An Automated Visual Inspection System for Bare Hybrid Boards

Cem Eskenazi

August 1985

Computer Vision and Robotics Laboratory
Department of Electrical Engineering
McGill University, Montréal

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of
Master of Engineering.

Abstract

Visual inspection forms an important part of the manufacturing process of hybrid circuits. Currently, such inspection is performed by human inspectors. In this thesis, a computerized automated visual inspection system is designed for the inspection of bare hybrid boards. This system detects defects on different layers of the circuits, analyzes these defects and gives descriptive as well as statistical information about them. The system consists of low level fault detecting routines and a rule-based production system. The rule-based system performs the general management of the system as well as expert analysis of defects. The system is tested on real samples of hybrid circuits and its performance is satisfactory.

Résumé

L'inspection visuelle joue un rôle important dans le procédé de production des circuits hybrides. Aujourd'hui, ce type d'inspection est accomplie par des inspecteurs humains. Cette thèse presente un système automatique d'inspection visuelle désigné pour l'inspection des plaques hybrides sans composantes. Ce système détecte des défauts aux différents niveaux des circuits, analyse ces défauts et en donne de l'information descriptive aussi bien que statistique. Le système consiste en deux niveaux: les routines à bas niveau et un système expert. Le système expert performe la gérance générale du système aussi bien que l'analyse experte des défauts. Le système est éprouvé sur des examplaires réels de circuits hybrides et montre une performance satisfaisante.

Acknowledgements

My deepest gratitude goes to Prof. M. D. Levine, my supervisor. He not only made my adaptation to a new culture and to a new university very comfortable, but was of tremendous assistance in guiding and helping me throughout this thesis research. His encouragement and constructive criticisms helped me enjoy this work and accomplish it with pride. I shall be eternally grateful to him for all his help.

All the members of our Computer Vision and Robotics Laboratory have given a lot of help and encouragement during my years at McGill. To mention the names of all the friends would be impossible here, but I am grateful to each and every one of them. My particular thanks to Pierre Parent who was our system manager during a large part of my research, and Mike 'der Mouse' Parker for whom it seems there is no unsolvable software problem.

My very special thanks to Pamela Craford for her encouragement and moral support.

Table of Contents

Chapter 1 The Hybrid Circuit Visual Inspection Problem.

1.1 Introduction

Hybrid circuits are electronic circuits which are built by combining the traditional printed circuit board design philosophy with a printing method called *thick film* technology. In hybrid circuits, one or more layers of conductors are printed on a ceramic board, separated wherever necessary by layers of dielectrics. Resistors which are printed on top of the conductors are trimmed in order to achieve the desired resistance. Discrete components such as capacitors and integrated circuits are soldered on top of all the printed layers on the circuit, which is completed by the addition of a lead frame and overglazing. See Figure 1.1 for a picture of a hybrid circuit.

The manufacturing process of the hybrid circuits involves several steps of printing, drying, firing in ovens, soldering, and other specialized steps such as laser cutting or abrasive trimming. Faults occurring during these production steps are not always detectable or locatable by electrical tests but can still be fatal to the operation of the circuit. Therefore, another method for the inspection of the circuits is necessary. In order to understand why visual inspection has been selected by today's manufacturers as well as in the system presented in this thesis, a close look at the manufacturing process of hybrid circuits is necessary. Section 2 of this chapter is a detailed description of hybrid circuits.

This thesis deals with the design and testing of an automated visual inspection system for hybrid circuits. The design of such a system involves two steps: Creating the appropriate algorithms to detect the commonly occurring defects during the manufacturing of hybrid circuits, and designing a rule-based production system which will analyze the data provided by these algorithms and will give information about the appropriate actions to take when

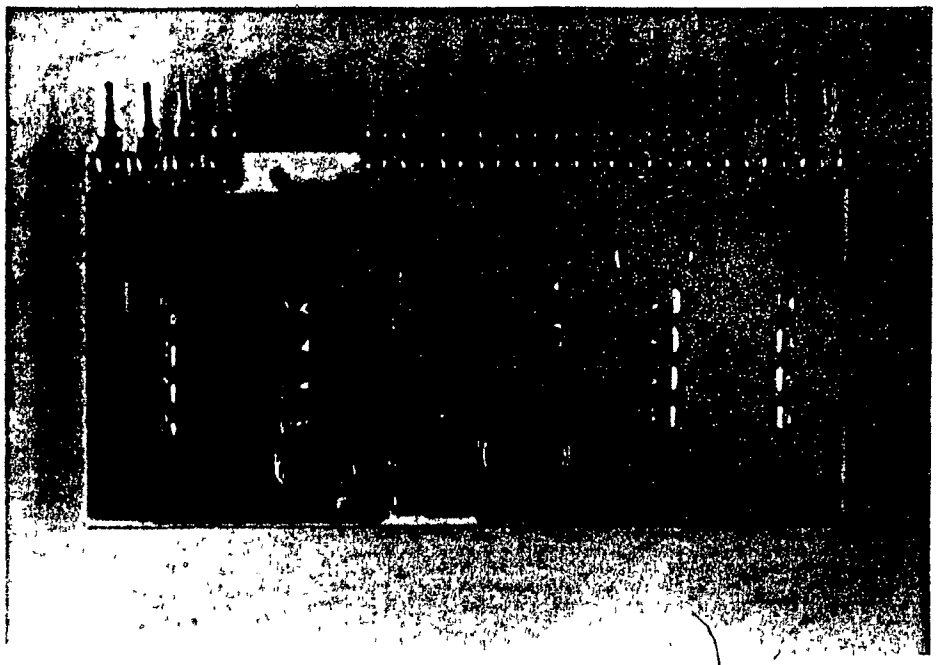


Figure 1.1 A hybrid circuit

such defects are found. Such a system has been modeled and implemented in this thesis. The system is able to analyze the two-dimensional features of hybrid circuits during their production, detect errors, and generate detailed information about them.

1.2 Description of Hybrid Circuits

1.2.1 Introduction

There are many manufacturers of hybrid circuits in today's market. They have various types of hybrid circuits, with different components, layers, and layouts. Naturally, their design rules and quality specifications also vary, but the main philosophy remains the same. Throughout this thesis, hybrid circuit design and inspection rules are inspired from the related Northern Telecom documents ([Hopkinson80a], [Hopkinson80b], [Lofthouse81], [Northern81]) and from personal conversations with Northern Telecom engineers. The automated visual inspection system is also designed to satisfy Northern Telecom's design rules and inspection criteria, but can be easily modified to fit any other set of rules.

1.2.2 How Are Hybrid Circuits Built

As mentioned earlier, hybrid circuits are made of three basic parts: *ceramic substrate,*

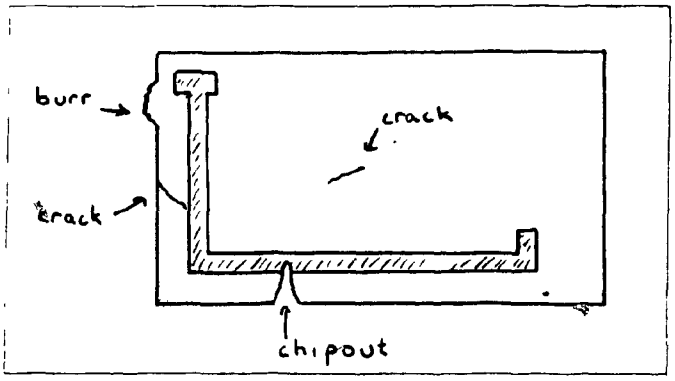


Figure 1.2 *Typical substrate defects*

printed layers, and *discrete components* The printed layers generally consist of *conductor*, *dielectric*, *resistor* and *overglaze* layers A description of the various parts of the hybrid circuit follows

1.2.2.1 Ceramic Substrates

Ceramic substrates are thin ceramic plates on which the layers are printed and the components soldered. They can be large plates on which several circuits are printed, and the plate is cut into separate circuits before the mounting of discrete components. Ceramic substrates should be clean and should not have any cracks or missing pieces that would affect the proper operation of the circuit The faults occurring on ceramic substrates which might affect the operation of the circuit depending on their size or location are the following: cracks, chipouts that reduce the width of conductors or of areas of solder pads beyond a certain limit, chipouts that damage resistors, burrs that will increase the width or length of the substrate more than a certain value or keep the lead frame attachment more than a certain distance from the substrate These faults are illustrated in Figure 1 2. Obviously, some of these can be detected before any printing occurs, whereas others can be classified as fatal or acceptable only after layers or discrete components are placed on the substrate. Of course, there is even the possibility that some of these faults will not appear at the beginning of the production process but will occur during the process, perhaps resulting from improper manipulation or accidents. Therefore, the ceramic substrate should be inspected not only at the beginning of the assembly but also during different stages and in particular when the whole circuit is assembled.

1.2.2.2 Conductor Layers

Conductor layers are the first layers printed on the substrate. As opposed to printed

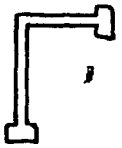
circuits where conductors are obtained by etching on a surface totally covered with a metal layer. hybrid circuit conductors are obtained by being printed with a process called thick film technology. This technology consists of first manufacturing a "screen" of the pattern that is wanted on the circuit, and then printing a paste of desired qualities on the substrate through that screen. The screen is made of a very fine steel mesh where the pattern is obtained by covering the unwanted parts of the pattern by a layer of paint that clogs the holes of the mesh. When this screen is put on top of the substrate and the material to be printed is pressed on it with a rubber edge commonly known as a *squegee*, only those parts of the screen that were not covered by paint pass the material onto the surface of the substrate, resulting into the desired pattern being printed on the substrate. Figure 1.3 summarises the thick film printing process

The pattern on the screen itself is prepared by a photographic process not unlike the one used for printed circuit boards. The patterns are designed manually. Since one screen can be used to print tens of thousands of substrates, the design of the screens constitutes a very minor part of the whole production process. The photographic films used to manufacture the screen are 'perfect' models of the circuit. In the system described in this thesis, the digitized pictures of such films are used as models of the circuit under consideration for part of the analysis performed. It is hoped that in the future the patterns for the screens will be computer generated by a CAD/CAM system. Under these circumstances the data will be directly available to the inspection system without going through the possible distortions caused by the digitizing process.

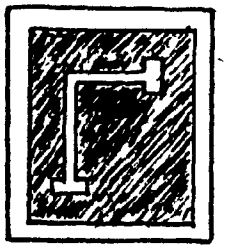
The materials that are printed using the thick film technology are pastes of the appropriate electrical qualities (e.g. conductive for conductor layers, resistive for resistors, non-conductive for dielectrics), and it is necessary to dry and fire them so that they can be fixed and hardened on the substrate. Firing is done in special ovens where the proper temperature and ventilation is maintained. Errors can occur during any of the steps of this process and therefore the circuits should be examined at every stage.

Conductors are generally printed on only one side of the substrate, but in complicated circuits, it is sometimes advantageous to print them on both sides, in particular to obtain additional connections between leads. On the main side of the circuit, there can be several layers of conductors printed on top of each other, allowing more flexibility for the design of the circuits. These layers are electrically separated by printed dielectric materials, except in locations where electrical connections between layers are wanted.

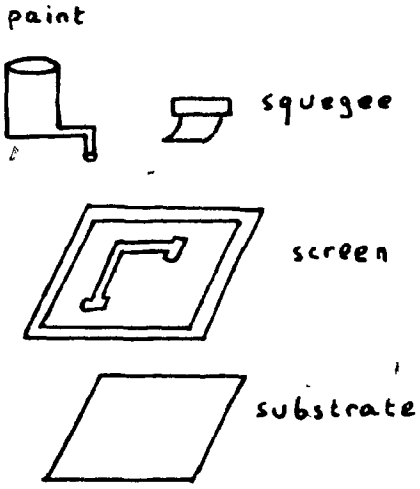
Various parts of conductor layers are meant to accomplish different tasks. *Conductor*



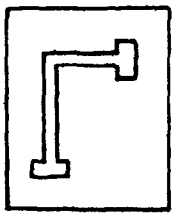
a Pattern to be printed



b Pattern is printed on a steel mesh by a photographic process



c The screen is pressed onto the substrate, paint is smeared onto it with a squegee



d Substrate has pattern printed on it. The paint dries and hardens after firing.

Figure 1.3 Thick film printing process

paths are generally long and narrow parts of the circuit which electrically connect two points in the circuit. *Solder pads* are rectangular shapes of a certain minimum area on which discrete components are soldered. Finally, *probe pads* look similar to solder pads but they are used as conductive points on the circuit for electrical tests. From their functions it can be easily understood that solder pads and probe pads should never be covered by non-conductive materials, although conductor paths should be protected by an overglaze layer.

Faults that can occur in conductor layers are the ones that can affect their electrical

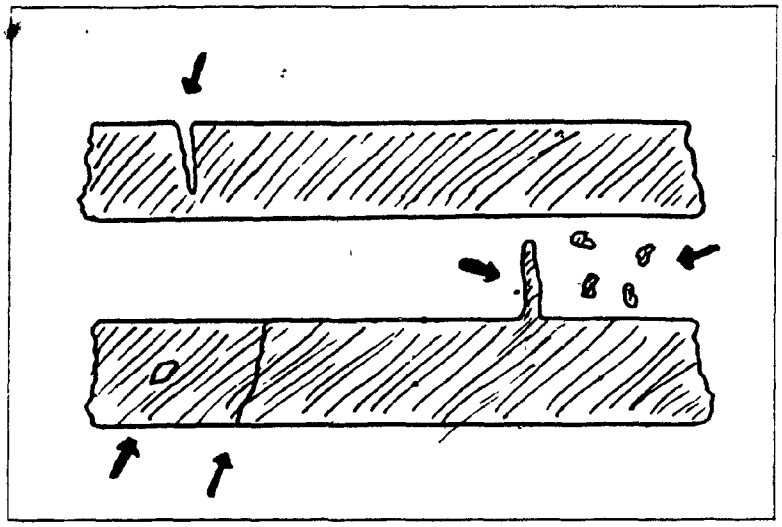


Figure 1.4 Typical conductor defects

properties, causing open or short circuits. Therefore reduced widths of conductor paths, reduced areas of solder or probe pads, voids or porosities are acceptable only up to a certain point since an excess of any of these would cause an open circuit. Reduced distances between two conductors, spreading, splattering or extensions of the conductive material might cause short circuits and therefore should not exceed a certain threshold either. Figure 1.4 gives examples of errors which might occur on conductor layers.

1.2.2.3 Dielectric Layers

Dielectric layers are used as insulation between layers of conductors where they are supposed to be electrically insulated from each other. They are printed the same way as the conductor layers and require the same cycles of drying and firing. To insure proper insulation between conductors, and since testing the reliability of the dielectric layers is very difficult at the time of production, two identical layers are printed at the same locations.

Dielectric materials which are supposed to insulate two conductor paths should overlap the bottom conductor at the crossover location with a certain allowed tolerance so that the risks of the two layers touching each other are minimized. Also, dielectric layers shouldn't have any cracks, chipouts, or pinholes that would create any risk of connection between conductors that are supposed to be insulated. Finally, smearing or sputtering of the dielectric material is allowed only when this does not affect the conductivity of parts of the circuit such as solder or probe pads. Figure 1.5 gives several examples of faults that can occur on dielectric layers.

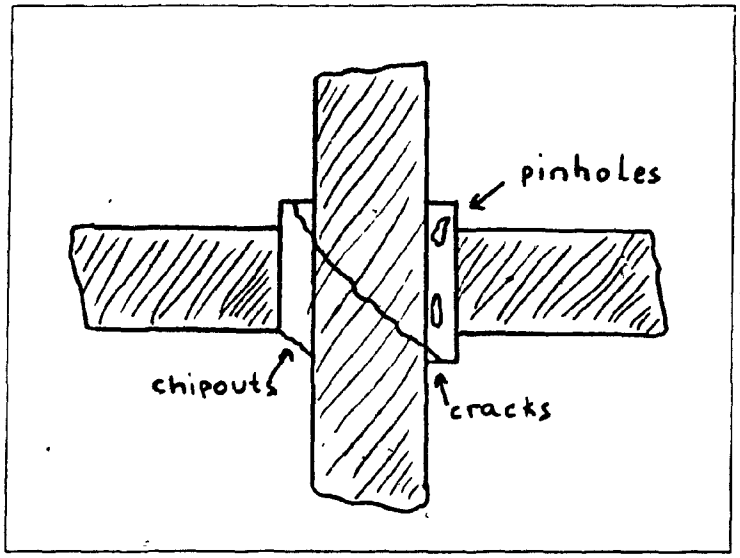


Figure 1.5 Typical dielectric defects

1.2.2.4 Resistors

One of the big advantages of hybrid circuits over printed circuit boards is that resistances (sometimes called resistivities) can be created by printing them on the substrate instead of soldering them on as discrete components. These resistances are printed with the same methods as the previous layers, using resistive materials The resistors generally are rectangularly shaped areas touching two or more conductors, forming the desired resistance between them. Voids, cracks, blisters, peeling and other imperfections of the resistive material can alter its resistive characteristics and should be allowed only up to certain limits. Also, the overlap area of the resistors with the conductors they are supposed to be electrically connected to should not be less than certain limits, and the distance between the resistors and other parts of the circuit should exceed certain limits.

Resistor Trimming. Obviously it is impossible to accurately obtain the desired resistor values at the time of printing. The printed material can expand slightly causing a greater area and therefore lower resistance. The layer might be too thin or too thick, affecting the resistivity of the material. The overlap between the resistor and conductors can be smaller than expected, causing higher resistance. Since resistance tolerance in some electrical circuits can be very small, a method should be used to 'tune' the resistance to its desired value. This is accomplished by *trimming*.

Trimming the resistor means making one or several cuts in the resistor so that its area is effectively diminished, thereby increasing its resistance. The trimming is done until

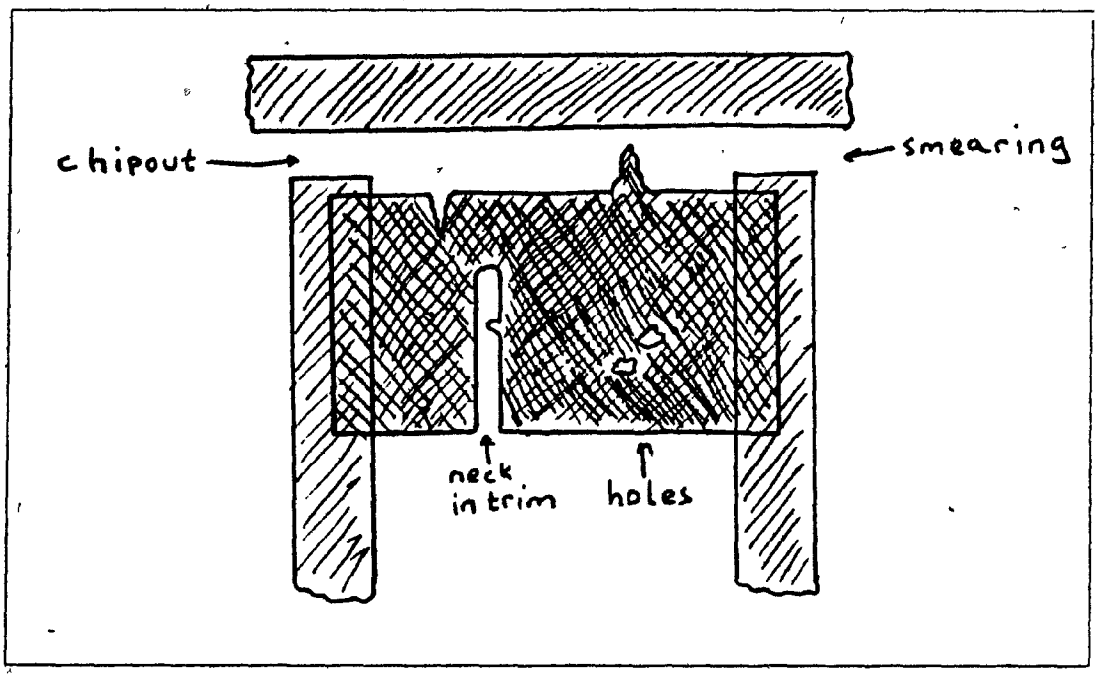


Figure 1.6 Typical resistor defects.

the desired resistance is achieved. Two methods of trimming are currently used. One of them is called air trimming or abrasive trimming. Air trimming consists of blowing sand or similar hard materials onto the resistor, causing the erosion of part of the resistor. The other method, laser trimming, consists of cutting lines into the resistor with a laser beam. This method is certainly cleaner, easier to use with the proper equipment, and is more accurate.

It is expected that trimming itself may cause some problems during the manufacturing of hybrid circuits. Basically three kinds of faults might appear once the trimming is done. One of them is dirt (resistive or conductive) stuck in a trim, sometimes referred to as 'detritus in the kerf'. This will obviously change the value of the resistor. The other problem is misaligned cuts which may partially impinge on the conductors. The third kind of problem are very narrow strips of resistive material left on either end or between trims. In this last case, although the resistor seems to have the proper value during the production, relatively high currents might burn the narrow strips, therefore causing the value of the resistor to change. Figure 1.6 shows an example of resistor with various faults indicated on it.

1.2.2.5 Overglaze Layers

The last layer that gets printed on a hybrid circuit before the soldering of discrete components is the *overglaze layer*. This one is a transparent, coloured, non-conductive layer which serves to protect the circuit from any physical damage in its future use. Of course, since the discrete components are still to be mounted on the circuits and electrical tests are still to be performed, solder pads and probe pads must be left uncovered. Apart from this important criterion, occasional imperfections that might occur on the overglaze layer are not fatal to the operation of the circuit since they do not have any electrical effects and therefore they are acceptable.

1.2.2.6 Lead Frame Attachments and Discrete Components

Once the layers are printed on the substrate with the thick film technology, the discrete components and lead frame attachments can be soldered onto it. A mask similar to the ones used for the printing of the two-dimensional patterns of the circuit is created for the solder pads, and a solder paste is printed on the circuit at the appropriate pads. Discrete components such as capacitors, integrated circuits, other semi-conductors, relays, coils and the like are placed on top of the solder paste, and the circuit is once again fired. This process hardens the conductive solder paste which fixes the components in their final locations. Soldering of the discrete components is also a source of several faults in the final product, such as misplaced, misaligned, and missing components; solder joint quality may also be inspected ([Merrill84]). However, the faults related to these three-dimensional discrete components placed on the hybrid circuits are outside the scope of the research reported here and are being considered by other researchers in our laboratory.

1.3 Visual Inspection of Hybrid Circuits

1.3.1 Comparison of Electrical and Visual Inspections

There are two practical ways to test any kind of electronic circuit. One of them is electrical, the other is visual. Tactile testing could also be used for testing the existence of three dimensional components on the circuit, but the amount of information this kind of test would be able to give would be too little to make the technique useful.

Electrical tests are very reliable for the inspection of the printed layers of a hybrid circuit: relevant inputs and expected outputs can be easily determined. These tests will

most certainly detect any defects which will interfere with the proper operation of the circuit under consideration, such as open or closed circuits. They are easily computerized and can be applied in very short periods of time. These tests are used in the production of hybrid circuits at various stages and are valuable in detecting most of the defective circuits.

Despite its advantages, electrical testing has two major shortcomings. One of them is that an electrical test does not give enough information to determine the cause of the problem. For example, consider a test of the resistance between two points in a circuit and suppose the result of the test was infinite resistance, i.e. an open circuit, contrary to what was to be expected. This certainly gives an indication of a fault in the circuit, but does not give any information about the cause of the problem. In this particular case, the cause might be an open on a conductor line, in which case the location is still to be determined, a missing or misplaced component on the path, a missing probe pad, or a pad accidentally covered with dielectric material, such as the overglaze layer.

The other major drawback of the electrical test is that it cannot detect weaknesses of the circuit which after a certain period of operation might become fatal faults. To quote the above example again, if a conductor is too narrow at one point, or if a dielectric between two conductors has a crack which in the future might conduct, or if a resistor is not properly aligned with the conductors it is supposed to touch and the contact point is too narrow for the currents that will go through them, an electrical test will not be able to detect any of these.

Visual inspection overcomes all the drawbacks of electrical inspection. With visual inspection, faults can be located and specified. Even the reasons for an occurring fault can be deduced by visually inspecting a hybrid circuit. Moreover, imperfections which do not cause a malfunction of the circuit at the time of production but which can become fatal to its operation after a certain time of operation can also be detected by visual inspection. To understand this more clearly, let us examine Table 1 which summarizes the possible faults that may occur on the substrate and the printed layers of a hybrid circuit.

As it can be seen in Table 1, most of the defects do not affect the immediate operation of the circuit. They are considered as defects only because they constitute a hazard for the future. Also, the faults that do affect the performance of the circuit can only be detected, but not described, by electrical tests. A visual inspection can describe their location, their size or extent, and probably the reason for their occurrence. Besides, a visual inspection can be performed at every stage of the production, including before and after drying and firing, thereby providing much more detailed information about defects, whereas the majority of

Substrate	cracks chipouts burrs
Conductor	reduced widths reduced areas voids and porosities reduced distances between conductors spreading, splattering, extensions
Dielectric	reduced areas cracks, chipouts, pinholes smearing, sputtering
Resistor	voids cracks, blisterings, peelings reduced overlap area with conductors reduced distance to other parts of circuit dirt in trim misplaced trim narrow strips of resistive material left by trims
Overglaze	covered pads

Table 1. Faults that can occur during the production of the bare board

the electrical tests can be performed only after a certain number of layers have been printed.

1.3.2 Human Visual Inspection

Visual inspection is indeed the method used today by hybrid circuit manufacturers. At every stage of the production, all the boards or usually just a percentage of them are visually examined under a microscope by inspectors and the errors are recorded. Frequent occurrences of a certain defect prompts the inspectors to flag the manufacturing process. Random occurrences of defects result in the repair of the boards or to their rejection, according to how much time and effort would be involved in repairing the circuit.

Figure 1.7 illustrates the assembly process for hybrid circuits. The two dimensional visual inspections needed during this process are indicated in the diagram.

1.4 Rationale for Automated Inspection

The visual inspection of hybrid circuits is a very repetitive process. It involves inspecting thousands of boards up to tens of times for defects whose dimensions are in the order of fractions of millimeters, microscopic in the case of the laser trim. Any of these defects can be missed by electrical tests but can be fatal to the operation of the board, which is inadmissible for industrial standards. Because of the repetitive and boring nature of hybrid

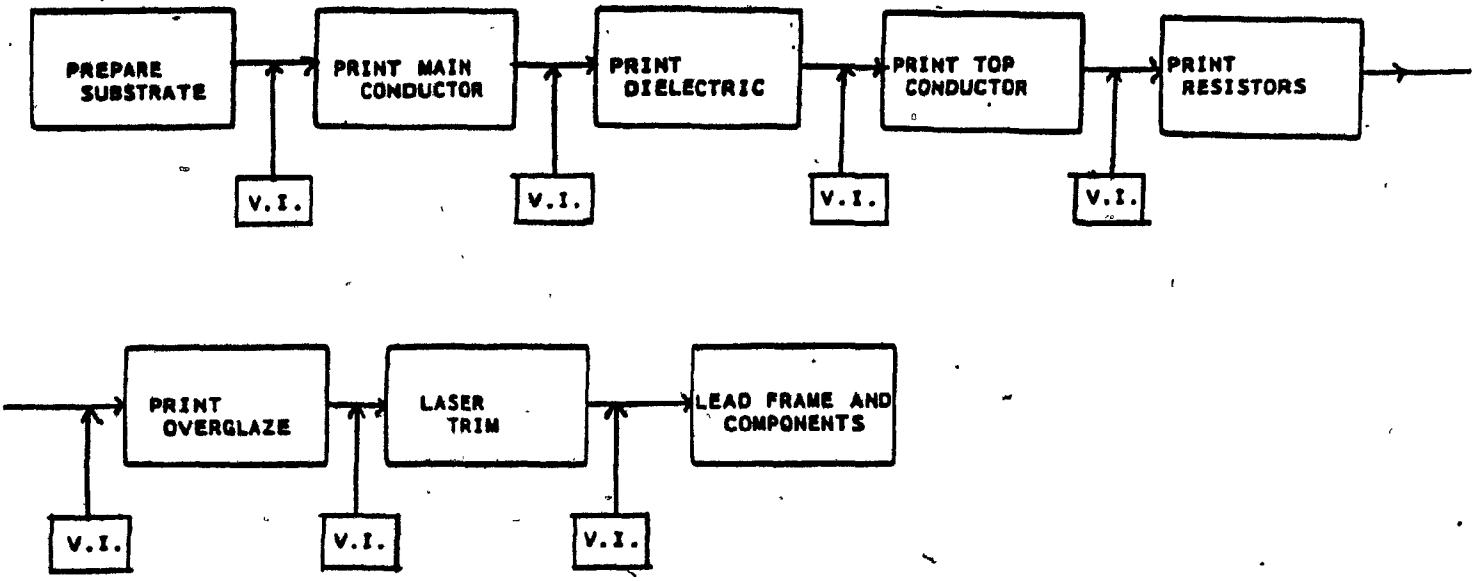


Figure 1.7 Assembly process for hybrid circuits with the location of required two-dimensional visual inspections.

circuit visual inspection, human inspectors can easily overlook some of these defects. This is aggravated by the fact that currently used hybrid circuit production technologies have an error rate of 20% to 30%. Also, the inspection might have to be done in unfavourable environments.

In an automated inspection system, once the inspection methods are perfected, there is no danger of overlooking defects. An automated system can examine every point of every single circuit with same 'concentration'. An automated visual inspection system can

1. find defects
2. report their exact location
3. report their dimensions
4. report their nature
5. compute statistics on defects

6. pass all of this information to an automated repair station or notify a human operator in case of ambiguities.

Another advantage of automated inspection is that it can be performed as part of the production process, thereby matching the high speed of the production process. Boards do not have to be transported to some inspection site every time they need to be inspected. The production and inspection systems can be integrated into one large system.

The idea of an automated inspection system is closely related to the idea of automatic repair. Contemporary robotics research provides us with means for handling and repairing electronic circuits using robot arms, without the intervention of human operators. The data generated by an automatic inspection system can be directly communicated to a repair station where circuits can be fixed if such an operation is feasible and economical. Such a repair system is currently being designed by other researchers in our laboratory ([Levine84], [Mansouri84]).

1.5 Contribution of This Thesis to the Hybrid Circuit Inspection Problem

The research conducted in this thesis consists of designing an automated visual inspection system for the two-dimensional hybrid circuit inspection problem. By two-dimensional inspection we mean the inspection of the stages of the circuit in which a three-dimensional image of the board is not required. These stages include the bare ceramic substrate as well as the printed layers, i.e., conductor, dielectric, resistance and overglaze layers. Such a design can be divided into three distinct tasks:

1. Finding proper algorithms to detect defects.

2. Building a rule-based production system which will manage the general operation of the system as well as the fault analysis task.

3. Designing the hardware for such a system.

By proper algorithms we mean efficient algorithms which are particularly designed to detect the kind of faults that can occur during the manufacturing process. These algorithms should also be general enough to be applicable to other similar inspection problems, e.g. printed circuit boards or new additions to hybrid circuits. The managing-analysing production system should also be suitable for the particular task but flexible enough to accept new

rules or type of circuits. Finally, the hardware includes the proper computer equipment on which such an inspection system would run as well as robot arms, microscopes, conveyors belts and the like which would be used to manipulate and view the circuits.

The first two tasks are designed, implemented and discussed in this thesis. The rule-based production system and the fault detection algorithms successfully find defects or suspected defects on any layer of the hybrid circuits, report their location, size, and the inspection rules they break. The production system is designed such that its modification or additions to it are very simple to perform. Therefore, if the threshold value of a certain rule must be changed according to production needs, if a new layer must be added, or if a different type of electronic circuit must be inspected, all one would have to do is to alter some of the productions or add new ones. This allows great flexibility in adapting this visual inspection system to various needs. A number of methods have been proposed and implemented for the hybrid circuit visual inspection problem, and a few papers have been published about them ([Arlan79], [Arlan80], [Chien75], [Nakashima84], [Lee78], [Zimmerman82]). These papers deal with methods for automatically detecting hybrid circuit defects from two-dimensional images, or with hardware setups for the inspection problem. However, none of them mention a full inspection system which combines proper algorithms to detect faults as well as an organizing production system which analyses these faults and generates information about the appropriate actions to take. To our knowledge, the work described in this thesis is the first research of such dimensions.

1.6 Review of Chapters

The general organization of this thesis is as follows: Chapter 2 reviews past research on the same subject or similar fields. Automated visual inspection in general, as well as the particular case of printed circuit boards and hybrid circuits are examined. Chapter 3 is a general introduction to rule-based systems. The concept of a rule-based system is studied, and its application to the general management of an inspection system and to the decision-making aspects of fault detection are discussed. Chapter 4 is the actual detailed description of the inspection system designed in this thesis. Finally, Chapter 5 gives application examples plus a general discussion of the hardware needed to implement such a system in an industrial environment.

Chapter 2 Historical Background

2.1 Automated Visual Inspection

In Chapter 1, we mentioned several reasons for using visual inspection during the production of electronic circuits. We also discussed the advantages of automating this kind of inspection. In this chapter, we will review the methods that have previously been designed for this purpose, classify them, and discuss their effectiveness and drawbacks. This will not be an exhaustive listing of all the methods used in automated visual inspection, but rather a sample of them for the classification and evaluation of various approaches.

2.1.1 Applications of Automated Visual Inspection to Electronic Circuits

Hybrid circuit inspection is one of the newest fields of automated inspection of electronic components. During the last decade, numerous applications of visual inspection have been attempted in industry. Several articles have been published about these methods and their applications ([Chin82]). Inspection systems have been used in the following areas related to electronic circuits ([Levine81]):

- printed circuit boards
- printed circuit drilling
- wire bonding
- photomasks

- integrated circuits
- hybrid circuits .

Other experimental systems have been designed in related electronic production areas, such as light bulb filaments, connectors, relays, etc. A comprehensive listing of the literature that has been published in these areas can be found in [Chin82].

2.1.2 Classification of Approaches

All the methods designed for the inspection of electronic circuits can be classified into two generalized categories. These are *template matching* and *design rule checking*. Before mentioning some of the research done in electronic circuit visual inspection, it is a good idea to describe the two methods briefly, illustrating them with examples.

2.1.2.1 Template Matching

Template matching is the most obvious and straightforward method for visually inspecting any kind of manufactured product. It consists of having an 'ideal' model for the product in question, and comparing the product to this ideal model, or *template*. The simplest form of this method consists of a pixel-by-pixel comparison of the image to the template, and is known as *image subtraction*. Image subtraction involves a simple exclusive-or operation between to images. The resulting image is a defect map of the product. Figure 2.1 illustrates this method. Two examples of this kind of inspection can be seen in [Sandland77] and [Skinner77].

It is quite apparent that without some means of approximation and adaptation, direct image substraction is very sensitive to edge misalignment errors, misregistration, and other mechanical and human errors. Besides, there can be cases where a perfect match of the circuit and the ideal model is not absolutely necessary for the proper operation of the circuit. Slight shrinking or swelling of circuits are two such cases ([Chin82]). In order to overcome these difficulties, variations of the image subtraction methods have been used. The general idea in these cases is to examine local features on a circuit instead of performing a global inspection of the whole circuit. Such methods are known as *feature matching* or *pattern matching*.

Typically, standard features are extracted from the ideal model and from the circuit to be inspected. Then these features, or patterns, are compared, instead of comparing the

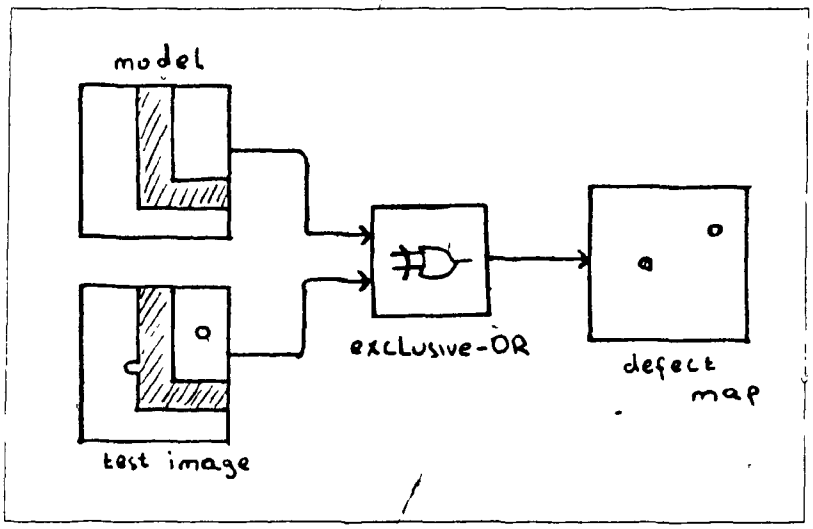


Figure 2.1 The image substraction method

whole circuit with the whole model. This strategy compresses the image data for storage and reduces the sensitivity of the input data. The selection of adequate features is itself a difficult problem, since an electronic circuit can contain a great variety of 'patterns' and a 'standard' set of shapes is not well-defined for a given type of circuit The following articles are examples of implementation of the feature matching approach. [Jarvis80], [Krakaner78], [Takaki78].

The pattern matching approach in its simplest form does not entirely solve the approximation problem mentioned above in the image subtraction case More complex pattern matching methods have been designed to overcome this difficulty. These fall into two categories: *approximation* and *syntactic* methods In the approximation methods, the features that are extracted are not compared on a point-by-point basis Instead, the patterns and the circuit are approximated. This not only saves computation time and storage, but also minimizes false alarms due to lack of an exact match Such approximations include polygonal approximations, description by properties such as angles of corners or spatial relations of the corners to each other, low resolution approximations, Fourier transforms, etc. ([Pavlidis78]). The syntactic methods take a different approach. They describe a large set of complex objects using small sets of pattern primitives and structural rules A structural description of these primitives can than be determined, forming a 'grammar' A set of primitives can then be formed to describe common defects, and these primitives can be searched for on the circuit under examination. [Pavlidis76] and [Bjorklund77] give examples for the application of this approach to the PCB inspection problem.

2.1.2.2 Design Rule Checking

A very different approach to automated visual inspection is inspection by design rule checking. This approach does not rely on ideal models; instead, it checks whether the circuit under inspection conforms to the original design rules. This is basically the method a human inspector uses to detect defects There are two obvious advantages to this approach· no storage is necessary for a model, and the criteria for the defects are much more flexible than in template matching. However, this method also has disadvantages when compared to template matching. First, there can be cases where design rules are satisfied even though the patterns are incorrect; this would result in possibly fatal errors passing the tests Also, designing algorithms to find those parts which are in conflict with the design rules can be quite a tedious task. Humans perceive distinct parts and characteristics on a circuit very easily. For example, conductor lines, probe pads, or distances between them are quite apparent for an inspector Telling a computer to check the distance between two conductor lines on the other hand, requires teaching the computer the concept of conductor lines, areas between them, and the width of these areas. Finally, because of the complexity of the problem, such algorithms can be quite expensive computationally.

Examples of such methods designed for electronic circuit inspection are *dimensional verification* and *spatial filtering*. Dimensional verification algorithms are based on the assumption that the width of any pattern on the circuit must be within certain limits, any parts of the circuit which exceed these limits have suspected defects on them. Figure 2.2 illustrates this idea.

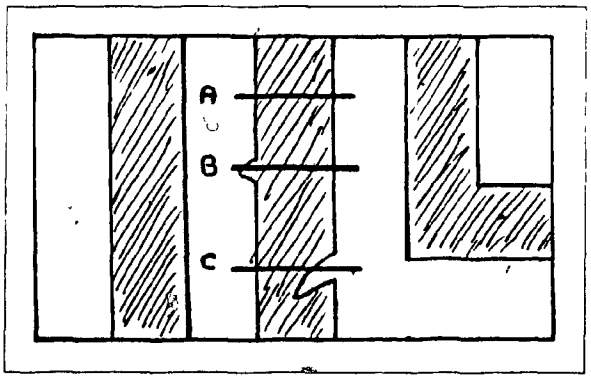


Figure 2.2 Dimensional verification Measurement A is accepted, B and C are rejected

[Thissen77], [Nakashima79], [Bentley80], [Sterling79], [Restrick77] describe various implementations of the dimensional verification approach. Filtering of a circuit image to

find defects involves finding high frequency spatial variations, i e. small areas on the circuit. In this case, it is assumed that defects are either printed regions or 'missing' regions on the circuit whose area is smaller than of any normal part of the circuit. One way of detecting these small areas on an image is by filtering its Fourier Transform ([Watkins69], [Almi77], [Axelrod72]). Another interesting method is the expansion-contraction method proposed by Ejiri *et al* ([Ejiri73]) In this method the input pattern is expanded and then contracted by the same amount, which has the effect of filling defects described as 'holes' A comparison of the resulting image with the original gives a map of these defects The inverse operation, contraction-expansion, finds protrusions. The sum of these operations has the effect of filtering high frequency components of the image. Figure 2 3 describes this method.

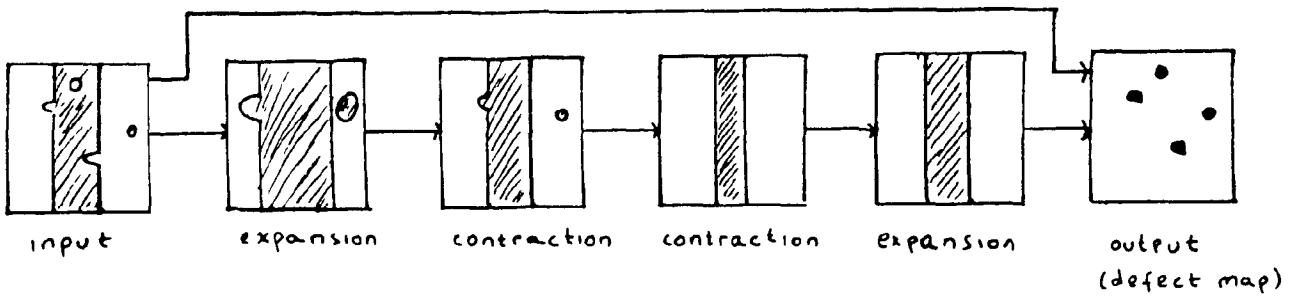


Figure 2.3 The expansion-contraction method

2.2 Visual Inspection of Printed Circuit Boards

The PCB inspection problem is one on which extensive research has been performed over the years Since visually PCBs are very similar to hybrid circuits, their automatic inspection presents the same kind of problems as in the case of hybrid circuit inspection. In fact, with some minor exceptions, a PCB could be considered the same as the conductor layer of a hybrid circuit. Therefore, the research that has been done for the PCB inspection problem is relevant for the topic of this thesis. One main difference between the two inspection problems is the fact that much smaller error limits are used for hybrid circuits. Thus, a more detailed analysis can be tolerated in their case. This section will give an overview of methods designed specifically to inspect PCBs, while the next section will focus on the hybrid circuit inspection methods.

Image Subtraction. Direct image subtraction for PCB inspection has been described by several authors ([Fehrs72]). Various methods of speeding up the comparison process

have been reported. Peterson ([Peterson74]) has designed a method where the PCB to be inspected and the ideal model of the circuit are scanned by two cameras in parallel and the inspection is performed with high speed. Olsen ([Olsen76]) prepared a transparent mask with a light blocking pattern corresponding to the desired PCB. When he registered this mask over the PCB under inspection and shone an appropriate colour of light onto the setup, faults appeared as bright spots on a dark background.

Feature Matching. Jarvis ([Jarvis80]) designed a method where each conductor-substrate border pattern extracted from the PCB is matched again a set of predetermined 5x5 pixel patterns. Areas which do not match any of the patterns are subject to further analysis to determine whether they are flaws. Such analysis includes computation of conductor areas, the ratio of area to length, boundary length, and others. Only a few hundred 5x5 templates are sufficient to perform the inspection on a complete circuit. This method eliminates the need for precise alignment but requires a careful preparation of the templates.

Chin *et al.* ([Chin78]) reported an inspection system which can be trained by a human operator. Interaction is used to extract features, eliminate noise, define a graph model of the circuit, encode models in compact data structures, etc Once the system 'learns' about the model, it directs the scanner which analyses the circuit to examine the expected features. Since the system knows where to look for different expected features, it does not require the scanning and processing of the entire PCB image, and is therefore more economical.

Dimensional Verification. This idea has been widely used in the inspection of PCBs. Although dimensional verification will not detect every kind of error encountered in PCBs, most faults could be considered as being either too narrow or too wide. This does assume that the circuit is designed according to strict dimensional constraints. [Danielsson79] discusses several dimensional verification algorithms.

Cheng ([Cheng78]) describes a system where PCBs are mounted on an X-Y table and scanned by a laser beam. Pixels are counted during the scan and conductors that are too narrow or too wide are detected. Thissen ([Thissen77]) creates a model for the circuits to be examined consisting of center lines for all the conductor strips on the circuit, with threshold width values associated with every line. Nakashima *et al.* ([Nakashima79]) use a laser diffraction technique to measure the width of conductors. Restrick ([Restrick77]) uses linear solid state arrays to measure the line widths of conductors.

Spatial Filtering. As mentioned before, Ejiri *et al* designed a method where the input pattern is expanded and contracted by the same amount, giving a map of small fault areas

([Ejiri73]). Essentially, this method filters high frequency components of the circuit, and these are possible defects This method is one of the two main methods used in this thesis to detect faults on different layers of hybrid circuits. Digital or analogue Fourier Transforms are other methods used by various authors to filter out high frequency regions of PCBs ([Almi77], [Axelrod72]).

2.3 Visual Inspection of Hybrid Circuits

Hybrid circuits are relatively new in the market. Although the production process is quite sophisticated, there are no fully operational automated inspection systems to inspect the circuits during or after the production. Apart from electrical tests which can only detect certain aspects of the circuits, all visual inspection is done by human inspectors. Only a few fully automated visual inspection methods have been reported in the literature.

Lee ([Lee78]) reports an experimental inspection center designed at Honeywell Information Systems. The center uses existing digitizing technologies coupled with a minicomputer. Computer controlled transport of the circuit allows full coverage of the circuit; an algorithm has also been designed for the registration of the circuit with respect to the model images. The system inspects only the first two conductor and dielectric layers. After precise alignments and noise elimination, it does a direct comparison between the image of the circuit and the screen. The resulting difference map is analysed by algorithms not described by the author. The system takes six minutes per layer to run, is quite accurate, but lacks the flexibility of an 'intelligent' repair station where interaction, rule modification or addition are possible.

Arlan *et al.* have developed a system where the image subtraction is performed in a video processor by comparing video frames with the stored model on video disk ([Arlan79], [Arlan80]). The authors claim that their system can detect 0.5 – 2 mils wide open and short circuits due to the high resolution of Vidicon cameras. Again the system relies on direct comparison of circuit layers to the screens, and no details are given about how the defect map is processed once it is obtained.

Zimmerman *et al.* ([Zimmerman82]) report an automated visual inspection system called AVIS. AVIS performs a one-dimensional electro-optical scanning using a translation table. The edges of the circuit are detected with a hardware oriented edge detector, and these edges are compared to the edges of a pre-stored ideal model. The match criterion is the number of unmatched pixels per line segment in the edge map. The system does not output any kind of information about defects.

Chien and Snyder ([Chien75]) built a system of programs to process images of hybrid circuits. To achieve this they used a procedural model of the circuit. Nakashima *et al.* ([Nakashima84]) designed methods for the recognition of parts in hybrid microcircuits. One of their methods recognises thick film conductor patterns using matrix filtering. Another method uses peak sharpening of the matching degree curve in template matching, thereby increasing the signal to noise ratio.

2.4 Conclusion

A visual inspection system for the hybrid circuit manufacturing process should perform the following tasks: Inspect the circuits, report the presence, location and description of defects, keep statistics and communicate with a repair station. The examination of various visual inspection methods and systems encountered in the literature shows clearly that although several valuable tools have been proposed and used, a system accomplishing all the tasks listed above has not yet been designed. The system described in this thesis is an attempt to combine various fault detection methods described in the literature with an rule-based expert system which has the ability of taking 'intelligent' decisions about the information given by these methods.

Chapter 3 The Approach: Rule-Based Systems

3.1 Introduction

There are basically two approaches to the automated visual inspection problem as reported in the literature: one of them is *template matching*, the other is *design rule checking*. These two approaches were mentioned in the review of the historical background on visual inspection. As will be seen in Chapter 4, the system described in this thesis takes advantage of both approaches and uses a rule-based system to combine and analyze the data coming from either. In this chapter, we intend to give a formal description of rule-based systems.

3.2 Rule-Based Systems

Rule-based systems are a subset of a general class of systems called *knowledge-based systems*. The term *knowledge-based* is somehow vague since every existing system, including procedural systems, must be based on some sort of knowledge. The dividing line between the two kinds of systems is that in a knowledge-based system, the knowledge is well defined and easily identifiable within the basic structure of the system. Rule-based systems are a subset of knowledge-based systems where this identification is very clear: the knowledge and the control mechanism are distinct. The *rules* of a rule-based system contain the knowledge and form the world model. *Production Systems*, first introduced by Post ([Post43]), form a tool that has been used for the implementation of rule-based systems. The next sections contain a description of production systems and some examples of rule-based systems encountered in the literature.

3.2.1 Production Systems

In its simplest form a production system is formed by three components (Figure 3.1):

1. a data structure, or working memory, forming the world model
2. a set of rules (or productions)
3. a rule interpreter.

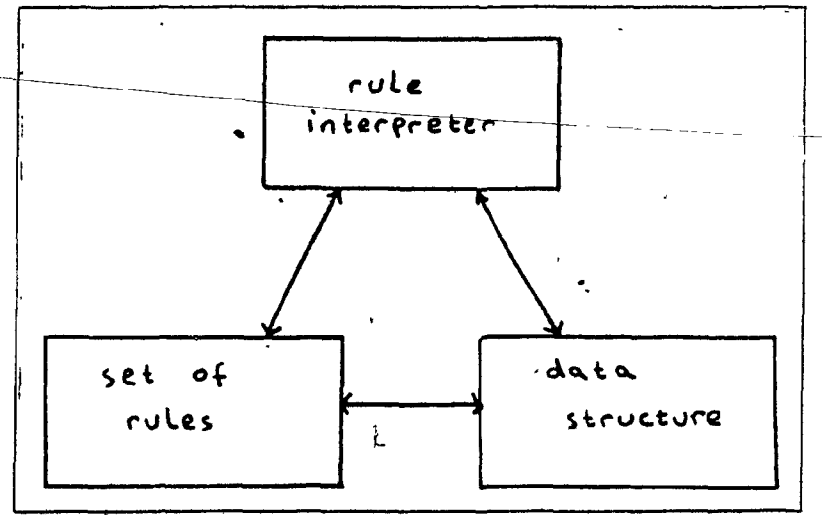


Figure 3.1 A production system

The world model is basically a finite collection of symbols. New symbols get added to this collection with the risk of losing old ones. The rules are condition-action pairs. The 'left hand side' of the rule is the representation of a certain state of the world. If this state matches the current state of the world model, the rule 'fires'. In other words, the 'right hand side' or the action associated with the rule gets executed. This possibly modifies the state of the world. In such a simple form of a production system, the role of the rule interpreter is trivial. It compares the condition component of rules with the world model and executes the appropriate actions. More complicated rule interpretation schemes have been designed. These will be described in the next section.

The characteristics of production systems which make them suitable for artificial intelligence programs are the following:

1. The database is the only method of communication between rules. This communication is also unique in the sense that all the rules have access to all the parts of the database ([Davis77])

2. The rules are independent entities. They can be modified, deleted, or new rules can be added without seriously affecting the performance of the system ([Winograd75]).

3. Rules represent primitive actions at the conceptual level. The action executed by the rule might be a complex process, but it is considered to represent a simple unit of behaviour within the context of the system.

4. Productions systems offer a unified approach to control and data. Rules have a well defined format readable by the system, and the system can modify rules, add new rules or delete them through the actions of existing rules.

Production systems can be used in every application that procedural systems could be used in; they are both Turing equivalent. The choice depends on the particular application, and is a matter of programming ease and efficiency. It should be noted that the production system philosophy is closer to the way humans reason. The database can be compared to the short-term memory of humans. The long-term memory would then be analogous to the set of rules, and the 'intellect' would be the rule interpreter.

3.2.2 Control in a Rule-Based System

The control cycle in a basic production system is simple: find a rule whose conditions match the data, and then execute the action associated with that rule. In complex systems, the rule selection become an important problem since at various stages more than one rule's conditions can match the data. Therefore, some method of selecting one out of a set of rules which could be fired must be designed. This problem is known as *conflict resolution*, and the set of candidate rules are known as the *conflict set*. The decision problem therefore becomes:

1. [Match] Evaluate the left hand sides of the productions to determine which are satisfied given the current state of the world.

2. [Conflict Resolution] Select one production with a satisfied left hand side. If no productions have satisfied left hand sides, return control to the user.

3. [Act] Perform the actions specified in the right hand side of the selected production.

4. If the action was to halt the system, return control to the user; otherwise go to step 1.

There are many different methods of selecting rules ([Davis77]). These include the following:

1. **Implicit order:** Rules are tested in the same order they are stored. When a match occurs, the process executes the related action, and either restarts at the top of the list, or continues from the last rule that it had selected.

2. **Rule order:** All rules are given prespecified priorities, and the matching rule with the highest priority fires. This enables fast rule matching, but adding new rules to the system becomes difficult.

3. **Data order:** The rule that matches the most important data item is selected. This means of course, that the data elements must be ordered and the matching process becomes slower

4. **Generality order:** The most specific rule gets selected. This is also fast but requires extra work when adding new rules.

5. **Temporal order:** The rule that has most recently fired gets selected. This is a fast method as well, but it risks being biased towards a few of the rules at the expense of using other sources of knowledge.

More complicated conflict resolution methods have been used in the literature ([Feigenbaum71], [Davis76], [Davis80], [Zucker78]). OPS5, the production system used in this thesis, modifies the basic control cycle mentioned above in the following manner ([Forgy81]):

1. [Conflict Resolution] Select one production with a satisfied left hand side. If no productions have satisfied left hand sides, return control to the user.

2. [Act] Perform the actions specified in the right hand side of the selected production.

3. [Match] Evaluate the left hand sides of the productions to determine which are satisfied given the current state of the world.

4. If the action was to halt the system, return control to the user; otherwise go to step 1.

This cycle is more convenient than the classical one because when the cycle ends, the conflict set is consistent with the current contents of the working memory. Using this cycle, the OPS5 system applies a conflict resolution strategy called *LEX* ([Forgy81]). LEX contains four rules which are applied in order:

1. Discard from the conflict set the instantiations that have already fired. An *instantiation* is an ordered pair of a production name and a list of working memory elements satisfying the production's left hand side. If there are no instantiations that have not fired, conflict resolution fails and no instantiation is selected.

2. Order the instantiations on the basis of the recency of the working memory elements, using the following algorithm to compare pairs of instantiations: First compare the most recent elements from the two instantiations. If one element is more recent than the other, the instantiation containing that element dominates. If the two elements are equally recent, compare the second most recent elements from the instantiations. Continue in this manner either until one element of one instantiation is found to be more recent than the corresponding element in the other instantiation,or until no elements remain for the instantiation. If one instantiation is exhausted before the other, the instantiation not exhausted dominates: if the two instantiations are exhausted at the same time, neither dominates.

3. If no one instantiation dominates all the others under the previous rule, compare the dominant instantiations on the basis of the specificity of the left hand sides of the productions. Count the number of tests that have to be made in finding an instantiation for the left hand side. The left hand sides that require more tests dominate.

4. If no single instantiation dominates after the previous rule, make an arbitrary selection of the dominant instantiation.

The LEX strategy has several advantages: It prevents instantiations from executing more than once, it forces the production system to attend to the most recent data in working memory, and it gives preference to productions with more specific left hand sides ([McDermott78]).

3.2.3 Some Rule-Based Systems in Literature

Davis and King classify rule-based systems into two basic categories ([Davis77]). The first one is the psychological modelling of human behaviour. The second class is a type of rule-based system which can be described as a performance-oriented expert system. The former was motivated by the similarity of human cognitive processes to the structure of production systems. The database of finite length is analogous to human short term memory. Such systems are attempts to formulate a set of production rules which will reproduce equivalent human behaviours. On the other hand, the performance-oriented expert systems contain the knowledge about one particular task and try to achieve competent behaviour in accomplishing the task. Large application oriented artificial intelligence systems belong to the latter category.

Several systems attempt the psychological modelling of human behaviour. The PSG system was used to explain memory scanning tasks ([Newell73]). PAS II studied learning and adaptive behaviour ([Waterman74]), and VIS simulated the descriptive ability of visual images ([Moran73]).

Typical examples of knowledge-based expert systems are: MYCIN medical consultation program ([Shortliffe76]). DENDRAL, a system which inferred chemical structures for organic molecules ([Feigenbaum71]). TEIRESIAS, which used faulty MYCIN consultations to track down and correct the rules that caused the errors ([Davis76]). META-DENDRAL, which inferred rules of mass-spectrometry for possible later use by DENDRAL ([Buchanan76]). [Feigenbaum77] gives a review of selected expert systems; [Rychener81] includes a bibliography on such systems. [Hayes-Roth83], [Buchanan84] and [Weiss84] are three recent books published on the subject.

Chapter 4 Description of the System

4.1 Introduction

This thesis deals with the design and implementation of a visual inspection system for bare hybrid boards. Only the software for such a system has been implemented. This chapter will provide a detailed description of the system, putting particular emphasis on the rule-based system which governs the entire inspection process, and on the fault-detection modules. An overview of the hardware requirements for an industrial application of the system can be found in the next chapter.

Our inspection system consists of a rule-based production system which has two distinct tasks: general management and fault analysis. Programs which are implementations of fault detecting algorithms run as sub-modules to the rule-based system. In the next section, first the fault detection algorithms will be described in detail, and then the rule-based system itself will be examined.

4.2 Description of the System

One of the advantages of using a rule-based system for the inspection of circuits is that the fault detection methods can be relatively simple, straightforward mechanisms, since the analysis rules will subsequently examine every fault candidate and will give detailed information about them. In this thesis, two basic methods are used to detect defects on hybrid circuits: One is direct comparison or matching of the circuit under inspection against the ideal model of the circuit, and the other is the expansion-contraction method first introduced by Ejiri *et al.* ([Ejiri73]). As it will be seen in the following sections, the

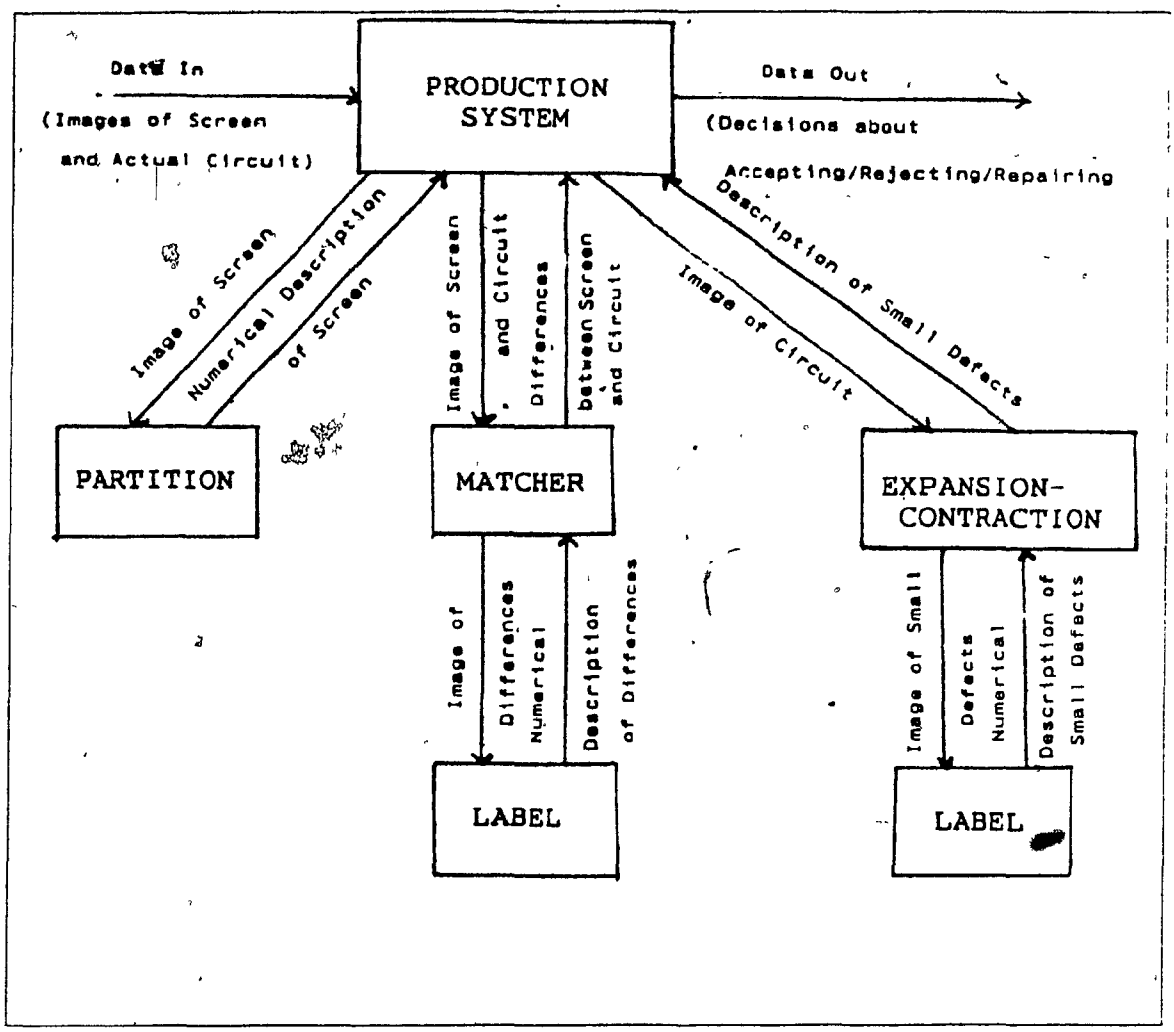


Figure 4.1 Block diagram of the system

first one of these methods is more suitable for finding large defects, while the second is superior for smaller or high frequency defects. The program modules implementing these two methods will be referred to as MATCHER and EXPCONTR throughout this chapter. A third module, PARTITION, is used to dissect the circuit under analysis into relatively simple polygons and classify these polygons. The data obtained from this module is used by the rule-based system to perform a more intelligent analysis of the defect candidates. The module LABEL takes the picture outputs of MATCHER and EXPCONTR and returns numerical values which are used by the rule-based system for the analysis of the faults. Figure 4.1 is a block diagram which shows the communication between the above mentioned modules. The next sections will describe all the modules in detail.

4.2.1 Fault Detecting Modules

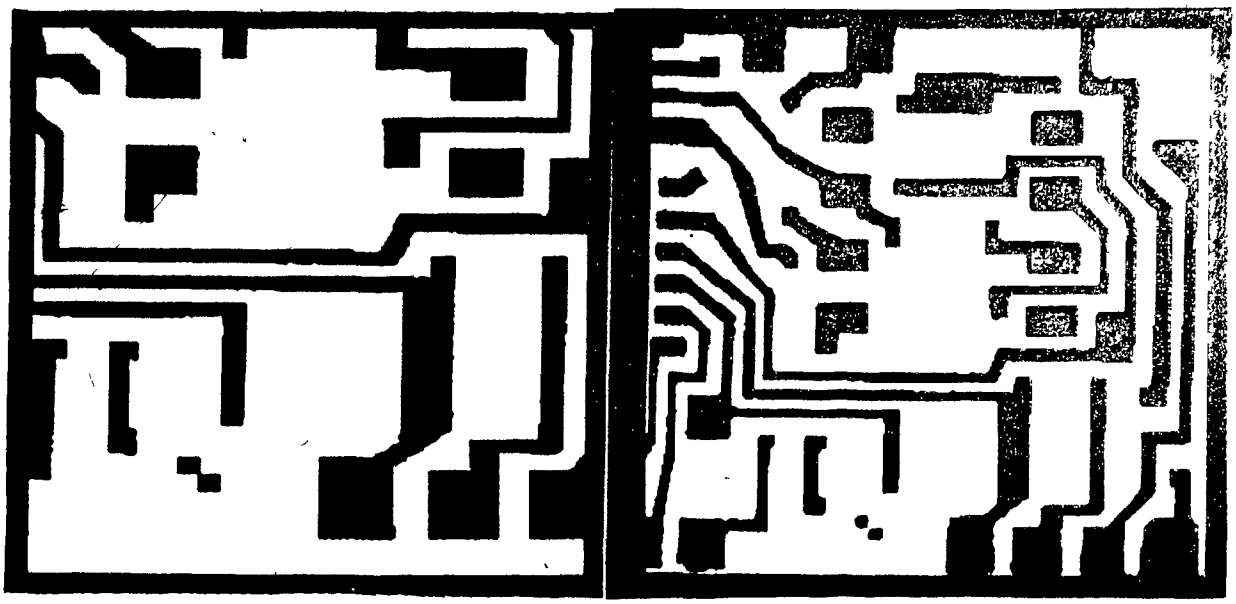
4.2.1.1 The MATCHER module

As mentioned in Chapter 1, the screens that are used for the thick film printing process can be viewed as ideal models for the circuits under inspection They constitute an image of what the circuit 'should' look like Very often, the circuit ends up not quite looking exactly like the screen that was used to print it Smearing, sputtering, cracking and shrinking are some of the reasons such a discrepancy can occur Every one of these mismatches are potential defects, and should be examined

The MATCHER module's purpose is to do a direct comparison between the circuit under inspection and the image of the screen. This task is not as straightforward as it appears because of scale changes and misregistration that can occur during the digitization of the circuit and the screen. Because of such changes, the image and the model must be rotated and rescaled with respect to each other in order to achieve a perfect match If this is not done properly, the matching process can be expected to give errors even at locations where there actually is a perfect match

Such a rotation/rescaling operation would be a complex task to perform if the images did not offer some sort of reference point Fortunately, manufacturers of hybrid circuits use markers for the printing of every layer to insure proper alignment of the layers with respect to each other. The markers are special shapes which are electronically non-functional They are printed with the rest of the layer at two distinct points of the circuit. Figure 4 2 shows examples of circuits and masks which contain such markers These markers are perfect reference points for the alignment and rotation of the images that the MATCHER program handles.

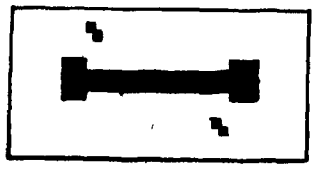
The MATCHER program works in two distinct modes: the *teaching* mode and the *inspection* mode. The aim of the teaching mode is to interactively define the markers that will later be used as reference points for the registration and scaling of the images In this case, the only input to the program is the image of the screen. This image, which ideally is binary, is in practice a grey level image because of quantization errors in the digitization process. In order to binarize the image, its histogram is computed, and a 'valley' value is found in the histogram. This value is the lowest histogram value between two peaks of the histogram indicating, respectively, the black and white regions of the images. The 'valley' grey level is taken as the ideal threshold value and the image is binarized according to this threshold. The resulting image is displayed on a monitor, and an operator selects



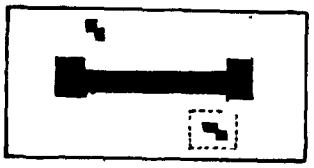
Screen Actual circuit

Figure 4 2 Portions of a screen and of a circuit containing the markers which are used for the alignment of circuits

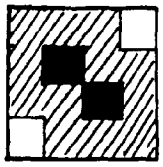
one of the markers by defining a window around it with the help of a joystick Once the marker is approximately extracted, the program computes its exact location A mask which will be used for template matching is also created out of the picture of the marker The mask consists of a grey level image where the shape of the marker has positive values and the region around it has negative values This is a way of creating a mask out of a template such that a correlation between the mask and the image will give maximum response at perfect match locations ([Levine85, Ch 5]) The other marker, which has the same shape as the first one, is located automatically by performing a template matching on the circuit The mask created out of the first marker is used as a template in this search Once the two markers are located, the following information is stored as data to be used in the inspection mode the mask which is used for the template matching to find the markers, the exact pixel locations of the two markers, the distance between them, and the direction of the line segment which joins them. The distance and the direction are stored to be used as references for the scaling and rotation operations which will be performed during the matching in the inspection mode. Figure 4.3 is a step by step explanation of the MATCHER program in teaching mode. It should be noted here that were the screen prepared by a CAD/CAM system and stored in computer memory, access to the perfect binary image of the screen would greatly simplify the operation described above.



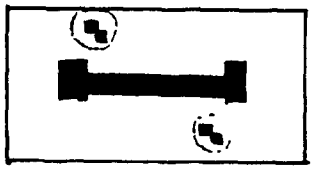
a Image of the mask gets thresholded



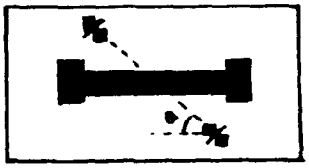
b A marker is extracted



c A mask is created out of the marker



d Exact location of both markers is found



e Distance and orientation of markers are computed

Figure 4.3 Step by step explanation of the MATCHER program in teaching mode

The template matching performed during the teaching mode as well as the inspection mode is a two-stage template matching ([VanderBrug75a], [VanderBrug75b]) This approach first employs a scaled-down version of the template which is matched against a proportionately scaled-down version of the image This preliminary search provides a rough estimate of where the pattern is in the image. A second search with the actual size image and template only around the estimated area will give the exact location of the pattern. The reason for performing such a two-level match is to save computation time. The complexity of a template matching operation on a $N \times M$ image and a $n \times m$ template is $N \times M \times n \times m$. If the same template matching is done with an image and a template which are respectively half the size of the original, the complexity becomes

$$\frac{N}{2} \times \frac{M}{2} \times \frac{n}{2} \times \frac{m}{2}, \quad \text{or} \quad \frac{N \times M \times n \times m}{16}.$$

In other words, a rescaling factor of r results into a simplification of r^4 in the matching operation. Once the approximate location of the template is found in the image, an exact

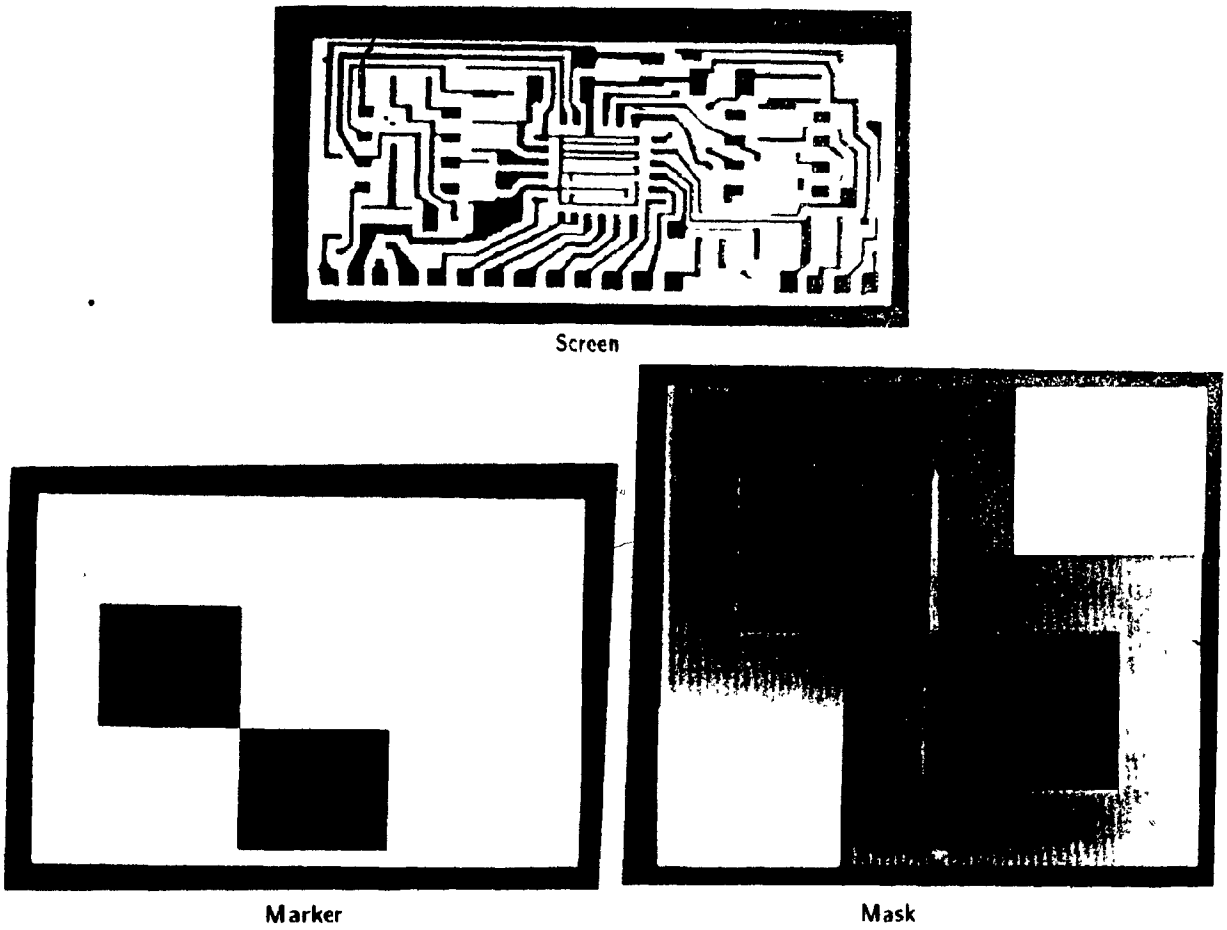


Figure 4.4 Picture of a conductor screen the marker which is extracted from it and the mask which is created from the marker

matching with the actual size image and template is performed only around the area of that location. This second stage of matching takes much less computation than a full template matching since it is done on a very small area of the picture

The pictures used during this thesis work were approximately 800 × 500 pixels For the two stage template matching, reduced pictures of 256 by approximately 160 were used, with proportionately reduced masks. After considering different values, the size 256 was chosen as an optimum value for maximum reduction without significant loss of information Pictures of a sample image and its related images (mark, mask) can be seen in Figure 4 4.

In the inspection mode, the MATCHER module takes as input the data obtained in the teaching mode, the image of the screen, and the image of the circuit under inspection. The latter is obtained by illumination from the bottom and capturing a picture through a camera attached to a microscope. There are two possible ways of illuminating a circuit:

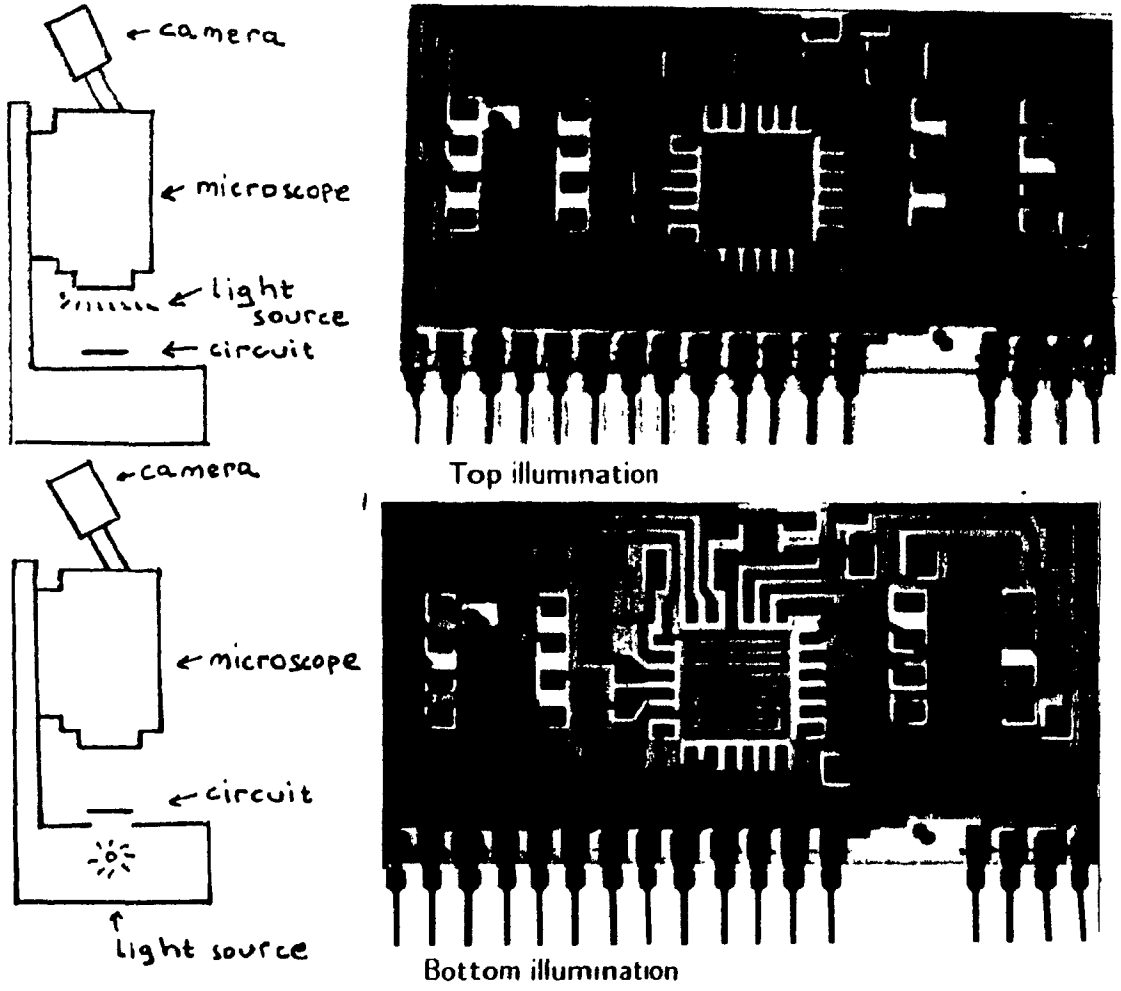


Figure 4.5 Pictures of a hybrid circuit using top and bottom illumination

from the top and from the bottom (see Figure 4 5).

Top illumination undoubtably gives more visual information about a circuit, since every shade and colour variation can be seen But because of specularities, reflections and porosities in a material, analyzing a top illuminated image of a circuit by computer is a very difficult task. Deciding which parts of the circuit belong to which layers (eg, conductor, dielectric) requires colour information and a high level analysis of the image. Low level computations such as histogram thresholding are not sufficient in this case since every material appears to reflect a wide range of grey levels in the histogram range. Thus, an analysis of the circuit using top illumination requires a lot of high level computation. In a real time industrial inspection environment this might be unrealistic with current technologies. Bottom illumination does not give all the information top illumination provides. It provides

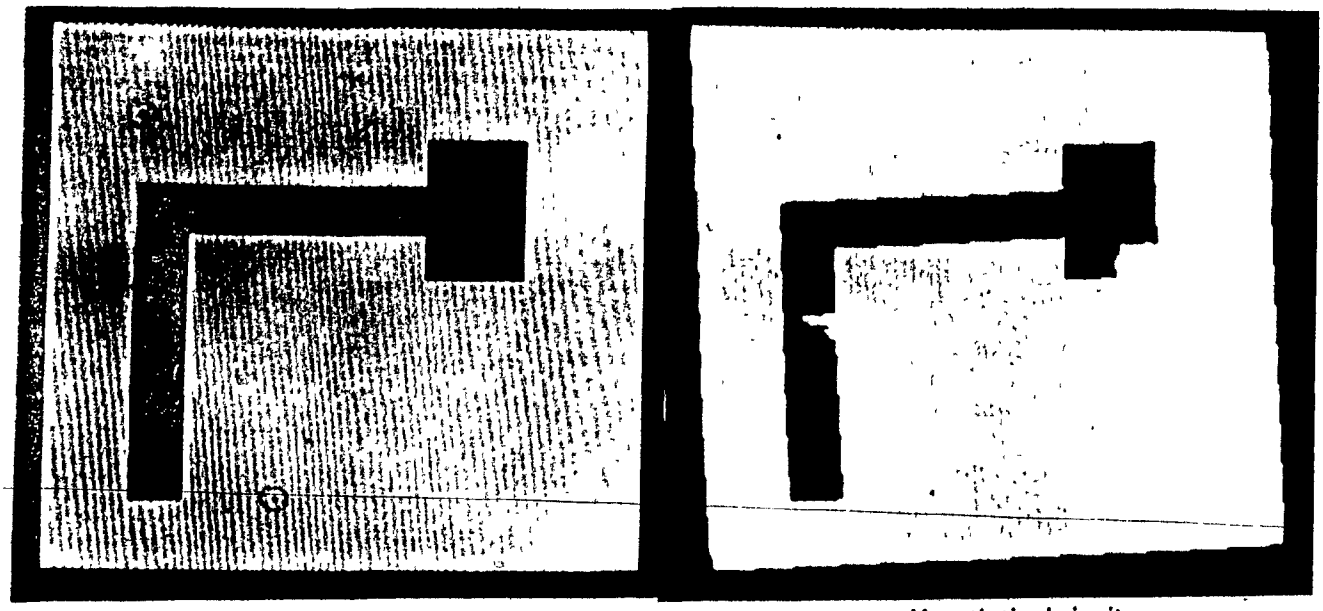
only an outline of the printed layers. But since all the printed layers are either opaque or translucent, this information is quite sufficient for the analyses to be performed. The only important loss of information is in the areas of the circuit where a part of a layer is printed on top of another layer which is opaque. For example, when a dielectric layer (translucent) is printed on top of a conductor layer (opaque), the parts of the dielectric layer printed over conductor lines cannot be seen with bottom illumination. In this case, it is assumed that a defect occurring on the dielectric layer over the conductor will also show up partly outside the area of the conductor. This assumption may not always be valid of course, but another method of inspection would require either sophisticated lighting equipment, or a large amount of computation. It is hoped that any defect missed because of this overlap problem will show up during electrical tests.

Once the image of the circuit is obtained using bottom illumination, the MATCHER program searches for the locations of the markers on the circuit using two-stage template matching described above. When it finds the markers, it computes the distance between them and the orientation of the line connecting them. These values are compared to the corresponding values for the screen, and the image of the circuit is rotated, rescaled and registered so that it is aligned with the image of the screen. The rotation and registration were performed digitally during this research but should ideally have been done by moving the circuit on an $xy\theta$ stage. Once the two images are aligned, a simple subtraction gives an image of the differences between the two images. Since a rotation performed digitally leaves digitization errors, the resulting image must be cleaned up. The cleanup is performed by canceling every cluster of three pixels or less. If a rotation table were used, such an operation might not be necessary.

The image of the differences is later translated into data usable by the production system, through the module called LABEL. This module will be explained in another section. Figure 4.6 is a step by step explanation of the MATCHER program in the inspection mode.

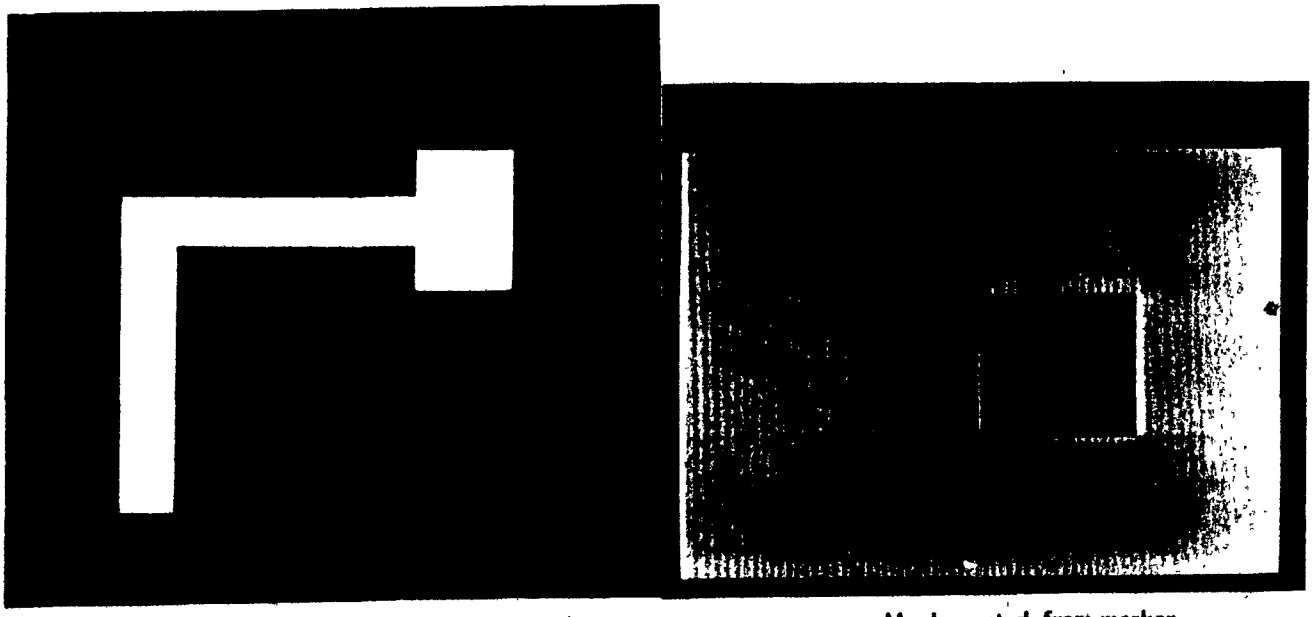
4.2.1.2 The EXPCONTR module

The expansion-contraction module, or EXPCONTR, is an implementation of the method suggested by Ejiri *et al.* in ([Ejiri73]). This approach takes advantage of the fact that most of the defects occurring on a printed circuit are small compared to the size of the circuit. Smearing, sputtering, cracking and other such incidents always cause small alterations to the printed material. The expansion-contraction method effectively extracts a map of these small defects.



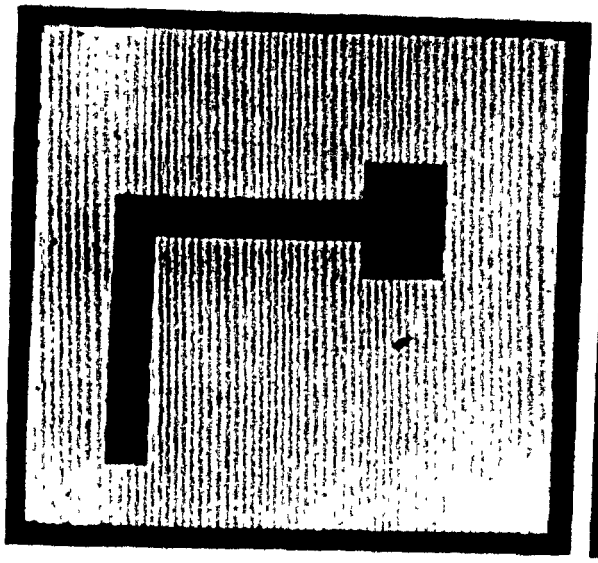
Hypothetical screen

Hypothetical circuit

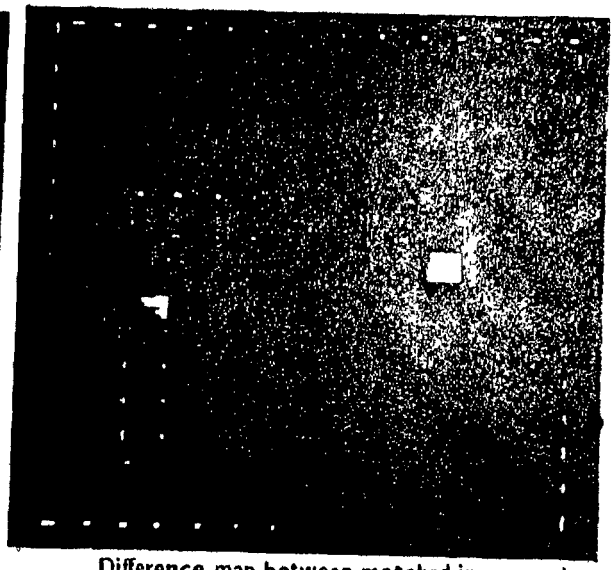


Marker on the screen is extracted

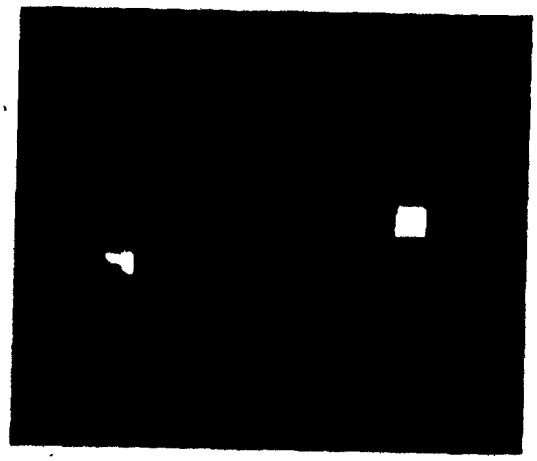
Mask created from marker



Match between screen and circuit



Difference map between matched image and screen



Cleaned up difference map

Figure 4.6 The MATCHER process on a hypothetical circuit. See Chapter 5 for applications on real circuits

The EXPCONTR module considers the circuit under inspection as a binary image of a set of complex polygons. The fault detection process is performed in two stages: expansion-contraction and contraction-expansion. In the expansion contraction stage, the borders of all the polygons are found, and they are expanded by a certain number of pixels

(see Figure 4.7). The border of the polygons is found using the following algorithm:

1. Scan the picture line by line starting from the bottom left until you reach a polygon pixel.

2. From the direction you arrived at the pixel, search its neighbours in a counterclockwise manner until you reach another polygon pixel.

3. Repeat this for every new pixel until the original pixel is reached again

The expansion can be performed digitally by specialized hardware or by optical means ([Ejiri73]), but in this thesis it was done in the following manner:

1. Find the border pixels of the polygon.

2. For every pixel on the border, define all the 8-connected pixels as border pixels.

3. Repeat the above procedure n times.

n is the number of pixels the polygons must be expanded. It is defined according to the inspection criteria for a particular circuit. Suppose that a particular constraint for the conductor lines states that the minimum width of a line should be half the width that the original design requires. Than a value of n equivalent to about half the width of the conductor would be appropriate. An expansion by that amount would fill in any nick or hole that might be found in a conductor line, and at the end of the expansion-contraction process, such a location on the circuit will be marked as a defect. Once the expansion is performed, the circuit is contracted by the same amount of n pixels. The contraction is done by finding the border and deleting it. This is done n times. The resulting image is identical to the original image except at locations where there are holes or nicks of a smaller scale than the size of polygons on the circuit. A difference of the resulting picture with the original one gives a map of holes or nicks on the circuit.

The second stage of the operation, the contraction-expansion, is the inverse of the previous one and is appropriate for finding extra printed elements on the circuit, resulting from sputtering, foreign objects and other accidents. This step is performed by contracting the circuit first, as described above, and then expanding by the same amount. In the

resulting image, all the parts of the circuit which are considerably smaller than the polygons will disappear. A difference between such an image and the original image will give a map of unwanted material on the circuit. The contraction amount n can be the same as in the expansion-contraction stage since most of design criteria about open or almost-open circuits are similar to the criteria for closed or almost-closed circuits.

Since both the expansion-contraction and contraction-expansion methods basically detect the high frequency components on a pattern, both will give false alarms at sharp corners (see Figure 4.8). The way to get around this difficulty is by taking into account the fact that false alarms will occur exactly at the same locations for both methods, and if there really is a defect on the circuit, only one of the methods will detect it. Therefore, an exclusive-OR operation between the resulting images of the two stages gives a map of the real defect candidates.

A list of errors that the EXPCONTR and MATCHER modules can detect, and the reason the two methods are used in parallel will be seen in another section, after the description of the PARTITION and LABEL routines.

4.2.1.3 The PARTITION module

Were the aim of this work just to detect defects on a hybrid circuit, the EXPCONTR and MATCHER modules would be sufficient to accomplish the task. But what is required is not only to detect defect candidates, but also analyse them according to the design criteria of the manufacturer. In order to accomplish this, the inspection system should not only know the location, size, etc. of defects, but also what type of circuit part the defect is located on. This is particularly important for the conductor layers, because the inspection criteria vary greatly for conductor lines, probe or solder pads, or other kinds of conductor components. Dielectrics, resistors and overglaze layers generally do not have components with different functional characteristics, but the analysis of their defects also vary according to what kind of conductor area they are printed on. This is why, given a circuit location for a particular defect, the inspection system should know what type of conductor is lying on that part of the circuit, if any.

The PARTITION module accomplishes this. It takes as input the image of the screens used to print the conductor layers, partitions these image into simple polygons, and outputs information about these polygons. This information consists of the type of polygon, the location of its corners, its center of gravity, and its area. It should be noted that what is required here is not the division of the image of the circuit into the simplest possible

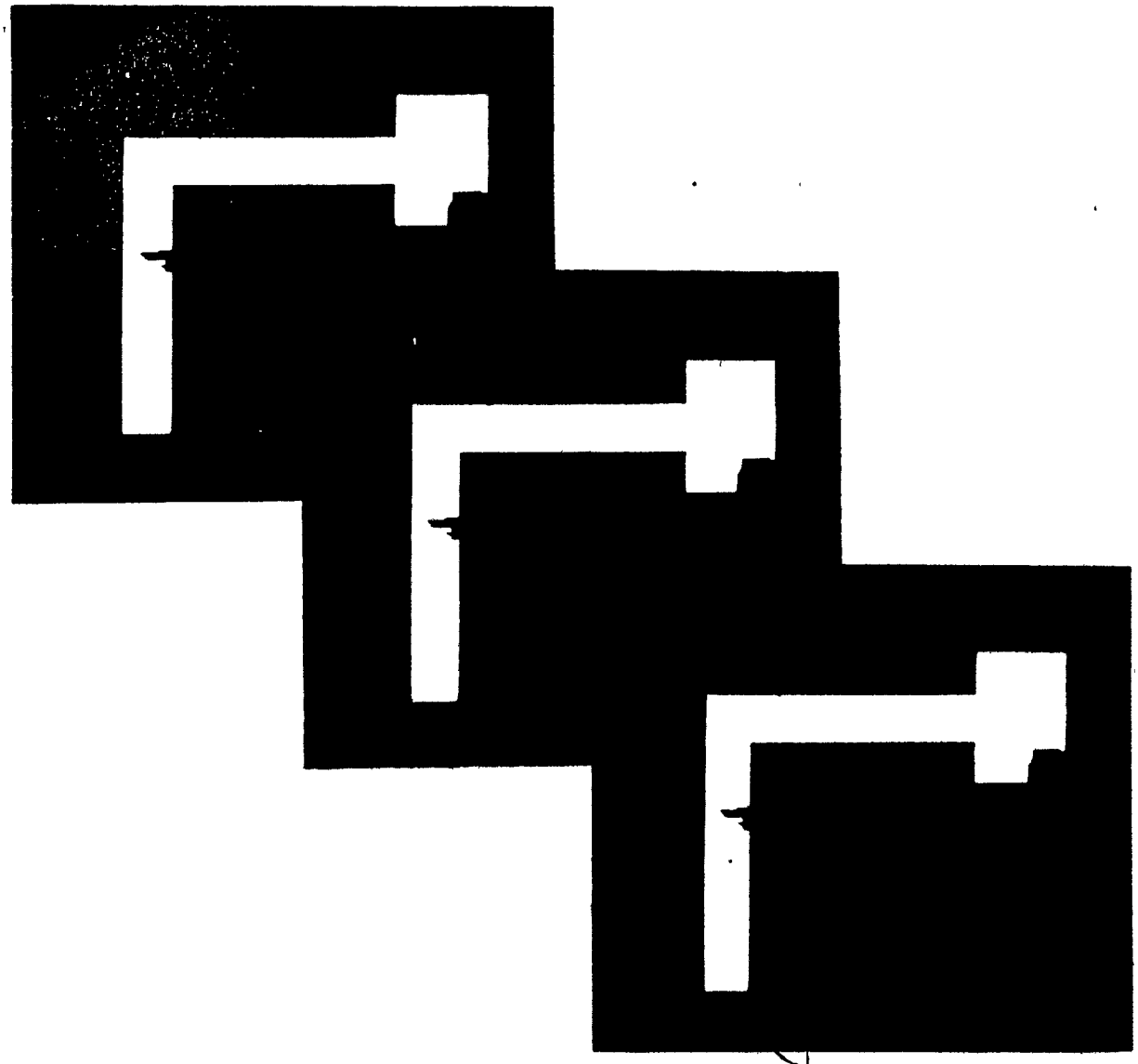


Figure 4.7a Expansion The polygons in the circuit are expanded by n (in this example 3) pixels. The red borders show the expansion.

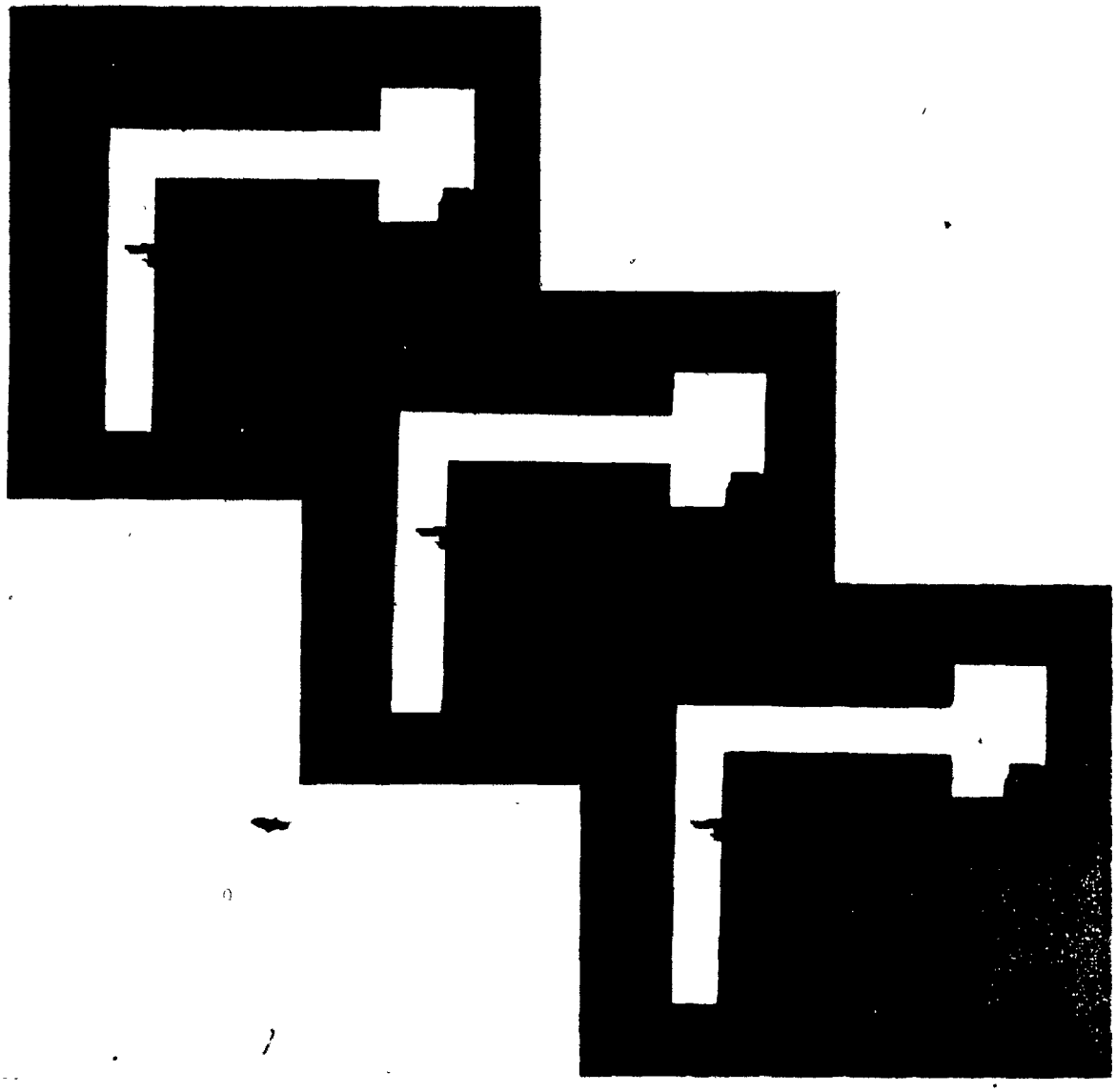


Figure 4.7b Contraction. The expanded polygons are contracted by the same number of pixels

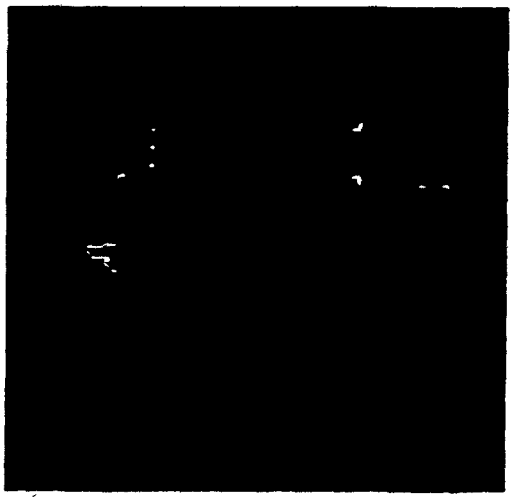


Figure 4.7c The result of the expansion-contraction process. The red areas show the differences between the original figure and the one created by expanding and then contracting the former

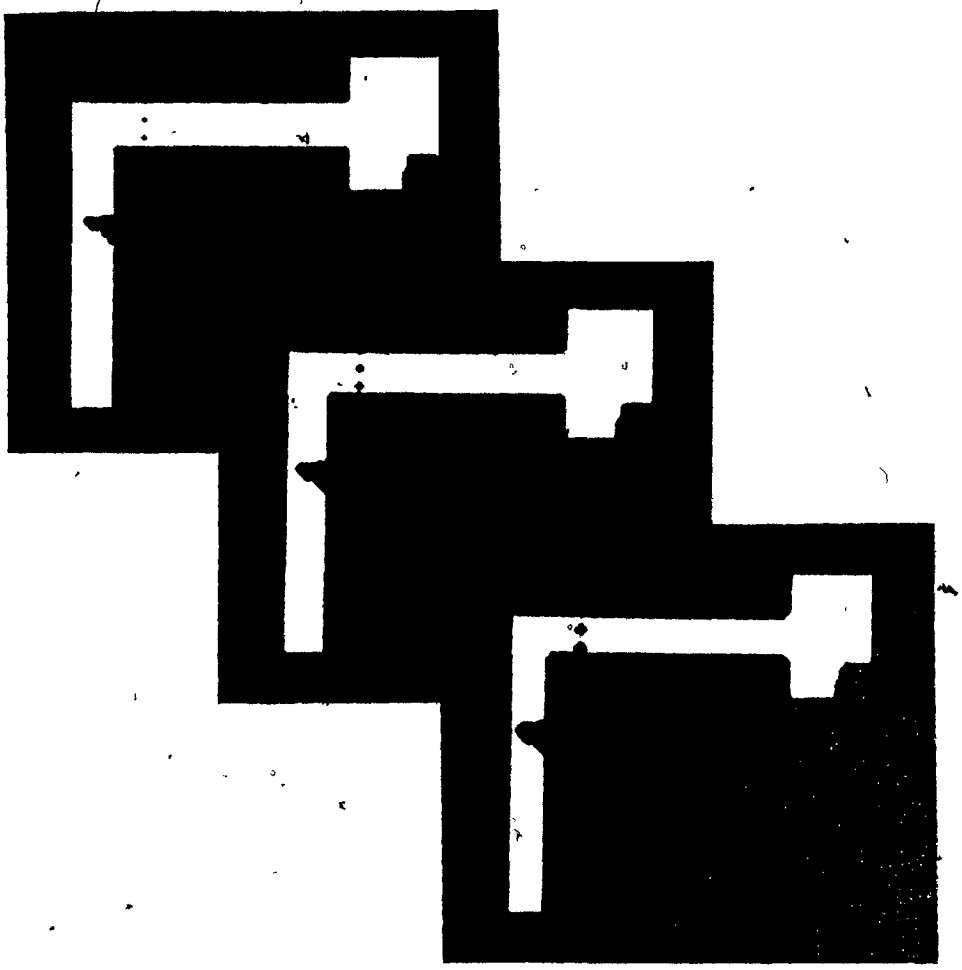


Figure 4.7d Contraction. The original figure is contracted by 3 pixels.

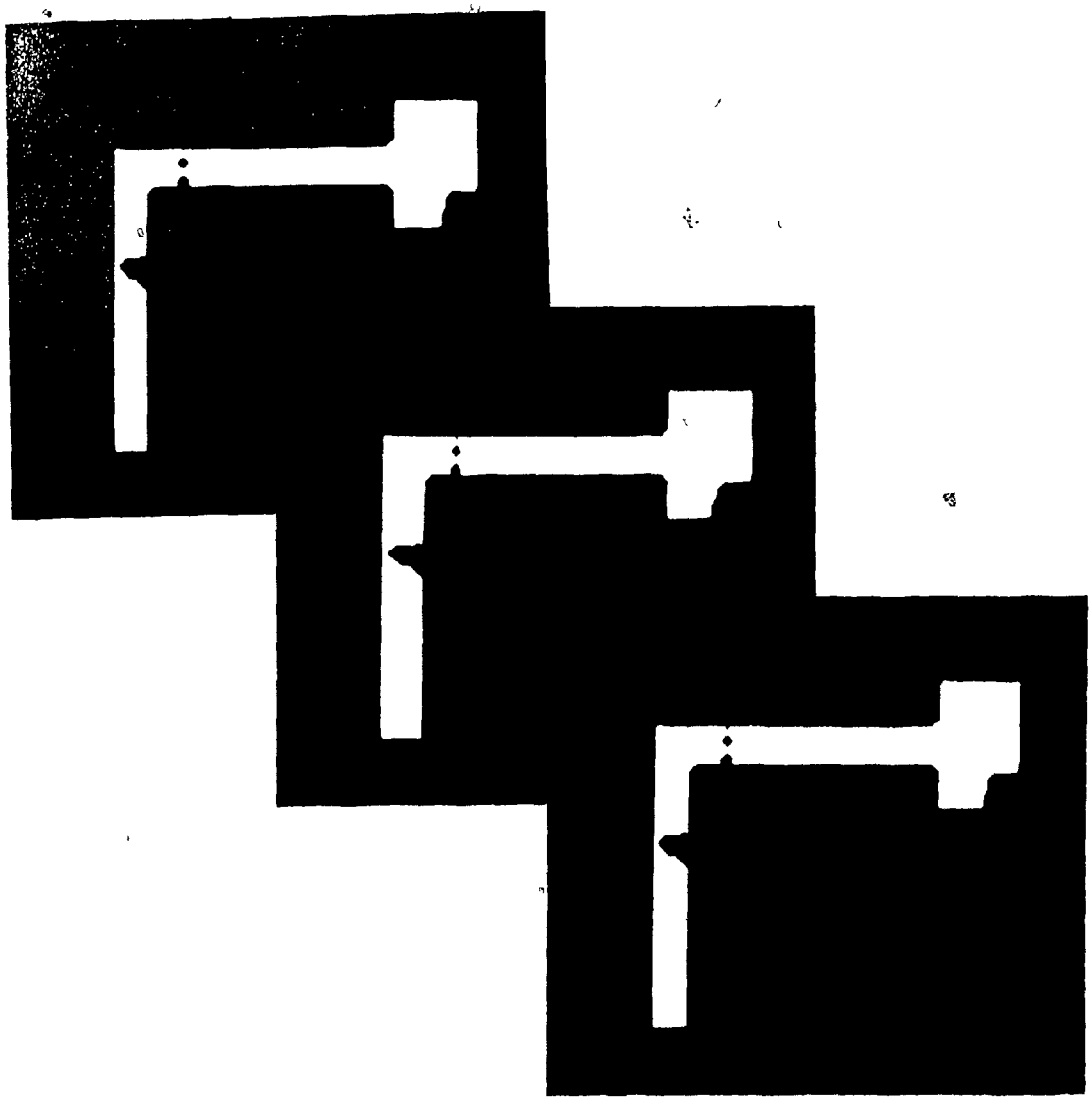


Figure 4.7e Expansion The contracted figure is expanded by three pixels The red regions show the added pixels

Figure 4.7f The result of the contraction-expansion process The red areas show the differences between the original figure and the one created by contracting and then expanding the former

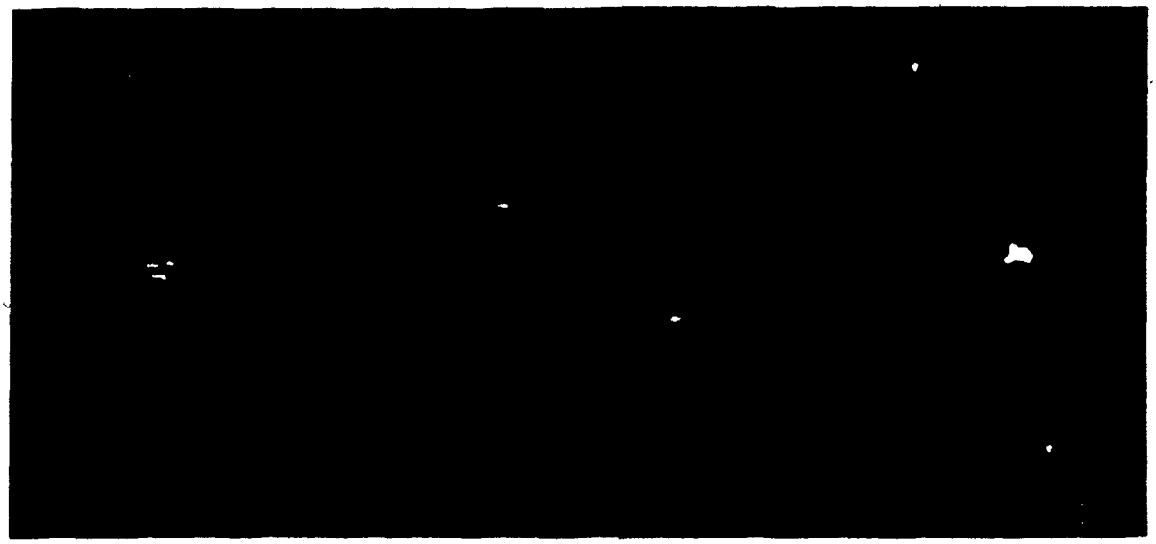


Figure 4.7g Combined result by exclusive-OR ing the two intermediate results seen in 7c and 7f and the error map obtained by comparing this result with the original image

Figure 4.7 The EXPANSION-CONTRACTION method

Figure 4.8 Illustration of false alarms at sharp corners

or simple convex polygons Such an operation would not give a great deal of information about the circuit s layout Rather what is required is information about the functional characteristics of any point of the circuit The PARTITION routine accomplishes this in five distinct steps

1. Corner detection
2. Primary segmentation
3. Secondary segmentation
4. Simplification of polygons
5. Classification of polygons

Corner detection. The original input is a binary image of the circuit Corners are detected with a modified version of the algorithm described in [Freeman77a] and [Freeman77b] ([Levine85 Ch 10] gives a good summary of the algorithm) First the boundary of the polygons in the image is found by the method described in the section about the EXPCONTR module. Once this is done, the boundary is known as a 'chain' of pixels. The basic idea in the detection of the corners of this chain is to 'move' a line segment of a certain length (say five pixels) along this chain and record the direction changes on the segment Once the whole boundary is scanned like this, values exceeding a certain threshold in the direction change function are chosen as corners. The reason a line segment is

used instead of just measuring the direction changes from pixel to pixel on the boundary is to avoid detecting false corners due to digitization. Figure 4.9 illustrates how the method works and shows a sample circuit with detected corners.

Primary segmentation. The *primary* and *secondary* segmentations use the data obtained by the corner detection algorithm to divide the circuit into simple polygons with distinct functional characteristics Let us define a unit distance as the width of a conductor line Figure 4.10a displays a hypothetical circuit where the segmentation is illustrated

Corners which are distant from each other by a unit distance (times a tolerance factor) can be connected to each other and thus form dissection lines which divide the original circuit. Corners $a - b$ and $c - d$ in the figure are such pairs The search starts from the bottom leftmost corner. Corners are skipped until one which has an inner angle of more than 180 degrees is reached (corner b is the first one) Among the remaining corners one which is close to that one by the unit length is searched (a is the one close to b). Once the connection is made, the smaller polygon is chopped off from the original one and the latter is updated. The algorithm is applied recursively to newly formed polygons because polygons which are chopped off can have further segmentations (eg when c and d are connected, the polygon which is formed can be dissected further at points $e - f$) Applying the algorithm only to corners with an inner angle greater than 180 degrees is useful in two ways· First, it drastically reduces the computation time since every connection must be made between two corners at least one of which has an inner angle of greater than 180 degrees. Thus a search to connect say the corner *start* to any other corner is a loss of time; second, it prevents connections outside the circuit, eg, between g and h

Secondary segmentation. At the end of the primary segmentation, there are some corners which could still be used as dissection points, by connecting them not to other corners this time, but to points on boundary lines Points i, j, k and l are such points. Such connections are called secondary segmentation. The search for these connections is done by extending either of the two lines forming a corner such as i (again with an inner angle of greater than 180 degrees), and trying to find an intersection point with any of the other segments ($m - n$ in the case of i) provided that the intersection point lies within the endpoints of the segment and that the distance between the corner and the intersection point is again within the unit distance limit.

Simplification of polygons. Most of the polygons formed by the above process are expected to be simple convex polygons but may appear to have many more corners forming slightly jagged lines because of the inaccuracy in the corner detection. At this step the

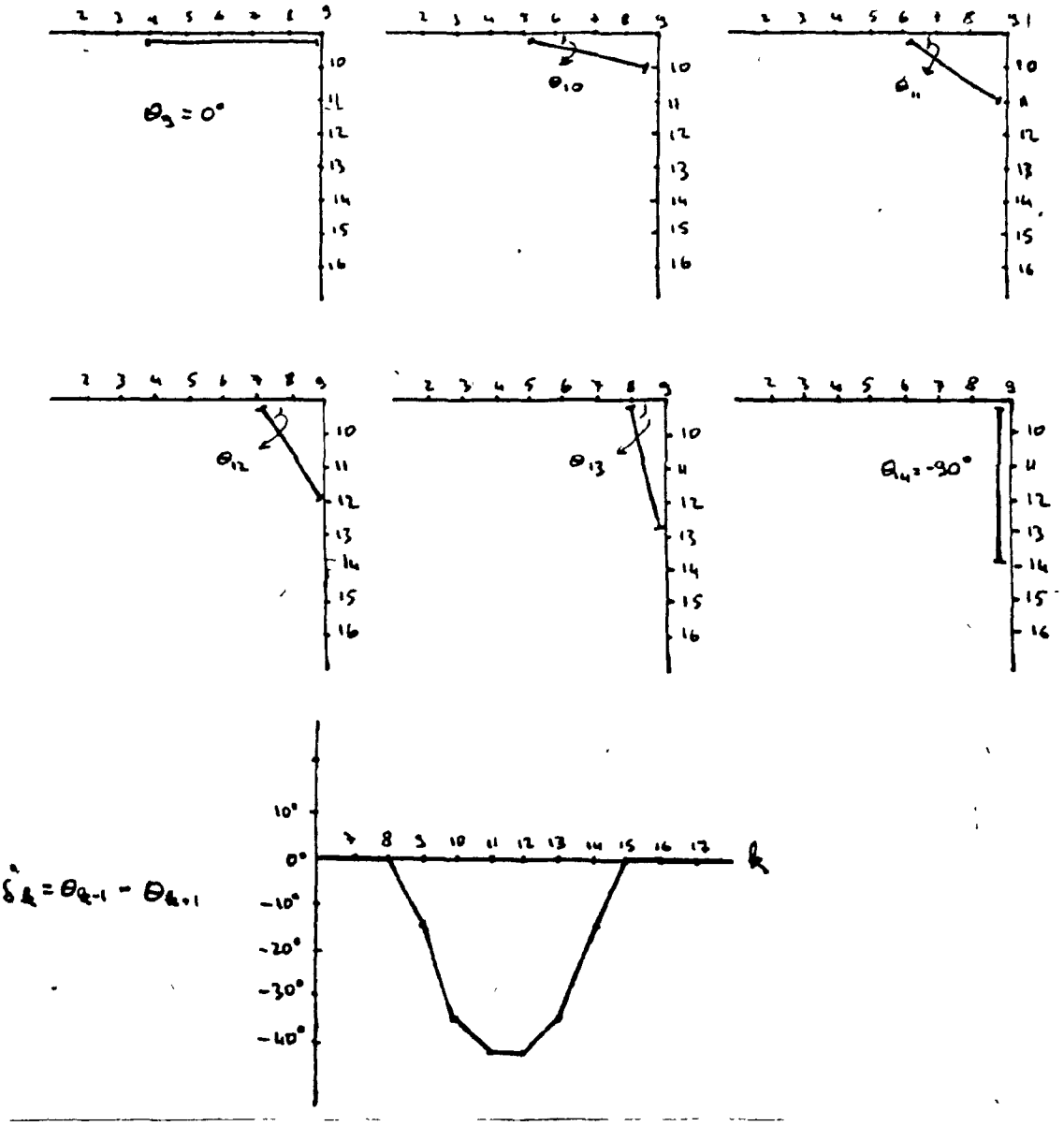


Figure 4.9a Corner detection by incremental curvature The incremental curvature δ_k for a 90 degree turn The line segment used to measure the curvature has a length of 5 pixels

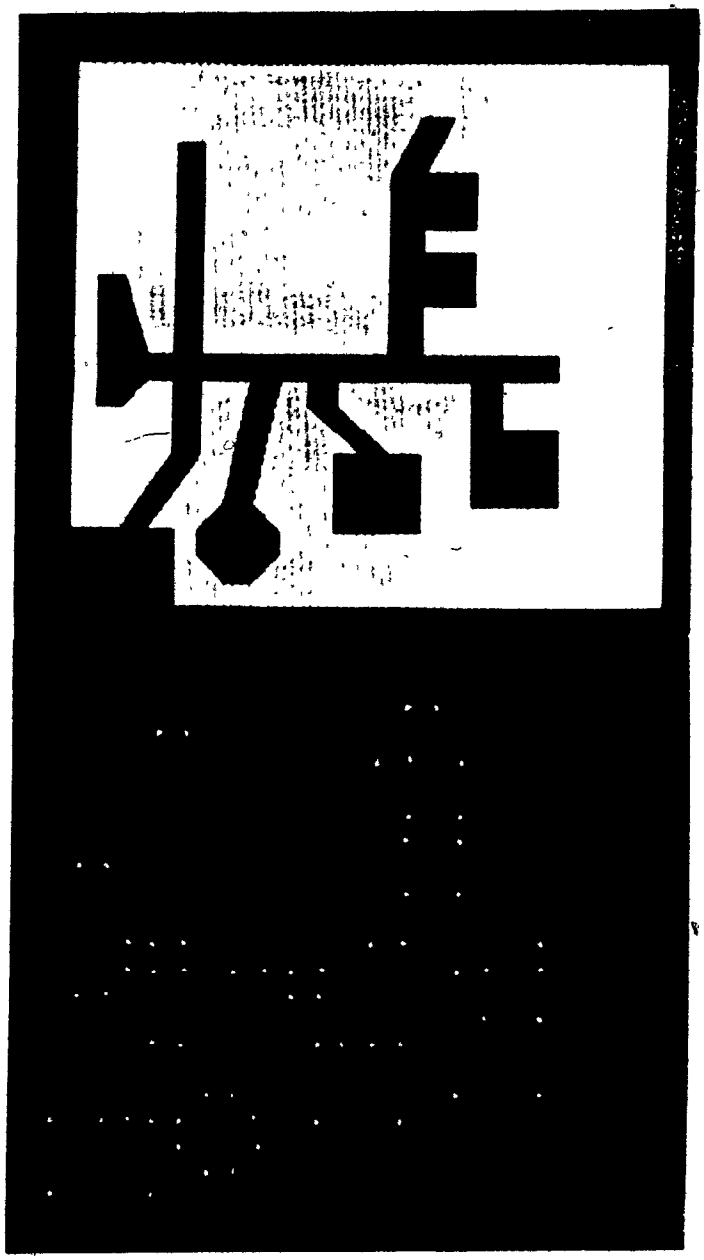


Figure 4.9b A hypothetical circuit and its detected corners

polygons are simplified in order to be able to classify them (see Figures 4.10d, 4.10e and 4.10f). The simplification is done by deleting any corner which has an inner angle very close or equal to 180 degrees and connecting its two neighbours. This way, for example the middle long rectangle-like shape of the test pattern in Figure 4.10e will be simplified into a rectangle (or rather a quadrilateral with inner angles close to 90 degrees) Figure 4.11 is an example of part of an actual circuit partitioned into simple polygons.

Classification of polygons. Once the simplification is made, the polygons can be

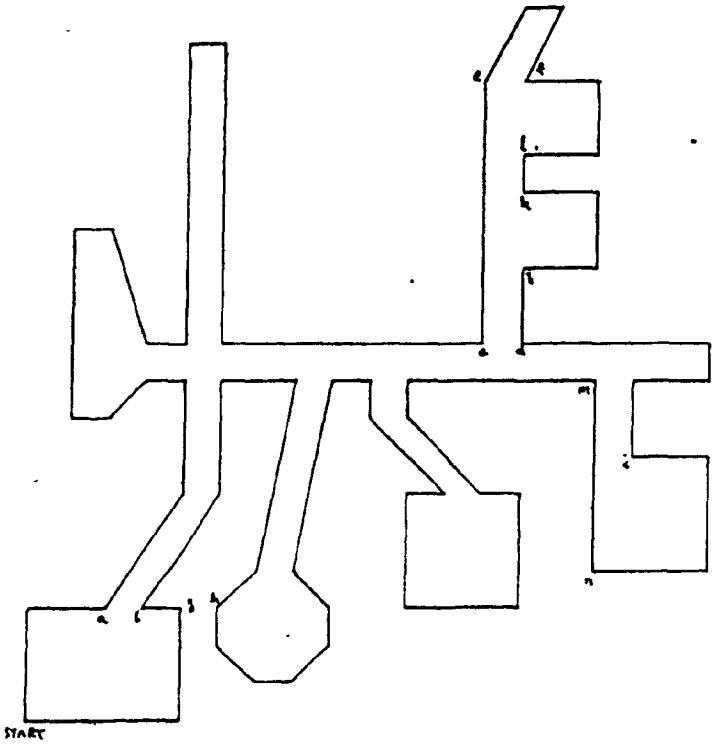


Figure 4.10a A hypothetical circuit input to the computer as a list of corners

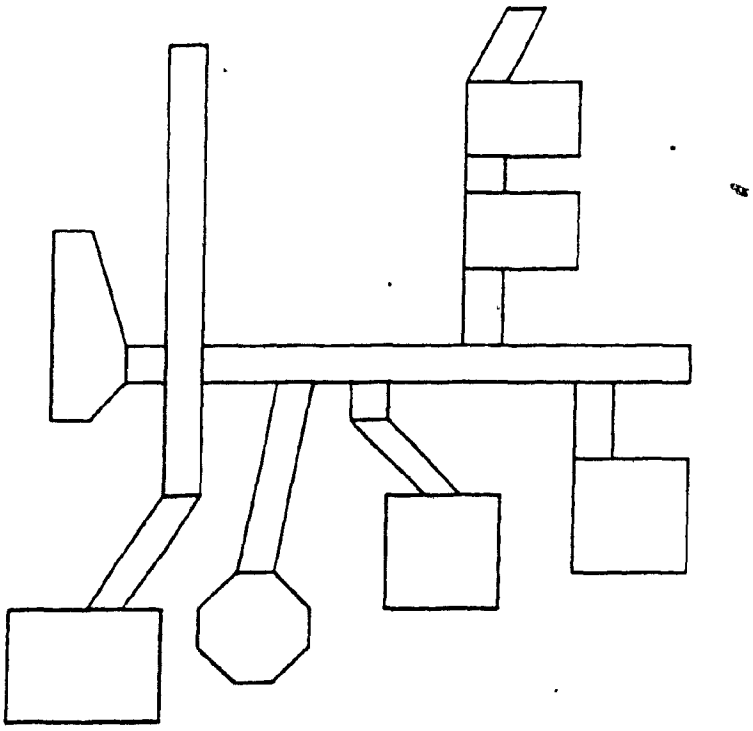


Figure 4.10b Segmentation of the circuit in Figure 4.10a

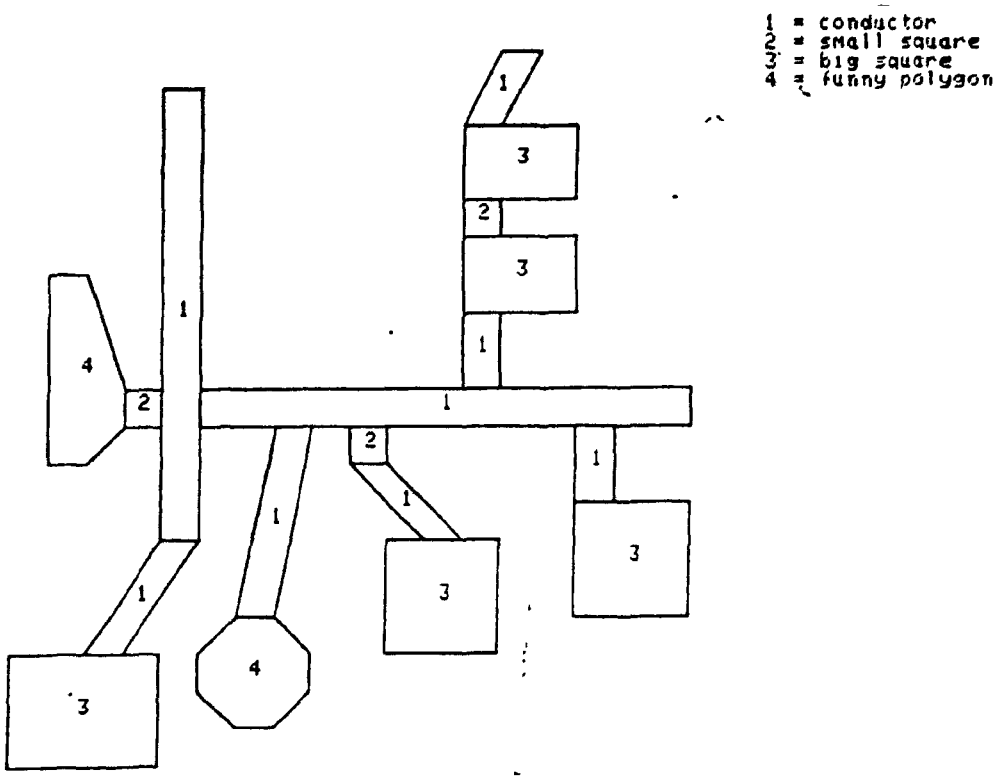


Figure 4.10c Labelling of the polygons found in Figure 4 10b

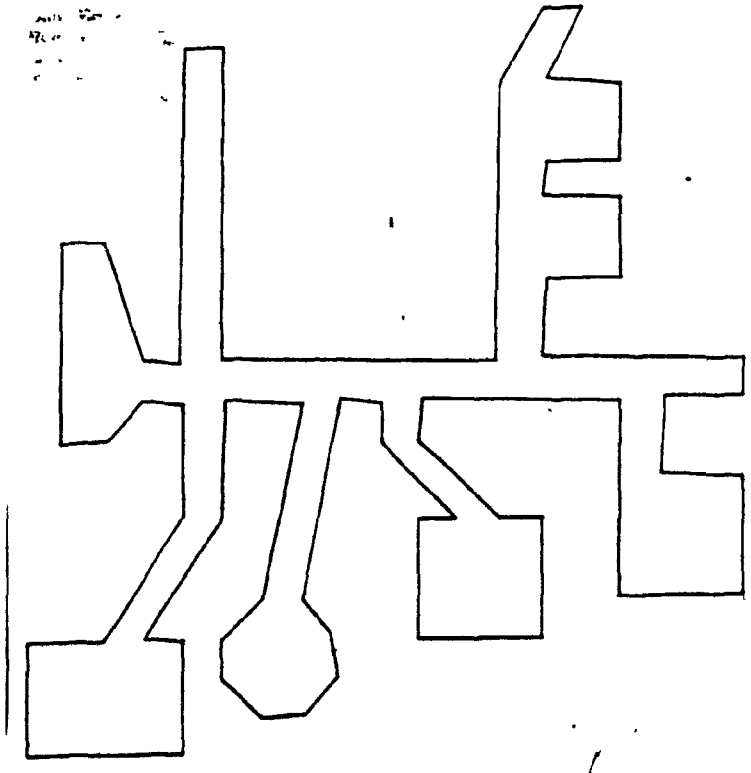


Figure 4.10d Same circuit input as a binary image. The corners are detected with the method described in Figure 4.9 Notice the jagged lines due to the inaccurracy of the corner detection.

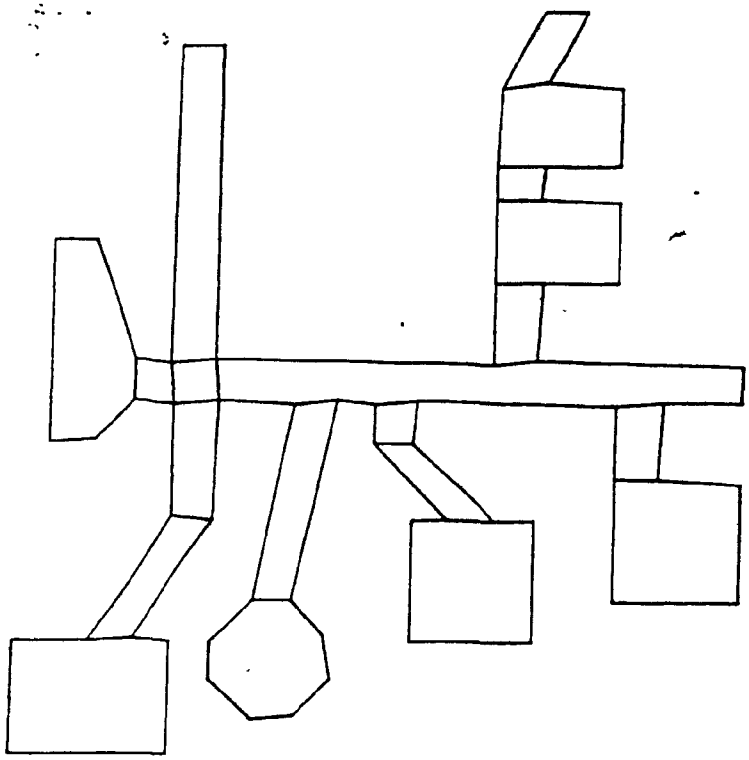


Figure 4.10e Segmentation of the circuit shown in Figure 4 10d

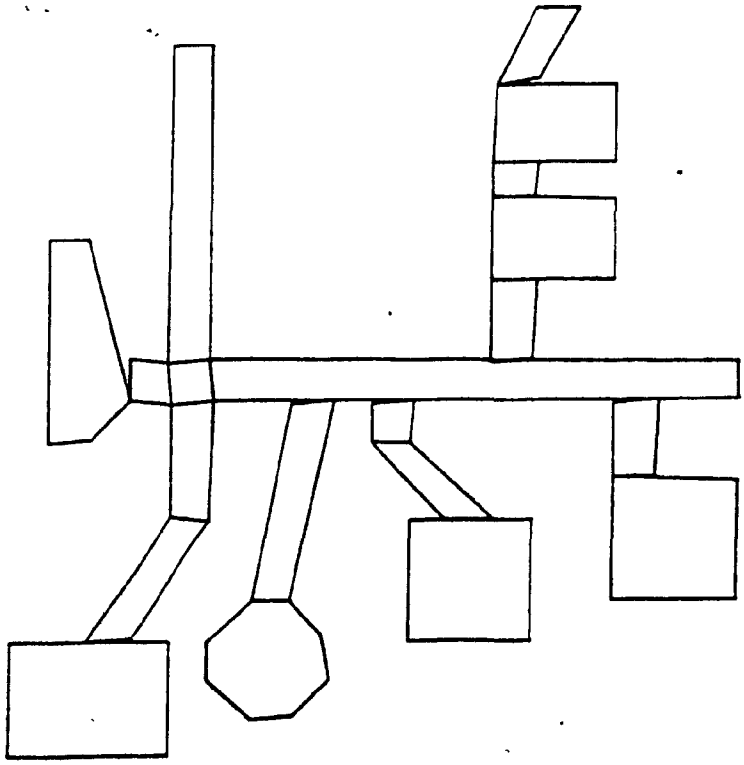


Figure 4.10f Simplification of the polygons found in Figure 4.10e

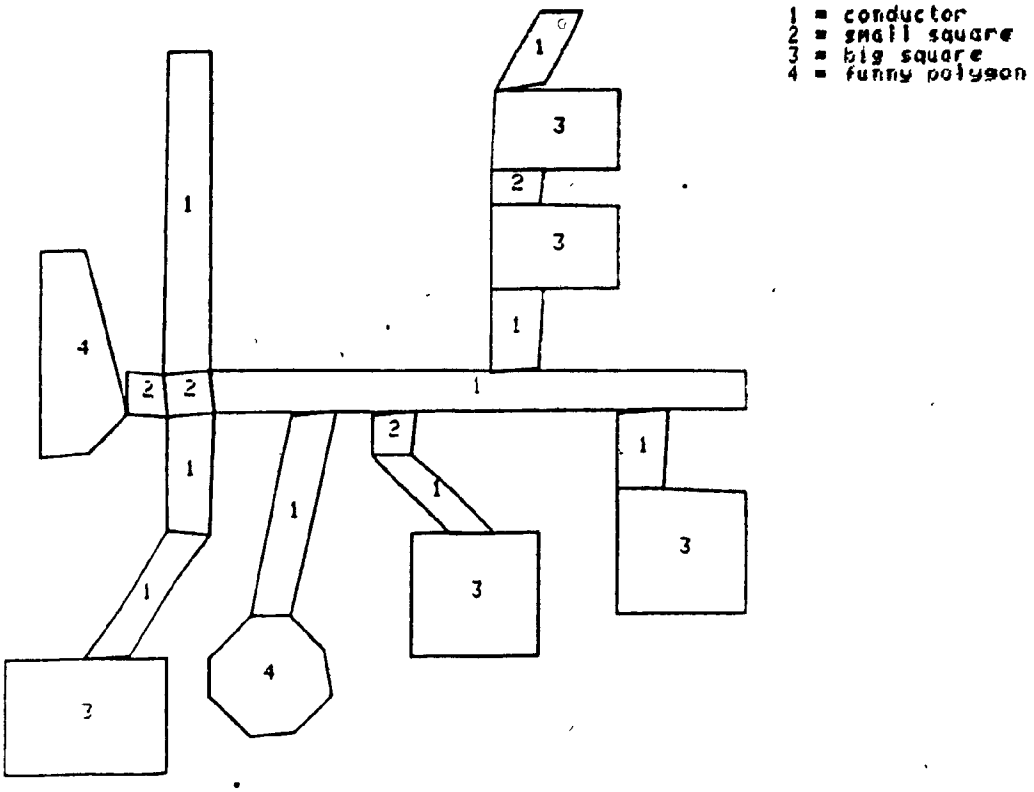


Figure 4.10g Labelling of polygons found in Figure 4 10f

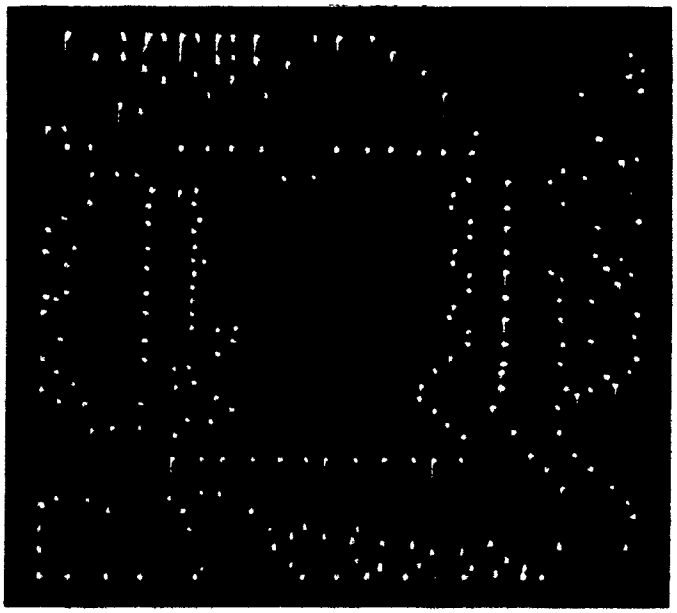


Figure 4.11a Corner detection on a sample circuit.

classified into one of four classes.

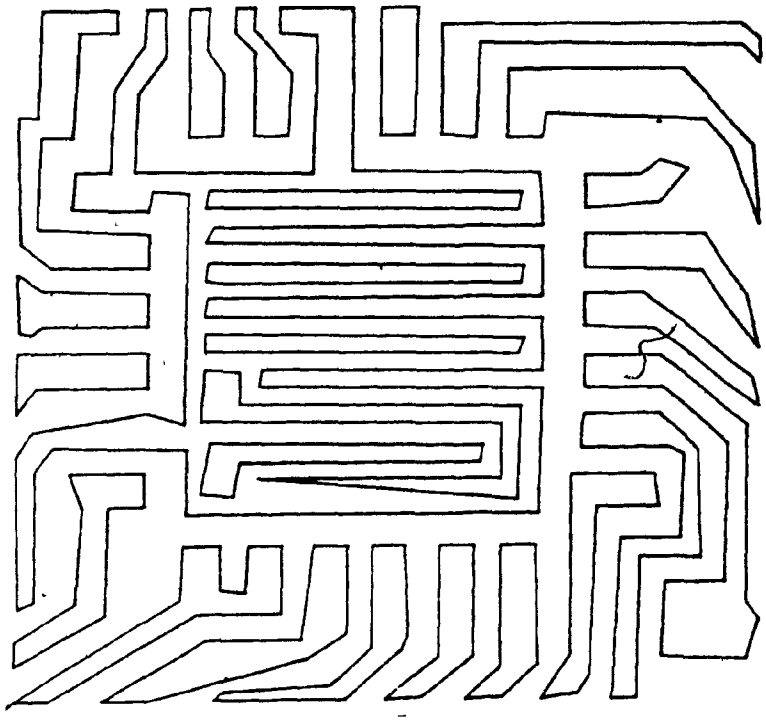


Figure 4.11b Same circuit as in Figure 4 11a drawn by connecting detected corners

1. conductor line
2. small square
3. big square
4. other polygon

The first three are supposed to be quadrilaterals. A *conductor line* is defined as a quadrilateral which has a width of unit distance and a length at least 1.5 times larger than the width. A *small square* has its width approximately equal to its length and also approximately equal to the unit distance. It is a part of the circuit which connects two other parts together like two conductor lines, two probe pads, etc. A *big square* has its width approximately equal to its length but they are both larger than the unit distance. A big square is likely to be a probe pad or a solder pad. The remaining polygons are the ones which cannot be classified as one of the three above. They either have more than four sides or do not look like rectangles. These are most probably parts of the conductor circuit which are there because of design aesthetics and do not have any electrical functionality.

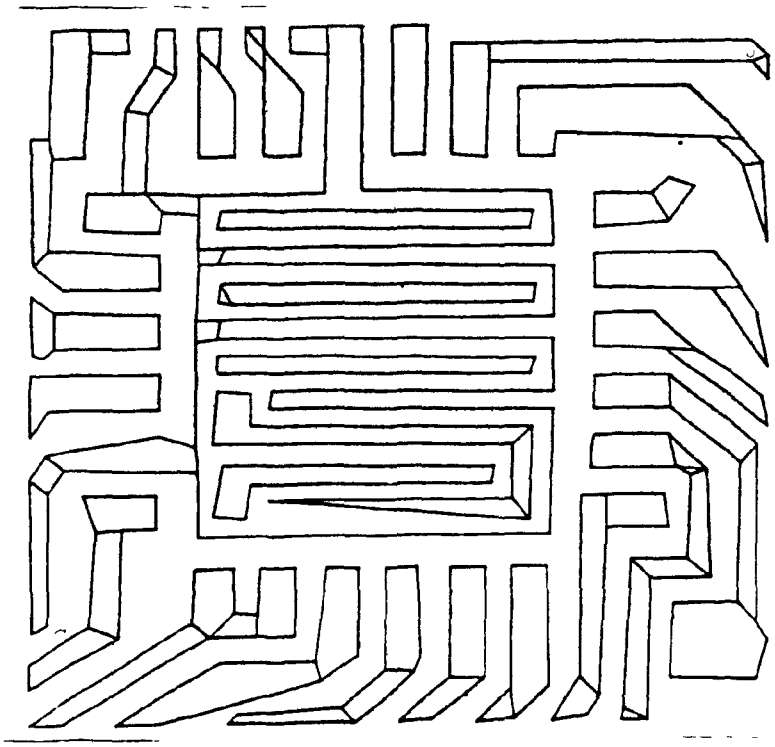


Figure 4.11c Segmentation

Examples of the different types of polygons described here can be seen in Figure 4 10g

Once the polygons are defined. they are labeled with numbers. and a set of feature data about them is passed to the inspection system. The data includes label. corner coordinates center. area. and type of polygon

4.2.1.4 The LABEL module

The LABEL module is a simple routine which takes as its input the defect map that either EXPCONTR or MATCHER has generated. and translates it into a list of data For each defect which appears as a connected cluster of pixels. it generates the minimum and maximum coordinates of the enclosing rectangle. the center and its area. The LABEL module computes this data by finding the boundaries of the defects (the same way as the EXPCONTR module finds the boundaries of polygons) and then performing computations on these boundaries.

4.2.1.5 Output comparison of the modules

The routines MATCHER. EXPCONTR and PARTITION provide sufficient data to the

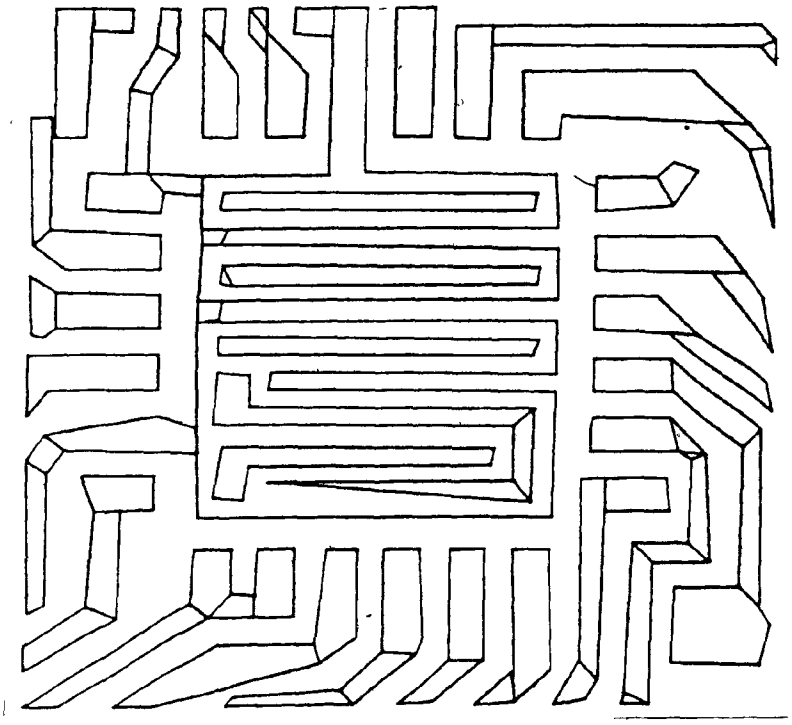


Figure 4.11d Simplification of polygons

inspection system for a good search and analysis of defects on any layer The MATCHER and EXPCONTR modules are suitable for finding defects of a different nature Nevertheless, both of them will give a certain level of evidence at locations which most likely contain defects. Therefore, the inspection system has the possibility of questioning both methods before arriving at a conclusion regarding a particular defect. This largely eliminates the occurrence of false alarms.

The MATCHER program is most suitable for finding defects involving relatively large areas. Because of digitization errors and slight imperfections during the production in reproducing the screen on the circuit, small unimportant mismatches might occur between the circuit and the screen. These are not important errors and they are deleted from the output of the MATCHER routine. However, errors covering a large area, such as a missing part of the circuit, an extra printed region due to too much paste, missing parts because of cracks or lifts will definitely be detected by the matching mechanism. The method will also detect some of the smearing, sputtering, pinholes, etc., adding stronger evidence to the findings of the expansion-contraction program.

The EXPCONTR module on the other hand, does not rely on any correspondence between the circuit and the screen, and is therefore more suitable for finding actual small

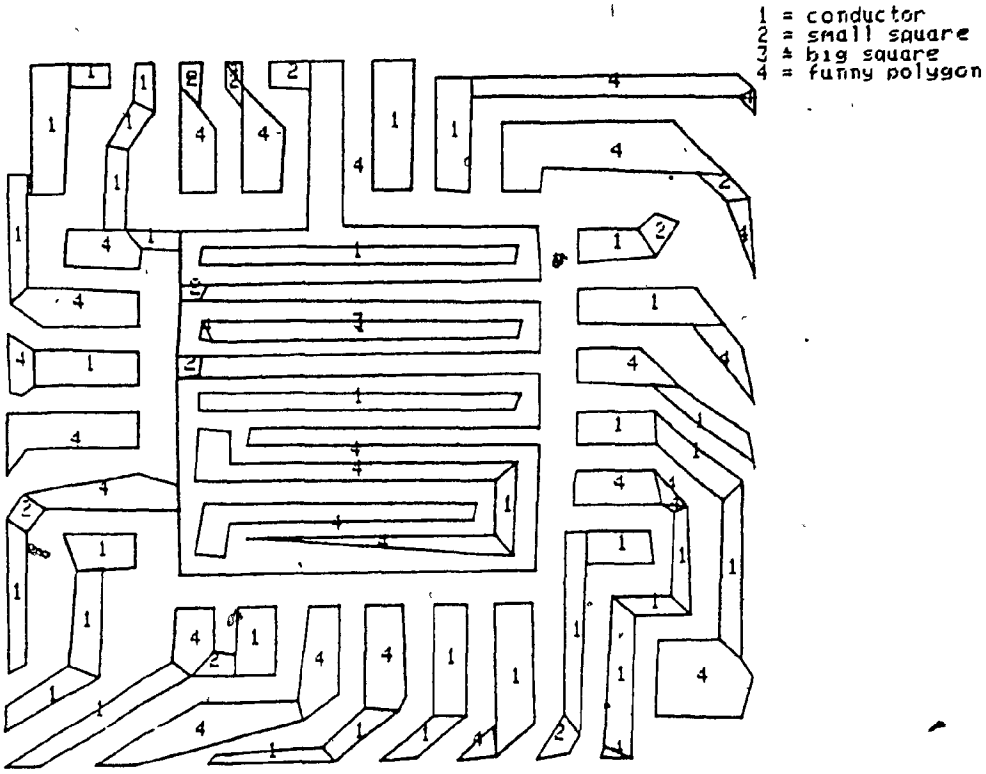


Figure 4.11e Labelling of polygons

defects as opposed to small unimportant discrepancies between the two. It will definitely detect cracks, pinholes, smearings, sputterings. It will also give evidence about larger defects since these generally tend to have jagged and therefore high frequency edges This evidence, combined with the evidence coming from the MATCHER program, have been largely sufficient for an in-depth analysis of defects. As stated earlier, once the defects are located, it is imperative to find out where with respect to the circuit's various components the defect actually occurs. This is accomplished using the data coming from the PARTITION program

4.2.1.6 Separation of Layers

An important problem in computerized visual inspection of a multi-layer circuit such as a hybrid circuit is the visual separation of the different layers. Since bottom illumination is used in this thesis, colour information cannot be used for such a separation. Instead, we use the fact that all layers are either translucent or opaque. After the printing of every layer, a bottom illuminated picture of the circuit is taken. The histogram of the picture is computed and its peaks are related to various circuit layers. For example the peak around black indicates the conductor lines and the resistors, since both of these are

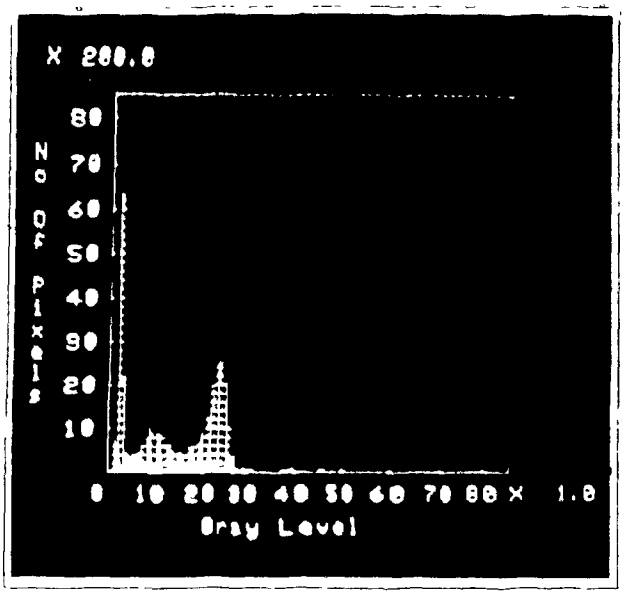


Figure 4.12 Histogram of a circuit with conductor and dielectric layers (the circuit can be seen in Figure 5 4a) The first peak corresponds to the conductor layer which is the darkest layer The other two peaks correspond to the dielectric layer and bare substrate respectively

opaque The peak with the highest grey level corresponds to the substrate which is the most translucent of all of the components of the circuit In between these are two peaks indicating the dielectric layer and the overglaze layer The data obtained from the histogram is used in conjunction with the picture of the same circuit taken before the most recent layer was printed For example when the first conductor layer gets printed on the substrate the histogram of the image is computed and the image is thresholded at the only valley value This gives a binary image of the conductor When the dielectric layer gets printed on top of the conductor the histogram of the image has three peaks and two valleys The middle peak indicates the dielectric The image is thresholded at the two valley values and the dielectric is assigned a certain grey level this value indicates the presence of dielectric material on the circuit (see Figures 4 12 and 5 1)

But since some of the dielectric is hidden by the conductor a direct comparison of this image by the MATCHER module can be misleading Therefore information from the picture taken before the dielectric was printed is necessary to confirm the output from the matcher program. If a region is indicated as being a defect by the MATCHER program and the image of the conductor layer does not have a conductor strip over that region, than the region is really a defect, otherwise it is just hidden under a conductor The same reasoning applies to output from the EXPCONTR module. The same approach also applies to other

layers every time part of a layer is printed over part of another layer which is opaque

4.2.2 The Production System

The modules described above do the fault detection work and also take most of the computation time But what makes the inspection system 'intelligent' is the rule-based production system which uses the data provided by the modules The production system is the 'brain' of the operation It accomplishes two tasks managing the whole operation by running the right program, issuing the right command or message at the right time, and analyzing defects The next two sections will go into the details of both tasks

4.2.2.1 General Management

A specific succession of actions must be performed for the visual inspection of every hybrid circuit

1. Visual inspection of the circuit after the printing of every layer
2. Report of the circuit s state to a general database, derivation of statistics from the collected data
3. Generation of error messages for defects on the circuit as well as mishaps in the operation
4. Provision of interaction with a human operator whenever necessary

A rule-based production system is more suitable for such tasks than a procedural computer program.

The management part of the production system in this thesis performs the following tasks: inquires whether the inspection consists of a new circuit, in which case extraction of markers and other initializations are necessary; or a known circuit, in which case the data already stored about the circuit are used.

The system then proceeds to obtain images of every layer, and runs the necessary fault detection programs. In the process, whenever necessary, it stores the data for future use. Then it applies the inspection rules to analyze the data obtained from the previous steps. During the analysis stage, the system requires stored information about the circuit, such

```
(p doPartition
   (goal
    ^analysisType 1                    , teaching mode
    ^layerType    conductor
    ^partition    notDone)
-->
   (write (crlf) doing the partitioning (crlf) )
   (call partition)
   (modify 1 ^partition    done
             ^analysisType 2 ))
```

Figure 4.13a Example of control rule If we are in the teaching mode and the layer under inspection is a conductor layer and partitioning has not been performed yet perform the partitioning

as the image of the screen used to generate the particular layer, or the image of the previous layer At various stages, the operator can display visual information on a monitor or numerical information on a terminal (see Figure 1 7) The system is supposed to communicate with a robot repair system running simultaneously, but this communication is part of another project in our laboratory At this point, our system gives visual and numerical information as well as an English description about defects and stores the information if required

The OPS5 production system interpreter is used for the implementation of the system The next chapter will give more information about the programming aspects of the project. What should be noted here is that due to the almost English like style of the OPS5 rules, and to the fact that is is easy to modify a production system adding or modifying any rules to the system is quite simple This comment applies to general management rules as well as to defect analysis rules Therefore, if a new layer gets added to a certain type of hybrid circuit, or if more interaction is required for some reason, or if some inspection rule gets modified because of industrial needs, the addition or modification of rules should prove to be trivial

4.2.2.2 Defect Analysis

The defect analysis rules in the rule-based system are direct translations of the industrial inspection rules into OPS5 productions. Since the data about the location, size, area, orientation of defects is fed to the system by the inspection modules, the productions' only task is to 'translate' the inspection rules into the OPS5 language and then apply them to the defects. As stated above, modifying or adding rules is a relatively simple task. Figure 4.13 gives examples of defect analysis rules as well as general management rules.

```
(p areaOfFaultOnPad
   (apply rules)
   (loopIndex ^value <index>)
   (statusWord ^circuitPart 3          ; a fault found on a probe pad by
               ^testType {>= 2 <= 3} ) ; the matcher program (or both)
   (partDescription ^25%area <x>)
   (faultValues ^area >= <x>
                ^centerX <centX>
                ^centerY <centY>)
-->
   (remove 1)
   (write (crlf) area of fault number <index> at location
      <centX> <centY> is more than 25% the area of pad (crlf)) )
```

Figure 4.13b Example of analysis rule If a defect is found on a probe pad by the MATCHER program and if the area of the defect is greater than 25% of the area of the pad issue an error message

4.3 Conclusion

As stated earlier one of the great advantages of using a rule-based system for an inspection process is the ease by which one can add or modify rules In its current state the system described in this chapter contains the necessary control and inspection rules for the inspection of the two-dimensional layers of hybrid circuits These rules can be altered to suit the criteria of different manufacturers and the requirements related to different types of circuits

Currently, the system contains approximately twenty control rules These are related to the selection of programs to run, images to read in, and decisions about which inspection rules to apply in different cases There are three or four inspection rules per manufacturer s inspection criterion The reason there is not a one-to-one correspondance between inspection criteria and inspection rules in the production system comes from the limitations of the OPS5 production system interpreter in performing arithmetic operations on the left hand side of rules ([Forgy81])

Chapter 5 Application Examples and Conclusion

5.1 Introduction

This chapter will give some detail about the programming languages and techniques used in the implementation of the thesis It will then show some application examples and finally will end with suggestions for industrial hardware requirements and possible improvements

5.2 Programming Notes

As mentioned in Chapter 4. the production system interpreter used for this thesis is the OPS5 interpreter. written for a LISP environment [Forgy81]. The OPS5 interpreter has the advantage of accepting LISP routines for simple computations or communications It also has the disadvantage of not having sufficient memory space since it runs under LISP In fact. it was found impossible to load large picture arrays into the production system Therefore another high level programming language was necessary for the low level fault detection computations. VAX-FORTRAN. an improved version of FORTRAN-77 was used for the these computations. Again because of memory limitations. the inspection modules were designed as separate main programs. and the communication between them and the production system was done through mailboxes ([DEC82]). Since the only information passed from the FORTRAN programs to OPS5 was the numerical data about circuit polygons or defects (see Figure 4.1). this scheme solved the problem of memory limitations in OPS5. The inspection modules were written so that they could be run in two different modes: one mode is very interactive and gives the operator the possibility of displaying

different stages of the inspection and asking for numerical information. the other mode communicates only with the production system.

The whole system was developed on a Digital Equipment Corporation VAX-11/780 computer. Images were displayed on a Grinnell display system. The circuits were digitized using a WILD-M8 microscope. and the photographic transparencies which are used to create the screens were scanned on an Optronics P1000 drum scanner

5.2.1 The OPS5 production system interpreter language

The rule-based system written in the OPS5 production system interpreter language plays an important role in making this system 'intelligent'. This section will summarize the experiences and difficulties encountered during this thesis work in using the OPS5 interpreter

An OPS5 production system consists of *productions* which are formed by a *left-hand side* (LHS) and a *right-hand side* (RHS) The LHS is a group of logical conditions These conditions are related to the existence or absence of memory elements in the system's memory tables Values of variables can also be tested in the LHS conditions: e g whether the value of a variable is larger or smaller than the value of another variable [Forgy81] gives a detailed description of the LHS conditions Examples of rules are given in Figure 4 13 If all the conditions of the LHS of a production are satisfied. the production might be 'fired'. in other words. the actions described on its RHS might be performed. When more than one production have satisfied LHSs. the interpreter will chose one of the productions according to a scheme called *conflict resolution* Chapter 3 goes into a detailed description of conflict resolution strategies. OPS5 uses a strategy called LEX which is also described in Chapter 3. The actions described on the RHS are related to adding. removing or modifying memory elements. They can also involve input/output operations for communication with the 'world'

The OPS5 language is very versatile for implementing rule-based systems. An implementation of the rules of an expert system is done in an almost English-like manner. as can be seen in the examples in Chapter 4. Its conflict resolution features provide ways of getting around difficulties that would normally be encountered in programming the same expert system in conventional languages such as FORTRAN. For example. when a defect on a hybrid circuit is analyzed. there might be several plausible explanations about its nature. In a conventional language. finding the most plausible explanation would require explicit comparisons and implementations of decision making rules. In a production system

interpreter, all this is done by conflict resolution. if two or more LHSs are satisfied, the interpreter makes a decision about which production to fire.

Unfortunately, trying to implement conventional programming techniques in the OPS5 language becomes difficult. For example, in a sequential programming language a loop is very easy to implement, this is generally done with just one command. In OPS5, creating a loop becomes a major implementation problem; it takes at least three productions to simulate the behaviour of a loop, and the limits of the loop index have to be checked explicitly on every iteration. It could be argued that in an expert system a loop in the conventional sense is not needed, but in the case of a system like the one described in this thesis where general management and expert analysis are both performed by the same program, loops and other conventional programming techniques are necessary For instance, the OPS5 program used in this thesis analyses every defect as an expert system, but it has to perform the analysis on several defects: therefore it has to perform a 'loop' for every defect One possible solution to overcome this difficulty would be to use the OPS5 program as a subroutine callable from conventional languages like FORTRAN, therefore taking care of loops and other classical programming techniques from outside the OPS5 environment Such a technique was found impossible to implement due to the very different nature of FORTRAN and LISP OPS5

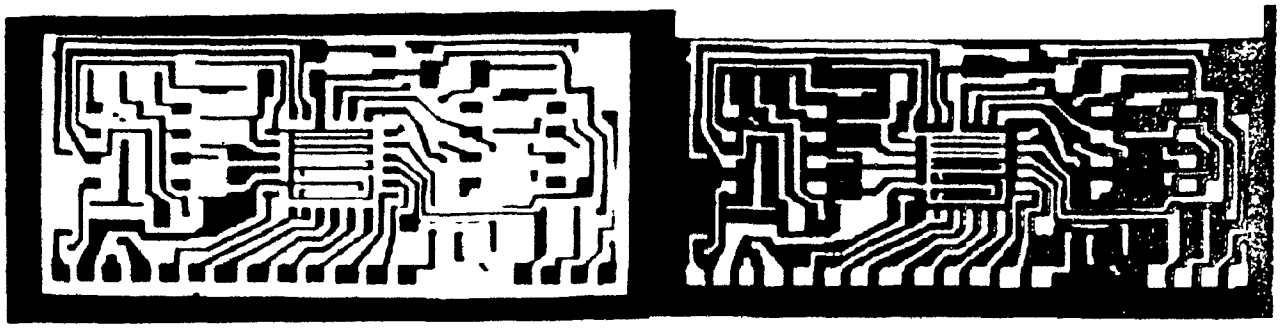
One other major weakness of the OPS5 programming language is its inability to perform simple arithmetic operations on the LHS For example, although implementing a rule which says "**if** $x \geq y$ **then** *action*" is very easy, one which says "**if** $x + y \geq z$ **then** *action*" requires writing two productions one of which forces the other to fire, and creating a new 'intermediate memory element. Forcing productions to fire is against the spirit of production systems, but is unfortunately needed in this case as well as in the case of implementing conventional programming techniques.

These and other difficulties encountered during the use of the OPS5 production system interpreter are closely examined in [Hong84].

5.3 Application Examples

The operations of the basic modules of the circuit have been illustrated in Chapter 4. Figures 5.1 to 5.5 give examples of the application of the system to real hybrid circuits.

Figure 5.1a shows the application of the MATCHER program to an actual circuit. First, the markers are extracted and the size and orientation of the screen found in the teaching



Screen

Extraction of markers

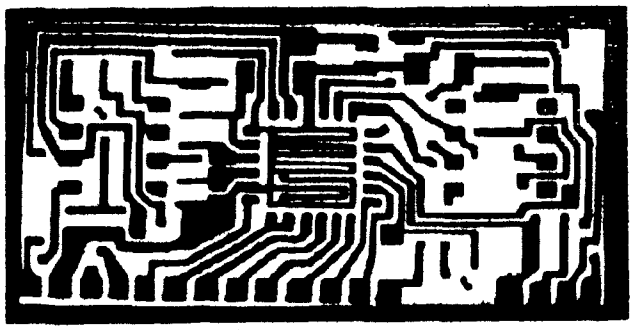
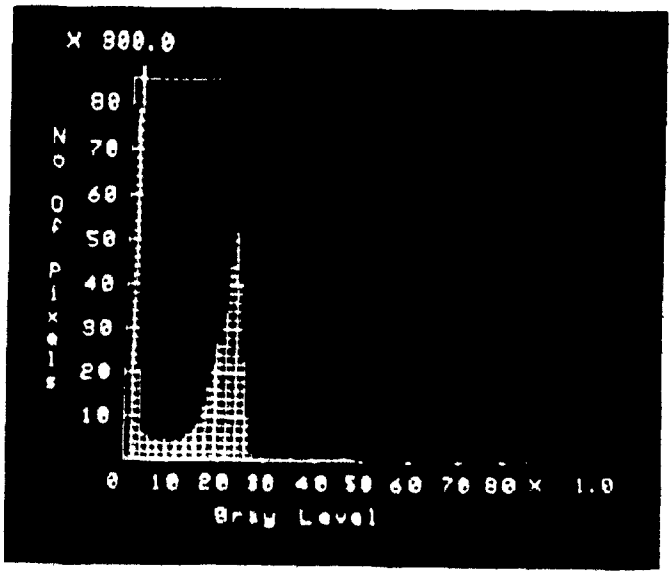
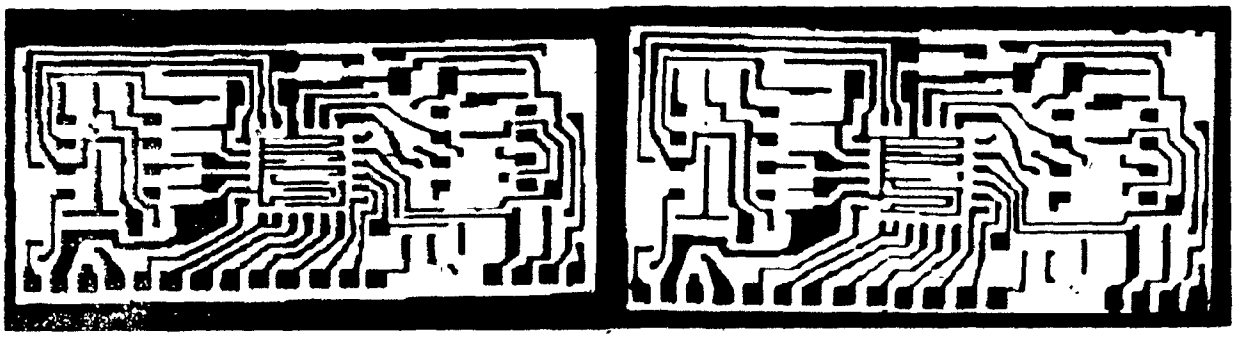


Image of circuit

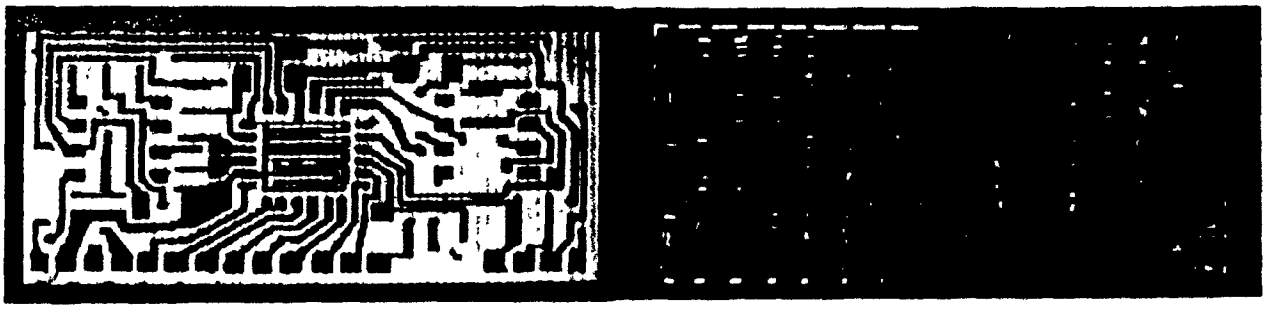


Histogram of the image



Thresholded image

Rotated and resized image



Match between circuit and screen

Difference between circuit and screen

Figure 5.1a Application of the matcher program to a real circuit The resulting image will be cleaned up and a perfect match will be found

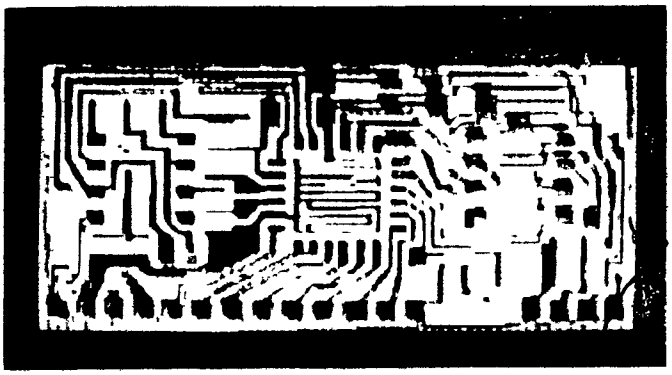


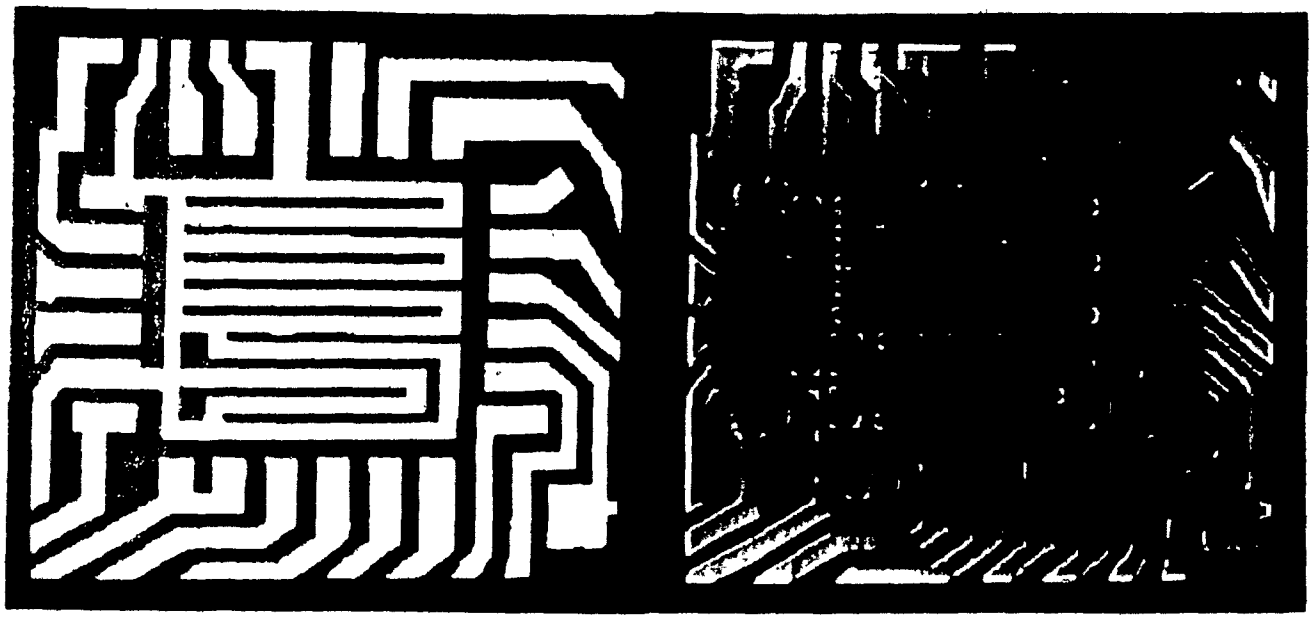
Figure 5.1b An example where the MATCHER program finds a mismatch between the screen and the circuit Note that although the markers correspond exactly the rest of the circuit is distorted This might be caused by a distortion during the printing or digitization processes

mode Then, the image is thresholded according to the valley found in its histogram Using the data from the teaching mode and the data obtained from the marker locations on the actual image, the circuit's image is rotated and rescaled Once this is done, the circuit and the screen are superimposed and the difference between the two is found This gives a map of the defects Figure 5 1b shows the result of the same operation on another circuit In this case, although the matching operation completes successfully it is observed that the circuit and the screen cannot be matched properly This is caused by a distortion either during the printing process or during the digitization process

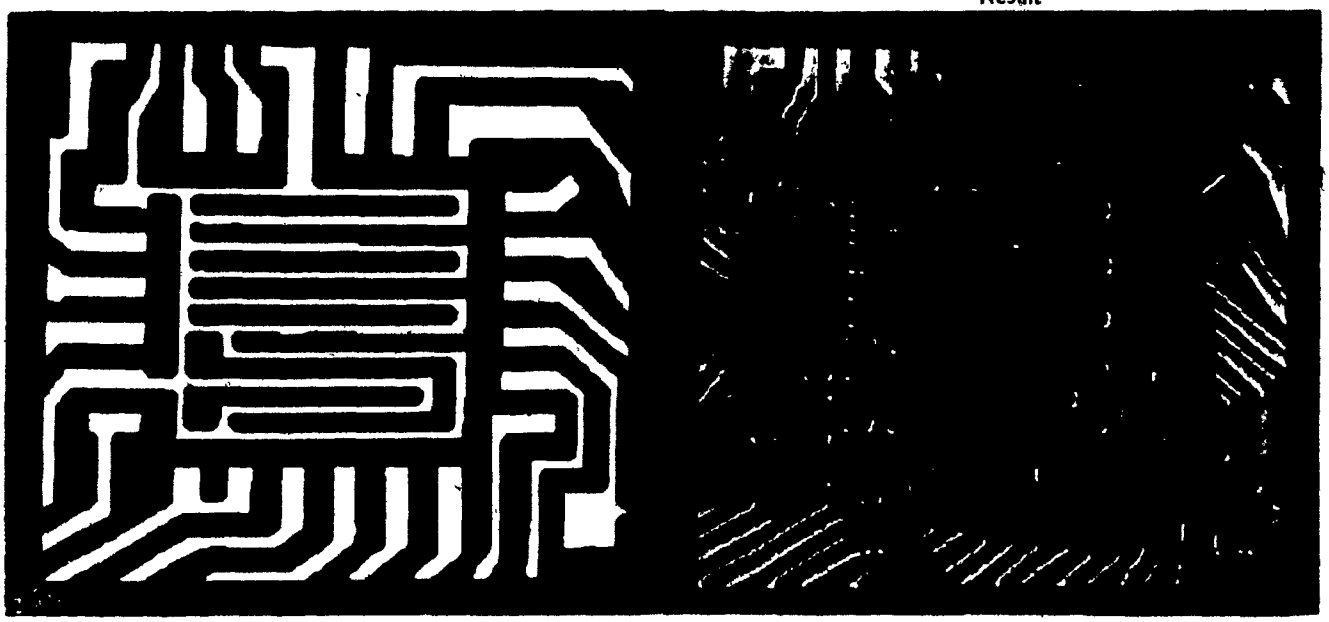
Figure 5 2 is an example of the application of the EXPCONTR program to a portion of a real circuit The defect map indicates parts of the conductor circuit which might be too narrow according to specifications An analysis of the defects by the production system will measure the conductor lines which contain the defects and will arrive at a final conclusion about the measurements

As mentioned in Chapter 4, additional layers are extracted from the image of the circuit by thresholding the histogram Figure 5 3 shows a hypothetical circuit where the dielectric is too narrow at the top right corner Once the image of the dielectric is extracted from the image of the whole circuit, the EXPCONTR program indicates the narrow region as a possible flaw. Figure 5 4 shows the same process on a real circuit

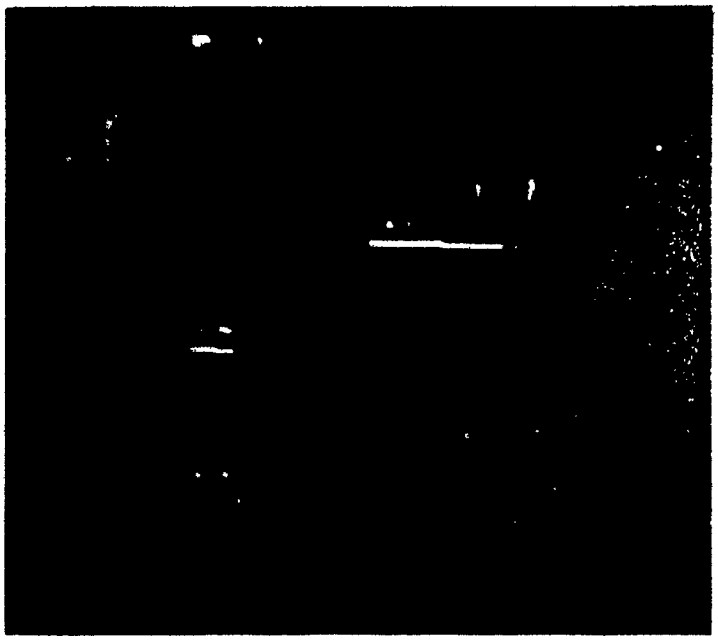
Finally, Figure 5.5 gives examples of defect analysis rules that give error messages At this point, this information is simply printed on the operator's terminal, but it could be used to fire up other programs which would control the robots and other devices to actually repair the circuits.



Expansion Result

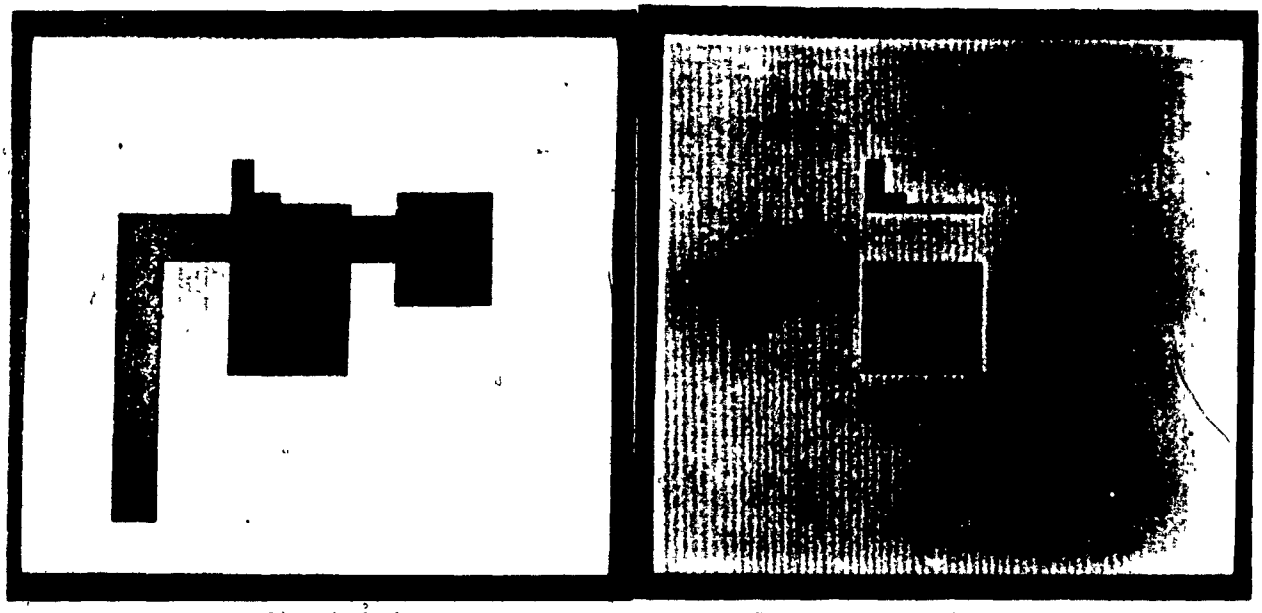


Contraction Result



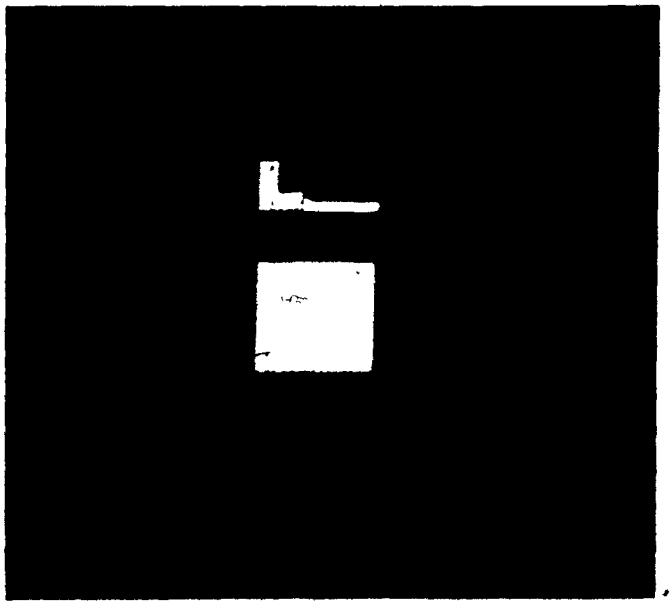
Defect Map

Figure 5.2 Application of the EXPCONTR program to a real circuit



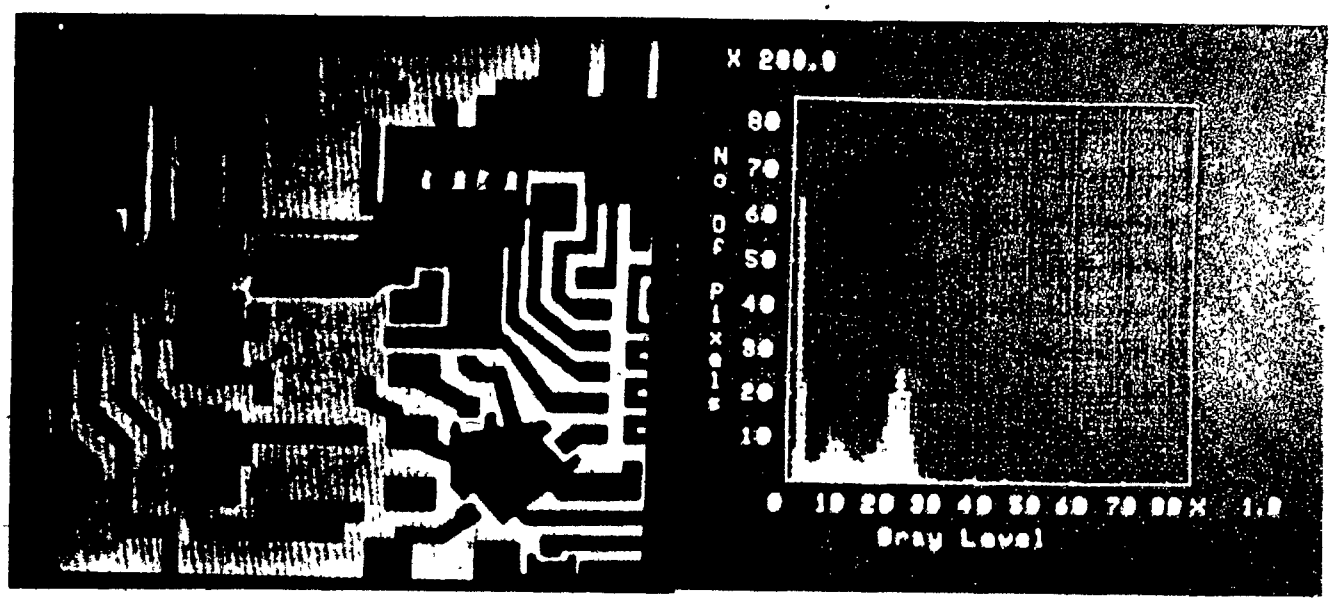
Hypothetical circuit

Extracted image of the dielectric layer



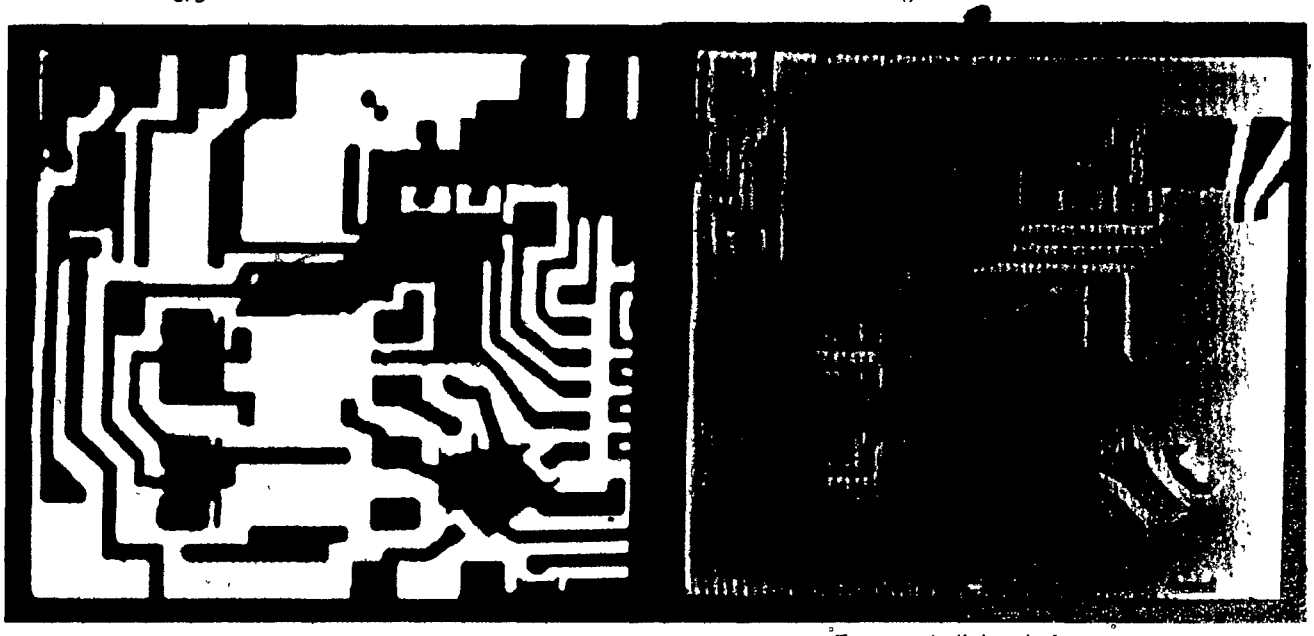
Detected defect

Figure 5.3 Detection of defects on the dielectric layer



Circuit with conductor and dielectric layers

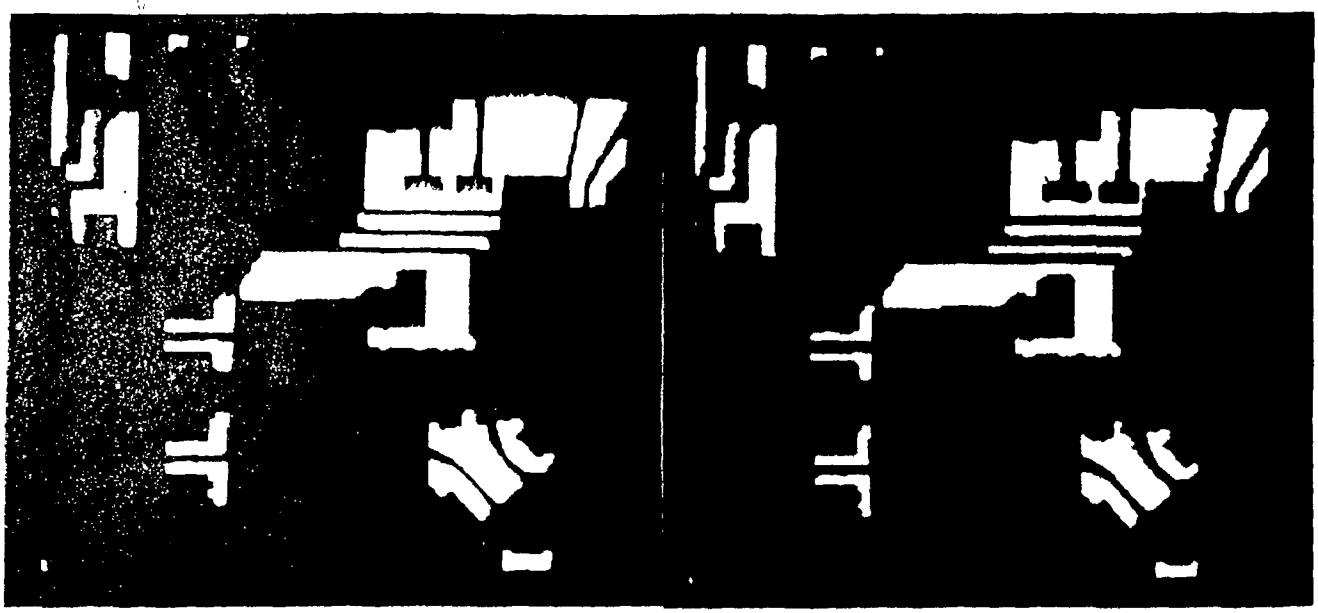
Histogram of the circuit



Thresholded circuit

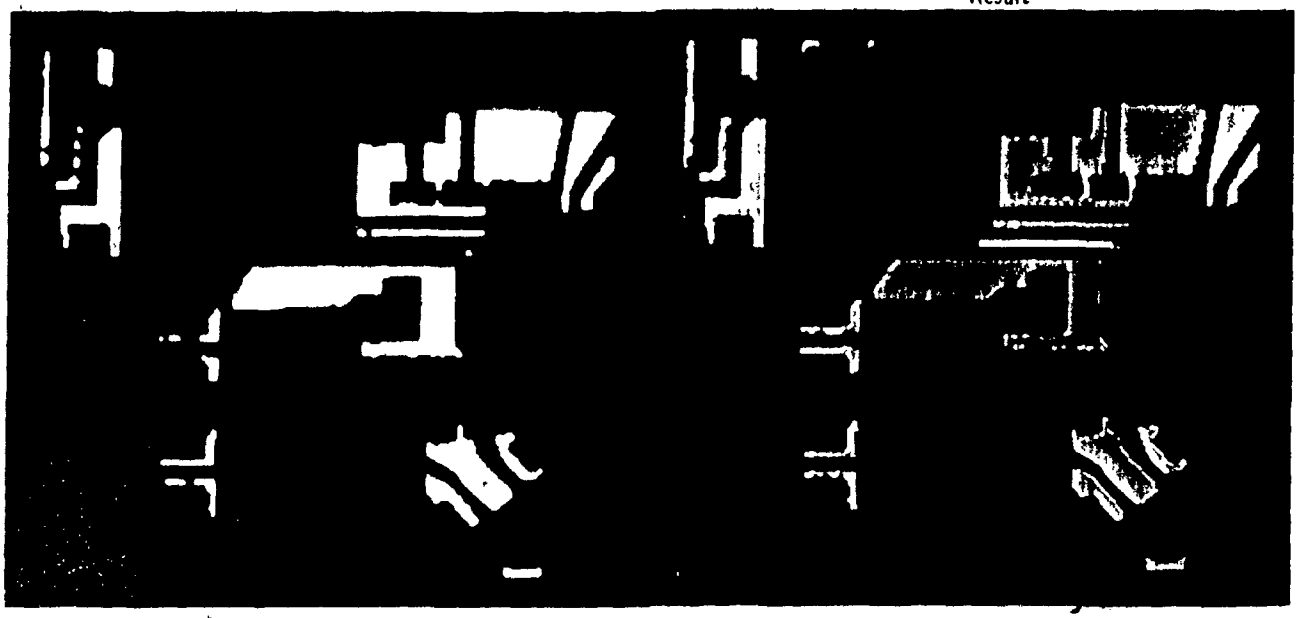
Extracted dielectric layer

Figure 5.4a Example of defect detection on the dielectric layer. Extraction of the dielectric layer from the image of a circuit



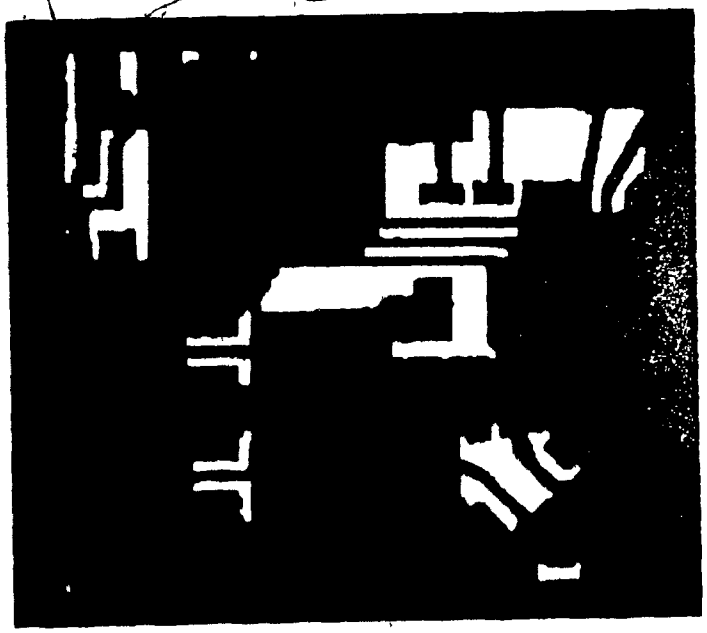
Expansion

Result



Contraction

Result



Combined results The light areas show potential defects

Figure 5.4b Example of defect detection on the dielectric layer Finding the defects with the expansion-contraction methods

```
12. getArrayValues 30 40 39 28 34 35
13. neckOnConductor 52 46 50 51 44

width of fault number 3 at location (54,117) is more than 75% the width
of the conductor it lies on

14. loop 19 46 49
15  getArrayValues 46 56 55 44 50 51
16  endLoop 68
17  loop 19 62 65
18  getArrayValues 62 72 71 60 66 67
19. endLoop 84
20  loop 19 78 81
21  getArrayValues 78 88 87 76 82 83
22. endLoop 100
23  loop 19 94 97
24. getArrayValues 94 104 103 92 98 99
25. endLoop 116
26. loop 19 110 113
27. getArrayValues 110 120 119 108 114 115
28. endLoop 132
29. loop 19 126 129
30. getArrayValues 126 136 135 124 130 131
31. endLoop 148
32. loop 19 142 145
33. getArrayValues 142 152 151 140 146 147
34. endLoop 164
35. loop 19 158 161
36. getArrayValues 158 168 167 156 162 163
37. neckOnConductor 180 174 178 179 172

width of fault number 11 at location (55,116) is more than 75% the width
of the conductor it lies on

38. loop 19 174 177
39. getArrayValues 174 184 183 172 178 179
40. areaOfFaultOnPad 196 190 194 195 188

area of fault number 12 at location (192,139) is more than
25% the area of pad
```

Figure 5.5 *Examples of analysis results* The system issues error messages with the location and description of defects. The names and numbers seen in the listing above are the names of the rules fired by OPS5 and the memory elements accessed by the rules This data is particularly useful for debugging a new or modified system.

5.4 Hardware Requirements and Suggested Improvements

The hardware used in this research is a general purpose research facility. The system

is not intended to be used in an industrial environment At this point, the analysis takes at least 30 minutes or more to completely analyze one 1 inch by 2 inch hybrid circuit In an real production environment this is completely unrealistic The production system itself takes an insignificant portion of the inspection time, but it could still be run in a faster environment, with dedicated hardware and a faster version of the production system than the LISP version With available array processors and other dedicated software, operations like convolutions and correlations can be performed within seconds instead of minutes With such hardware, it is realistic to expect an inspection time of one minute or less per circuit.

The bottleneck occurs at the fault detection modules, which require expensive computations on large picture arrays They could be improved in two ways by using specialized computer hardware, such as array processors, to speed up operations like template matching; and by using hardware to do some of the processing optically and mechanically instead of digitally Such hardware includes specialized optical or digital machinery to perform the expansion-contraction or matching operations, as well as $x - y$ and rotation stages

Of course, the overall project would include circuit manipulation and repair elements such as robots, conveyor belts, cameras, suction pumps, etc. Such a repair system is currently being designed in our Computer Vision and Robotics Laboratory. This system will be interfaced to the inspection system described in this thesis in conjunction with two other projects. These deal with the three-dimensional aspect of the inspection problem and with the solder joint inspection problem.

Apart from hardware improvements, the software of the system could be improved This is by no means the final version of a system applicable to industrial conditions, but its present performance proves its applicability to real manufacturing environments. Some of the improvements we suggest for the system are the following

1. The data provided by the PARTITION module is very important for the performance of the production system. This module is run once for every new circuit, and the information it generates is used for all the circuits of the same type to be inspected. Since this information is so crucial, and since the output is not always perfect because of digitization errors and noise, the system should provide a human technician a means of editing the data. Two methods of editing should be used. First, the corners detected by the program should be modifiable. This could be done with a joystick by displaying the circuit on a screen and changing locations, adding, or deleting corners. Then, once the polygon types are determined, the operator should

be able to modify these types, divide polygons into smaller ones, merge polygons, etc These two interactive steps would help create a more perfect model of the circuit for future use Also, as a separate type of polygon it might be a good idea to add the lead frame attachement pads, since some of the rules related to these pads are different than rules related to other kinds of pads

2. The generation of the screens by a CAD/CAM system would help in almost every stage of the production and inspection During the computerized design, engineers could indicate the types and exact locations of polygons used for the conductor layers, thereby eliminating the need for not only the interactive editing suggested above, but the role of the PARTITION module altogether The design of screens using CAD/CAM systems would also provide more accuracy and flexibility in design and modifications.

3. Statistics related rules have not been added to the production system Such rules would generate histograms of the types and locations of defects occurring on different layers, and would suggest corrective actions whenever certain defects occurred too often, indicating a possible flaw in the production process

4. Better software tools could certainly help the development of the system. The version of OPS5, the production system interpreter, used in this thesis has more programming flexibility than other versions of the same interpreter that we examined, but it also lacks certain abilities related to resource management * Because of its lack of memory space and array limitations, all the operations on digitized pictures had to be done in a completely separate environment, and the data obtained from these operations were passed to the production system in a somehow inconvenient way (see Section 2) At any stage, if additional information related to digitized pictures was needed, other dedicated lower level programs had to be run in order to provide the information. OPS5 has no means of performing the operations within itself. This difficulty caused the additions of new rules to be more difficult than we would expect it to be with an interpreter having the required capacity.

---

* Three other versions of OPS5 have been experimented with during this thesis work. They are the experimental BLISS version mentioned in [Forgy81], and pre-releases of versions 1 0 and 2 0 of VAX-11 OPS5, also written in BLISS

5.5 Conclusion

In this thesis we have designed and implemented an automated visual inspection system for two-dimensional layers of hybrid circuits. The system works well in our research environment It detects, analyzes and reports on defects on any of the printed layers of hybrid circuits. It would need specialized hardware and possibly some software adaptations to be used in an industrial production environment, but it has been proven that it can accomplish the task of inspecting and analyzing defects

REFERENCES

[Almi77] L U Almi and J Shamir. One-dimensional Fourier Transform for the Inspection of Photomasks. *Applications of Holography and Optical Data Processings*. Marom, Friesem and Wiener-Avnear. Eds Pergamon. 1977 pp 541-548

[Arlan79] L Arlan *et al*. High-Resolution Computer Controlled Television System for Hybrid Circuit Inspection. in *Proc. SPIE. Imaging Applications for Automat Industrial Inspection and Assembly*. Vol 182. July 1979. pp 130-139

[Arlan80] L Arlan. Thick-Film Hybrid Inspection With a Computer Controlled High-Resolution Imaging and Storage System. in *Proc. SPIE. Opt. in Metrology and Quality Assurance*. Vol. 220. Feb 1980

[Axelrod72] N N Axelrod. Intensity Spatial Filtering Applied to Defect Detection in Integrated Circuit Photomasks. *Proc. IEEE*. Vol 60. 1972. pp 447-448.

[Bentley80] W A. Bentley. Automated Optical Inspection of Multilayer Printed Circuit Boards. *Proc. SPIE. Optics in Metrology and Quality Assurance*. Vol. 220. February 1980.

[Bjorklund77] C M. Bjorklund and T Pavlidis. On the Automatic Inspection and Description of Printed Wiring Boards. *Proc. Int. Conf. Cybern. Soc.*, Princeton, NJ. 1977. pp 690-693.

[Buchanan76] B G. Buchanan *et al.*, Automatic Rule Formation in Mass Spectrometry by Means of the META-DENDRAL Program. Applications of Artificial Intelligence for Chemical Inference XXIII. *Jour. of the ACS*. Vol.98. 1976. pp. 61-68

[Buchanan84] Rule-Based Expert Systems. B.G. Buchanan and E.H Shortliffe. eds . Addison-Wesley. Reading. Mass.. 1984.

[Cheng78] C. Cheng. Automated Recognition System for Industrial Quality Assurance. *Proc. Society of Photo-Optical Instrumentation Engineers*. Vol. 155. 1978. pp. 78-82.

[Chien75] R. T. Chien. W. Snyder. Visual Understanding of Hybrid Circuits via Procedural Models. *Proc. of the 4th Int. Joint Conf. on Artificial Intelligence*. Vol. 2. 1975. pp. 742-745.

[Chin78] R T Chin, C A Harlow and S J. Dwyer, Automatic Visual Inspection of Printed Circuit Boards, in *Proc. SPIE. Image Understanding Syst. and Industrial Applications.* Vol. 155, August 1978, pp 199-213

[Chin82] R.T Chin and C.A. Harlow, Automated Visual Inspection, a Survey, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol PAMI-4, No 6, November 1982.

[DEC82] Vax-11 Fortran User's Guide, Digital Equipment Corporation, Maynard, Massachusetts April 1982, pp 3-20 – 3-23

[Forgy81] C. Forgy, OPS5 User's Manual, *Tech rep* CMU-CS-81-135, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, Penn , July 1981

[Danielsson79] P.-E. Danielsson and B Kruse, Distance Checking Algorithms, *Computer Graphics and Image Processing*, Vol. 11, 1979, pp. 349-376

[Davis76] R. Davis, Applications of Meta-Level Knowledge to the Construction, Maintenance and Use of Large Knowledge Bases, Memo HPP-76-7, Computer Science Dept , Stanford University, Stanford, CA, 1976.

[Davis77] R. Davis and J J King, An Overview of Production Systems, in *Machine Representations of Knowledge*, Elcock and Mehie, eds , John Wiley, 1977, pp.,300-334

[Davis80] Meta-Rules: Reasoning About Control, Artificial Intelligence, Vol 15, 1980, pp 179-222.

[Ejiri73] M. Ejiri, T. Uno, M. Mese, and S. Ikeda, A Process for Detecting Defects in Complicated Patterns, *Computer Graphics and Image Processing*, Vol. 2, 1973, pp. 326-339.

[Fehrs72] D.L. Fehrs, A Gated Optical Comparator for Automated Inspection of Microcircuit Patterns, presented at the Annu. Meet. Opt. Soc. Amer., San Francisco, CA, October 1972.

[Feigenbaum71] E.A. Feigenbaum, B.G. Buchanan, J. Lederberg, On Generality and Problem Solving: A Case Study Using the DENDRAL Program, Machine Intelligence 6, Edinburgh Univ. Press, 1971.

[Feigenbaum77] E.A. Feigenbaum, The Art of Artificial Intelligence: Themes and Case Studies of Knowledge Engineering, Memo HPP-77-25, Computer Science Dept., Stanford

Univ Stanford. CA. August 1977

[Forgy81] C. Forgy, OPS5 User's Manual, *Tech. Rep* CMU-CS-81-135, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, Penn , July 1981

[Freeman77a] H Freeman, Shape Description Via the Use of Critical Points, PRIP-77, *Proc. IEEE Computer Society Conference on Pattern Recognition and Image Processing*, Rensselaer Polytechnic Institute, Troy, N.Y , June 6-8, 1977, pp. 168-174

[Freeman77b] H Freeman, L S Davis, A Corner-Finding Algorithm for Chain-Coded Curves *IEEE Trans on Computers*, Vol C-26, No 3 March 1977 pp 297-303

[Fu77] K S Fu, S Y Lu, A Clustering Procedure for Syntactic Patterns, *IEEE Trans Systems, Man, and Cybernetics*, Vol SMC-7, No 10 October 1977, pp 734-742.

[Hayes-Roth83] Building Expert Systems, F. Hayes-Roth, D A Waterman, and D B Lenat, eds , Addison-Wesley, Reading, Mass , 1983

[Hong84] W. Hong and A Dill, Everything You Wanted to Know About OPS5 But Were Afraid to Ask (A Practical Guide to Using OPS5 at CVaRL), McGill University Computer Vision and Robotics Laboratory, Technical Report 84-5R, 1984

[Hopkinson80a] Flow Chart Depicting the Thick Film Systems Process for X793, prepared by George E. Hopkinson, October 1980, Northern Telecom, Aylmer, P Q

[Hopkinson80b] Flow Chart Depicting the Thick Film Systems Process for X712, prepared by George E. Hopkinson, October 1980, Northern Telecom, Aylmer, P Q

[Jarvis80] J F Jarvis, A Method for Automating the Visual Inspection of Printed Wiring Boards, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.PAMI-2, January 1980, pp. 77-82

[Krakaner78] L. Krakaner and T. Pavlidis, Visual Printed Wiring Board Fault Detection by a Geometrical Method, Computer Science Lab., Princeton University, Tech. Rep. 248, 1978.

[Lee78] D.T. Lee, A Computerized Automatic Inspection System for Complex Printed Thick Film Patterns, *SPIE Applications of Electronic Imaging Systems*, Vol. 143, 1978, pp. 172-177.

[Levine81] M.D. Levine and S.W. Zucker, An Annotated Bibliography on the Application

of Computer Vision to Electronics Inspection. McGill University Computer Vision and Robotics Laboratory. Technical Report 81-5. 1981.

[Levine84] M.D Levine. Robots and Hybrids. Invited paper. *Proc. International Symposium on Image Processing and Its Applications*. Univ of Tokyo. Tokyo. Japan. January 18-21. 1984.

[Levine85] M D Levine. Vision in Man and Machine. McGraw - Hill. 1985

[Lofthouse81] Visual Inspection of Thick Film Circuits. originated by John Lofthouse. Northern Telecom. November 1981

[Mansouri84] A -R Mansouri. A. Malowany and M D Levine. The Use of Vision Feedback in Printed Circuit Board Assembly. McGill University Computer Vision and Robotics Laboratory. Technical Report 84-6R. 1984.

[McDermott78] J. McDermott and C.L Forgy. Production System Conflict Resolution Strategies. in *Pattern Directed Inference Systems*. D A. Waterman and F Hayes-Roth. eds. Academic Press. New York. 1978. pp. 177-199.

[Moran73] T.P. Moran. The Symbolic Nature of Visual Imagery. Proc. Third IJCAI. Stanford. CA. August 1973. pp. 472-477

[Nakashima79] M. Nakashima. K Fujihara and T Inagaki. Automatic Mask Pattern Inspection for Printed Circuits Based on Pattern Width Measurements. *Proc SPIE. Image Applications for Automatic Inspection and Assembly*. Vol 182. April 1979. pp 38-48

[Nakashima84] M. Nakashima *et al*. Pattern Recognition Techniques for Hybrid Microcircuits. *Proc. of the International Conference on Robotics*. 1984. pp. 19-24

[Newell73] A. Newell. Production Systems: Models of Control Structures. in *Visual Information Processing*. Academic Press. New York. 1973.

[Northern81] Thick Film Hybrid Outgoing Visual Inspection Standard. Northern Telecom. document ID: HQM003. 1981

[Olsen76] O. A. Olsen. Visual Method for Locating Faults in Printed Circuit Boards. U.S Patent 3976383. August 24. 1976.

[Pavlidis76] T. Pavlidis. A Minimum Storage Boundary Tracing Algorithm and Its Application to Automatic Inspection. Princeton University. Technical Report 222. December

[VanderBrug75a] G J VanderBrug, A Rosenfeld Two-Stage Template Matching, Tech Rep 364, Computer Science Center, University of Maryland, March 1975

[VanderBrug75b] G J VanderBrug, Two-Stage Template Matching The Grayscale Case, Tech Rep 398 Computer Science Center, University of Maryland, August 1975

[Waterman74] D A Waterman, Adaptive Production Systems, Complex Information Processing Working Paper No 285, Dept of Psychology, Carnegie-Mellon Univ , December 1974

[Watkins69] L S Watkins Inspection of IC Photomasks With Intensity Spatial Filters *Proc. IEEE*, Vol 57, 1969, pp. 1634-1639

[Weiss84] S M Weiss and C A. Kulikowski, A Practical Guide To Designing Expert Systems, Rowan & Allanheld, Totawa, N.J , 1984.

[Winograd75] T Winograd, Frame Representations and the Procedural/Declarative Controversy, in *Representation and Understanding*, Borrow and Collins, eds , Academic Press, 1975.

[Zimmerman82] N J Zimmerman *et al.*, Automatic Visual Inspection System for Hybrid Circuits, *IEEE Workshop on Industrial Applications of Machine Vision*, Research Triangle Park, N C , May 3-5, 1982, pp 55-61

[Zucker78] S W Zucker, Production Systems with Feedback, in *Pattern Directed Inference Systems*, D A Waterman and F Hayes-Roth, eds , Academic Press, New-York, 1978, pp 539-555

1976

[Pavlidis78] T Pavlidis. A Review of Algorithms for Shape Analysis. *Computer Graphics and Image Processing*. Vol 7. No 2. April 1978. pp 243-258

[Peterson74] C Peterson. Automated Visual Inspection. in *Proc 2nd Int Joint Conf Pattern Recognition*. Copenhagen. Denmark. 1974

[Post43] E. Post. Formal Reductions of the General Combinatorial Problem. *Am Jour of Math* . Vol 65. 1943. pp. 197-268

[Restrick77] R C Restrick. An Automatic Optical Printed Circuit Inspection System. *Proc SPIE. Solid State Imaging Devices*. Vol. 116. 1977. pp 76-81

[Rychener81] M.D. Rychener. Knowledge-Based Expert Systems A Brief Bibliography. Technical Report. CMU-CS-81-127. Computer Science Dept . Carnegie-Mellon Univ . June 1981

[Sandland77] P Sandland. Automatic Inspection of Mask Defects. *Proceedings of the Society of Photo-Optical Instrumentation Engineers*. San Jose. California. Vol 100. April 4-5. 1977. pp. 26-35

[Shortliffe76] E. Shortliffe. Computer-Based Medical Consultations: MYCIN. New York. 1976

[Skinner77] J G Skinner. The Use of an Automatic Mask Inspection System (AMIS) in Photomask Fabrication. *Proceedings of the Society of Photo-Optical Instrumentation Engineers*. San Jose. California. Vol 100. April 4-5. 1977. pp 20-25

[Sterling79] W M. Sterling. Automatic Non-Reference Inspection of Printed Wiring Boards. *Proc. IEEE Conference on Pattern Recognition and Image Processing*. August 1979. pp. 93-100.

[Takaki78] A Takaki. S. Ishihara. M. Naruse and T Yamada. Automated Inspection System for Various Defects in Screen-Printed Patterns. *Proc. 13th Intr. Congr. High Speed Photography and Photonics*. Tokyo. 1978. pp.594-597.

[Thissen77] F.L.A.M. Thissen. An Equipment for Automatic Optical Inspection of Connecting Lead Patterns for Integrated Circuits. *Philips Tech. Rev.*. Vol. 37. no. 2. 1977. pp. 77-88.