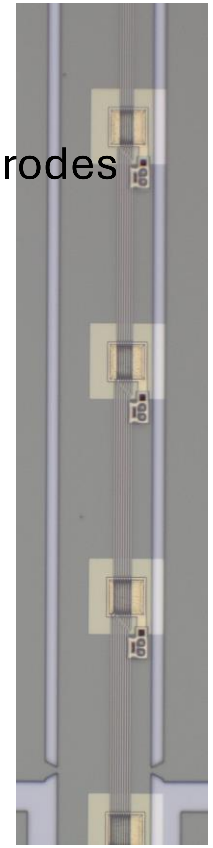






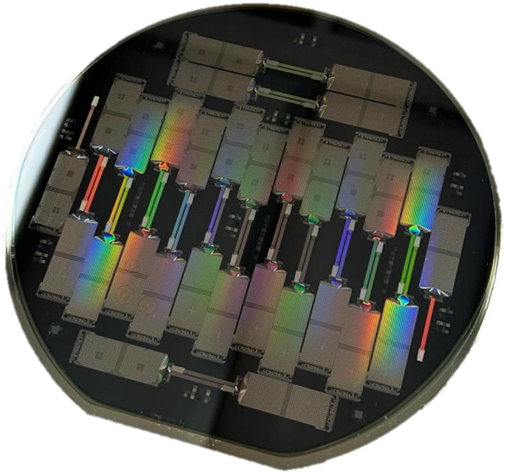
1024 ch
30 µm electrodes
1 µm traces





Less common EBL challenges

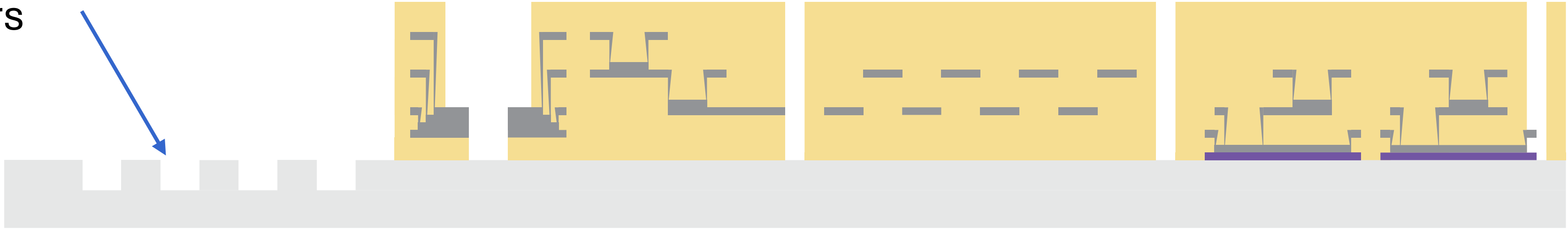
150 mm wafer scale ...



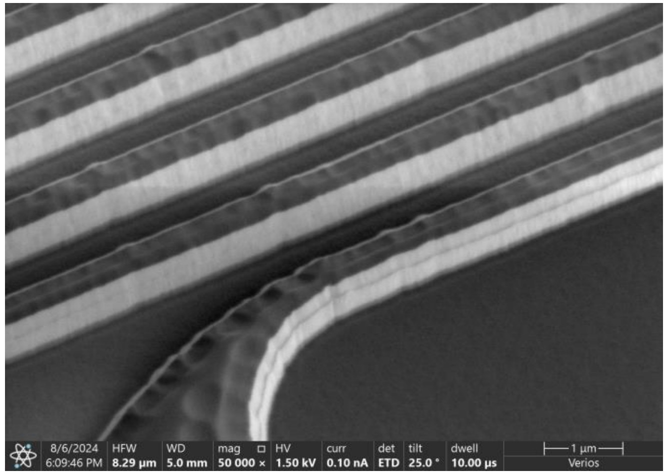
... 8 wafers/batch

Must design for throughput!

Negative alignment markers



~10-20 layers per fab, only global alignment → design ~500 nm overlay tolerance

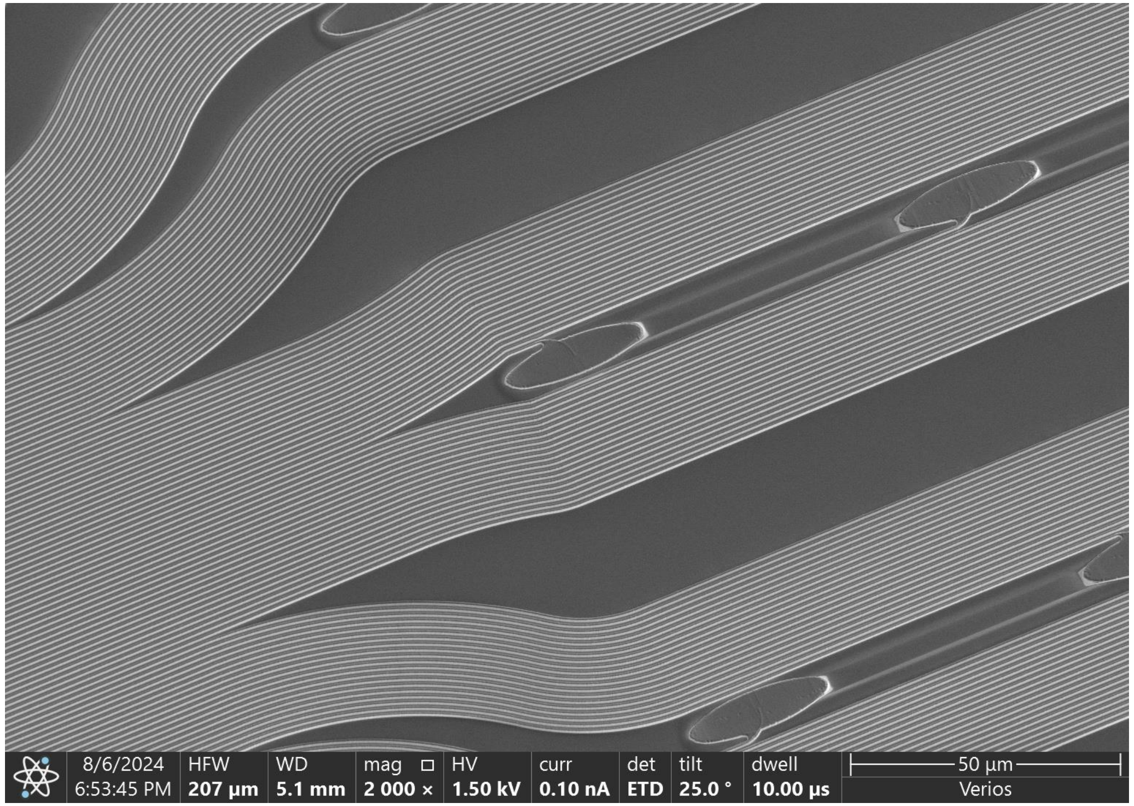


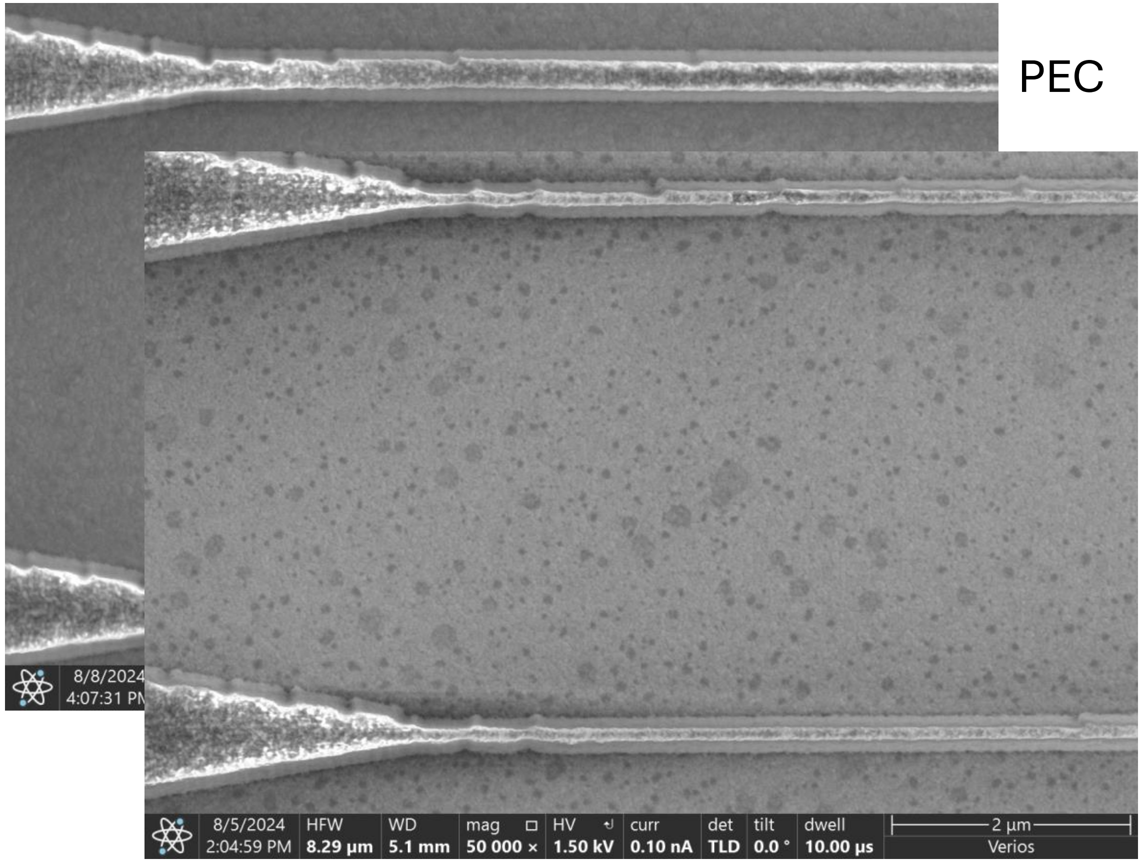
300 nm thick Ti-Pt-Au-Pt traces...

1. choose resist carefully
2. maximize resist thickness
3. maximize selectivity

Where is PEC useful?

Large variations in feature density





PEC

No PEC

Python is great for PEC

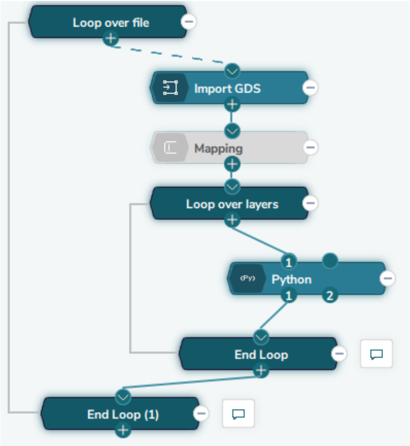
```python
# Extract layer
extract_out = BEAMER.extract_layer(extract_in, { 'ExtentMode' : 'User', 'CellName' : '', 'LayerSet' : layerNum, 'RegionLayer' : '', 'RegionLayerBehavior' : 'Clip', 'ResultMode' : 'Region/Instances', 'DoseClassificationShr
if BEAMER.get_area(extract_out)>0:

    # Try importing LEDB
    print('Trying LEDB import for ' + layoutName + ' L' + layerNum + '_' + layerDescr)
    try:
        filename_ldb = myDirectory + '/LEDB post-PEC intermediates/' + layoutName + '_L' + layerNum + '-' + layerDescr + '_Post-PEC.ldb'
        imported_ldb = BEAMER.import_ldb( { 'LayerSet' : '*', 'FileName' : filename_ldb } )
    except:
        print('LEDB import failed for ' + filename_ldb + '.  File may not exist.')
        no_LEDB_preexists = True
    else:
        print('LEDB import succeeded for ' + filename_ldb + '.')
        no_LEDB_preexists = False

    # If there is no preexisting LEDB, then heal, fracture, and PEC the layout, then save an intermediate LEDB
    if no_LEDB_preexists:
        heal_out = BEAMER.heal(extract_out, { 'SoftFrame' : 0.300000, 'HierarchicalProcessing' : True, 'SelectedLayerSet' : '*', 'LayerAssignment' : 'PerLayer', 'ProcessingMode' : 'Healing' } )
        fracture_out = BEAMER.fracture(heal_out, { 'FractureLayer' : '*', 'KeepResolution' : True, 'Resolution' : 0.001, 'BeamStepSize' : 1, 'CurveApproxTolerance' : 0.100000, 'CurveTolerance' : 1.000000, 'FractureAxis' :
        try:
            print('Trying PEC for ' + layoutName + '_' + layerNum + '-' + layerDescr + '.')
            PSF_ref_static_front = 'Substrate_Si_Thickness_675000_Energy_100_Layers_'
            PSF_ref_static_tail = '_Alpha_0_Beta_0_Eta_0_Gamma1_0_Nue1_0_Gamma2_0_Nue2_0_Simulator_mcTrace 1.1.0'
            if PSF_tag == 'Polyimide_2um':
                PSF_archive_ref = PSF_ref_static_front + 'PI 2000nm, Al 100nm_Resist_PMMA 600 nm_Z-Position_0.275_Electrons_20000000' + PSF_ref_static_tail
            elif PSF_tag == 'PtAuPt_on_2umPI':
                PSF_archive_ref = PSF_ref_static_front + 'PI 2000nm, Pt 25nm, Au 250nm, Pt 25nm_Resist_AR-N_7520new 600 nm_Z-Position_0.275_Electrons_20000000' + PSF_ref_static_tail
            elif PSF_tag == 'Polyimide_4um':
                PSF_archive_ref = PSF_ref_static_front + 'PI 4000nm, Al 100nm_Resist_PMMA 600 nm_Z-Position_0.275_Electrons_20000000' + PSF_ref_static_tail
            elif PSF_tag == 'TiPtAuPt_on_4umPI':
                PSF_archive_ref = PSF_ref_static_front + 'PI 4000nm, Ti 15nm, Pt 25nm, Au 250nm, Pt 25nm_Resist_AR-N_7520new 600 nm_Z-Position_0.275_Electrons_20000000' + PSF_ref_static_tail
            elif PSF_tag == 'Polyimide_6um':
                PSF_archive_ref = PSF_ref_static_front + 'PI 6000nm, Al 100nm_Resist_PMMA 600 nm_Z-Position_0.275_Electrons_20000000' + PSF_ref_static_tail
            elif PSF_tag == 'TiPtAuPt_on_6umPI':
                PSF_archive_ref = PSF_ref_static_front + 'PI 6000nm, Ti 15nm, Pt 25nm, Au 250nm, Pt 25nm_Resist_AR-N_7520new 600 nm_Z-Position_0.275_Electrons_20000000' + PSF_ref_static_tail
            elif PSF_tag == 'Polyimide_8um':
                PSF_archive_ref = PSF_ref_static_front + 'PI 8000nm, Al 100nm_Resist_PMMA 600 nm_Z-Position_0.275_Electrons_20000000' + PSF_ref_static_tail
            elif PSF_tag == 'TiPtAuPt_on_8umPI':
                PSF_archive_ref = PSF_ref_static_front + 'PI 8000nm, Ti 15nm, Pt 25nm, Au 250nm, Pt 25nm_Resist_AR-N_7520new 600 nm_Z-Position_0.275_Electrons_20000000' + PSF_ref_static_tail
            elif PSF_tag == 'Polyimide_10um':
                PSF_archive_ref = PSF_ref_static_front + 'PI 10000nm, Al 100nm_Resist_PMMA 600 nm_Z-Position_0.275_Electrons_20000000' + PSF_ref_static_tail
            else: PSF_archive_ref = 'null'
            PECresult = BEAMER.pec(fracture_out, { 'UserdefinedDoseClassFile' : '**filename**',
                            'MinFractureSizeMode' : 'Automatic',
                            'PSFArchiveIdentifierString' : PSF_archive_ref,
                            'PSFType' : 'Archive',
                            'PSFFileName' : '',
                            'BeamSize' : blur,
                            'DoseClassMode' : 'Accuracy',
                            'MaxNumOfDoseClasses' : 256,
                            'Accuracy' : 1.000000,
                            'MinDoseFactor' : 0.100000,
                            'MaxDoseFactor' : 10.000000,
                            'UserDefinedSeparationValue' : False,
                            'SeparationValue' : 0.100000,
                            'FractureGrid' : float(beam_step_size_nm)/1000,
                            'MinFractureSize' : 0.100000,
                            'MinFractureSizeShortRange' : 0.100000,
                            'LayerListForCorrection' : '*',
                            'LayerListForFullCorrection' : '*',
```

Export LEDB after PEC, before fracture. Avoid re-running PEC.

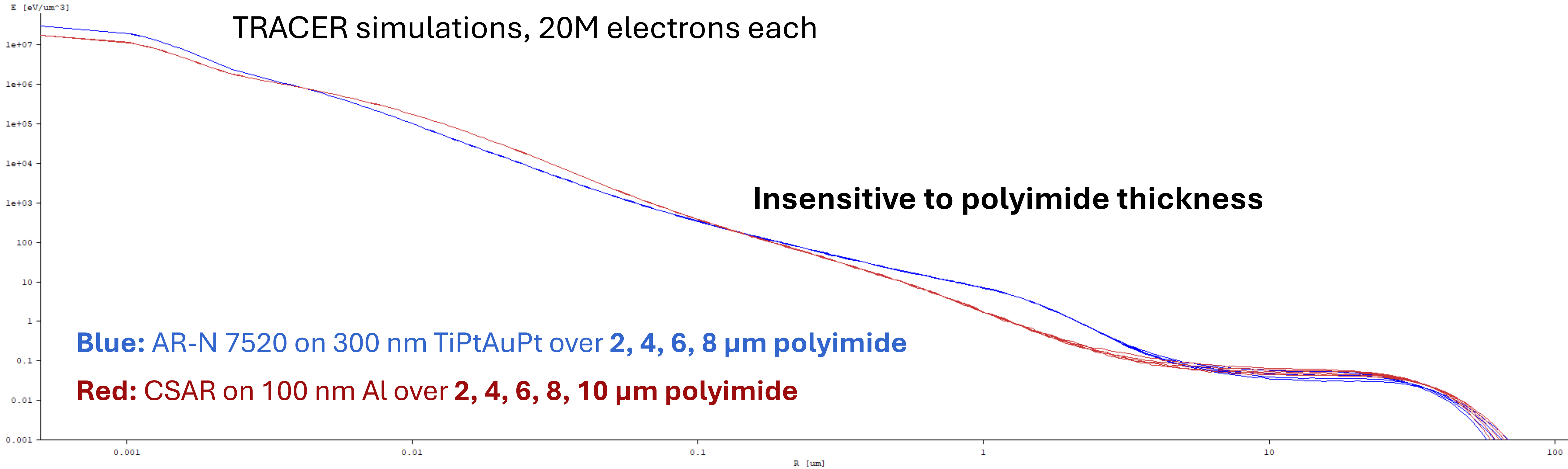
Only way to programmatically select PSF for PEC per layer. (Had I needed to!)



What about the insulating substrate?

It doesn't matter at 100 keV!

TRACER simulations, 20M electrons each



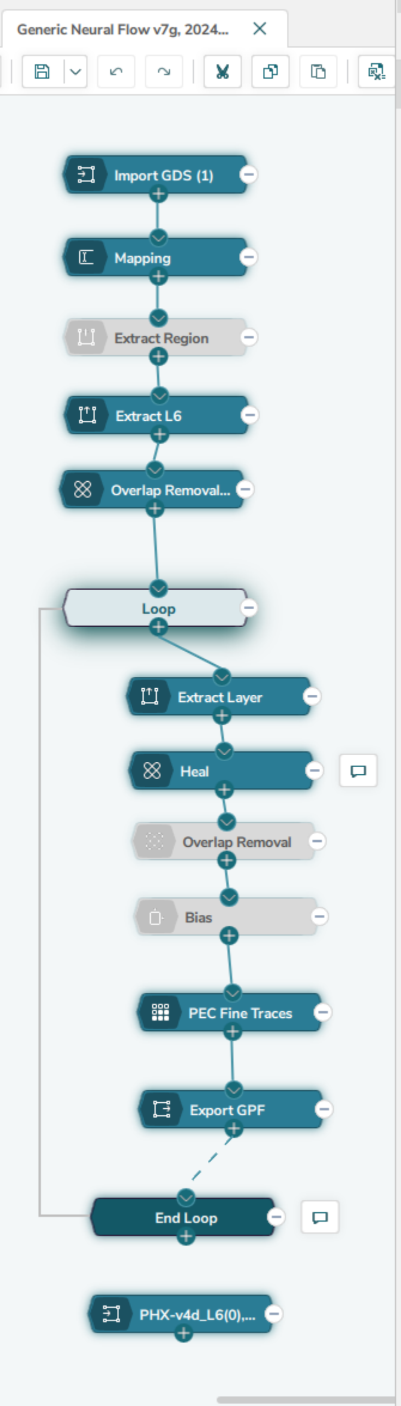
Insensitive to polyimide thickness

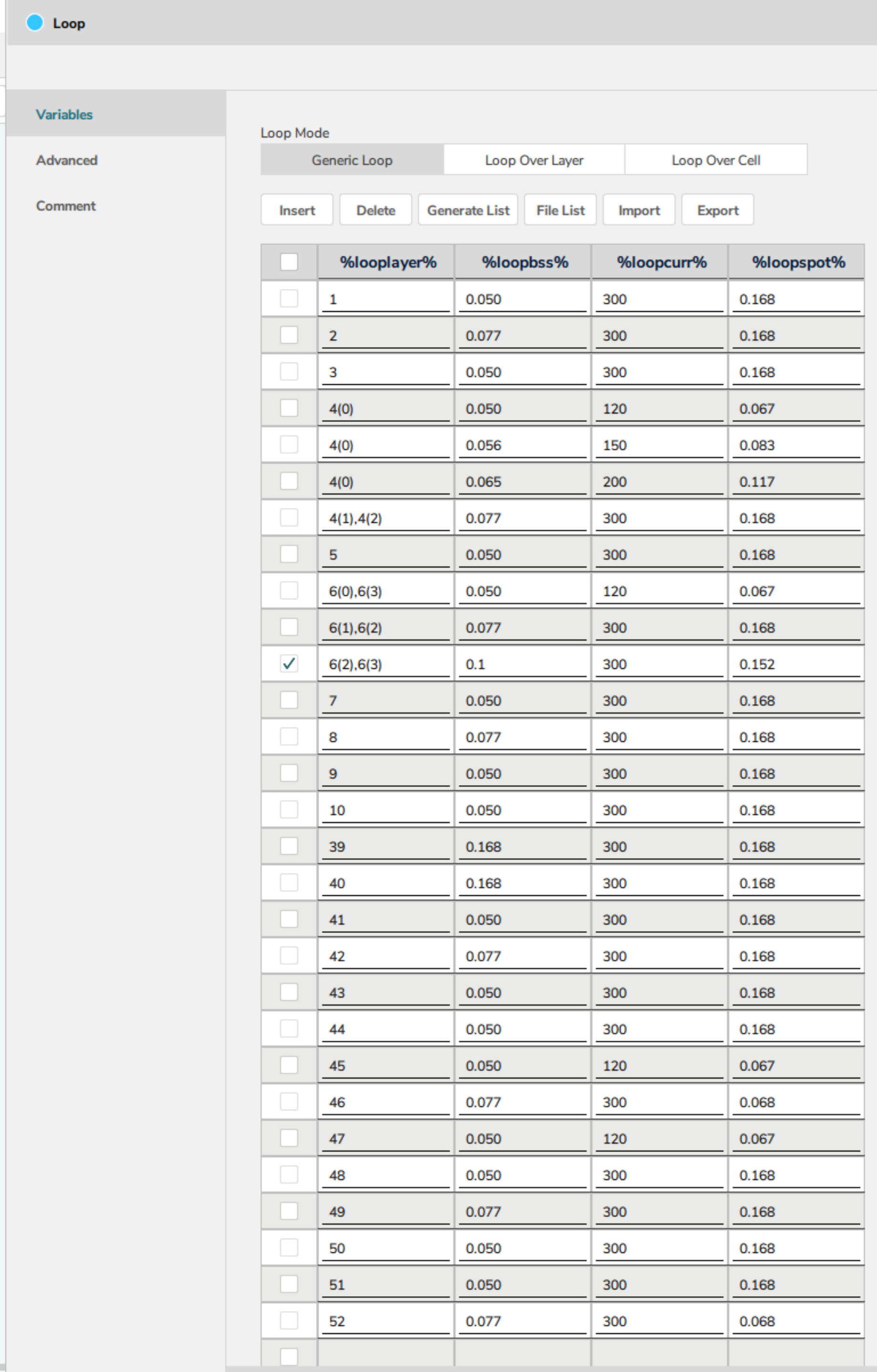
Blue: AR-N 7520 on 300 nm TiPtAuPt over **2, 4, 6, 8 µm polyimide**

Red: CSAR on 100 nm Al over **2, 4, 6, 8, 10 µm polyimide**

Manage many layers with loop variables

- Easy to run 1 layer or N layers
  - Run and ignore for ~24h
- No need to track which parameters to change and which modules they're in
- No need to maintain external list of values
- Parameterize output file naming with layer and beam info → clear and consistent





	%looplayer%	%loopbss%	%loopcurr%	%loopspot%
	1	0.050	300	0.168
	2	0.077	300	0.168
	3	0.050	300	0.168
	4(0)	0.050	120	0.067
	4(0)	0.056	150	0.083
	4(0)	0.065	200	0.117
	4(1),4(2)	0.077	300	0.168
	5	0.050	300	0.168
	6(0),6(3)	0.050	120	0.067
	6(1),6(2)	0.077	300	0.168
✓	6(2),6(3)	0.1	300	0.152
	7	0.050	300	0.168
	8	0.077	300	0.168
	9	0.050	300	0.168
	10	0.050	300	0.168
	39	0.168	300	0.168
	40	0.168	300	0.168
	41	0.050	300	0.168
	42	0.077	300	0.168
	43	0.050	300	0.168
	44	0.050	300	0.168
	45	0.050	120	0.067
	46	0.077	300	0.068
	47	0.050	120	0.067
	48	0.050	300	0.168
	49	0.077	300	0.168
	50	0.050	300	0.168
	51	0.050	300	0.168
	52	0.077	300	0.068

Design for throughput

Minimize required dose

AR-N 7520.17 new	$\boxed{105\ uC/cm^2}$
ma-N 2405	$260\ uC/cm^2$
CSAR 6200.18	$\boxed{275\ uC/cm^2}$
PMMA 950K	$680\text{-}750\ uC/cm^2$

$$f\ [MHz] = \frac{0.1 * I\ [nA]}{D\left[\frac{\mu C}{cm^2}\right] * s^2[\mu m^2]}, \qquad f < 125\ MHz$$

Maximize beam current

EBPG 5200 limit:	300 nA	167nm spot
>200 nm features:	120 nA	47-68 nm spot

Minimize beam step size

Smoother features

Beam defocus is overlooked knob

Split GDS layers into datatypes

e.g., layer 6(0), 6(1), 6(2)

Assign beam based by feature size

Datatypes within layer PEC together

Minimize aperture realignments

Minimize current changes

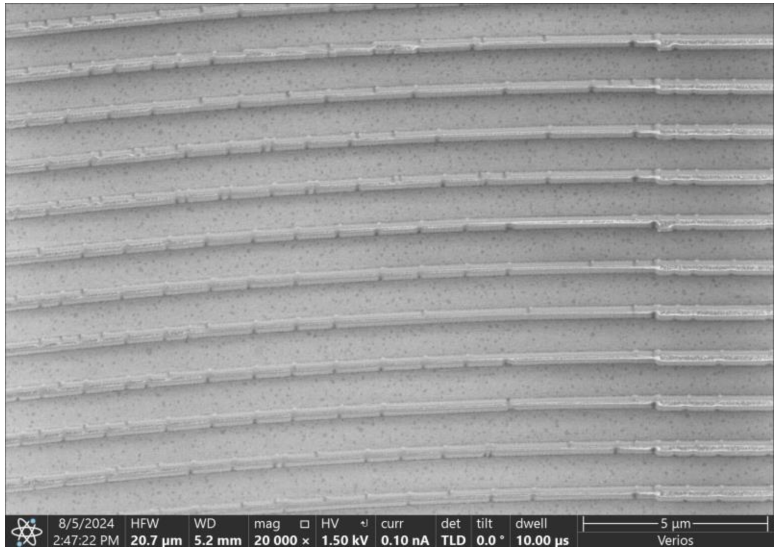
Minimize aperture changes

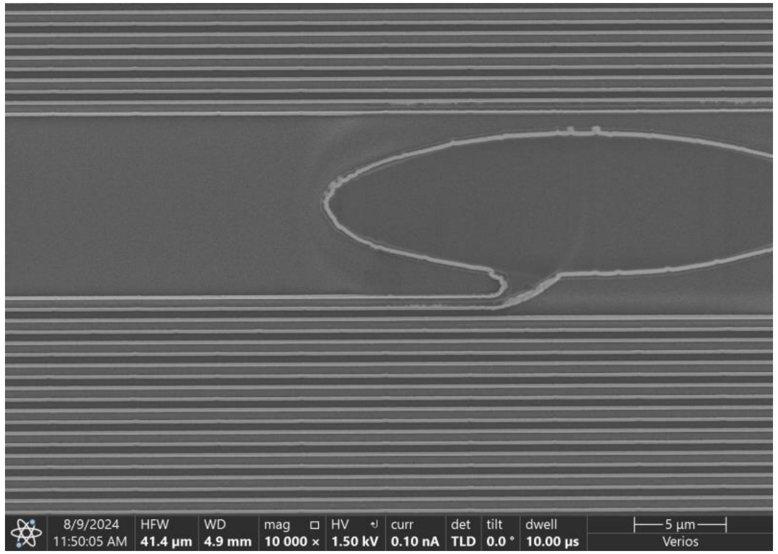
→ Common source of job crashes
(EBPG/BEAMS issue)

EBPG Fracturing and Tool Decisions

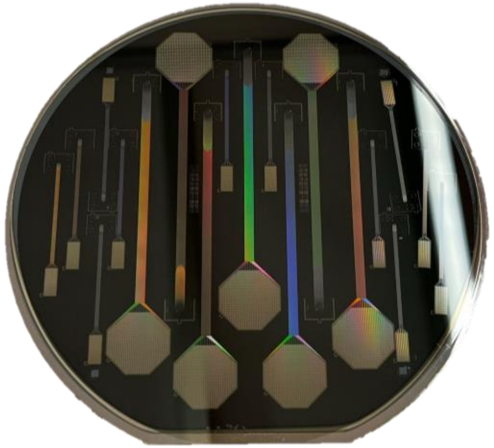
- 10 µm main field overlap
- Meander along dominant trace direction
- Curved fracturing → 7.3.0 fixed a 7.2.0 bug with incorrect fracturing!
- Jump & Settle

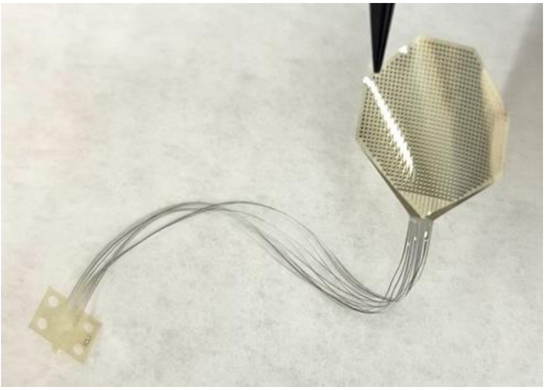
- Increase settling time by 10x to reduce notching at polygon boundaries (negligible write time impact)



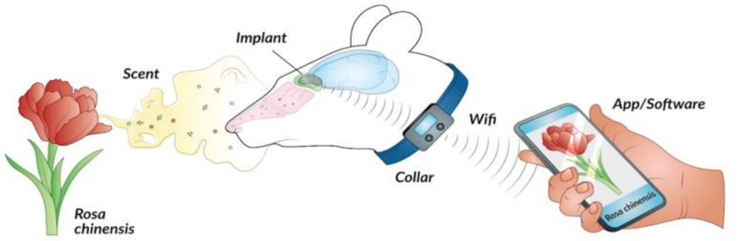


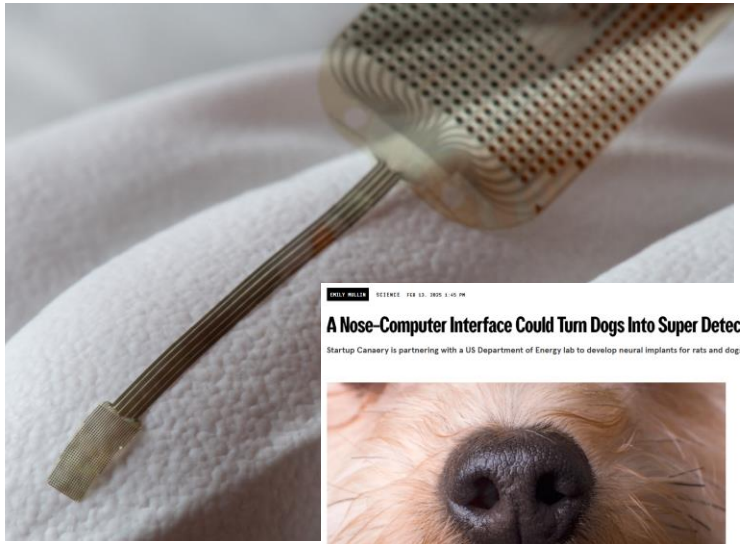
EBL implantables to date | Other EBL in progress





Visual Prosthesis
(industry collaborator)



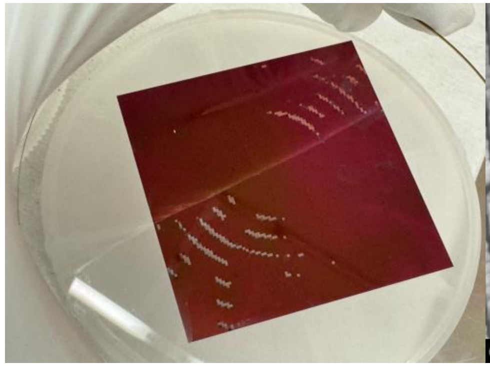


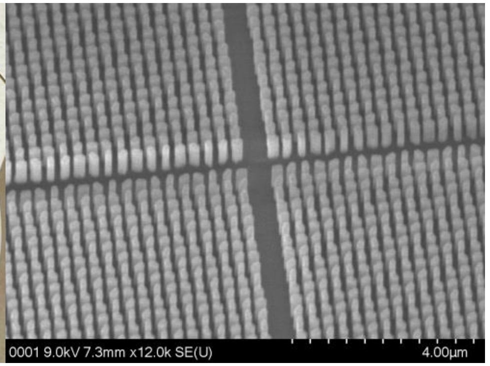
A Nose-Computer Interface Could Turn Dogs Into Super Detectors

Startup Canaery is partnering with a US Department of Energy lab to develop neural implants for rats and dogs that are capable of decoding what they smell.

WIRED

Olfactory Recording Array (collab. Canaery)





- Wafer-scale metalens array
- Quantum junctions, traps
- Photonic integrated circuits
- Gratings for fusion laser diagnostics

Funding & Support

Funding

NIH U01/UF1 NS107667

Canaery CRADA TC02420

NSF Convergence Accelerator, 24C0015

Facilities/Support

Thanks to Marvin, Kaustubh, and Roger for the endless support!