Design-aware Mask Inspection

Abde Ali Kagalwalla[1], Puneet Gupta[1],
Chris Progler[2] and Steve McDonald[2]

[1]{abdeali,puneet}@ucla.edu

Electrical Engr., UCLA

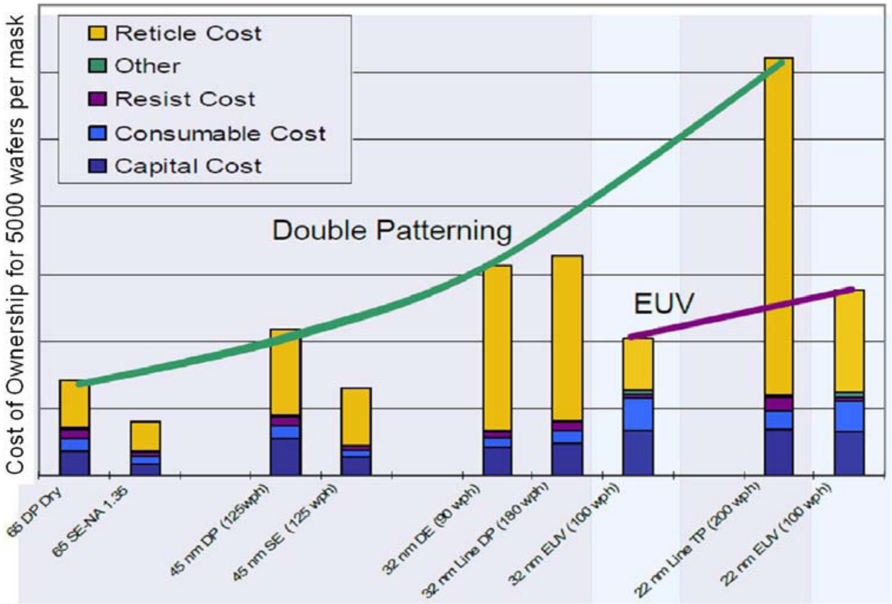
[2]Photronics Inc.

www.nanocad.ee.ucla.edu

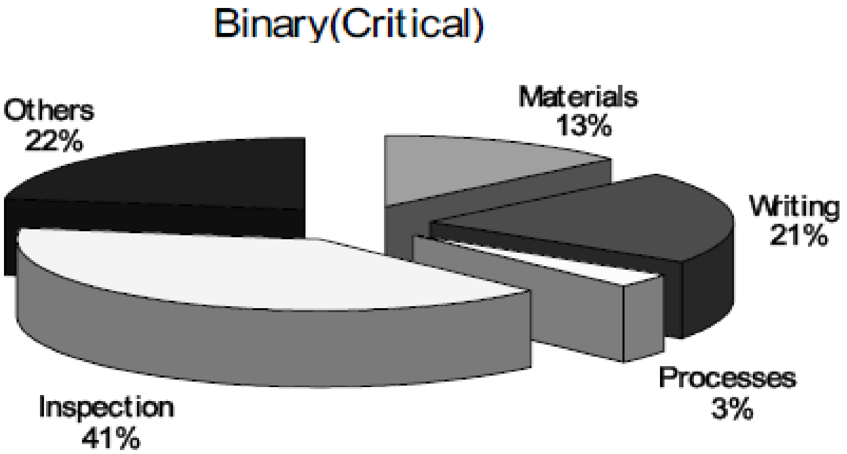
Outline

- **Motivation**
- Proposed design-aware inspection
- Results
- Conclusion & Future Work

Motivation



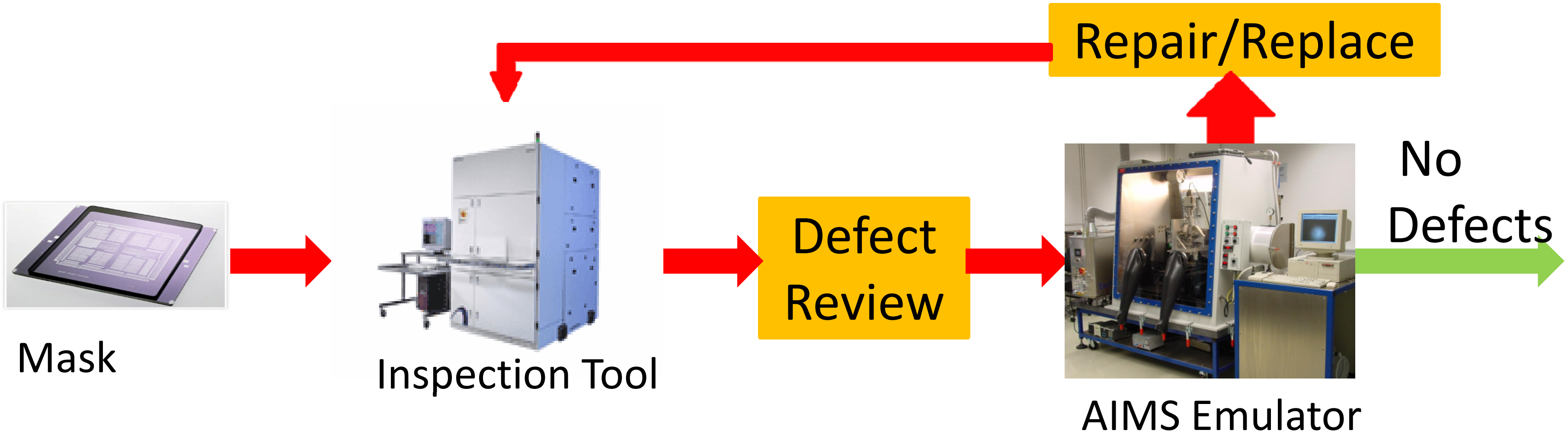
Mask cost increase with technology
Source: ITRS 2009



Mask manufacturing cost budget
Source: Dai Nippon Photomask at SPIE 2008

- Decreasing feature size & RETs → mask inspection challenging
- Reducing mask cost critical for low volume SoCs
- Mask cost expected to be worse for future patterning(EUV, nano-imprint)

Mask Inspection Primer



Repair/Replace

Mask

Inspection Tool

Defect Review

AIMS Emulator

No Defects

- Defect review often manual → Slow
- AIMS emulation ‘gold standard’ but tedious
- Defect repair/replacement expensive

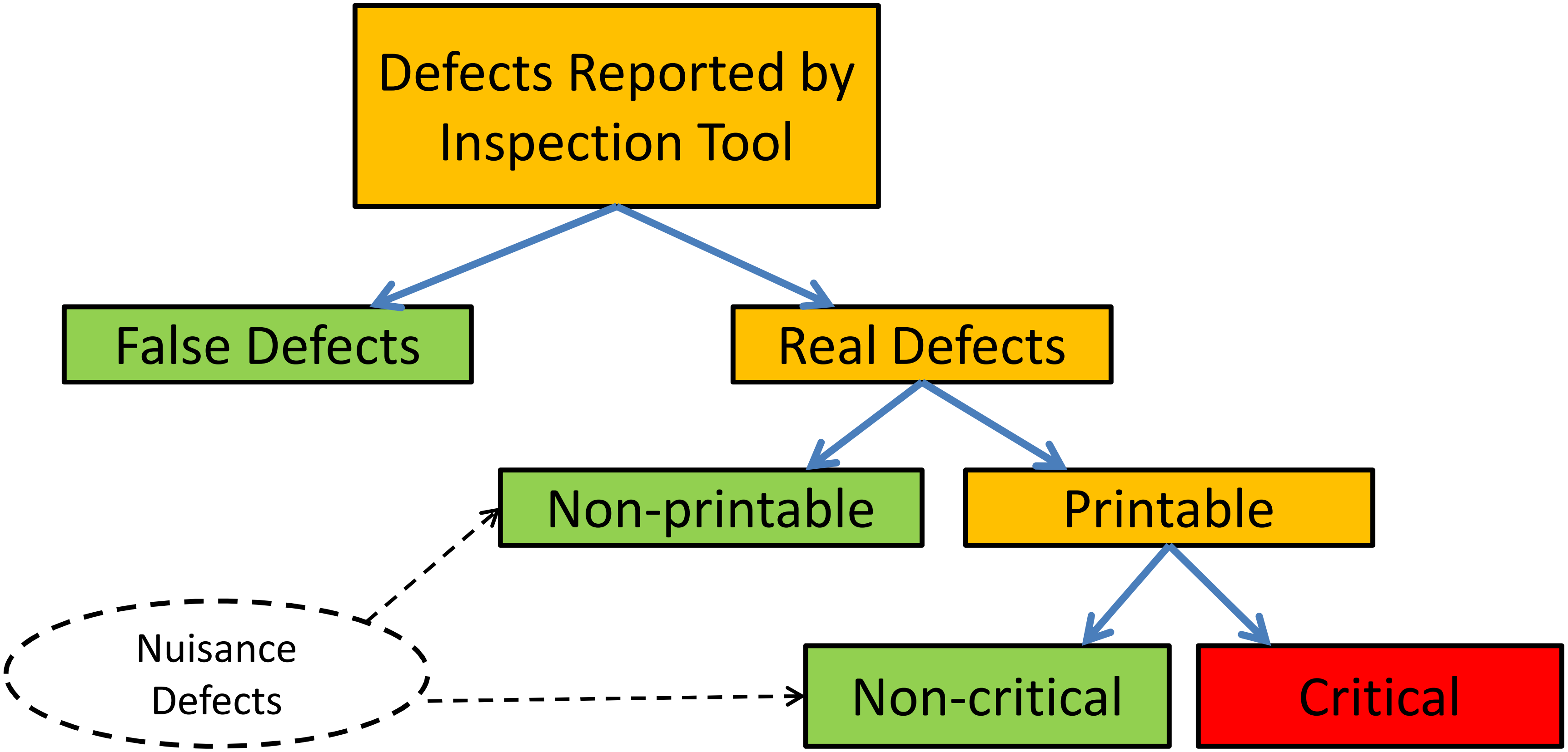
Mask Inspection Tool

- Gray-scale image comparison
- Intensity difference > threshold → Defect

- Allows adjustable threshold
- More common used term is sensitivity(s)
- Can choose from different pixel sizes(p)
- Inspection resolution = K(p/s)

- First pass yield
  - -Masks that pass inspection without repair/replacement
  - -Key metric for cost reduction
- Controlling defect count of tool critical for turnaround time

Why Design-Aware Inspection?



Defects Reported by Inspection Tool

False Defects

Real Defects

Non-printable

Printable

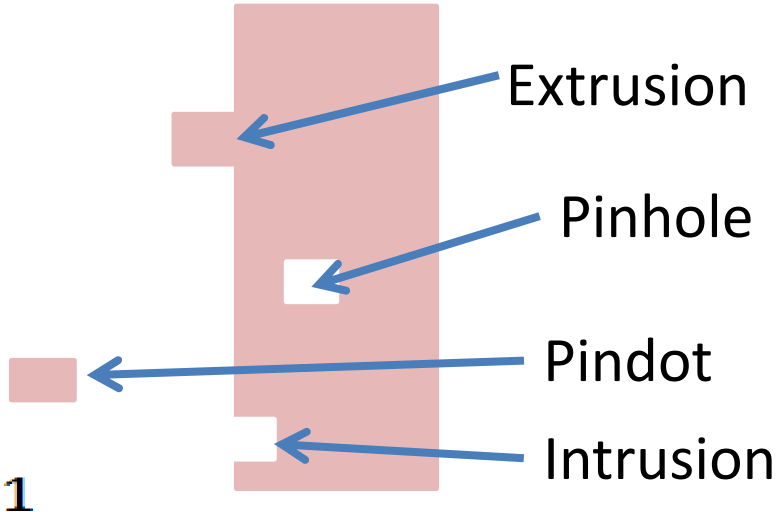
Non-critical

Critical

Nuisance Defects

Design-awareness to minimize false + nuisance defects reported without missing critical defects

Modeling Inspection Tool Defects



Defect Types:

- CD defects: Intrusion, extrusion
- Contamination: Pinhole, pindot

 # False Defects $=K\frac{\mathrm{Area}}{p^{\alpha}}\ \mathrm{erfc}(\frac{1}{\sigma s})$

- Models imaging system noise
- Typically models photon limited noise

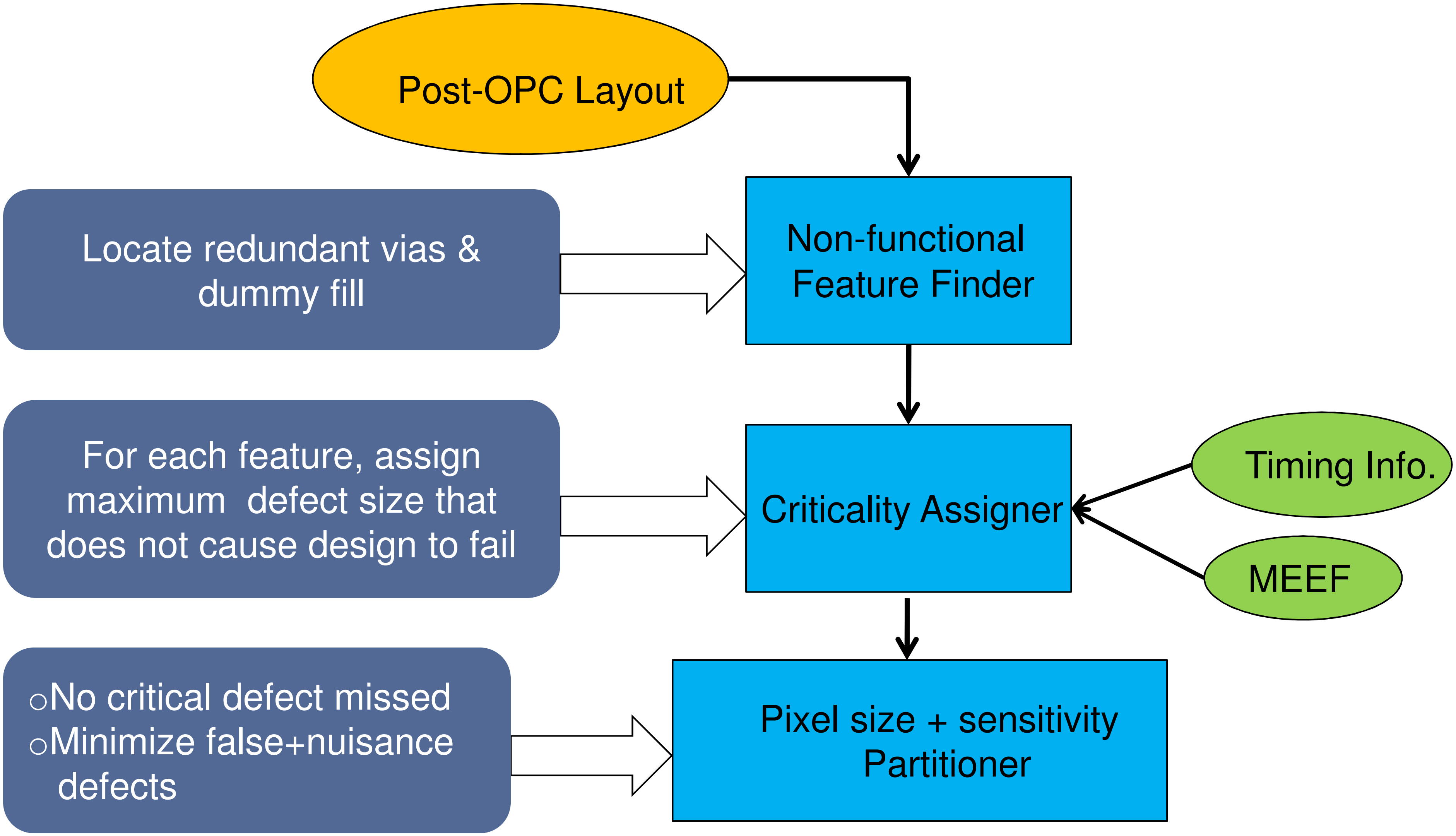
 # Nuisance Defects $=K_n Area\left(\frac{p}{s}\right)^{\beta}$

- Derived assuming negative binomial defect distribution

Outline

- Motivation
- **Proposed design-aware inspection**
- Results
- Conclusion & Future Work

Overview of our work



Post-OPC Layout

Locate redundant vias & dummy fill

Non-functional Feature Finder

For each feature, assign maximum defect size that does not cause design to fail

Criticality Assigner

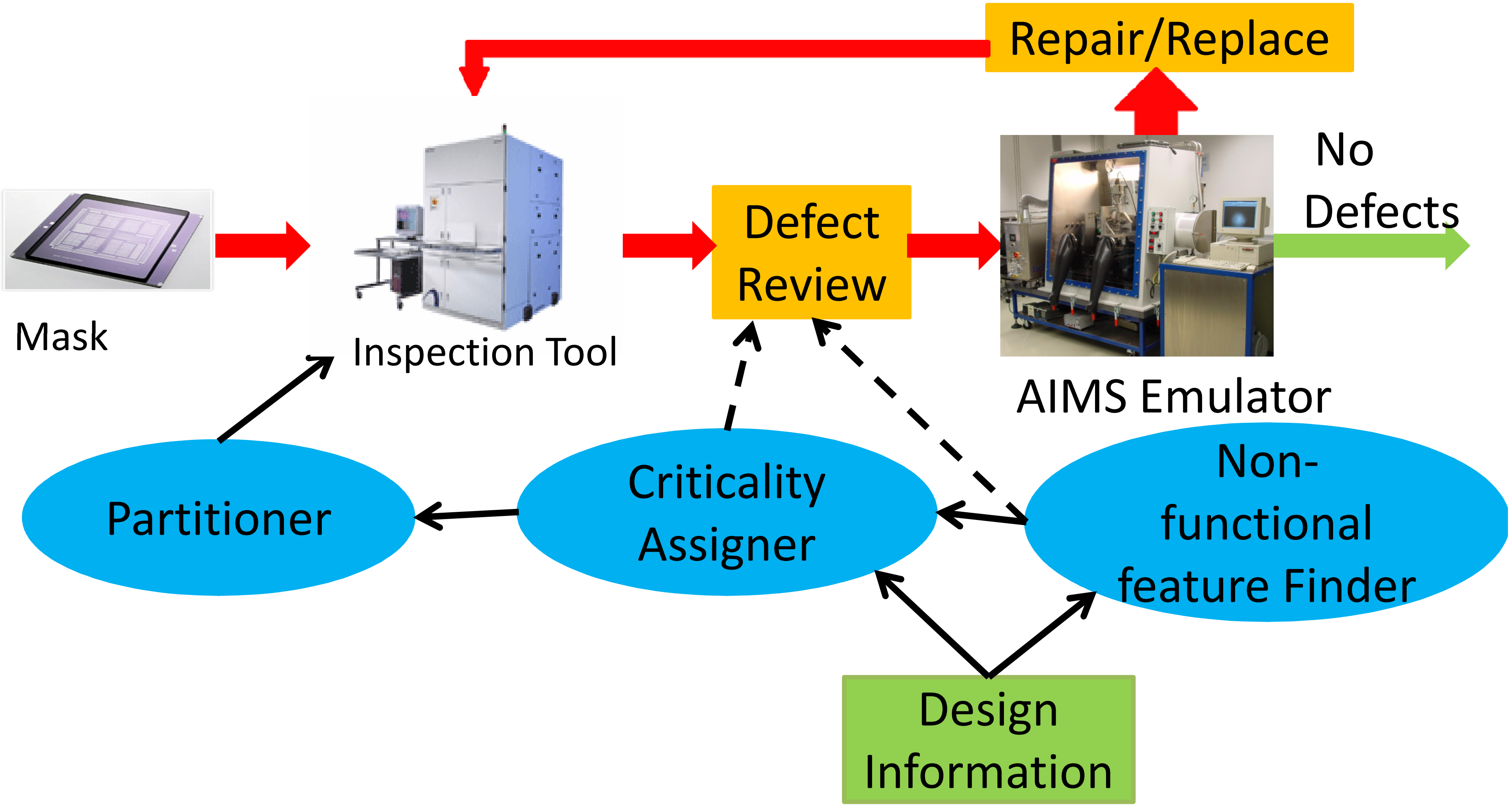
Timing Info.

MEEF

- ○No critical defect missed
- ○Minimize false+nuisance defects

Pixel size + sensitivity Partitioner

Proposed Design-aware Inspection Flow



Repair/Replace

Mask

Inspection Tool

Defect Review

AIMS Emulator

No Defects

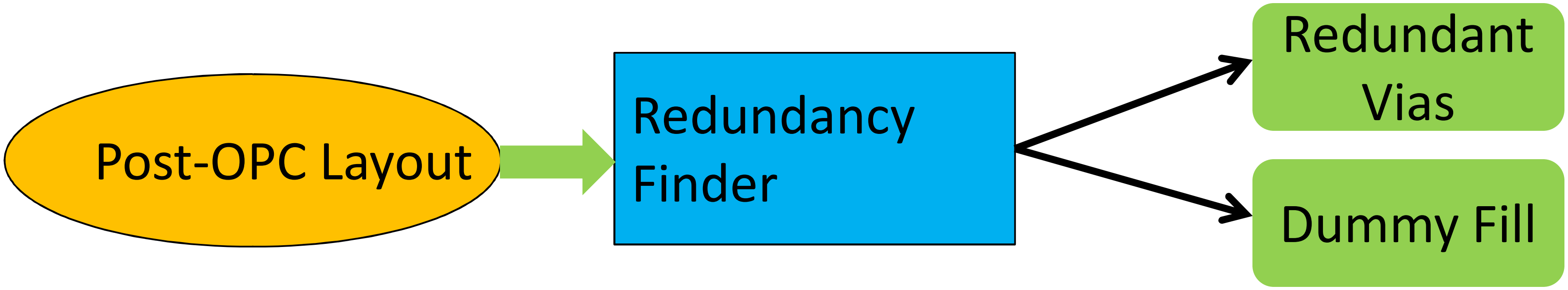
Partitioner

Criticality Assigner

Non-functional feature Finder

Design Information

Non-functional Feature Finder: Overview



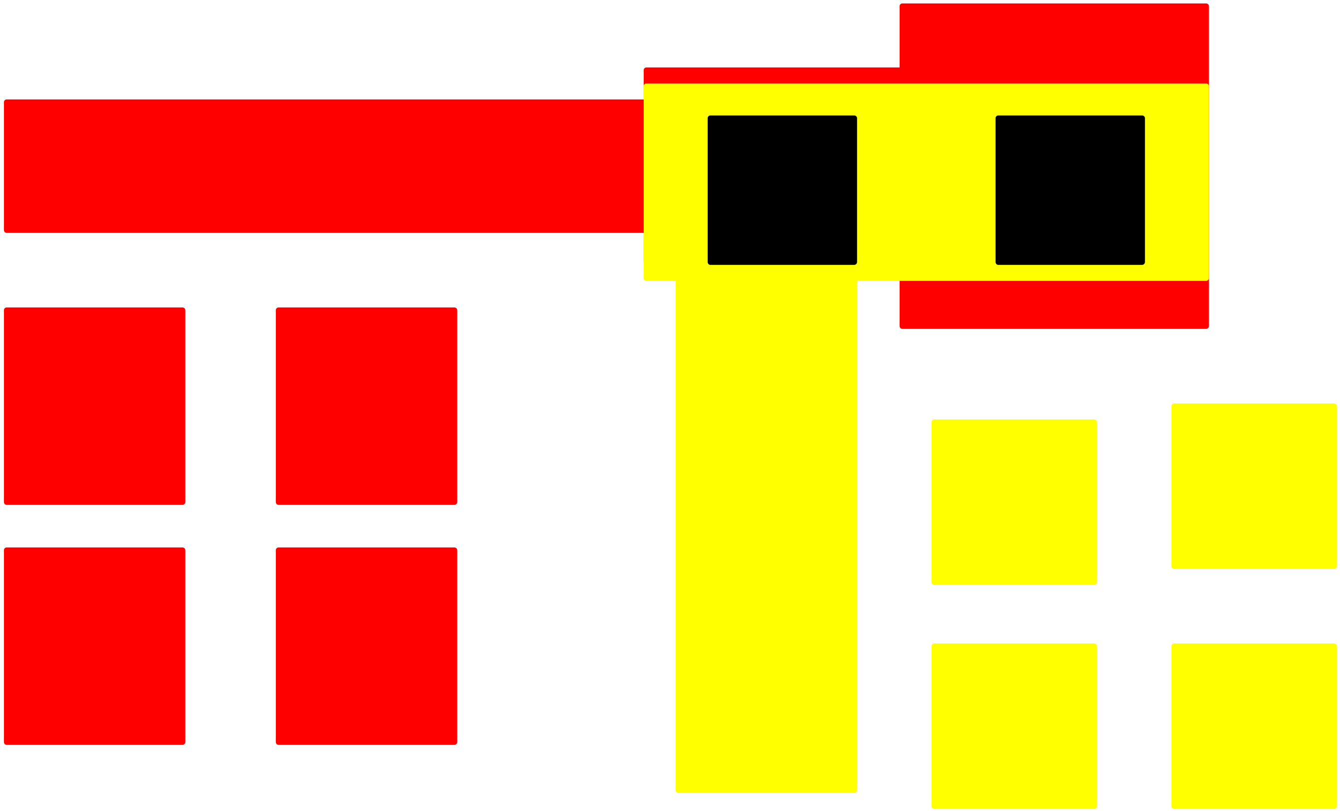
- Assume that layout has only rectilinear shapes
  - Valid for all digital designs

- Only floating fill with no via-connected fill considered
  - Consistent with most fill insertion tools

- Approach extensible to identifying other non-functional features like spare cells, non-tree routes and assists

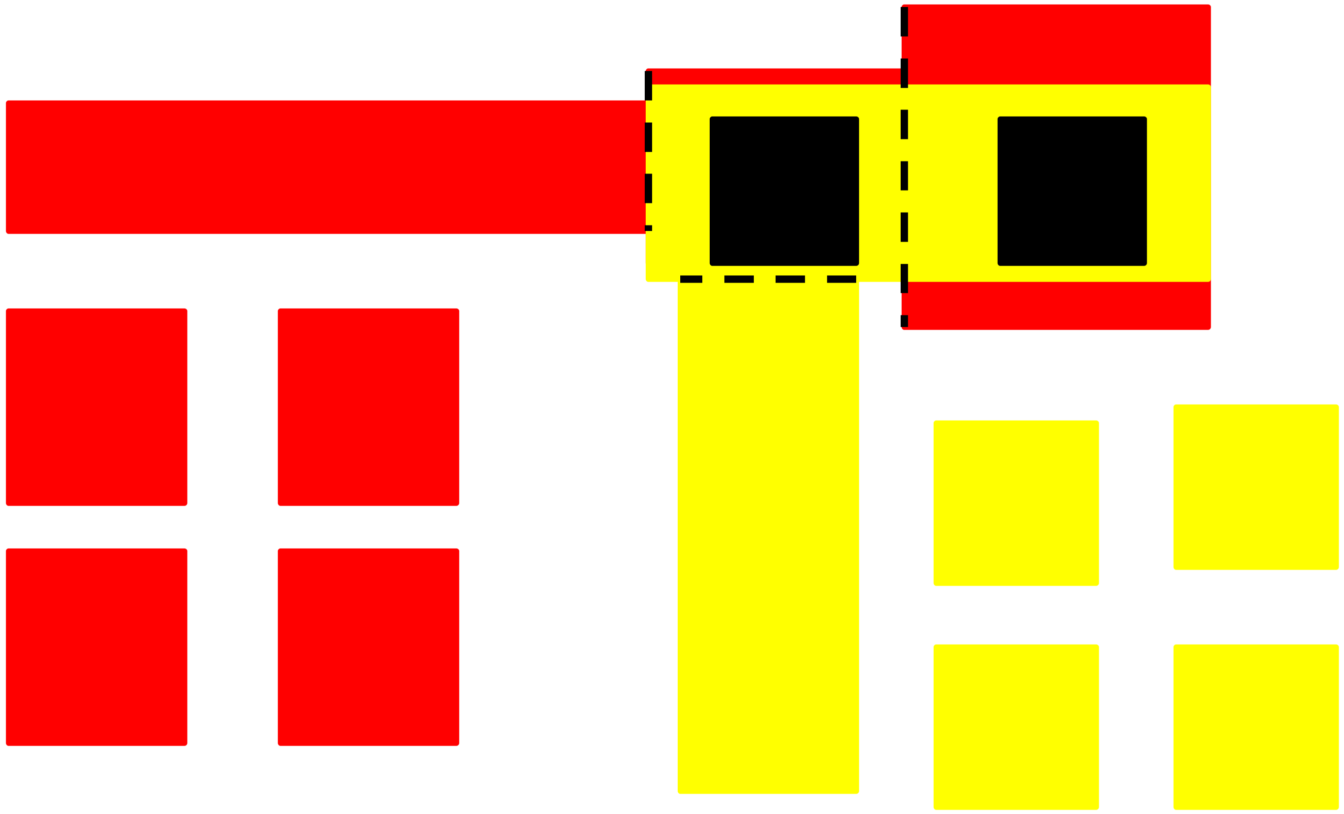
Non-functional Feature Finder: Algorithm Steps

Sample Layout



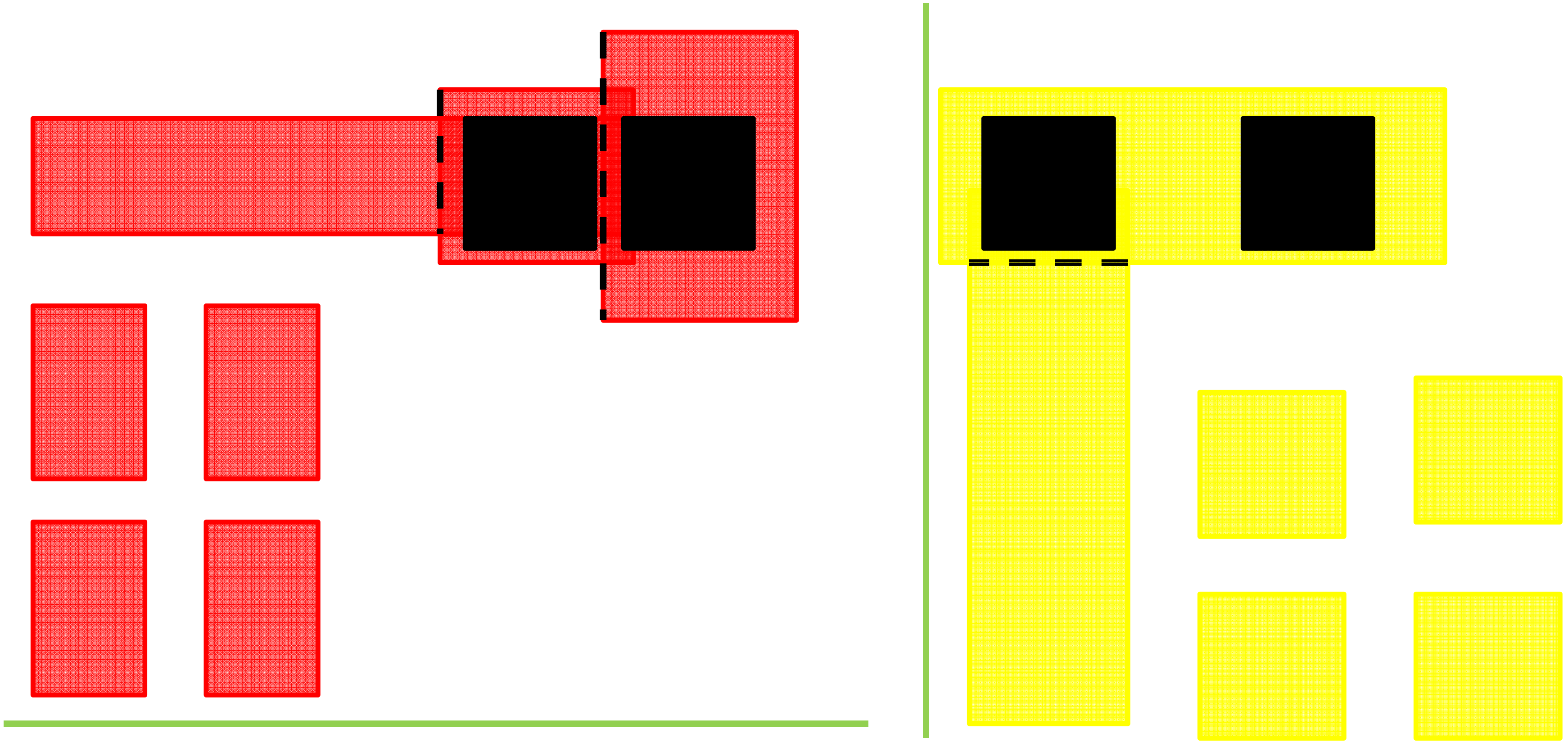
Non-functional Feature Finder: Algorithm Steps

Fracture polygons



Non-functional Feature Finder: Algorithm Steps

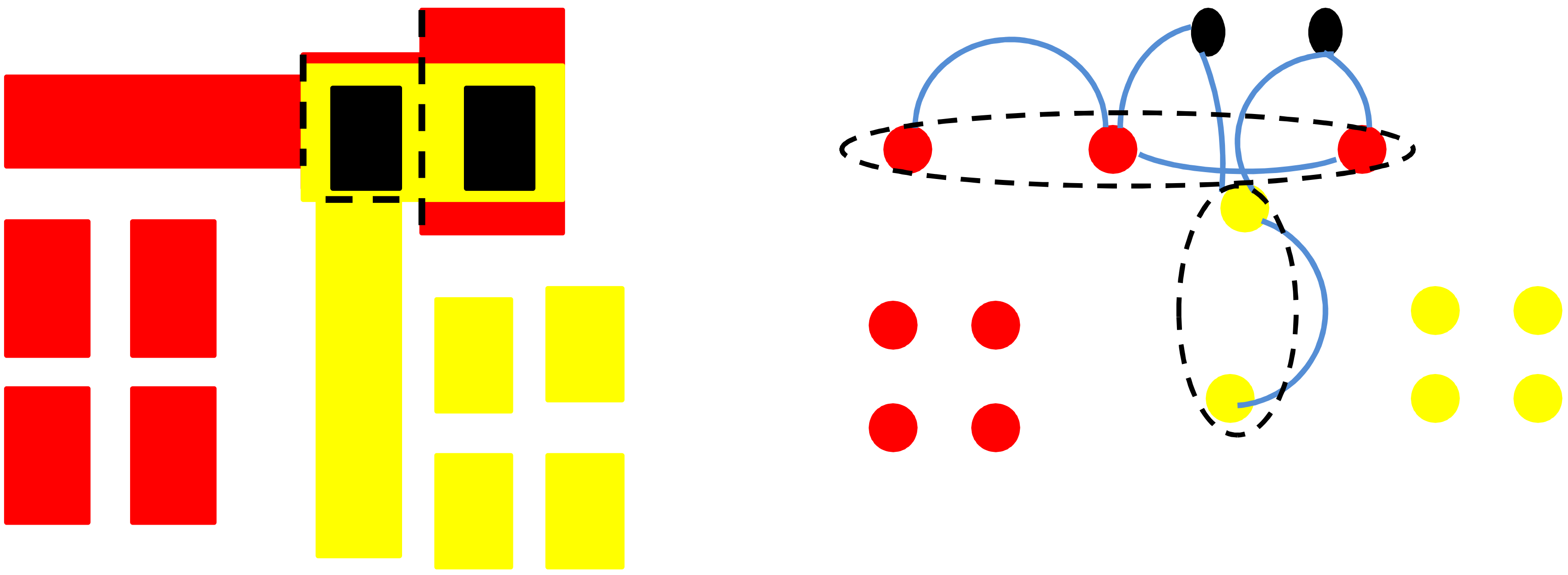
Scan-line for graph construction



Segment + interval trees to store scan-line events

Non-functional Feature Finder: Algorithm Steps

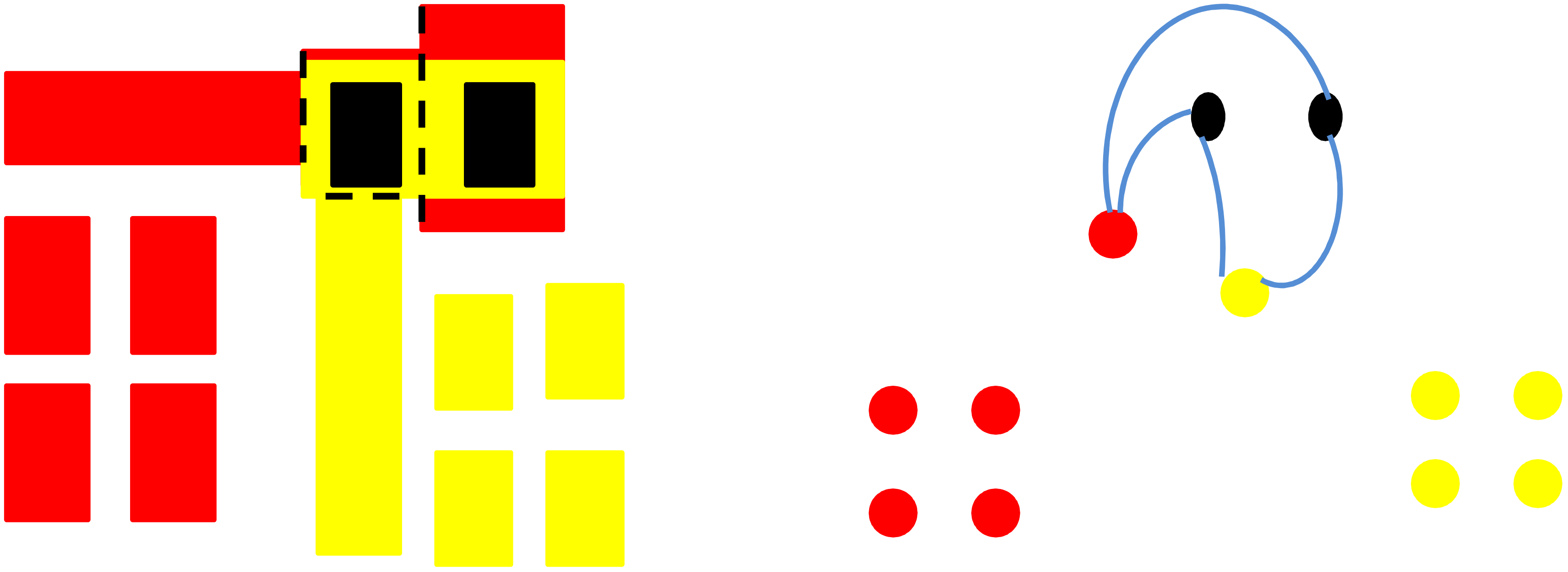
Merge Neighborhood Graph



Same color neighbor vertices merged

Non-functional Feature Finder: Algorithm Steps

Analyze Merged Neighborhood Graph



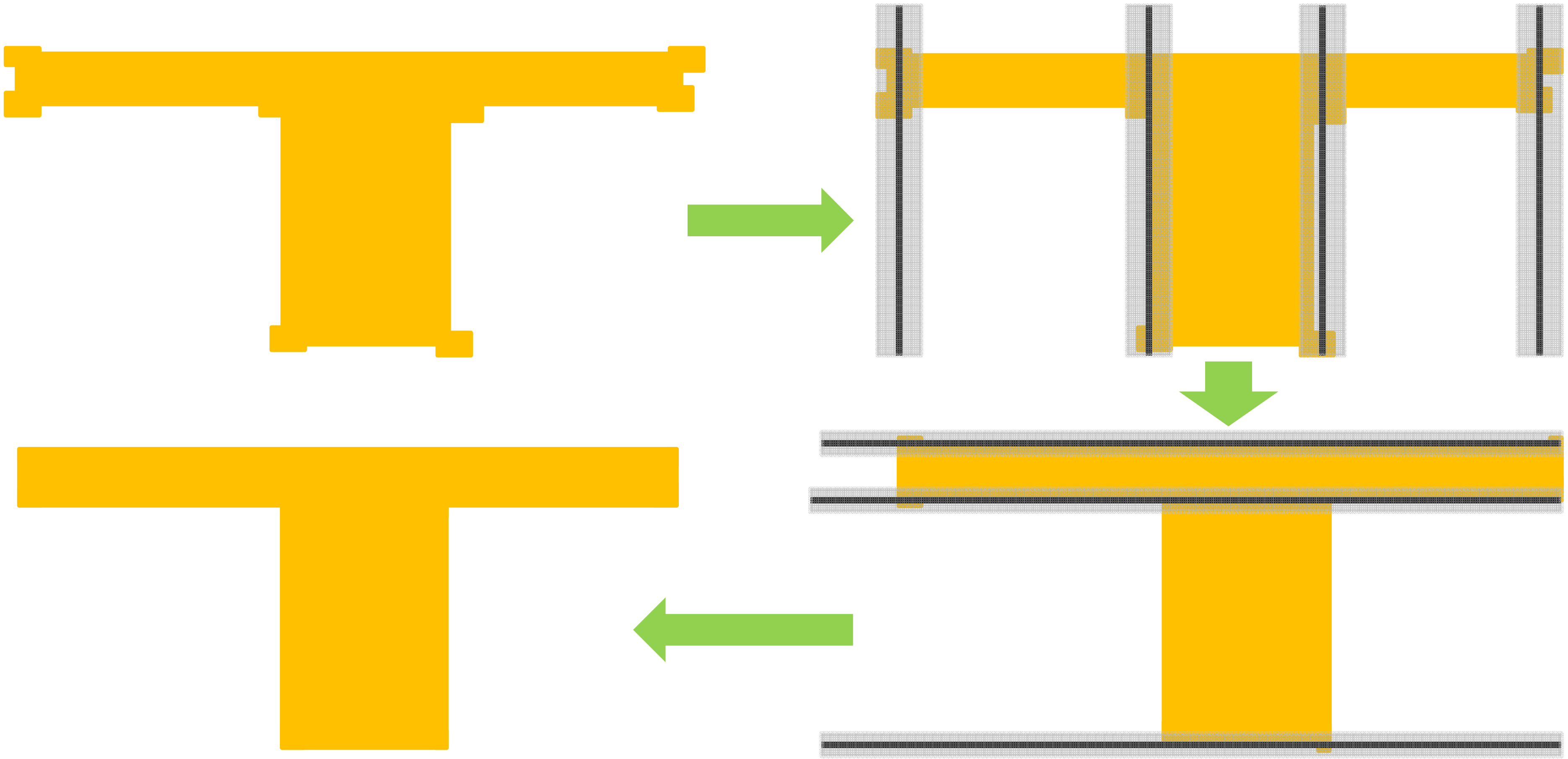
Cycles → Redundant vias
Isolated vertices → Floating fill

Non-functional Feature Finder: Algorithm Summary

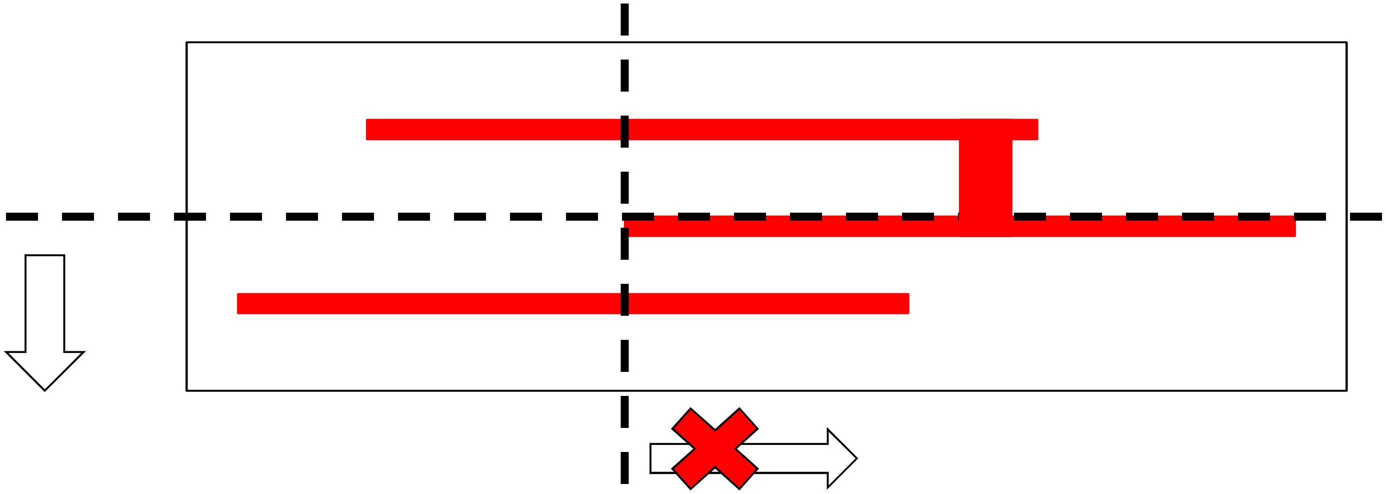
- Algorithm steps:
  - Fracture shapes
  - Neighborhood graph construction
  - Vertex merging
  - Cycle and isolated vertex finding

- **Scan-line based graph construction time critical step**

Runtime Reduction: Shape Simplification



Runtime Reduction: Scan-line speedup

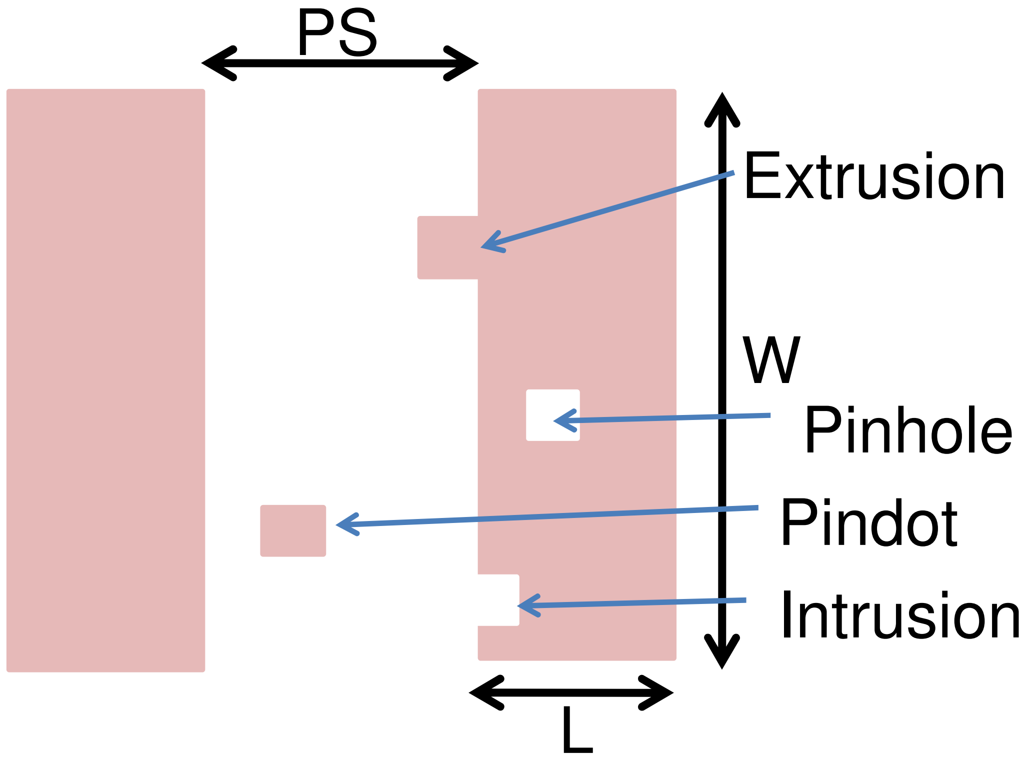


- Estimate routing direction
  - -Reduces average size of segment+interval trees

- Use separate interval+segment trees for each metal+via layer set
  - - Smaller tree size
  - - Easy to parallelize

Poly Layer Assignment

- Timing slack → Max. tolerable defect size
- Assume a fixed finite number K(=10) of defects per path
- Account for width /spacing to prevent opens/shorts



Assumption: Pinholes have no design impact

Metal/Via Layer Assignment

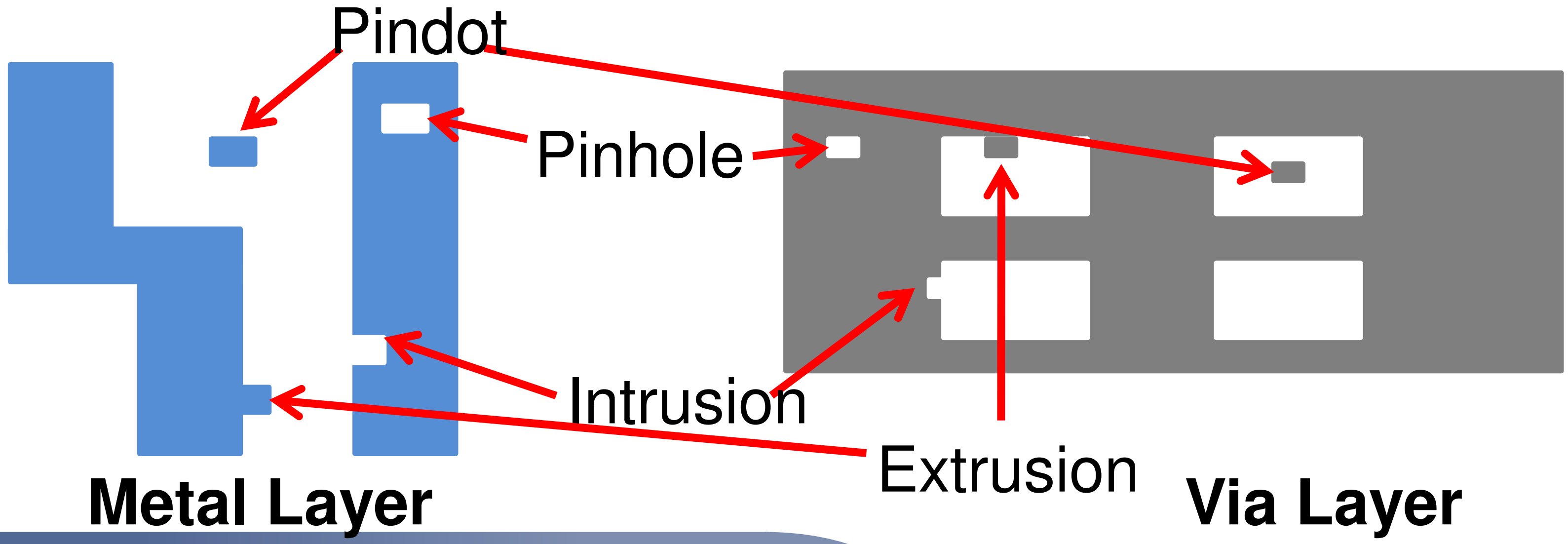
•Require only post-OPC layout for assignment!!

Metal Layer

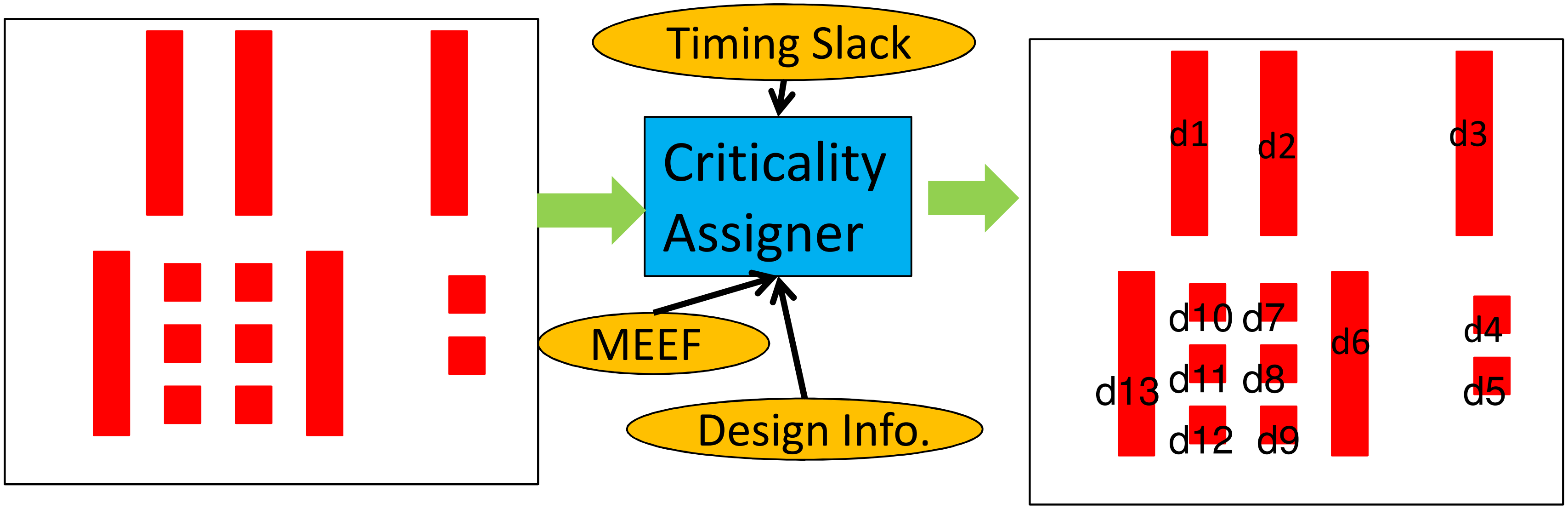
- Dummy features assigned larger minimum defect size

Via Layer

- Even smallest pinhole can cause short
    - Non-dummy metal intersect regions
- Redundant vias assigned higher defect size



Criticality Assignment



Timing Slack

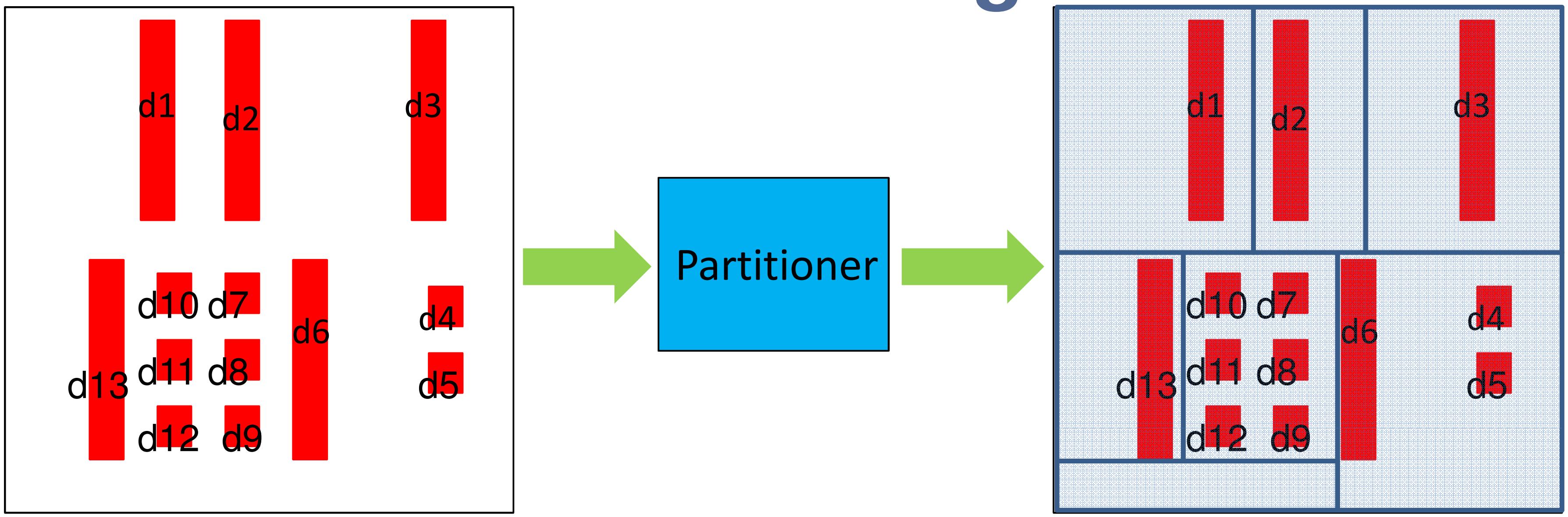
Criticality Assigner

MEEF

Design Info.

- CD (extrusion/intrusion) and contamination (pinhole/pindot) defects separately considered
  - -Inspection tools have different sensitivities for them
- *Assumptions:*
  - - Only binary, square defects considered
  - - MEEF=1 since modern Inspection tools adapt to it

Partitioning



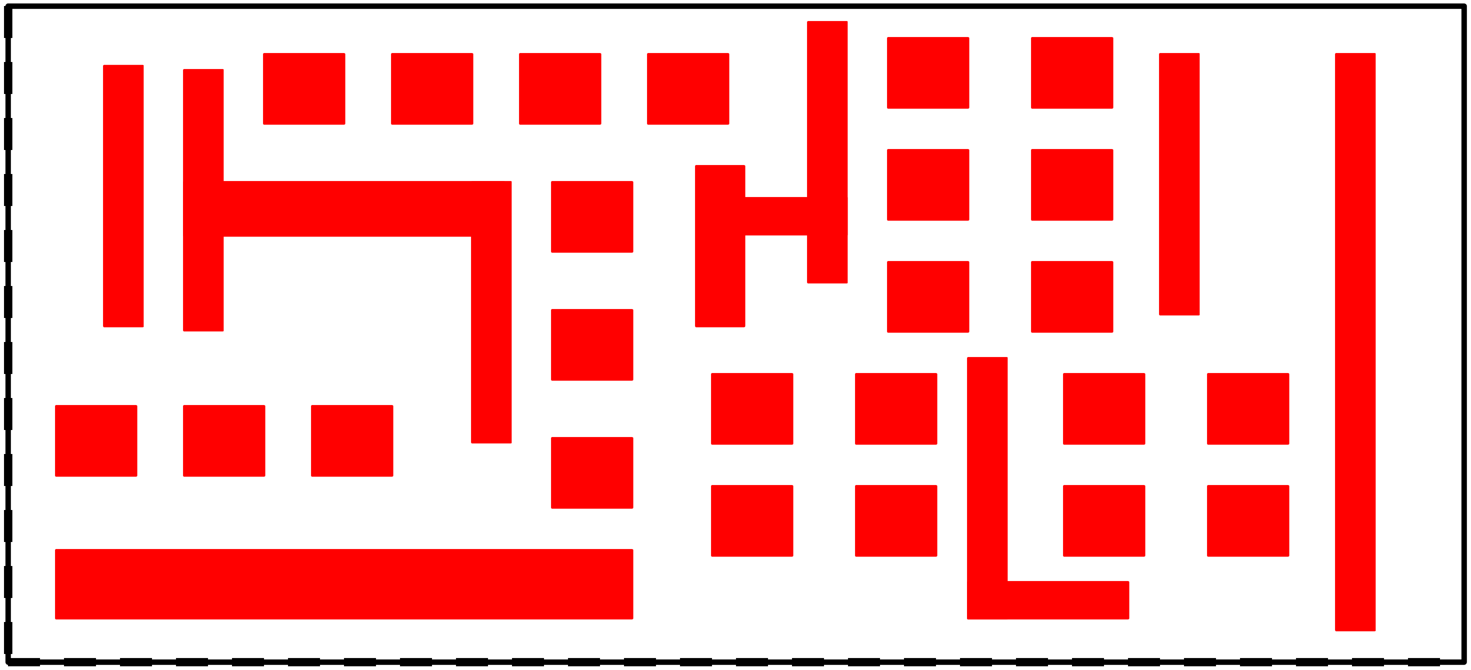
Goal: Partition with each region assigned a pixel size, sensitivity

Constraints:

1. CD tolerance of partition > Min. detectable defect size = K(p/s)
   - Ensuring no critical defects missed
2. Min. width/height of each partition > L_{min}
   - Inspection tool requirement

Cost Function: #False Defects + γ*#Real Defects

Partitioning Algorithm



- Scan-line based heuristic
  - Move vertical and horizontal lines across design
  - Max. tolerable defect of partition(p/s) → try all discrete p values and pick minimum cost value
  - Moving distance of L_{min} to meet width constraint

Outline

- Motivation
- Proposed design-aware inspection
- **Results**
- Conclusion & Future Work

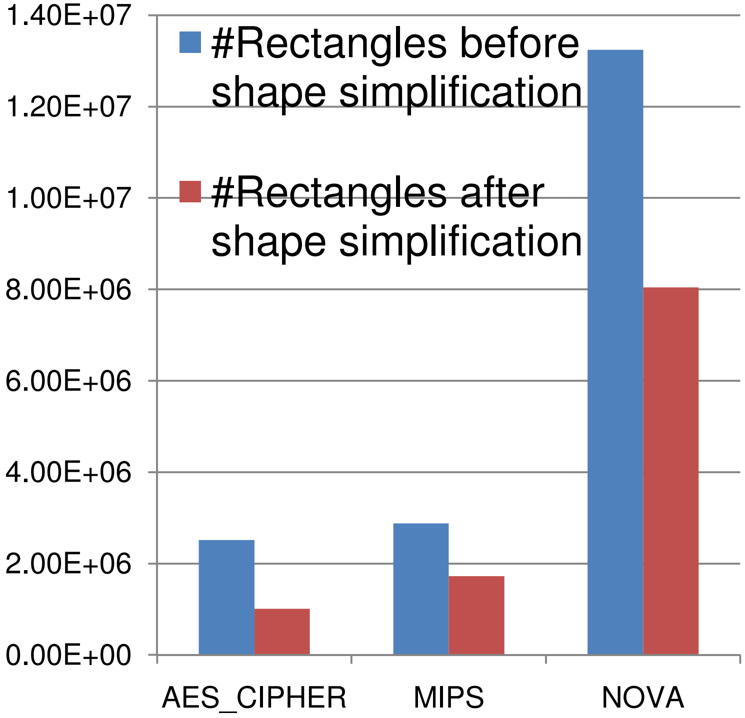
Experimental Setup

- All implementation done in C++ using OpenAccess API
- Test cases taken from opencores.org
    - SP&R → Cadence RC/Encounter + 45nm Nangate
    - OPC → Mentor Calibre
    - DRs → 45nm Free PDK
- Defect models fitted using commercial maskshop data
    - 800 reticles, 8000-15000mm^2
- Pixel sizes: 72nm and 90nm, Sensitivity:0-100
- L_{min} = 2.0um (wafer scale)

Design Name	**# Gates**	**Area (um^2)**
Aes_cipher(8-metal)	15467	102494
Mips(6-metal)	11577	59461
Nova(6-metal)	43156	268594

Experimental Results: Non-functional Feature

- Results verified using DEF file of designs
  - Almost 100% accuracy for both dummy fill and redundant vias

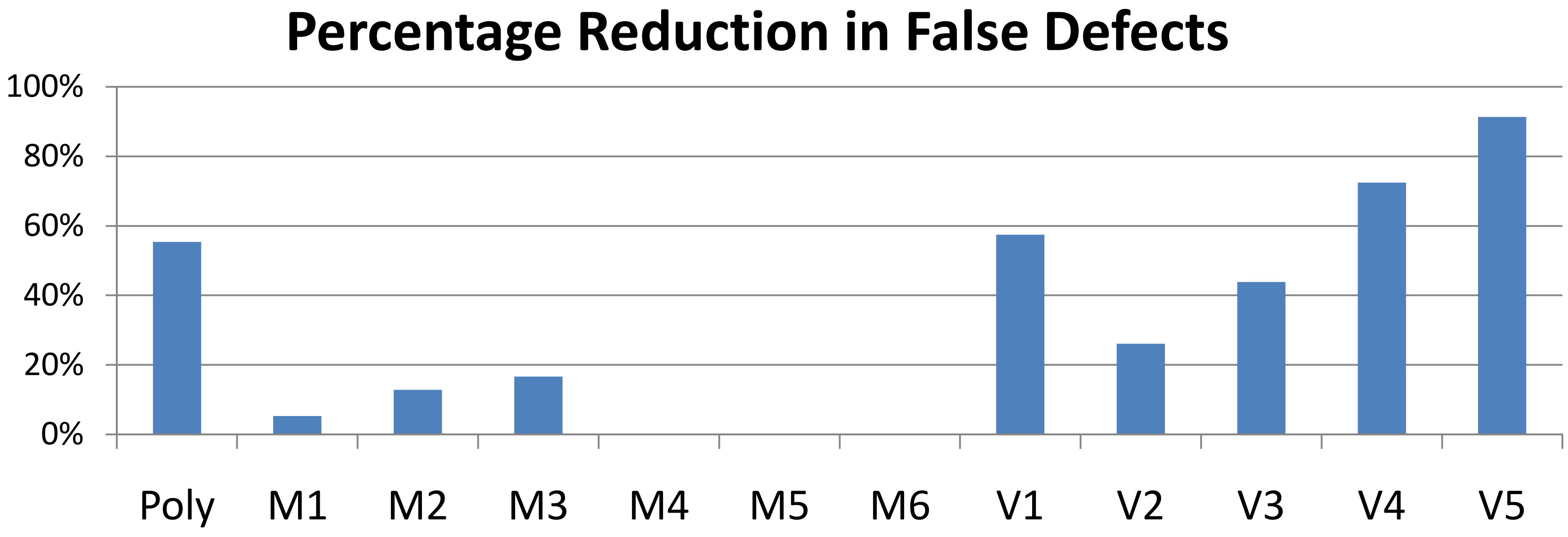


Design	#Double Vias	#Dummy Fill	Runtime (min)	Memory (MB)
Aes_ Cipher	131464	97772	8	910
Mips	44004	67341	5	1190
Nova	209623	303792	79	4814

Experimental Results: Partitioning

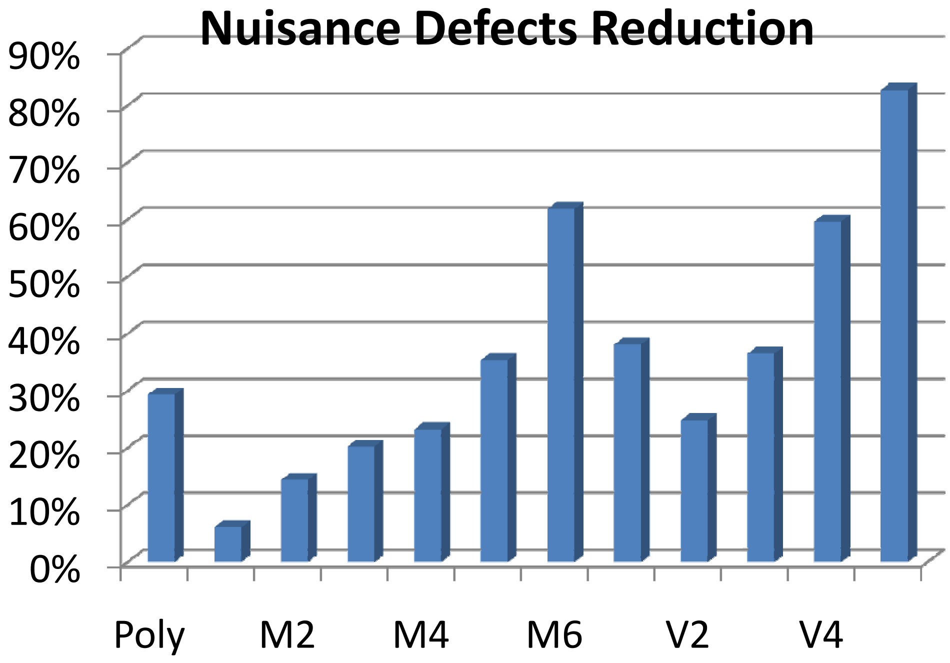
- Average false defect reduction over two designs (MIPS and NOVA)
  - Via layer: Most improvement → redundant vias
  - Higher metal layers: Zero improvement → Less defects

- Substantial improvement in defect review time

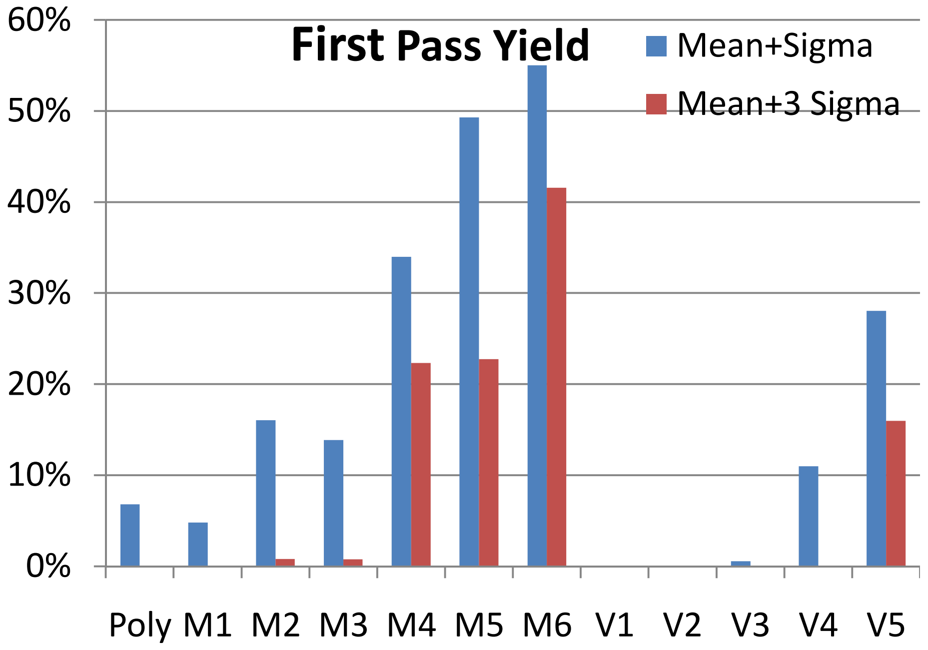

Percentage Reduction in False Defects

Experimental Results: Nuisance defect reduction

- Higher via, metal layers show substantial nuisance defect improvement
- For first pass yield, Monte Carlo simulation with 7-150nm defects distributed on the partitioned reticle area



Nuisance Defects Reduction



First Pass Yield

Outline

- Motivation
- Proposed design-aware inspection
- Results
- **Conclusion & Future Work**

Conclusion

- Proposed a comprehensive design-aware mask inspection methodology:
  1. Identified non-critical features with full accuracy in post-OPC layout
  2. Method for evaluating criticality of shapes using timing slack, non-critical info and design rules
  3. Partitioning algorithm to inspect different regions with different pixel size and sensitivity

- Up to 4X reduction in false defects with up to 55% improvement in first pass yield achieved by design-aware inspection

Future Work

- Current approach assumes mask shop has complete mask set of design
  - Techniques to work with limited design data
- Better false defect model
- Study tradeoffs of tuning only sensitivity versus sensitivity + pixel size

THANK YOU!