Rapid In-Line Detection of Macro Defects in Semiconductor Manufacturing

Dick de Roover, Gwen W. van der Linden, Sarbajit Ghosal, Robert L. Kosut, Jon L. Ebert, Abbas Emami-Naeini

SC Solutions, Inc.
1261 Oakmead Pkwy
Sunnyvale, CA 94085
Email: roover@scsolutions.com

APC Conference XXVI 2014
Ann Arbor, MI
10/01/2014
Content

- Background & Motivation
- Macro Defect Detection Problem
- The Idea
- Development of the Hardware/Software Solution:
  - Concept feasibility
    - Use of a commercial scanner & results
    - Development of required algorithms
    - Sample die estimation & defect detection results on process wafers
  - Custom hardware design & performance
    - Custom hardware design
    - Sample results on process wafers
  - Defect clustering
    - Clustering methods
    - K-means algorithm
    - Sample results
- Summary & Conclusions
The manufacturing of Integrated Circuits (ICs) on semiconductor wafers involves hundreds of complex and expensive process steps.

Defects can occur during any of several steps such as etch, resist removal, and can be caused by particle contamination, incomplete process, process variations, dislocations, scratches, cracks, etc.

Defects range in size from submicron to visually-detectable “macro” defects that may be as large as several inches long and span multiple dies.

Defects reduce yield.

With increase in number of steps in semiconductor manufacturing, defect detection becomes more critical, however, current Defect Detection tools are slow and expensive, and not every wafer is inspected.

There is a critical need to quickly and inexpensively inspect each and every wafer without affecting throughput.
Macro Defect Detection Problem

Conventional way: off-line detection of only a sample of wafers, expensive equipment (> $M)

Needed Solution: in-line detection of every wafer with relatively inexpensive equipment (< $50k)
The Idea

Detect large fraction of macro defects right at the process step by in-situ auto-inspection of every wafer using:

1. inexpensive scanning hardware
2. sophisticated algorithms
Details of the Idea: Defect Detection & Classification

Rapid In-line Macro-Defect Detection

Image from Inexpensive Hardware

Distortion Correction

Defect Detection

Defect Classification

Wafer scanner

Detector

mirrors

light source

glass plate

moving wafer

or

High-speed camera

or

patterned/unpatterned wafers

Optics, alignment, aliasing, ...

Root-Cause Analysis (RCA)

Algorithms

Network Interfaces

User Interfaces

GPUs
Concept Feasibility: Use of Commercial Scanners
Our First Scan of a Wafer Using a Commercial Scanner

Scanned using HP Scanjet G3010

Red circle is wafer edge.

Yellow “circle” is 3mm exclusion

Red dot is wafer center

Blue dot is notch
Resolution determines maximum detectable defect size

Scanning time increases rapidly with increasing resolution

<table>
<thead>
<tr>
<th>Resolution (DPI)</th>
<th>Pixel Size (µm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>300</td>
<td>84</td>
</tr>
<tr>
<td>600</td>
<td>42</td>
</tr>
<tr>
<td>800</td>
<td>32</td>
</tr>
<tr>
<td>1200</td>
<td>21</td>
</tr>
</tbody>
</table>
Development of Required Algorithms

- **No a priori information** regarding the wafer, die-size, edge, etc. is required

- **Algorithms to detect wafer orientation**
  - Detect edge of wafer
  - Find center of wafer
  - Find the notch
  - Rotate wafer to align image with center-notch

- **Algorithms to detect die size & locations**

- **Algorithms to detect defects**
  - Estimate reference die (no defects)
  - Use statistics to compare dies to reference die (zero false-positives)

- **Algorithms to classify defects**
  - Identify clusters
  - Pattern recognition
  - Root-cause analysis
Wafer Orientation & Die Detection

Scanned wafer – scaled, rotated, aligned to notch

Notch detection

Die Location detection

Die Size detection
Resolution impacts quality of ‘reference die’

Algorithms needed to compensate for image distortion such as ‘aliasing’, line curvature, etc.

Algorithms are time consuming
Sample Results: Die Estimation & Defect Detection

- **300 dpi**
  - 155 bad dies marked

- **600 dpi**
  - 162 bad dies marked

- **800 dpi**
  - 158 bad dies marked

- Aim for 0 ‘false positives’
- Use reference wafer (no defects) to determine thresholds
Custom Hardware Design & Performance
Having demonstrated concept feasibility with commercially available scanners, we can speed up the process and achieve better performance using custom hardware/software design.

To not affect throughput, scanning needs to occur in the order of seconds, while wafer is moving in or out of a tool, subsequent processing < 30sec.

**DEVELOPMENT TARGET**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Max. Wafer Diameter</td>
<td>300mm</td>
</tr>
<tr>
<td>Max. Carrier Size</td>
<td>450mm</td>
</tr>
<tr>
<td>Max. Optical Resolution</td>
<td>1365dpi</td>
</tr>
<tr>
<td>Min. Image Pixel</td>
<td>18.6μm</td>
</tr>
<tr>
<td>Min. Pixel Depth</td>
<td>8bit X 1</td>
</tr>
<tr>
<td>Max. Scan Time</td>
<td>4sec</td>
</tr>
<tr>
<td>Max. Process Time</td>
<td>25sec</td>
</tr>
<tr>
<td>False Positive Defect</td>
<td>0</td>
</tr>
</tbody>
</table>

Target: hardware that detects defects as small as 10μm at scanning speed in the order of seconds, but still at a favorable price compared to off-line inspection equipment (<$50k)
### Custom Hardware Design

<table>
<thead>
<tr>
<th>Diagram Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scanner cross-section view, with top and front cover. Wafer tray fully retracted.</td>
<td>Same view, but with top and front covers removed, showing the wafer tray.</td>
</tr>
<tr>
<td>450mm</td>
<td>300mm</td>
</tr>
<tr>
<td>Scanner cross-section view with covers removed and wafer tray fully extended.</td>
<td>Same view, with inner cover removed, showing the linear stage.</td>
</tr>
</tbody>
</table>
System Components and Performance for 300mm wafer

- **System components:**
  - High-resolution line camera (16000 pixels per line ~ pixel size of 18.7μm for 300mm wafer)
  - Light sources, Lenses, Mirrors
  - High-speed linear motor
  - Software

- **Achieved performance with custom hardware:**
  - A 300mm wafer was scanned at 1280dpi in 3.5 seconds
  - Subsequent processing for defect detection in 8.5 seconds
  - Defects as small as 5.6μm were detected
Hardware Setup
Sample Result: 300 mm process wafer – raw image
Sample Result: edge, notch & dies detected
Sample Result: die error map
Defect Clustering
Two classes of clustering techniques:

1. **Hierarchical or agglomerative**: These algorithms start with each point in its own cluster. Clusters are combined based on their proximity to each other, using one of many possible definitions of closeness.

2. **Point assignment**: Points are considered in some specified order, and each point is assigned to the cluster into which it best fits.

We have used the k-means algorithm, a widely used point assignment method for classifying...
K-means algorithm

Pseudo-code* of k-mean clustering shown below:

Initially choose \( k \) points that are likely to be in different clusters;
Make these points the centroids of their clusters;
FOR each remaining point \( p \) DO
  find the centroid to which \( p \) is closest;
  Add \( p \) to the cluster of that centroid;
  Adjust the centroid of that cluster to account for \( p \);
END;

- Iterative loop was added to above algorithm:
  - In each iterative step, starting cluster centers were assigned value of the final center of previous iteration until convergence was reached (i.e., no change in cluster composition).

Results with k-means Algorithm

Sample defect data set

Data in four clusters with best-fit ellipses determined for each cluster for quantitative characterization
Iterative determination of clusters using the k-means algorithm:

- Defect within blue square is starting cluster center (centroid)
- Defect shown in purple star is the new cluster center
- Clustering converges in three iterations
Advantages of using the k-means algorithm for defect clustering:

1. Simple implementation
2. Fast for low-dimensional data like defect data

Disadvantages of the k-means algorithm for defect analysis:

1. Since k-Means is restricted to data which has some sort of a center (centroid), it cannot handle data of spatially-varying densities such as annuli
2. Total number of clusters must be specified
3. Does not identify outliers
Implementation of Defect Cluster Map
Summary & Conclusions

- Evaluated different hardware solutions to build an inexpensive industrial-grade scanner that detects defects in the order of 10 to 20 μm
- High-resolution images can be scanned in a few seconds; subsequent processing is in the order of 10’s of seconds
- High-resolution imaging (1365dpi) allows for defect detection in the 10 to 20 μm range
- Effective defect clustering using k-means algorithm
- Addresses defect detection and clustering for both patterned and un-patterned wafers

Is it possible to quickly and inexpensively detect macro defects at every step on every wafer without affecting throughput?

We think yes!

The next question is: is the industry ready for it?