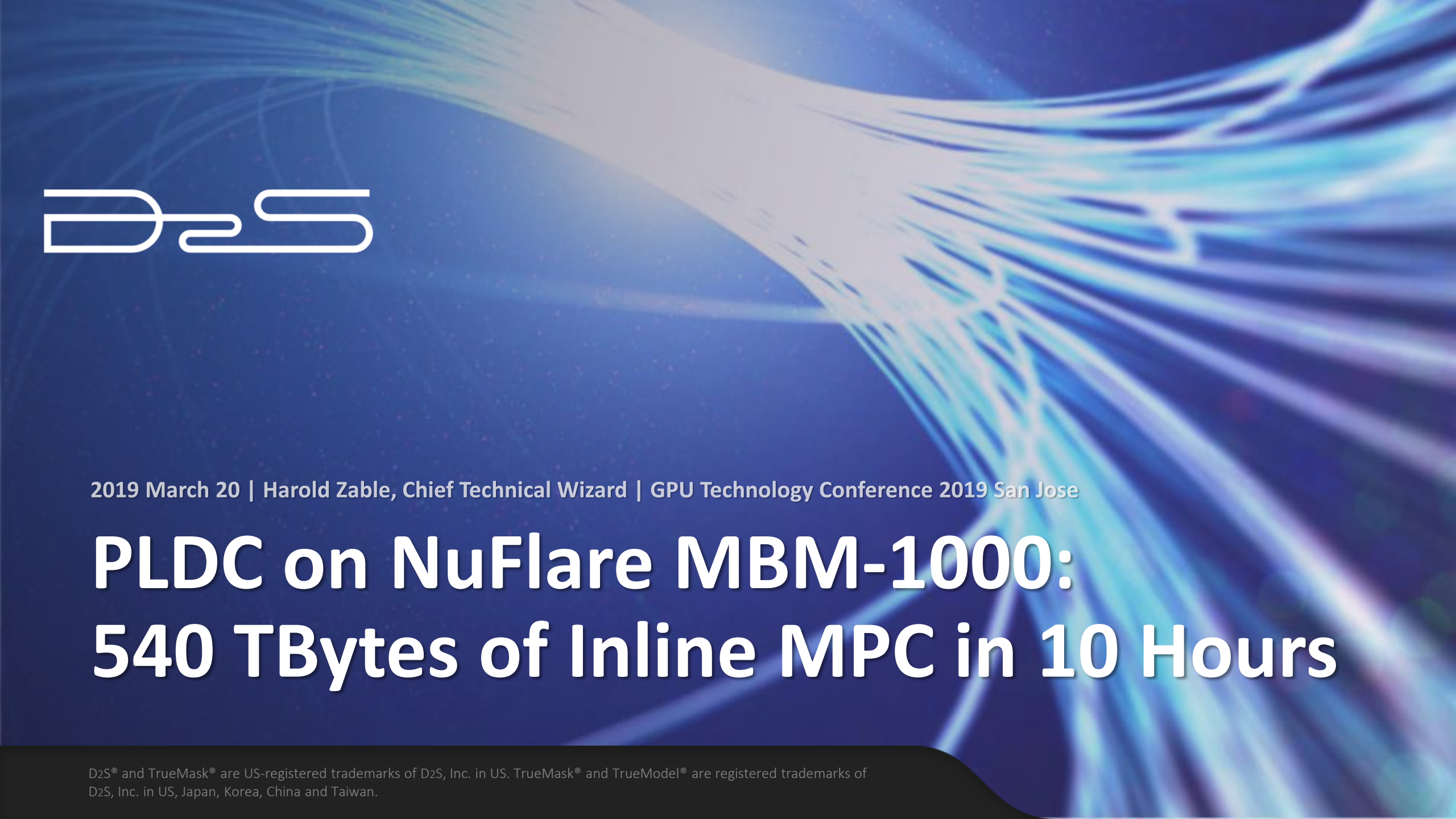
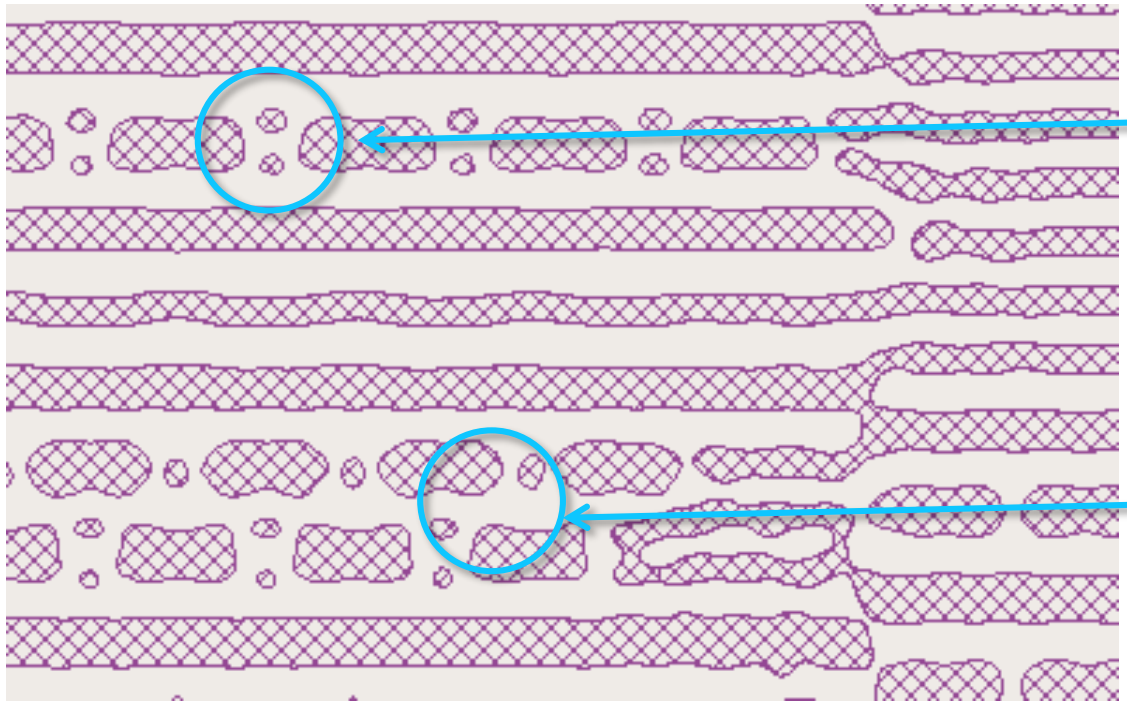


2019 March 20 | Harold Zable, Chief Technical Wizard | GPU Technology Conference 2019 San Jose

PLDC on NuFlare MBM-1000: 540 TBytes of Inline MPC in 10 Hours

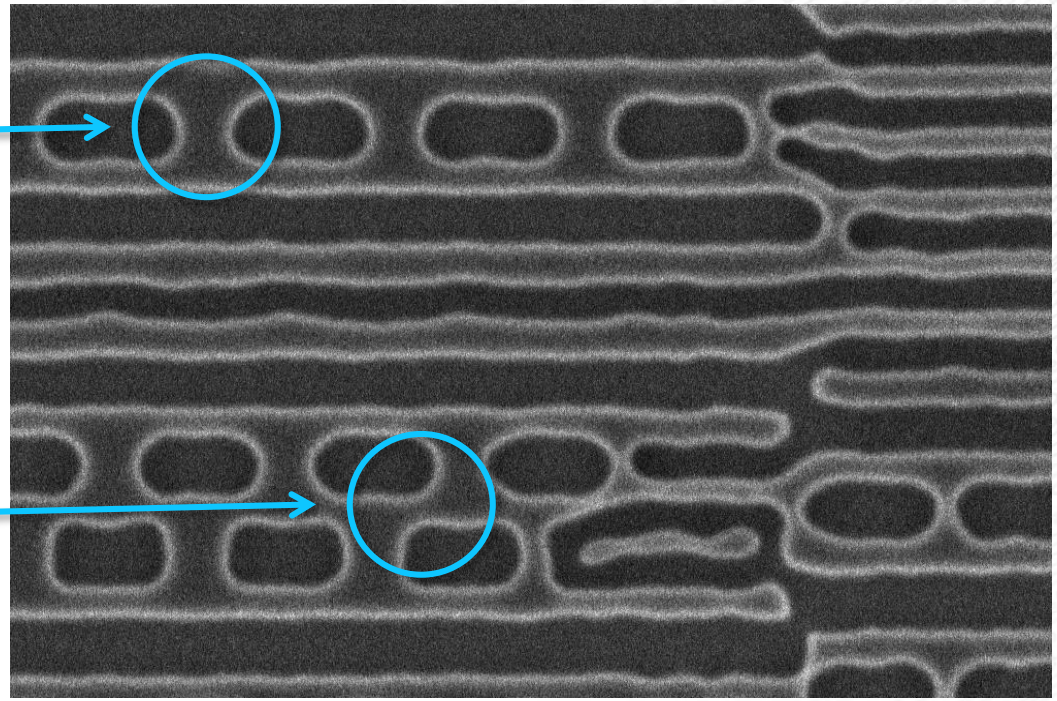
You Can't Always Get What You Want

You want to make this:



Design

You get this:



Electron Microscope Picture

We Start With Many Questions

- We need to ask the big questions about the talk **title**:
  - What is PLDC? Or any of the other acronyms in the title?
  - Who would want to do it?
  - Where and when would we be doing 540 TBytes of MPC?
    - What is MPC?
  - Why is it important to do it inline?
  - How do we use GPUs to do it?
- To answer these questions, we need to talk about how integrated circuits are made, and how that process has changed recently.

Integrated Circuits are Made of Many Layers

- Integrated Circuits are cut out of **wafers**, which are made up of layers of many different materials, stacked on top of each other:
  - Si as substrate.
  - Metal for connections.
  - Diffusion or implant for particular electrical properties.
- Layer designs are extremely complex.

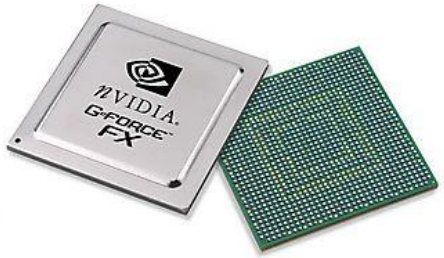
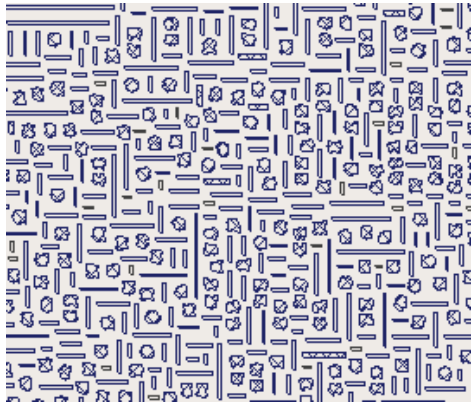


Image Credit Gpureview.com



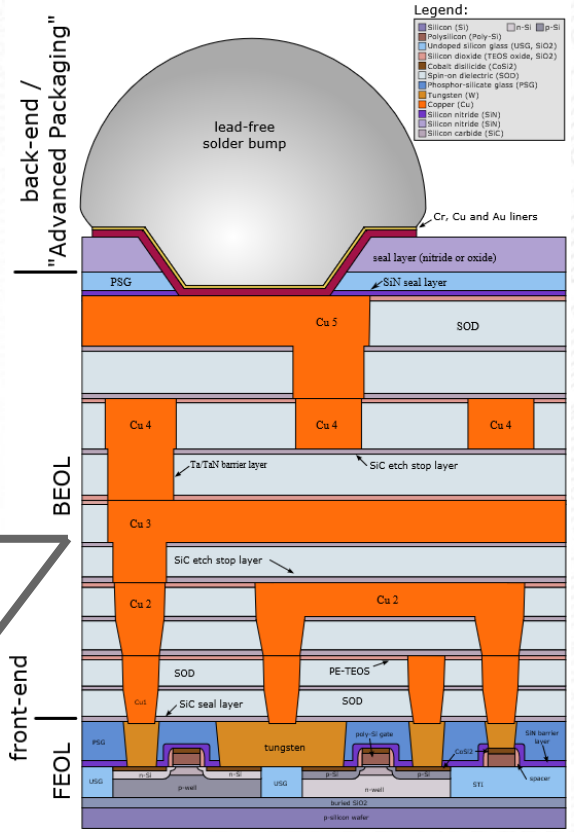
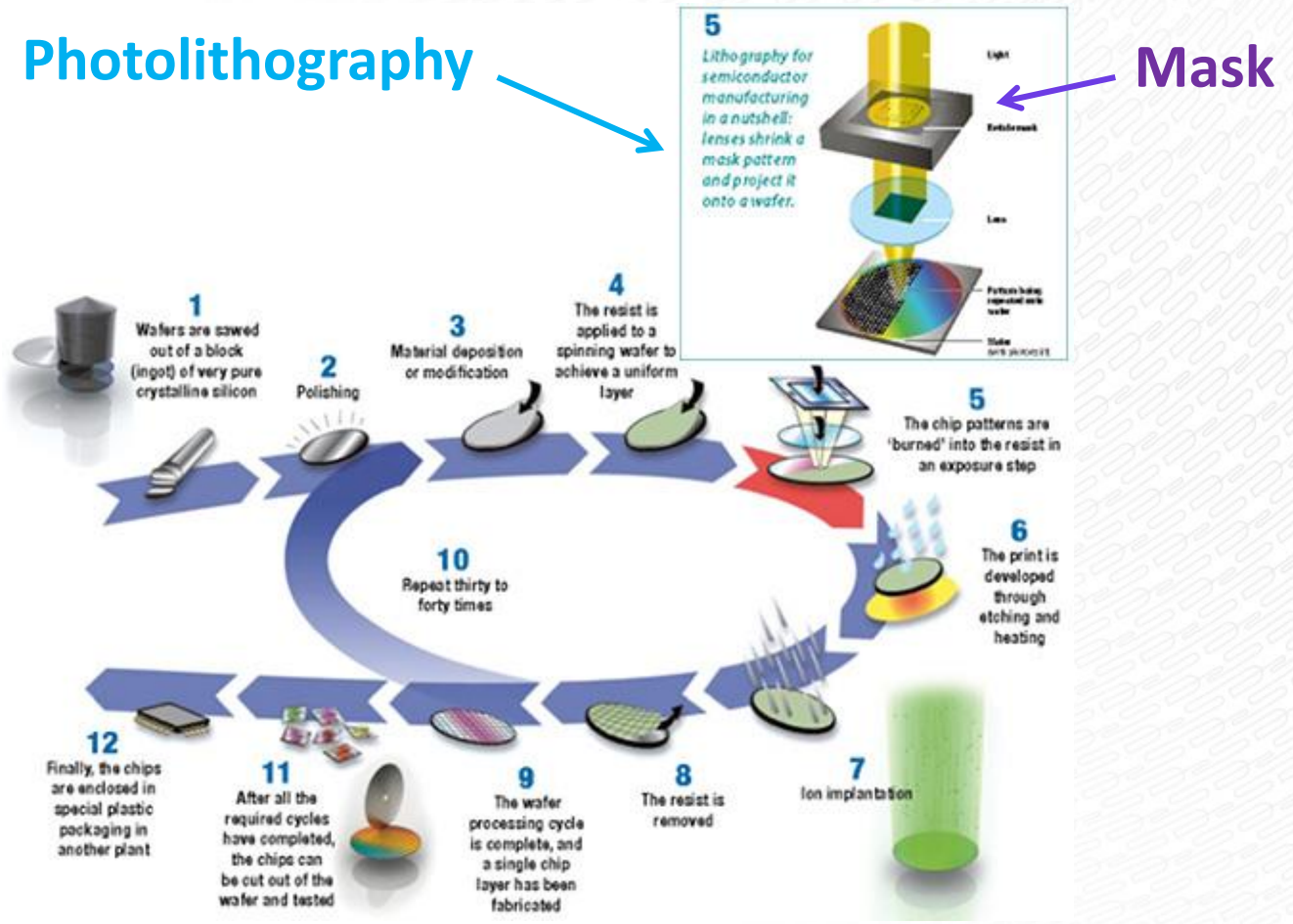


Image Credit Cepheiden, WikiMedia Commons

It Takes Many Steps to Make a Single Layer

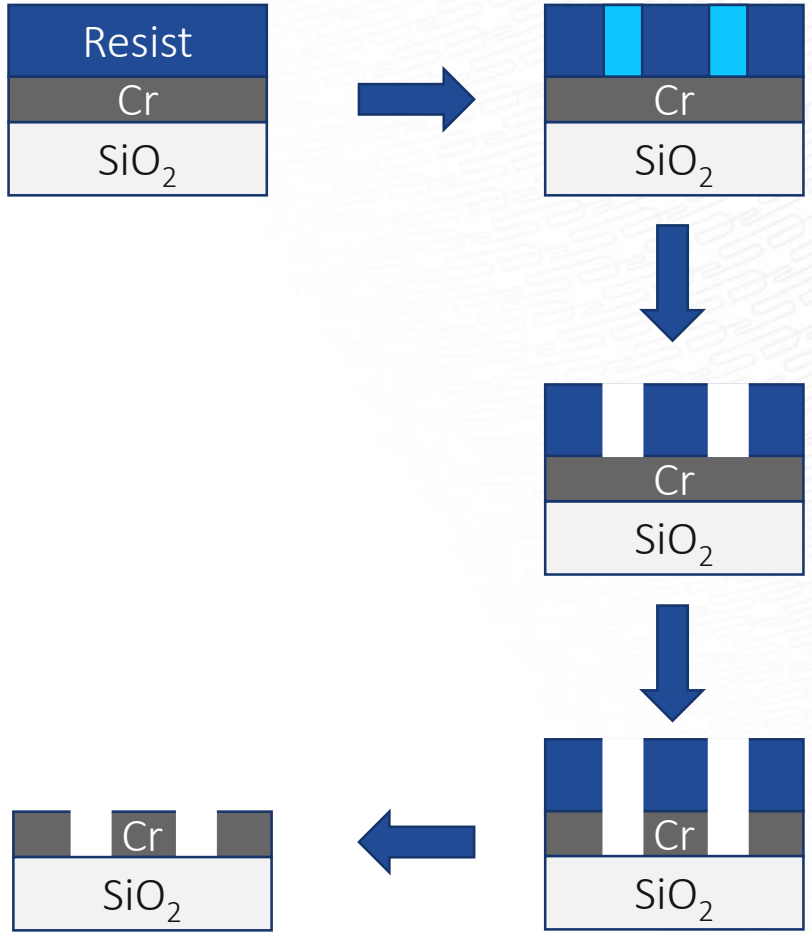
- Each individual wafer layer takes hours to process.
  - It can take a month from tape out to first silicon.
- Fabs process hundreds of wafers each hour, making integrated circuits cheap by mass production.
- Shining light through a **mask** to project an image of the desired pattern onto the wafer is called photolithography.



https://bits-chips.nl/artikel/asml-for-beginners

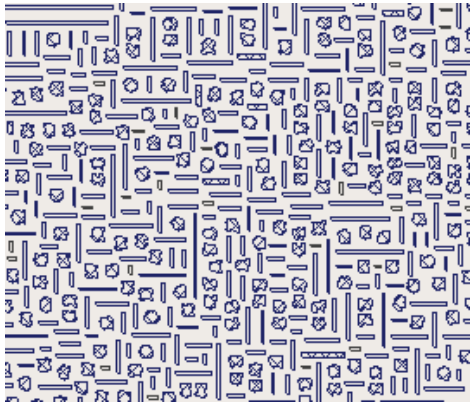
Masks are Made from Exposure, Develop, and Etch

- Making a mask is a completely separate process, much like making a single layer of a wafer:
  - Apply photoresist.
  - Expose the pattern.
  - Develop the photoresist.
  - Etch away material (Cr) underneath.
- Conventional masks are 4x larger than wafer layers.
- EUV masks are made of more complex materials, but the fabrication process is similar.



Masks Are Exposed by eBeam Lithography

- **NuFlare** makes machines that expose mask photoresist using an electron beam (eBeam).
- Software fractures the pattern into rectangles and right triangles.
  - Every component and wire is broken into shots.
- Writing a mask this way can take hours to days.
  - Goal: <**10 hours** per mask.



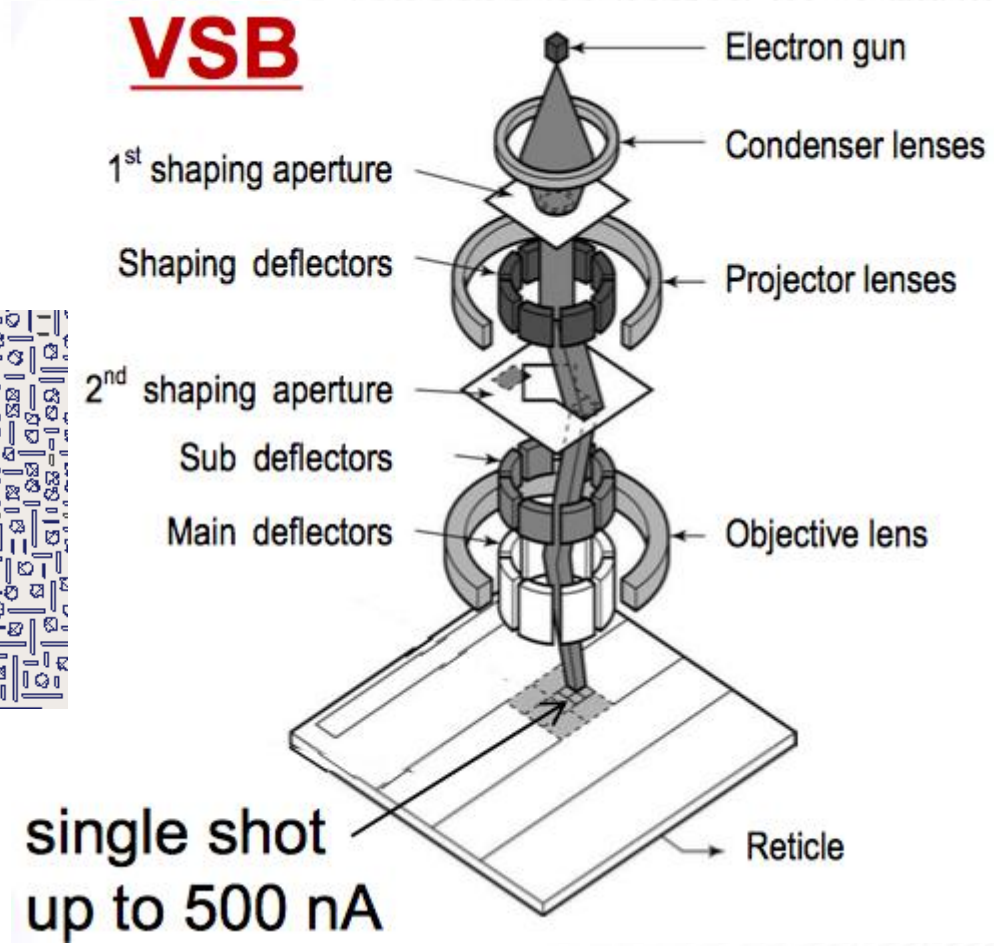
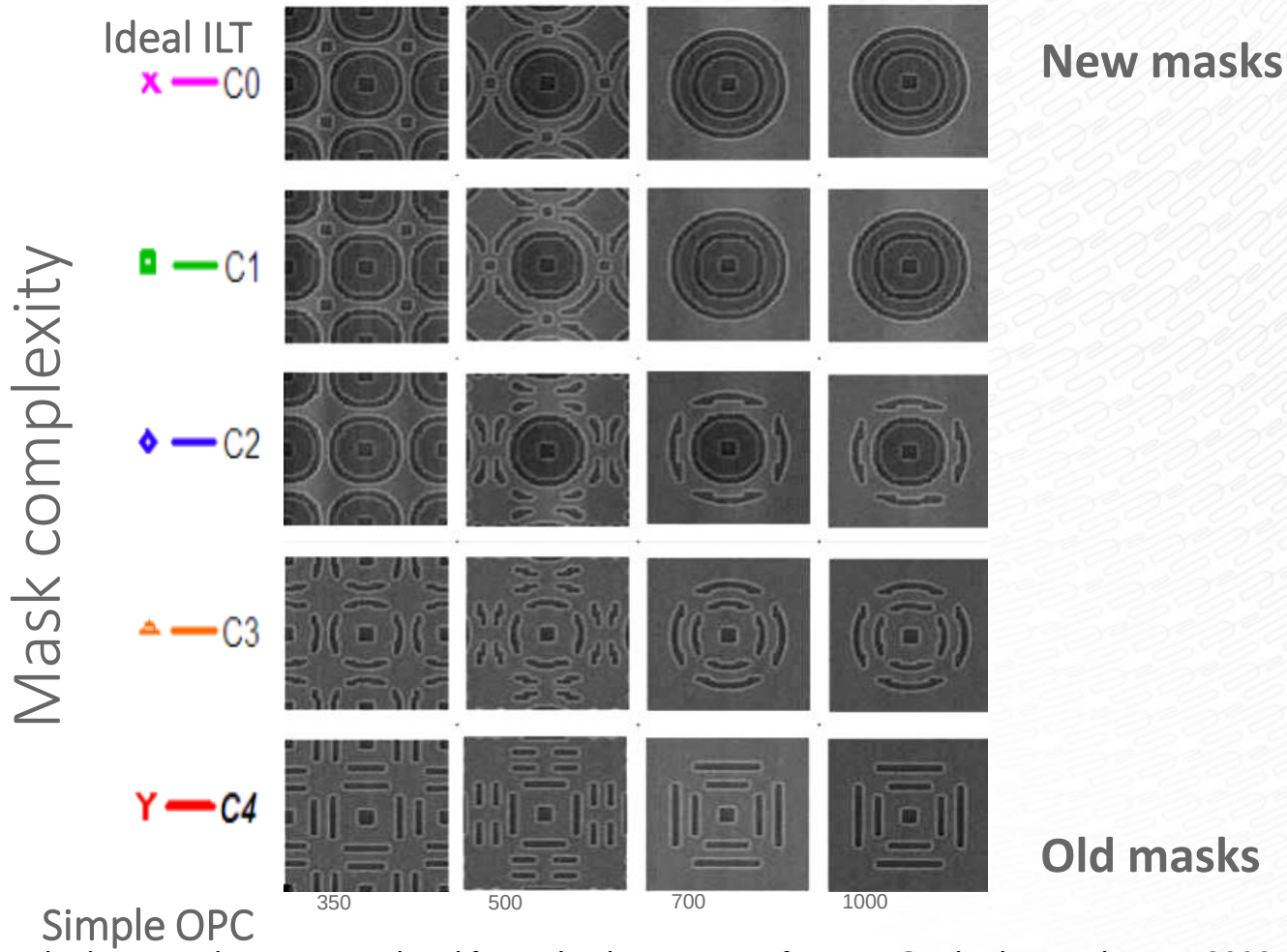


Image from semiengineering.com, 2017-01-08

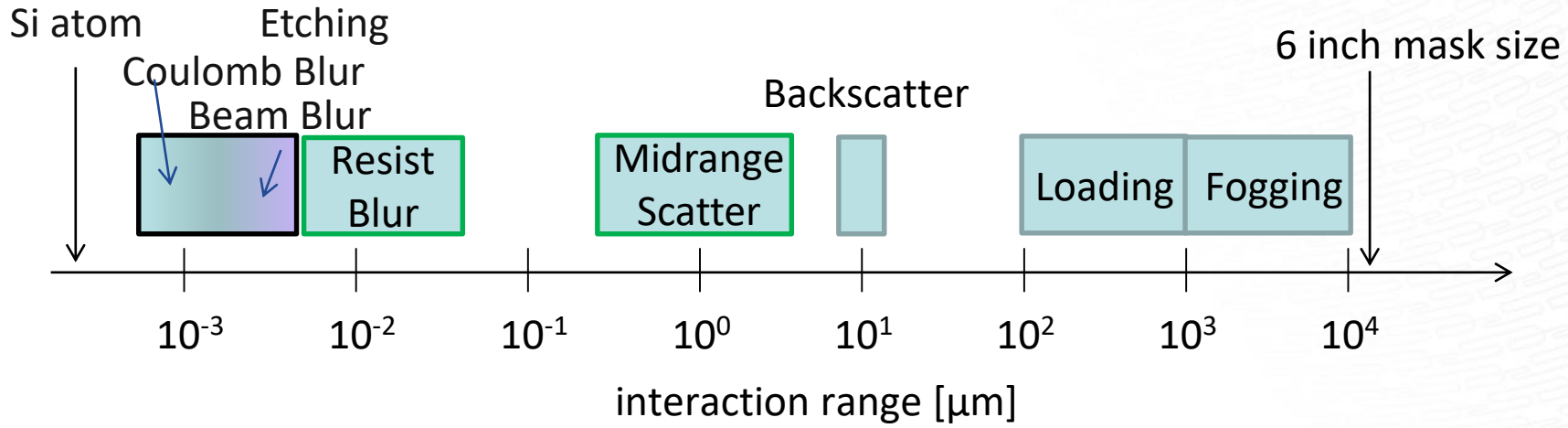
As Decades Pass, Mask Complexity Soars

- VSB mask writers write rectangles.
- Wafer feature size is now substantially smaller than the wavelength of light used to expose it. Techniques that make that possible (OPC, ILT) require drawing irregular curvilinear shapes, not just rectangles.
- Breaking circles down into rectangles will take a **lot** of rectangles.



The base study on conventional fracturing is courtesy of Byung-Gook Kim, et al., PMJ 2009

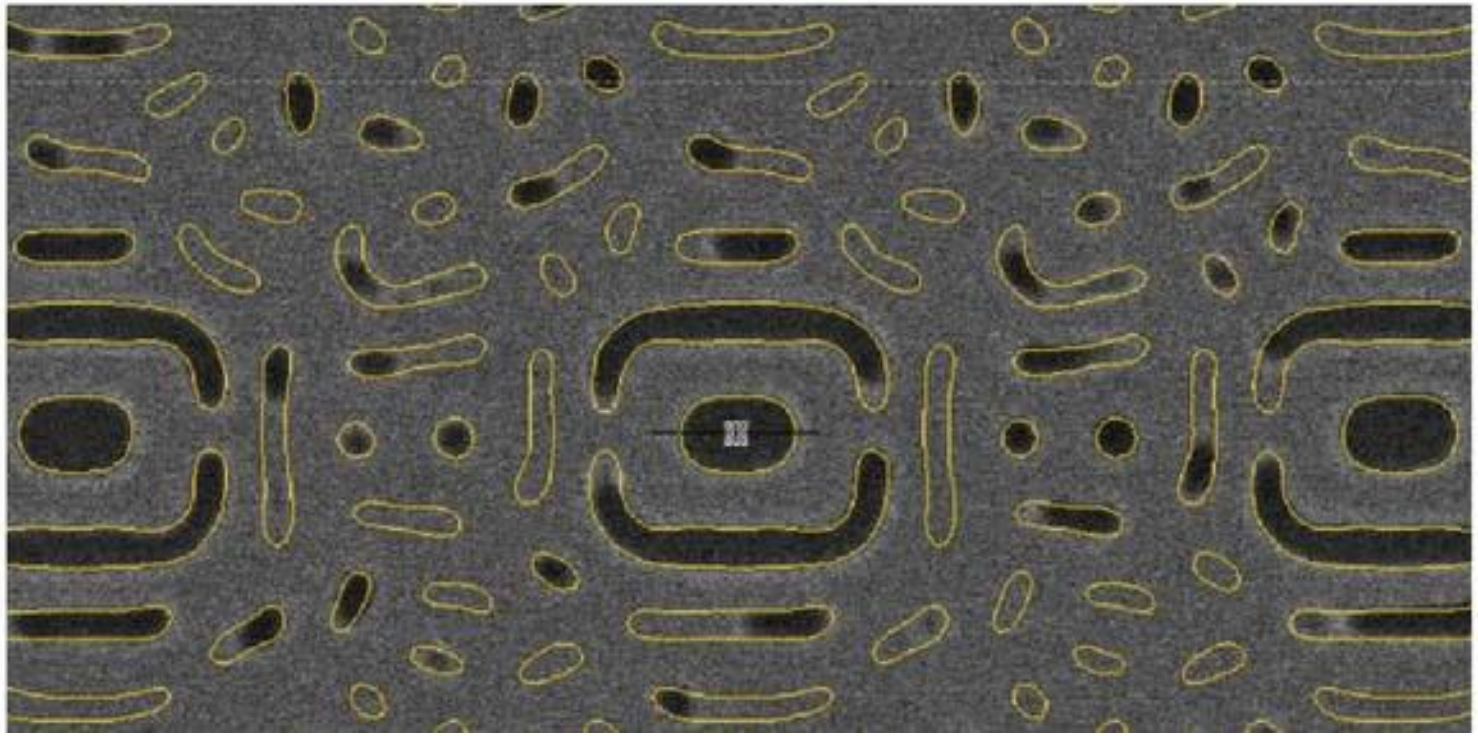
eBeam and Etch Mask Physics is Complicated



- Different physical effects at different length scales.
- Many effects can be modeled with a Point Spread Function (PSF).
- Some (like etching) cannot.
- D2S has a mathematical model called TrueModel® to describe eBeam physics.
  - We use simulation to study these effects.

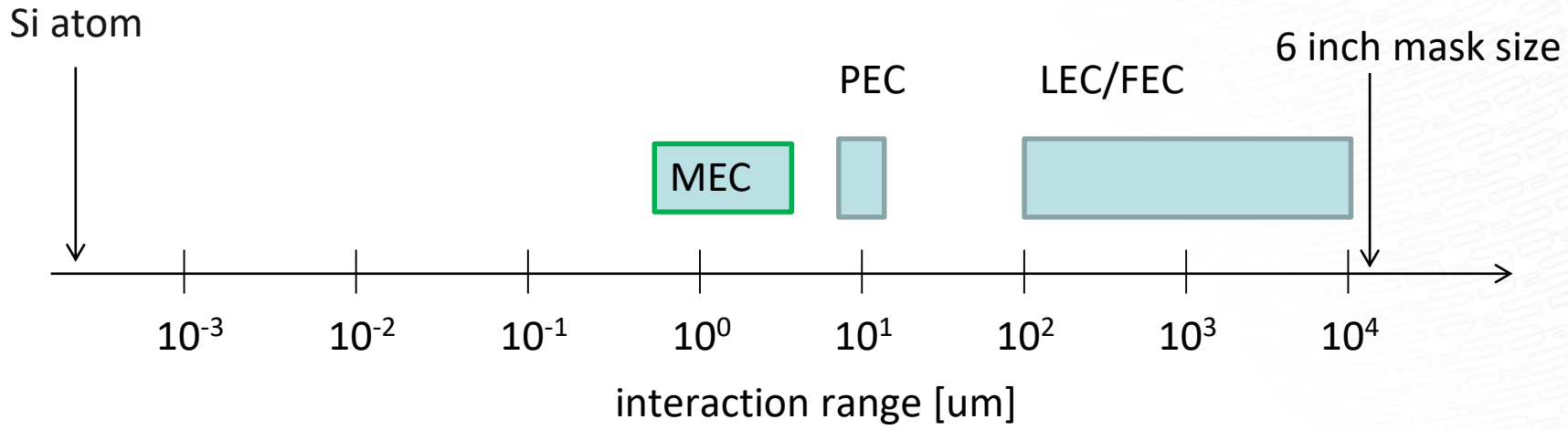
Masks Don’t Always Print Correctly

Features are small enough that you have to consider eBeam and mask physics and compensate for it with Mask Process Correction (**MPC**).



Ingo Bork *et. al*., Proc. SPIE 10810, Photomask Technology 2018

Existing Systems Correct Inline at Certain Scales



- Some types of MPC targeted for particular effects are part of VSB machines, or are **inline** corrections. These include PEC, LEC, FEC, and recently MEC.
- For the rest of the effects, MPC is computed separately by an additional offline system.
  - This system must make a separate pass through the entire mask.

Multi-beam is a New Way to Write Masks

- To deal with the added complexity, NuFlare is now making a 10nm pixel-based machine: the **MBM-1000**.
- Instead of fracturing mask geometry into rectangles, we **rasterize** it into pixels.
- Each pixel has its own dose.

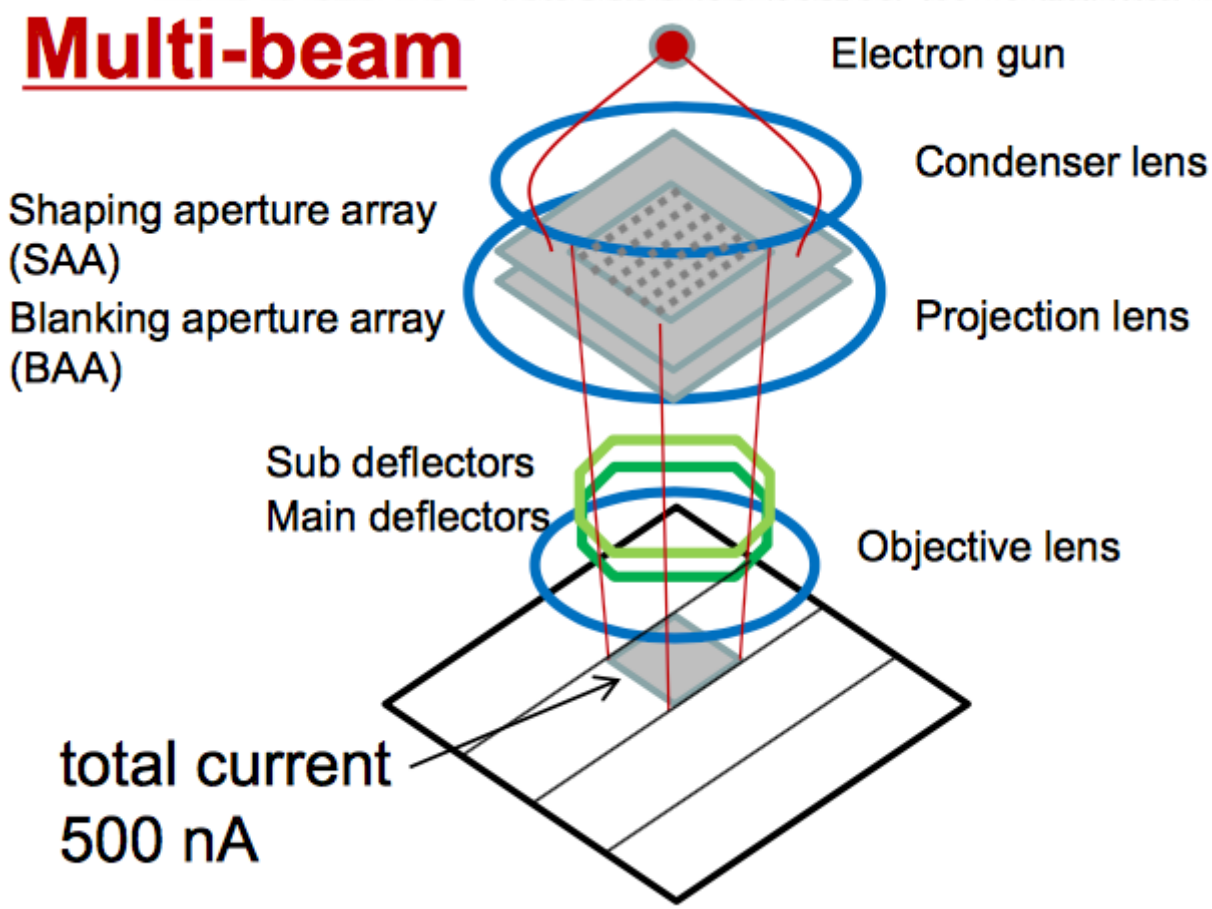
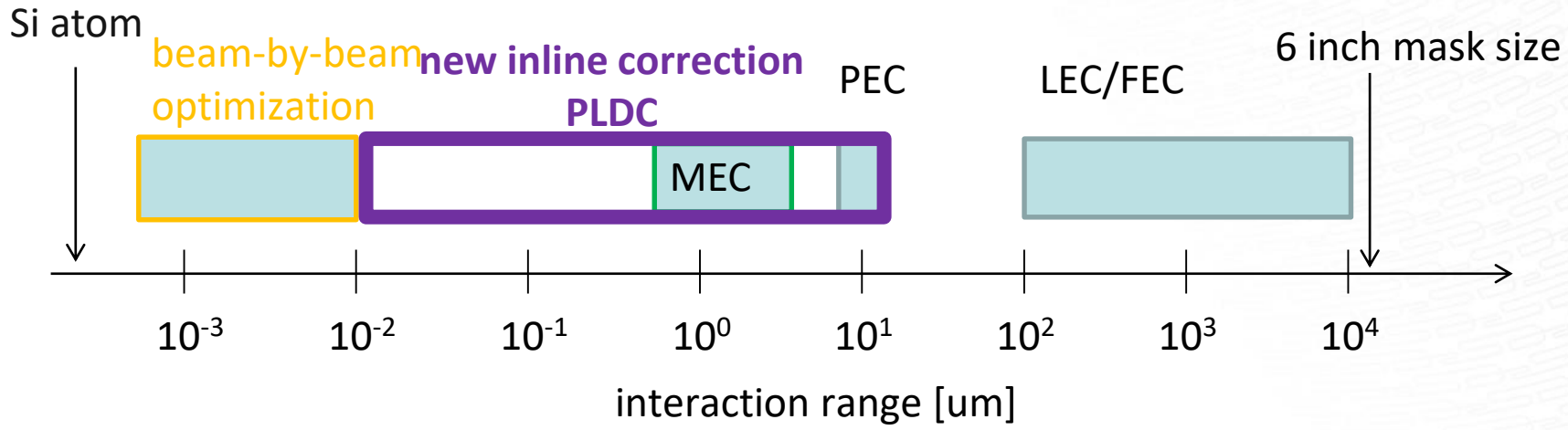


Image from semiengineering.com, 2017-01-08

Different Systems Correct at Different Scales



- With the arrival of the Multi-beam machines NuFlare has also introduced Pixel Level Dose Correction, or **PLDC**.
- PLDC works because we have access to all the pixel data, so we can adjust all the pixels individually to compensate for physical effects.

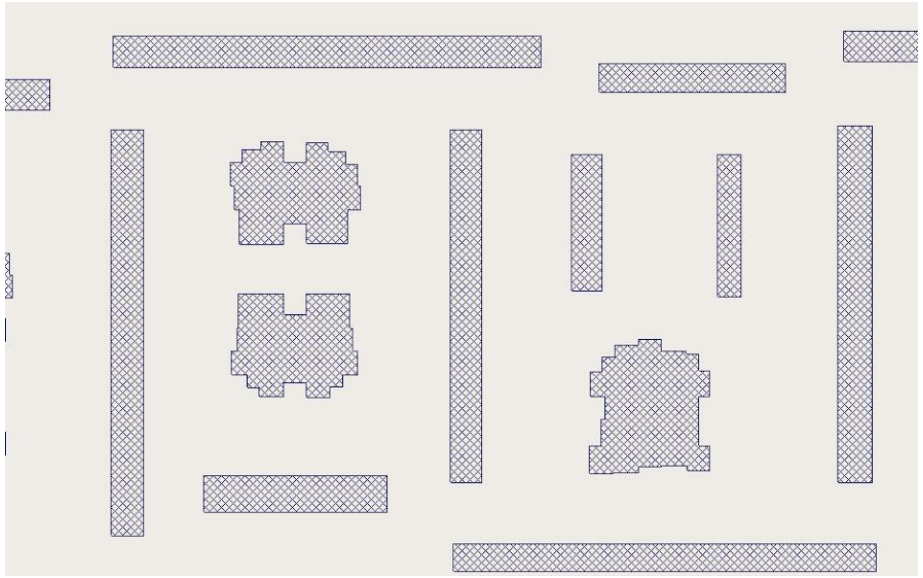
PLDC Requires a Huge Amount of Data

- Printable mask area is 100mm x 130mm.
- Pixel size is 10nm x 10nm.
- NuFlare's machine prints at 2 shifted locations.
- Pixel intensity data is 2 bytes/pixel.
- $(100*10^{-3}*135*10^{-3})/(10*10^{-9}*10*10^{-9})*2*2 = 540*10^{12}$ = **540 TB**.
  - That's 100,000 DVDs, 200 ImageNets, or $100K of consumer-grade SSDs.
  - Take a 16 Mpixel camera, replace each pixel with a camera, and count pixels.
- We must be able to write the mask in **10 hours**.
  - 540 TB / 10 h = 15 GB/s = 7.5 pixels/ns.
- FDR/QDR InfiniBand connections are 3-4 GB/s. Use 3-5 connections.

D2S Computational Design Platform (CDP)

- PLDC was implemented using the D2S 5th generation CDP:
  - 888 TFLOPS (SP)
  - Water-Cooled
  - Reliable, Redundant, Recoverable for 24/7 Clean Room

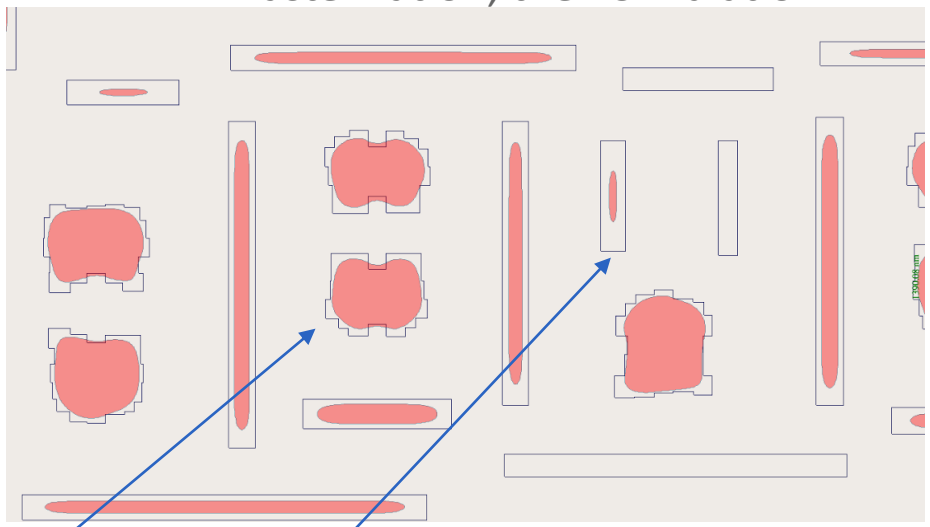
What Does PLDC Look Like?



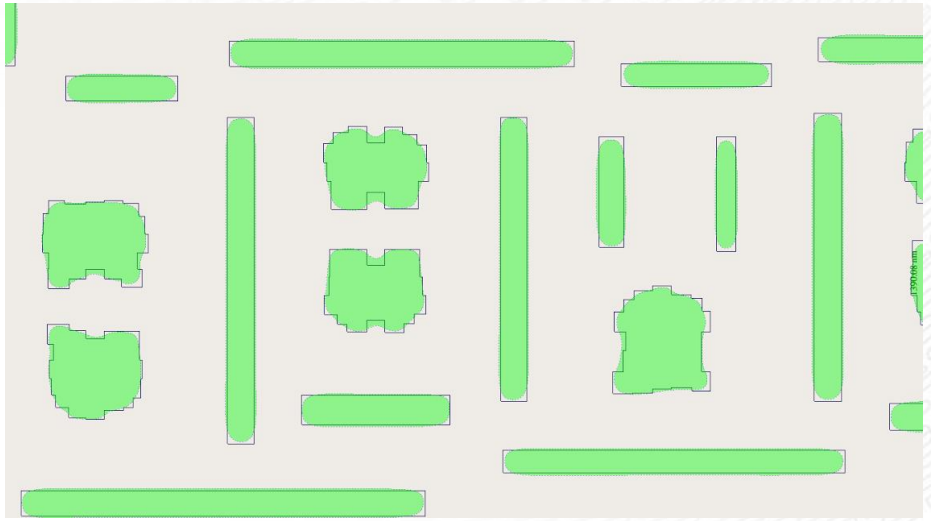
- Since PLDC works at a pixel level, we can display eBeam intensity as grayscale.
- PLDC tends to enhance pixels near the edges, so interior pixels appear dim.
- Larger scale corrections also tend to reduce pixel values, as exposure energy comes from farther away.

Simulation Shows that PLDC Should Work

Rasterization, then Simulation



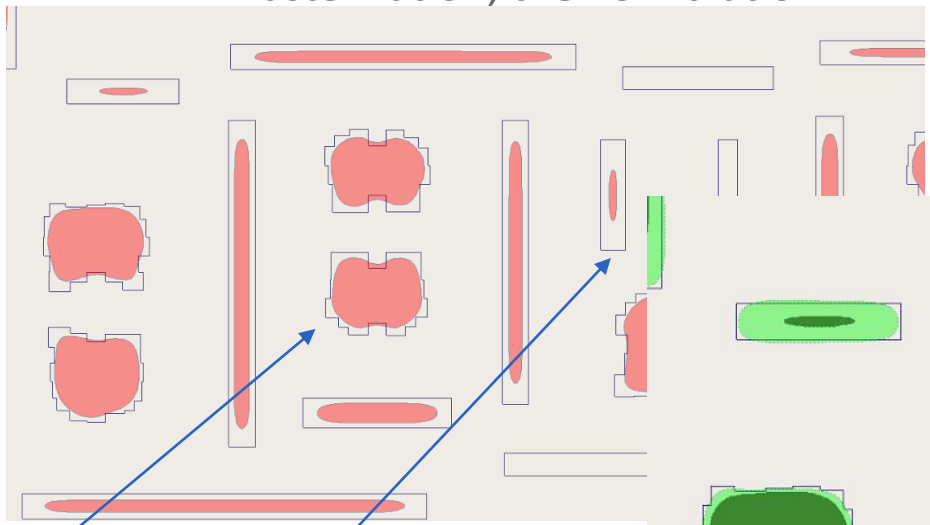
PLDC, then Simulation



- Pure rasterization misses some features entirely. PLDC includes them.
- PLDC adjusts figures individually. All shapes and sizes improve.
- Even within each shape, different areas get appropriate improvements.

Simulation Shows that PLDC Should Work

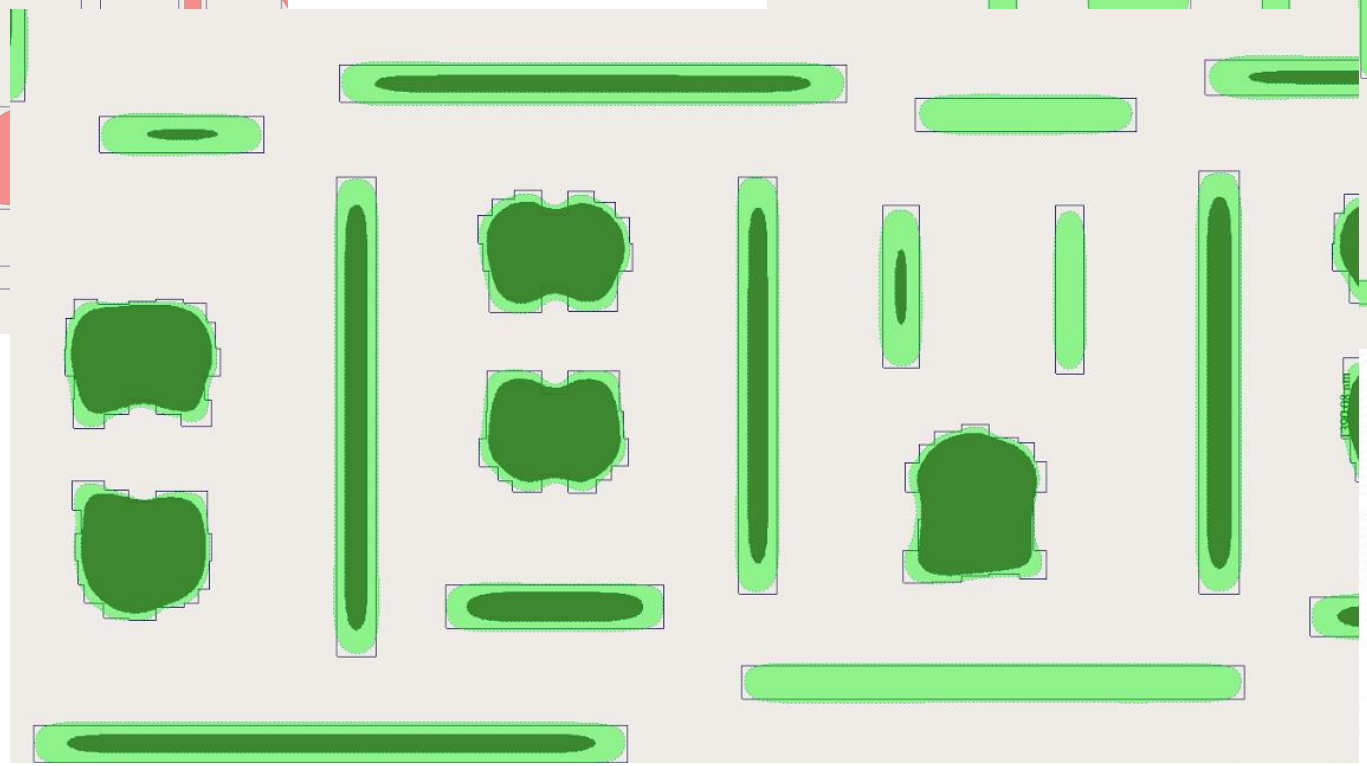
Rasterization, then Simulation



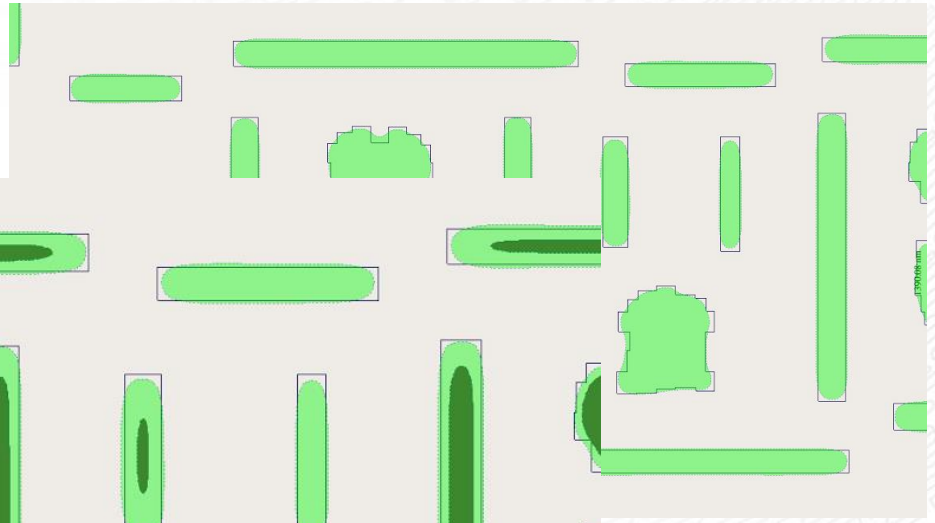
218x244nm

63.7nm

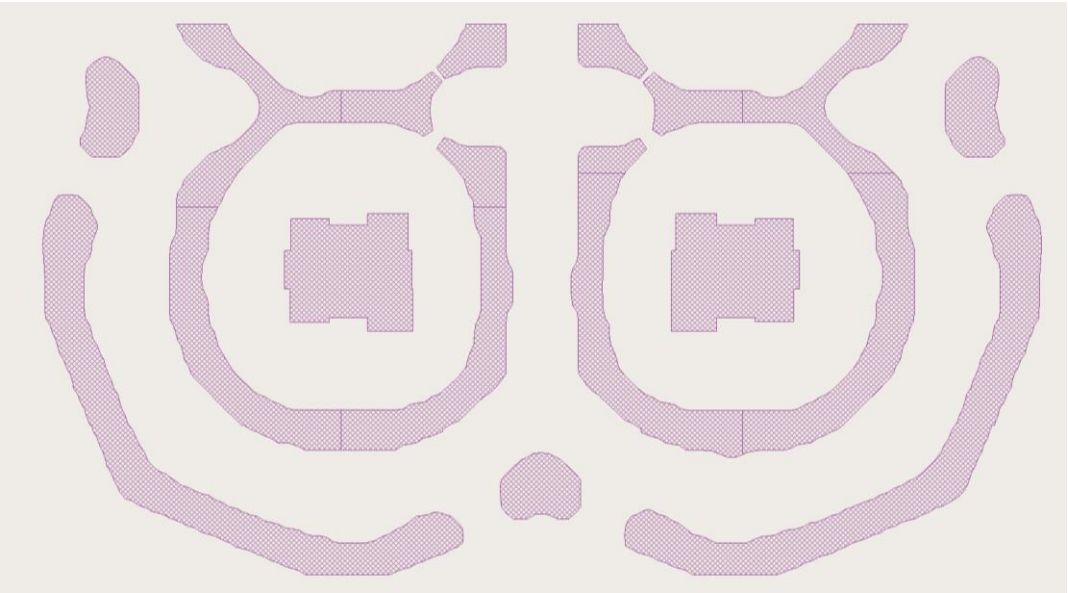
Both, Overlaid



PLDC, then Simulation



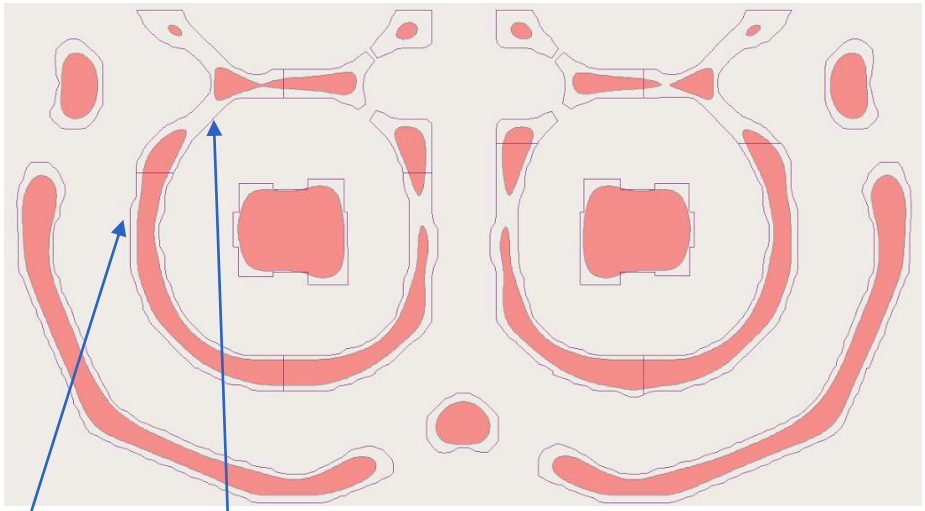
PLDC Works for Curvilinear Shapes



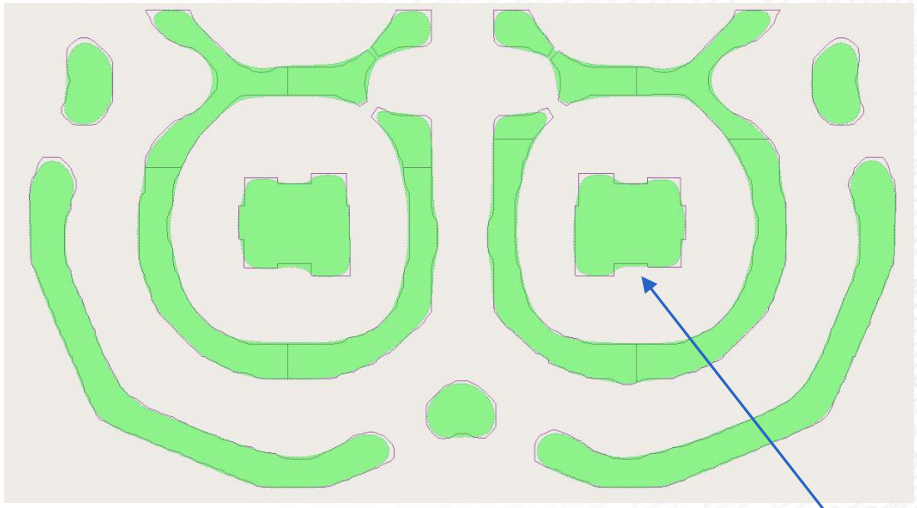
- ILT and other Resolution Enhancement Technologies produce curvilinear results.
- PLDC works on pixels, so it can be applied to these newer, more complex shapes as easily as older, rectangular shapes.

Simulation Shows PLDC Curvilinear Enhancement

Rasterization, then Simulation



PLDC, then Simulation

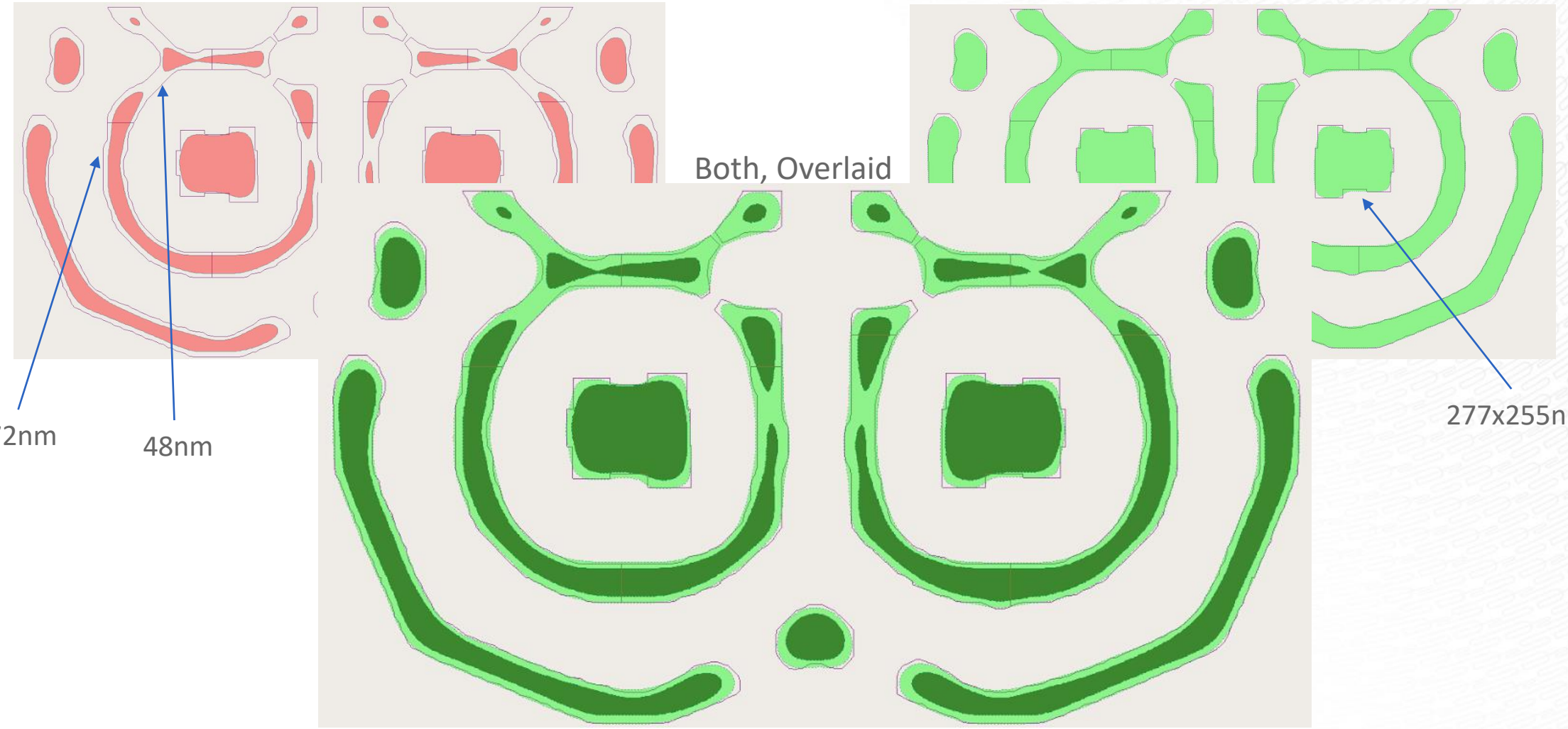


- The value of pixel-level enhancement is clear within individual figures.
- Gaps at narrow points of the shapes are filled in, while places needing only minor adjustment are gently enhanced.

Simulation Shows PLDC Curvilinear Enhancement

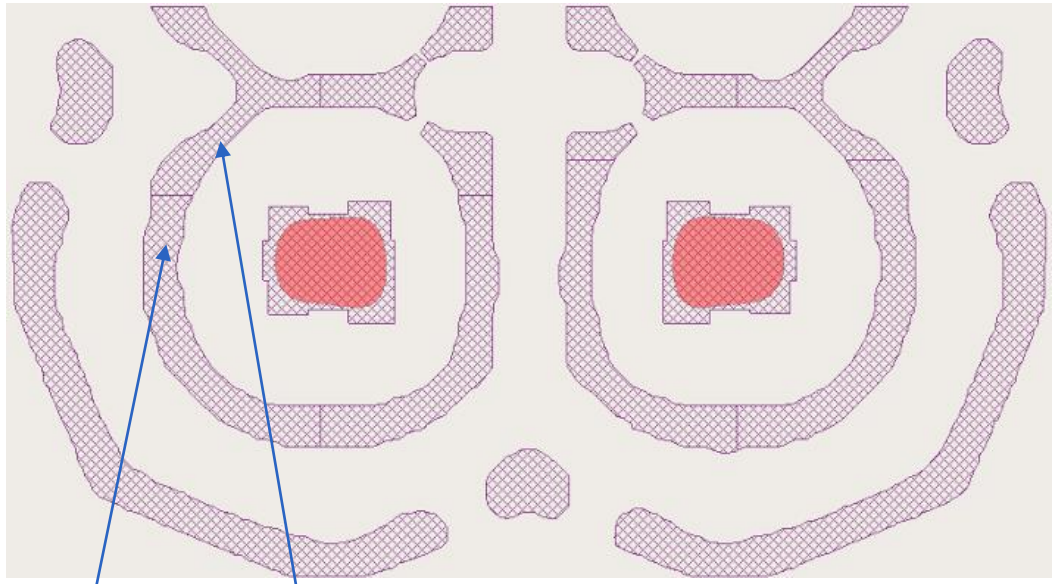
Rasterization, then Simulation

PLDC, then Simulation

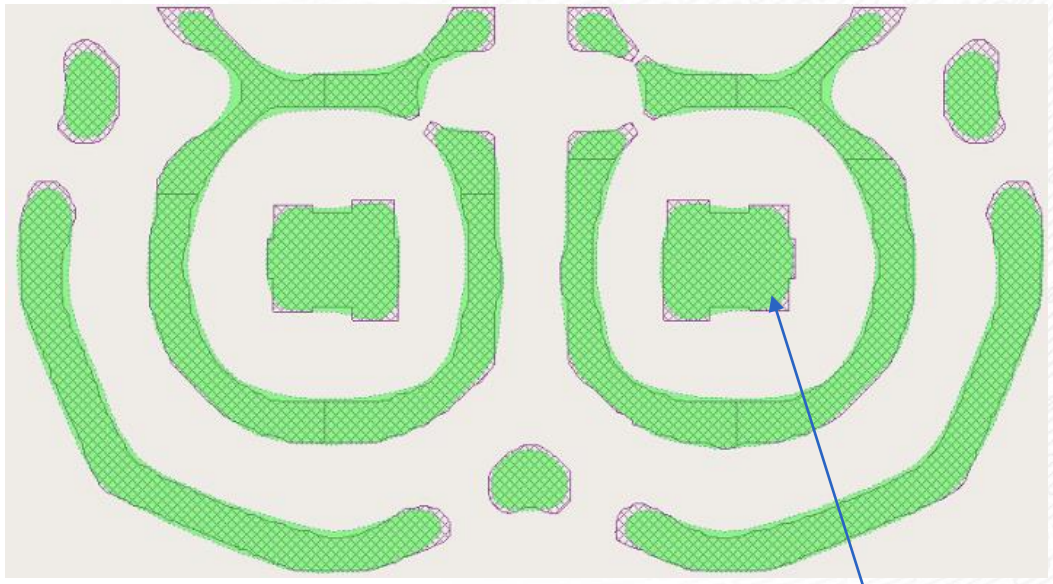


PLDC Enhances Curvilinear Printability

Rasterization, then Simulation



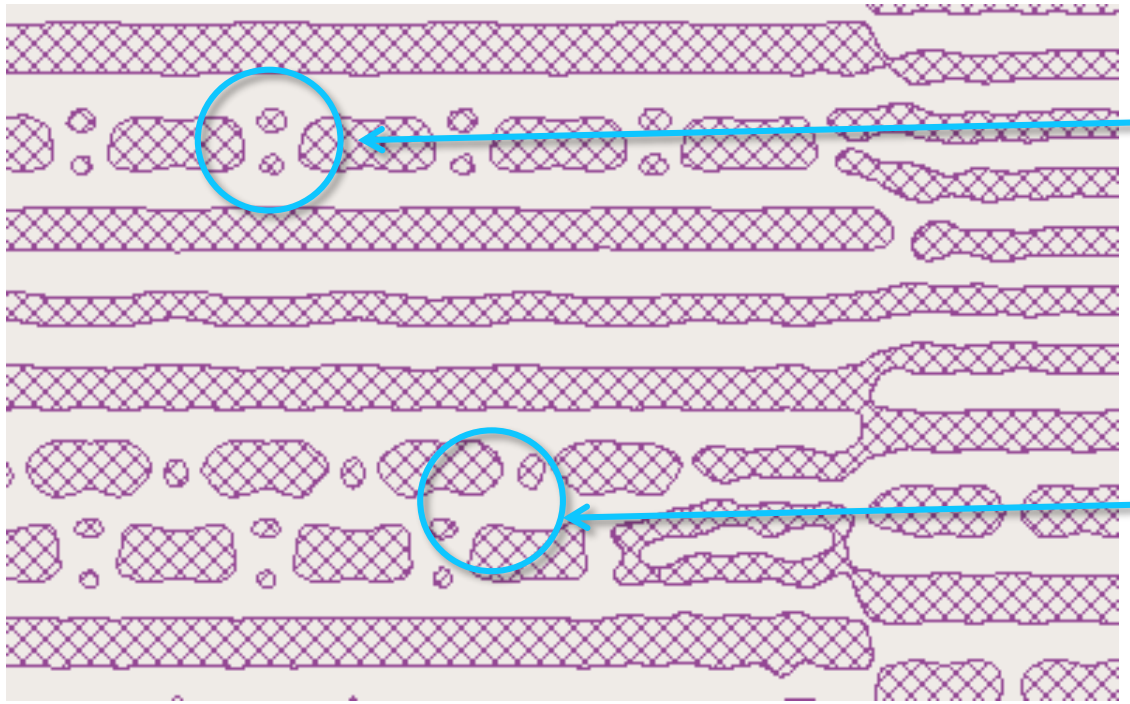
PLDC, then Simulation



- At 2/3 the previous size, features that were completely missing with raw rasterization are present and properly shaped with PLDC.
- PLDC allows mask making machines to print more detailed masks.

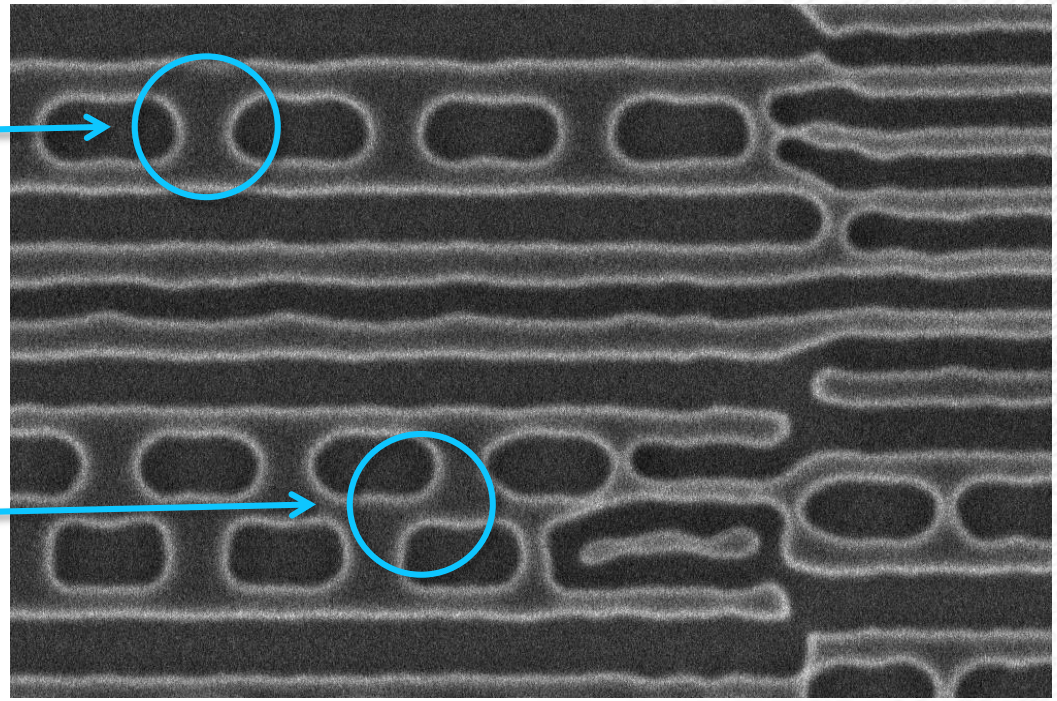
You Can't Always Get What You Want, But…

You want to make this:



Design

Without PLDC, you get this:

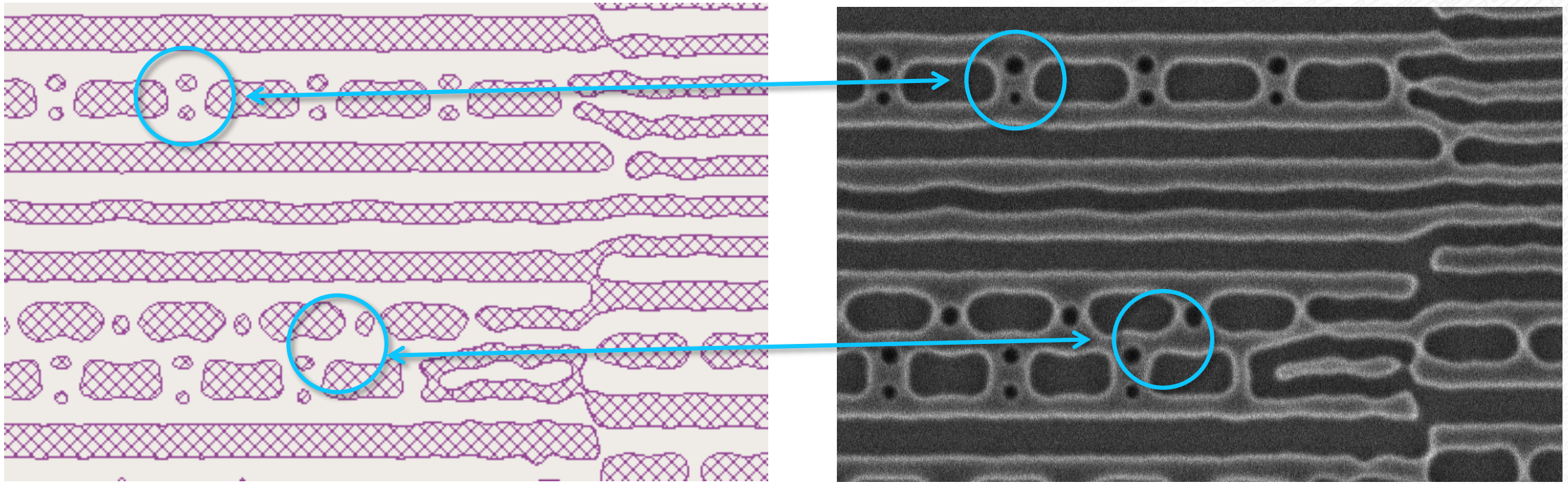


Electron Microscope Picture

If You PLDC… You Get What You Need

You want to make this:

With PLDC, you get this:

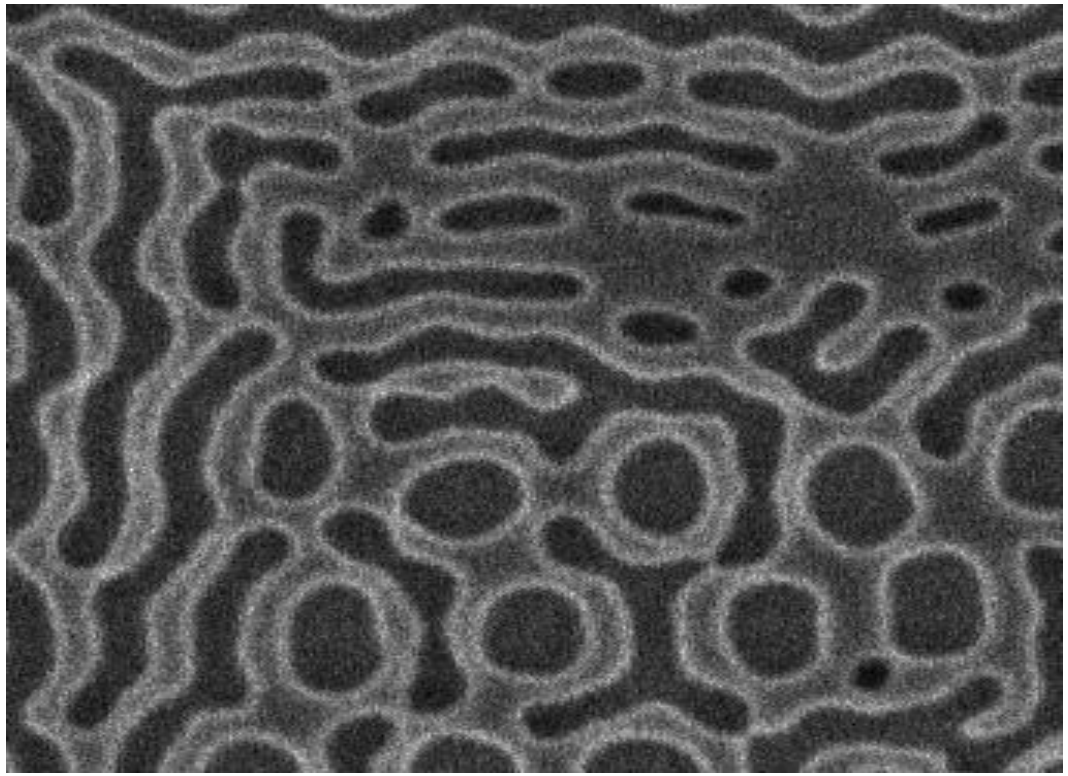


Design

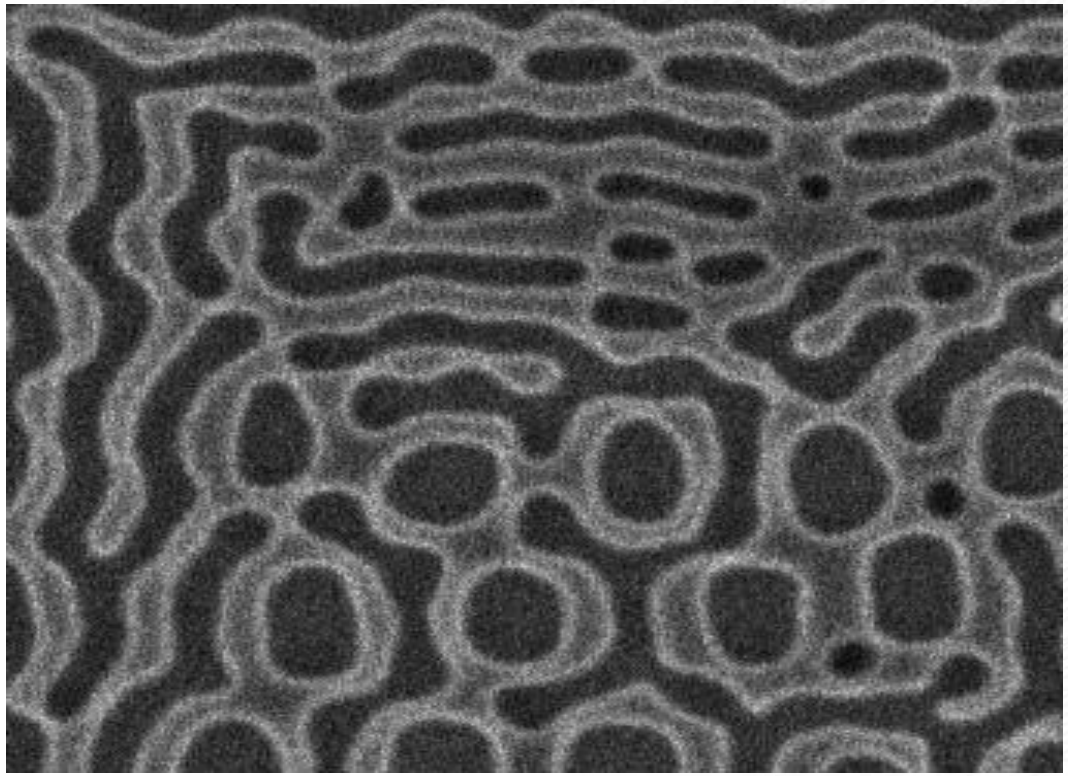
Electron Microscope Picture

Under the same exposure conditions, PLDC fills in the details.

PLDC Fills In Missing Features?

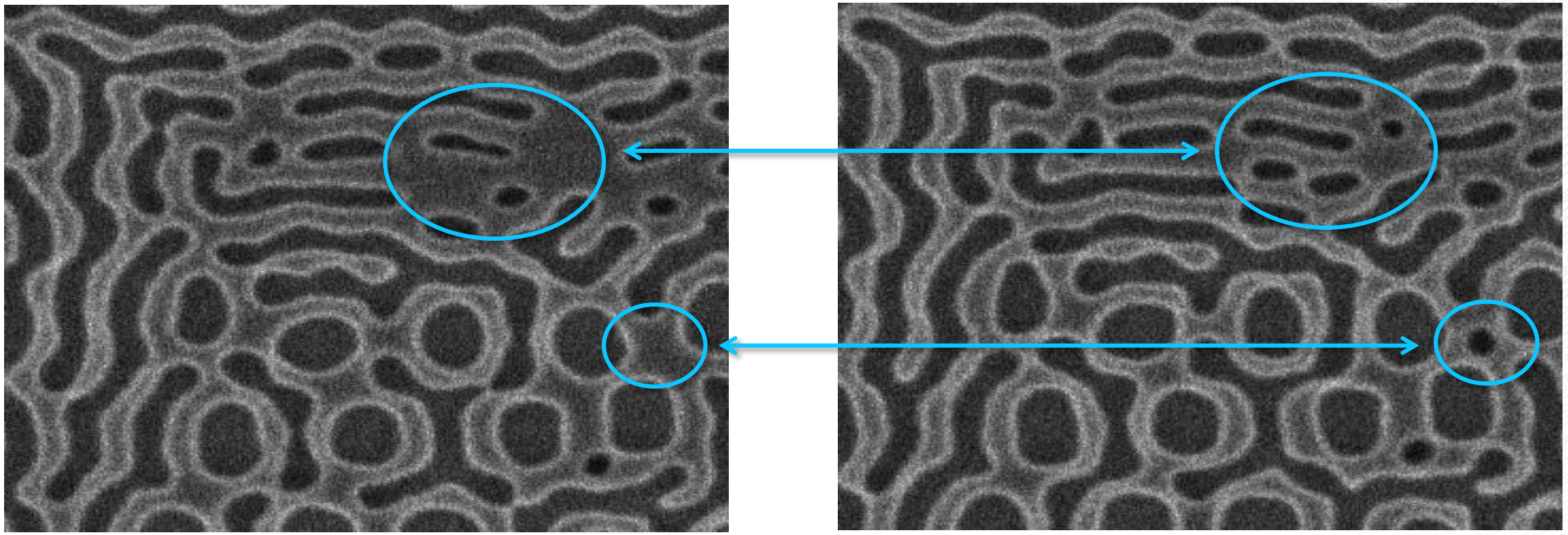


Without PLDC



With PLDC

PLDC Fills In Missing Features!



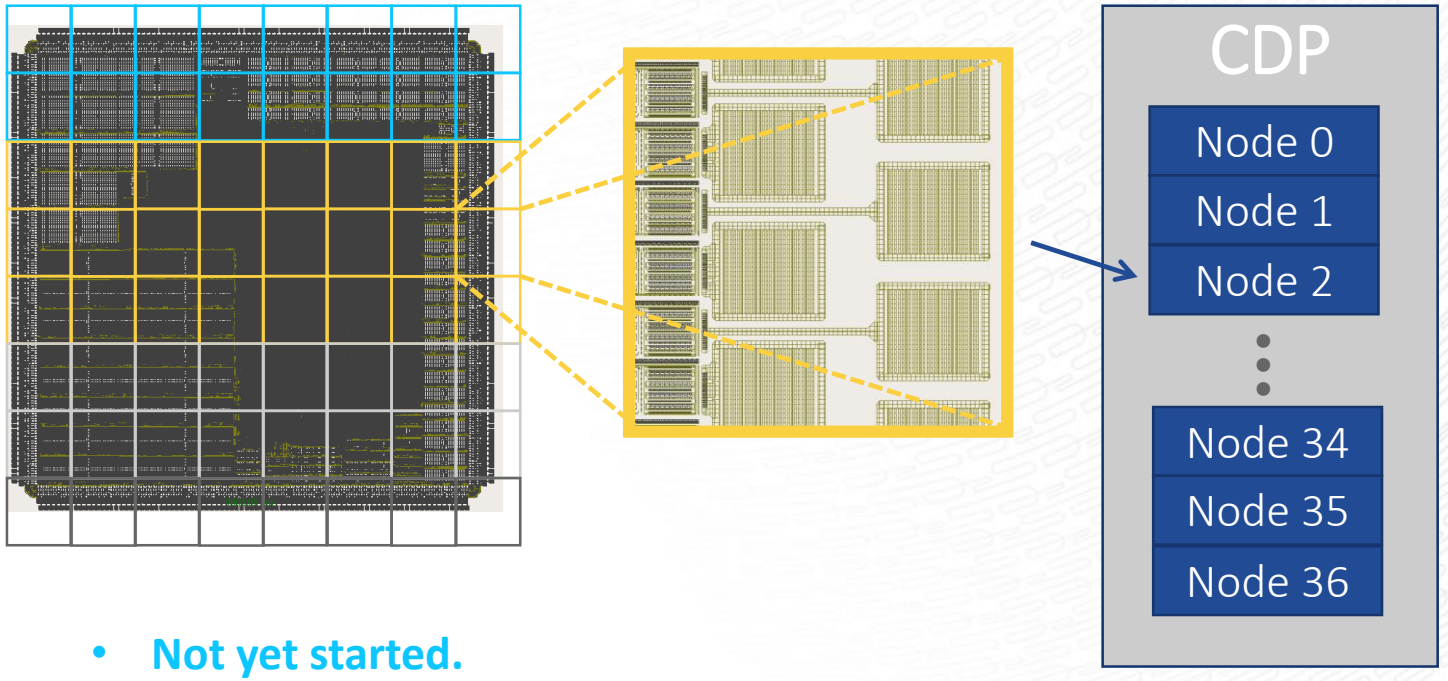
Without PLDC **With PLDC**

Answering the Big Questions

- We've answered five of the six big questions about PLDC:
  - Who? Mask makers building state-of-the-art integrated circuits.
  - What? Pixel Level Dose Correction calculates values for all 540 TBytes
  - Where? On a CDP (GPU/CPU cluster) as part of a multi-beam mask writer.
  - When? Inline with mask writing to save time and money.
  - Why? To achieve optimal results based on eBeam mask physics.
- For the rest of the talk, I'll explain How we use GPUs for PLDC.

Dividing up the Work I: Geometrically

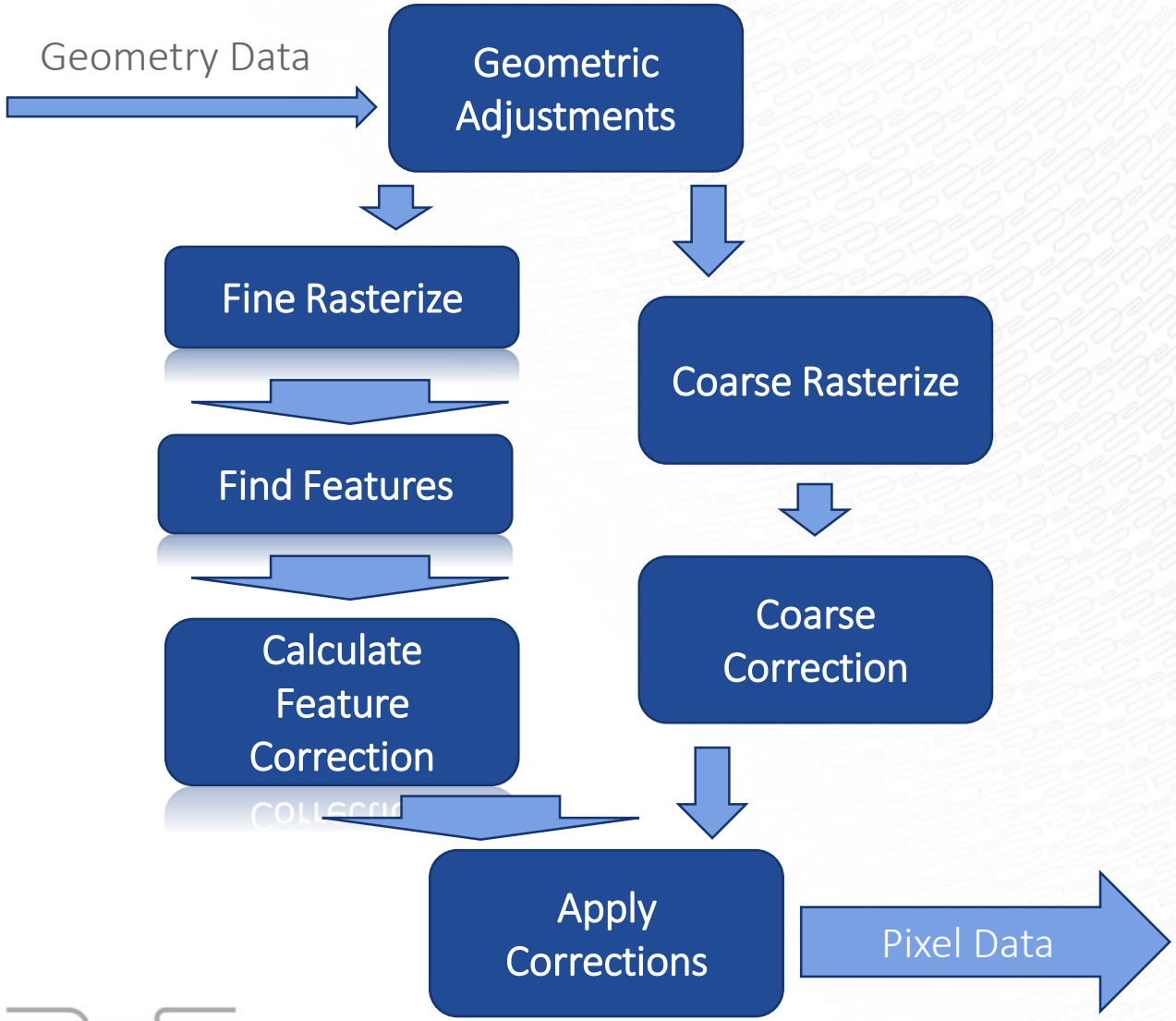
- The simplest way to divide up the work is geometrically. Different partitions are assigned to different nodes.
- Our inline process stays just ahead of the eBeam mask writing.
- Load balancing is achieved with a custom queueing system, reserving memory and processing time for both GPUs and CPU cores.



- **Not yet started.**
- **Computation in process.**
- **Buffered for writing.**
- **Completed.**

Dividing up the Work II: Coarse vs. Fine

- CPU reads the geometry. GPU makes adjustments.
- Coarse and fine rasterization and corrections are calculated on separate CPU processes with GPU help.
- The results of all these corrections are combined by CPU before output.
- Each chunk of work is designed to keep together SIMD calculations for optimal GPU usage.



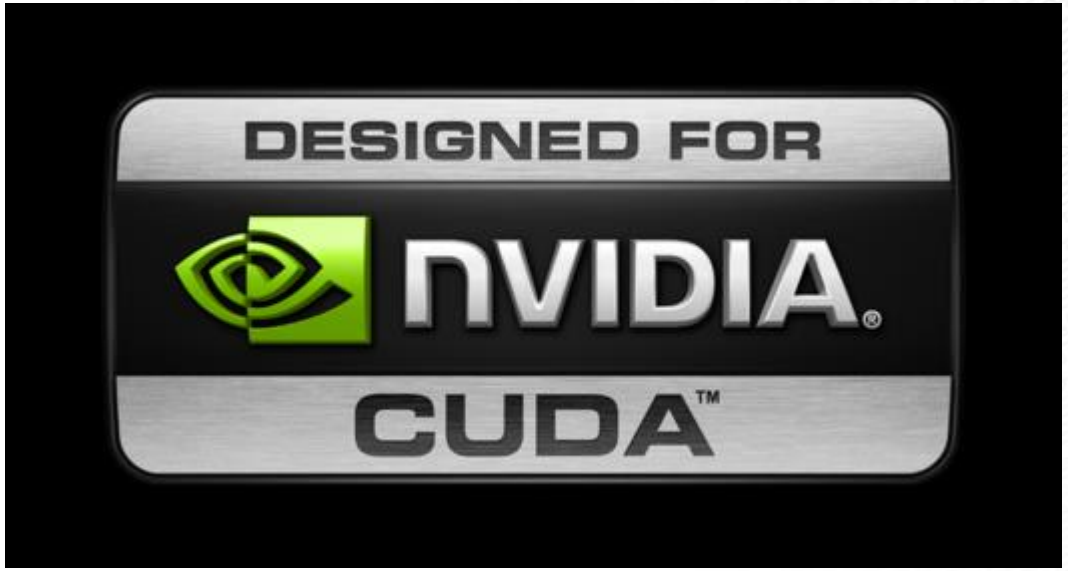
System Reliability

- The D2S CDP (compute cluster) is a small part of a multi-million dollar machine that is expected to be up 24/7.
- System is built with redundancy and “no single point of failure”
- The custom queueing system monitors all partitions for failure or timeout and can restart any partition if needed.
- Check all error codes from CUDA calls and Linux system calls.
- Failure is not an option.

CUDA Best Practices

- At D2S, we've been writing CUDA code for over 10 years.

- Our code is used in production.
  - Our systems are reliable.
  - Our systems are fast and efficient.
  - Our code must last beyond a single research project.

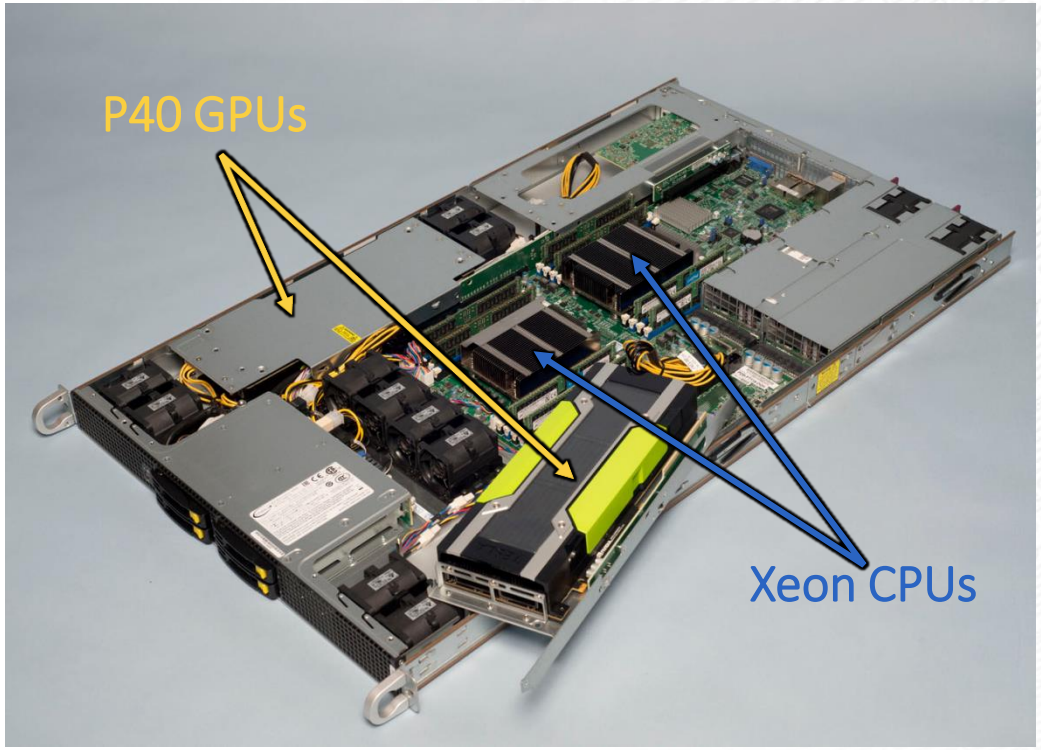


CUDA Best Practices: Software Hygiene

- Check every return value.
  - CUDA errors persist until you clear them.
  - Debugging and recovery require you catch errors quickly.
  - Check CUDA kernel calls, too!
- Wrap every call.
  - We use a C++ wrapper with CUDA errors converted to exceptions.
  - Template wrappers for memory allocation restore C++'s strong typing.
- Use memory arenas.
  - cudaMalloc() can be slow, so batch allocations are often useful.

CUDA Best Practices: Sharing GPUs

- D2S nodes have multiple GPUs.
- We have multiple jobs running on each node.
- We generally use advisory locks for GPU allocation.
- Memory allocation constraints led to the creation of an interprocess wrapper around memory allocation.



One node from a CDP with multiple GPUs and CPUs.

CUDA Best Practices: Use Weird Memory Modes

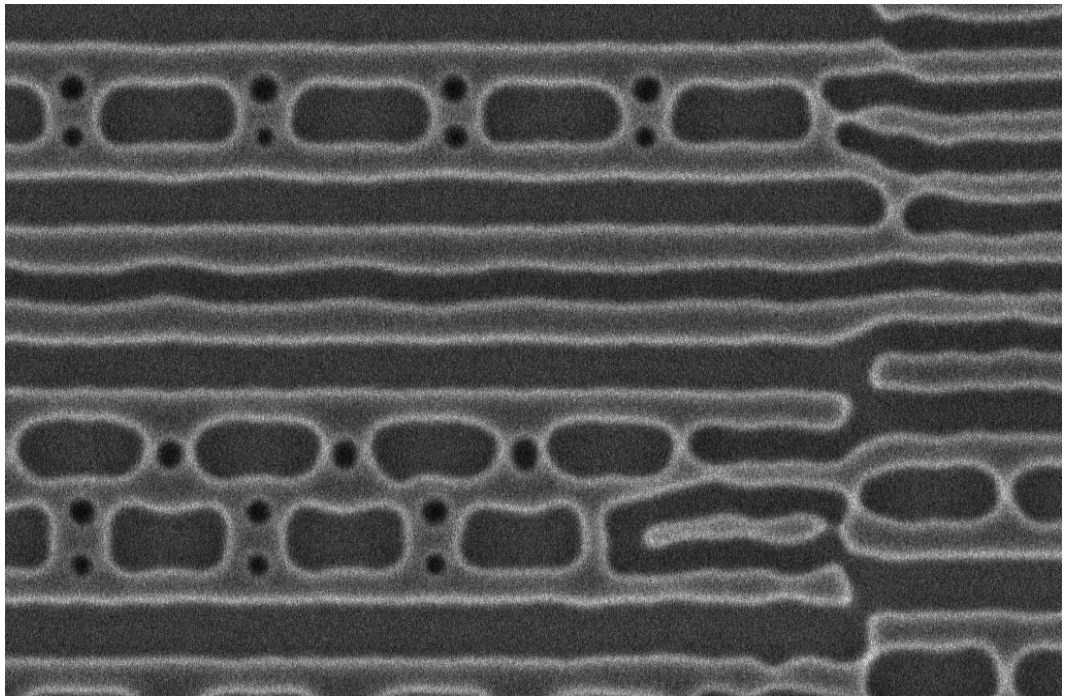
- If data is going to be manipulated multiple times, put it on the GPU with cudaMalloc() and leave it there.
- If you've got a large chunk of CPU data that you want to touch and then leave alone, use cudaHostRegister() to map it into GPU space.
- If you've got an interpolated table of values, use cudaTexture.
- Check performance carefully; GPUs can do the unexpected.
  - Interprocess mutex behavior for some memory transactions.
- Assume that whoever is reading your code doesn't understand the memory modes. Comment appropriately.

CUDA Best Practices: Use the Profiler

- nvvp is a fine tool. Use it when you're ready to profile.
- As with all coding tasks where performance really matters:
  1. Prototype the data path first.
  2. Make it work correctly.
  3. Make it work quickly.
- Early optimization is rarely properly targeted.
- Measure before you optimize.

Now We Understand the Talk Title

- PLDC on NuFlare MBM-1000:
  540 TBytes of Inline MPC in 10 Hours



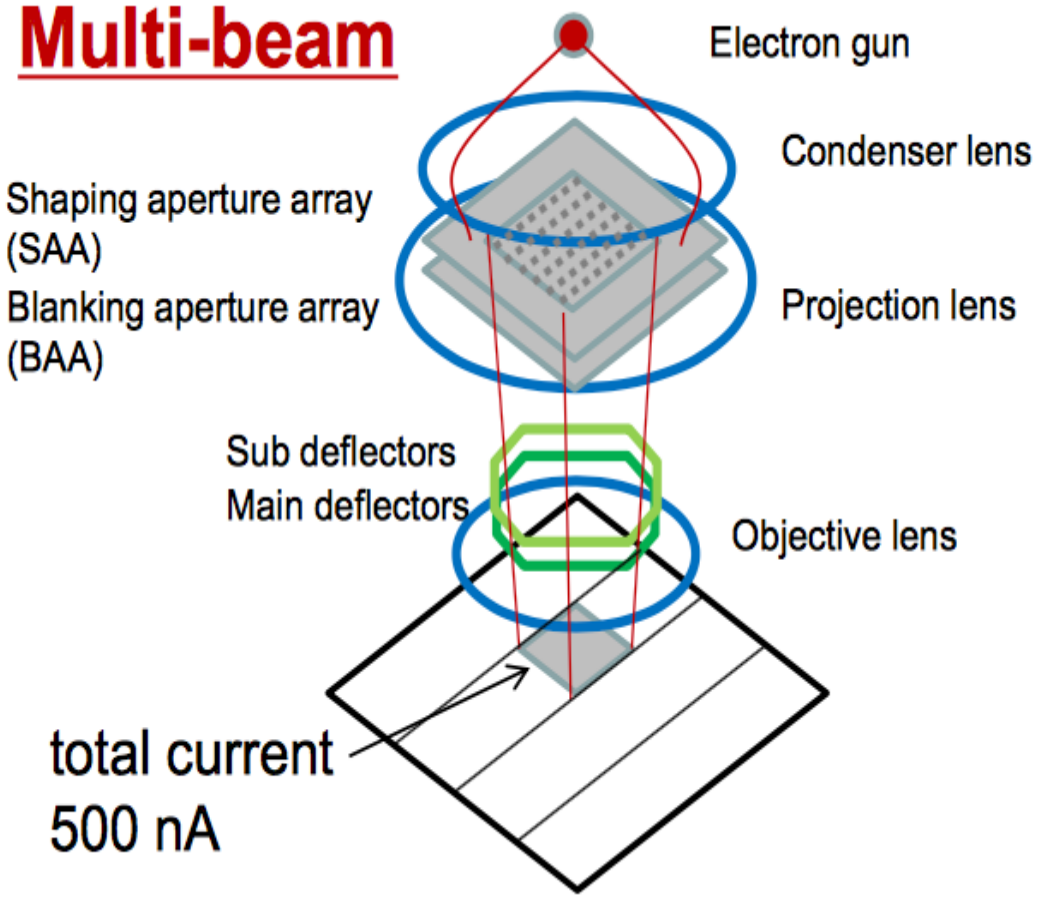


Image from semiengineering.com, 2017-01-08

Use GPUs to Make GPUs

Image credit: nvidia.com

harold@design2silicon.com