AUTOMATED CLASSIFICATION OF DEFECTS IN INTEGRATED CIRCUIT MANUFACTURING

Frederick Y. Wu
Virginia H. Brecher
Paul B. Chou
Debra M. Parisi
Ravi Rao
Martin C. Sturzenbecker
IBM T.J. Watson Research Center
P.O. Box 218, Yorktown Heights, NY 10598

Abstract

Inspection of semiconductor wafers during manufacturing is a critical process which has become highly automated. Following inspection, typically a sample of defects is reviewed by a human operator for the purpose of defect classification. Classification is the assignment of a defect to one of a number of pre-defined types that are related to the cause or consequence of the defect. Such information may be used for process diagnosis and/or yield prediction.

At present, manual classification can be more time and labor intensive than the inspection itself, even when only a small sample of defects is reviewed. Furthermore, manual classification is frequently inaccurate and inconsistent. Automation of defect classification can potentially lower manufacturing costs by reducing labor, improving tool productivity, and increasing data accuracy.

In this paper we describe an implementation of Automated Defect Classification in a manufacturing environment. We will discuss the practical aspects of the installation and acceptance of the system, as well as the technical problems and solutions.

1 Introduction

Visual inspection of patterned wafers is a key part of the process of integrated circuit manufacture. Since the entire manufacturing procedure is lengthy and expensive, inspection at critical process levels is needed to ensure quality, predict yield, and detect and diagnose process problems. There exist commercial inspection tools which rapidly detect defects on patterned wafers with excellent sensitivity. This equipment is optimized for defect *detection* at high speed, and is unable to extract much information for defect *classification*. The requirements and technical challenges of automating defect classification for semiconductor devices have been described earlier [1]. The system described in this paper performs classification of defects that have already been located on a wafer by inspection equipment.

According to current practice in many semiconductor manufacturing plants, after defects have been detected and located, a subset of them are reviewed by an operator. The wafers are loaded onto a review station consisting of a microscope and automated positioning stage, which moves the wafer to each defect site using the coordinates determined by the inspection tool. At each site, the operator must first visually locate the defect, and then assign the defect to one of a number of predefined categories. The resulting data are compiled and analyzed to predict yield or diagnose problems.

Classification of defects by an operator is slow, costly, inaccurate, and inconsistent. Whereas a state-of-the-art inspection tool can inspect a 200 mm wafer in about 20 minutes, reviewing a sample of 100 defects frequently takes an operator much longer. Furthermore, the task is tedious and subjective, leading to results that are dependent on the operator as well as the degree of operator fatigue. The goal of automated defect classification (ADC) is to reduce greatly the level of operator involvement by automating the classification of routine defects. This would save operator labor, improve data consistency, and lessen the tedium of the task.

2 The Manufacturing Environment

Integrated circuit manufacturers have made huge capital investments in process equipment, including inspection and review tools. Intense competition drives manufacturers to improve tool and personnel productivity and minimize additional investment. Consequently, automation of defect classification should be done using existing tools with minimal additional hardware. Automated classification could be done on either an inspection tool or a review station. Since the cost of a review station is a small fraction of the cost of an inspection tool, most manufacturers prefer to keep the inspection equipment performing inspection, rather than slowing it down by using it for review as well. For these reasons, our ADC system is designed to be retrofittable onto review stations.

Review stations contain general purpose microscopes which are frequently used for a wide variety of purposes in a manufacturing line. An ADC system must be tolerant of the multiple uses of review equipment, and either have complete control over the equipment, or be able to diagnose an improper setup.

Since the personnel that operate review tools generally are unskilled except for specialized defect training, an ADC system must be very simple to use. The line engineers have defect knowledge but also some limited computer skills. They can learn how to set up ADC for new products and process levels as long as no programming is involved.

There are many ways in which defects can be categorized. Each manufacturer or even each manufacturing facility may have a different scheme of classification that it finds most useful. The defect classes are also likely to differ from level to level and from product to product. Under these conditions it is clear that an ADC system should be adaptable to user-defined defect categories. Certainly the performance of ADC may be expected to vary depending on how distinguishable the classes are by visual observation. Furthermore, new defect classes may arise as the process evolves or as new types of equipment are introduced. ADC should be capable of learning the new classes with a minimum of engineering effort.

The line engineers want to be informed of new or unusual defects. They also require a minimum degree of accuracy of the automatically classified defects. In most cases they prefer to have ADC refuse to classify a defect rather than to mis-classify it. Thus, ADC should be able to distinguish between straightforward defects and those that are questionable or not previously defined.

A general ADC system should be able to work with all types of semiconductor products, including both memory and logic devices. However, as a starting point we developed ADC for periodic patterns such as are found in memory devices and cache arrays of microprocessors. It is well known that the speed, sensitivity, and reliability of defect *detection* on periodic patterns is superior to that on random patterns. Since defects must be re-detected before they can be classified on a review station, it follows that defect *classification* is likely to be faster and more accurate on periodic patterns than on random patterns.

3 System Description

3.1 Overview

We constructed an ADC system for the 16 Mb DRAM manufacturing line in IBM's Burlington (Vermont) plant. The system was designed to function with existing review stations, each of which was comprised of a Reichert Polyvar SC microscope integrated with a Kensington Laboratories MS4 stage and robotic handler. A station was controlled by an IBM PS/2 running in-house review software, connected to a token-ring network. Each station already had a high-quality color CCD camera mounted on the microscope and a color video monitor. For ADC, we added a color frame grabber board into the PS/2, and attached an IBM RS/6000 workstation to the network.

The review station software was modified to perform diagnostics (described later), grab defect images, transmit images to the workstation, and receive classification results from the workstation. The ADC algorithms run on the workstation, and do not use any special hardware. The algorithms consist of four main modules, briefly described below.

3.2 Defect Detection

The detection of defects is performed by comparison of the defect image with reference patterns. During preparation of ADC for an inspection level, reference images of non-defective device regions are gathered and used to create statistical mean and standard deviation images. Another reference, called the label image, is created semi-automatically from the mean image. The label image has a label (usually a material or device structure name) assigned to each pixel.

The defect detection algorithm aligns the defect image with the reference mean image by searching for a maximum correlation coefficient. It then compares the defect image to the mean image, pixel-by-pixel. Any defect image pixel that differs from the corresponding mean image pixel by more than some factor times the standard deviation for that pixel is considered a defect pixel. In addition, each defect pixel is analyzed to determine whether it matches any of the known labels and a defect label image is created.

3.3 Defect Grouping

Having located every defect pixel, we next group connected pixels into regions and calculate basic region parameters such as area, perimeter, and centroid. The defect detection process frequently breaks up a defect into multiple disconnected regions, even though they are probably all part of the same defect. To counteract this, a second grouping is performed based on spatial proximity of regions, to form clusters. Each cluster is considered a defect.

3.4 Defect Features

The system has a large set of defect characteristics, or features, which it can compute. For example, the *size* features include the maximum, minimum, average and sum of the region areas in a cluster, as well as the area of its convex hull. Typical *shape* measures include circularity and eccentricity; color measures include the first and second moments of the red, green, and blue measures.

To minimize the computation load features are calculated only if required for accurate classification. The features fall into several broad categories: size, shape, texture, location, color and contrast.

3.5 Classification

Although we could have selected any of several classification methods for this application, several characteristics of integrated circuit defect classification led us to use *fuzzy inference* techniques [2, 3, 4, 5]. The defect category experts, line engineers and operators, frequently describe defect types in vague or imprecise terms. Different engineers and operators will occasionally disagree over a defect call, implying that there is some overlap between classes, or some uncertainty in their definitions. These conditions are handled well by fuzzy logic. Defects also are exceptional events, and may not be accurately modeled by standard statistical approaches. Finally, some defect types are very rare and, especially when applying ADC to a new product or level, it can be difficult to compile a large set of example images for each class.

Fuzzy inference is a computational architecture for encoding rulebases using fuzzy logic. Each rule associates features with a particular class by means of reference sets, or linguistic terms which are defined as fuzzy sets on the domains of feature components. For example, we may encode a rule such as "If the area of the defect is large or the number of regions is very large, then the defect is a gross defect." "Area" and "number of regions" are features, "large" and "very large" are fuzzy sets defined on the values of these features. A given value for area may be "large" to a degree which ranges from zero to one.

Rules operate, or *fire* in parallel; that is, independently of one another. The defect feature vector is compared to the fuzzy sets of each rule, and an overall degree of similarity is determined. This similarity implies a degree of membership in the class indicated in the conclusion of the rule. When all rules have fired, their outputs are combined to yield a net degree of membership of the defect in each class. This value can be considered a level of confidence that the defect belongs to that particular class, but it is not a probability distribution, which would have the constraint that the memberships in all classes must sum to one.

Defects that do not match any of the defined classes are called "unknown" and returned to the operator for manual classification following completion of all sampled defects on the wafer.

4 System Training and Testing

4.1 Manual Rule Generation

An engineer who has thorough and authoritative knowledge of the defect class definitions can use the ADC system to write a rulebase using English-like terms and syntax. In this mode, the system is used to compute features from a small set of sample defect images. The engineer defines fuzzy sets based on the feature distributions, and chooses names for the fuzzy sets, such as "HIGH" or "VERYLOW". Rules are then written using the fuzzy set names and the feature names, as in the example in Figure 1.

4.2 Automatic Rule Generation

The ADC system also has the capability to generate rules automatically if it is given a suitable set of expert-classified defect images. The automatically generated rules follow the same syntax as manually generated rules, and can therefore be analyzed or edited if necessary. More details can be found in [6].

4.3 Testing

The ADC system has a facility by which expert-classified defect images can be selected for inclusion in a testing database. The database consists of all the features extracted from all the selected defect images. An engineer can test a newly developed rulebase by using the testing database as input and observe the results in several ways. The displayed results show the accuracy of the rulebase in each class and the fraction of defects classified.

4.4 Thresholding

For every defect, the fuzzy inference procedure determines a confidence level corresponding to each class. For some defects there is no uncertainty; the confidence level is 1.0 for one class and 0.0 for all others. For other defects the highest confidence level is less than 1.0 and/or there are non-zero confidence levels in other classes.

The accuracy of classification is generally correlated with the confidence level computed by ADC. This characteristic makes it possible for the engineer to select the accuracy desired in each class by setting thresholds on the confidence levels. Another measure that is useful for this purpose is the ratio between the highest confidence level and the next highest confidence level. This thresholding capability gives the line engineer control over the tradeoff between classification accuracy and classification rate (the fraction of defects classified by ADC).

5 Manufacturing Implementation

5.1 Architecture

There were numerous issues that had to be resolved in order to get ADC to interface smoothly with existing manufacturing procedures. As mentioned earlier, our approach was to minimize the changes to the hardware and software already in the manufacturing line. The ADC software runs on a remote workstation linked to the review stations by a token-ring network. In this client-server architecture, each PS/2 that controls a review station is a potential client for the ADC server. This configuration allows efficient use of the ADC server and does not impose much additional computation load on the review station controller.

The system can be operated in two modes: the user-friendly XWindow-based engineering mode or the on-line production mode. In the engineering mode, a user with detailed knowledge of the defect classes and basic knowledge of machine vision can interactively develop, test, and update a custom solution for a selected product and inspection level. The solution thus developed, consisting of reference images, a fuzzy rulebase, and a configuration file, can be used directly in the production mode. In the production mode, the operator simply initiates an ADC session for the current review station, product, and level, and waits for the ADC results to be returned.

Figures 2 and 3 are examples of the engineering mode display, showing ADC results for defects at two levels of the 16 Mb DRAM product.

5.2 Diagnostics

Since the review station is also used for non-ADC purposes, the microscope is frequently left by the operators with settings that are not suitable for ADC. Among the parameters that may not be correct are the magnification, the illumination level, and the field and diaphragm apertures. In the ideal case (and in future systems), these settings should be under computer control so that they can be properly set during ADC initialization. Given the existing hardware constraints, we designed ADC to check for the proper settings by image analysis. As part of ADC initialization, a test image from a non-defect region of the wafer is acquired and evaluated for illumination level, illumination uniformity, scale, and orientation. If an error is detected, an appropriate message is displayed on the operator's monitor.

5.3 Data Management

Images are acquired by the review station computer and sent over the network to the ADC server. As soon as an image has been acquired the station proceeds to the next defect site and acquires the next image. In parallel, the ADC server processes each image as it is received. For each defect, it computes the confidence levels and applies the thresholds that have been set by the line engineer for the particular class, level, and product. After all defect images have been classified by the server, a file containing the results is sent back to the review station computer. The review station program then incorporates the ADC results into the defect file and presents to the operator the short list of defects to be reviewed manually. The latter defects are those which ADC could not classify with enough confidence to ensure the desired accuracy. Classifications by both ADC and operators are encoded for traceability to the source of the classification. Additionally, users with appropriate authority can override the ADC classification.

6 Performance

The ADC system has been used by the IBM Microelectronics Division Burlington 16 Mb DRAM line for about one year. It is being used at two of the major inspection levels, and extension to other levels is in progress. The Burlington line engineers determine the accuracy required in each class, and ADC is tested against those targets before it is used in the line. Typically ADC achieves those accuracy targets, which range from 80% to 100%, on the 5 or 6 most common defect classes, which constitute about 80% of all defects found. In this past year, well over 100,000 defects have been classified automatically, and the use of ADC continues to expand.

7 Acknowldgements

The authors wish to acknowledge the contributions of Dale Albrecht, Carol Boye, and Larry Wyckoff of IBM Microelectronics to the success of the ADC project.

References

[1] P. Chou, A. Rao, M. Sturzenbecker, and V. Brecher, "Automatic defect classification for integrated circuits," in *Machinve Vision Applications in Industrial Inspection*, vol. 1907, pp. 95–103, SPIE, 1993.

[2] J. C. Bezdek and S. K. Pal, eds., *Fuzzy Models for Pattern Recognition.* IEEE Press, 1992. New York.

```
fuzzy\_rule\_base SCRATCH (defuzzification=none, orgate=sum)
begin
if HULLECCEN is VERYHIGH
and SPARSENESS is HIGH
and MEANBCNTRST is LOW
and INTCNTRATIO is VERYLOW
then DEFCLASS is SC weight 1.0
```

Figure 1: An example of a rulebase.

[3] H. Ishibuchi, K. Nozaki, and H. Tanaka, "Pattern classification by distributed representation of fuzzy rules," in *IEEE International Conference on Fuzzy Systems*, pp. 643–650, March 1992. San Diego.

[4] J. Keller and D. Hunt, "Incorporating fuzzy membership functions into the perceptron algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 7, no. 6, pp. 693–699, 1985.

[5] L. Zadeh, "Outline of a new approach to the analysis of complex systems and decision processes," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 3, pp. 28–44.

[6] D.M. Parisi and M.C. Sturzenbecker, "Application of Fuzzy Inferencing Techniques to Automated Classification of Defects in Semiconductor Wafers," to be published in proceedings of Fuzzy Logic '94, September, 1994, San Diego.

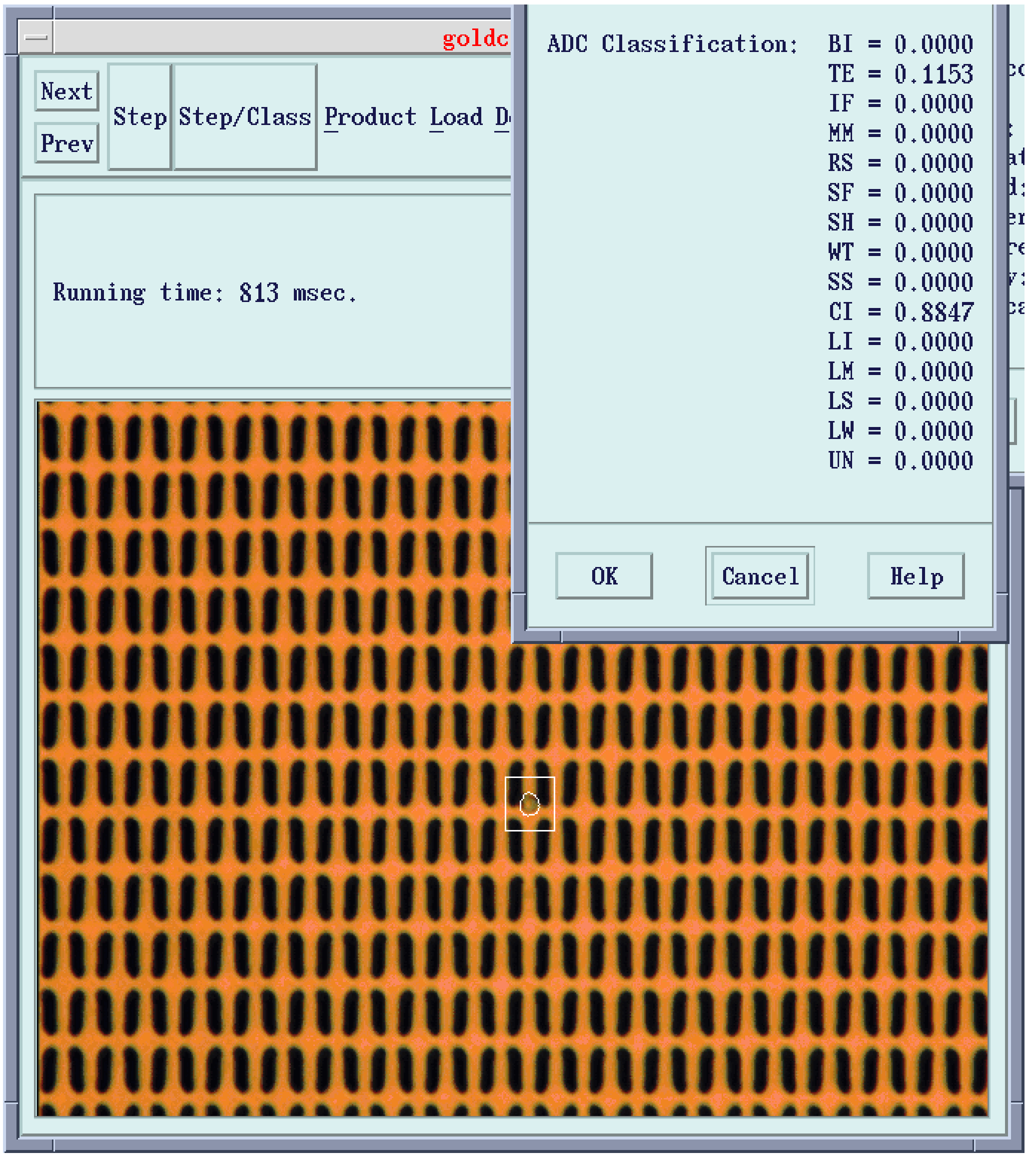

Figure 2: Example of a classified defect.

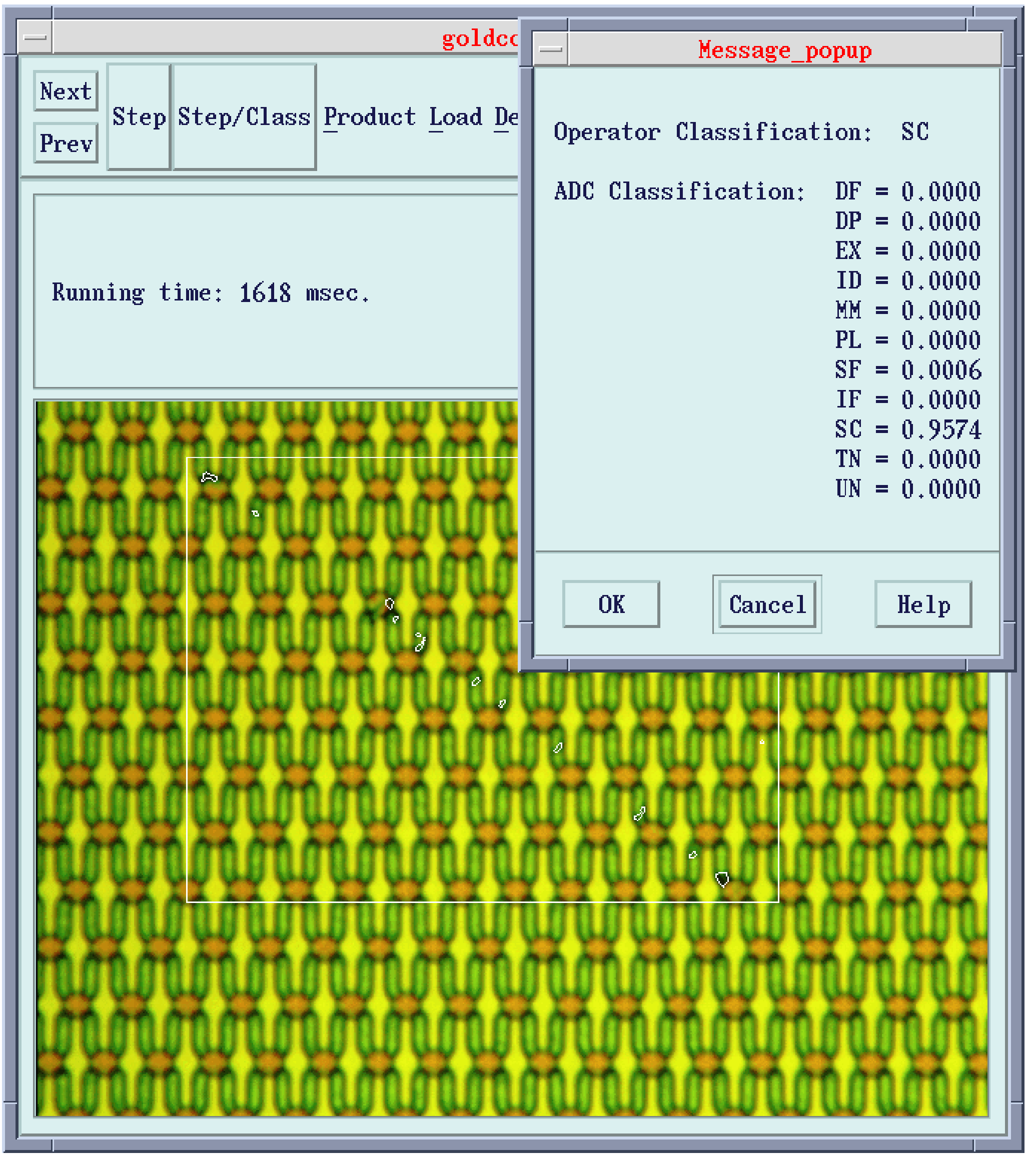

Figure 3: Example of a defect classified as SCRATCH on a more complex pattern.