INFORMATION TO USERS

The most advanced technology has been used to photograph and reproduce this manuscript from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.
Defect detection in periodic VLSI circuits using digital image processing

Malhis, Luai Mohammed, M.S.
The University of Arizona, 1990
DEFECT DETECTION IN PERIODIC VLSI CIRCUITS

USING DIGITAL IMAGE PROCESSING

by

Luai Mohammed Malhis

A Thesis Submitted to the Faculty of the
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

In Partial Fulfillment of the Requirements
For the Degree of

MASTER OF SCIENCE
WITH A MAJOR IN ELECTRICAL ENGINEERING

In the Graduate College
THE UNIVERSITY OF ARIZONA

1990
STATEMENT BY AUTHOR

This Thesis has been submitted in partial fulfillment of requirements for an advanced degree at The University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department or the Dean of the Graduate College when in his or her judgment the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED: [Signature]

APPROVAL BY THESIS DIRECTOR

This thesis has been approved on the date shown below:

[Signature]  April 27, 1990

Robin N. Strickland  Date
Associate Professor of Electrical
and Computer Engineering
ACKNOWLEDGMENTS

I am indebted to my advisor Dr. Robin N. Strickland whose encouragement, guidance, and assistance made this work possible. I am also grateful to Dr. Robert A. Schowengerdt and Dr. Harold Parks for their support and suggestions. I would also like to thank Mr. Alexander kran and Mr. Carl Beuschel of the IBM General Technology Division for supporting this project. I am very thankful to my friends Mr. Zuhua Mao and Mr. Timothy Cowell for their valuable suggestions.
TABLE OF CONTENTS

LIST OF ILLUSTRATIONS ................................................. 6
LIST OF TABLES .......................................................... 8
ABSTRACT ............................................................................. 10
1. INTRODUCTION ................................................................. 11

2. PUBLISHED METHODS FOR DETECTING DEFECTS
   IN PERIODIC PATTERNS ................................................... 16
   2.1 Optical Spatial Filtering ........................................... 16
   2.2 Design-Rule Checking Approach ................................. 17
   2.3 Reference Comparison Approach ................................. 18
      2.3.1 The Self Reference Image ................................... 18
      2.3.2 The Golden Image ............................................. 20
   2.4 Conclusion ............................................................... 21

3. GENERATING A DEFECT FREE IMAGE ................................... 23
   3.1 The Inertia Method .................................................... 24
      3.1.1 Inertia Definition ............................................. 24
      3.1.2 The SGLDM ..................................................... 27
      3.1.3 Inertia For Period Measurements ............................ 29
      3.1.4 Reduction In Computation Time ............................. 30
   3.2 The Power Spectrum Method ........................................ 31
      3.2.1 Power Spectrum Definition .................................... 32
      3.2.2 Parameter Selection ........................................... 33
      3.2.3 The Limitations of Using The Power Spectrum
            Method for Period Estimation ............................... 39
      3.2.4 Period Estimation ............................................. 40
   3.4 Comparison Between The Inertia Method
      And The Power Spectrum Method ................................... 41
   3.4 Algorithm Definition ............................................... 44

4. LOCATING DEFECTS AND GENERATING A DEFECT MAP ............ 48
   4.1 The Local Method for Defect Detection .......................... 48
   4.2 Illustrative Example ............................................... 52
   4.3 The Global Approach For Defect Detection ..................... 54
   4.4 False Alarms ......................................................... 54
   4.5 Morphological Filtering ........................................... 55
   4.6 Performance Analysis of The Differencing Technique ....... 59
      4.6.1 Performance Of The Local Method ......................... 60
         4.6.1.1 Error In Period Estimation ......................... 60
         4.6.1.2 Nonuniform Illumination ............................... 63
         4.6.1.3 Misalignment ........................................... 66
         4.6.1.4 Nuisance Defects ....................................... 66
# TABLE OF CONTENTS - Continued

- 4.6.1.5 Window Size ........................................69
- 4.6.1.6 Defect Detection Results Using
  The Local Method ........................................70
- 4.6.1.7 False Alarm Rate ....................................71
- 4.6.1.8 Probability Of Defect Detection
  Using The Local Method ..................................71
  - 4.6.1.8.1 Nonuniform Illumination ....................77
  - 4.6.1.8.2 Nuisance Defects ............................79
- 4.6.2 The Global Method Performance .....................81

5. DEFECT CLASSIFICATION ...................................93

- 5.1 Defect Description .......................................94
- 5.2 System Implementation ...................................97
- 5.3 Component Labeling ......................................98
  - 5.3.1 The Labeler Algorithm Description ...............98
- 5.4 Images Needed By The Defect Classifier ............102
- 5.5 The Defect Classifier ..................................105
- 5.6 Defect Classification Results .........................108

6. CONCLUSIONS ................................................125

REFERENCES ...................................................128
LIST OF ILLUSTRATIONS

Figure

3.1 Typical Inertia Function Of Periodic Texture, Using Periodic IC Images..........................26
3.2 The Effect Of Zero Padding On Frequency Resolution..................38
3.3 Periods In Periodic IC Image........................................46
3.4 Generating The Defect Free image.................................47
4.1 Division Of The Image Into Blocks...............................50
4.2 Block Divided Into Windows......................................51
4.3 The Average Absolute Difference Values..........................53
4.4 The Effect Of Morphological Filtering............................58
4.5 The Effect Of Period Miscalculation................................62
4.6 Illumination Effect On Defect Detection.............................64
4.7 How Averaging Reduces The Effect Of Nonuniform Illumination On Defect Detection.................65
4.8 The Local Method Performance......................................68
4.9 Illustration Of The Probability Computation For Defect Detection..................................72
4.10 Distribution Function \( \Phi(z) \)........................................74
4.11 Illustration Example..................................................77
4.12 Illustration Example..................................................78
4.13 Defect Detection (test 1,2)..........................................84
4.14 Defect Detection (test 3,4)..........................................85
4.15 Defect Detection (test 5,6)..........................................86
4.16 Defect Detection (test 7,8)..........................................87
4.17 Defect Detection (test 9,10).........................................88
4.18 Defect Detection (test 11,12).................................89
LIST OF ILLUSTRATIONS - Continued

4.19 Defect Detection (test 13,14) .........................90
4.20 Defect Detection (test 15,16) .........................91
4.21 Defect Detection (test 17,18) .........................92
5.1 The Circuit For Defect Classification ................95
5.2 Original Binary Image ................................100
5.3 Initial Labeled Image ................................101
5.4 Final labeled Image ................................102
5.5 The Four Images Needed By The Defect Classifier...104
5.6 The Defect Classifier ................................107
5.7 Classification (test 1) .................................109
5.8 Classification (test 2) .................................110
5.9 Classification (test 3) .................................111
5.10 Classification (test 4) .................................112
5.11 Classification (test 5) .................................114
5.12 Classification (test 6) .................................115
5.13 Classification (test 7) .................................116
5.14 Classification (test 8) .................................117
5.15 Classification (test 9) .................................118
5.16 Classification (test 10) ...............................119
5.17 Classification (test 11) ...............................120
5.18 Classification (test 12) ...............................121
5.19 Classification (test 13) ...............................122
5.20 Classification (test 14) ...............................123
5.21 Classification (test 15) ...............................124
LIST OF TABLES

Table

3.1 Inertia Measurements........................................30
3.2 Comparison Between the Inertia and the Power Spectrum Methods.................41
3.3 Comparison Between the Inertia and the Power Spectrum Methods Using an Actual IC Images........43
4.1 The Average Absolute Difference Values For An Actual IC Image......................52
4.2 Kernel Size Vs. False Alarm Area..................................57
4.3 Defect Types..........................................................59
4.4 The Relationship Between Defect Area And The Estimated Period Accuracy........60
4.5 The Effect of Nonuniform Illumination................................63
4.6 The Window Size Effect..............................................69
4.7 Defect Detection Results By The Local Method......70
4.8 The Relationship Between Defect Area And The Estimated Period Accuracy........81
4.9 Nonuniform Illumination Effect.................................81
4.10 Defect Detection Results By The Global Method....82
5.1 Classification Results (test 1)..............................109
5.2 Classification Results (test 2)..............................110
5.3 Classification Results (test 3)..............................111
5.4 Classification Results (test 4)..............................113
5.5 Classification Results (test 5)..............................114
5.6 Classification Results (test 6)..............................115
5.7 Classification Results (test 7)..............................116
5.8 Classification Results (test 8)..............................117
<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.9</td>
<td>Classification Results (test 9)</td>
<td>118</td>
</tr>
<tr>
<td>5.10</td>
<td>Classification Results (test 10)</td>
<td>119</td>
</tr>
<tr>
<td>5.11</td>
<td>Classification Results (test 11)</td>
<td>120</td>
</tr>
<tr>
<td>5.12</td>
<td>Classification Results (test 12)</td>
<td>121</td>
</tr>
<tr>
<td>5.13</td>
<td>Classification Results (test 13)</td>
<td>122</td>
</tr>
<tr>
<td>5.14</td>
<td>Classification Results (test 14)</td>
<td>123</td>
</tr>
<tr>
<td>5.15</td>
<td>Classification Results (test 15)</td>
<td>124</td>
</tr>
</tbody>
</table>
ABSTRACT

A defect detection algorithm applicable for periodic VLSI circuitry is presented in this thesis. Even though the algorithm is based on the reference comparison approach, the periodicity of the circuit eliminates the need for the so called "golden wafer". The suggested algorithm has demonstrated the ability to detect defects of small area (0.023% of the image area). In addition, the algorithm was 93% successful in defect detection and has a false alarm rate of 0.067 per inspected frame, based on testing 20 frames.

Moreover, a defect classification algorithm for one periodic circuit type is also implemented. The defect classifier was 95% successful in correctly classifying 20 defects.
CHAPTER 1

INTRODUCTION

Over the years computer vision techniques have been used extensively for detecting and classifying defects in many materials such as wood, glass, steel, printed circuit boards, and others. For the past few years, special attention has been given to the inspection of integrated circuits. As integrated circuits become smaller and more complex, manual inspection becomes very difficult and costly. Breakthroughs in computer electronics, image processing, and pattern recognition have resulted in reliable industrial visual inspection equipment, such as the KLA 2020. However, this inspection equipment is expensive and not flexible enough to accommodate the wide variety of integrated circuits that exist. Daily advances in integrated circuit manufacture create the need for cheap, fast, accurate, and flexible visual inspection systems. Since current hardware inspection equipment is expensive and designed for a very specific application, inspection by software is the logical alternative. Even though software programs developed for circuit inspection are much slower
than hardware inspection equipment, they are cost effective and easier to update. As faster hardware and specialized processing chips become available, this gap in speed may be narrowed.

Recently, a number of defect detection algorithms have been developed for detecting defects in integrated circuits. These algorithms generally fall into two categories: first, the differencing approach and second, design-rule verification [Chin, 1987]. In the differencing approach, the wafer being inspected is compared with a flawless version of the same wafer. The differencing approach is flexible and powerful because it is easy to define a set of feature criteria which facilitate defect detection. However, the differencing approach is plagued by the need for large storage space for the reference wafer, precise registration between the inspected wafer and the reference wafer, and similar gray level contrast. In design-rule verification, wiring and pads are checked to determine if they fall within design requirements. The design-rule approach is less sensitive to registration than the differencing approach, but misses large flaws and distorted features.

The characteristics of the inspected integrated circuit dictate the technique used in developing an algorithm for defect detection. The new line of integrated circuits, Very Large Scale Integrated (VLSI) circuits, are normally of
complex and variable structure. Some VLSI circuits are of rectilinear, periodic, or circular structure. In addition, each circuit type, such as periodic, can have many different basic patterns. Defects in VLSI circuits can be of many different types: shorts, opens, over-etching, under-etching, pad size violations, and spurious metals [Chin, 1987]. With current image processing techniques, it is almost impossible to develop a defect detecting algorithm that is applicable for all circuit types or even for all patterns of one circuit type. Almost all currently developed algorithms are highly application specific; they require prior knowledge of the circuit and the possible defect types.

In addition to defect detection, the second important aspect of integrated circuit inspection is the problem of defect classification. Using defect characteristics (shape, location, size, and contrast), a rule based system can be developed to identify defect types. The developed rule based system should use enough human knowledge so that it too will classify defects expertly. The accuracy and sophistication of such systems varies with the number of possible defect types and their characteristics.

This thesis focuses on detecting and classifying defects in periodic VLSI circuits via computer vision techniques. Since most VLSI circuits have a periodic structure, such as DRAMs, developing an algorithm for defect
detection in periodic circuits is very worthwhile. A digital image of the inspected circuit is acquired, then image processing techniques are employed to extract the defect and generate a defect map, showing its shape and location. When dealing with periodic circuits, the periodicity of the circuits overcomes many of the drawbacks of the differencing technique. The repetitiveness of the circuit makes it feasible to extract its basic structure pattern. Then a reference image, needed for comparison, is constructed from the basic structure pattern. Because the defect free reference image is constructed from the inspected wafer the need for excessive storage space, exact registration, and similar contrast are eliminated. The problem then becomes finding suitable feature criteria on which the comparison must be based. The selected criteria must contain adequate information about the image to be affected when a defect occurs.

The algorithm to do this consists of two major steps:

1) Extract the basic structure pattern and generate a reference image.

2) Compare the reference image and the inspected image, then generate a defect map of all possible defects.

The above approach assumes that the inspected circuit does not contain too many defects which interfere with generating the reference image.
Along with the defect detection algorithm suggested above, a simple rule based system for defect classification is implemented. The suggested system uses the generated defect map and a set of selected defect features to attempt to distinguish and categorize potential circuit defects. Even though the suggested defect detection algorithm is applicable for all periodic circuit types, the defect classification system is applicable for the one specific periodic circuit. For defect classification, prior knowledge of the circuit structure and its possible defects is crucial in designing an expert system for defect classification.
CHAPTER 2

PUBLISHED METHODS FOR DETECTING DEFECTS IN PERIODIC PATTERNS

During the past few years, a number of algorithms have been developed for the purpose of detecting defects in periodic patterns, such as those found in memory chips and fabrics. Three of these methods are: optical spatial filtering, design-rule checking, and reference-comparison.

Following is a brief review of some of the published work on using these methods for defect detection in periodic patterns.

2.1 Optical Spatial Filtering

The optical spatial filtering method, which uses two structurally identical lenses, has lead to the development of the Insystem 8600. The method consists of a Fourier plane spatial filter combined with a holograph. The two lenses are designed to perform optical and inverse optical Fourier transforms. When one of the lenses optically Fourier transforms a given image, the diffracted light corresponding
to periodic frequencies will accumulate at specific points in the Fourier transform plane. Defects normally have very few spatial frequencies overlapping the frequencies of the periodic structure. Therefore, if a filter is used to block the points corresponding to the periodic structure frequencies and then the other lens is used to perform the inverse Fourier transform of the remaining points, defect information will be imaged in the output plane. This method for defect detection is much faster than employing image analysis algorithms. However, expensive optical components and difficulties in designing Fourier transform filters are the limitations in employing this technique for practical applications [Fusek, et al., 1987].

2.2 Design-Rule Checking Approach

Several commercial systems for the inspection of printed wiring boards are based on the design rule verification method [Danielson, 1979 and Bentley, 1980]. This method compares the inspected patterns of wiring and pads with some predefined design rules. Prior knowledge of the inspected circuit pattern structure is required for employing this method. In addition, this technique is known to miss large flaws and distorted features during wafer inspection [Chin, 1987]. This method is rarely used for detecting defects in complicated periodic patterns such as those of VLSI circuits; because, the reference comparison approach (below)
proved to be a much simpler, more accurate, and more general method.

2.3 Reference Comparison Approach

Most of the suggested algorithms for defect detection are based on the reference comparison scheme. This method appears in two types:

1) Self reference image - two identical regions are compared in the same image.

2) Golden image - the image under test is compared with an ideal image with perfect parts.

Following is a review of the work that is based on both types of the reference method for defect detection in periodic patterns.

2.3.1 The Self Reference Image

In 1986, Clark, Parui, You, and Hashim developed an algorithm for detecting defects in fabrics. Their method is based on period estimation and a pixel to pixel differencing scheme. First, the period in both directions is estimated, then the absolute difference between each pixel and the two neighboring pixels on both sides, and one period apart from the inspected pixel, is determined. A threshold value is used to flag pixels as defective or nondefective. This comparison scheme is repeated for both the horizontal and
the vertical directions of the inspected image [Clark, et al., 1986].

A similar approach was used by the IBM Western Research Center in developing the IBM P300 inspection system. This system assumes periodicity in the horizontal direction only, and determines the period before employing the inspecting algorithm. The inspection algorithm is composed of two parts: low level and high level. The low level algorithm compares each pixel with its two neighboring pixels on both sides, and one period apart from the inspected pixel. Unlike the method used by Clark et al. to flag potential defective pixels, the IBM P300 uses a more sophisticated comparison scheme. The high level algorithm is a combination of a statistical method and a repeated application of the low level algorithm on different images of the inspected wafer. The high level algorithm is used to eliminate false alarms from the defect map [Dom, et al., 1988].

In both systems each pixel is compared with two neighboring pixels in order to eliminate the ambiguity that might exist if one pixel is used in comparison [Dom, et al., 1988].

Konishi, Misono and Kato [1982] have developed an algorithm for detecting defects in CCD arrays. Their method is based on cell-to-cell comparison. A differential operator and a thresholding scheme are used to extract the
pattern boundary and to generate a binary edge image of the inspected CCD wafer. Taking advantage of the repetitive structure of the CCD array, a cell-to-cell comparison is made on the binary image to locate defects. However, this approach is very sensitive to noise and is limited to patterns of simple structure.

2.3.2 The Golden Image

Yoda et al. [1988] has investigated defect detection and classification in periodic VLSI circuit. In his approach a differencing comparison scheme between two images, A and B, is employed for detecting defects. Both images, A and B, are acquired using the same single optical setup, but image B is obtained one period delayed compared to image A. Since subtraction generates too many false alarms, Yoda et al. use morphological size filtering and morphological shape filtering to reduce the number of false alarms in the defect map.

The OSI Corporation has developed a system based on comparison between the inspected wafer and a similar but defect free wafer. To avoid the problems that arise in employing image to image comparison schemes, such as illumination effects and image to image registration, OSI developed a unique template matching algorithm that avoids direct image to image comparison. Little detail of their
template matching algorithm has been revealed. Since this system requires two images, a defective image and defect free image, it can inspect all wafers with repetitive or random patterns [Dralla, et al., 1987].

2.4 Conclusion

The reference comparison approach is the most suitable approach for detecting defects in periodic patterns. Both types of reference comparison approach, self reference and the golden wafer methods, have advantages and disadvantages. The self reference comparison method eliminates the need for the reference wafer, but requires a complicated comparison algorithm for locating and identifying defects [Dom, et al., 1988]. On the other hand, employing the golden wafer method for defect detection enables a simple comparison scheme to be employed [Yoda, et al., 1988]; however, nonuniform illumination effects, storage space for the defect free wafer, and image to image registration limit the practical application of this method. For periodic patterns, a combination of the self reference approach and golden wafer approach can be employed which would eliminate the disadvantages of using either one. The golden wafer can be extracted from the inspected wafer and a fast and simple comparison scheme can be employed for defect detection. The following chapters discuss a new approach which employs image processing techniques for defect detection in periodic
patterns. A comparison between the above suggested methods and our method will be presented in the concluding chapter.
CHAPTER 3

GENERATING A DEFECT FREE IMAGE

The first challenge in using the differencing technique is to obtain a defect free reference wafer to be compared with every tested wafer. The tested wafer must have the same characteristics as the reference wafer. Many comparison techniques require storing an image of a defect free wafer, but this imposes constraints on how testing must be conducted. Making sure that every tested wafer is registered with the reference wafer and that both wafers have similar contrasts is costly and time consuming. However, for periodic images the reference wafer can be deduced from the defective wafer itself. By computing the periodicity in both the horizontal (x) and the vertical (y) directions, the wafer's basic structure pattern can be extracted and used to generate a defect free reference wafer. The reference wafer will be exactly registered with the defective wafer and both wafers will have similar contrast.
3.1 The Inertia Method

A texture is defined as a structure composed of a large number of basic patterns, or elements, and a placement rule defining the distribution of such. Therefore, texture analysis techniques can be employed for the analysis of periodic VLSI circuits. Textures may be analyzed using a statistical approach or a structural approach. The statistical approach is generally applicable, while the structural approach is only used if the placement rules of the basic structure pattern can be extracted. For periodic images, computing the periodicity in both horizontal and vertical directions will result in extracting the basic structure pattern and its placement rules. Conners and Harlow [1980] have suggested the use of a texture attribute based on second order statistics for this purpose: the inertia.

3.1.1 Inertia Definition

The inertia is defined as:

\[
I(\theta,d) = \sum_{i=0}^{Ng-1} \sum_{j=0}^{Ng-1} (i - j)^2 S(i,j,\theta,d)
\]  

\(d = 0,1,2 \ldots L.\)

where

- \(i,j\) are summation indices over the gray level range.
- \(\theta\) is the direction of computing the inertia. (If \(\theta = 0^\circ,\)
then the inertia is computed in the horizontal (x) direction. If $\theta = 90^\circ$, then the inertia is computed in the vertical (y) direction.)

d is the intersampling distance.
L is the length of the image in the direction $\theta$.
Ng is the number of possible gray levels in the image.
S SGLDM computed for a particular d and $\theta$. (See below.)
I is the inertia measured for a given d and $\theta$.

The inertia is a periodic function in the direction in which the original image is periodic. The zeros of I are used to measure the period. Since for real images, the inertia is not zero at multiples of the texture period, because of nonuniform illumination from one part of the image to another, the period is computed as the distance between two local minima in I. In addition, period length can be measured as the difference between zero and the first local minima. In this case the last method is used for period measurements. A typical inertia function for periodic images is shown in Fig. 3.1.

The inertia can be computed in the horizontal and vertical directions, at angles of 0 and 90 degrees, respectively. This leads to the corresponding period in each direction and the extraction of the basic texture pattern.
Figure 3.1 Typical Inertia Function Of Periodic Texture, Using Periodic IC Images.
3.1.2 The SGLDM

Computing the inertia requires computing the Spatial Gray level Dependence Matrix (SGLDM). The SGLDM contains the second order statistics of a texture. It is the estimated probability of going from one gray level value to another for each combination of \( d \) and \( \theta \). Each entry in the SGLDM corresponds to the number of times a certain pair of gray levels are found in the original image, at a pair of pixels with displacement \( d \) in the direction \( \theta \). Probabilities are taken by dividing each entry in the SGLDM by the sum of all entries in the SGLDM. The SGLDM is a square matrix, and its dimensions are equal to the number of possible gray level values in the image.

Examples of the SGLDM and the resulting inertia are given below.

**Pixel values (10 X 10 image).**

<table>
<thead>
<tr>
<th>x: 0 1 2 3 4 5 6 7 8 9</th>
</tr>
</thead>
<tbody>
<tr>
<td>y:</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>9</td>
</tr>
</tbody>
</table>

The inertia can be computed for this image in the
horizontal direction ($\theta = 0^\circ$) and the vertical direction ($\theta = 90^\circ$) for displacement values of 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

Samples of the SGLDM computed for several displacement values are given below.

**SGLDM ($d = 1, \theta = 0^\circ$)**

```
  i: 0 1 2 3 4 5 6 7
 j: 0 3 2 2 0 0 0 0 0
  1 3 3 2 3 0 0 0 2
  2 0 2 1 3 2 0 1 0
  3 0 1 0 1 9 3 0 1
  4 0 0 0 4 0 4 3 0
  5 0 0 0 1 0 0 3 3
  6 1 2 1 1 0 0 5 5
  7 0 1 2 1 0 0 5 4
```

At row 3 column 5 in the SGLDM above, the entry 3 is found. This corresponds to three occurrences of the gray level pair (3,5), where the gray level value 5 is one pixel, $d = 1$, to the right, $\theta = 0^\circ$, from the gray level value 3. Examining the original image, these occurrences are found at pixel locations (row,column): (5,1), (5,2), (5,6), (5,7), and (6,6) (6,7).
SGLDM: $d = 5, \theta = 0^\circ$.

\[
\begin{array}{cccccccc}
  & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 3 & 2 & 2 & 1 & 0 & 0 & 0 & 0 \\
2 & 0 & 1 & 1 & 3 & 1 & 0 & 0 & 0 \\
3 & 0 & 0 & 1 & 5 & 0 & 1 & 0 & 0 \\
4 & 0 & 0 & 0 & 0 & 0 & 4 & 1 & 1 \\
5 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 1 \\
6 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 6 \\
7 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 3 \\
\end{array}
\]

SGLDM: $d = 2, \theta = 90^\circ$.

\[
\begin{array}{cccccccc}
  & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 1 & 2 & 0 & 0 & 2 & 2 \\
2 & 0 & 0 & 0 & 1 & 2 & 0 & 2 & 3 \\
3 & 0 & 0 & 1 & 2 & 0 & 0 & 2 & 2 \\
4 & 4 & 3 & 0 & 2 & 3 & 0 & 2 & 1 \\
5 & 2 & 2 & 2 & 0 & 1 & 3 & 0 & 1 \\
6 & 0 & 1 & 0 & 3 & 0 & 1 & 1 & 1 \\
7 & 6 & 0 & 2 & 1 & 3 & 1 & 1 & 6 \\
\end{array}
\]

3.1.3 Inertia For Period Measurements

The inertia values in the above image are tabulated below:
Table 3.1 Inertia Measurements.

<table>
<thead>
<tr>
<th>θ</th>
<th>d</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>0°</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0°</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>0°</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>0°</td>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>0°</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>0°</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>0°</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>0°</td>
<td>7</td>
<td>10</td>
</tr>
<tr>
<td>0°</td>
<td>8</td>
<td>11</td>
</tr>
<tr>
<td>0°</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>90°</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>90°</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>90°</td>
<td>2</td>
<td>9</td>
</tr>
<tr>
<td>90°</td>
<td>3</td>
<td>11</td>
</tr>
<tr>
<td>90°</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>90°</td>
<td>5</td>
<td>11</td>
</tr>
<tr>
<td>90°</td>
<td>6</td>
<td>16</td>
</tr>
<tr>
<td>90°</td>
<td>7</td>
<td>15</td>
</tr>
<tr>
<td>90°</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td>90°</td>
<td>9</td>
<td>10</td>
</tr>
</tbody>
</table>

Assuming the image above is periodic then:

(1) Period length in horizontal direction = 5.

(2) Period length in vertical direction = 4.

3.1.4 Reduction In Computation Time

Because calculating the inertia requires the use of all of the possible gray level values in the image, it behooves us to reduce this number in order to perform the computation in a reasonable time. Gray level reduction can be accomplished in one of two ways:

1) Reducing the number of quantum levels to 16 equally spaced steps (linear reduction).
2) Equal probability quantizing (EPQ).

Both methods are frequently employed to transform, with little or no loss of information, the original image into a new image containing fewer gray level values. Equal probability quantizing is best used when the image has a low contrast. EPQ simply means that the number of pixels in the quantized image is the same for each gray level value. Linear reduction is employed if the image has high contrast. It is also easier and faster to compute. In this case linear reduction works well in preserving most of the image structure because of the flatness of the original image histogram. Thus, reducing the 256 possible gray level values in an image to 16 gray level values, for example, reduces inertia computation by a factor of 256.

Since one is normally dealing with large images (256 x 256), reducing inertia computation overhead is essential. In periodic images each row and column has periodic characteristics. Consequently, the inertia (of an image) can be computed using one row and one column only. From the inertia measurements, the period in both horizontal and vertical directions can be computed.

3.2 The Power Spectrum Method

The periodicity of the image can also be computed by evaluating the energy distribution in the Power Spectrum.
For a periodic image, all the energy in its true power spectrum will be concentrated at frequencies representing the periodicity of the image. However, in practice only a finite segment of the image is available for power spectrum computation. The computed spectrum then is that of the periodic extension of this segment, rather than that of the ideal original image.

3.2.1 Power Spectrum Definition

For a periodic signal \( f(k) \) the discrete Fourier transform (DFT) is defined as:

\[
F(n) = \sum_{k=0}^{N-1} f(k) e^{-j2\pi kn/N}
\]

where

\( F(n) \) is the DFT.

\( f(k) \) is the input periodic signal.

The input is \( N \) real values of the periodic signal \( (k = 0, 1, 2, \ldots, N-1) \) and the output is \( N \) complex values of the frequency spectrum \( (n = 0, 1, 2, \ldots, N-1) \).

The power spectrum of the periodic signal \( f(k) \) is

\[
H(n) = |F(n)|^2
\]

where \( n = 0, 1, 2, \ldots, N-1 \).
3.2.2 Parameter Selection

Before computing the power spectrum for a given signal, two parameters are selected: the sampling frequency \((fs)\), and number of samples \((N)\). The spatial extent of the \(N\) consecutive signal values is \(N(1/fs)\). This is also the fundamental period of the DFT analysis, so the fundamental frequency of both the DFT and the power spectrum is the reciprocal, \(fs/N\). In addition, the power spectrum is only defined at integer multiples of the fundamental frequency \((n = 0, 1, 2, \ldots, N-1)\). This means that the frequency of either the DFT or the power spectrum resolution then is

\[
\Delta f = \frac{fs}{N}
\]

The sampling frequency must be at least twice the frequency of the signal to be inspected. Ideally the number of samples is selected to provide a desired frequency resolution as described by Eq. 4. In this study, however, the sampling rate is fixed by the camera system and the number of samples is fixed at 256. Thus, the inspected signal is truncated after 256 samples. Because both the sampling rate and the number of samples are fixed, frequency resolution is fixed at 1/256 of the sampling rate. This cannot be improved.

Nevertheless, because the power spectrum is only defined at integer numbers \((n = 0, 1, 2, \ldots, N-1)\), accuracy in
period estimation is affected by the number of image periods in the inspected sampled data. If the periodic signal is truncated at an integer multiple of the image period, then period estimation is very accurate. Theoretically it is perfect in this special case. However, if the periodic signal is truncated at other than an integer multiple of the image period, spectral leakage results and period estimation is not accurate. This results from the limited resolution described above. Hamming windowing and zero padding can be employed to reduce the effect of spectral leakage. Hamming windowing is used to suppress the abrupt discontinuities at the end of the data record resulting from truncating the number of samples to 256. Zero padding increases the number of samples to a greater number than 256, without changing the sampling rate. Thus it interpolates between spectral values and decreases frequency intervals in the power spectrum. Therefore, zero padding is used as a way to increase frequency resolution to help reduce the picket fence effect. The following equation can be used to compute frequency resolution in the power spectrum when zero padding is used:

\[
\Delta f = \frac{fs}{N + M} = \frac{fs}{L}
\]  

where

\( M \) is the number of zeros added at the end of the data record.
\( L \) is the length of the padded data record = \( M + N \).

The more zeros added the better frequency resolution is achieved. For examples, if \( \Delta f = 1 \) when \( N = 256, M = 0, \) and \( L = 256, \) then, adding 256 zeros at the end of the data buffer, i.e. \( N = 256, M = 256, L = 512, \) will improve frequency resolution (by decreasing the fundamental analysis frequency, \( \Delta f \)) to 0.5. (See Fig. 3.2.)
Figure 3.2 The Effect Of Zero Padding On Frequency Resolution.
(a) The periodic signal, period length is 102.4 pixels.
(b) The power spectrum without zero padding. Buffer length is 256. Period length estimated from this spectrum is 128 pixels.
(c) The power spectrum with zero padding. Buffer length is 512. Period length estimated from this spectrum is 102.4 pixels.
3.2.3 The Limitations of Using The Power Spectrum Method For Period Estimation

To employ the power spectrum method for period estimation in periodic images, a frequency resolution of less than 0.016 (see below) is needed to achieve an accuracy in period estimation to within one pixel of the actual period. Frequency resolution can be increased by padding zeros to the end of the data buffer, as described above. Therefore, the minimum length \( L \) of the data buffer needed to achieve this frequency resolution will now be computed. (See below.)

We have \( N = 256 \) data. With a normalized sampling interval \( (\Delta f) \) of 1, the sampling frequency \( (f_s) \) is 256. Assume the 256 sampled data contains at least 2 periods, i.e. period length is less than or equal to 128 pixels. If the period length is 128 pixels, then the data contain 2.0 periods. Thus the frequency corresponding to this period is 2.0. If period length is 127 pixels, then the data contain 2.016 periods and the corresponding frequency is 2.016. If these two cases are to be distinguished in the power spectra, then the required resolution \( \Delta f = 2.016 - 2.0 = 0.016 \). This is the maximum \( \Delta f \) value needed to achieve less than one pixel accuracy in period estimation for any arbitrary period length.

If \( \Delta f = 0.016 \), then \( L \) is found as follows:
\[ \Delta f = \frac{f_s}{L} \]

i.e. \( L = 16000 \).

Thus the buffer length must be at least 16000 samples to achieve accuracy in period estimation to within one pixel of the actual period.

To employ FFT techniques in power spectrum computation the minimum record length needed is 16384. Thus, before computing the power spectrum 16128 zeros must be appended to the 256 sampled data.

3.2.4 Period Estimation

The following procedure is used for period estimation using the power spectrum:

1) One row and one column of the inspected image are obtained.
2) A Hamming window is applied to the data.
3) The 256 data buffer is zero padded to 2048 (Maximum buffer size permitted in the system used for this study).
4) The power spectrum is then computed.
5) The first peak in the power spectrum (above zero frequency) is located.
6) The period is estimated by dividing buffer length (2048) by the location of the first peak.
Because the maximum buffer length (L) that can be used in the procedure above is limited to 2048, due to the limitations of the compiler used to perform this procedure, 
\[ \Delta f = \frac{256}{2048} = 0.125. \]

3.3 Comparison Between The Inertia Method And The Power Spectrum Method

The following table compares the power spectrum method to the inertia method for period estimation using manually generated periodic signals. In the table below \( N = 256 \), \( L = 2048 \), and \( \Delta f = 0.125 \).

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>( a )</td>
<td>( b )</td>
<td>( c )</td>
<td>( d )</td>
<td></td>
</tr>
<tr>
<td>84</td>
<td>84</td>
<td>85.3</td>
<td>3.05</td>
<td></td>
</tr>
<tr>
<td>84.50</td>
<td>85</td>
<td>85.3</td>
<td>3.03</td>
<td></td>
</tr>
<tr>
<td>120.47</td>
<td>120</td>
<td>120.47</td>
<td>2.125</td>
<td></td>
</tr>
<tr>
<td>56.88</td>
<td>57</td>
<td>56.88</td>
<td>4.5</td>
<td></td>
</tr>
<tr>
<td>77.57</td>
<td>78</td>
<td>78.76</td>
<td>3.3</td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>64</td>
<td>64</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.2

Comparison Between the Inertia And The Power Spectrum Methods.

key:

\( a \) - Actual period length (in pixels).

\( b \) - Period length measured by the inertia method (in pixels).

\( c \) - Period length measured by the power spectrum method (in pixels).

\( d \) - Number of periods in the sampled data.
Due to the spectral leakage effect and a frequency resolution of 0.125, the periodicity of only the signals which are truncated to x.0, x.125, x.25, x.5, x.625, x.75, and x.875 periods, where x is an integer value, can be exactly estimated using the power spectrum method. In these cases, the power spectrum method is more accurate than the inertia method. (See rows 3 and 4 in the table 3.2.) However, the inertia method is not affected by truncating the periodic signal to either an integer multiple of the period or a non-integer multiple of the period. Even though the inertia method is only accurate to within one pixel of the actual period, the inertia method is more accurate than the power spectrum method when the signals are truncated to any real number of periods other than x.125, x.25, x.5, x.625, x.75 and x.875, where x is an integer. (See rows 1, 2 and 5 in table 3.2.) Thus on average, the inertia method is more accurate than the power spectrum method.

Since, during image digitizing, signals are truncated to a real number multiple of the period, period estimation using the inertia method is more reliable than the power spectrum method.

The following table compares the power spectrum method to the inertia method using actual periodic images. The same row and column used in inertia computations are used in power spectrum evaluations.
Table 3.3
Comparison Between the Inertia and the Power Spectrum Methods Using an Actual IC Images.

Key

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
<th>g</th>
<th>h</th>
<th>i</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>42</td>
<td>42</td>
<td>42.67</td>
<td>97</td>
<td>97</td>
<td>97.5</td>
<td>14.32</td>
<td>22.68</td>
</tr>
<tr>
<td>2</td>
<td>42</td>
<td>42</td>
<td>42.67</td>
<td>98</td>
<td>97</td>
<td>97.5</td>
<td>27.18</td>
<td>27.10</td>
</tr>
<tr>
<td>3</td>
<td>43</td>
<td>43</td>
<td>42.67</td>
<td>96</td>
<td>96</td>
<td>97.5</td>
<td>19.32</td>
<td>22.17</td>
</tr>
<tr>
<td>4</td>
<td>42</td>
<td>42</td>
<td>42.67</td>
<td>97</td>
<td>97</td>
<td>97.5</td>
<td>22.42</td>
<td>24.86</td>
</tr>
</tbody>
</table>

Computing periodicity using the inertia method is seen to be very accurate. The root mean square difference between the original image and the defect free image is smaller when the inertia method is used to extract the basic structure pattern than when the power spectrum is used to extract the basic structure pattern. Therefore, period estimation by the inertia method is much closer to the actual image.
periodicity than is the periodicity estimated by the power spectrum method. The inertia method is more accurate, because the tested images were truncated to 6.1 periods in the horizontal directions and and 2.67 periods in the vertical direction. In addition, estimates of period based on inertia computation match the actual period. (The "actual period" in this case is measured by hand from the inspected image.)

3.4 Algorithm Definition

The following algorithm is used to generate a defect free reference wafer:

1) Obtain an image of the defective circuit. (This will be called the original image.)

2) Compute the inertia in the original image as described above. Use the inertia measurements to determine the periodic dimensions in both the horizontal and vertical directions.

3) Having found the horizontal and vertical periods, the original image can be divided into blocks. Each block contains one basic structure pattern and has the dimensions of one period in both the horizontal and vertical directions. (See Fig. 3.3.)

4) Average all the blocks together to generate an average block.
5) Use the average block to generate a defect free image by simply replacing every block in the original image by the average block, call this the reconstructed image (See Fig. 3.4.)

In the above algorithm an assumption is made that many of the blocks in step 4 do not contain any potential defects.

A method for extracting the circuit's basic structure pattern has been developed. Thus, a defect free image, to be compared with the inspected image, is generated. A comparison technique based on the differencing method for defect detection will be presented in the next chapter.
Figure 3.3 Periods In Periodic IC Image.

Image is 256 x 256 pixels.
Period in horizontal (x) direction = 60 pixels.
Period in vertical (y) direction = 42 pixels.
Each block is 60 x 42 pixels.
Each block contains one basic structure pattern.
Figure 3.4 Generating The Defect Free Image.

(a) Original defect image.
(b) Defect free reference image.
CHAPTER 4

LOCATING DEFECTS AND GENERATING A DEFECT MAP

A defect in a periodic image is considered to be any inhomogeneity in the image structure. The detection of inhomogeneities can be achieved by comparing an image with a defect free reference image. Then a set of suitable feature criteria is selected on which to make comparison. In order to compute a feature which is sensitive to local inhomogeneities, one must work at a local scale within the image. This can be achieved by dividing the original image into subimages, and computing a feature, needed for comparison, in each of these. The average absolute difference is a suitable comparison criterion because it is very sensitive to changes in gray level values.

Two approaches, local and global, based on the differencing method can be employed for detecting defects.

4.1 The Local Method For Defect Detection

After generating a defect free image from the original defective image (Chapter 3), the following steps are used
for locating defects.

1) Divide the image into 64 x 64 pixel blocks. At the resolution used herein i.e. 256 x 256 pixel images, this yields 16 blocks. (See Fig. 4.1.)

2) Divide each block into smaller 4 x 4 pixel windows. Each larger block, therefore, consists of 256 windows. (See Fig. 4.2.)

3) For each window in a given block in the original image, compute the average absolute difference between the original image and the defect free image.

4) Rank the 256 average absolute difference values in each block, and use the maximum value to represent the absolute difference for that block. Since each block is now identified by one absolute difference value, and the image is divided into 16 blocks, the image contains 16 different absolute difference values.

5) Compute the mean and standard deviation of the 16 average absolute difference values.

6) A block is considered to be defective if its average absolute difference value is more than one standard deviation above the mean of the average absolute difference values.

7) After the defective blocks have been identified, the defect map is generated by subtracting these blocks in the original image from the corresponding blocks in the defect free image.
In the above algorithm two tradeoffs are evident. First, the window size determines whether or not the smallest defects are resolved. A $4 \times 4$ pixel window was selected simply because anything smaller yields too many false alarms. Second, the choice of one standard deviation cutoff criterion for the average absolute difference measurements is a compromise between the possibility of having more than one defect in the image and generating too many false alarms in the defect map.

**Figure 4.1** Division Of The Image Into Blocks.

Image is $256 \times 256$ pixels.
Each Block is $64 \times 64$ pixels.
Figure 4.2 Block Divided Into Windows.

Block is 64 X 64 pixels.
Each window is 4 X 4 pixels.
4.2 Illustrative Example

Suppose the average absolute difference between the original defective image and the reference defect free image is computed for each block, and the following set of absolute difference values resulted:

<table>
<thead>
<tr>
<th>Block Number</th>
<th>The Average Absolute Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>45</td>
</tr>
<tr>
<td>2</td>
<td>27</td>
</tr>
<tr>
<td>3</td>
<td>43</td>
</tr>
<tr>
<td>4</td>
<td>71</td>
</tr>
<tr>
<td>5</td>
<td>44</td>
</tr>
<tr>
<td>6</td>
<td>110</td>
</tr>
<tr>
<td>7</td>
<td>28</td>
</tr>
<tr>
<td>8</td>
<td>53</td>
</tr>
<tr>
<td>9</td>
<td>51</td>
</tr>
<tr>
<td>10</td>
<td>40</td>
</tr>
<tr>
<td>11</td>
<td>21</td>
</tr>
<tr>
<td>12</td>
<td>42</td>
</tr>
<tr>
<td>13</td>
<td>70</td>
</tr>
<tr>
<td>14</td>
<td>55</td>
</tr>
<tr>
<td>15</td>
<td>25</td>
</tr>
<tr>
<td>16</td>
<td>43</td>
</tr>
</tbody>
</table>

Table 4.1 The Average Absolute Difference Values For An Actual IC Image.

Mean = 48.00
Standard deviation = 21.93

In this example, blocks 4 and 6 are considered to be defective. Whereupon, Blocks 4 and 6 in the original defect image are subtracted from the same blocks in the defect free image. The defect map consists of a defect in block number 6, and a false alarm in block number 4. Then, morphological filtering operations (opening and closing) are applied to
remove very small false alarms from the defect map. Block 6 has an average absolute difference of 110 which is much greater than the average absolute difference of all other blocks. This is a strong indication that block 6 contains a defect and can be singled out as the defective block. Figure 4.3 shows the average absolute difference for each window of the inspected image above.

Figure 4.3 The Average Absolute Difference Values.

- a - b - (a) The original defective image.
- c - d - (b) The defect free image.
- d - (c) The average absolute difference for each 4 x 4 window.
- d - (d) The initial defect map, showing the defect in block number 6 and a false alarm in block number 4.
4.3 The Global Approach For Defect Detection

A simpler approach for detecting large defects can be employed instead of the local method discussed above. After generating a defect free reference image, the defect map is generated by simply subtracting the original defective image from the defect free reference image. This method is more global and faster for detecting defects than the local method. However, this global approach fails to detect defects of small size and low contrast due to the existence of false alarms in the image.

4.4 False Alarms

False alarms are created due to the following factors:
(1) Miscalculating the period by one or two pixels.
(2) Nonuniform illumination.
(3) Misalignment from one block to another. (The image is periodic at angles slightly different from 0 and 90 degrees.)
(4) Nuisance defects (defined below).

False alarms caused by variation in brightness can be greatly reduced when averaging is used in reconstructing the defect free reference image. Moreover, false alarms of this type can be further reduced if the image is normalized before processing. Image normalization uniformly distributes the brightness throughout the image.
Little can be done to reduce false alarms caused by misalignment, other than ensuring that images are digitized with careful alignment relative to wafer layout. Any misalignment will also contribute an error when the period is estimated. False alarms generated by miscalculating the period by one pixel are harmless and can be eliminated by averaging.

False alarms that naturally exist in the image, i.e. nuisance defects, are the most harmful of all. These anomalies are small and exist as a result of small differences in size of some parts of the basic structure pattern of the tested wafer, or as a result of some foreign substance on the wafer surface. These false alarms can not be considered as defects because they do not alter the tested wafer performance. Most of these false alarms are located during defect detection and are removed by applying morphological filtering operations. (See below.)

4.5 Morphological Filtering

Morphological filters are employed to eliminate small noise patterns from the defect map. The two fundamental concepts in morphological filtering are: erosion and dilation [Serra, 1982]. Erosion reduces a blob area while dilation increases a blob area. The rate at which an object is eroded or dilated is dependent on the kernel size
employed for this process. The larger the kernel size the more an object is shrunk (eroded), or grown (dilated). To eliminate false alarms from the defect map, without changing the defect area, an erosion followed by dilation (opening), is employed for this process. However, dilation followed by erosion (closing), will fill small holes in a blob.

Even though opening and closing are dual with respect to complementation they are not inverse operations [Peters, Strickland, 1986]. Opening and closing have a nonlinear effect on the shape and area of the opened or closed blob. In addition, opening and closing tend to smooth the boundaries of the opened or closed blob. (See Fig. 4.4.) Both the area, in pixels, and shape of the false alarms controls the size of the kernel employed to remove such false alarms from the defect map. The larger the false alarms area, the larger the kernel size used. However, if two false alarms have the same area, but one is concentrated and the other is spread, a smaller kernel size is used to remove the spread false alarm.

The following table summarizes the relationship between false alarm area and the size of the morphological filter needed to remove this false alarm from the defect map in one pass of the filter. The false alarms in this table are of random shape.
<table>
<thead>
<tr>
<th>Kernel Size</th>
<th>Area of False Alarm</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 x 2</td>
<td>3</td>
</tr>
<tr>
<td>3 x 3</td>
<td>11</td>
</tr>
<tr>
<td>4 x 4</td>
<td>25</td>
</tr>
<tr>
<td>5 x 5</td>
<td>40</td>
</tr>
<tr>
<td>6 x 6</td>
<td>60</td>
</tr>
<tr>
<td>7 x 7</td>
<td>83</td>
</tr>
<tr>
<td>8 x 8</td>
<td>110</td>
</tr>
</tbody>
</table>

Table 4.2 Kernel Size Vs. False Alarm Area.

For most the inspected images in this study, a 4 x 4 kernel size proved to be an adequate kernel size to remove the false alarms from the defect map and yet retain the defect in the defect map. However, for small size defects, i.e. less than 25 pixels, a smaller kernel size is used, because using a larger kernel will result in removing the defect from the defect map. (See table 4.7.)
Figure 4.4 The Effect Of Morphological Filtering On Defect Shape.

(a) The defect map before morphological filtering.
(b) The defect map after morphological filtering.
(c) The defect map before morphological filtering.
(d) The defect map after morphological filtering.
4.6 Performance Analysis of The Differencing Technique

A number of defect images were used to test the approach discussed above. All images were supplied by IBM in 3½ x 4½ ins. color Polaroid format. The images consist of four different defect types, as listed below.

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>&gt; 50 pixels</td>
<td>bright</td>
<td>connects two gates</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>&lt; 50 pixels</td>
<td>bright</td>
<td>an extension of a gate</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>10 - 80 pixels</td>
<td>similar</td>
<td>on a trench</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>15 - 90 pixels</td>
<td>dark</td>
<td>on a trench</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.3 Defect Types.

Key:

a - Defect type.
b - Defect area.
c - Defect average gray level compared with background average gray level.
d - Defect location.

Defect type 1 is a fatal bridging defect. Defect type 2 is a nonfatal bridging defect. Defects type 3 and 4 are void type defects.

When digitizing the images above, the magnification is chosen so that each image contains at least two periods in both directions. A minimum of two periods is needed for accurate period measurements. In addition, to generate a defect free reference image using averaging, the inspected image must contain a number of defect free cells.

The effects of false alarms and of changing window size
upon the results of this approach are the focus of the following discussion.

4.6.1 Performance Of The Local Method

Following is a quantitative analysis of how these false alarms affect algorithm performance:

4.6.1.1 Error In Period Estimation

The method of inertia for extracting the basic texture pattern is very accurate. Estimates of period based on inertia computation match the actual period to within the tolerance afforded by digital sampling of the camera system. (The "actual period" in this case was measured by hand from the original Polaroid prints.)

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>15</td>
<td>4</td>
<td>3x3</td>
</tr>
<tr>
<td>1</td>
<td>46</td>
<td>20</td>
<td>4x4</td>
</tr>
<tr>
<td>2</td>
<td>110</td>
<td>65</td>
<td>7x7</td>
</tr>
<tr>
<td>3</td>
<td>350</td>
<td>100</td>
<td>8x8</td>
</tr>
<tr>
<td>4</td>
<td>Defect free image is too distorted. No defect can be detected.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.4 The Relationship Between Defect Area And The Estimated Period Accuracy.

Key:

a - Error in periodicity calculation (in pixels).
b - Minimum defect area that can be detected (in pixels).
c - Maximum false alarm area in defect map (in pixels).
d - Size of morphological operator (in pixels).
The data in the table above are manually generated, i.e. the defect free image is generated with 1, 2, 3, or 4 pixels more or less than the actual period length. The table above shows that as the error in period estimation increases (column a), the false alarm area increases (column c). Thus small defects cannot be detected (column b). For example, for a 1 pixel error in period estimation, defects of area less than 40 pixels cannot be detected.

Therefore, accurate defect detection relies heavily upon accurate period estimation. Large defects may go undetected when a small error in periodicity occurs. This is mostly a result of the increasing distortion of the computed defect free image with increasing period estimation error. Figure 4.5 show the effect of period estimation upon defect detection.
Figure 4.5  An example of the effect of period miscalculation on defect detection. In this example, the error is 3 pixels in both the horizontal and vertical directions. The defect free image (4.5(b)) is defocused due to the error in the period.

---------  (a)  The original image of defect type 2.
- a - b - (b)  The defect free image.
---------  (d)  The defect map with false alarms.
- c - d - (d)  The defect map after applying morphological filtering.
4.6.1.2 Nonuniform Illumination

The table below summarizes the effect of nonuniform illumination on the defect detection scheme.

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>15</td>
<td>4</td>
<td>3x3</td>
</tr>
<tr>
<td>16</td>
<td>35</td>
<td>10</td>
<td>3x3</td>
</tr>
<tr>
<td>32</td>
<td>50</td>
<td>25</td>
<td>5x5</td>
</tr>
<tr>
<td>48</td>
<td>80</td>
<td>50</td>
<td>6x6</td>
</tr>
<tr>
<td>64</td>
<td>150</td>
<td>100</td>
<td>7x7</td>
</tr>
<tr>
<td>80</td>
<td>300</td>
<td>240</td>
<td>10x10</td>
</tr>
<tr>
<td>100</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.5 The Effect Of Nonuniform Illumination.

Key:

a - Maximum background gray level variation (in gray level).
b - Minimum defect area that can be detected (in pixel).
c - Maximum false alarm area in defect map (in pixel).
d - Size of morphological operator (in pixels).

To study the effect of nonuniform illumination on defect detection, the gray level values of 20% of the inspected image pixels are reduced by 16, 32, ... gray levels.

The local method for defect detection is very sensitive to changes in background gray level variations, since it is based on the differencing method. With a 16 gray level reduction in the gray level values of 20% of all image pixels, defects of small area and low contrast can not be detected. However, the averaging used to generate the defect
free image lessens the impact of nonuniform illumination upon defect detection. This can be seen if the defect free image is reconstructed from one basic structure pattern rather than the average of all the basic structure patterns in the image. (See Figs. 4.6 - 4.7 for comparison.)

Figure 4.6 An example of nonuniform illumination effect on defect detection. In this example, the maximum illumination difference is 48 gray levels. The defect free image is generated using averaging.

--- (a) The original defective image of defect type 2. 
- a - b - (b) The defect free image, generated by averaging. 
--- (c) The defect map with false alarms. 
- c - d - (d) The defect map after applying morphological filtering.
Figure 4.7 Nonuniform illumination effect on defect detection. The Defect free image is generated using the upper left corner basic structure pattern as the basic building block in generating the defect free image.

(a) The original defective image of defect type 2.
(b) The defect free image, generated using the upper left corner basic structure pattern.
(c) The defect map with false alarms.
(d) The defect map after applying morphological filtering.

Comparing the defect maps before applying morphological filtering (Figs. 4.6(c) and 4.7(c)), Fig. 4.6(c) contains fewer false alarms than Fig. 4.7(c). Therefore, averaging reduces the impact of nonuniform illumination on defect detection.
4.6.1.3 Misalignment

Since misalignment causes error in period estimation, the basic circuit structure pattern can not be extracted. Hence, a defect free reference image for registration with the defect image cannot be generated. Therefore, for this case the technique fails in defecting defects of any size.

4.6.1.4 Nuisance Defects

Tested wafers normally contain nuisance defects along with the actual defects. These false alarms are normally small and have high contrast. The number of these false alarms limits the performance of the local differencing method. If no false alarms exist in the image, this method is 100% successful in locating defects and generating a defect map for all the defects listed above. On the other hand, the presence of too many false alarms will cause the local differencing method to fail in detecting defects, in particular, defects of type 2 and 3. To study the performance of the local method in the presence of these false alarms, the four following cases are examined.

Case 1 Defect types 1 and 4. Large and high contrast defects.
In this case false alarms can be removed by morphological filtering operators because the defects are much larger than false alarms. For this case
the local method is always successful in locating and isolating the defect.

**Case 2** Defect types 2 and 4. Small and high contrast defects.
Blocks containing false alarms are detected as defective blocks in addition to the true defective blocks. False alarms cannot be removed from the defect map if their size is greater than the defect size. For this case, the local method is always successful in locating the defect, but false alarms may also be generated.

**Case 3** Defect type 3. Large and low contrast defects.
In addition to the defect, if the image contains more than two false alarms, the algorithm fails in detecting the defect because false alarms have higher contrast than the defect. However, if the image contains two or less false alarms, defects can be located and false alarms are removed from the defect map. For this case, the local method is not always successful in locating the defect.

**Case 4** Defect type 3. Small and low contrast defects.
For this case, if any false alarm is present in the image, the local method will not succeed in locating the defect. Nuisance defects are more visible than such defects.
Figure 4.8 shows the cutoffs in defect contrast and area for successful detection by this method. The data used to generate this graph were taken directly from the set of test images used in this thesis.

Figure 4.8 The Local Method Performance As A Function Of Defect Contrast And Area.

- $x$: the absolute difference in gray level between defect contrast and background.
- $y$: defect area in pixels.
4.6.1.5 Window Size

In order to detect small defects, 0.023% of the image area (the area of the smallest defect in the inspected images), a 4 x 4 window size was used. Increasing the window size to 8 x 8 will reduce the effect of false alarms, but will also decrease algorithm sensitivity to small defects. For defect detection it would be optimum to choose a window having the defect’s expected size. For the set of testing images used herein, a 4 x 4 window is proven to be suitable. If prior information exists about defect size, a different window size may be chosen. Nevertheless, for large sized defects (more than 100 pixels) the global approach which simply subtracts the original defective image from the defect free reference image will result in defect detection. Then, false alarms found in the defect map are removed by morphological filtering operations.

The table below summarizes the relationship between window size and the area of the smallest detectable defect.

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>4x4</td>
<td>15</td>
</tr>
<tr>
<td>8x8</td>
<td>40</td>
</tr>
<tr>
<td>16x16</td>
<td>50</td>
</tr>
<tr>
<td>32x32</td>
<td>100</td>
</tr>
</tbody>
</table>

Table 4.6 Window Size Effect

key:

a- Window size (in pixels).
b- Defect area (in pixels).
Because the average absolute difference in each window is used to detect defective blocks, increasing window size (the number of pixels averaged) will reduce the sensitivity of the local method to small defects. For example, if the window size is increased to 8 x 8 pixels, defects of area less than 40 pixels cannot be detected.

4.6.1.6 Defect Detection Results Using The Local Method

The table below summarizes the test results for the local method with a 4 x 4 window size:

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
<th>g</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>376</td>
<td>60</td>
<td>4x4</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
<td>396</td>
<td>60</td>
<td>4x4</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>1</td>
<td>16</td>
<td>60</td>
<td>3x3</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>3</td>
<td>350</td>
<td>60</td>
<td>4x4</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>257</td>
<td>76</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>46</td>
<td>60</td>
<td>4x4</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>269</td>
<td>60</td>
<td>4x4</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>1</td>
<td>144</td>
<td>60</td>
<td>4x4</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>1</td>
<td>34</td>
<td>60</td>
<td>4x4</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>25</td>
<td>60</td>
<td>3x3</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1</td>
<td>16</td>
<td>60</td>
<td>2x2</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>1</td>
<td>50</td>
<td>60</td>
<td>3x3</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>49</td>
<td>60</td>
<td>4x4</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>1</td>
<td>102</td>
<td>50</td>
<td>4x4</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>1</td>
<td>32</td>
<td>50</td>
<td>3x3</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>2</td>
<td>150</td>
<td>50</td>
<td>4x4</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 4.7 Defect Detection Results By The Local Method.

Keys:

a - Defect type.
b - Image number of this defect type.
c - Number of defects of this type in the circuit.
d - Defect area (in pixels).
e - Threshold value.
f - Size of morphological filter (in pixels).
g - Number of false alarms in the final defect map.

Defect detection is successful if the number of false alarms in the final defect map is zero.

4.6.1.7 False Alarm Rate

False alarm rate is either measured by the number of false alarms located per inspected image or by the ratio of the area of these false alarms to the area of the inspected image [Dom, 1988]. In this case, false alarm rate is measured as the number of located false defects per inspected image. For the local method, the false alarm rates before employing the morphological filtering and after employing morphological filtering are 0.33 and 0.067 respectively, based on testing 20 images.

4.6.1.8 Probability Of Defect Detection Using The Local Method

In the local method, the average absolute difference between the original image and the defect free image is computed for each window. This value is that window's contrast relative to background. Consider the contrast of all windows in the image to be the image contrast relative to background. Assume image contrast to be the random
variable $C$ which has the normal or gaussian distribution, i.e. $C = N(\mu_C, \sigma_C)$. In addition, assume the contrast distribution of the defect in the image to be the random variable $D$ which also has the normal distribution, $D = N(\mu_d, \sigma_d)$. Also assume $\mu_d > \mu_c$. The density function of the random variable $C$ is

$$f_c(GL) = \frac{1}{\sigma_c/2\pi} e^{-\frac{1}{2} \left( \frac{(GL - \mu_c)^2}{\sigma_c^2} \right)} \quad (1)$$

The density function of the random variable $D$ is

$$f_d(GL) = \frac{1}{\sigma_d/2\pi} e^{-\frac{1}{2} \left( \frac{(GL - \mu_d)^2}{\sigma_d^2} \right)} \quad (2)$$

See Fig. 4.9 for illustration.

Figure 4.9 Illustration Of The Probability Computation For Defect Detection.
The value \( t \) in Fig 4.9 is used as the cutoff criterion to isolate the defective windows from the defect free windows. The area \( A \) in Fig. 4.9 is the probability of having a defect in the image not detected. The area \( B \) is the probability of false alarm detection. The sum of these two areas gives the total error probability. If a given pixel has the same probability of being defective or not defective, then the total probability error is minimal at the point \( t \) where \( f_c(GL) = f_d(GL) \). [Schowengerdt, 1983].

Solving for \( t \), we get:

\[
t = \frac{\mu_d^2 - \mu_c^2}{2(\mu_d - \mu_c)} \quad \text{if} \quad \sigma_d = \sigma_c
\]

or

\[
t = \frac{\sigma_d^2 \mu_c - \sigma_c^2 \mu_d \pm \sigma_d \sigma_c [(\mu_d - \mu_c)^2 + 2(\sigma_d^2 - \sigma_c^2)(\ln \sigma_d - \ln \sigma_c)]^{1/2}}{2(\sigma_d^2 - \sigma_c^2)} \quad \text{if} \quad \sigma_d \neq \sigma_c
\]

Then the probability of defect detection is

\[
P(D > t) = \int_t^\infty f_d(GL) \ dGL, \quad \text{(4)}
\]

\[
= 1 - F_d(t),
\]

\[
= 1 - \frac{1}{\sigma_d/2\pi} \int_{-\infty}^t e^{-(GL - \mu_d)^2/2\sigma_d^2} \ dGL.
\]

The integral above cannot be evaluated by elementary
methods, but can be represented in terms of the integral (Fig 4.10)

\[ \text{Erf}(z) = \frac{1}{\sqrt{2\pi}} \int_0^z -\frac{u^2}{2} e^u \, du, \quad (5) \]

\[ = \Phi(z) - \frac{1}{2}. \]

![Graph showing the distribution function \( \Phi(z) \) of the normal distribution with mean 0 and variance 1.](image)

**Figure 4.10** Distribution Function \( \Phi(z) \) Of The Normal Distribution With Mean 0 And Variance 1.

Where \( \Phi(z) \) is defined as

\[ \Phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^z -\frac{u^2}{2} e^u \, du, \quad (6) \]

which is the distribution function of the normal distribution with mean 0 and variance 1. If we set \( (t - \mu)/\sigma = u \), then
\[ F(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{(t-\mu)/\sigma} e^{-u^2/2} \, du, \]  

(7)

\[ F(t) = \Phi\left(\frac{t - \mu}{\sigma}\right) \]  

(8)

Therefore,

\[ P(D > t) = 1 - \Phi\left(\frac{t - \mu_D}{\sigma_D}\right) \]  

(9)

Since \( \Phi(-t) = 1 - \Phi(t) \), then

\[ P(D > t) = \Phi\left(\frac{\mu_D - t}{\sigma_D}\right). \]  

(10)

The probability of detecting false alarms is defined as:

\[ P(C > t) = 1 - F_C(t) \]

\[ = 1 - \frac{1}{\sigma_C/2\pi} \int_{-\infty}^{t} e^{-\frac{(GL - \mu_C)^2}{2\sigma_C^2}} \, dGL, \]

\[ = 1 - \Phi\left(\frac{t - \mu_C}{\sigma_C}\right). \]  

(11)

Example:

For the set of images in this study, \( C = N(47.62, 13.87) \) and \( D = N(99.62, 28.93) \).

From 3, the optimal cutoff value \( t = 69.71 \).

From 10, the probability of defect detection,

\[ P(D > t) = \Phi\left(\frac{99.62 - 69.71}{28.93}\right) = 0.8508 \]
From 11, the probability of false alarm detection,
\[ P(C > t) = 1 - \Phi\left(\frac{69.71 - 47.62}{13.87}\right), \]
\[ = 1 - 0.9411 = 0.0559. \]

The computed probability of defect detection (0.8508) is low. A lower cutoff value can be used to improve the probability of defect detection at the cost of false alarm detection probability. However, false alarms can be removed from the defect map using morphological filtering operations. We have found a good value of \( t \) to be \( \mu_C + \sigma_C \).

Thus \( t = 47.62 + 13.87 = 61.49 \).

Then, the probability of defect detection becomes,
\[ P(D > t) = \Phi\left(\frac{99.62 - 61.49}{28.93}\right), \]
\[ = 0.9066. \]

The probability of false alarm detection,
\[ P(C > t) = 1 - \Phi\left(\frac{61.49 - 47.62}{13.87}\right), \]
\[ = 1 - 0.8413 = 0.1587. \]

Using \( t = \mu_C + \sigma_C \) as the cutoff value, then (10) becomes
\[ P(D > t) = \Phi\left(\frac{\mu_d - (\mu_C + \sigma_C)}{\sigma_d}\right). \quad (12) \]

The probability of defect detection is a function of \( \mu_d \).
μ_C, σ_d and σ_d. The higher the defect contrast relative to image contrast, the higher the defect detection probability.

4.6.1.8.1 Nonuniform Illumination

Assume the inspected image contains nonuniform illumination and this nonuniform illumination is weighted left to right according to the following equation (Fig. 4.11):

\[
f(m) = \frac{(a - 1)}{w} \cdot m + 1. \tag{13}
\]

where

- \( a \) is a real number \( \leq 1.0 \).
- \( m \) is pixel position in the horizontal direction.
- \( w \) is image length in the horizontal direction.

The change in image contrast, due to nonuniform illumination, at each pixel location is computed by the following equation:
\[ d(m) = \left| GL \left( \frac{a - 1}{w} m + 1 \right) - \left( \frac{a + 1}{2} \right) GL \right| \]  

where

\[ d(m) \] is the absolute difference at pixel location \( m \) due to nonuniform illumination (in gray level).

\[ GL \] is the gray level value at that pixel location.

Figure 4.12 describes the shape of \( d(m) \). Alternatively,

\[ d(m) = \left( |m - \frac{w}{2}| \right) \left( \frac{1 - a}{2} \right) \frac{GL}{w/2} \]  

(15)

Figure 4.12 Illustration Of Equation (15).

If image contrast without nonuniform illumination is the random variable \( C \), the effect of nonuniform illumination on this random variable distribution is defined by the following equation:

\[ K = C + d(m). \]  

(16)

where

\[ K \] is image contrast distribution with nonuniform illumination.

\[ C \] is image contrast distribution with uniform illumination.

\[ d(m) \] is the change in image contrast due to nonuniform illumination, at pixel location \( m \).
The random variable $K$ will have the normal distribution with
\[ \mu_K = E(D) = E(C + d(m)), \]
\[ = \mu_C + d(m), \text{ and} \]
\[ \sigma_K = E((C + d(m) - \mu_C - d(m))^2) \]
\[ = \sigma_C \] (17) (18)

Let the cutoff value $t = \mu_K + \sigma_K$, to get a high defect detection probability, then from (10)
\[ P(D > \mu_K + \sigma_K) = \Phi\left(\frac{\mu_d - (\mu_K + \sigma_K)}{\sigma_d}\right) \]
\[ = \Phi\left(\frac{\mu_d - (\mu_C + d(m) + \sigma_C)}{\sigma_d}\right) \] (19)

Because $d(m) \geq 0$ for $a \leq 1.0$, nonuniform illumination reduces the probability of defect detection, while the probability of false alarm detection stays the same (0.1587).

4.6.1.8.2 Nuisance Defects

The effect of nuisance defects on the image contrast ($C$) is defined by the following:

Let $\alpha$ represents nuisance defect density per unit area. Then
\[ \alpha = \frac{n}{w^2} \] (20)

where
- $n$ is the number of nuisance defects in the image.
- $w$ is the image dimension in both directions.
Assume the nuisance defects area distribution to be gaussian and represented by the random variable \( S = N(\mu_s, \sigma_s) \). Also assume the nuisance defect contrast distribution to be gaussian and represented by the random variable \( G = N(\mu_g, \sigma_g) \).

In addition, assume the nuisance defects are uniformly distributed all over the image. Let the image contrast distribution in the presence of nuisance defects be the random variable \( H \).

The following equation shows the relationship between \( C \), \( H \), and the nuisance defects density, area, and contrast:

\[
H = C + \alpha \mu_s G
\]  

(21)

Because the sum of two normal random variables is a normal random variable, the random variable \( H \) will be normal with 

\[
\mu_h = \mu_c c + \alpha \mu_s \mu_g, \text{ and}
\]

\[
\sigma_h^2 = \sigma_c^2 + \alpha^2 \mu_s^2 \sigma_g^2.
\]

Let the cutoff value \( t = \mu_h + \sigma_h \), to get a high defect detection probability, then from (10)

\[
P(D > \mu_h + \sigma_h) = \Phi\left(\frac{\mu_d - (\mu_h + \sigma_h)}{\sigma_d}\right)
\]

(22)

\[
= \Phi\left(\frac{\mu_d - (\mu_c + \alpha \mu_s \mu_g + [\sigma_c^2 + \alpha^2 \mu_s^2 \sigma_g^2]^{\frac{1}{2}})}{\sigma_d}\right)
\]

Thus, the larger the nuisance defects density \( (\alpha) \), contrast \( (G) \), or area \( (S) \) the smaller the defect detection probability.
4.6.2 The Global Method Performance

The same set of images used for testing the local method for defect detection was used for testing the global method for defect detection. The following tables summarizes the effect of error in period estimation and of nonuniform illumination on this approach for defect detection:

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>15</td>
<td>4</td>
<td>3x3</td>
</tr>
<tr>
<td>1</td>
<td>70</td>
<td>40</td>
<td>5x5</td>
</tr>
<tr>
<td>2</td>
<td>150</td>
<td>65</td>
<td>6x6</td>
</tr>
<tr>
<td>3</td>
<td>Defect free image is too distorted. No defect can be detected.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.8 The Relationship Between Defect Area And The Estimated Period Accuracy.

Key

a - Error in periodicity calculation (in pixels).
b - Minimum defect area that can be detected.
c - Maximum false alarm area in defect map.
d - Size of morphological operator (in pixels).

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>15</td>
<td>4</td>
<td>3x3</td>
</tr>
<tr>
<td>16</td>
<td>35</td>
<td>10</td>
<td>3x3</td>
</tr>
<tr>
<td>32</td>
<td>70</td>
<td>50</td>
<td>5x5</td>
</tr>
<tr>
<td>48</td>
<td>160</td>
<td>120</td>
<td>9x9</td>
</tr>
<tr>
<td>64</td>
<td>250</td>
<td>180</td>
<td>10x10</td>
</tr>
<tr>
<td>80</td>
<td>Illumination is too great. No defect can be detected.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.9 Nonuniform Illumination Effect.

Key

a - Maximum background gray level variation.
b - Minimum defect size that can be detected.
c - maximum false alarm size in defect map.
d - Size of morphological operator (in pixels).

The table below summarizes the test results of the global method for defect detection:

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
<th>g</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>376</td>
<td>60</td>
<td>4x4</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
<td>396</td>
<td>60</td>
<td>4x4</td>
<td>10</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>1</td>
<td>16</td>
<td>60</td>
<td>3x3</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>3</td>
<td>350</td>
<td>60</td>
<td>4x4</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>46</td>
<td>60</td>
<td>4x4</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>269</td>
<td>60</td>
<td>4x4</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>1</td>
<td>144</td>
<td>60</td>
<td>4x4</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>1</td>
<td>34</td>
<td>60</td>
<td>4x4</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>25</td>
<td>60</td>
<td>3x3</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1</td>
<td>16</td>
<td>60</td>
<td>2x2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>1</td>
<td>50</td>
<td>60</td>
<td>3x3</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>49</td>
<td>60</td>
<td>4x4</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>1</td>
<td>102</td>
<td>60</td>
<td>4x4</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>1</td>
<td>32</td>
<td>60</td>
<td>3x3</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>2</td>
<td>150</td>
<td>60</td>
<td>4x4</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 4.10 Defect Detection Results By The Global Method.

Key

a - Defect type.
b - Image number of this defect type.
c - Number of defects of this type in the circuit.
d - Defect area (in pixels).
e - Threshold value.
f - Size of morphological operator (in pixels).
g - Number of false alarms in the final defect map.

Defect detection is successful if the number of false alarms in the final defect map is zero.

The false alarm rate for the global method is very large
compared with that of the local method, since, in the global method, the whole defective image is subtracted from the defect free image.

Comparing the two approaches above, the global approach generates more false alarms than the local approach, i.e. has a higher false alarm rate. In addition, the local method is less sensitive to nonuniform illumination and error in period estimation than the global method. The local method is 93% successful in detecting defects for the set of images used, while the global method is only successful in 60% of the cases. Therefore, the local method shows a better performance in defect detection than the global method.

Illustrative examples of test images and the corresponding defect maps for the two approaches above are shown in Figs. 4.13 - 4.19. In addition, the two approaches above were tested on images of a second periodic integrated circuit. Examples of defect detection results are shown in Figs. 4.20 - 4.21.

An algorithm for detecting defects in periodic circuit patterns has been developed. This method has a low false alarm rate and is able to detect defects of small area (0.023% of the area of the inspected images). The next chapter presents a defect classifier for these detected defects. An attempt is made to classify each detected defect into one of the four categories above.
Figure 4.13 Defect Detection (test 1,2).

- a - b - (a) Original image of defect type 1. (test 1)
- c - d - (c) Defect map superimposed on test 1 image.

- d - (d) Defect map superimposed on test 2 image.
Figure 4.14 Defect Detection (test 3, 4).

- a - b - (a) Original image of defect type 1. (test 3)
- c - d - (c) Defect map superimposed on test 3 image.

- (b) Original image of defect type 1. (test 4)
- (d) Defect map superimposed on test 4 image.
Figure 4.15 Defect Detection (test 5, 6).

- (a) Original image of defect type 2. (test 5)
- (b) Original image of defect type 2. (test 6)
- (c) Defect map superimposed on test 5 image.
- (d) Defect map superimposed on test 6 image.
Figure 4.16 Defect Detection (test 7,8).

- a – b – (a) Original image of defect type 2. (test 7)
- c – d – (c) Defect map superimposed on test 7 image.
- e – f – (d) Defect map superimposed on test 8 image.
Figure 4.17 Defect Detection (test 9,10).

- a - b - (a) Original image of defect type 3. (test 9)
- c - d - (b) Original image of defect type 3. (test 10)
- (c) Defect map superimposed on test 9 image.
- (d) Defect map superimposed on test 10 image.
----- Figure 4.18 Defect Detection (test 11,12).
- a - b - (a) Original image of defect type 4. (test 11)
- c - d - (c) Defect map superimposed on test 11 image.
------ (d) Defect map superimposed on test 12 image.
Figure 4.19 Defect Detection (test 13, 14).

- a - b - (a) Original image of defect type 4. (test 13)
- c - d - (b) Original image of defect type 4. (test 14)
- c - d - (c) Defect map superimposed on test 13 image.
- c - d - (d) Defect map superimposed on test 14 image.
Figure 4.20 Defect Detection (test 15, 16).

- a - b - (a) Original image of circuit type 2. (test 15)
- c - d - (b) Original image of circuit type 2. (test 16)
- (c) Defect map superimposed on test 15 image.
- (d) Defect map superimposed on test 16 image.
Figure 4.21 Defect Detection (test 17,18).
- a - b - (a) Original image of circuit type 2. (test 17)
- c - d - (b) Original image of circuit type 2. (test 18)
- (c) Defect map superimposed on test 17 image.
- (d) Defect map superimposed on test 18 image.
CHAPTER 5

DEFECT CLASSIFICATION

A rule based system for defect classification in one periodic VLSI circuit type is presented herein. This system is designed to classify the defects detected by the differencing method (chapter 4). Developing a system that is applicable for all periodic circuit types is very complicated due to each circuit's peculiar defect types and basic structure pattern. Prior knowledge of the inspected circuit structure is very crucial in building a defect classification system.

Defect classification is an application of expert system technology. The heart of the system is found in the data base and consists of a set of rules to describe different defect types and their characteristics. The advantage in using a human expert for defect classification lies in the flexibility of human response and in the power of human sensory perception in pattern recognition. However, computer based visual recognition is far from that of human. The main reason for building such systems is the great demand and
short supply of human experts. In building a defect classifier the human expert is encouraged to document his expertise in a form that is reproducible and testable by others, which is cost effective and an efficient utilization of human expertise.

5.1 Defect Description

The circuit this defect classifier is dealing with is shown in Fig. 5.1a. The circuit consists of three major visible parts: gate, trench and background. (See Fig. 5.1b.)
In addition, there are four possible defect types that can occur in this circuit. Each defect is characterized by its location, average gray level, and size.

Below is a description of each defect:

**Type 1 Defect** Fatal Bridging:
- Location: Connects two or more gates.
- Average gray level: bright, similar to the gate class gray level.
- Size: large in comparison to other defects.

**Type 2 Defect** Nonfatal Bridging:
- Location: an extension of a gate.
- Average gray level: bright, similar to gate class gray level.
- Size: smaller than the type 1 defect.
**Type 3 Defect** Void (Underlying Material Visible):
Location: on a trench.
Average gray level: similar to background gray level.
Size: variable.

**Type 4 Defect** Foreign Matter on Wafer:
Location: on a trench.
Average gray level: much darker than a normal component.
Size: variable.

Based on the above defect descriptions the following rules are used to classify defects:

**Rule 1:**
If the average gray level of the defect is similar to gate class gray level and the defect connects two or more gates, then it is a type 1 defect.

**Rule 2:**
If average gray level of the defect is similar to gate class gray level and the defect is an extension of one gate, then it is a type 2 defect.

**Rule 3:**
If the average gray level of the defect is similar to background class gray level and located on a trench, then it is a type 3 defect.
Rule 4:

If the average gray level of the defect is dark and located on a trench, then it is a type 4 defect.

No conclusion can be made about any other defect in the defect map that does not fit one of the above rules. For example, if a defect has a bright gray level and is located on a trench, it is classified as unknown.

To implement the above rules, the defect classifier first utilizes the defect's gray level to classify it into either of three groups: (a) defect 1 and 2, (b) defect 3, or (c) defect 4. If in the first group, it then checks to see if the defect is connected to one or two gates to classify it further as either a type 1 or a type 2 defect. Defects in either group b or c must be located on a trench to be classified as a type 3 or a type 4 defect, respectively. Otherwise, no conclusion is made about this defect.

5.2 System Implementation

Such classification systems are normally implemented in a high level menu driven format using a language such as PC CONSULTANT PLUS. However, this system is implemented in a low level conventional programming format because it is faster and more efficient to classify defects in an algorithm format, or low level programming, as may be shown by the following:
1) The data base is small. There are only four rules needed for defect classifications.

2) The defect classifier needs to scan the inspected image directly to extract the needed features for defect classification.

3) The amount of symbolic data needed to describe the inspected circuit and the defect is very large which complicates data transfer between the low level part (defect detection and circuit components extraction) and the high level part (defect classification).

4) This system is only applicable for only one periodic circuit type, therefore, only one algorithm is used to classify all defects.

5.3 Component Labeling

The defect map normally contains more than one defect. Since the generated defect map is binary (gl = 0 or 255), individual components are separated by giving each defect a different label (symbol). The labeled defect map is then used to extract each defect's features.

5.3.1 The Labeler Algorithm Description

For binary images, each object in the binary map consists of a set of eight connected pixels, each pixel is connected to another pixel of the same object by at least
one of its eight immediate neighbors. Consider the following definitions:

<table>
<thead>
<tr>
<th>P1</th>
<th>P2</th>
<th>P3</th>
<th>L1</th>
<th>L2</th>
<th>L3</th>
</tr>
</thead>
<tbody>
<tr>
<td>P4</td>
<td>Pc</td>
<td>L4</td>
<td>Lc</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Binary Image File          Label File

\( P_c \) = current pixel at coordinates \((x, y)\).
\( P_1 \) = pixel at coordinates \((x-1, y-1)\).
\( P_2 \) = vertical neighbor pixel at coordinates \((x, y-1)\).
\( P_3 \) = pixel at coordinates \((x+1, y-1)\).
\( P_4 \) = horizontal neighbor pixel at coordinates \((x-1, y)\).
\( \text{L}_1 \) = label at coordinates \((x-1, y-1)\), previously assigned.
\( \text{L}_2 \) = label at vertical neighbor, previously assigned.
\( \text{L}_3 \) = label at coordinates \((x+1, y+1)\), previously assigned.
\( \text{L}_4 \) = label at horizontal neighbor, previously assigned.
\( L_c \) = label at current pixel, to be assigned.

The initial label \( k \) is set to 0. The binary image is then scanned from left to right and top to bottom. The logical value for each pixel is then read and labels assigned to the pixel as follows:

1) If the logical value of the current pixel is 0 (GL = 0), go to the next pixel in the binary image.
2) If the logical value of the current pixel is 1 (GL = 255) and none of its neighboring pixels, \( \text{L}_1 \), \( \text{L}_2 \), \( \text{L}_3 \), or \( \text{L}_4 \), has a label, increment \( k \) and set \( L_c = k \). Go to the
3) If the logical value of the current pixel is 1, and only one of its neighbors, L1, L2, L3, or L4, has a label assigned to it, set Lc = Li. Go to the next pixel.

4) If the logical value of the current pixel is 1, and more than one of its neighbors have been labeled, then set Lc = Minimum(Li). Go to the next pixel.

5) Repeat the above four steps for each pixel until the entire binary image is scanned.

After performing the above algorithm on the binary image, an initial labeled map is produced. For some cases the initial labeled map may contain more than one label for the same object. (See example below.) In this case the equivalence table (defined below) is used to relabel the labeled map. Consider the following binary image:
After the initial labeling algorithm is performed, we obtain:

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0
0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 2 2 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Figure 5.3 Initial Labeled Image.

The above image contains one object. However, two different labels are assigned to this blob. Therefore, it is necessary to merge labels through the equivalence table as follows:

1) Set the content of the equivalence table to its indices:
   - Table[1] = 1
   - Table[2] = 2
   - Table[3] = 3
   - Table[max] = max.

   For the image above the equivalence table will contain only two labels, Table[1] = 1 and Table[2] = 2.

2) Scan the initial labeled image left to right and top to bottom. If two different labels are adjacent, such as label 1 and label 2 in the image above, then the two labels can be combined by replacing the contents of
Table[2] by the Table[1] value. (Now Table[1] = 1, Table[2] = 1.)

3) Scan the initial labeled image, and replace the old labels by the new labels. In the image above, each pixel has a 2 as its label, the 2 will be replaced by 1 as the new label.

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 1 1 1 1 1 1 1 0 0
0 0 0 0 1 1 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
0 1 1 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Figure 5.4 Final Labeled Image.

5.4 Images Needed By The Defect Classifier

The following set of images are needed by the defect classifier algorithm:

1) The original defective image, the same image used for defect detection. (See Fig. 5.5a.)

2) The labeled binary map of the gate class. (See Fig. 5.5b.)

The gate class is extracted from the defective image using local thresholding. Local thresholding is used to avoid nonuniform illumination effects. The following procedure is used to extract the binary gate map:
a - The image is divided into blocks each of size 32 x 32.

b - Compute the threshold value in each block:

\[ T = G_{\text{max}} - K(G_{\text{max}} - G_{\text{min}}). \]

where

\[ T \quad \text{threshold value.} \]
\[ G_{\text{max}} \quad \text{maximum gray level value in the block.} \]
\[ G_{\text{min}} \quad \text{minimum gray level value in the block.} \]
\[ K \quad \text{constant} = 0.4. \]

c - Any pixel within the block that has a gray level greater than \( T \) is given a gray level value of 255. Pixels having gray level value less than or equal to \( T \) are given a gray level value 0.

The gate class is extracted easily by thresholding because of the great contrast between the gate class and the surrounding region. The labeling algorithm is then used to give a unique label to each gate in the binary gate map.

3) The binary map of the trench class. (See Fig. 5.5c.)

In this case, the defect free image, not the original defective image, is used to extract the map of the trench class for two reasons. First, if the defect is located on a trench part, a portion of the trench part is missing in the original defective image while the trench class is complete in the defect free image. Second, the edge boundary of the trench class shows the darkest gray level in the defect free
image, hence, thresholding can be used to extract the trench class.

4) The labeled defect map. Each defect in the defect map is given a unique label. (See Fig. 5.5d.)

Figure 5.5 The Four Images Needed By The Defect Classifier.

-------- (a) Original defective image.
- a - b - (b) The labeled map of the gate class.
-------- (c) The binary map of the trench class.
- c - d - (d) The labeled defect map.
--------
5.5 The Defect Classifier

The four images above are processed simultaneously to classify the defects in the defect map as follows:

1) Scan the labeled gate map and the trench class map to compute the average gray level for the gate class, the trench class, and the background class in the original defective image.

2) The following is done for each labeled defect in the defect map: A) Use the labeled defect map to extract the defective portion in the original image and compute its average gray level.

B) Compare the defect's average gray level with that of the gate class, if they are similar then it is either a type 1, a type 2, or an unknown type defect.

When the binary map of the gate class is extracted, defects of type 1 and 2 will also be connected to a gate in the binary map because the defect and the gate class have similar gray levels. In the labeled gate map, both the defect and the gate that is connected to the defect, will make one object, hence, both will have the same label. The labeled defect map and the labeled gate map are processed to classify the defect as either a type 1 or a type 2 defect as follows:

a) Use the defect map to extract the label of the gate that is connected to the defect in the labeled gate map.
b) Remove all gates that have a different label from the gate map. (At the end of this step the gate map only contains one labeled object, the defect and the gate that is connected to the defect.)

c) Remove the defective portion from the labeled gate map. (Now the labeled gate map contains one or two gates depending on the extracted defect type.)

d) Relabel the gate map, or use vertical and horizontal projection techniques to check, how many isolated objects are in the labeled gate map. If the new labeled gate map contains two gates then it is a type 1 defect. Otherwise, it is a type 2 defect.

e) If the defect is not connected to a gate part (isolated defect), then the defect is classified as of the unknown type.

C) Compare the average gray level of the defect with that of the background class. If they are similar and the majority of the defect’s pixels belong to the trench class, then it is a type 3 defect.

D) If the average gray level of the defect is dark, and the majority of the defect’s pixels belong to the trench class, then it is a type 4 defect.

E) Otherwise, no conclusion is drawn about the defect. For example, if a defect has an average gray level similar to the background gray level, but the majority of its pixels do not belong to the trench class, then no
conclusion can be made about this defect. It is classified as unknown.

A flow diagram of the defect classifier above is shown in Fig. 5.6.
5.6 Defect Classification Results

The system correctly classified the defects in 14 out of 15 defect maps. In the case where the system failed, the defect map contained two defects, one defect has the characteristics of a type 2 defect and the other defect has the characteristics of a type 4 defect. Actually, both defects are of type 3. This difference in classification is due to the visual perception difference between the human and the computer.

One of the problems in designing a classification system is the lack of firm visual end points or conclusions. This is due to the lack of information in the data base or the incomplete set of observations which are relevant to the problem.

This defect classification system is specific to one periodic circuit type. A similar approach may be used for the classification of defects in other periodic circuits. Final conclusive remarks about the suggested methods for defect detection and classification are presented in the next chapter.

The set of images used to test the above classifier and their corresponding classification results are shown in Figs. 5.7 - 5.21 and tables 5.1 - 5.15 respectively.
Figure 5.7 Defect Classification (test 1).
- a - b - (a) Original image of defect type 1.
- a - b - (b) Label map of the gate class.
- c - d - (c) Binary map of the trench class.
- c - d - (d) Labeled defect map.

The Average Gate Gray Level = 215
The Average Trench Gray Level = 96
The Average Background Gray Level = 64

The Average Gray Level Of Defect 1 = 200
The Area Of Defect 1 = 376 pixels.

Defect 1 connects two gates.
Defect 1 is type 1 defect.

Table 5.1 Classification Results (test 1).
Figure 5.8 Defect Classification (test 2).

- a - b - (a) Original image of defect type 1.
- c - d - (b) Label map of the gate class.
- " " " " (c) Binary map of the trench class.
- " " " " (d) Labeled defect map.

<table>
<thead>
<tr>
<th>The Average Gate Gray Level</th>
<th>= 221</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Average Trench Gray Level</td>
<td>= 102</td>
</tr>
<tr>
<td>The Average Background Gray Level</td>
<td>= 76</td>
</tr>
<tr>
<td>The Average Gray Level Of Defect 1</td>
<td>= 200</td>
</tr>
<tr>
<td>The Area Of Defect 1</td>
<td>= 396 pixels.</td>
</tr>
</tbody>
</table>

Defect 1 connects two gates.
Defect 1 is type 1 defect.

Table 5.2 Classification Results (test 2).
Figure 5.9 Defect Classification (test 3).

- a - b - (a) Original image of defect type 1.
- c - d - (b) Label map of the gate class.
- c - d - (c) Binary map of the trench class.
- d - (d) Labeled defect map.

The Average Gate Gray Level = 215
The Average Trench Gray Level = 98
The Average Background Gray Level = 70

The Average Gray Level Of Defect 1 = 186
The Area Of Defect 1 = 40 pixels.

Defect 1 connects two gates.
Defect 1 is type 1 defect.

The Average Gray Level Of Defect 2 = 87
The Area Of Defect 2 = 37 pixels.

0% of defect 2 pixels belong to a trench.
No conclusion can be made about defect 2.

Table 5.3 Classification Results (test 3).
Figure 5.10 Defect Classification (test 4).

- a - b - (a) Original image of defect type 1.
- c - d - (b) Label map of the gate class.
- c - d - (c) Binary map of the trench class.
- c - d - (d) Labeled defect map.
The Average Gate Gray Level = 225
The Average Trench Gray Level = 107
The Average Background Gray Level = 76

The Average Gray Level Of Defect 1 = 196
The Area Of Defect 1 = 350 pixels.
Defect 1 is an extension of a gate.
Defect 1 is type 2 defect.

The Average Gray Level Of Defect 2 = 206
The Area Of Defect 2 = 257 pixels.
Defect 2 connects two gates.
Defect 2 is type 1 defect.

The Average Gray Level Of Defect 3 = 85
The Area Of Defect 3 = 44 pixels.
54% of defect 3 pixels belong to a trench.
Defect 3 is a type 3 defect.

The Average Gray Level Of Defect 4 = 39
The Area Of Defect 4 = 43 pixels.
55% of defect 4 pixels belong to a trench.
Defect 4 is a type 4 defect.

The Average Gray Level Of Defect 5 = 228
The Area Of Defect 5 = 76 pixels.
Defect 5 is an extension of a gate.
Defect 5 is type 2 defect.

Table 5.4 Classification Results (test 4).
Figure 5.11 Defect Classification (test 5).
- a - b - (a) Original image of defect type 2.
- c - d - (b) Label map of the gate class.
- c - d - (c) Binary map of the trench class.
- c - d - (d) Labeled defect map.

The Average Gate Gray Level = 221
The Average Trench Gray Level = 107
The Average Background Gray Level = 75

The Average Gray Level Of Defect 1 = 216
The Area Of Defect 1 = 46 pixels.

Defect 1 is an extension of a gate.
Defect 1 is type 2 defect.

Table 5.5 Classification Results (test 5).
Figure 5.12 Defect Classification (test 6).

- a - b - (a) Original image of defect type 2.
- c - d - (b) Label map of the gate class.
- c - d - (c) Binary map of the trench class.
- c - d - (d) Labeled defect map.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>The Average Gate Gray Level</td>
<td>= 222</td>
</tr>
<tr>
<td>The Average Trench Gray Level</td>
<td>= 98</td>
</tr>
<tr>
<td>The Average Background Gray Level</td>
<td>= 67</td>
</tr>
</tbody>
</table>

The Average Gray Level Of Defect 1 = 217
The Area Of Defect 1 = 269 pixels.

Defect 1 is an extension of a gate.
Defect 1 is type 2 defect.

The Average Gray Level Of Defect 2 = 209
The Area Of Defect 2 = 98 pixels.

Defect 2 is an extension of a gate.
Defect 2 is type 2 defect.

Table 5.6 Classification Results (test 6).
Figure 5.13 Defect Classification (test 7).

- a - b - (a) Original image of defect type 2.
- c - d - (b) Label map of the gate class.
- (c) Binary map of the trench class.
- (d) Labeled defect map.

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Average Gate Gray Level</td>
<td>217</td>
</tr>
<tr>
<td>The Average Trench Gray Level</td>
<td>101</td>
</tr>
<tr>
<td>The Average Background Gray Level</td>
<td>70</td>
</tr>
<tr>
<td>The Average Gray Level Of Defect 1</td>
<td>219</td>
</tr>
<tr>
<td>The Area Of Defect 1</td>
<td>144 pixels</td>
</tr>
</tbody>
</table>

Defect 1 is an extension of a gate.
Defect 1 is type 2 defect.

Table 5.7 Classification Results (test 7).
Figure 5.14 Defect Classification (test 8).
- a - b - (a) Original image of defect type 2.
- c - d - (b) Label map of the gate class.
- (c) Binary map of the trench class.
- (d) Labeled defect map.

The Average Gate Gray Level = 214
The Average Trench Gray Level = 104
The Average Background Gray Level = 76

The Average Gray Level Of Defect 1 = 213
The Area Of Defect 1 = 34 pixels.

Defect 1 is an extension of a gate.
Defect 1 is type 2 defect.

Table 5.8 Classification Results (test 8).
Figure 5.15 Defect Classification (test 9).
- a - b - (a) Original image of defect type 3.
- c - d - (b) Label map of the gate class.
- c - d - (c) Binary map of the trench class.
- c - d - (d) Labeled defect map.

The Average Gate Gray Level = 222
The Average Trench Gray Level = 102
The Average Background Gray Level = 74

The Average Gray Level Of Defect 1 = 58
The Area Of Defect 1 = 25 pixels.

100% of defect 1 pixels belong to a trench.
Defect 1 is a type 3 defect.

Table 5.9 Classification Results (test 9).
Figure 5.16 Defect Classification (test 10).
(a) Original image of defect type 3.
(b) Label map of the gate class.
(c) Binary map of the trench class.
(d) Labeled defect map.

The Average Gate Gray Level = 224
The Average Trench Gray Level = 107
The Average Background Gray Level = 71

The Average Gray Level Of Defect 1 = 79
The Area Of Defect 1 = 16 pixels.

100% of defect 1 pixels belong to a trench.
Defect 1 is a type 3 defect.

Table 5.10 Classification Results (test 10).
Figure 5.17 Defect Classification (test 11).

- a - b - (a) Original image of defect type 3.
- c - d - (b) Label map of the gate class.
- (c) Binary map of the trench class.
- (d) Labeled defect map.

| **The Average Gate Gray Level** | = 226 |
| **The Average Trench Gray Level** | = 100 |
| **The Average Background Gray Level** | = 77 |

| **The Average Gray Level Of Defect 1** | = 187 |
| **The Area Of Defect 1** | = 22 pixels. |

Defect 1 is an extension of a gate.
Defect 1 is a type 2 defect.

| **The Average Gray Level Of Defect 2** | = 21 |
| **The Area Of Defect 2** | = 41 pixels. |

95% of defect 2 pixels belong to a trench.
Defect 2 is a type 4 defect.

Table 5.11 Classification Results (test 11).
Figure 5.18 Defect Classification (test 12).
- a - b - (a) Original image of defect type 4.
- c - d - (b) Label map of the gate class.
- e - f - (c) Binary map of the trench class.
- g - h - (d) Labeled defect map.

Table 5.12 Classification Results (test 12).

<table>
<thead>
<tr>
<th></th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Average Gate Gray Level</td>
<td>223</td>
</tr>
<tr>
<td>The Average Trench Gray Level</td>
<td>99</td>
</tr>
<tr>
<td>The Average Background Gray Level</td>
<td>71</td>
</tr>
<tr>
<td>The Average Gray Level Of Defect 1</td>
<td>25</td>
</tr>
<tr>
<td>The Area Of Defect 1</td>
<td>49 pixels</td>
</tr>
</tbody>
</table>

100% of defect pixels belong to a trench.
Defect 1 is a type 4 defect.
Figure 5.19 Defect Classification (test 13).

- a - b - (a) Original image of defect type 4.
- c - d - (b) Label map of the gate class.
- c - d - (c) Binary map of the trench class.
- c - d - (d) Labeled defect map.

The Average Gate Gray Level = 227
The Average Trench Gray Level = 105
The Average Background Gray Level = 75

The Average Gray Level Of Defect 1 = 8
The Area Of Defect 1 = 102 pixels.

93% of defect 1 pixels belong to a trench.
Defect 1 is a type 4 defect.

Table 5.13 Classification Results (test 13).
Figure 5.20 Defect Classification (test 14).

(a) Original image of defect type 4.
(b) Label map of the gate class.
(c) Binary map of the trench class.
(d) Labeled defect map.

The Average Gate Gray Level = 174
The Average Trench Gray Level = 87
The Average Background Gray Level = 70

The Average Gray Level Of Defect 1 = 37
The Area Of Defect 1 = 32 pixels.

93% of defect 1 pixels belong to a trench.
Defect 1 is a type 4 defect.

Table 5.14 Classification Results (test 14).
Figure 5.21 Defect Classification (test 15).
- a - b - (a) Original image of defect type 4.
- c - d - (b) Label map of the gate class.
- (c) Binary map of the trench class.
- (d) Labeled defect map.

The Average Gate Gray Level = 191
The Average Trench Gray Level = 100
The Average Background Gray Level = 83

The Average Gray Level Of Defect 1 = 23
The Area Of Defect 1 = 150 pixels.

95% of defect 1 pixels belong to a trench. Defect 1 is a type 4 defect.

The Average Gray Level Of Defect 2 = 33
The Area Of Defect 2 = 95 pixels.

100% of defect 2 pixels belong to a trench. Defect 2 is a type 4 defect.

Table 5.15 Classification Results (test 15).
CHAPTER 6

CONCLUSIONS

Visual inspection algorithms for periodic VLSI circuits have been developed. Integrated circuit inspection consists of two parts: defect detection and defect classification. The emphasis of this thesis was on defect detection more so than on defect classification due to the following. Since the basic structure pattern of the periodic circuit is irrelevant in developing a defect detection algorithm, it is possible to build a defect detection system that is applicable for all periodic circuits. On the other hand, prior knowledge of the inspected circuit pattern and its possible defect types are very crucial in building a defect classification system. Therefore, the defect classification system will be very circuit dependent. In addition, before classifying the defects in an inspected circuit, a reliable defect detection system is needed.

Even though the developed defect detection algorithm is based on the reference comparison technique, the periodicity of the inspected circuit eliminated the need for using
another defect free (reference) image. Exploiting image periodicity, a defect free image is extracted from the defective image. Thus, the problems of wafer to wafer comparison such as contrast, registration, and storage space for the reference wafer are eliminated.

The inertia method proved to be more reliable than the power spectrum method for period measurements. Moreover, because direct image to image subtraction generates too many false alarms, a local differencing scheme is employed to generate the defect map. This defect detection method proved to be successful in detecting defects of small area (0.023% of image area) and low contrast. However, the limitations of this system are:

1. The inspected image must contain a number of defect free cells.
2. The inspected image must be periodic at angles of 0 and 90 degrees only.
3. It is sensitive to errors in period measurements and to nonuniform illumination.
4. It is very sensitive to nuisance defects, the presence of too many nuisances defects might cause the system to fail in detecting defects of low contrast and small area.

Following is a comparison between the performance of our defect detection system and the performance of other
existing systems such as the IBM P300 and the system developed by Yoda. Our system is 93% successful in detecting defects and has a 0.067 false alarm rate per inspected frame based on 20 inspected frames. The IBM P300 for example is 96% successful in detecting defects and has a 0.05 false alarm rate based on 226,488 inspected frames. The system suggested by Yoda is 95% successful in detecting defects and has a 0.5 false alarm rate based on a large, but unspecified, number of frames. Since our system was tested using small number of frames compared to the other systems, no firm conclusion can be made on whether this system is better or worse than either system.

Finally, a classifier system for the defects in one periodic circuit type was suggested. This system utilizes defect characteristics, such as contrast and location relative to the main circuit components, to classify defects. The system was 95% successful in correctly classifying 20 defects. The limitation of this system is that it is applicable to only one periodic circuit type.
REFERENCES


