Yield Improvement of VLSI Layout Using Local Design Rules

Thesis submitted by

Gerard A. Allan

for the degree of

Doctor of Philosophy

Edinburgh Microfabrication Facility,
Department of Electrical Engineering,
University of Edinburgh,
Scotland.

1992

Abstract

The demand for larger more complex systems on a single IC has shown a steady increase and to date has been met by improvements in fabrication technology. In the future it may not be possible to satisfy this demand in the same way, as it will become increasingly expensive to obtain the required process improvements. It seems likely that the demand for even larger single chip systems will continue and that the commercial success of these devices will become more heavily dependent on their yield. At the same time there is also a continuing trend towards more automate layout generation and these layouts are usually less dense than those produced using traditional hand-crafted designs.

This thesis addresses the problem of maximising the yield of circuit layouts and introduces a yield improvement concept of Local Design Rules. These are integrated circuit layout rules that are used to increase a circuits yield by making more efficient use of the circuit area. The rules define a more optimum feature size and spacing of components in relation to the surrounding layout geometry. This enables the "unused" silicon to be reclaimed and used to enhance the circuit yield without violating the layout design rules.

The type of circuit and nature of circuit layout to which local design rules can be applied to give useful yield improvement are discussed highlighting the problems in a fabrication process that can be improved by this type of layout manipulation. The impact of layout changes on the circuit performance that have been made on the suggestion of local design rules is addressed.

Algorithms for the automatic application of track displacement, track width increase and contact increase local design rules are presented along with a spatial data structure suitable for efficient design rule checking of the suggested layout changes. These algorithms have been implemented and used to apply local design rules to integrated circuit layouts. Finally, several examples are presented with results from Monte Carlo yield simulations.

Declaration

I declare that this thesis has been completed by myself and that all the work included in this thesis is entirely my own except where otherwise indicated.

Gerard A. Allan.

Acknowledgements

I would like to thank my supervisor Dr Anthony Walton, who has provided invaluable help and advice during all the stages of this work. My thanks also go to the EMF staff who have answered all my questions, even the stupid ones, and provided both assistance and on the spot tutorials.

I would also like to thank my fellow PhD students, who have over the years proved an endless source of help and humour, without whom I could have finished this work 6 months earlier.

Finally, and most importantly, I would like to thank my parents, family and friends without whose support this thesis could not have been achieved.

This work was funded by the SERC.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The fabrication of integrated circuits is a complex process that can have many hundreds of process steps. Developments in fabrication technology have resulted in a continual reduction in the feature size of integrated circuit components. Feature sizes as small as 0.6μm are now being introduced into commercial plants[1]. Not only are feature sizes being reduced but the actual area of ICs has shown a tendency to increase, as more functionality is demanded. This means that the manufacturing complexity of integrated circuits is being driven from two directions; smaller feature sizes and larger chip area. This places a heavy burden on process engineers, requiring continual improvements in the fabrication process in order to produce ICs with an acceptable yield. Materials used in the fabrication of these devices must be ultra-pure and the environment in which the devices are processed must be free of contaminants.

The cost of process improvements and tooling up for the next generation of semiconductors is prohibitively expensive. There are only a few companies that are able to commit the required investment and even then it is often only with the help of national governments[2]. Other IC manufacturers must attempt to stretch their existing processes to meet the demand for greater functionality and better performance demanded by their customers. Since the required investment is so great even a small increase in return can amount to a substantial sum. The return on investment from a fabrication plant is dependent to a large extend on

the yield of the product(integrated circuits) that the plant produces. This implies that any improvement in yield is of considerable commercial interest.

1.1 Defect Sensitivity

It has long been recognised[3, 4] that the yield of an integrated circuit is dependent on the "active area" of the circuit. This area is not the physical dimensions of the chip, but a calculated value that is based on the defect sensitivity. Defect sensitivity is often defined in terms of the critical area[5]. The critical area, and therefore the defect density, is determined by the circuit geometry, in particular, the size and proximity of the circuit components that define the circuit.

RAM cell designers appear to be most aware of the importance of layout to the final yield of the circuit. RAM cell design in a commercial environment[6] typically involves tuning both the layout and the fabrication process to minimise the critical area and so optimise the yield.

While RAM cell designers have been intimately concerned with the yield obtained from a given circuit layout, designers of general purpose digital circuits have neglected the possibility of fine tuning a layout to achieve increases in yield. Great care is taken to reduce the physical area that an integrated circuit design will occupy but, so long as the layout does not violate any of the design rules, no further modification would normally be made. Little or no effort is expended to maximise the yield of the layout inside the area already defined by the circuit.

There are many reasons why circuit designers do not attempt minor modifications that would improve circuit yield:

- no tools may be available to measure the yield of the layout
- IC manufacturers do not usually inform designers of the defect levels in the manufacturing process. This makes it difficult to determine the yield or relative yield of different layout options

- designer have traditionally not been responsible for the yield of their circuit, since that is seen as the domain of the process engineer.

Whatever the reasons for the lack of interest in the past, in the future commercial pressures may force circuit designers to take some of the responsibility for the yield of the circuits they design. If that is the case then tools to aid the designer will be required.

1.2 Objectives

This thesis addresses the issue of layout modification to improve the defect sensitivity and hence the yield of circuit layout. Layout modifications can only improve the yield of integrated circuits when they affect, for the better, process steps that are ordinarily the source of yield losses. Ideally such a process step should be modified to reduce to zero the number of defects that it introduces. Where this cannot be achieved, because of either the high cost of modifying the process or the inability of current technology to solve the problem, it makes sense to reduce the sensitivity of the layout to those defects that will be introduced.

The main objective of this work was to develop a method to increase the yield of integrated circuits by modifying the circuit layout to decrease the sensitivity of the circuit to defects. Defects are inevitably present in the manufacturing process, therefore, reducing the defect sensitivity of a circuit will in general increase the circuit's yield.

While yield improvement was the overall aim there were a number of objectives and goals which are required to fulfill this.

- **Examine Layout and Layout Rules**

  Layout generated by a number of sources particularly layout that had been automatically generated, was examined in order to determine the effect of layout rules on circuit area. If changes in layout rules do not greatly affect

the circuit area and give increased defect sensitivity then the design rules should be modified.

- **Suitable Layout Modifications**

  If, in order to decrease the defect sensitivity, the layout is to be modified, these modifications will have to be simple in order that they can be easily accomplished, either by hand or automatically. Modifications that require the re-working of large areas of a circuit, in order to accommodate a layout change, are unacceptable, even where yield improvements can be achieved.

  It is necessary to determine a set of simple layout changes that can be made, which do not impact greatly on neighbouring layout and yet reduce the defect sensitivity.

- **Measure Change in Defect Sensitivity**

  The effect any layout modifications has on the defect sensitivity of the circuit layout must be modelled or measured in some way, in order to determine if the objective of decreasing the defect sensitivity has been achieved.

- **Automate the Process of Layout Modification**

  The process of layout modification is time consuming and therefore not suitable to be done by hand. If the process is to be used in a commercial environment, it must be automated to allow layout modifications to be made with as little intervention from the circuit designer as possible.

- **Description of Layout Modifications**

  In order to automate the layout modifications they must be described in a form that can be easily interpreted by a computer program. That is a simple set of rules or tests must be developed that indicate whether an individual piece of circuit layout can be usefully modified within the constraints imposed by its immediate environment.

- **Efficient Algorithms**

  The automation process will be required to operate on large circuit layouts.

In order for these modifications to be made in a reasonable time, algorithms that are time efficient must be researched and tailored to suit the particular requirements of the automation software.

- **Selective Control of Automation Process**

  It was considered desirable that a circuit designer should have the option of applying the layout changes selectively. This would be done by indicating those pieces of layout that are to be changed. This implies some sort of interactive control of the layout modification process must be made available.

- **Parallel Execution**

  Equally desirable is a fast response time, so that very large designs can be processed quickly. Even with efficient algorithms very large designs are likely to take too long for commercial use. To speed up this response time the automation process should be capable of processing a layout in parallel. This requirement may affect the design of the layout modification algorithms.

- **Direct Future Research**

  The objectives of this work were limited to simple layout modifications. Another aim was to gain enough experience with these simple layout modifications to develop an understanding of what could be achieved by more detailed modifications. This experience will be used to determine how practical it would be to design a system to automatically apply much more sophisticated layout manipulations to increase circuit yield.

1.3 Overview

This work describes a method for decreasing the defect sensitivity of IC layout. This method uses what have been termed Local Design Rules (LDR). These are rules that can be used by a designer, but more usefully by a computer program, to test whether a particular piece of circuit layout, from an existing completed IC design, can be modified to increase the circuit yield. The LDRs do not increase the circuit area, but instead use any redundant space that is available between the existing circuit layout geometry. That is LDRs suggest layout modifications that make more effective use of the original circuit area.

Whether LDRs can be usefully applied to a given circuit design is highly dependent on both the type of layout and the fabrication process to be used to manufacture the IC. Since LDRs make use of redundant space, it follows that layout generated by automatic layout tools is more suited to this process than an equivalent hand crafted layout, since hand crafted designs are nearly always more dense than automated designs.

The implementation of a computer program called LocDes that is used to apply LDRs to circuit layout is described. The algorithms that are used by LocDes to perform the layout modifications and the associated design rule checking are presented.

1.3.1 Using LocDes

The LocDes program is a post-processor of integrated circuit layout. Only when the design of an integrated circuit or cell has been completed would LocDes normally be used. The program searches the layout for instances of circuit geometry that it considers non-optimal. That is a piece of layout for which there is a LDR that describes conditions that match those of the particular piece of layout. The LDR will then prescribe a layout modification that can be used to reduce the

defect sensitivity of the layout. The modification is only applied if none of the layout design rules are violated.

Because modifications would usually be made to completed integrated circuit designs, those designs whose simulated performance is close to the performance specifications may not be suitable for LDR modifications. This is due to the fact that some layout changes may adversely affect the performance of a circuit (this is discussed in greater detail in chapter 5). Where performance is critical and the expected performance of a device is so close to its specifications limits that indiscriminate layout modifications cannot be contemplated, the circuit designer can use an interactive version of the program. This allows the designer to select which layout is to have LDRs applied, ensuring that critical areas of the circuit remain unaffected. An alternative is to apply LDRs earlier in the layout process. By applying LDRs to individual circuit cells before simulation of their electrical characteristics, the designer can build the circuit from these blocks. The designer will then be able to determine the performance of the circuit in the usual way and make any modification to the already "enhanced" cell in the normal way.

1.3.2 Outline

Chapter 2: Background

This chapter gives some background material necessary to the understanding of the work in the later chapters. The definition of yield as used in this thesis is given along with some discussion on the causes of yield loss. Yield prediction is explored both in terms of chip area and the defect sensitivity of the circuit layout. A number of layout based yield prediction methods are discussed.

Layout based yield prediction is to a large extend dependent on the defect density and size distribution. Results are presented for an experimental determination of defect density size distribution based on metal serpentine and comb test structures.

Chapter 3: Yield Simulation

As there was no suitable yield simulator available, when this work was undertaken, a method of analysing the results obtain from the application of LDRs to circuit layout was required. This chapter describes the implementation of a yield simulator that uses the Monte Carlo method to estimate yield. This simulator was designed to predict the yield of tracks (metal1, metal2, polysilicon shorts and breaks), to enable the effectiveness of LDRs to be established, by comparison of the enhanced layout with the original version. Details of defect placement and fault analysis used by the simulator are presented and discussed.

Chapter 4: Design Rules

This chapter introduces integrated circuit layout design rules. These rules determine the minimum size and spacing of circuit components. The influence of these rules on circuit area, for different circuit layouts, is investigated using a circuit compactor. The results of these investigations and their implications for yield improvement strategies are discussed. The importance of defect sensitivity and the possibility of decreasing it for non-optimal layouts is highlighted,

Chapter 5: Local Design Rules

This chapter introduces Local Design Rules (LDR). These rules are used to describe the conditions under which a specific layout modification can be made. LDRs to increase track width, displace tracks and increase contact/via size are presented. The conditions under which these LDRs can be used and the effect the layout modifications have on the defect sensitivity, performance and reliability of circuits is explored. Layout modifications to increase yield are not recommended for all circuit types and fabrication conditions, and an explanation is given of those circuit types and fabrication processes that may prove suitable.

Chapter 6: The LocDes Program

The LocDes program is presented. This program has been designed to automatically apply LDRs to integrated circuit layout. The program can operate with either the X windows user interface, allowing selected layout to be modified, or in a batch mode, where all of the layout is adjusted where possible. The algorithm-

s used by the program to implement, track width changes, track displacement and contact/via adjustments are given. The parallel operation of the program is discussed. This allows very large circuit layouts to be processed in a reasonable time.

Chapter 7: Design Rule Checking

The execution speed of the LocDes program is largely dependent on the design rule checking that is required to ensure that all the layout changes performed by the program do not violate the design rules. This chapter presents and explains the design rule checking algorithms used for each of the LDRs that can be applied by the LocDes program. The speed of these algorithms is in turn highly dependent on the speed of the geometry searches. The adaptive multiple storage binary tree data structure used to achieve fast region searches is explained.

Chapter 8: Results

This chapter presents results from the application of LDRs to circuit layouts made by the LocDes program. These results, in the form of the differences in defect sensitivity, were obtained using the Monte Carlo yield simulator presented in chapter 2. Results for LDRs applied to a routing network and a random logic layout are presented and discussed. The difficulty of determining the effect of LDRs on the defect sensitivity of the circuit layout is highlighted, since LDRs can simultaneously affect a number of different fault mechanisms.

Chapter 9: Discussion and Conclusions

This chapter summarises the work presented in this thesis. The future role of LDRs and LocDes within the IC design process are discussed. Possible improvements to LocDes to make it more usable in a design environment are discussed, as are possible improvements to the type and range of LDRs. A more efficient method of yield simulation using analytical techniques based on polygon manipulation is highlighted. The lessons learned during the work involved in this thesis have been used to develop a plan for an automatic layout system capable of producing IC layout with a yield comparable to hand-crafted designs.

Chapter 2

Background

This chapter provides some background to yield and yield prediction. The importance of defect size distribution and density is explored. Some experimental results for defect size distribution are presented that are in agreement with other reported results.

2.1 Introduction

The manufacture of integrated circuits is a complex and exacting task. At present it is not possible to control the material quality or process parameters to the extent that a 100% defect free product can be produced. It is not likely that such a state of perfection will be reached in the foreseeable future. Once it is accepted that defects will occur, the task of process engineers and designers of integrated circuits is to achieve the highest value of yield and predict accurately what that yield will be.

There are two types of yield loss – structural and performance. Shorts and opens are examples of structural faults. Performance faults add resistance and capacitance; the circuit may still work but not at the desired specification. The fraction of chips that pass a number of electrical tests of dc connectivity and performance is known as the yield. There are a number of mechanisms that

cause yield loss and points in the production line at which yield can be measured. We will consider the yield to be the number of circuits working when tested before the wafer is scribed and individual chips are wire bonded and packaged.

To date the most important way of improving both the performance of MOS chips and their density has been the use of scaling. Scaling has physical and process limits. The physical limits are absolute. The process limitations are the result of the lithographic process. The dominant limiting factor has been shown to be drain area breakdown – merging of the source and drain [7]. Sub-micron geometries have already been achieved commercially by many fabricators and the trend is now to 0.5 micron levels and lower. These scaled down devices are more process sensitive, relying on thinner and better oxides. Major improvements in the quality of the materials used in the manufacture and the cleanliness of the environment in which the circuits are manufactured must be achieved if such circuits are to be manufactured at a profitable level of yield.

While the major increases in yield in ICs are the result of improved fabrication techniques, the yield at any point in time is also determined by the layout rules employed by the designer. The dimensions of circuit features determine the yield of the circuit and the overall circuit size. To maximise profits it is necessary to find the optimum layout rules that give the maximum number of working chips from a given wafer area[8, 9]. As the fabrication process becomes more complex, with an increasing number of layers and process steps, the interaction of these layers makes it increasingly important to optimise the layout rules to obtain high yields and compact circuits.

The circuit designer can affect yield by making the right choice in the size and position of circuit features. It is also possible to design circuits that are tolerant to faults and so make it possible for a circuit to perform its function even though it is less than perfect. The size of the circuit is also very important since larger circuits tend to have a lower yield than small circuits. A choice must be made between fitting a system on a single chip or a dividing it into a number of different chips. An understanding of this relationship with area is important for a designer

working on a large system so that an informed decision of the optimum chip size can be made[10].

An important factor to consider in development of a new process is the learning curve, whereby the yield improves by approximately 20% [11] every time the number of batches that have been processed doubles. In the early stages of production this can lead to very significant improvements in yield.

2.2 Types of Yield Loss

Yield loss can arise from many different causes. IC fabrication is a complex process which requires very strict control of many process parameters such as temperature, humidity, concentration of doping agents, and purity of all materials used. Because of the complexity of the process it is not altogether surprising that some circuits do not function correctly. The total yield loss of a functionally correct design is often divided up into gross yield losses and faults from random defects.

2.2.1 Gross Yield.

Gross yield losses are usually caused by manufacturing errors affecting the whole or a large part of the wafer [11]. Excessive variation in processes parameters such as process time, temperature, doping concentrations and lithographic faults, are usually the cause of this form of yield loss. Gross yield losses can often be found by visual inspection of wafers or measurements made during the process from test structures formed in the kerf[12]. Wafers with gross yield loss can removed from the production line before the processing is completed. The test structures are used to ensure that circuit elements, transistors, vias, contacts etc. are within the process window (i.e., within the range of parameters that will ensure working devices). The distribution of device parameters within the process window is of special interest to manufacturers who sort their product by performance.

While LDRs could in some circumstances be used to decrease the sensitivity to some types of global defects (for example, increasing contact overlap to reduce sensitivity to mis-alignment) the effectiveness of such LDRs will be limited. In general a global defect results in a large area of unusable wafer. Unless layout modifications can be applied to all the susceptible components in this area (e.g. all contact overlaps) and give a significant degree of protection, it is likely that there will still be some defects that will result in circuit faults. Consequently it is not recommended the LDRs be used to solve gross yield loss problems.

2.2.2 Random defects.

Random defects are spot or small area defects. Spot defects are found randomly their concentration normally varies over the wafer surface as a function of radius. Typically the concentration of defects is greater towards the edge of the wafer [13, 14]. Stapper[11] suggests that more than 80% of yield loss is due to random rather than gross defects, while a figure of 60% is suggested by Mangir[15, 16].

These random defects are the result of local process fluctuation that only affect a very small area of a device. If a circuit is made more tolerant to these types of defect, an increased production yield will result.

In mature MOS fabrication processes a few small area defect types dominate the total yield loss. These are extra and missing material defects, oxide pinholes and junction leakage[17-19]. These are discussed in greater detail below.

Some other defects that cause localised circuit faults are hillocks, which cause shorts between conducting layers and step coverage breaks where thinning at crossovers causes tracks to break. However these defects are due to global rather than local process disturbances[20]. They are best resolved by adjusting the process itself, rather than using LDRs to reduce the sensitivity to a global error.

Extra and Missing Material Defects

These defects are caused by dust particles on masks/reticles, the wafer surface and chemicals used in the fabrication process. The dust particles lead to unexposed resist, which when developed results in an unwanted "pinhole". This pinhole will result in a missing material defect if the next stage is an etch or an extra material defect will be formed, if the following stage is a deposition.

Because of the close association of extra and missing material defects to the lithographic process these defects are often called photo or lithographic defects[18].

Missing material defects are responsible for metal/polysilicon track breaks while extra material defects cause track shorts as well as missing contacts/vias (the contact cut is not etched away).

The sensitivity of a circuit to metal and polysilicon extra/missing defects can be reduced by LDRs. In particular width increase LDRs reduce the sensitivity to missing material defects and track displacement LDRs reduce the sensitivity to extra material defects. Yield loss from missing contacts can also be attacked by using LDRs to increase the contact size, so that a larger defect is required to totally close the contact.

Oxide Pinholes

Oxide pinholes are primarily caused by a deficiency of oxygen at the Si-SiO_2 interface, tensile stress, surface imperfections and contamination[21-23]. The yield of oxide layers is a function of the area of oxide.

In general it is not possible to reduce the area of critical oxide, since gate area and the overlap area of conductor are fixed(figure 2–1). This makes it difficult to design LDRs to decrease the sensitivity to this type of defect. However, where separate layers form tracks running on top and parallel(figure 2–1) to each other it may be possible to displace one or both of the tracks to reduce the area

of overlap and therefore the probability of a fault from an oxide pinhole. For example, this situation can occur between polysilicon and metal 1 as shown in figure 2–2. An LDR to perform this layout modification is feasible, but has not been implemented. The occurence of such parallel wires is low in most layout styles (metal and poly are often run perpendicular to each other) so that it is unlikely that this adjustment would be required very often.

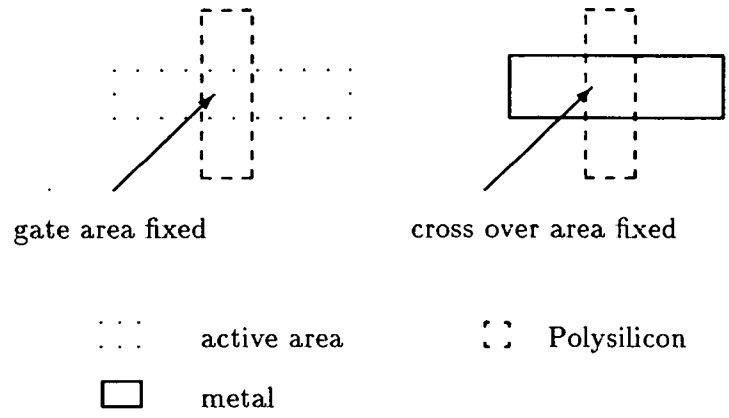


Figure 2–1: Fixed Area of Oxide

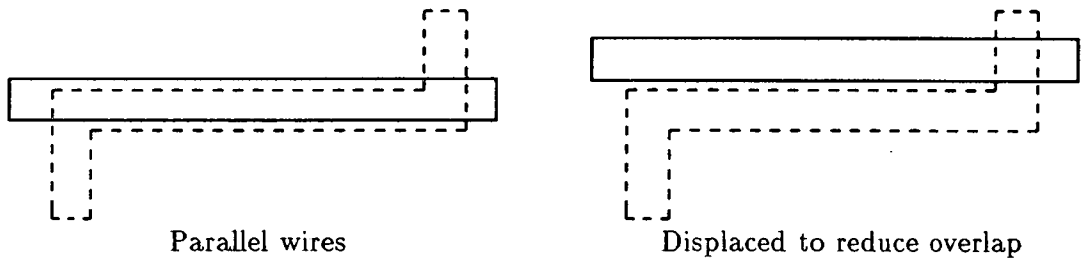


Figure 2–2: Displacement to Reduce Critical Oxide Area

Junction Leakage

Junction leakage occurs where crystal defects or contamination are present at a junction[24]. The high electric fields at these points cause large current leaks across the junction, resulting in poor performance or a failed device. Junction leakage defects can be a substantial source of yield loss in RAMs[17-19, 24]. This is not such a problem in general purpose digital circuits as the density of active

devices is usually much lower. The yield loss is a function of diffusion area and periphery.

There is little that could be done to affect the sensitivity of circuits to these defects since diffusion areas are usually at the minimum required to form the active devices and connect to other conducting layers. Consequently, the number of layout changes that could be made is small.

2.2.3 Fault Clustering.

The distribution of spot defects is random though in a complex manner. It has been observed that the defects on a wafer have a tendency to form clusters[25]. It is thought that this occurs as a result of clouds of particles in the materials used in the fabrication process. This implies that if there is a defect present the probability that another defect is present on an adjacent site increases. If there are a number of related defects then the probability of there being a large cluster of defects increases. This has important consequences for the yield and reliability of redundant circuits (section 2.9).

There is some dispute as to the definition of what constitutes a defect cluster[26]. This is a result of practical difficulties in measuring clusters. For instance, what appears to be a cluster of defects may in fact be unrelated random defects that have the same position on the wafer map but have been caused by different process steps. It is often difficult to determine the cluster size. Even if it can be shown that the defects were all caused by the same process step it is not clear whether a particular defect, near what is considered the cluster edge, is in fact a random defect close to a cluster or indeed part of another adjacent cluster. Attempts to define mathematical models that overcome these difficulties have been hindered by the lack of relevant data[27]. There is an understandable reluctance by IC manufacturers to release the raw wafer map data that can be used to verify existing models and aid in the proposition of new models

The Influence of Clusters on LDRs Effectiveness

How much defect clusters will affect the usefulness of layout modifications in reducing defect sensitivity is dependent on the ratio of the faults that are caused by single defects and those caused by defect clusters. If a large percentage of all circuit faults are due to single defects then layout modifications will affect the final yield by reducing the number of single defect faults. However, if defect clusters are the major source of circuit faults, then the density of clusters will determine whether layout modifications can improve the yield.

If layout modifications give $x\%$ reduction in single defect sensitivity then the corresponding reduction for defect clusters is $100 \times (\frac{x}{100})^n\%$, where n is the number of defects per circuit area or cluster density. For example, if a 10% reduction in defect sensitivity is obtained for a single defect the equivalent reduction in defect sensitivity for a defect cluster of density 2 defects/circuit area is only 1%.

There is no reliable data concerning either the defect density of clusters or the ratio of faults caused by single defects and defect clusters. Consequently, it is not possible to draw any conclusions on the impact of the defect clustering phenomena on the effectiveness of layout modification in reducing defect sensitivity. Also, since each process has unique defect statistics it is likely that some processes will be more suitable than others.

2.3 Yield and Chip Area.

2.3.1 Poisson Statistics.

There is a large volume of work on the statistics of yield which relates yield to the defect density and chip size. One of the earlier papers is by Murphy[3]. Murphy noted that an exact calculation of yield could only be made by examining each circuit on its individual merits. Considering this to be too difficult a task an

attempt to relate the yield of a chip to its area is given. Assuming the defects to be randomly distributed, Poisson statistics are used to generate an equation for yield

$$Y = \exp(-DA) \tag{2.1}$$

Y = yield

D = Defect density (defects/area)

A = Area

This equation states that the logarithm of yield decreases linearly with the area of chip. This equation is not in agreement with results found in practice, giving an overly pessimistic estimate for large chips. Equation 2.1 assumes random defects while in practice it is found that the defects are distributed non-randomly. This disagreement can be overcome by regarding the problem as a sum of sub areas each with a different defect density.

$$Y = \int_0^\infty \exp(-DA)\, f(D)\, dD \tag{2.2}$$

$f(D)$ = probability of defects

The defect probability function ($f(D)$) must be found experimentally. Equation 2.2 assumes that there is only one defect type or one major defect type. This unrealistic assumption can be overcome by calculating the yield for every defect type so that the yield of a process with n defect types can be expressed by equation 2.3.

$$Y_{total} = Y_n\, Y_{(n-1)}\, Y_{(n-2)} \ldots . Y_2\, Y_1 \tag{2.3}$$

2.3.2 Window Method.

Experimental results have shown that the Poisson method of yield calculation consistently gives lower predicted yield than those observed in large chips. This is due to the assumption that defects are randomly distributed. Evidence that

defects are not randomly distributed can be seen graphically by use of the "window method" originally described by Seeds[28]. In this method a wafer map of defective and working chips is divided into areas of integral numbers of chips. An estimate of the number of working larger chips obtainable can be found by counting the number of larger areas that have only working chips in them. However great care has to be taken when using this method as the sensitive area of a chip may differ from its actual area and the ability of a circuit to operate with a performance fault is liable to vary from one chip design to another. This method can only be considered accurate where the larger chip is equivalent to the sum of the smaller chips. This does not occur often. The "window method" does however suggests that the Poisson statistics are not an accurate prediction of yield for larger chips.

2.3.3 Boltzmann Statistics

It can be shown [29] that the Poisson model is false since it is based on Boltzmann statistics in which all spot defects are considered distinguishable. Boltzmann statistics give,

$$Y = (1 - \frac{1}{N})^{NAD} \tag{2.4}$$

$$\lim_{N \to \infty} Y = \exp(-DA) \tag{2.5}$$

N = Number of cells

A = Active area of cell

D = Average Defect Density of Randomly distributed defects.

However defects can be considered to be caused by a number of distinct mechanisms producing a number of indistinguishable defects.

2.3.4 Bose-Einstein Statistics

Using Bose-Einstein statistics in which all defects are considered indistinguishable[29]. The equation for yield becomes

$$Y = \frac{1 - 1/N}{1 + AD - 1/N} \tag{2.6}$$

$$\lim_{N \to \infty} Y = \frac{1}{1 + AD} \tag{2.7}$$

For a more general case with n different defect types the yield then becomes

$$Y = \frac{1}{1 + AD_1} \times \frac{1}{1 + AD_2} \times \cdots \times \frac{1}{1 + AD_n} \tag{2.8}$$

Each of the terms in the yield equation represents a defect mechanism with defect density D(n). This statistical representation tends to over-estimate the yield obtained from large chips.

2.3.5 Experimental Methods

The defect density has been found experimentally to be distributed according to the gamma function[19]. If this function is used as the model for defect densities the resulting equation for yield is a negative binomial distribution. The resulting yield equation is empirical, not based on any physical model or mathematical theory. This gamma function produces a yield function that is not linear with the logarithm of yield and actually over-estimates the yield of larger chips but gives a good fit within the range of chip sizes in the data base.

$$Y = \frac{1}{(AD/n + 1)^n} \tag{2.9}$$

Where n = (mean/standard deviation)^2 of defect density
measurements taken from different wafers within a batch
and a number of different sites on the wafers.

The yield of a process cannot be well modelled by a simple mathematical or physical model. There are a number of parameters involved and they do not

obey any simple relationship. Often the best approach is to empirically form an equation for yield. This has the benefit of at least being accurate within the data base provided which for most purposes is all that is required. One method is to use the relationship derived by Murphy with a gamma function of defect density using an experimentally determined mean and standard deviation of faults in test devices[30].

Data from commercial fabrication lines has shown that there are difficulties in fitting yield data to yield models. Stapper suggested that this was because the defect sensitivity of circuits is not directly proportional to area. Stapper[19] demonstrated that yield data for ROS [1] circuits can be more easily interpreted by comparing the yield of the chip versus the number of data bits (sub-circuits) rather than the area. This suggests that yield is not so much a function of circuit area as defect sensitivity and where defect sensitivity is not proportional to area the standard yield models will not hold. Some methods of yield prediction using defect sensitivity are given in section 2.8.

2.3.6 Chip Area and Wafer Size.

Yield can also be affected by the size of the chip relative to wafer size. The optimum chip size for maximum silicon usage is dependent on the relative size of chip to wafer. Gupta[10] gives the results of a computer program to give the optimum placement of square chips on a wafer. An empirical formula is generated from the computer program output.

$$N = \frac{W}{Z^2 - \frac{3.54}{Z/R} + 0.94} \tag{2.10}$$

N = number of chips

[1] ROS read only store, memory chips that are fabricated with a fixed set of information

Z = chip size (length of side)

R = effective radius

W= slice area (usable area of wafer)

An effective wafer radius is used since the edges of the chip are not suitable for chip production. This is due to defects introduced by handling and optical distortion at the edges of the wafer[13, 14]. The size of this effective radius will depend on the process used but can commonly be expected to be of the order of 10mm less than the actual wafer radius. The formula can be used to find the amount of wafer actually used as chip area and therefore the wastage. The results show that for chip sizes greater than 0.3 of the effective radius the empirical formula becomes inaccurate. At this size careful choice of the chip size can lead to significantly less wastage of wafer area.

2.4 Defect Density and Size distribution.

The shortage of published data on defect densities in the literature can be attributed to the variation between processes and the commercial sensitivity of such information. However the state of the art is approximately 1 defect/cm^2[31]. There is a shortage of data on the clustering of defects and more clustering information would be useful to the designer of fault tolerant circuits.

It has been observed that smaller defects are more common than large defects, this is true of both air-borne and chemical defects[9]. It has further been observed that below some diameter defect densities fall to zero since they are no longer resolved.

Stapper observed on memory devices that defect size fell off as $\frac{1}{x^n}$, where x is the defect diameter. Small defect sizes could not be observed and were assumed to rise to a peak then fall of linearly as shown graphically in figure 2–3. The

normalised distribution function is,

$$h(x) = \frac{2(n-1)x}{(n+1){x_0}^2} \quad 0 \leq x \leq x_0 \tag{2.11}$$

$$h(x) = \frac{2(n-1){x_0}^{n-1}}{(n+1)x^n} \quad x_0 \leq x \leq \infty \tag{2.12}$$

Where the value of n is chosen to fit the experimental data

Stapper fitted equations 2.11 and 2.12 to process data with a best fit of 3.02. This value agrees well with other reported values for n. Stapper[4] reports another fabrication process gave n=3.1 for extra material defects and n=2.9 for missing material defects. For n=3 the equations 2.11 and 2.12 become,

$$h(x) = \frac{x}{{x_0}^2} \quad 0 \leq x \leq x_0 \tag{2.13}$$

$$h(x) = \frac{{x_0}^2}{x^3} \quad x_0 \leq x \leq \infty \tag{2.14}$$

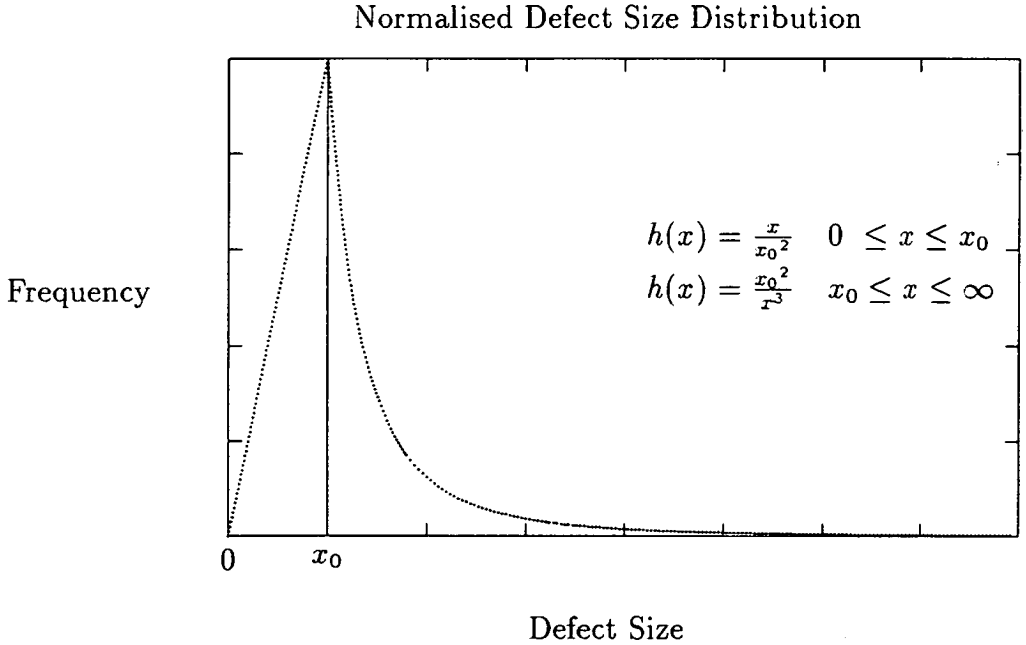


Figure 2–3: Normalised Defect Distribution

The most important part of the curve is the section for defects greater than x_0. The x_0 point is the minimum size defect that can be resolved by the lithographic

process. Since circuit geometry must always have greater width and separation than the minimum resolvable size it follows that defects smaller than this size will not cause circuit faults. The defect size distribution has shown itself to be remarkably constant with ever decreasing values of x_0. It is believed[32] that the distribution will remain proportional to the cube of the defect size for the foreseeable future.

Alternative distributions have been proposed notably by Ferris-Prabhu[33] and Maly[34]. Ferris-Prabhu presented a more general distribution than that given above, which for suitable terms becomes equivalent to equations 2.13 and 2.14.

Maly suggested that the defect size was determined by mask defect diameter combined with the line width distortion that occurs when the mask pattern is applied to the IC. He also suggested that there is more than one source of defects. Measurements showed that there were two sources of defects, dust particles and the mechanical contact of the mask and wafers[2]. The probability density function for a single defect source was modelled by a Rayleigh distribution of,

$$f_r(x,\alpha) = \frac{x}{\alpha^2}\exp(-\frac{x}{2\alpha^2}) \quad x > 0 \tag{2.15}$$

$$f_r(x,\alpha) = 0 \quad x = 0 \tag{2.16}$$

where α is the distribution parameter.

The combination of air-borne and mechanical contact defects were modelled as,

$$\begin{aligned} f(R) &= [\beta_1 f_r(R-m_1,\alpha_1) + (1-\beta_1) f_r(R-m_2,\alpha_2)]\beta_2 + \\ &\quad (1-\beta_2) f_r(R-m_3,\alpha_3) \end{aligned} \tag{2.17}$$

β_1 and β_2 are weights for dust and mechanical defects

m_1, m_2, α_1 and α_2 characterise dust defects

m_3 and α_3 characterise defects caused by mechanical contact

---

[2]This source of defects has been eliminated in modern processes that no longer use contact printing.

In a modern process with no contact printing $\beta_2 = 1$ so that equation 2.17 becomes

$$f(R) = \beta_1 f_r(R - m_1, \alpha_1) + (1 - \beta_1) f_r(R - m_2, \alpha_2) \tag{2.18}$$

2.5 Experimental Determination of Defect Density and Size Distribution

A series of metal 1 comb and serpentine structures similar to those shown in figure 2–4 were fabricated using the EMF 6μm CMOS process.

- A serpentine structure can be used to detect missing material defects. Defects that break the track connecting the probe pads will register as a high resistance (open circuit) between the pads.

- A comb structure can be used to detect extra material defects. Where such a defect connects the interdigitated tracks the resistance between the two probe pads will be reduced in comparison to the high open circuit resistance when no defect has been detected.

A series of these structures were fabricated with dimensions of 1.5, 2.0, 2.5 ... 9.0, 9.5μm. These dimensions refer to track width in the serpentine structures and track separation in the comb structure. To ensure the mode of failure would be shorts in the comb and breaks in the serpentine structure all track widths in the combs and all track separations in the serpentines were 10μm.

The experimental run consisted of 11 wafers with 37 measurable test sites giving a sample size of 407 for each of the structures. The combined results from all 11 wafers are given as wafer maps in figure 2–5 showing all the serpentine results and figure 2–6 the comb results. It can be seen from both of these figures that the number of defects tends to be greater towards the edge of the wafer and that the incidence of shorts (comb structure) is greater than breaks (serpentine

structure). This can be seen more clearly by taking the log of the defects found as shown in figures 2–7 and 2–8.

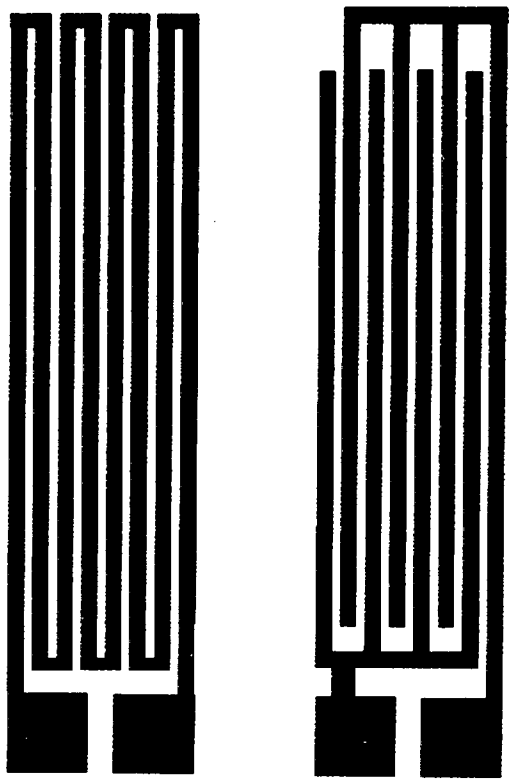


Figure 2–4: Serpentine and Comb Test Structure

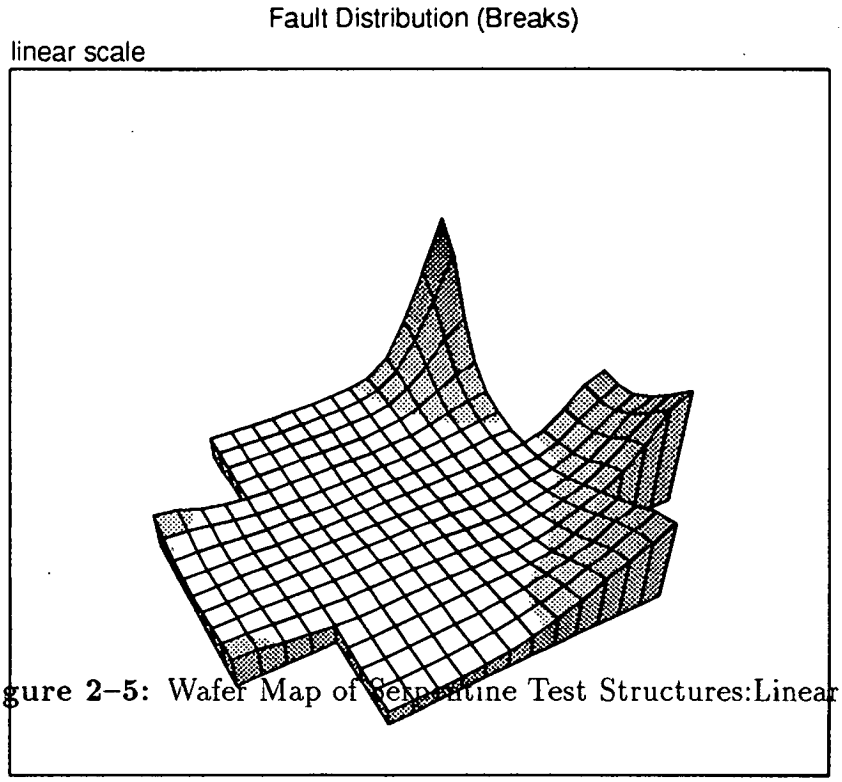


Figure 2–5: Wafer Map of Serpentine Test Structures:Linear Scale

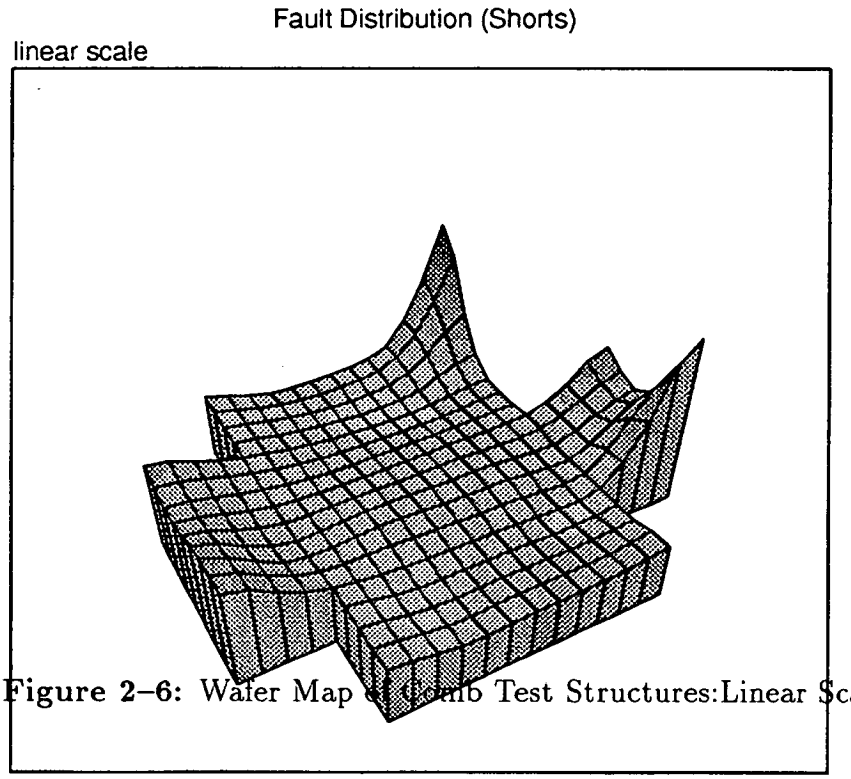


Figure 2–6: Wafer Map of Comb Test Structures:Linear Scale

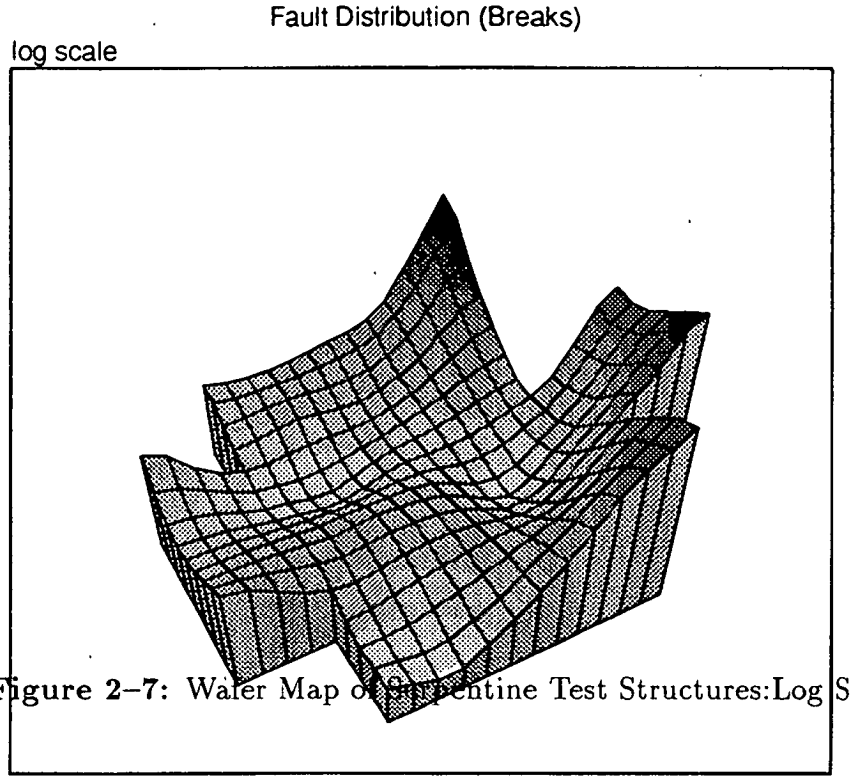


Figure 2–7: Wafer Map of Serpentine Test Structures:Log Scale

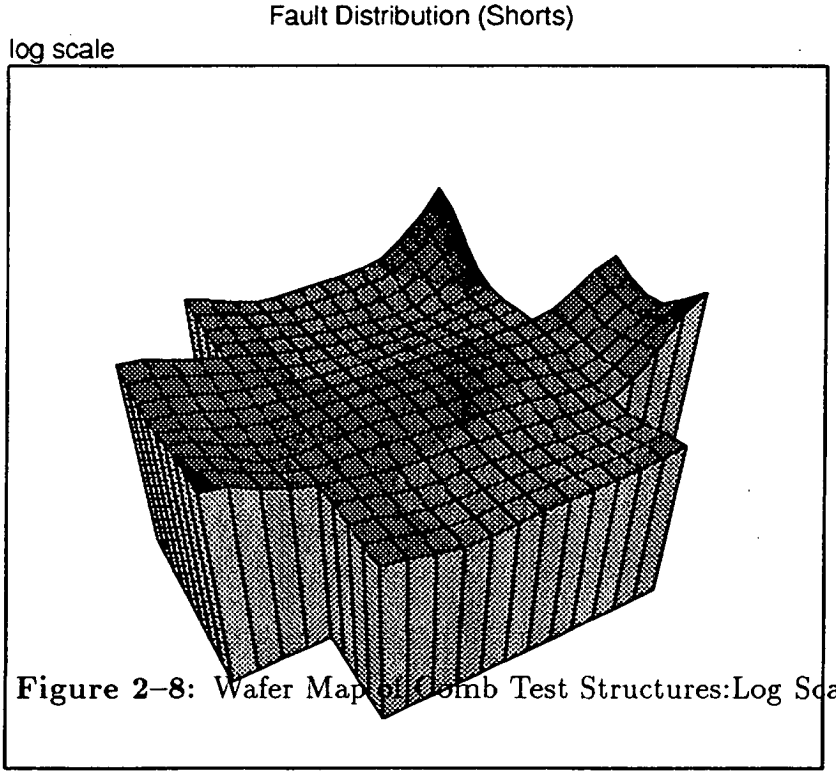


Figure 2–8: Wafer Map of Comb Test Structures:Log Scale

2.5.1 Defect Density

The defect density of the process can be calculated from the number of defects detected by the test structures and the "active" test area. In the 1.5μm serpentine test structure the active area is equal to 1.5 times the total conductor length between the two probe pads. Which in this case was 4mm (the test structure length) times 10 (the number of wire lengths) giving an active area of 60,000μm^2. The active area of a comb structure is similarly calculated from the length and width of the conductor separation.

The defect density is found from,

$$Defect\ Density = \frac{D}{A \times N_{wafer} N_{sites}} \tag{2.19}$$

D = Total Number of defects

A = Active area

N_{wafer}= Number of wafers

N_{sites} = Number of test sites per wafer

The accuracy of the defect density can be found by calculating the confidence interval for D, the total number of defects. Using this confidance interval for D in equation 2.19, the confidence interval for the defect density can be found. The 100(1-μ) percent confidence interval for D is given by,

$$y - \frac{\sqrt{y(n-y)}}{\sqrt{n-1}} t_{\mu/2;n-1} \leq D \leq y + \frac{\sqrt{y(n-y)}}{\sqrt{n-1}} t_{\mu/2;n-1} \tag{2.20}$$

y = Total Number of defects measured

n = $N_{wafer} \times N_{sites}$

$t_{\mu/2;n-1}$= 1.96 for μm = 0.05 n > 120

The defect density and 95% confidence interval for the comb and serpentine structures are given in tables 2–1 and 2–2.

Comb Structure Experimental Results (Shorts)					
Structure Size (μ)	Defect Density ($defects/cm^2$)	95% Conf. Inter.			Rel Error (%)
1.5	9.87	9.06	to	10.68	16.5
2.0	4.32	3.73	to	4.90	27.0
2.5	2.30	1.89	to	2.71	35.8
3.0	1.47	1.16	to	1.78	42.1
3.5	1.11	0.85	to	1.36	45.6
4.0	1.07	0.84	to	1.30	43.0
4.5	0.87	0.68	to	1.07	45.1
5.0	0.60	0.44	to	0.76	52.7
5.5	0.58	0.43	to	0.72	51.2
6.0	0.55	0.41	to	0.68	50.1
6.5	0.52	0.39	to	0.64	49.6
7.0	0.36	0.26	to	0.46	58.2
7.5	0.32	0.22	to	0.41	59.8
8.0	0.26	0.18	to	0.35	64.5
8.5	0.18	0.11	to	0.25	76.1
9.0	0.19	0.13	to	0.26	71.0
9.5	0.22	0.15	to	0.29	64.0

Table 2–1: Measurements of Defect Densities from Comb Test Structure

Serpentine Structure Experimental Results (Breaks)					
Structure Size (μ)	Defect Density ($defects/cm^2$)	95% Conf. Inter.			Rel Error (%)
1.5	4.01	3.32	to	4.71	34.7
2.0	2.01	1.56	to	2.46	44.5
2.5	1.08	0.78	to	1.38	56.0
3.0	0.70	0.47	to	0.92	64.5
3.5	0.47	0.30	to	0.65	73.1
4.0	0.34	0.20	to	0.48	81.4
4.5	0.29	0.17	to	0.41	83.5
5.0	0.23	0.13	to	0.34	88.0
5.5	0.20	0.11	to	0.29	91.8
6.0	0.18	0.10	to	0.27	90.5
6.5	0.16	0.08	to	0.23	94.7
7.0	0.16	0.09	to	0.23	89.2
7.5	0.16	0.09	to	0.23	86.8
8.0	0.15	0.08	to	0.21	88.0
8.5	0.15	0.09	to	0.22	83.5
9.0	0.12	0.06	to	0.17	91.8
9.5	0.11	0.06	to	0.16	93.2

Table 2–2: Measurements of Defect Densities from Serpentine Test Structure

2.5.2 Defect Size distribution

The defect size distribution can be found from the change in defect density as a result of increased track width and separation in the serpentine and comb structures. Subtracting the defect density of a 2μm structure for that of the 1.5μm structure gives the contribution to the defect density due to defects less then 2μm and greater then 1.5μm. The defect size distribution of extra material defects (shorts) found from the comb structure defect densities is given in figure 2–9. A similar distribution for missing material defects calculated from the serpentine defect densities is given in figure 2–10. Curves of the form $\frac{K}{Defect\ Size^3}$ can be found to fit the data within the error bars, derived from the 95% confidence interval of the defect densities. This suggests that the defect size distribution is proportional to $\frac{1}{Defect\ Size^3}$ in agreement with reported defect size distributions of other fabrication processes.

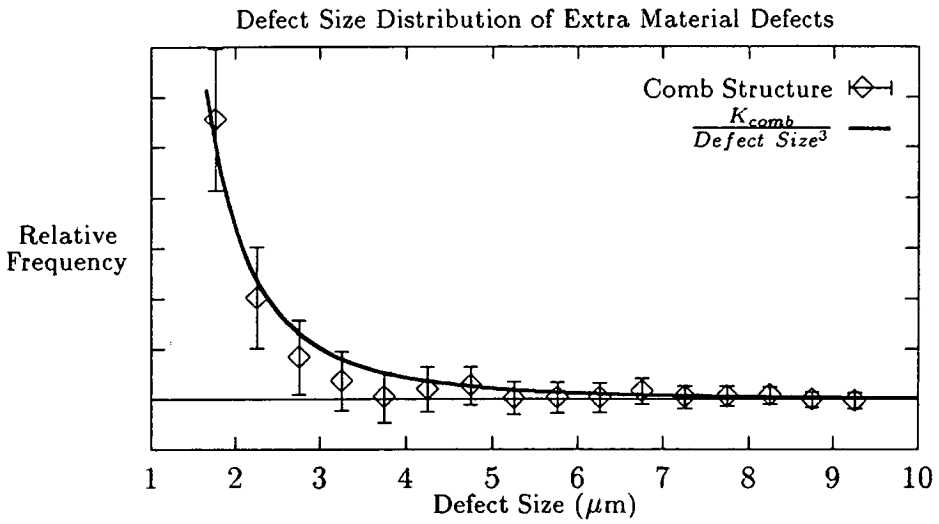


Figure 2–9: Defect Size Distribution Generated from Comb Test Structure

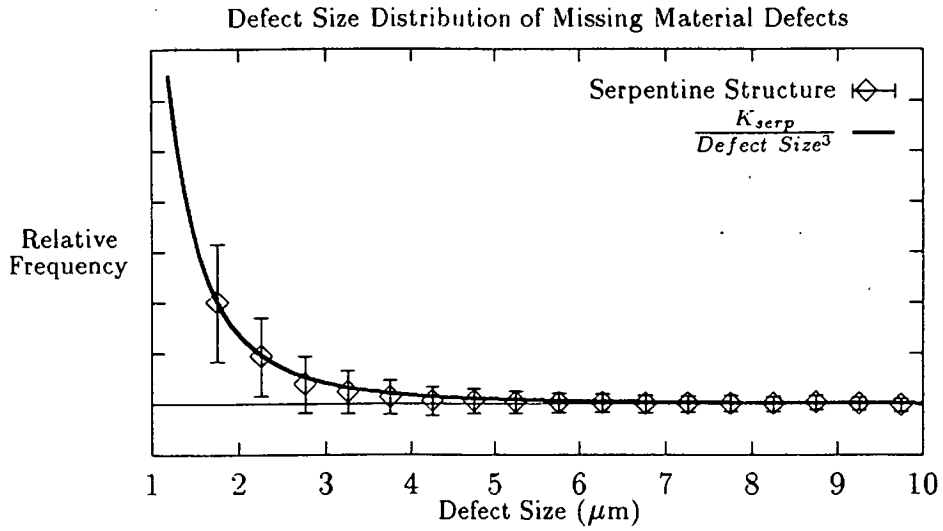


Figure 2–10: Defect Size Distribution Generated from Serpentine Test Structure

2.6 Determination of Optimal Layout Rules.

In order to optimise the yield from a process the optimal layout rules must be determined. These rules should give the IC designer a complete list of the acceptable line widths and spacings for all of the layout geometry. They should also indicate the spacing and line widths required between different layers to avoid undesired interaction between these layers e.g. metal width over polysilicon for good step coverage. Layout rules should, where possible, be kept as simple as possible to enable a designer to work with them competently and efficiently. The rules should where necessary take account of the type of circuit being designed. The design of very regular structures like memories requires very detailed design rules if high yielding circuits are to be fabricated. Logic circuits may not require the same detail. In-house designs allow for the use of a more detailed design rule set, where the designer works through many designs and where it is therefore worth the investment of time to learn and use a more complex rule base giving a better representation of the optimal layout rules. Where the fabricator is trying

to sell the process to designers in other companies it may be better to have very simple design rules to reduce the investment in retraining designers required by companies when changing between processes.

2.6.1 Test Structures.

The design/layout rules are normally determined experimentally using test structures. Test structures for layout rules must be simple, to allow easy interpretation of the results and should be worst case layouts, sensitive to misalignment and dimensional tolerances.

Ipri[9] introduces 3 test structures for CMOS/SOS, to examine some common circuit features and determine the statistical probability of success for different feature sizes. A process analysis structure is used to examine the continuity of different conductor types and their interaction. A number of different masks are made, forming conductors of different widths, to study the effects of line thickness on yield. The number of defects is found by sequentially interrogating lengths of the interconnect. A spacing array structure tests the definition of lines spaced a given distance apart. This has two levels, with the second level stepping over the first, allowing the effect of crossovers to be studied. A contact array structure tests contacts between conductors, a number of contacts chains of different lengths are interrogated to check for broken contacts.

Ipri [35] also describes a test vehicle specially designed for the MOS transistor to measure mask to mask interaction. The transistor array was designed to enable simple dc testing of cells of CMOS/MOS inverters to find open and short circuits. Strings of inverters are tested and the yield is plotted against the number of transistors. The design rules are varied in an attempt to find an optimum set of design rules. The design rules for epi-silicon to polysilicon, polysilicon gate overlap, source drain spacing and contact position are investigated with this structure

In each of the test structures described by Ipri different masks of each level are used giving results for various conductor widths and contact sizes. Using a large number of test chips it was possible to attempt to optimise the circuit feature sizes to maximise the yield of the process. These optimised features are the basis for layout rule generation.

A similar procedure has been adopted by Groves [36] but in this case hardware and software were developed to automate the collection and analysis of data. An automated stepper and computer directed prober were used to generate the large amount of data necessary. The intelligent system collects only the required data, bypassing any of the test structures that are of no interest. Even still, a large amount of data is generated that must be reduced to a manageable size and a form that can be easily interpreted. For this Groves recommends comprehensive data analysis software to reduce data to the form of graphs, scatter plots, bar charts, wafer plots etc. The highest level of data representation is the layout rules with the confidence level of all the rules.

A system of this kind is required if a process is to be continuously monitored to give yield information. This kind of information is also useful for yield prediction and to fine tune a fabrication line to obtain maximum yield levels.

Improvements in Test Device Design.

The development of optimum design rules requires a large number of test structures to gather sufficient information to make an accurate estimate of yield of the devices with different feature sizes. The test devices use a large area of silicon for the pads which are used to probe the device when under test. Walton [37] presents a method of reducing this area by introducing on-chip switching. This is implemented using a decoder on chip to switch connections from the pads to one test structure at a time. This allows the test structures to be addressed individually and greatly reduces the number of pads used. As the decoder is essential for the operation of the test structures the decoders are designed using conserva-

tive layout rules in order to reduce the possibility of failure. This technique can greatly increase the number of test devices that will fit on a test chip.

As the yield of silicon processes has improved it has become more difficult to measure yield without using large batches. This has resulted in the use of longer conductor lines as test patterns with the result that the defect causing the failure is often difficult to find in such long lines. A solution to this is given by Sprogis [38] who gives the design of a test structure called the defect diagnostic matrix (DDM). This test structure is an improvement on existing structures for yield analysis in lines of metal, polysilicon etc, that could probably be developed for use with contact chains. The DDM is used by IBM for highly automated tests of metal and polysilicon lines replacing conventional finger/serpentine test structures. The DDM isolates defects and is more sensitive to defect clusters than conventional structures. The isolation of defects allows for a more efficient investigation of faults by optical or SEM inspection. The DDM is made from an array of small defect sensitive cells. A long conductor is isolated into small areas by sequentially testing tap transistors selected by a decoder. The DDM is used as a line monitor that can be correlated to yield. It was found to predict a logic product yield to within 3 percent. The percentage of defects found by operators was 95% compared with 45% using conventional monitors. It can be designed to fit on the kerf and, because the design is simple, it can quickly be transferred to successive technologies.

2.6.2 Test Structures for Defect Clusters

In the design of fault tolerant circuits and circuits with built-in redundancy the number of defects and their distribution is an important design consideration. The majority of work on yield has made the simplifying assumption that the distribution of defects is random. This approach may be adequate for conventional circuits that require only one defect to completely disable the circuit and consequently little error results from this assumption for small to medium size chips. However for fault tolerant circuits the distribution of defects and clusters

of defects becomes a more important concern. Information on the amount of clustering of defects is useful to the engineer investigating fault tolerant architectures as it will to a large extent determine the amount of redundancy required of a circuit to maximise the yield.

The electrical measurement of fault clusters in wafers or chips is a more difficult task than measuring the random defects. This is due to the fact that devices to detect the existence of a number of closely packed faults must be fault tolerant structures themselves and give information about the regions that have a large number of faults. One attempt at a design to find the extent of defect clusters by using a fault tolerant test chip so that data on the number of defects and spatial distribution can be obtained more effectively than with a conventional test chip is given by Lewis [39].

The design consists of an array of cells of area $0.015mm^2$. Each cell contains an XOR gate and some by pass circuitry. The array consists of a number of rows of cells; the circuit can be configured to bypass any number of defective XOR gates in a row and can detect faults in the bypass wiring, but only if the fault in the wiring does not occur in a cell that has a fault in the logical part of the cell as well. In the case of a wiring and logical fault in the same cell the whole row appears defective. The data obtained from the structure is of a form suitable to be bit mapped, pin-pointing areas of defects.

This design does not meet all the criteria because two faults, one in the logic and one in the bypass circuit of the same cell, will cause a whole row of devices to fail, leaving no information on the size of the fault cluster that caused the failure, except that there were two or more faults. This means that the device is good only for characterising fault clusters that are not dense or which, through good fortune, only cause damage to either the logic or the bypass circuitry of a cell but not to both.

2.7 Statistical Determination of Layout Rules.

The design rules have a strong impact on the yield and compactness of chips. An increasing number of layout layers has resulted in a greater complexity in the calculation of design rules. The experimental calculation of design rules for optimum yield is a costly process that has to be redone every time the fabrication process is changed by the introduction of new equipment or practices.

2.7.1 STRUDEL

Razdan [40] presents a general methodology for design rule development with a statistical design rule developer (STRUDEL). STRUDEL uses the statistical variations in the manufacturing process to find a probability of failure of a particular geometry. There are local and global variations in the process, the global variations are calculated by a process and device simulator, Fabrics II [41]. Local variations are calculated from probability density functions.

The method used is to sum all the local and global variations and compare the probability calculated, (that for instance two wires will not short circuit), with some threshold value. If the probability of failure is greater than the threshold value then the separation is increased and the calculation is repeated. Iteration of this calculation takes place to find the distance at which the design rule is found to give a probability of failure that matches the threshold value.

2.8 Yield Prediction from the Circuit Layout.

The foregoing has considered yield only in relationship to the area or active area of a circuit. Another way of determining the yield of a chip is to determine the process yield of sub-elements of a circuit. The product of the sub-elements yield values gives the yield of the whole circuit. In its simplest form this means determining the yield for contacts, vias, polysilicon lines etc and splitting a circuit into those elements of known yield. The yield of such circuit elements can be found experimentally from test devices. It is also possible to determine the yield of a circuit from the defect size density distribution and the circuit layout. These procedures are complementary to the yield models that have been developed and are a natural extension. Indeed Murphy[3] only developed his yield model because he considered that the analysis of circuit geometry was too difficult an operation.

2.8.1 VLASIC.

Walker[42] presents a program VLASIC which uses a Monte Carlo method to predict the yield in CMOS circuits. In this program defects are generated using a random number generator with characteristics matching the distribution of defects found experimentally. The program places these randomly generated defects on the circuit layout and determines if the defect would case a fault. By generating large numbers of faults it is possible to simulate a layout to give results with the desired confidence level.

The main merit of VLASIC is its accuracy. It can be used to predict the effect of any defect type that can be modelled as a change in the geometry of the circuit layout. This enables very detailed analysis of a layout with the added advantage that the actual nodes at which the fault occurs are listed. As a consequence, it is possible to classify those areas of the layout which are most susceptible to faults.

This detail is only obtained at the expense of a large amount of CPU time. Because of this, the VLASIC program is generally only used to analyse RAM cell

layout. The cost of the analysis can be amortised over many (MBits) RAM cells per chip and very large production runs. Though with the advent of a parallel version of VLASIC[43] simulation of larger circuits becomes more feasible.

Relevance to LDRs

The type of detailed analysis provided by VLASIC is not required when examining the results of LDRs on large layouts, especially since the time to generate meaningful results is prohibitive. However, VLASIC could be used to analyse small "template" layout parts to determine LDRs for some of the more complex layout situations, such as increasing contact sizes where there are varying amounts of surrounding geometry.

2.8.2 Virtual Layout.

A method to predict the yield of a CMOS circuit has been given by Maly[44, 45], who developed a technique to generate a simplified virtual layout from a circuit layout in CIF[3]. This method produces a very simple layout with all or many of the properties of the original circuit layout. This method is suitable for regular structures but does not take into account the effects of defect clusters on the yield.

The virtual layout is effectively a collection of statistics that describe the circuit layout. These statistics include the sum of lengths of track within a given width range for all the interconnect layers, the number and type of contacts/vias, transistor sizes etc.

The virtual layout is generated by polygon manipulations, such as shrink/bloat operations, on the original layout. For example, to find metal 1 wires of width greater than 2μm, all metal 1 geometry is shrunk by 1 μm causing

[3]Caltech Intermediate Format

geometry 2μm or less to disappear. The remaining polygons are then bloated by the same amount. The resulting geometry is subtracted from the original metal 1 layout to give all metal 1 forming parts less than 2μm[46]. By generating statistics from other shrink/bloat values (e.g. 3μm) the combined length of track of tracks over a width range (2–3μm) are generated by subtracting the smaller shrink/bloat length of track from the larger one. The length of these 2–3μm metal 1 wires is one of the statistics that makes up the virtual layout for the circuit under consideration. A similar procedure is used to determine total amount of other track widths and their spacings.

This method produces results that are less accurate than can be achieved with VLASIC, since the layout has been simplified. For example, all wires are assumed to be straight and are grouped within pre-defined width ranges. The generation of the virtual layout removes much of the circuit detail, so that no information on "problem" areas in the layout can be made available. This method is much faster than that used by VLASIC and could be used on even the largest circuits.

Comparison of the different virtual layouts directly is possible enabling the effects of LDRs to be clearly seen. For example, a track displacement LDR will produce a reduction in the "length" of closely space tracks and a corresponding increase in those tracks that are more distant (see section 8.1.1).

2.8.3 RYE.

Chen[47] presents the most advanced yield simulation to date. The program RYE (Realistic Yield Evaluation) simulates the yield of CMOS circuits. RYE uses analytical methods rather than Monte Carlo methods of simulation; this gives a reduced CPU time. Results given by Chen suggest that the analytical method is approximately 2 orders of magnitude faster than the Monte Carlo method in a simulation where results to the same confidence level were obtained by both methods. The RYE program also exploits hierarchical layout of a circuit, avoiding recalculation on identical circuit parts. This feature can greatly speed

up a simulation depending on the nature of the circuit for which yield prediction is required.

Rye uses analytical methods to calculate the probability of failure of simple layout patterns. The probabilities for larger more complex layout, such as macro-cells, are determined hierarchically from these lower level simple layouts. This allows Rye to be used to diagnose low level faults and pin-point the actual source of the problem to specific geometry.

The capability of identifying problem areas would make Rye a useful tool to be used with LocDes in two ways:

- It can be used to highlight problem areas in the original layout to which LDRs can be directed to improve.
- It could also be used as a yield checker. It is more sophisticated than a simple design rule checker and could to be used after LDRs have been applied to a layout. This would ensure that no additional problem areas had been added by the changes intended to increase the circuit yield.

Rye can be used more readily in the process of yield maximisation because of it's greater efficiency, in comparison to the Monte Carlo method, e.g. VLASIC.

2.8.4 McYield

McYield is a software system that predicts functional yield[48]. It can produce accurate yield estimates for a 1 million transistor chip in approximately 1 CPU hour (VAX 8650). This is much faster than any other published technique. McYield uses the concept of critical areas[49]. The critical area is the area of layout in which a defect type can cause a fault. McYield uses a modified version of the Magic layout system[50] to generate these critical areas from which estimates of the yield can be made.

The magic system stores layout in terms of tiles. That is for a given layout layer the mask geometry is split into rectangular regions (tiles). The space be-

tween these tiles is "white" space and is stored as "white" tiles. Thus, the whole layout area is "paved" with tiles. This allows rapid calculation of the critical area since the both the mask geometry and the space between the geometry is represented by a similar data structure. McYield also utilises the hierarchy of the layout to reduce the required calculation.

McYield is well suited to be used in conjunction with the LocDes program. The rapid yield prediction (the most efficient reported to date) would permit a shorter cycle time to enable faster optimisation of LDRs, even when applied to large complex layouts.

The output of the system is a report on the yield loss due to each defect type and the overall yield. McYield not only indicates the total yield loss but the source of the yield loss, making it a valuable tool not only for circuit designers but also for process engineers, who can use the output to determine which circuit features are the main detractors from the yield.

2.9 Fault Tolerant Techniques.

One technique that has been suggested to increase circuit yield is the use of redundancy to give a circuit the ability to tolerate a single or multiple faults and still function correctly. The only commercial circuits to have exploited fault tolerant techniques are memory devices or chips with large memory blocks. The most popular approach is to have spare rows or columns that can be connected into the memory circuit if necessary. The improvement in yield is variable and often the figures are difficult to obtain, as manufacturers consider such information to be useful to their competitors. However, the information that is available suggests that the yield improvement can be from 30 fold increase in the yield of prototypes to a 3-5 fold increase for mature circuits (Bell labs). Intel claim as little as 1.5 fold yield improvement [51].

The use of fault tolerant circuits has lead to the idea of using the whole wafer as a circuit - wafer scale integration. There have been attempts to produce commercial wafer scale memories[52, 53].

The earliest attempts at wafer scale integration have used a discretionary wiring approach [54, 55]. This requires a mask of a unique wiring pattern for each wafer to wire the operating cells together. Later it was possible to obtain the same effect using an E-beam to define the wiring pattern [56, 57].

A knowledge of the defect density and the clustering of defects is required in the application of fault tolerant techniques, since, if there are defects in a circuit that result in the use of the available redundancy, the possibility that there are also defects in neighbouring circuitry increases, because of the existence of fault clusters. It is important that redundant units that perform the same function should be as widely separated as possible.

Circuit defects depend not only on the defect densities but also on the design rules and are probably particular to a given process. There is some disagreement on the magnitude of yield improvement obtained from relaxing design rules. This is not surprising since different processes will respond differently to design rule changes. It is considered that the relaxation of design rules for critical circuit parts (e.g. data buses) is a good thing in fault tolerant circuits[16].

Another approach to improving yield has been the repair of partially working circuits. The most obvious repair technique is discretionary wiring where circuits are probed and a wiring pattern is developed to connect the operating circuits together. This has a number of disadvantages including the contamination and damaging of circuits during the probing stage and the expense involved in individual masks or E-beam patterning of the metal layer. Dense memories may use laser repair surgery to blow a fusible polysilicon link [58] and it is also possible to make connections with a laser by changing a high resistance polysilicon connection into a low resistance connection [59, 60]. Cutting and welding with the same laser is feasible. An alternative is to have connections made or broken by electrical currents using metal or polysilicon links. The electrical techniques [61, 62] have

the same capabilities as the laser but require extra circuitry. The high current path needs large transistors and guard rings are placed around the fusible links. Memory chip manufacturers seem to be evenly split between laser and electrical fusing of links. It is also possible to make chip repairs by individual connections during packaging [63]. For example only working columns might be connected to the chip connections in partially working chips. Motorola offset some of their yield losses in their 32K EPROM by blowing polysilicon fuse links on partially working chips to form a 16K EPROM[64]. Similar techniques are used by other manufacturers[18]. Another method is to have a volatile or non-volatile register that stores information on the state of circuit blocks to determine if they are non-defective.

One argument put forward against the use of redundancy is that a defect cluster that causes a fault, subsequently overcome by the use of redundancy, may cause other faults that do not cause a failure when tested but do give rise to "weaknesses" in the circuit. This can cause the circuit to fail when in operation, after leaving the factory, causing a reliability problem. Thus it is considered by some (notably the Japanese[65]) that redundant circuits are inferior and should not be used. Redundancy is seen as a crutch for product of an inferior quality. This was the reaction in 1983 and it has not changed much. While a Japanese manufacturer, Fujitsu, will use redundancy in 600 MBit RAM wafer scale memory [53], it is at the chip level with some redundancy of the common signal lines. This contrasts with a 16 MBit RAM from an American manufacturer, IBM[66]. This large chip has bit redundancy, a word redundancy system and error correcting circuitry. The chip can survive more than 5000 defective cells and still have 96% of its error correcting ability available to deal with soft errors.

2.10 Conclusions

This chapter has given some of the background to the work described in this thesis. The importance of defect sensitivity to yield has been highlighted. Methods of yield prediction, based on the combined effects of defect sensitivity and defect size distribution, as reported in the literature have been discussed. The experimental results of defect size distribution from metal comb and serpentine test structure have been presented and these are in agreement with results reported by others.

Chapter 3

Yield Simulation.

This chapter describes the motivation behind and the development of a Monte Carlo yield simulator. This simulator can used to estimate the change in yield as a result of layout modifications such as track displacement and increased track width.

3.1 Introduction

The design of a yield simulator that is able to estimate the production yield of an integrated circuit is beyond the scope of this work. The actual percentage of working chips produced from a particular layout and fabrication process was not the prime concern. What was required was a simulator that could measure the differences in yield that were the result of small layout changes, in particular changes in metallisation layers as a result of the application of local design rules. A measure of a layout's sensitivity to circuit faults as a result of spot defects was required. The development of such a simulator was only undertaken since none was available from another source. One simulator that is suitable for this type of work is McYield[48] that should become available soon from the Microelectronics Center North Carolina (MCNC).

3.2 Mextra Based Yield Simulation.

An initial attempt was made to build a simulator from available CAD tools. This resulted in a very basic simulator developed using Mextra[67]. Mextra is a circuit extractor that generates a circuit netlist and netlist statistics from layout geometry in Caltech Intermediate Format (CIF)[68] format. This simulator was limited to extra material defect (figure 3–1) simulation. Defects were added to

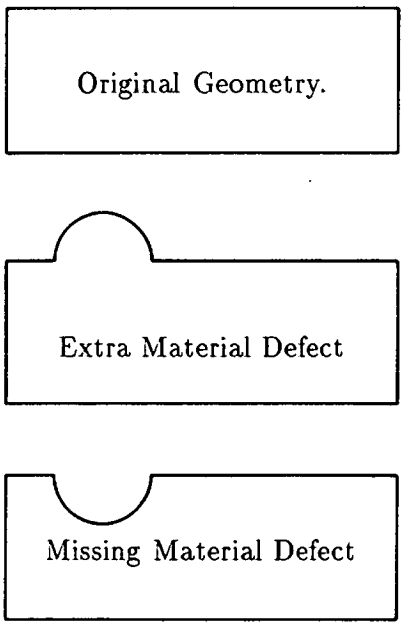


Figure 3–1: Missing and Extra Material Defects

the CIF description of the circuit under test and the resulting netlist statistics compared with the original statistics. When applying extra metal defects there are three possible outcomes,

1. **Extra node**

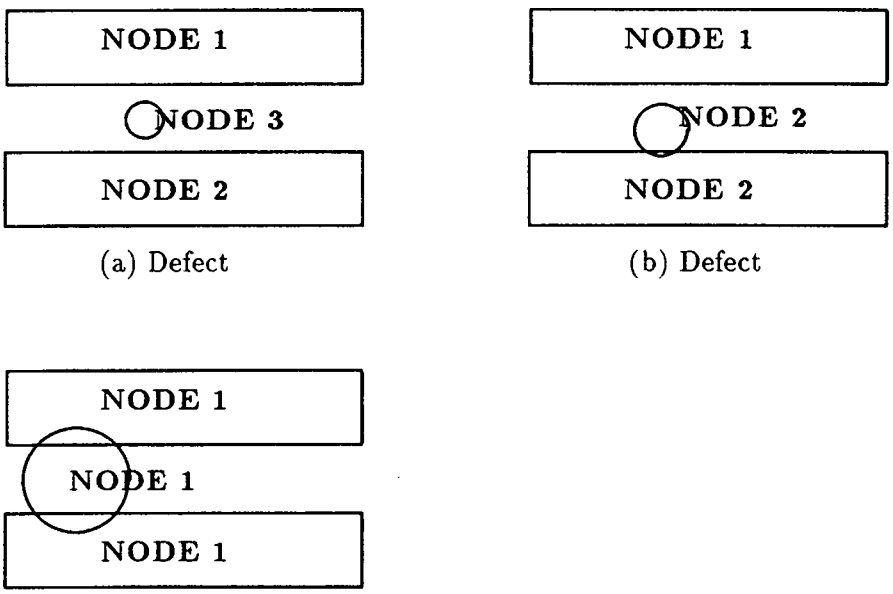
   If the defect is not placed on any other metal geometry then an extra electrical node is generated. This node consists solely of the defect geometry (figure 3–2(a)) and the defect has no effect on the structural yield of the circuit.

2. **Same number of nodes**

   If the defect is placed on existing metal geometry but does not cause a short to another node (figure 3–2(b)) then the netlist statistics will remain unchanged. The defect has no effect on the structural yield of the circuit.

3. **Fewer Nodes**

   If the defect is placed so that it comes in contact with metal geometry that forms more than one electrical node, the result is a reduction in the number of nodes as a short between these nodes has been formed (figure 3–2(c)). This type of defect would normally cause a change in the circuit function and is therefore a fault.



(a) Defect

(b) Defect

(c) Defect causing Fault (Short)

Figure 3–2: Extra Material Defects on Layout.

These defects can also cause reliability and performance problems which cannot be easily evaluated. Only defects that cause dc faults can be found using this type of analysis.

The Mextra based program produced results for extra material defects but, missing material defects (figure 3–1) could not be simulated. The program was very slow; each defect required approximately 5 CPU Sec of a SUN 3/60 using a

small circuit with 10 transistors. A very large number of defects over a range of defect sizes is required to give an accurate picture of the susceptibility of a circuit to defects. Thus the execution of the program to give useful results took many hours.

The program is inefficient, the main reasons for this are,

- Each defect requires two unix processes to be generated, the defect placement and the mextra process.
- The programs used, cat[1] and mextra, are not tuned to the job in hand. This is particularly true of mextra in which there are many redundant routines and operations.
- Communication between processes is via the file system; *cat* generates a file which is read by mextra which produces an output file of statistics that is read by another process to interpret and summarise the results.

While the mextra simulator provided some initial results it also showed that a yield simulator would have to be purpose built to be fast enough to give accurate results in a reasonable length of time. This and the need to simulate missing material defects resulted in the design and implementation of the Monte Carlo yield simulator.

3.3 Monte Carlo Yield Simulator

The Monte Carlo Yield Simulator is a purpose built simulator, written in the C programming language. The program simulates both missing and extra material defects on a CIF format circuit layout. The basic functions of defect placement and the resulting netlist analysis for extra material defects is performed by one UNIX^{TM} process with routines designed specifically to perform these tasks.

---

[1] cat is a standard unix command, used in this case to append a defect to an existing CIF file.

Missing material defects use an extra process per defect, though communication is through a socket[2] rather than the file system.

The simulator can be broken into three separate operations, initialisation, defect placement and defect analysis.

3.3.1 Initialisation

The initialisation procedure loads in the CIF representation of the circuit and processes the layout to produce,

- A spatial data structure (described in Chapter 7) containing the circuit layout boxes, each labelled with the electrical node number that it helps form.

- An array of linked lists; these lists contain pointers to all the geometry that forms a single electrical node. Each node has its own list.

The algorithm for the initialisation is shown in figure 3–3. The first step is to read in and parse[3] the CIF file. Each physical layer of layout (metal 1, polysilicon etc.) is stored in an array as layout boxes and a binary tree is generated that references the data held in the array. The binary tree allows very fast searches to be performed (Section 7.5). In order to generate electrical nodes the transistors

---

[2]A UNIX socket allows interprocess communication.

[3]The CIF file is parsed using a modified version of cif2ps, a public domain program that generates postscript output from a CIF input. This modified version flattens, removes the hierarchy, of the CIF input, giving as output the list of layout boxes that describe the complete layout. Cif2ps was originally written by Arthur Simoneau, The Aerospace Corporation, El Segundo, Calif with contributions by Marc Lesure, Arizona State University, Tempe, AZ and Gordon W Ross, The MITRE Corporation, Bedford, MA

```
Procedure Initialise
    begin
      Initialise data structures.
      Read in CIF File.
      Load layout data into Binary trees.
      Make symbolic layer for transistors (ACTIVE AREA AND POLYSILICON)
      Make ACTIVE AREA equal to ACTIVE AREA NOT TRANSISTOR layer
      for all layout Boxes do {
        Give Box an initial unique node number.
      }
      for all track layers do {  /* Metal1, metal2 Polysilicon etc. */
        for all Box of layer do { /* For every piece of layout of type layer. */
          Generate List of all other boxes of same layer that touch Box.
          Label all boxes in List with the same unique node number.
        }
      }
      for all Contact layers do { /* Contact Cut, Via. */
        for all Contacts of Contact layer do {
          Find the node numbers (Node1, Node2) of both track layers forming contact.
          Label all boxes of Node2 on all track layers
          with the node number Node1.
          Label Contact with node number Node1.
        }
      }
      Generate array of electrical node linked lists.
    end
```

Figure 3–3: Yield Simulator Initialisation Algorithm.

must be identified. The transistors are found by producing the logical AND of polysilicon and active area (figure 3–4(b)). In preparation for node generation the source and drain of the transistor are isolated by making a new active area, generated from the logical NOT of the transistor layer and the active layer (figure 3–4(c)). The original active area is then discarded.

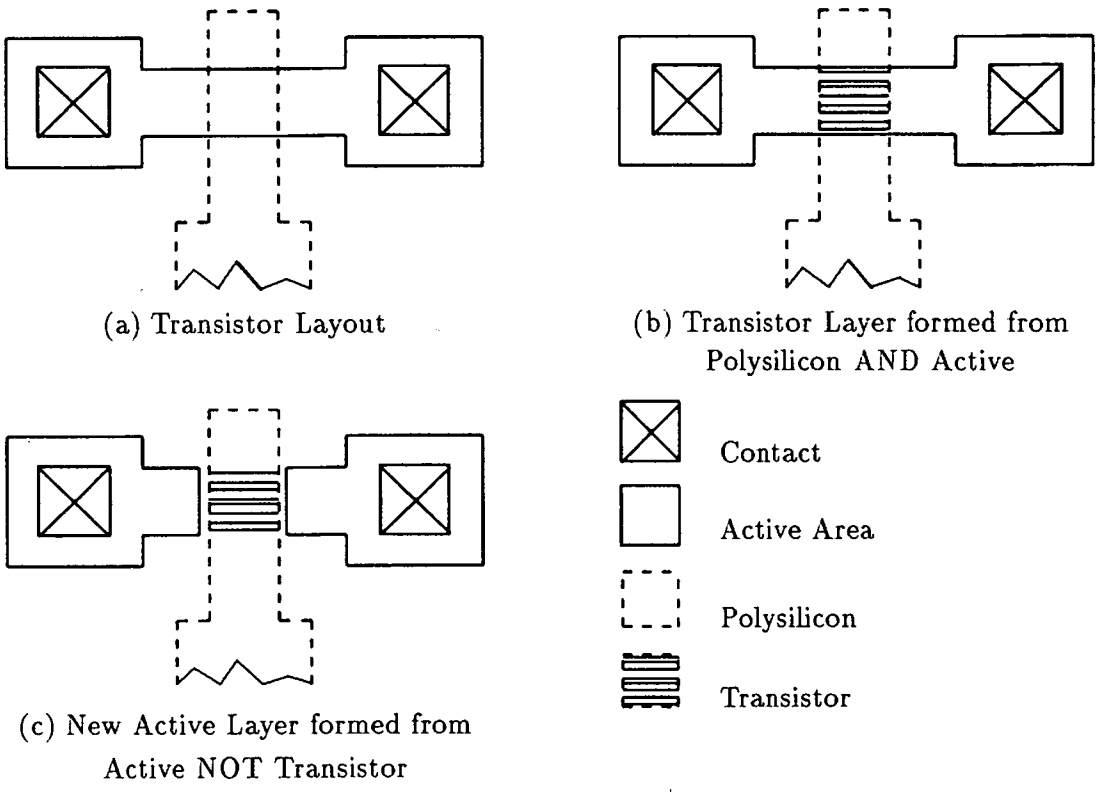


Figure 3–4: Transistor Source and Drain Generation from Layout.

Electrical Node Generation.

The new active area and transistor layer are also stored in arrays referenced by binary trees. Each layout box is given a unique node number. This is to ensure that even if a box is unconnected to any other layout it still has a unique number. The electrical nodes are formed by first cycling through all the boxes of track layers (polysilicon, active area, metal1 and metal2) to generate groups of boxes of the same layer that are connected. This is done by marking connecting boxes with the same unique node number. These groups are then formed into

```
Procedure Defect Size and Position.
        begin
          Max = DefectDis(MINSIZE).
          do
            Size = random()×(MAXSISE−MINSIZE)+MINSIZE.
          Until random()×Max <= DefectDis(Size).
          Xcoord = random()×(Xmax-Xmin)+Xmin.
          Ycoord = random()×(Ymax-Ymin)+Ymin.
          Generate Defect.
        end
```

Figure 3–5: Defect Generation Algorithm.

electrical nodes by cycling through all the contacts/vias to find groups of boxes that are electrically connected in the circuit. The boxes in these connected groups are then given the same unique node number marking then as part of a single electrical node.

A linked list for each node, referencing every layout box that is part of the node, is formed and placed in an array of these lists. These lists are used later to speed up the generation of sub-circuits when missing material defects are evaluated.

3.3.2 Defect Placement.

The size and position of defects are generated using the algorithm shown in figure 3–5. The layout area is defined by Xmin, Ymin, Xmax and Ymax. This algorithm generates random[4] defects with the distribution of DefectDis(Size), assuming that the maximum number of defects is found at the minimum defect

[4]The random() routine used is a standard unix function for the generation of pseudo random numbers.

size. This assumption holds true for all reported processes defect distributions at "interesting" defect sizes, but if not Max could be defined as the most frequently found defect size.

The value MINSIZE is the minimum defect size required for the simulation. This is the minimum design rule separation, for extra material defects, or the minimum design rule width for missing material defects. Defects smaller than this do not cause structural circuit faults. If the minimum width is the same value as the minimum separation then, the set of defects generated can be used for both defect types.

The Xcoord, Ycoord and Size are used to generate the defect. All defects are simulated as approximations to a circular defect, since using a circular defect would greatly increase the execution time of the program with little increase in the accuracy of the result. The defect is represented by five[5] rectangles (figure 3–6(a)). It is possible to obtain better approximations to a circle with more rectangles (figure 3–6(b)) but once again this slows execution with little increase in the accuracy of the result.

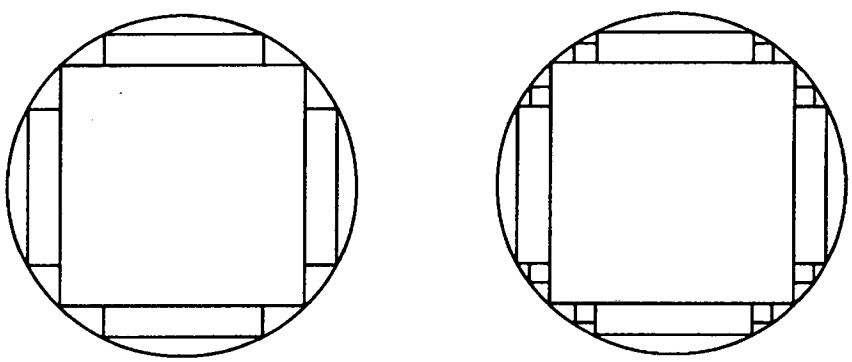


Figure 3–6: Circular Defect Approximation.

[5]In practice it is possible to use three overlapping rectangles.

```
Procedure ShortTest
        begin
          Get Defect.
          NODENUM = -1 /* Guaranteed not to exist */
          SHORT = FALSE
          for all Boxes in Defect do {
            Find List of circuit boxes touching defect Box.
            if List has members then {
              for all Members of List do {
                if NODENUM = -1 then
                  NODENUM = node number of Member
                else if NODENUM NOT = node number of Member then
                    SHORT = TRUE
              }
            }
          }
          return SHORT.
        end
```

Figure 3–7: Extra Material Defect Fault Detection Algorithm.

3.3.3 Defect Analysis – Shorts

The algorithm used to determine whether an extra material defect has caused a short is shown in figure 3–7. In this algorithm a defect is generated and all the boxes that form the defect are tested to determine whether they interact with any circuit layout geometry. When the first interaction is found the node number of the geometry is noted. If there are further interactions with the defect, the node of that geometry is compared to the first interacted node. If these nodes are different then a short has occurred. That is the defect has connected two electrically separate nodes, causing a circuit fault to be created, that will change the circuit function.

This assumes that all nodes in a circuit are important to the circuit function and that shorted nodes do actually cause circuit malfunction. In most real circuits this is a reasonable assumption, though it is possible that redundant layout or geometry used as labelling, for mask inspection etc, could give false results. Therefore, it is necessary to inspect layouts to be simulated to ensure that such problems will not arise.

3.3.4 Defect Analysis – Breaks

The algorithm used to determine whether a missing material defect has caused a break is shown in figure 3–8. In this algorithm a defect is generated and the interaction of the defect with the layout is tested. If there is no interaction then there is no fault (figure 3–9(a)), otherwise all geometry for each node that intersects the defect is extracted using the linked lists, formed during the initialisation procedure. This geometry is used to form a sub-circuit. The logical NOT of defect and sub-circuit is generated, representing the layout of the sub-circuit with a missing material defect. This new sub-circuit is processed to determine the number of nodes, in the same way as the original circuit layout was processed[6]. The number of nodes in the sub-circuit with the defect is compared to the original number used to form this circuit. If there are the same number of nodes as there were originally a circuit fault has not been generated (figure 3–9(b)) but, if there are a greater number then a fault has been generated (figure 3–9(c)).

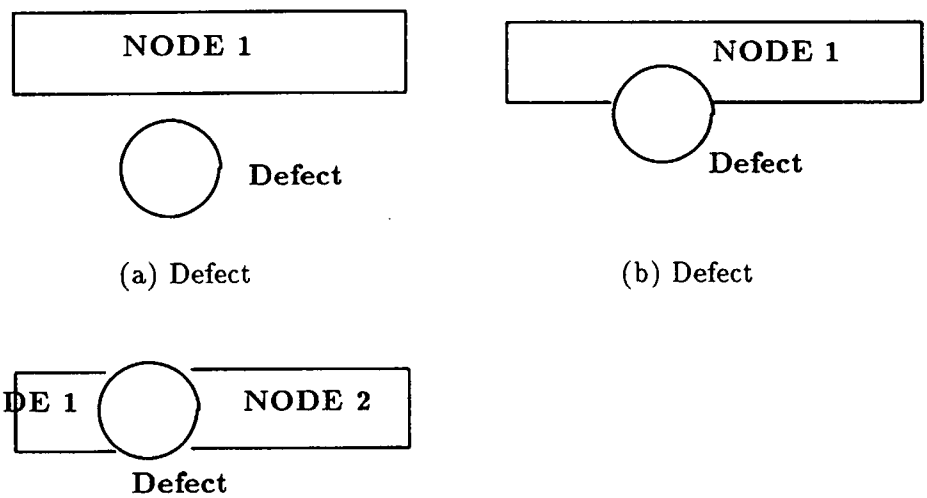
It has been assumed that all geometry is important to the circuit function and that any break will cause a circuit malfunction. This is not necessarily true. For example power lines may form a network that can survive a number of breaks. If only a single cell is "shown" to the simulator it is unable to determine whether

---

[6]This is done by opening a socket to another process running the same program, but with different arguments, that returns, via the socket, the number of nodes in the sub-circuit.

```
Procedure BreakTest
        begin
          Get Defect.
          BREAK = FALSE
          for all Boxes in Defect do {
            Add Nodes of circuit boxes touching defect Box to NodeList.
          }
          if there are nodes in NodeList then {
            Make Sub-circuit from Nodes in NodeList.
            Make Sub-Circuit = Sub-circuit NOT Defect.
            Generate Nodes from Sub-circuit.
            if Nodes in Sub-circuit > Nodes in NodeList.
              then BREAK = TRUE.
          }
          return BREAK.
        end
```

Figure 3–8: Missing Material Defect Fault Detection Algorithm.



(a) Defect

(b) Defect

(c) Defect causing Fault (Break)

Figure 3–9: Missing Material Defects on Layout.

the power supply has been disconnected by a single break. It is also possible that some geometry is redundant and does not fulfill a circuit function. These problems could be overcome with some programming effort but this has not been done. Instead care is taken that simulated circuits do not contain redundant geometry. This is quite easy to do since only small circuits are simulated because of the long execution times. Power lines do not pose a problem since the simulator is primarily intended to measure relative yield, that is the difference between two similar circuits. It can be assumed that a power line break is a circuit fault if the same assumption is made of all the cells simulated to find the relative yield. Also, power lines tend to be wider than other tracks and are therefore less prone to breaks.

3.4 Results

The Monte Carlo yield simulator gives, as output, a list of the defects that resulted in circuit faults. The defect size, position and defect type are given.

Xcoord	Ycoord	Defect Size	Defect Type	Fault
3402	102	5.3	Missing	Break
1039	3242	6.7	Extra	Null
1129	2157	3.5	Missing	Null
500	1532	2.9	Extra	Short
etc.				

This raw output must be further processed to give a summary of the results in a more understandable form. This is done using a number of awk[7] scripts to analyse the data and generate output for a variety of graphs and tables.

---

[7]Awk is a pattern scanning and processing language available on most versions of Unix

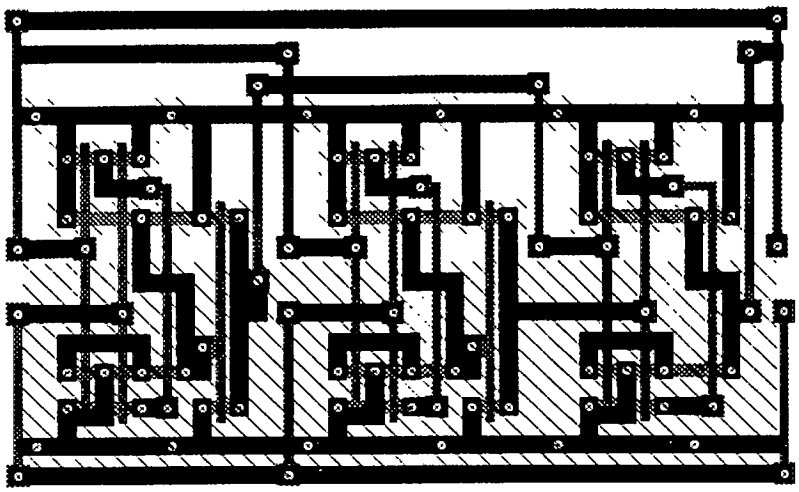


Figure 3–10: CMOS cell used in Yield Simulation.

3.4.1 Simulation Results.

The circuit in figure 3–10 was used in a simulation of 100,000 metal defects. The metal is the darkest layer in the figure. The extra material defects were generated using a size distribution of $\frac{1}{(Defect\ Size)^3}$ with a minimum size of 2.5μm, which is the minimum separation of metal to metal separation, Missing material defects have a lower bound of 3μm as this is the metal width. The maximum defect size generated for both defect types was 15μm. The results have been summarised in table 3–1. This table gives the total number of defects in a size range and the number of circuit faults that resulted from these defects, the 95% confidence interval for the circuit faults and the relative error of this interval.

The graph of figure 3–11 shows the percentage of faults that occur as a result of a given defect size. This shows that defects that are a size comparable to the minimum separation of metal have a very low probability of causing defects and that this probability increases as defect size increases. This does not imply that these smaller defects can be ignored. The graph in figure 3–12 shows the relative distribution of circuit faults and the size of the defect that caused the fault. This indicates that because of the larger number small defects present, a significant number of faults are caused by these defects.

Yield Simulation Results (Shorts)						
Defect Size (μ)	Total Defects	Faults	95% Conf. Inter.			Rel Error (%)
2.5 – 3.5	46467	495	451.63	to	538.37	17.5
3.5 – 4.5	19373	792	737.98	to	846.02	13.6
4.5 – 5.5	9708	841	786.67	to	895.33	12.9
5.5 – 6.5	5516	791	739.98	to	842.02	12.9
6.5 – 7.5	3381	688	642.11	to	733.89	13.3
7.5 – 8.5	2281	639	596.95	to	681.05	13.2
8.5 – 9.5	1589	553	515.77	to	590.23	13.5
9.5 – 10.5	1160	477	444.14	to	509.86	13.8
10.5 – 11.5	871	424	395.07	to	452.93	13.6
11.5 – 12.5	667	355	329.72	to	380.28	14.2
12.5 – 13.5	493	301	279.76	to	322.24	14.1
13.5 – 14.5	408	262	243.00	to	281.00	14.5
Yield Simulation Results (Breaks)						
4.0 – 5.0	41965	1324	1253.82	to	1394.18	10.6
5.0 – 6.0	20197	2183	2096.51	to	2269.49	7.9
6.0 – 7.0	11823	1972	1892.55	to	2051.45	8.1
7.0 – 8.0	7742	1574	1504.59	to	1643.41	8.8
8.0 – 9.0	5235	1150	1091.28	to	1208.72	10.2
9.0 – 10.0	3672	898	846.94	to	949.06	11.4
10.0 – 11.0	2686	687	642.67	to	731.33	12.9
11.0 – 12.0	2125	562	522.14	to	601.86	14.2
12.0 – 13.0	1566	434	399.27	to	468.73	16.0
13.0 – 14.0	1227	308	278.22	to	337.78	19.3
14.0 – 15.0	1032	249	222.05	to	275.95	21.6

Table 3–1: Results of Extra and Missing Material Defect Simulation on a CMOS layout.

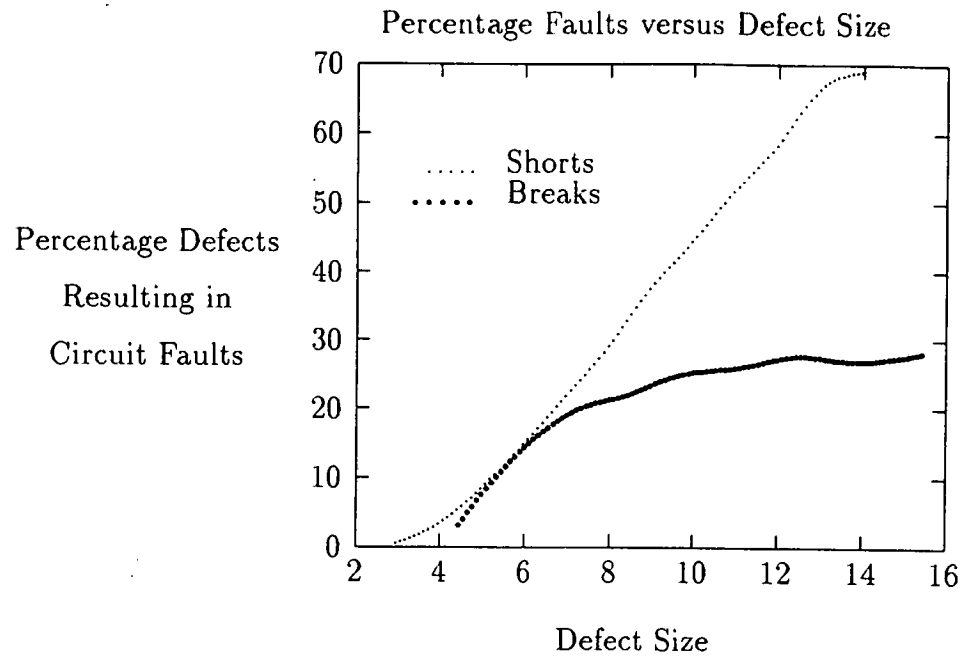


Figure 3–11: Graph of Faults(%) versus Defect Size on CMOS layout.

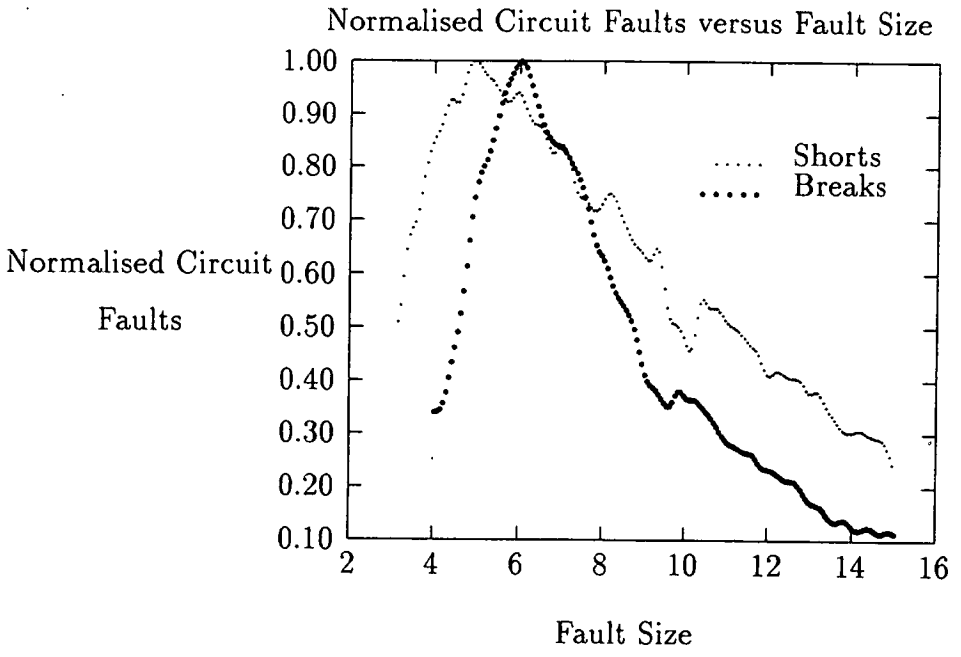


Figure 3–12: Graph of Normalised Faults versus Fault Size.

3.5 Conclusions

This chapter has explored the design of a Monte Carlo yield simulator. The simulator is capable of measuring the defect sensitivity of metal and polysilicon tracks. It is primarily intended to be used to measure the effect of layout modifications, such as changes in track width and separations, on the defect sensitivity of a circuit layout. Results generated using this simulator from layouts processed by the LocDes program are given in chapter 8.

Chapter 4

Design Rules.

This chapter introduces the IC layout design rules that define the minimum sizes and separations of layout components. The effect of these rules on circuit area and their potential use as a method of increasing yield are explored.

4.1 Introduction

The layout of integrated circuits is bound by a set of design rules. The design rules give minimum values for track width, track separation and contact sizes (figure 4–1). They also set out the minimum requirements to create active devices. Design rules for transistors define the minimum overlap and separation of the N, P and well implants.

There are occasions when specialist circuit parts, such as RAM, use a different set of design rules, that have been optimised for that particular circuit type. However, normally the design rules are applied over the whole of layout area and will therefore be referred to as Global Design Rules (GDR).

The purpose of these rules is to maximise the yield, performance and reliability of circuits. The rules are obtained from an analysis of the fabrication process and are usually particular to that process.

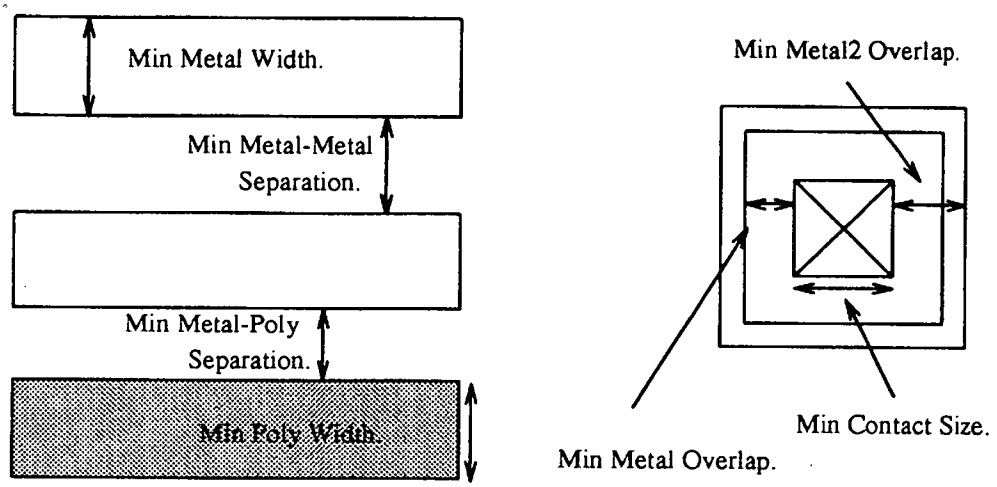


Figure 4–1: Examples of Global Design Rules.

4.2 Influence of Design Rules on Circuit Area.

The area of a circuit layout is proportional to the square of the feature size. So that if all design rules were scaled by an equal amount the resulting circuit area would be Scale factor^2 $\times$ area. Design rules are rarely scaled in this way. Improvements in fabrication technique do not take place equally in all process steps simultaneous. The simple relationship between area and scale factor is instructive but in practice can only be used with a completely revised fabrication process that is very similar to an existing process.

4.3 Single Design Rule Changes.

The influence of single design rule changes on the layout area of a circuit was studied using layouts generated by the STIX [69] circuit compactor. STIX is a symbolic[1] layout editor that fleshes then compacts symbolic geometry to produce mask level layout suitable for fabrication. These operations are controlled by design rule files that determine the width and separation of layers and the construction of circuit elements such as transistors and contacts. STIX was used to generate a number of different layouts of two circuits, a dynamic shift register (figure 4–2) and a six transistor RAM cell (figure 4–3). Various circuit layouts were generated from the symbolic representation of the circuit, using design rule sets that had a single design rule modified in each version of the circuit produced. This enabled the effect of that rule on circuit area to be determined.

STIX was used in preference to hand layout, as a large number of examples were required. It would have been too time consuming to do this by hand and the results would have depended to some degree on the alertness of operator and time taken for each layout, producing uneven results. Since a lot of commercial layout is presently automated it is not a serious handicap to use this method. In general, hand-crafted layout is more compact than machine layout but since the point of interest is changes in layout area, the difference between hand and machine layout will not be great.

The design rules tested were,

- Transistor Overlap of Gate.
- Metal Width.

---

[1] A symbolic layout editor takes input in a simlified form that represents the actual mask levels. E.g. transistors are defined as a symbol that is converted to the correct mask layout by the software at the layout generation stage.

- Contact Size.
- Contact Overlap.

The two circuits were tested for each design rule. These give different results showing that layout area is dependent on both the design rules and the nature of the circuit. Both circuits were generated initially using ES2[2] 2 micron CMOS design rules. These rules were then adjusted about the normal ES2 value to determine the effect on the layout area. In order to make the results more general and easier to interpret the design rules have been normalised.

4.3.1 Transistor Overlap of Gate.

A transistor in CMOS technology is formed at the layout level by placing a polysilicon track over active area. The active area will require threshold and well implants depending on the transistor type and the process used. Due to the possibility of misalignment between mask stages it is important that the polysilicon has extra overlap of the active area (see figure 4–4). There will always be a degree of misalignment between the masking layers because of limits in placement accuracy. The amount of misalignment that can be tolerated is to some extent dependent on the transistor overlap and the overlap used for other polysilicon structures such as contacts.

The results for transistor overlap of gate for the shift register and RAM cell are given in figure 4–5 and figure 4–6. These indicate that the area of the shift register is relatively insensitive to the overlap. Even at 1.5 times the original value the circuit area is increased by only 0.1%. The RAM cell is much more responsive. Over the region tested the cell area varies by approximately ±3%. The difference between the two circuits occurs because the RAM cell is dominated by transistors that are very closely packed in comparison to the shift register cell

[2]European Silicon Structures

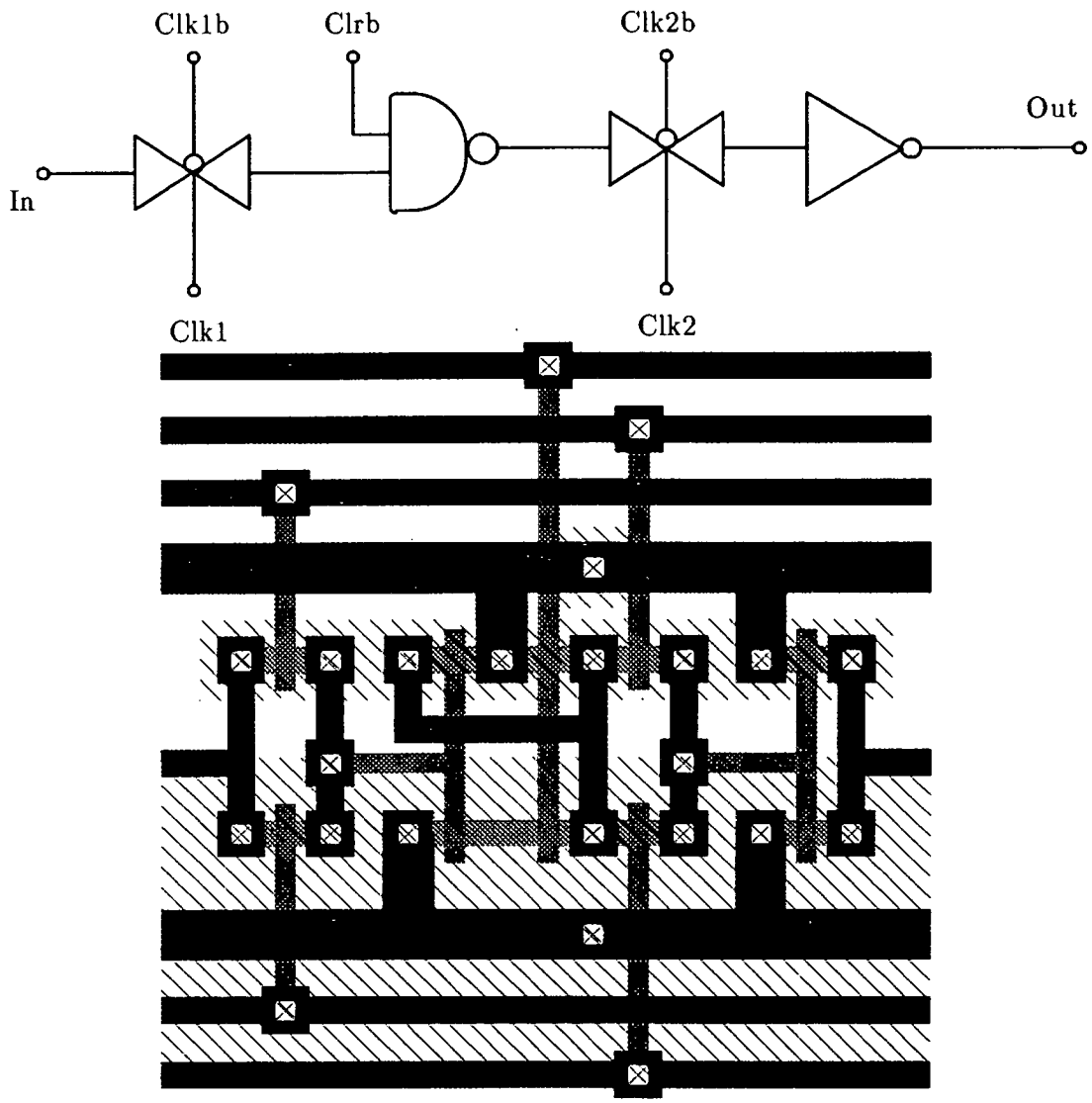


Figure 4–2: Dynamic Shift Register Cell

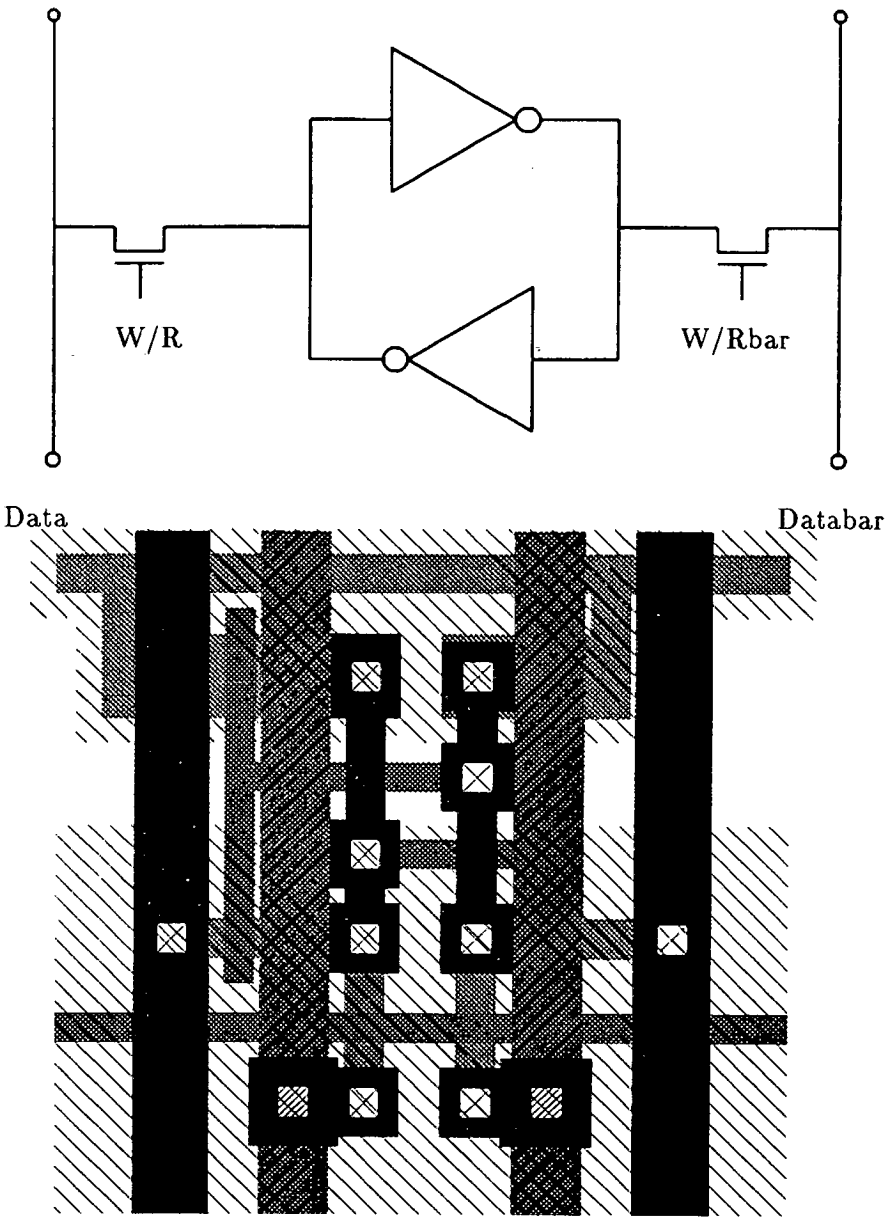


Figure 4–3: 6 Transistor RAM Cell

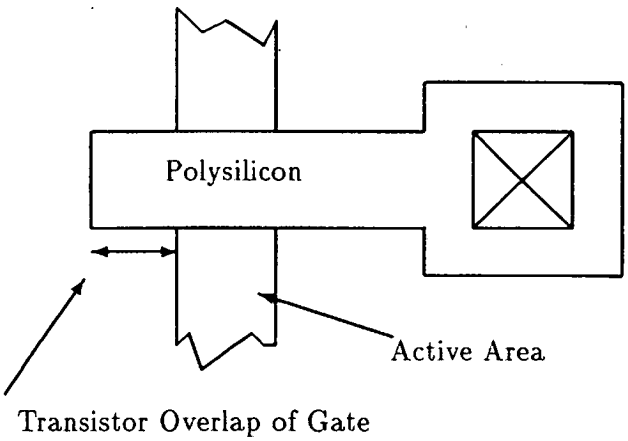


Figure 4–4: Transistor Overlap of Gate

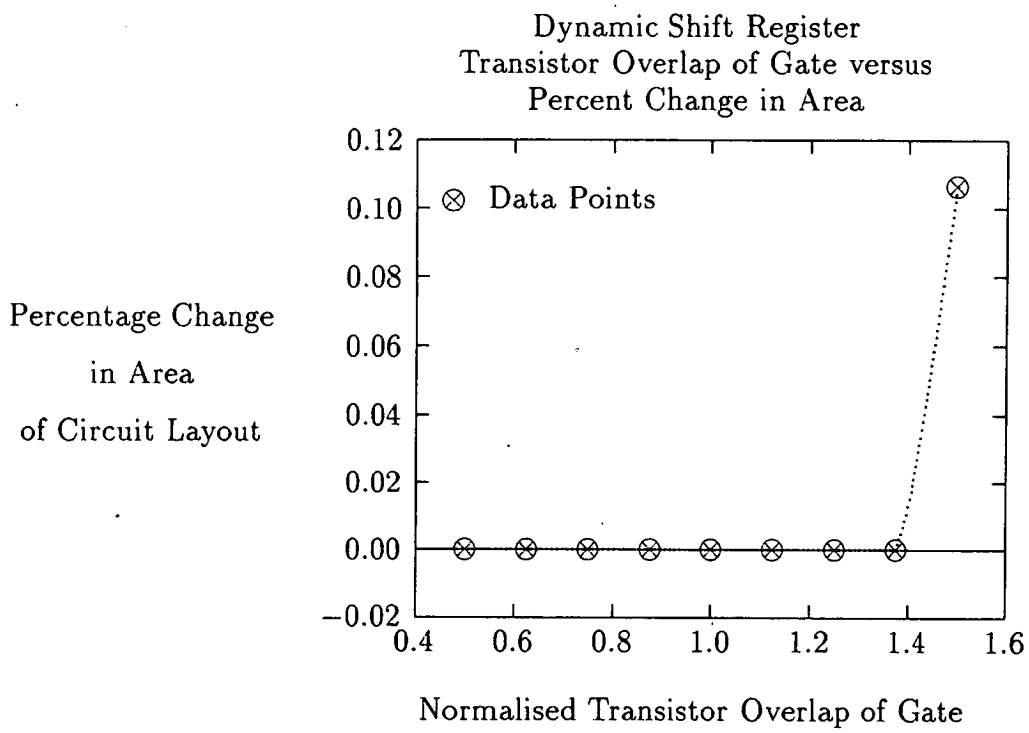


Figure 4–5: Results from Changing Transistor Overlap of Gate:Shift Register

that has interconnect and contacts acting as separators between transistors. This causes the RAM cell to be sensitive to transistor design rules and the shift register to be largely unaffected.

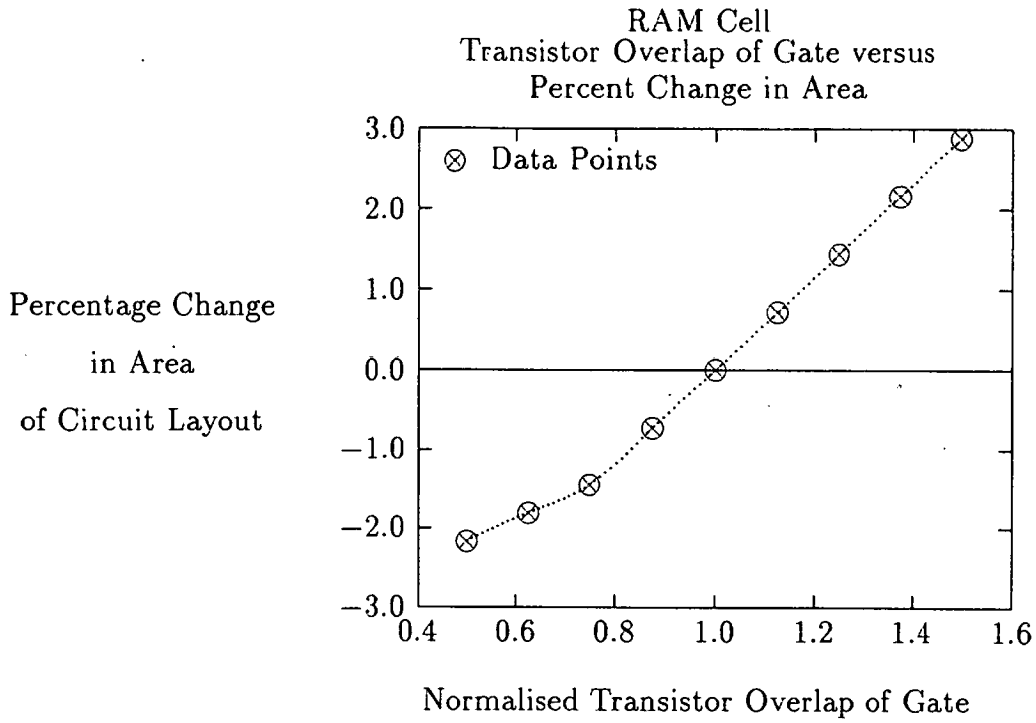


Figure 4–6: Results from Changing Transistor Overlap of Gate:RAM Cell

4.3.2 Metal Width.

Metal is used as interconnect between active elements such as transistors. It is used to make contacts with polysilicon and active layers. If there is more than one metal layer this too can be connected to the first metal layer. The RAM cell and shift register both use only a single layer of metal for interconnect.

The width of the metal was adjusted in both the RAM and shift register cells. The RAM cell was totally unaffected by changes in the metal, over the tested range of 75% to 125% of the original width. The shift register results over the same range are given in figure 4–7. This shows that the cell area varies by approximately ±1.5%.

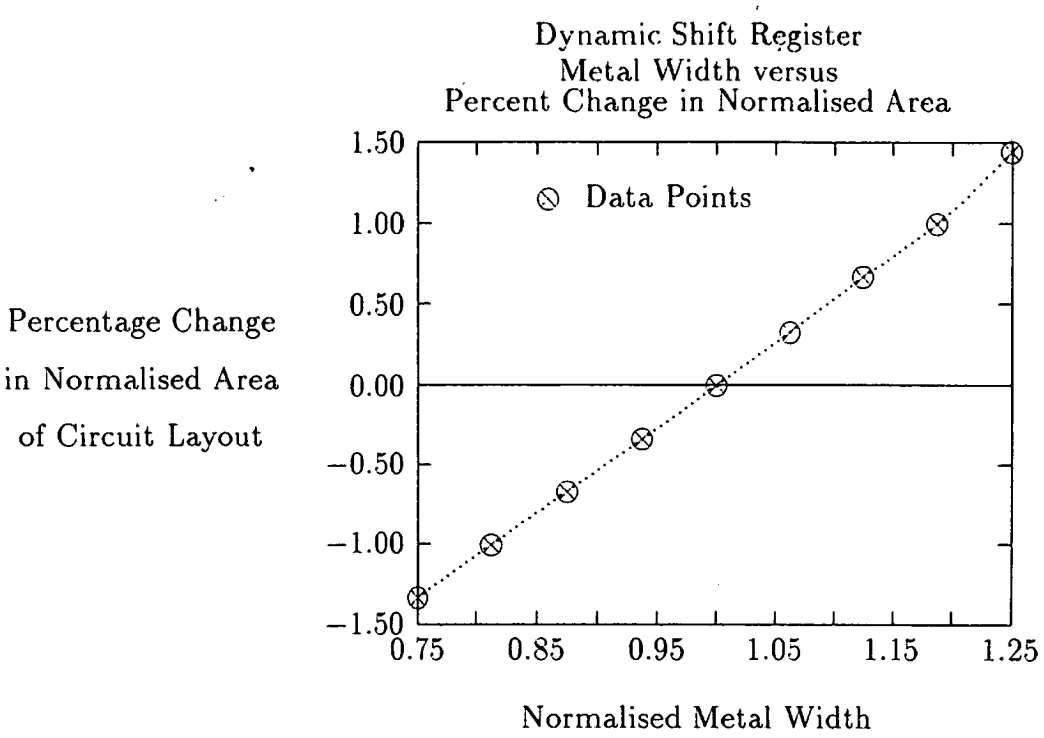


Figure 4–7: Results from Changing Metal 1:Shift Register Cell

4.3.3 Contact Size.

To make an electrical connection from one conducting layer to another of a different type contacts are used. Usually contacts are from metal to another layer e.g., polysilicon, active area or a second metal layer.

The design rule determining contact size in the two circuits was adjusted over a range of ±50% of the original size. The resulting cell area changes are given in figure 4–8 and figure 4–9. These show that the contact size has a significant impact on the area of both circuits.

4.3.4 Contact Overlap.

The contact overlap is used to provide protection against misalignment in much the same way as transistor overlap (section 4.3.1). Misalignment of contact layers causes either contacts to be badly formed, having a high resistance, or an open contact that can result in a circuit fault.

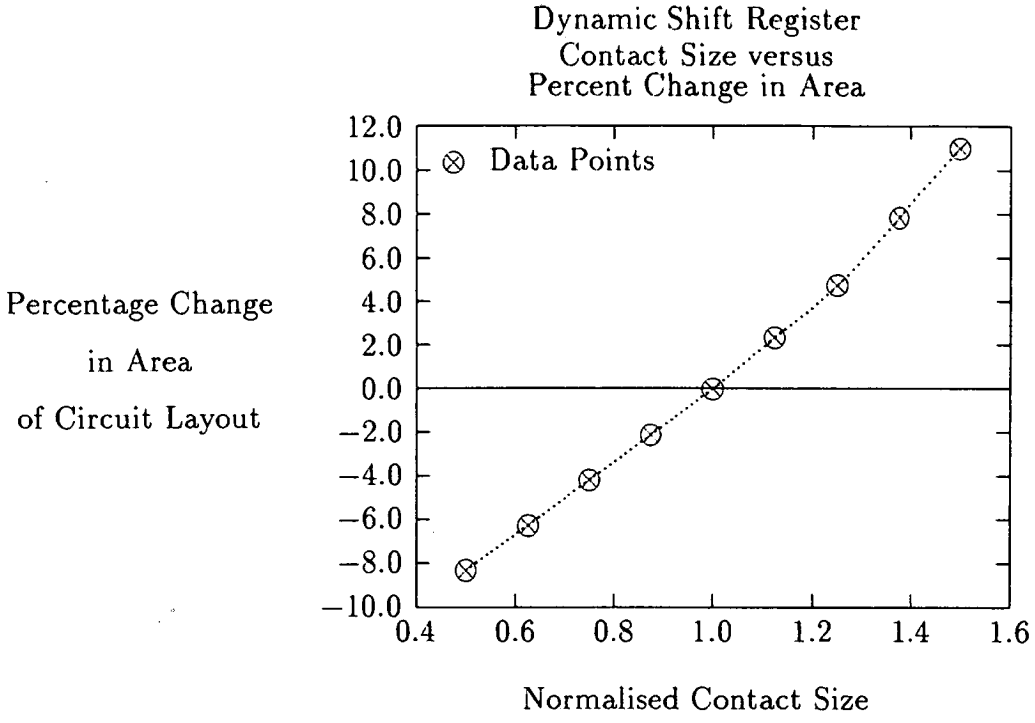


Figure 4–8: Results from Changing Contact Size:Shift Register Cell

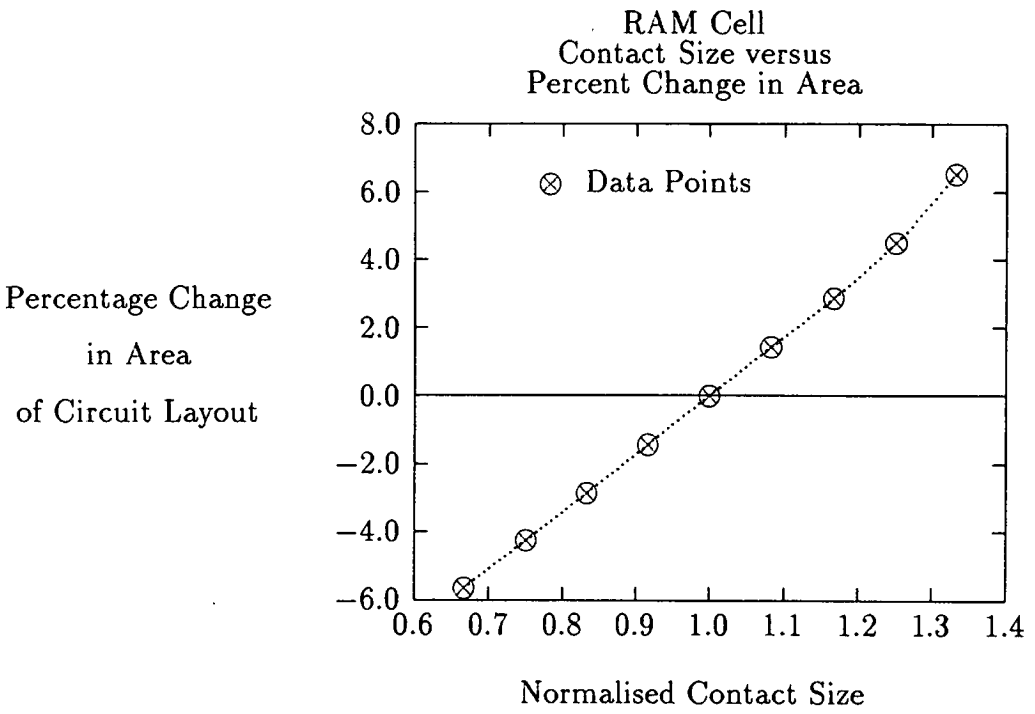


Figure 4–9: Results from Changing Contact Size:RAM Cell

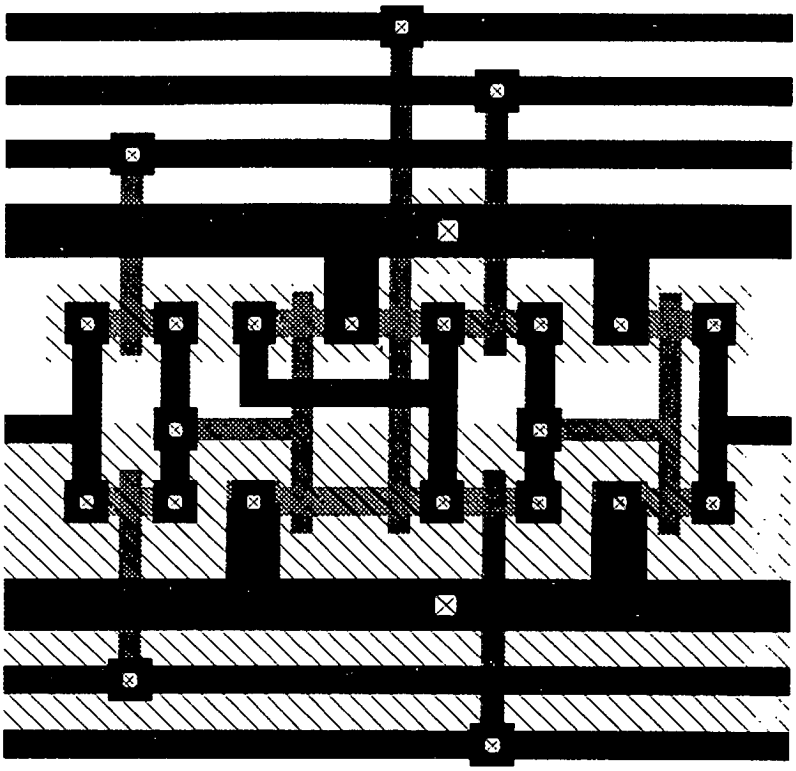


Figure 4–10: Shift Register with Contact 75% of Original Size.

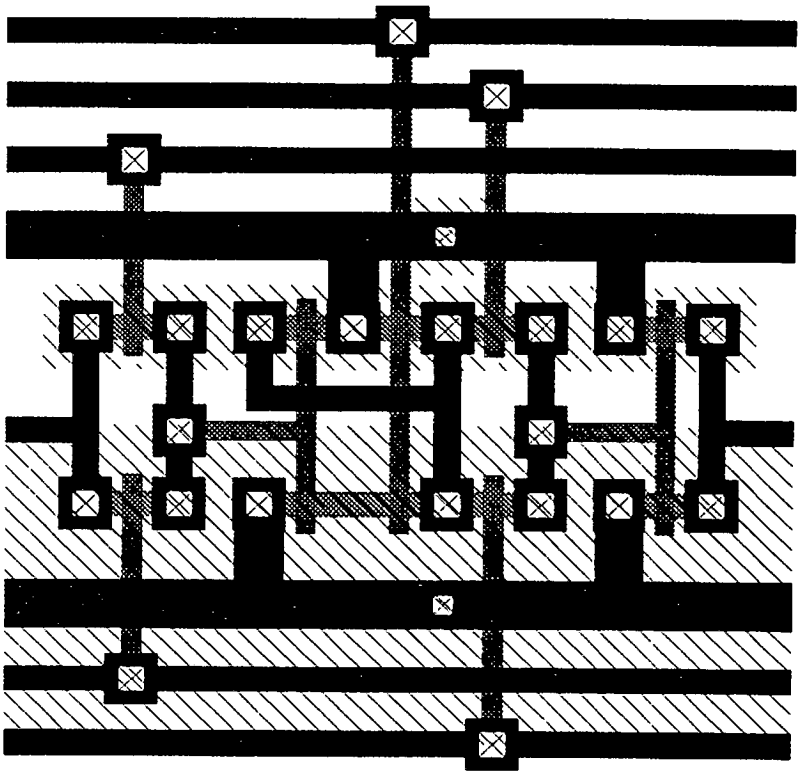


Figure 4–11: Shift Register with Contact 125% of Original Size.

Changes in contact overlap are likely to have an effect similar to increasing the contact itself. The results, given in figure 4–12 and figure 4–13, obtained by adjusting contact overlap by ±50% in the RAM and shift register cells reflect this similarity.

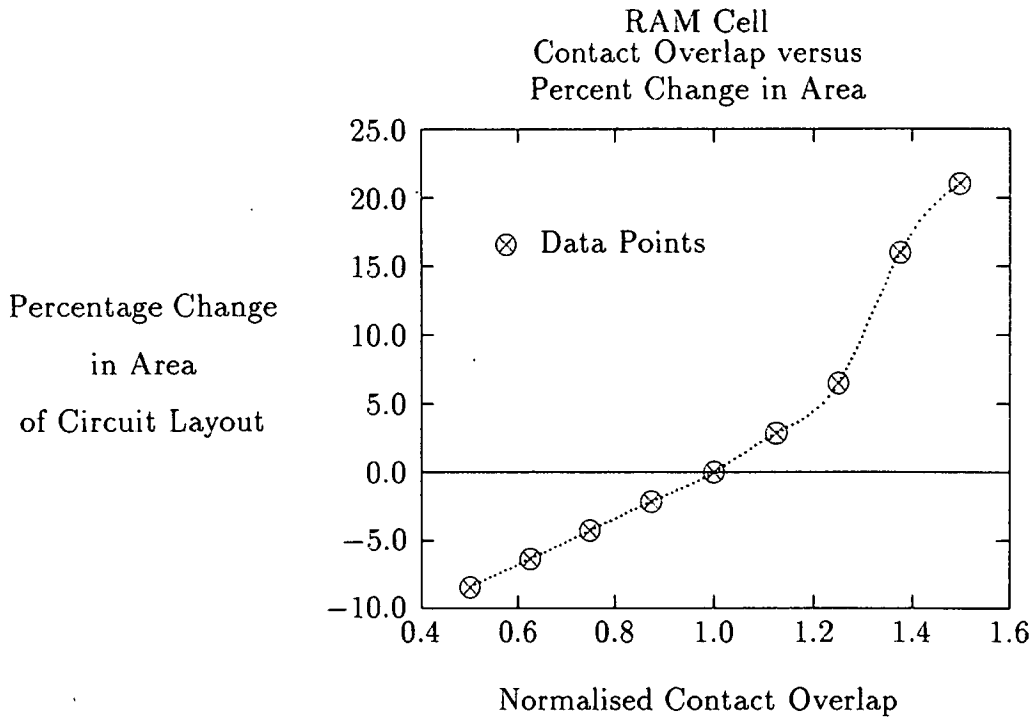


Figure 4–12: Results from Changing Contact Overlap:RAM Cell

4.4 Layout Style.

The yield of a circuit is not only dependent on the layout rules but on the implementation of the circuit and the layout style used. If the wrong design decisions are made, they can have just as great an effect on the circuit area and yield as the design rules. As an example of this consider a change in the layout of the shift register. The clock and data lines are originally on a single metal layer. Five new layouts are generated with increasing use of a second metal layer to carry these signals. The resulting layouts for 5 and 3 metal lines are shown in figures 4–15 and 4–16. The effect of the increased metal 2 on the area of the circuit is

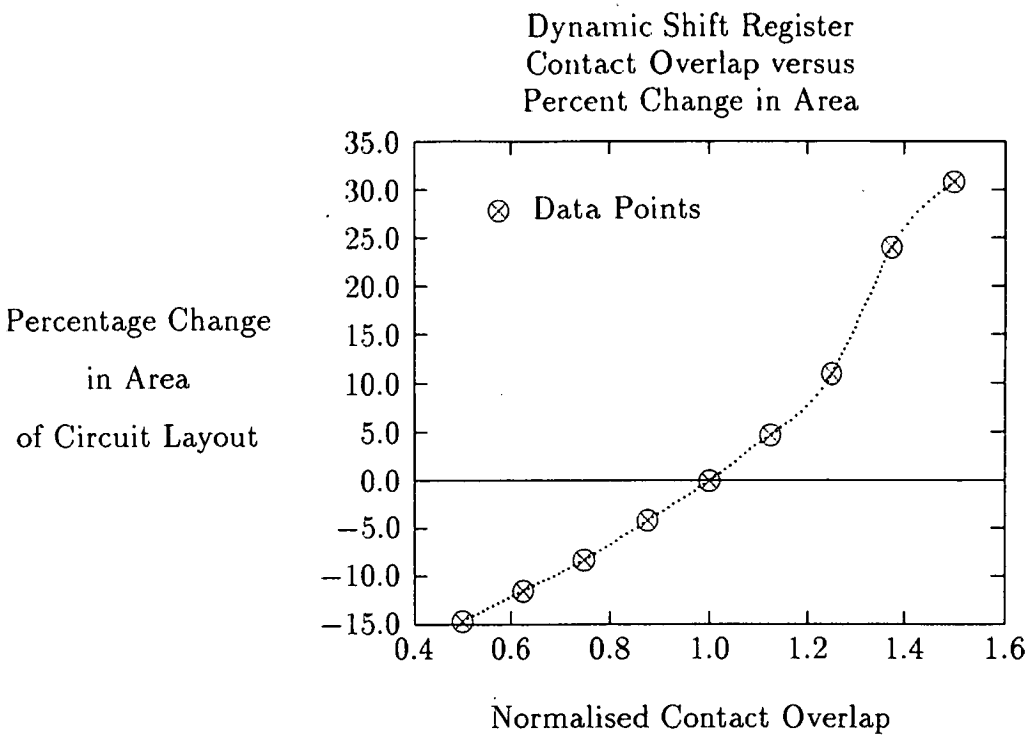


Figure 4–13: Results from Changing Contact Overlap:Shift Register Cell

shown in figure 4–14. This graph shows that there is a reduction in area by using up to two metal 2 lines but that further use of metal 2 increases the size of the circuit again. This simple experiment indicates that the yield of a circuit is the responsibility of the circuit designer as much as the process engineer.

4.5 Defect Sensitivity.

IC layout can be generated either manually or automatically from a higher level input. E.g. Layout can be produced from a symbolic representation of the circuit using a circuit compactor[69]. Whatever method is used, a great deal of effort is usually expended to produce as compact a design as possible. That effort is subject to diminishing returns and consequently nearly all layout is sub-optimal in that it occupies more area than the minimum required for that circuit function.

Achieving the minimum layout area is not equivalent to maximising the yield, though it is often viewed as such by circuit designers. It has been reported in the

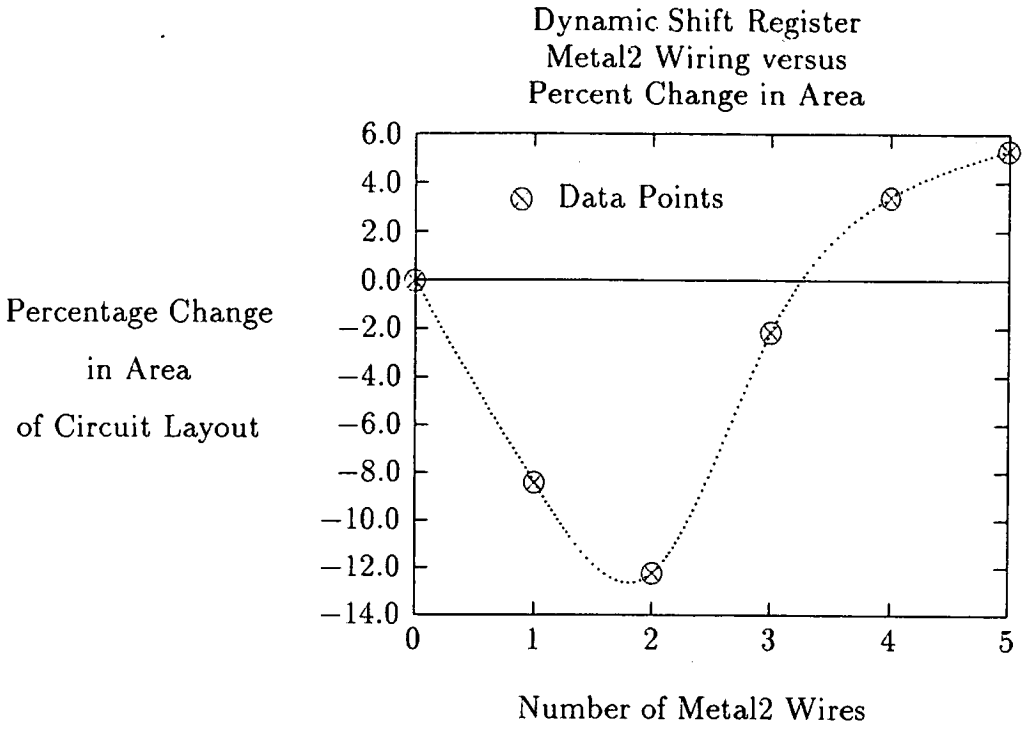


Figure 4–14: Use of Metal 2 in Shift Register Cell.

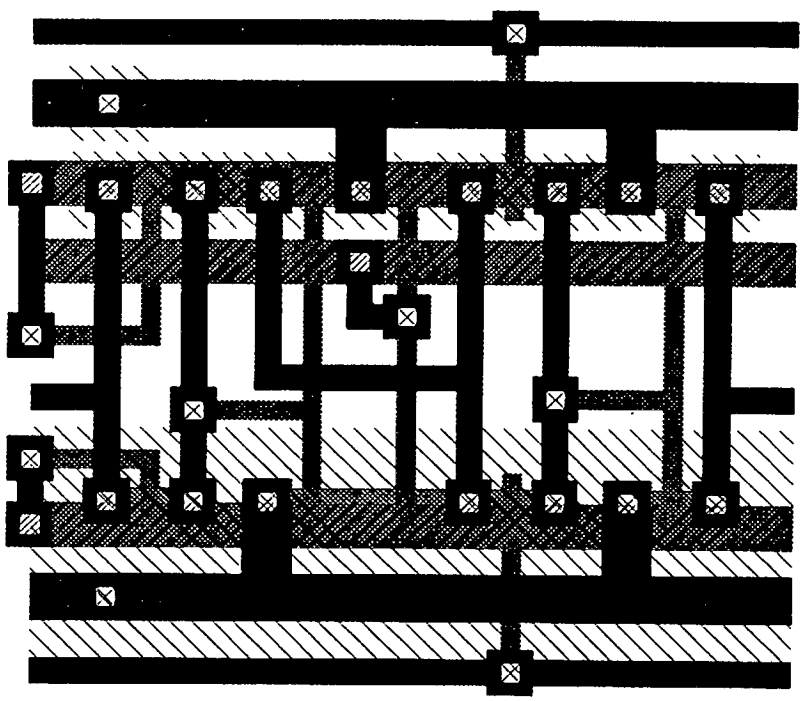


Figure 4–15: Three Metal 2 Wire Layout of Shift Register Cell.

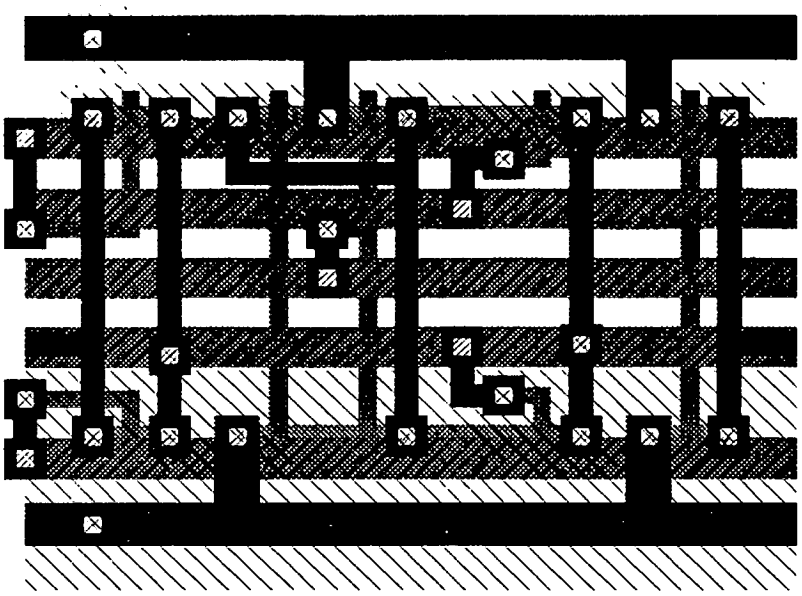


Figure 4–16: Five Metal 2 Wire Layout of Shift Register Cell.

literature [19] that while there appears to be a good relationship between circuit area and yield this is in fact a reflection of the closer relationship between defect sensitivity and yield. This is most easily seen where different chips that have the same area are fabricated resulting in different yields. It follows from this that the goal of the circuit designer should be to optimise the defect sensitivity of the circuit layout. The practice of area minimisation stems from the well known yield equations relating area to yield[3, 29, 19] and perhaps more importantly because of the difficulty of measuring the defect sensitivity of a layout compared to the ease of determining the area.

The minimisation of circuit area is at least a first approximation to the optimum layout, assuming that the global design rules (GDR) are a good reflection of the fabrication process. It is possible however, that this minimum area is not the optimum area. If an increase in circuit area by a given percentage can be used to decrease the defect sensitivity by a greater percentage, the yield of the circuit will be higher for the larger area. Whether this can be achieved for a given design will depend on the circuit design itself and the distribution of the different defect types that determine the defect sensitivity of the layout.

The GDRs are optimised to give "good" layout from a single global set of rules. These rules are not necessarily optimised for the local layout conditions or particular circuit types in which they are used. For example the cost of increasing the transistor overlap in the shift register cell (figure 4–2) by 1.5 of its original value is a 0.1%(figure 4–5) increase in area. If the resultant increase in yield from such a change was greater than 0.1% then the yield would be improved by changing that design rule. However if the shift register cell was only part of a circuit layout on which there was also a RAM cell of the same type as in figure 4–3, that cell would be increased by 3%(figure 4–6). This may be greater than the yield improvement provided by the rule change. For a more optimum layout it would be necessary to apply different rules to different cells.

The GDRs are not optimum for the layout of all circuit types. They also do not give any information on the defect sensitivity of different layout options. This omission causes circuit designers to be unable to determine the best layout to maximise the yield of the circuit.

4.6 Conclusions

This chapter has introduced integrated circuit layout rules. These rules define minimum size and spacings of layout. It has been shown that while, some of these rules can have a negligible effect on the layout area for particular circuit types, the area generally increases as design rule widths and separations are increased. Consequently, there does not appear to be any potential benefit from using layout modifications, involving adjustment of the global design rules of cells, to reduce the circuit defect sensitivity. The cost associated with such a procedure cannot therefore be justified and another approach must be taken.

Chapter 5

Local Design Rules.

This chapter introduces Local Design Rules. These are IC layout rules that define the optimum feature size and spacing in relation to the surrounding geometry and are used to increase the yield of ICs. The yield of IC devices is a major factor in determining their commercial success. This has led to designers of very high volume parts actively designing for yield. At the same time, and sometimes on the same device, other designers trade silicon area for ease of layout. Local design rules are used to reclaim some of this lost area to enhance the yield of the product.

5.1 Local Design Rules.

It is proposed that the yield of circuits fabricated using layout generated from a GDR set can be increased by searching for instances of local non-optimal layout and adjusting the circuit layout to produce a reduction in the defect sensitivity. In practice this done with a set of design rules called the Local Design Rules (LDR) that define a more optimum value of layout geometry size and separation in relation to the local layout conditions. With these rules layout geometry can be increased if sufficiently distant from other geometry and/or can be separated further from neighbouring geometry where there is enough space for a displacement.

The LDRs are applied to a circuit layout that has been generated using a set of GDRs. When these rules are applied there are two further conditions that must be met,

1. No global design rules are violated.

2. There is no increase in the overall circuit area.

The first condition permits automatic design rule checking to be done both before and after the application of the LDRs to ensure that no "accidental" changes have been made that violate the normal design rules. The second condition is to ensure that minimal extra work is required when converting a large design. If the LDRs are applied to individual cells it is important that their area is not changed as this would destroy existing hierarchical designs based on these cells. The coordinates of all the cells would have to be changed to account for the area change. Where LDRs are applied to a whole chip, it is equally important to have the resulting layout occupying the same area, since chips are usually a standard size.

The LDRs do not contradict the GDRs. The GDRs determine minimum size and spacing and hold over the whole design, but they should not be used to determine the maximum feature size or spacing of circuit components by always using the minimum value. While it is valid to attempt an initial layout with minimum sizes to reduce the circuit area, once the area has been defined the best use of any redundant space should be made to reduce the defect sensitivity of the layout.

It is intended that the yield of the resulting layout will be greater than the initial GDR layout. This can only be guaranteed if the fabrication process is understood well enough to ensure that the LDRs are an accurate reflection of the relative yield of the layout options available. No changes are made to the layout except those for which there is evidence to suggest that the changed layout will have a higher yield. This implies that LDRs are process specific in the same way as the GDRs.

5.2 Defining the Local Design Rules.

The complete definition of the LDRs requires a greater knowledge of the process than the GDRs, since the problem is no longer a "simple" matter of finding one rule set to maximise yield. This usually means maximising the yield of regular test structures. To define the LDRs, optimum layout in a variety of situations must be determined. This in practical terms means that it must be known where an increase in track width or contact size will result in an increase in yield, since associated with these adjustments is a reduction in geometry separation. The information required can be found using test structures as is the normal procedure for GDR generation. A full description of the LDRs will require a greater range of test structures, though there may be sufficient information to make a first approximation to the LDRs from the results of test structures designed to determine GDRs.

Yield simulation with defect size and distribution data [70, 71, 20, 46, 42] can also be used to determine LDRs. For example simulation can determine where a change in conductor width will increase the overall yield. The number of faults that are the result of shorts between nodes is increased by widening and hence reducing separation of the conductors, but at the same time the number of faults caused by breaks is reduced. If the overall number of faults can be decreased then there will be a corresponding increase in yield. The optimum value of track width/separation can be found by simulating a range of different layout options. The results can be used to generate LDRs.

5.3 Potential Layout Changes.

There are a number of potential layout changes that can be made; the most important for modern MOS fabrication processes are:

Track displacement (figure 5–1(b)).

Two metal or other layer tracks can be moved further apart from each other, thus reducing the probability of a short forming between the tracks [45, 44].

Increased track width (figure 5–1(c)).

A metal or other layer track can be increased in width. A spot defect on this wider track will be less likely to cause a complete track break [45, 44].

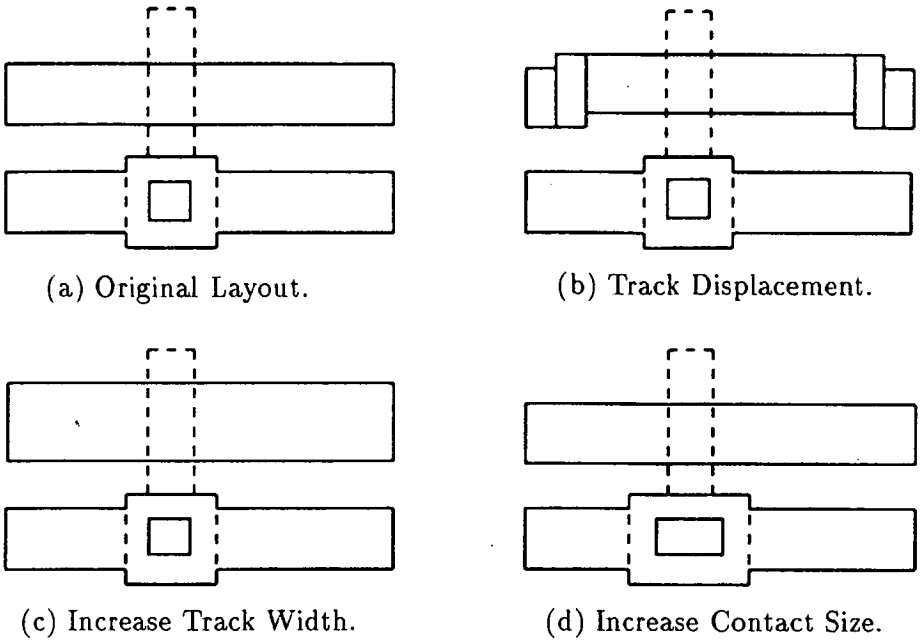
Increased contact size or number of contacts (figure 5–1(d)).

A contact can be increased in size[1], larger contacts have a higher yield [72-74]. The contact overlap must be adjusted to fit the larger contact to avoid violation of the GDRs. It is also possible to generate another contact where there is sufficient space so that if one contact fails the other can still provide a connection.

These changes can give an increased protection against spot or random defects. There are other changes that could be made, (e.g., in order to give protection against misalignment of polysilicon, increases in polysilicon contact overlap and transistor gate overlap could be made). This is less likely to result in an increase in yield, as misalignment usually occurs over the whole circuit or wafer so that, unless every contact and gate overlap were changed, any increase in yield would

---

[1]Larger contacts, particularly those increased in both the X and Y direction, may require an increase in the overlap of the contact beyond that normally required by the GDRs. This would avoid problems that are created with over-etch of large contacts in some processes optimised for the minimum contact and via size.



(a) Original Layout. (b) Track Displacement.

(c) Increase Track Width. (d) Increase Contact Size.

Figure 5–1: Potential Layout Changes to Enhance Yield.

probably be small. A greater variety of changes may be made for immature fabrication processes, where specific process steps have not been perfected. Layout changes can then be used to decrease the defect sensitivity to that process step.

5.3.1 Track Displacement.

The probability of shorts between same layer geometry such as metal tracks can be reduced by displacing the tracks to increase their separation. It can be shown that the optimum separation is where the track is equidistant from the surrounding tracks. Figure 5–2 shows a simple layout that was simulated using the Monte Carlo Yield Simulator of Chapter 3. The position of the metal track was varied by increasing the separation. The results of this simulation are shown in figure 5–3. It can be seen that the defect sensitivity is at a minimum where the track is at the midpoint between the surrounding geometry.

Local design rules for track displacement make use of this result. Tracks

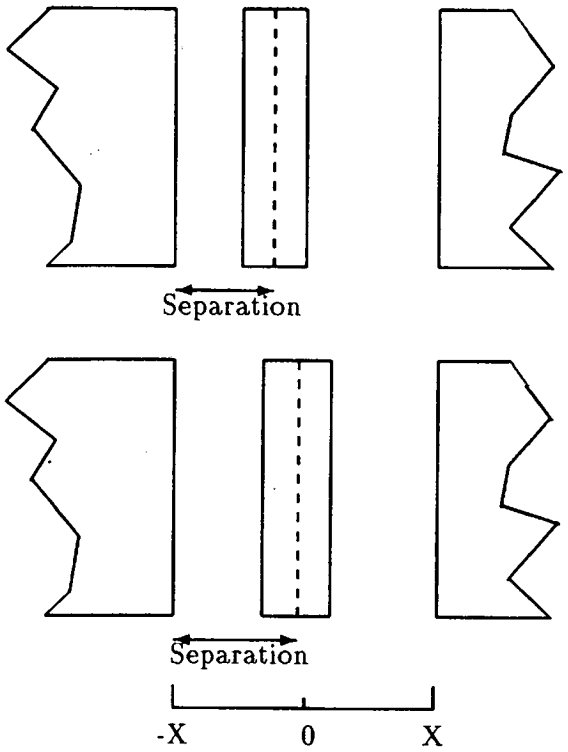


Figure 5–2: Test for Optimum Separation of Geometry.

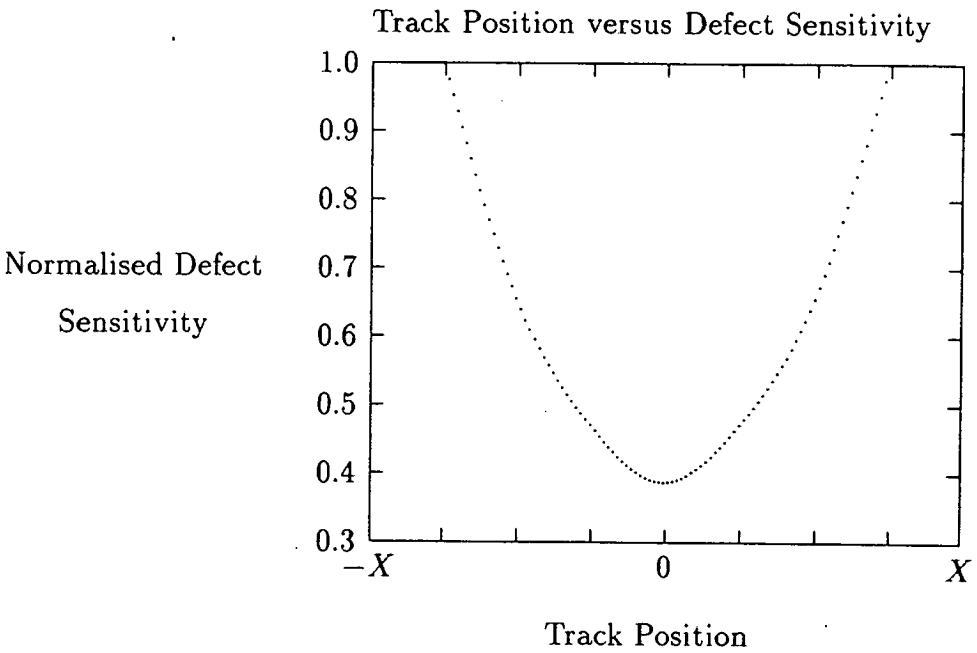


Figure 5–3: Optimum Geometry Separation Results.

are displaced away from their close[2] neighbouring same layer geometry until no further displacement is possible or they are equidistant from the surrounding same layer geometry. Tracks are only displaced away from geometry that is of the same layer and does not form the same electrical node as the track. Where geometry forms the same node a defect would not cause a short since the geometry is already electrically connected.

5.3.2 Change in Fault Size Distribution

The results of simulation of the layout in figure 5–4 are given in the graph of figure 5–5. This graph shows the number of faults caused by different sized defects before and after the displacement of the track in figure 5–4. The difference between the curves can be more easily seen in figure 5–6 along with another curve showing the results of a smaller displacement half the size of the original. These curves represent the improvement in defect sensitivity of the new layout after the track displacement. It can be seen that the number of faults that are caused by smaller(2.5–3×minimum feature size) defects is reduced and that there is a corresponding increase in the intermediate sized faults.

The reduction in the number of these smaller faults is a direct result of the increased separation of the track from its closest neighbouring geometry. However as the displaced track is moved away from the nearer track it also moves towards geometry on the opposite side. This results in an increase in the number of intermediate sized faults that were previously defects which were unable to bridge the gap. Faults produced by even larger defects are largely unaffected by the track displacement.

---

[2]The definition of *close* is dependent on the defect size distribution for the process used to fabricate the circuit. This would normally be 2 or 3 times the minimum layer separation.

Where track displacement LDRs are applied to a circuit there will in general be a range of both displacement amounts and length of displaced track. The resulting change in the distribution of fault size is effectively a weighted sum of curves similar to those in figure 5–6. Such a curve is shown in figure 5–7.

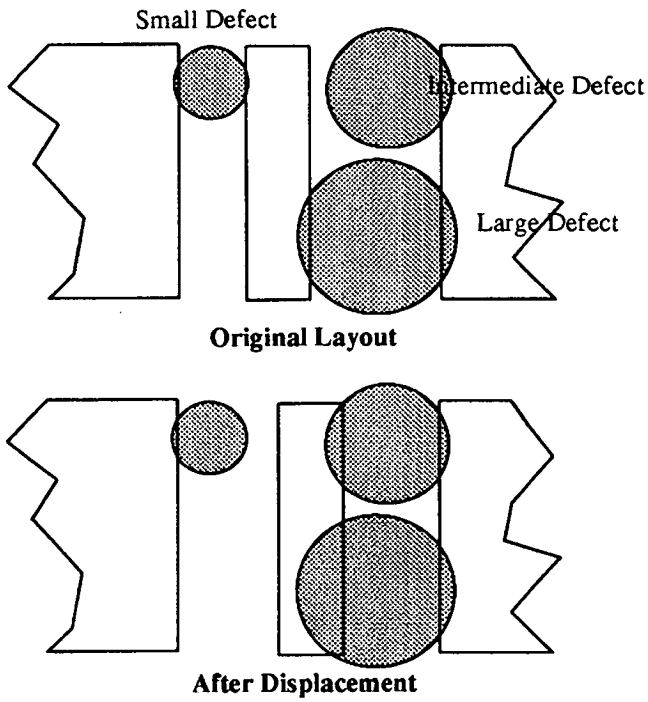


Figure 5–4: Effect of Displacement on Fault Size Distribution

The actual change of fault size distribution in a fabrication process will depend on the defect size distribution. If we assume a defect distribution of $\frac{1}{Defect\ Size^3}$ [4] and combine this with the difference in fault occurrence after track displacement (figure 5–6) the curves of figure 5–8 are generated. Figure 5–8 gives the difference in fault size distribution as a result of a track displacement. The area under this curve is proportional to the change in yield, a negative area is a yield improvement.

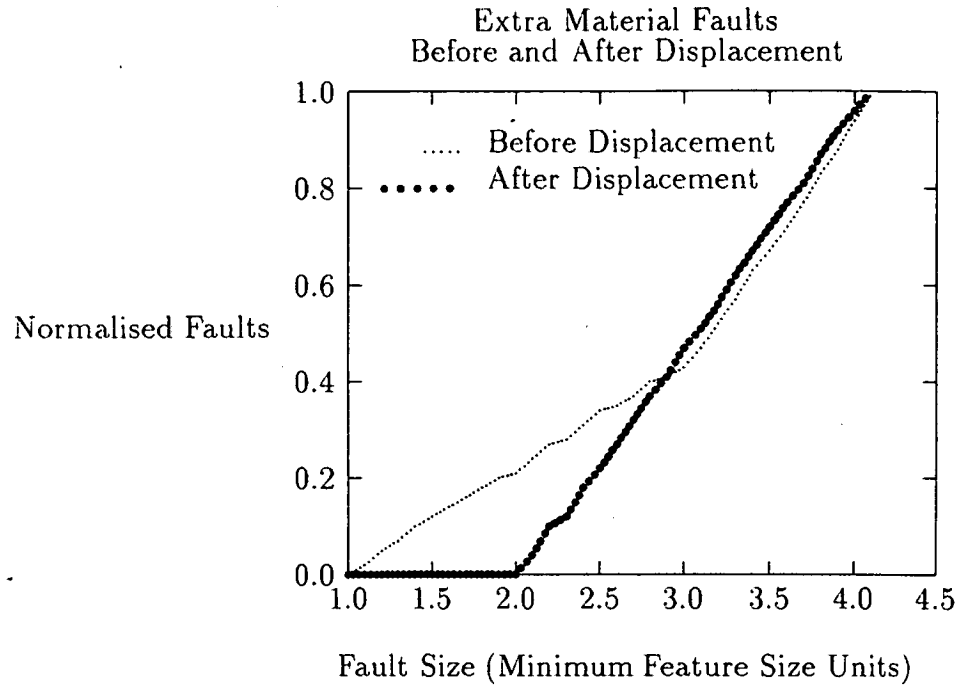


Figure 5–5: Distribution of Faults Sizes Before and After Displacement.

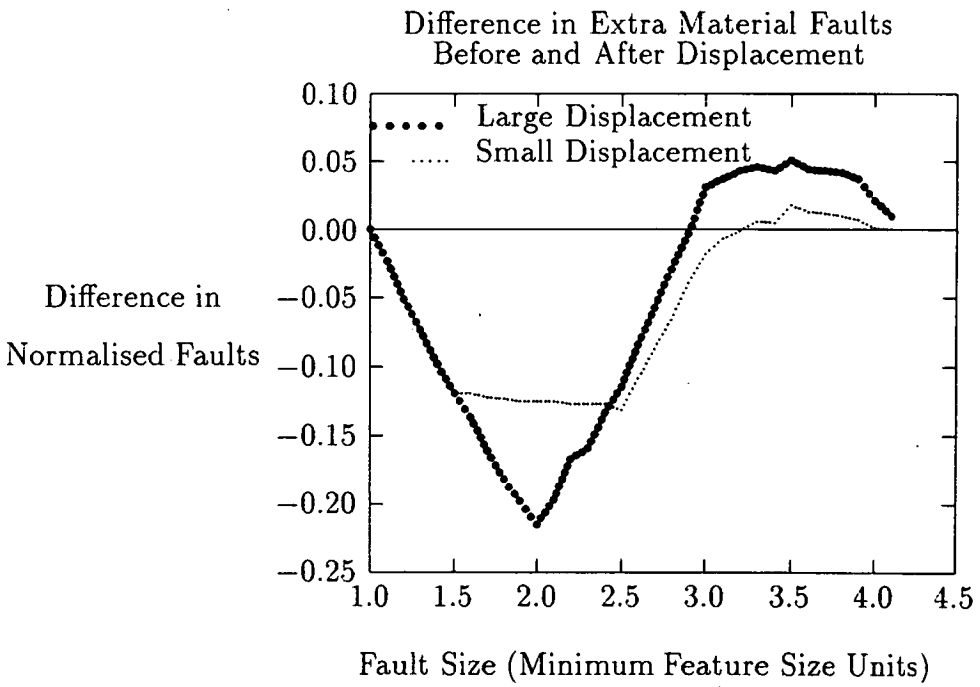


Figure 5–6: Difference Between Distribution of Faults Sizes Before and After Displacement.

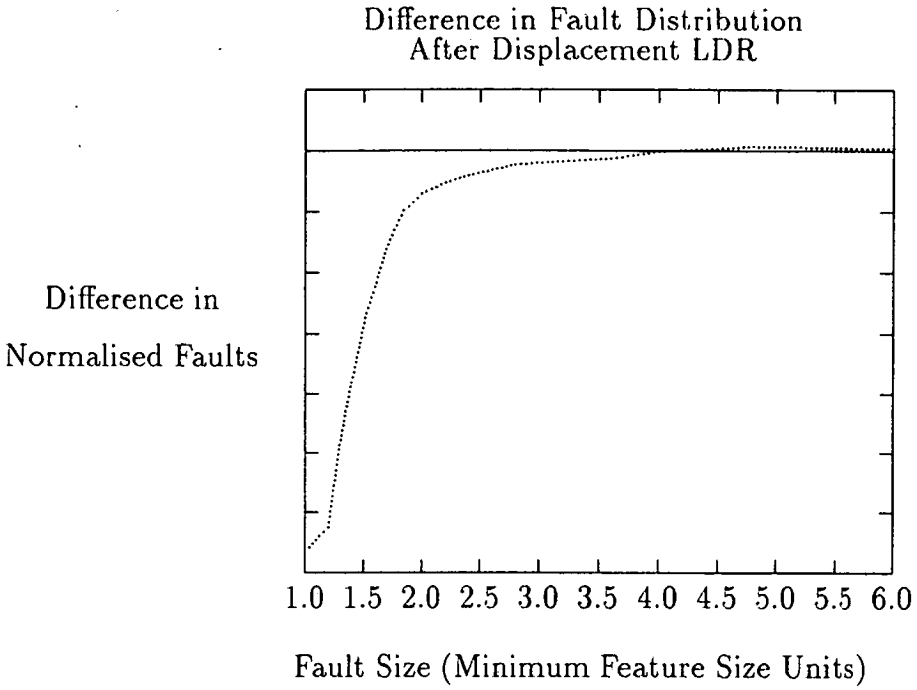


Figure 5–7: Standard Change in Fault Distribution from Track Displacement LDR Application to Circuit Layout.

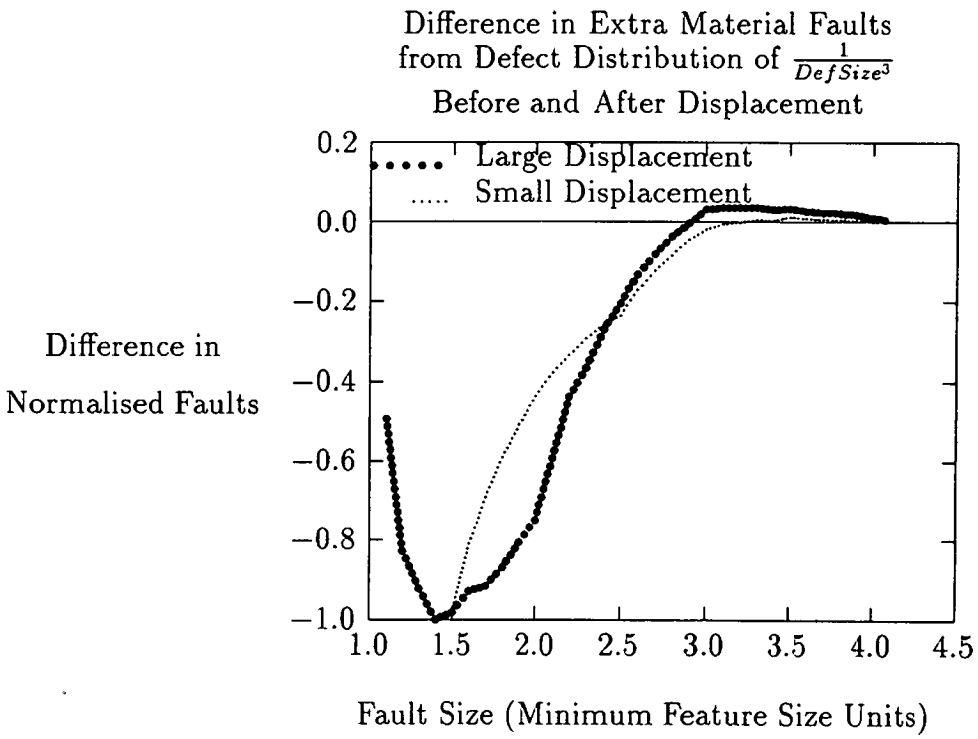


Figure 5–8: Difference in Fault Size Distribution After Displacement of Track.

5.3.3 Displacement Strategies

The are two possible displacement strategies,

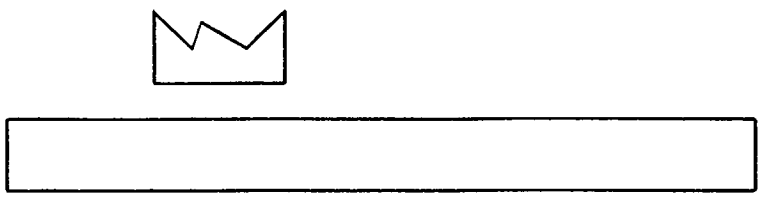
- **Local Displacement**

  Local Displacement involves displacing the track only where neighbouring geometry is present, the track then returns to its original position. This is shown in figure 5–9(b).

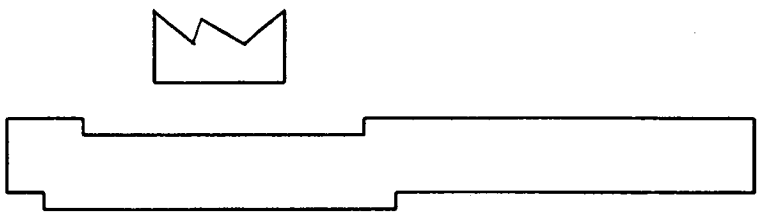
- **Continued Displacement**

  In continued displacement the track is displaced where neighbouring geometry is present but does not return to the original track position after the influence of the neighbouring geometry is "behind" it. The track is only guaranteed to return to its original position at the track end. This is shown in figure 5–9(c).

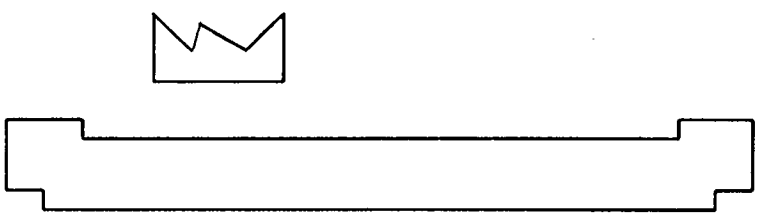
The best displacement method to use is the one that gives the greatest increase in yield and the least change in performance. Both methods can potentially give the same increase in yield since they result in track that is equidistant from neighbouring geometry, where that geometry is sufficiently close to have a significant effect on the probability of a fault being generated by a defect. Therefore the best method to use is that which results in the shortest track length; this is equivalent to the fewest displacements. The continued displacement method will result in the same or fewer displacements than the local displacement method. A track undergoing local displacement always returns to the original path at the earliest opportunity and then further along the track may be displaced again in the same direction as before. Continued displacements result only in displacements that actually improve the yield. Though in many cases this will result in exactly the same displacements using either methods. The optimum displacement strategy is continued displacement. This method cannot be applied easily in circuits that have been split into a number of blocks for automatic application of LDRs.



(a) Original Layout



(b) Local Displacement



(c) Continued Displacement

Figure 5–9: Possible Displacement Strategies

Where a circuit is split into smaller blocks to be processed (Chapter 5) the geometry at the boundaries must match. If the position of any geometry was not dependent on some local conditions that could be easily determined by the neighbouring block there would be great difficulty in correctly forming the circuit from the blocks. Continuous displacement of track results in changes in track position that may have their origin far from the point on the track they affect. Consequently, at present only the local displacement strategy is implemented[3].

5.3.4 Increase Track Width.

The probability of a track break can be reduced by increasing the width of the track. This is shown from a simulation of the layout of figure 5–10. The width of the metal track is increased with a resulting decrease in the sensitivity to track breaks shown in figure 5–11. When the width of the track is increased there is an equal reduction in the separation between the track and the neighbouring geometry. This causes an increase in the probability of a short being formed by an extra material defect between track and same layer, separate node, geometry. This is shown in figure 5–11.

In order to define local design rules to be used to increase track width the defect size distribution of both extra and missing material defects of the adjusted layer must be known. In general the LDR for track width increases is defined such that the track width is increased until any further increase would result in the increased probability of a short being greater than the reduction in probability of a break. For example, if the distribution of missing and extra material defects were identical, the track width could be increased until the separation and width

---

[3]One way of using continued displacement within a circuit split into blocks is to permit movement where tracks do not intersect the boundary and therefore have no matching tracks in a neighbouring block. All tracks that intersect the boundary can be displaced using the local displacement method.

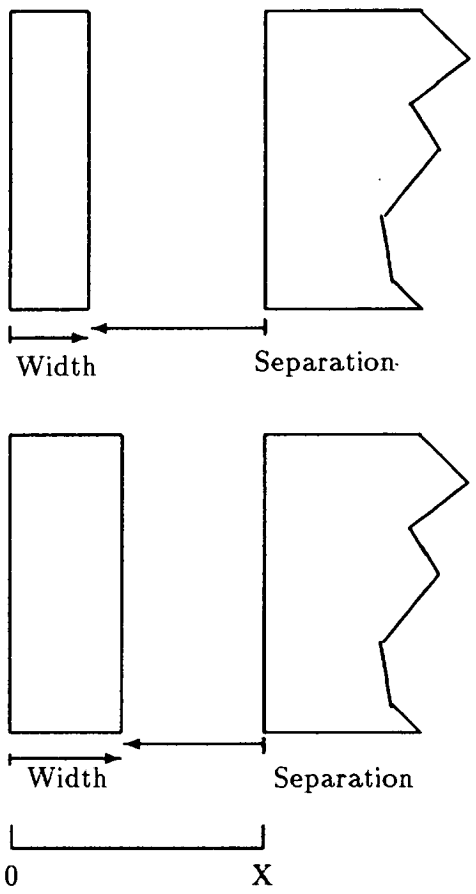


Figure 5–10: Test for Optimum Width/Separation of Geometry.

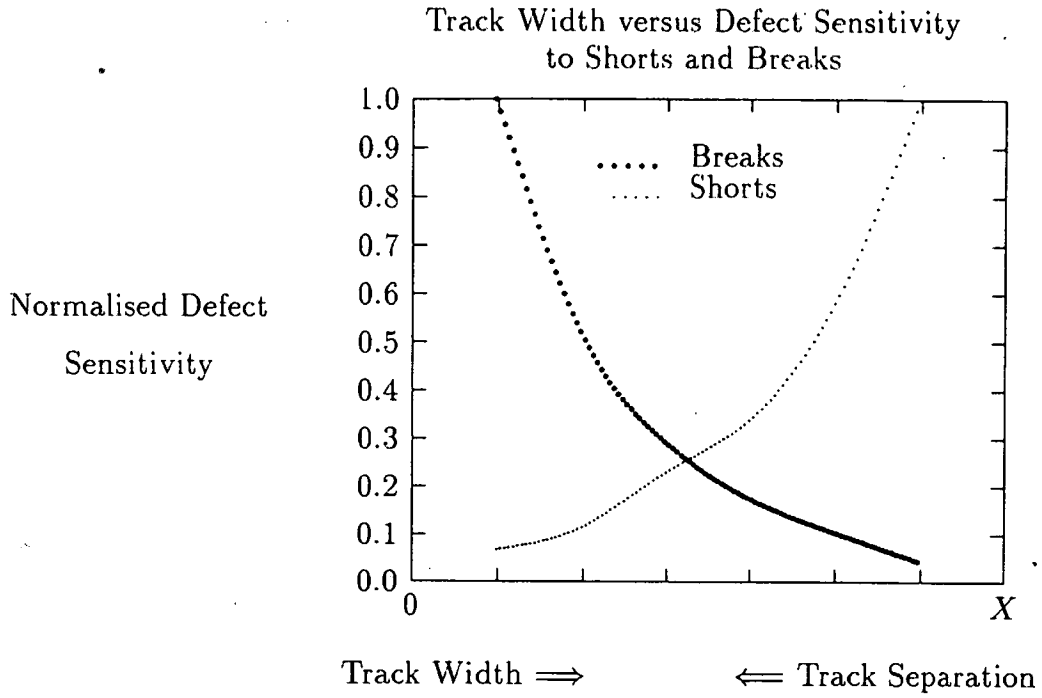


Figure 5–11: Optimum Width/Separation Results.

were equal (figure 5–12). Where there are more extra than missing material defects the width of track should be less than the separation (figure 5–12).

5.3.5 Increased Contact Size

The probability of a badly formed contact being either completely open or with a high resistance can be reduced by increasing the contact size[72-74]. Determining whether a given contact should be increased with a corresponding increase in the contact overlap layers is a more difficult task than for the previous cases of increased and displaced track. There are a minimum of three separate layers involved in the formation of a contact. The two connected layers and the contact cut itself. Typical metal-polysilicon and metal-metal2 contacts involve only three layers. Active area-metal contacts can involve five layers or more.

1. Active Area.

2. Metal.

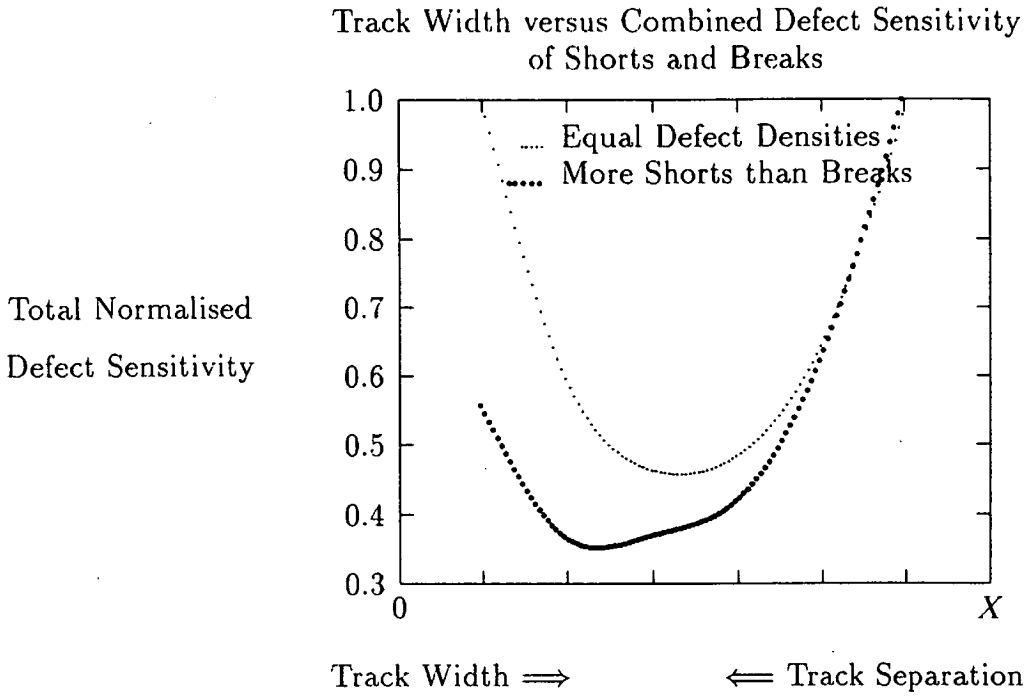


Figure 5–12: Combined Results of Extra and Missing Material Defects for Increased Geometry Width/Separation.

3. Contact Cut.

4. Implant P or N type.

5. Well layer N or P type.

Determining LDRs for structures with a large number of layers is obviously more difficult. The following discussion is confined to contacts with three layers. Consider the layout of figure 5–13, where a three layer contact structure is formed. In this case increasing the contact size results in,

- A decrease in missing material faults for both polysilicon and metal. A small portion of the metal and polysilicon wire are increased in width by the extra overlap.

- An increase in the number of extra material metal and polysilicon faults. This is a result of the corresponding reduction in the metal and polysilicon separation over a small portion of the track

- A decrease in badly formed contacts.

For this layout there is a corresponding increase in both polysilicon and metal extra material defects as there is other polysilicon and metal to interact with. This is not always the case; for example figure 5–14 shows a layout where there is no polysilicon with which a short can be formed. Even if this layout was only a part of a larger layout in which there was other polysilicon geometry the probability of an extra material defect forming a short is very low, because the polysilicon would be separated by a significant distance and the probability of large defects is low, assuming a standard defect size distribution. There are occasions in which there is no other geometry close enough to a contact to be shorted by an extra material defect in which case no fault contribution is obtained from the extra material defects of either contact layer.

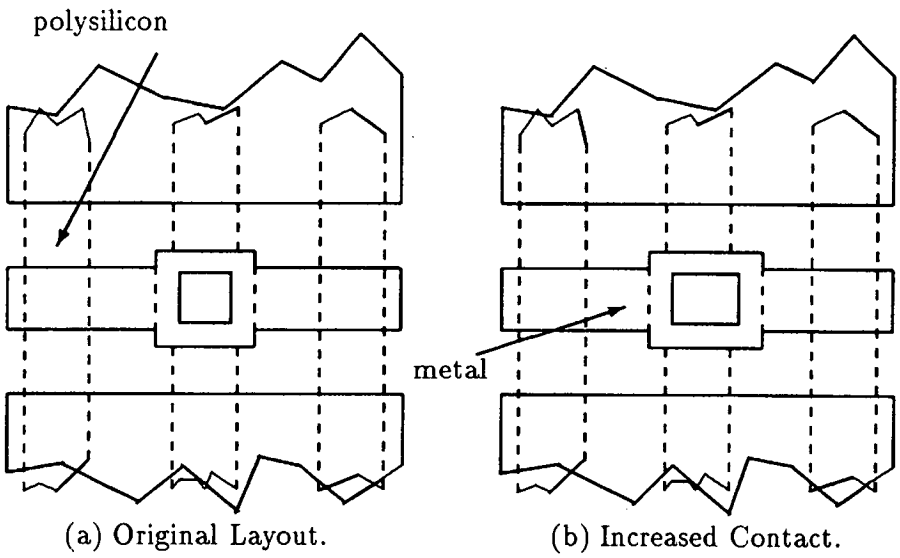
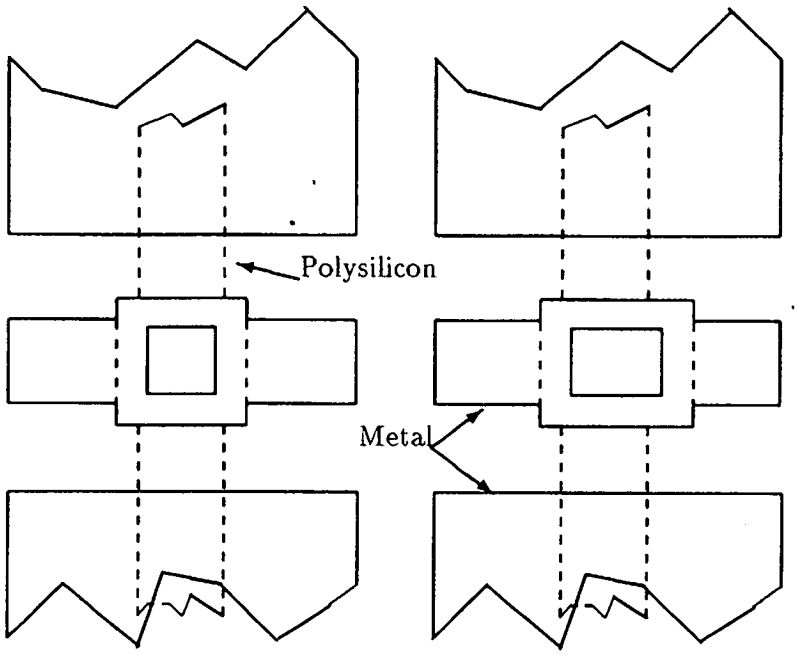


Figure 5–13: Increased Contact Size with Polysilicon Interaction.

In many GDRs it is recommended that a number of small contacts are preferred to one large contact. Once contacts reach an appropriate size they can be split to accommodate this rule.

A contact change is made where the reduction in yield from increased overlap (extra material defects) is less then the increase in the contact yield plus any



(a) Original Layout. (b) Increased Contact.

Figure 5–14: Increased Contact Size without Polysilicon Interaction.

gains made from the increased overlap (missing material defects). Because of the complexity of the LDRs for contacts, a large margin of error will be required. In practice it is recommended that changes to contacts are only permitted where there is a particularly bad problem with contact yield. This situation can arise and was described by Lorenzetti in the presentation of his paper [48] at Grenoble. He described a 1.2 million transistor chip designed by Northern Telecom that gave 0% yield. Lorenzetti examined the layout of the chip with the McYield yield simulator and found that more than 50% of the yield loss was due to the vias (metal-metal 2 contacts). The layout of the chip was adjusted by hand increasing the size of the vias within the GDRs where there was available space. When the chip was re-fabricated it gave a yield of approximately 1%. While this is not a particularly large yield it is very much better than nothing.

5.4 Performance

It is important that any changes that result from the application of LDRs to a layout do not cause a degradation in performance of the circuit. For this reason no changes that affect the size of active devices should be made. Where timing paths are critical care must be taken with layout adjustments so as to avoid the critical path. For example fan out from a single inverter may require that the signal reaches the inputs to the next gates simultaneously. The interconnect forming these time critical paths will either require special treatment, or be left unchanged.

The performance of circuits is such an important consideration that great care must be taken to ensure that any performance change does not push a circuit or circuit part outside of the allowable specifications. Where performance is critical, or it is suspected that layout changes may have adversely affected the circuit performance, circuit simulation using software such as SPICE[75] can be used to determine the extent of performance change.

In a more automated digital circuit layout environment, such as IDA (Integrated Design Aids) [76], transistor sizing software (e.g. TILOS[76, 77] – TImed LOgic Synthesizer) can be used to optimise the circuit performance and later be used to check any LDR adjustments. That is, if, after LDRs have been applied to the layout, the TILOS program suggests that the transitor sizes are still optimum the layout can be accepted. However, if transistor re-sizing is required then the performance of the circuit has been degraded and the layout should be modified, by re-sizing transistors or un-doing the LDR modifications, to allow the circuit to meet it's required specifications.

5.4.1 Increased Contact Size.

Increased contact/via size can result in a decreased contact resistance and increased overlap capacitance. The change in resistance as a result of increasing

contact size is dependent on the contact type. In particular, the contact resistance is dependent on the relative resistance of the two layers connected by the contact. Where the resistance of the layers is approximately equal, as in a metal 1 - metal 2 contact the resistance is proportional to the contact area.

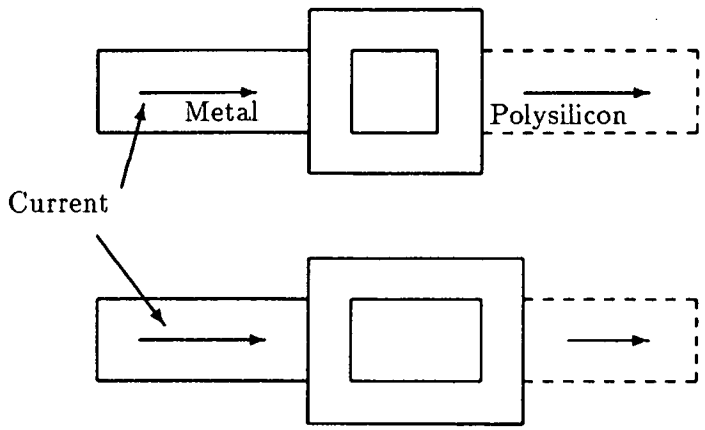
In contacts that connect layers with dissimilar resistances the change in contact resistance is dependent on the geometry of current flow and the contact size[37]. In particular the resistance is dependent on the width of the contact perpendicular to the current flow. This implies that an increased contact size might not lead to a reduced resistance. For example figure 5–15(a) shows a polysilicon contact in which the increased in size does not affect the contact resistance, as the contact edges perpendicular to current flow remain unchanged. Figure 5–15(b) shows a contact that has a reduced resistance as a result of a size increase.

The effect of overlap capacitance can be neglected because it is effectively shorted by the contact. The RC product will therefore be reduced in line with any reduction in the total contact resistance.

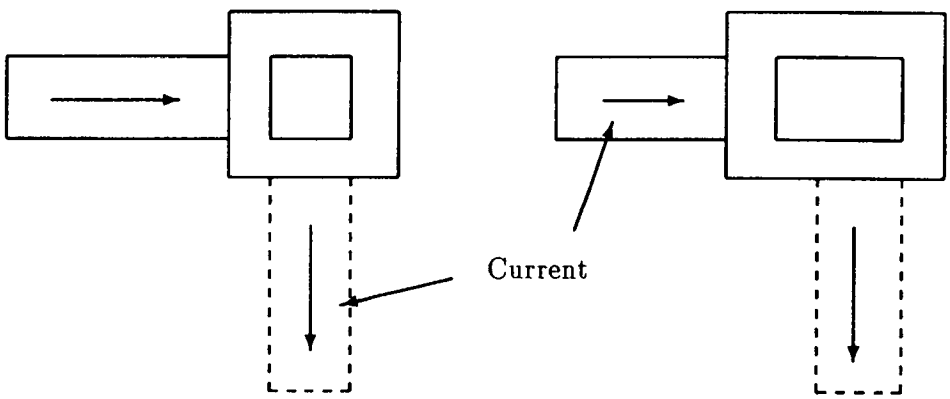
Increased substrate contact size can take place where the contact is sufficiently distant from active devices. Since they are not used to pass signals there will be no effect on performance of the circuit.

5.4.2 Track Displacement.

The displacement of a track results in a longer track, with an increase in RC. This is due to the small step in the track used to implement the displacement (figure 5–16). This will not necessarily cause a problem for metal tracks since the increase is small and, except for very long lines, the main resistive components in interconnect are contacts to other layers. Any reduction in performance will be very small and will only be apparent if the critical path is affected. If the critical path is known then any performance reduction can be avoided by ensuring that no large changes in track length are made on this path.



(a) Contact Resistance Unchanged.



(b) Contact Resistance Reduced.

Figure 5–15: Resistance and Contact Size in a Metal–Polysilicon Contact.

For the more resistive polysilicon tracks greater care should be taken. It may be best to avoid changes altogether unless significant yield improvements can be made.

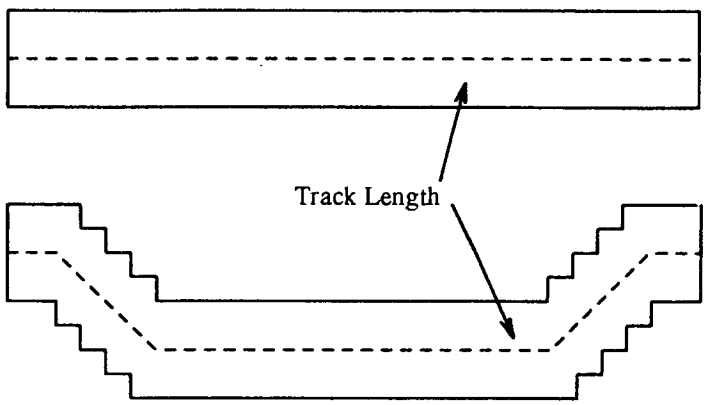
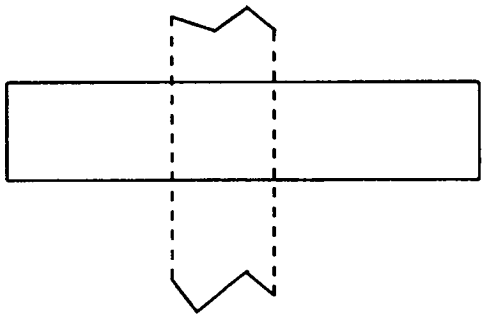


Figure 5–16: Increased Track Length with Displacement.

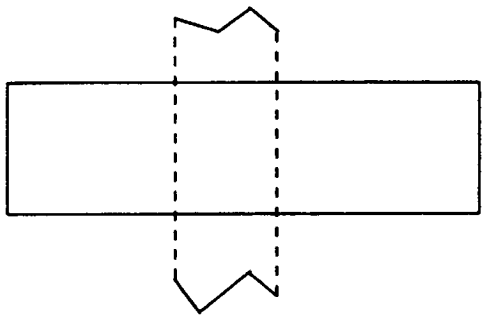
5.4.3 Increased Track Width.

Increased track width will cause an increase in the capacitance of the circuit but will also decrease the resistance of the circuit. The RC product will be reduced for small geometry tracks because the fringe capacitance, which is a sizable fraction of the total, is not significantly increased. Capacitance between tracks on the same conductor level will not be larger than that allowed for in the GDRs since the minimum track separation is still determined by the GDRs. However, the RC of the circuit as a whole will be increased since the RC of the transistors driving the node formed by the now wider track is increased since the capacitance of the track has increased and there is no change in the output impedance of the transistors.

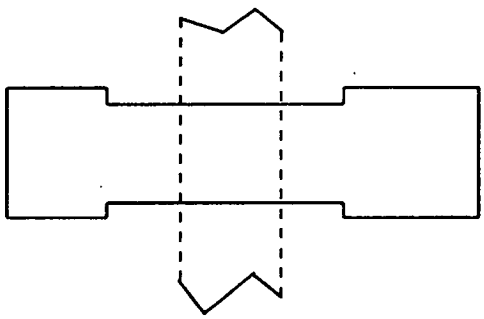
There is a possibility of greater cross talk where the capacitance between crossovers is increased. To avoid this, overlapping geometry that does not form a single electrical node should not be altered (figure 5–17).



(a) Original Layout.



(b) Increased Crosstalk.



(c) No Increased Crosstalk.

Figure 5–17: Track Width Increase and Cross Talk

5.5 Reliability

The reliability of a component is also of fundamental importance. There is little point in increasing yield at the price of the component reliability. The reliability of the circuits optimised using LDRs can also be improved. This requires that when the LDRs are calculated reliability issues are taken into account so that no layout changes will be made that cause reliability to be reduced.

There are occasions when high reliability is of greater importance than yield and LDR sets that increase reliability at the expense of yield can be developed. For example the width of interconnect can be increased to reduce the possibility of electromigration. Track widths that are larger than optimum for yield enhancement may be used to improve the reliability. Care should be taken since reduced separation can also increase the probability of failure from electromigration.

5.6 Suitable Fabrication Processes.

The effectiveness of LDRs is highly process and to some degree circuit dependent. They are best suited for fabrication processes that have,

- yield loss from shorts between tracks,
- yield loss due to poor contacts,
- yield loss from metal breaks.

While it is difficult to generalise about fabrication processes, there are some features common to many, though there are of course exceptions. From the fault categories that can be affected by LDRs, spot or random defects, one of the main yield detractors is the metallisation. Metallisation is prone to defects because

the wafer surface is no longer locally flat. The is due to localised deposition of preceding layers and covering oxide growths.

To avoid problems from electromigration, track widths are often wider than required for optimal yield. If the reliability of the track was not a concern the width could be reduced with a resulting increase in yield. One consequence of this for modern MOS processes is that the number of faults as a result of extra material defects exceeds the number caused by missing material defects. Sometimes this is by a very large amount [70]. This implies that track displacement LDRs are likely to be more important than track width LDRs.

The yield of contacts is often a problem in immature processes, but as the process matures they should become less of a problem. LDRs can be used to increase the yield of a fabrication process throughout its life. In the early life of a process, redundant space might be used primarily to increase contact size and, as the process matures and is perfected, the emphasis may change to track displacement LDRs. Contact size and track displacement LDRs are not mutually exclusive and in many instances can both be used fully in the same layout.

5.6.1 Volume Production.

While LDRs can be used regardless of the number of circuits to be fabricated, they are most efficiently used where there is a large volume of production. One reason for this is that the process of applying LDRs has a cost associated with it. The operation requires computer and operator time. Another reason is that where a small set number of chips are produced for a customer there is often over production to ensure that there are the required number of working devices. In this case where there is already a significant number of wasted devices there is little point in increasing the yield.

Where there is large volume production of a part that requires a large number of batches to complete the order, or a chip that is in continuous production such as a CPU or any device that has a part number, LDRs can make a useful contribution

to the yield. The point at which LDRs become useful will depend on the yield gained from their application and the number of devices required.

5.7 Suitable Circuits for LDRs.

In principal any circuit can be improved by the application of LDRs apart from those with no redundant space. Few circuits fall into this category except perhaps RAM and ROM designs and a few other very regular structure where a lot of effort has been applied to maximise yield. In practice some circuits are more suitable than others. LDRs are best suited for circuits that are,

- **Digital in nature.**

  Analog circuits are very dependent on the resistance and capacitance of their component parts and occasionally require that a number of paths have identical RC components. The circuit function would have to be well understood in order to apply LDRs with confidence. Digital circuits can tolerate a much greater degree of parameter variation and are therefore more suitable. To a large extent LDRs can be applied to a digital circuit layout with little regard for the circuit function. The only exception is on the critical path of a circuit that is already uncomfortably close to its required operating frequency.

- **Irregular Layout.**

  Irregular layout results in redundant space and can be the result of poor layout design or the circuit function. Some circuit functions can be laid out in a neat and regular fashion e.g., a 6 transistor RAM cell (figure 5–18). Other functions do not lend themselves so easily to neat layout e.g. XNOR gate (figure 5–19). Circuit functions on a larger scale show a similar variability in the use of available space. Where there is space available LDRs can be used.

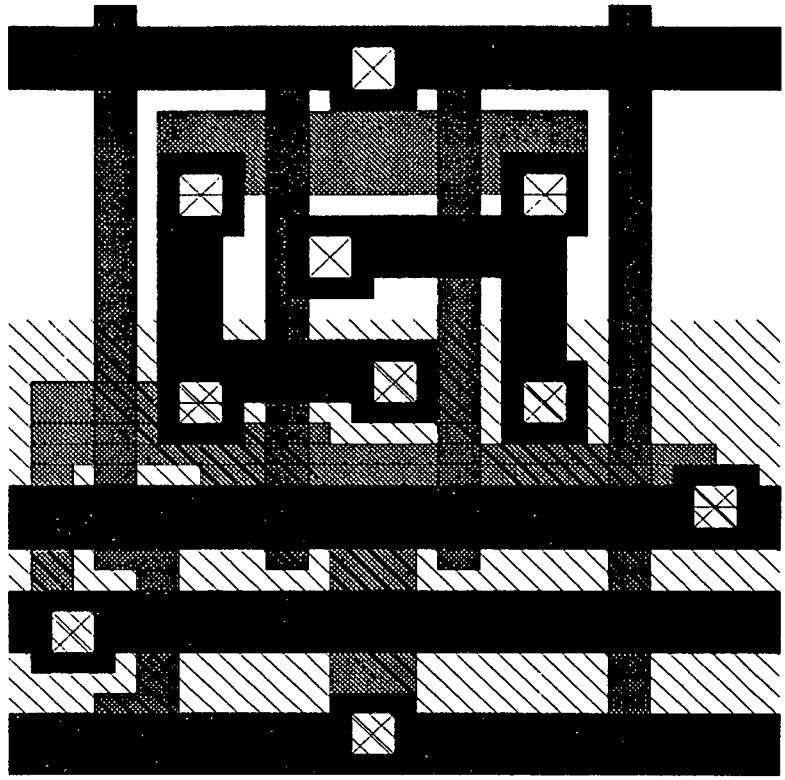


Figure 5–18: RAM Cell with Little Scope for LDRs.

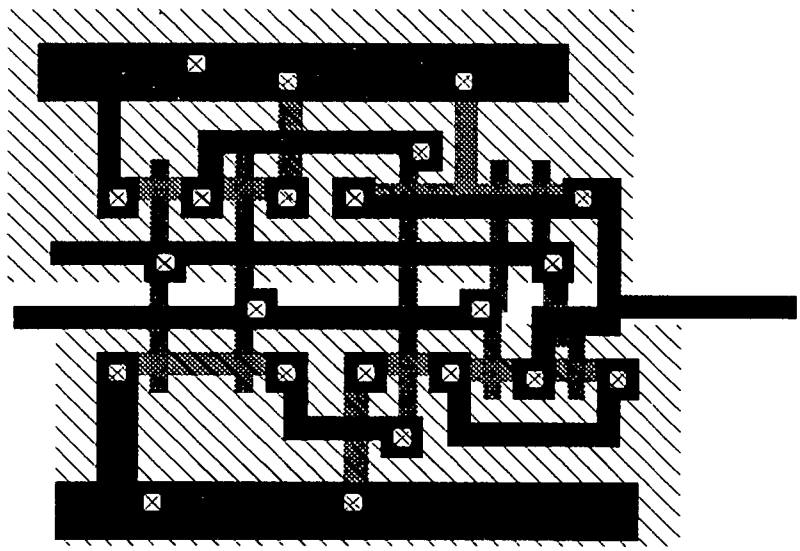


Figure 5–19: 2 Input XNOR Gate with Scope for LDRs.

- **Interconnect Intensive.**

  In a two metal MOS process, interconnect consists of metal, metal2 and metal-metal2 contacts. Yield loss from the metallisation stages is often a main contributor to total yield loss[78]. Interconnect is often irregular (figure 5–20) allowing track displacement LDRs to be used. There is also usually sufficient space to permit increased contact LDRs to be applied at least along the direction on the track. Circuit layouts with a lot of interconnect, particularly where it is irregular, can be improved with LDRs.

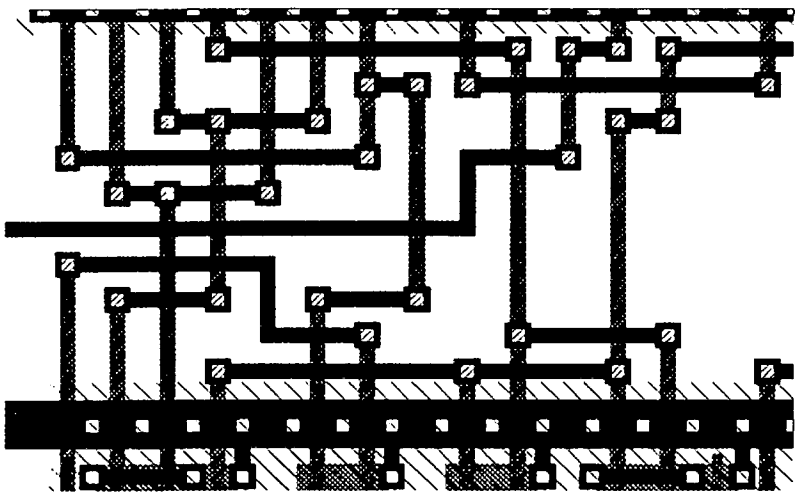


Figure 5–20: Example of Interconnect with Scope for LDRs.

- **Automated Layout.**

  LDRs are of most use where there is incomplete use of the layout area. Automated layout systems often result in inefficient layout, less dense than comparable hand-crafted layout. Even automated layout systems that generate layout by routing together hand-crafted cells such as GENESIL[79] can result in poorer layout than a full custom design. This is because the cells have been hand crafted for a generic CMOS process and then manipulated by scaling and over/under-sizing individual layers to produce layout to meet the intended fabrication processes design rules. GENESIL uses a program GENECAL[80] to accomplish this task as efficiently as possible.

As a consequence of the poorer use of layout area, automatically generated circuit layout is well suited to the application of LDRs.

5.8 Applying Local Design Rules.

Applying LDRs by hand would be very time consuming and also prone to error. To address this problem a program, LocDes, acting as a post-processor of CIF format layout has been developed. The LocDes program takes a circuit layout generated from a global design rule set as input to produce as output an enhanced layout with a reduced defect sensitivity but having the same function and performance specifications.

5.9 Conclusions

This chapter has introduced and developed the concept of local design rules. These rules describe the conditions under which a specified layout modification can be made. These conditions would normally refer to the position of neighbouring circuit geometry and, in particular, the required separation of geometry before a layout modification can be undertaken. The purpose of these rules is to enable layout modification to be described in a manner that can be used in a computer program to automatically apply these modifications.

The use of LDRs may have an effect on the performance and reliability of the circuit to which they are applied. Because of this, and the costs associated with applying LDRs, it is recommended that layout modifications are only considered for digital circuit layouts that are destined for large volume production. Ideally this production will take place in in a fabrication process that in which the yield loss is due to one or more of the process steps for which the defect sensitivity has been reduced by the LDRs.

Chapter 6

The LocDes Program.

This chapter describes the implementation of LocDes. This program applies local design rules to circuit layout to produce an enhanced layout that has a reduced defect sensitivity. The LDRs can either be applied to individual pieces of circuit geometry or iteratively over the whole circuit layout. The algorithms used to apply track width, track displacement and contact local design rules are presented.

6.1 The LocDes Program

The purpose of the LocDes program is to automatically apply local design rules to IC layout. These LDRs are used to described to the program the particular layout conditions that should, where possible, be modified and the precise modification that is to be attempted.

6.1.1 Local Design Rules

Local design rules consist of a list of one or more conditions that have to be satisfied before a layout change should be attempted. They also include, the layout change itself and another list of one or more conditions that the new layout must meet. This last list will always include the global design rules.

For example, a track displacement LDR operates on track segments. The segment is tested to determine whether it is equally distant from the neighbouring geometry on either side of it. If the inequality is greater than a value specified by the LDR, a displacement can be attempted. The size of the displacement is specified by the LDR. The layout modification is provisionally made and tested to see if it meets all of the conditions set by the LDR.

Since the GDR rules must always be satisfied, any other conditions required by an LDR will normally be more rigourous than those already included in the GDR set. For example, in a track width increase LDR, an extra condition could be that the separation of modified track should be greater than that required by the GDRs. This could be to help ensure that shorts between tracks are not increased by more than the reduction in track breaks.

6.1.2 Design Rule Checking

To simplify the problem Manhattan layout was assumed. The basic principles are not affected by this restriction and the implementation is greatly eased.

The application of LDRs requires a considerable amount of design rule checking of both the local and global design rules. The design rule checking that is undertaken by the LocDes program is the most computationally expensive part of the the program. Therefore, the internal representation of data and the algorithms used by the program are designed to implement design rule checking efficiently.

Normally design rule checking is done using line segments and an algorithm based on the scan-line technique[81, 76, 82, 83]. However the design rule checking undertaken by LocDes is different from the usual design rule checking in a number of ways.

1. Not all the layout geometry needs to be checked. Only where layout has been changed is it necessary to design rule check that particular piece of

layout. All other layout is assumed correct, since it is either original layout, that has previously been processed by a design rule checking program, or modified layout that has already undergone a check from LocDes.

2. Only small sections of layout need to be checked. The layout to be checked is either a small track segment[1], a contact or a contact overlap. All of which are small relative to the total layout size.

3. A single rule violation terminates the design rule checking phase. Since only those layout changes that cause no design rule violations are acceptable there is no need to find further violations when a broken design rule is found.

4. A large number of very similar but separate design rule checks are required. For example a track displacement will require design rule checks on a large number of track segments, each of which is only in a slightly different position from the last (since they form the same track).

These differences make it is more efficient to have a design rule checking algorithm with a low initialisation overhead before each individual "run" of the check procedure. However, a large overhead is acceptable where it occurs only once before any of the "runs".

Algorithms based on the scan-line technique require an initial sort of line segments and further sorting during processing both of which are computationally expensive. While, it is possible to minimise the amount of sorting in similar "runs", the initialisation time is still high.

A more efficient algorithm for the particular requirements of the LocDes program is based on layout that is represented as a collection of rectangular boxes

---

[1]Both track displacement and track width increase LDRs split tracks into small segments before they are processed

and a design rule checking algorithm that uses region queries rather than the scan-line technique. The data structure and algorithms used to implement the rule checking are presented in chapter 7.

6.1.3 Implementation

The LocDes program was originally written in C but was converted to C++. The main body of the code is still effectively C code but the C++ compiler used, g++[2] gave much better error checking which resulted in the discovery of a couple of previously undetected coding errors.

---

[2]G++(Ver. 1.37.0) is the C++ compiler produced by the GNU Foundation released under their copyright conditions known as the Copyleft. In short these conditions state that source code should be available for any distributed program containing any of their code.

6.2 Operating Modes

The program can be operated in two different modes,

1. With a graphics user interface, where the user defines individual pieces of layout geometry to which selected LDRs are applied.

2. In batch mode where the program applies LDRs in a predefined order evenly over the whole design space.

6.2.1 X Window User Interface

The LocDes program has been provided with an X window user interface[3] (figure 6–1). The interface was built using the ET++[84], an object oriented application framework, implemented in C++. This enabled a menu driven interface to be built in a very short time.

The interface allows the user to apply selected LDRs (from pop-up menus) to individual pieces of circuit geometry indicated by the mouse. For example in figure 6–1 the LDRs have been applied to one track(displacement) and a contact(increase). It should be noted that in this case the GDRs have not required an edge separation with the underlying polysilicon for the track displacement.

This type of tool can be used to apply LDRs to the layout of individual cells forming the database of a cell based automated layout tool. The application of LDRs can be controlled to avoid design rule violation across cell boundaries and can ensure that there will be no performance degradation of the resulting cell. This is done by applying LDRs selectively within the cell boundary, keeping all

[3]A single binary suitable for X, NeWs and Sunwindow window environments can be generated using the support for these window systems provided by ET++.

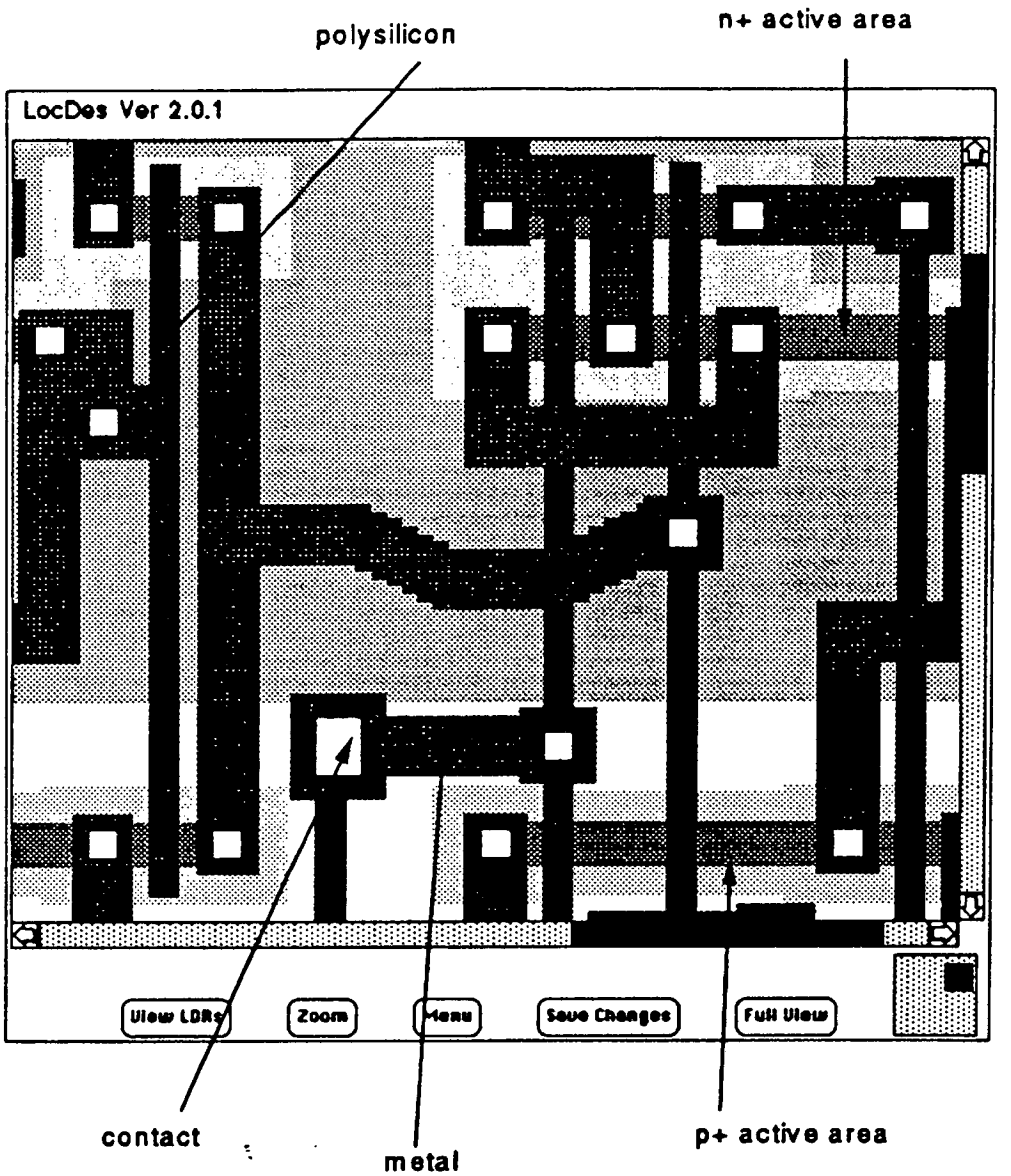


Figure 6–1: LocDes X Window User Interface.

geometry changes within the original GDR envelope and taking care that changes, along the critical path, do not result in a deterioration of performance.

6.2.2 Batch Processing

In the batch processing version of the LocDes program, LDRs are applied using a heuristic algorithm, that makes adjustments iteratively. Changes in layout geometry occur in a number of small steps. After a layout change for a given piece of geometry has been made, attempts to change other neighbouring geometry are made before making additional adjustments to the first piece of geometry. This is to ensure that there is an even application of LDRs, since a large number of small changes evenly applied give greater yield improvement than a small number of large changes. While this procedure will not necessarily give the optimal layout, it will produce a good approximation to it in a reasonable time.

Order of LDR Application.

The optimum application of LDRs would require very intensive analysis of the layout and a prediction of the yield of a large number of potential circuit layouts. Not only would this be very difficult to implement but it would require excessive execution times. Rather than proceed in this manner, LDRs are applied iteratively over the whole design space in a sequence defined by the user as shown in figure 6–2.

The order in which the LDRs should be applied is determined by the yield improvement resulting from their application. Where it is clear that a particular LDR, applied over the circuit area, can give a greater yield improvement than any other LDR then it should obviously be used first. The application of a contact size or track width LDR tends to restrict the use of another LDR because geometry has been increased in size thus reducing the space available for other LDRs. Track displacement LDRs also use up space when they are applied since the track is slightly longer and is wider at the step (figure 5–16). This LDR also redistributes

```
LDRorder[0]=Displace.
LDRorder[1]=Contact.
LDRorder[2]=Displace.
⋮
LDRorder[N]=0.

Procedure ApplicationOfLDRs
        begin
          i=0.
          while LDRorder[i] do {
            switch(LDRorder[i]) do {
              case Contact:
                DoContactLDR().
                break.
              case Displace:
                DoTrackDisplaceLDR().
                break.
              case Width:
                DoTrackWidthLDR().
                break.
            }
          i=i+1.
          }
        end
```

Figure 6–2: Application of LDRs Algorithm.

some of the available space by moving geometry away from close neighbouring geometry. This creates "new" space that may allow the successful application of other LDRs.

The situation where the use of one LDR gives greater yield improvement than others may well be common, particularly in a newer process that has a specific problem with a single process step or steps. The only known example of the application of what might be termed LDRs occurred in such a process where the yield of metal to metal 2 contacts was poor (page 98).

Where a particular LDR is not a clear winner over other LDRs it is less obvious how the optimum yield improvement is to be obtained when applying a mixed set of LDRs. At the present time there is no easy answer to this problem. The solution will require use of a fast yield predictor. At the time of writing, the only one suitable for the task is the McYield Simulator[48]; other simulators are significantly slower. As has been mentioned before the McYield program is not generally available at this time (Oct 1991).

Using a fast yield simulator it would be possible to investigate how the order of LDR application affects the yield. By experimenting with different mixes of LDRs a suitable application sequence could be found for a given process.

Execution Time

The LocDes program execution time is dependent on both the specific layout and the design rules. A Sun 4/60 CPU second gives an average of 180 attempted contact changes or 1.5mm length of track checked for displacement or width increase. Since as many as four iterations are required for the best results, the total execution time can be a number of hours for large circuits.

6.3 Loss of Hierarchy

Most CIF files are hierarchical in nature, with layout cells being defined in terms of other cells. The internal representation of the data within the LocDes program is a flattened view, where all the hierarchy is removed. This can cause an explosion in the amount of data required to represent a design. The size of the resulting data can cause the execution speed of large designs to be greatly reduced. This problem is addressed in Section 6.6.

The hierarchy of the design is removed because, were it retained, the application of LDRs would become restricted by cell to cell interaction. An LDR applied to geometry that is within the maximum design rule[4] of either its cell boundary or the boundary of any overlapping cell has to take into account geometry outside the cell that will interact with it. Where a cell is enclosed in another cell (nearly all cells are) care must be taken to ensure that no geometry from the enclosing cell interacts with the cell geometry.

There are at least three ways to apply LDRs to designs, other than the chosen method of flattening all hierarchy.

1. **Conservative Application of LDRs**

   The LDRs are applied to cells such that any changes that are made to the layout within a cell are at least the maximum design rule away from the cell boundary or the boundary of an overlapping cell. One problem is that an instance of a cell in a design may have geometry from its enclosing cell pass over the cell or sufficiently close to the cell boundary to interact with the cell geometry. For such cells only layout changes that caused no design

[4]More accurately the largest spacing given by the design rules that interact with the type of geometry under consideration.

rule violation in any of the instances of that cell in the current design can be made.

This procedure would greatly restrict the application of LDRs since most designs are made up from a number of small cells, many of which overlap. Large cells are generally constructed from these smaller cells. For many circuit layouts the results would be very much less than optimum. For example some designers make contact structures as a single cell to avoid drawing all the layers forming the contact every time one is needed. Such a layout style would make contact LDRs impossible.

2. **Flatten on Demand**

   Another method that could be used is to flatten cells only where required. That is, where a cell interacts with neighbouring, overlapping or enclosing cells, the interacting cells are flattened. This would almost always result in a completely flattened design and is therefore nearly equivalent but less efficient than flattening the design at the start of execution.

3. **Partial Flattening**

   A combination of the two above methods results in partial flattening of the design. Cells of a given size have all their contents flattened, so that the resultant design is made up from fewer but larger cells. This will only be useful where the design uses more than one instance of these larger cells. Where there is extensive use of overlapping cells or enclosing cells that contain geometry that overlaps the large flattened cells, separate instances of the large cells are required. Partial flattening will, in some instances, allow a portion of the hierarchy to remain. Designs that have few large cells, with several instances, or have geometry that overlaps these larger cells (routing etc.) will result in little gain over a flattened layout.

Each of these methods was considered impractical either because the application of the LDRs would be restricted or the resulting layout would be almost identical to applying LDRs to a flattened design. The partial flattening method

may provide better results, faster execution time, smaller layout file sizes, in some designs but in others little improvement would be gained and because the process is more complex the execution time may well be longer.

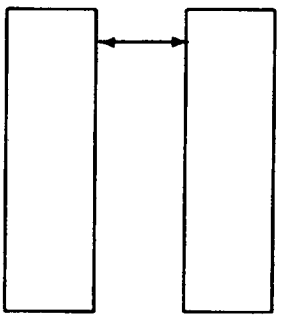
Removing all the hierarchy from the layout was considered the best method of applying LDRs because it allowed their unrestricted application and was simple to implement.

6.4 Initialisation

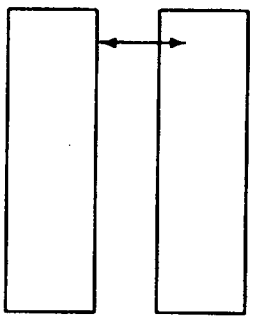
The program take as input a circuit layout in CIF[68] format. The algorithm of figure 6–3 is used to generate an internal representation of the layout in a format suitable for efficient manipulation. This algorithm is identical to the one in figure 3–3 of Chapter 3 except that design rules are read in and all contacts/vias are given separate node identifiers even where they form the same electrical node. The LDR algorithms assume that layout of the same electrical node can be treated differently, in that minimum separation of layers can be less than the global minimum where they form the same node (figure 6–4). This is not true for contacts/vias; so, rather than have an exception to this rule for contacts, they are all given unique electrical node identifiers.

```
Procedure Initialise
  begin
    Initialise data structures.
    Read in Global Design Rules.
    Read in Local Design Rules.
    Read in CIF File.
    Load layout data into Binary trees.
    Make symbolic layer for transistors (ACTIVE AREA AND POLYSILICON)
    Make ACTIVE AREA equal to ACTIVE AREA NOT TRANSISTOR layer
    for all layout Boxes do {
      Give Box an initial unique node number.
    }
    for all track layers do {  /* Metal1, metal2 Polysilicon etc. */
      for all Box of layer do { /* For every piece of layout of type layer. */
        Generate List of all other boxes of same layer that touch Box.
        Label all boxes in List with the same unique node number.
      }
    }
    for all Contact layers do { /* Contact Cut, Via. */
      for all Contacts of Contact layer do {
        Find the node numbers (Node1, Node2) of both track layers forming contact.
        Label all boxes of Node2 on all track layers
        with the node number Node1.
        Label Contact with a unique node identifier.
      }
    }
    Generate array of electrical node linked lists.
  end
```

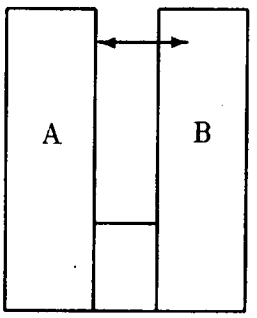
Figure 6–3: Initialisation of Data and Rules for LocDes Program.



(a) Minimum Separation



(b) Design Rule Violation.



In (c) extra material defects do not cause faults, as A is already shorted to B.

(c) No Design Rule Violation.

Figure 6–4: Design Rules for Same Node Geometry.

6.5 Applying Local Design Rules.

The following (simplified) algorithms are used by the program to adjust layout. The program assumes that the original layout has been passed by a design rule checker so that rule checking of adjusted layout is limited to the interaction of the adjusted geometry and the immediate environment.

6.5.1 Increase Track Width

The widths of tracks can be increased where local and global design rules permit. This will reduce the probability that a track will break when fabricated; it will also increase the risk of shorts between adjoining tracks. Many [5] MOS processes have greater yield loss from shorts than from breaks[70] so care must be taken in defining track width LDRs.

A small example layout is shown in figure 6–6(a). The bottom metal track has a track width LDR applied to it. The track is split into segments (figure 6–6(b)), each segment is tested in turn. If there is space above or below the segment greater than that required for the GDR and the LDR separation, a new wider segment is generated. All the design rules for the new larger segment are checked and if there are no violations the change in width is accepted (figure 6–6(c)). The resulting segments are then merged together (figure 6–6(d)). The algorithm used is shown in figure 6–5.

---

[5] No two fabrication processes are identical but there are features common to many mature production lines. Process that are at an early stage of development will often exhibit unique sources of yield loss

Procedure **WidthLDR** Track

begin

Split Track into Segments (see figure 6–6(b)).

for all Segments **do** {

if (Space Above Segment >= $LDR + GDR$) **AND**

(DesignRuleCheck = OK) **then** Increase Segment Size (Top).

if (Space Below Segment >= $LDR + GDR$) **AND**

(DesignRuleCheck = OK) **then** Increase Segment Size (Bottom).

}

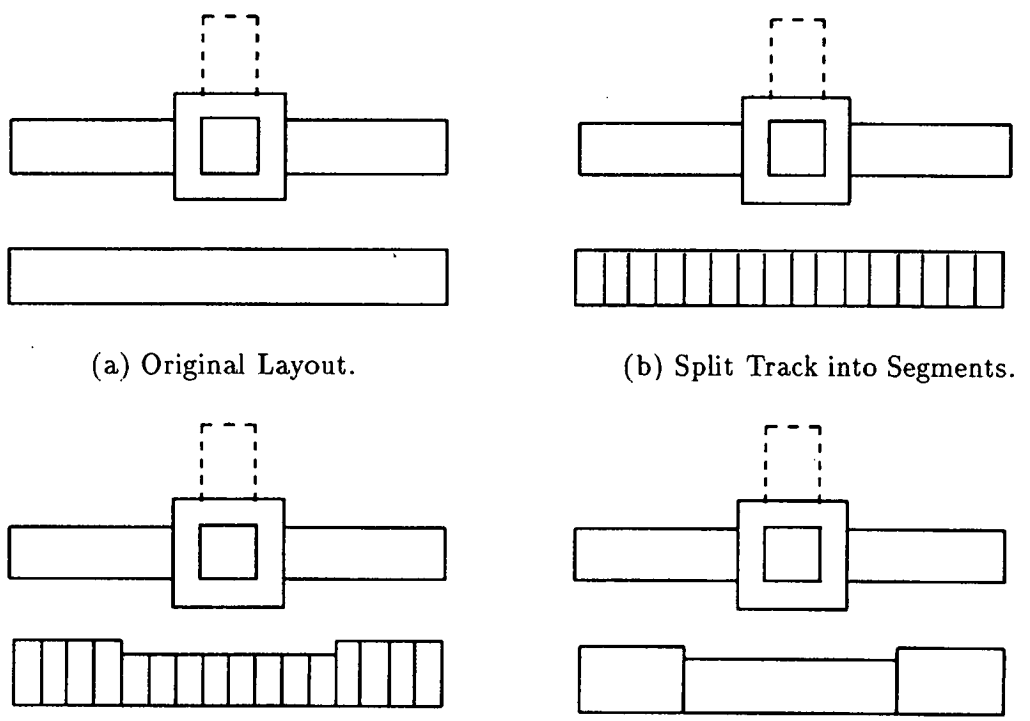
Merge Segments (see figure 6–6(d)).

end

AND OR = logical **AND OR**

\+ = arithmetical plus

Figure 6–5: Track Width LDR Algorithm.



(a) Original Layout.

(b) Split Track into Segments.

(c) Increase Segment Widths.

(d) Merge Segments Together.

Figure 6–6: Application of Track Width LDR.

6.5.2 Increase Contact Size

Yield loss from contacts can be a significant proportion of total yield loss. Using larger contacts where there is available space can reduce this loss. Where contacts are increased the overlap layers must also be adjusted to at least the minimum overlap specified by the design rules. This can give rise to some yield loss due to an increase in the probability of shorts between these layers and the surrounding layout geometry. This loss is offset against the gain in yield from the larger contact.

An example contact layout is given in figure 6–8(a). An LDR adjustment has been applied to this contact to give figures 6–8(b) and 6–8(c). A contact can be increased in four directions (± X and Y) with the directions being tested in turn until an adjustment has been made. In order for a contact change to be acceptable, each of the layers forming the contact must be checked to ensure that there is sufficient space to extend them without design rule violations. If all layers can be suitably modified then the contact and corresponding overlap layers are extended. Further attempts to increase the contact, in the same or another direction, can be attempted after the same procedure has been applied to the other contacts in the layout. The algorithm used is given in figure 6–7.

6.5.3 Track Displacement

In a mature process the dominant cause of yield loss can be shorts between metal tracks. This loss can be reduced, sometimes by a significant amount, by displacing tracks away from surrounding geometry where space permits.

Figure 6–10(a) gives an example of the top metal track being displaced using the algorithm in figure 6–9. The track is first split into segments. Where there are crossovers with design rules determining minimum edge separation, larger segments equal to the width of the crossover track plus two edge separations and two associated segments are formed as shown in figure 6–10(b). This is the

```
Procedure TryChangeContact Contact
        begin
          Success=FALSE.
          While all possible Directions /* +X -X +Y -Y. */
          AND Success=FALSE do {
            for all Layers forming Contact do { /* Contact, Metal, Poly etc. */
              if Space In Direction for Layer >= LDR[Layer] + GDR[Layer])
                AND (DesignRuleCheck = OK ) then Success=TRUE.
              else {
                Success=FALSE.
                Exit for Loop. /* Failed */
              }
            }
            if Success= TRUE then Increase Contact in Direction.
          }
        end
```

Figure 6–7: Contact LDR Algorithm.

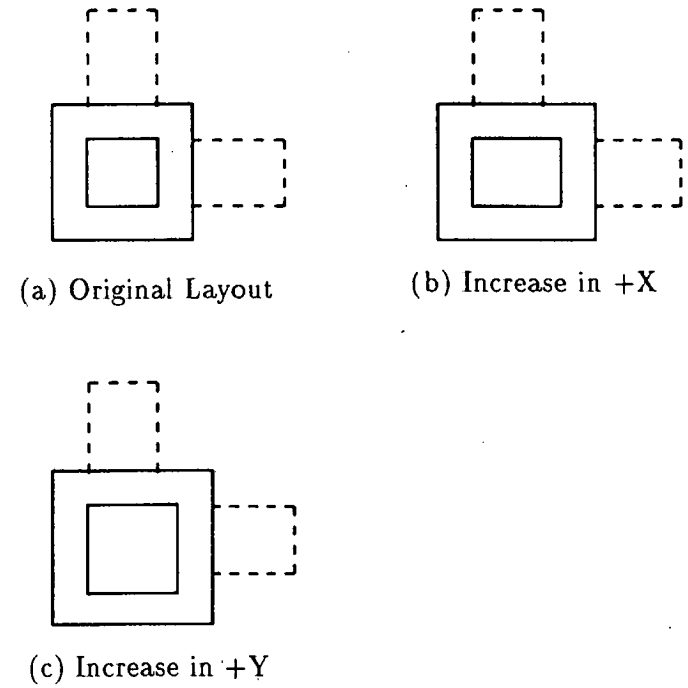


Figure 6–8: Application of Contact LDR.

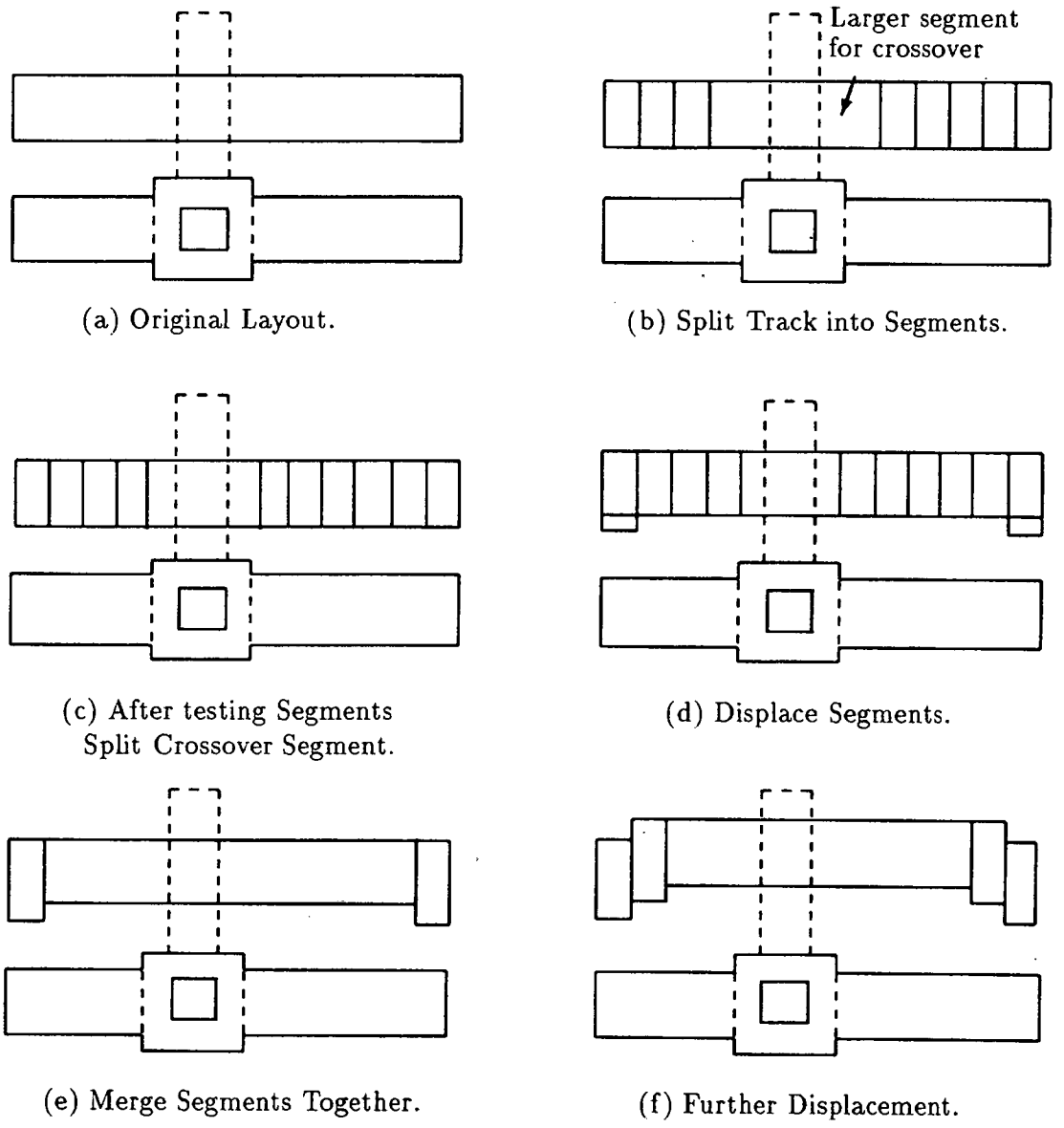
minimum length of track that can be displaced at the crossover without rule violation.

Each of the segments is examined in turn to determine if they can be moved. Normally a segment would be marked for a move when there is more space available on one side of the segment than the other, with the proviso that the displaced segment does not violate any design rules. If a "crossover" segment is marked for displacement, it is divided into three. The sub-segments at either end are the same size as the normal segments with the middle part being equal to the remainder of the "crossover" segment defined above (figure 6–10(c)). This is to ensure that during displacement the track width is not extended at the crossover point, as this would result in a large segment of the track increased in width. This procedure is only implemented if there are GDRs associated with edge separation. The marked segments are then collected into groups of continuous lengths of track that can be moved in the same direction. All of these blocks that are greater than a minimum length are displaced. This minimum, normally greater than the GDR minimum track width, ensures that displacements of very short

```
Procedure DisplaceLDR Track
    begin
      Split Track into Segments. /* small segments and crossovers */
      for all Segments do {
        if CheckMoveUpOK then Mark Segment Up.
        else if CheckMoveDownOK  then  Mark Segment Down.
        if Segment is Crossover AND is Marked then
          Split Up Segment to give normal Segment on each end
          (See figure 6-10(d)).
      }
      for Direction Up and Down do {
        Make Current Block
        for all Segments do {
          if  Segment Marked Direction then Add to Current Block.
          else if Current Block NOT Empty then {
            Add Current block to ListofBlocks.
            Make new Current Block.
          }
        }
      }
      for all Blocks do {
        if Block >= minimum movable size then {
          for all Segments in Block do {
            if first OR last Segment in Block
              then Extent Segment by Displacement LDR in Mark Direction.
            else Displace Segment by Displacement LDR in Mark Direction.
          }
        }
      }
      Merge Segments into Track (see figure 6-10(e)).
    end
```

Figure 6–9: Track Displacement LDR Algorithm

lengths of track that provide no yield improvement are excluded. The displacement is performed by extending the first and last segments of the block in the displacement direction and displacing, by the same amount, the remaining block segments (figure 6–10(d)). The resulting segments are then merged together to give the final layout (figure 6–10(e)).



(a) Original Layout.

(b) Split Track into Segments.

(c) After testing Segments
Split Crossover Segment.

(d) Displace Segments.

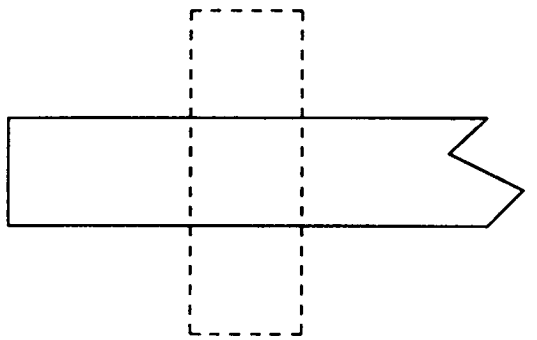
(e) Merge Segments Together.

(f) Further Displacement.

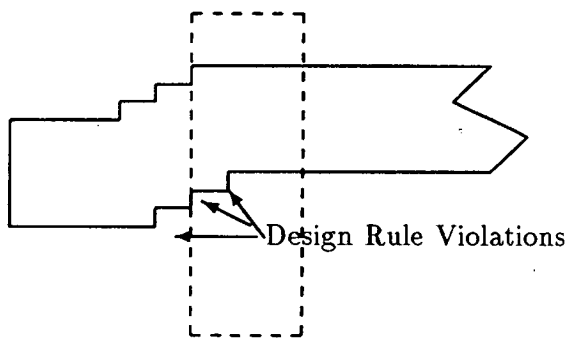
Figure 6–10: Application of Track Displacement LDR.

Displacement of Track at Crossovers

Crossovers regions of track are treated as a special case in track displacement LDRs where there is a minimum edge separation design rule for the layers forming the crossover. The results of a displacement could give rise to design rule violations, if the edge separation rule were to be ignored. This is shown in figure 6–11. The violations shown in figure 6–11(b) are technically layout errors but they would not necessarily result in a reduction in yield. The layout of figure 6–11 would produce a physical circuit layout similar to that shown in figure 6–12. The yield of this structure will often be the same as that produced by the more usual orthogonal crossover (figure 6–11(a)). This is of course dependent on the fabrication process.

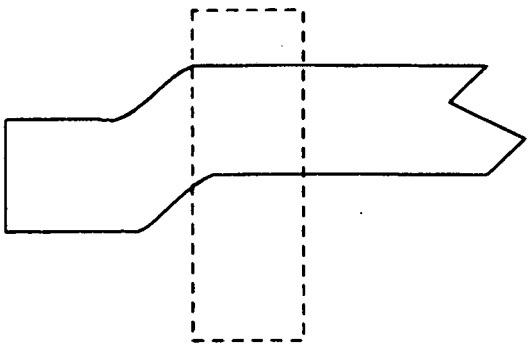


(a) Original Layout : Crossover



(b) Track Displaced with resulting Errors

Figure 6–11: Crossover with Track Displacement Resulting in Design Rule Violations.



Displaced Track as Fabricated

Figure 6–12: Resulting Physical Layout from Track Displacement at Crossover.

The need for special treatment of track crossovers will, for some fabrication processes, prove to be unnecessary. Where this is done the resulting layout will give many layout error warnings when passed thorough the global design rule checker. If care is not taken these may mask any "real" layout errors.

6.6 Large Circuit Layouts

Even using the most efficient data structures some circuit designs are too large to be processed with all the data in on board RAM. Where the data representing the layout is greater than this the time taken to page data in and out of RAM results in a large reduction in execution speed. The present limit on a SUN 3/80 with 16 MB of memory is 25,000 mask geometry boxes. This is about 5mm^2 of 3 μm CMOS. As many chip designs are larger than this, provision has been made for larger designs.

The application of LDRs is a local calculation that therefore permits designs to be split up and LDRS applied in smaller sections. Where the number of layout geometry boxes exceeds a specified limit (normally set at 15,000) the LocDes program divides the circuit layout into a number of segments. The layout can

be processed in 1, 2, n^2 n=2, 3, $\cdots$ segments depending on the number of boxes in the design. Each segment overlaps with its neighbours by at least twice the maximum design rule operating distance for the layout technology (figure 6–13). The whole design is completed by processing all the individual segments, deleting the overlap from the segments, and appending all results to a single file. This ensures that geometry at the edge of the segments does not violate global or local design rules from neighbouring segments.

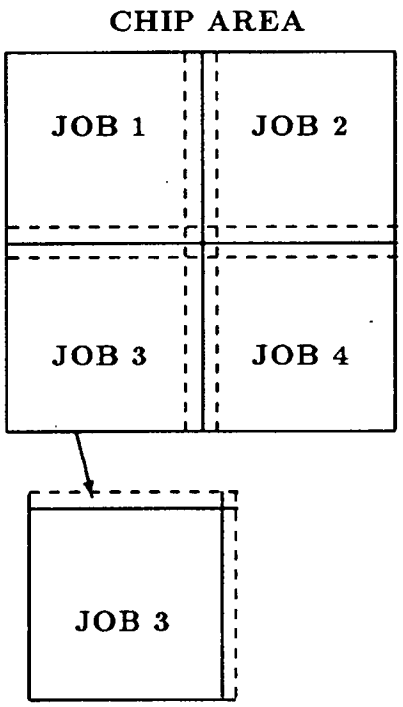


Figure 6–13: Segment Generation from Large Circuit Layouts.

The effect on the execution time of the program of segmenting large designs is shown in figures 6–14, 6–15 and 6–16. These graphs are the result of runs of the LocDes program using different numbers of layout segments to process a circuit layout. All runs involved a 3 micron double metal CMOS chip containing 22,000 layout geometry boxes. The circuit was processed for contact and metal

1 displacement LDRs rules for two iterations. Execution times were found using the UNIX time[6] command.

Load Time (figure 6–14).

The load time is the time taken by the program to do the initialisation of data structures. This was found by running the program without applying any LDRs. This shows that the load time increases as the number of segments is increased. This is not unexpected since the CIF parsing program is run for each segment to filter out all geometry boxes in the segment being processed.

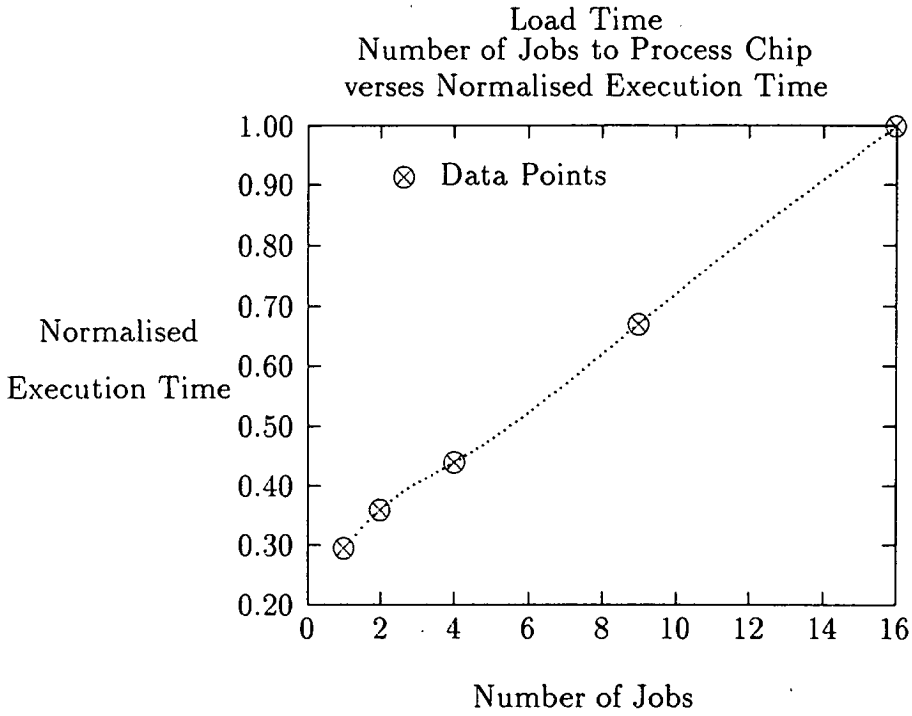


Figure 6–14: Graph of LocDes Load Time

Execution Time (figure 6–15).

The execution time versus number of jobs (segments) shows that there is an increase in execution time as the number of segments is increased. This is

[6]See Unix manual section 1

due to the extra computation required to select the appropriated section of the layout, clip off the segment overlap regions and to combine the results of all the segments.

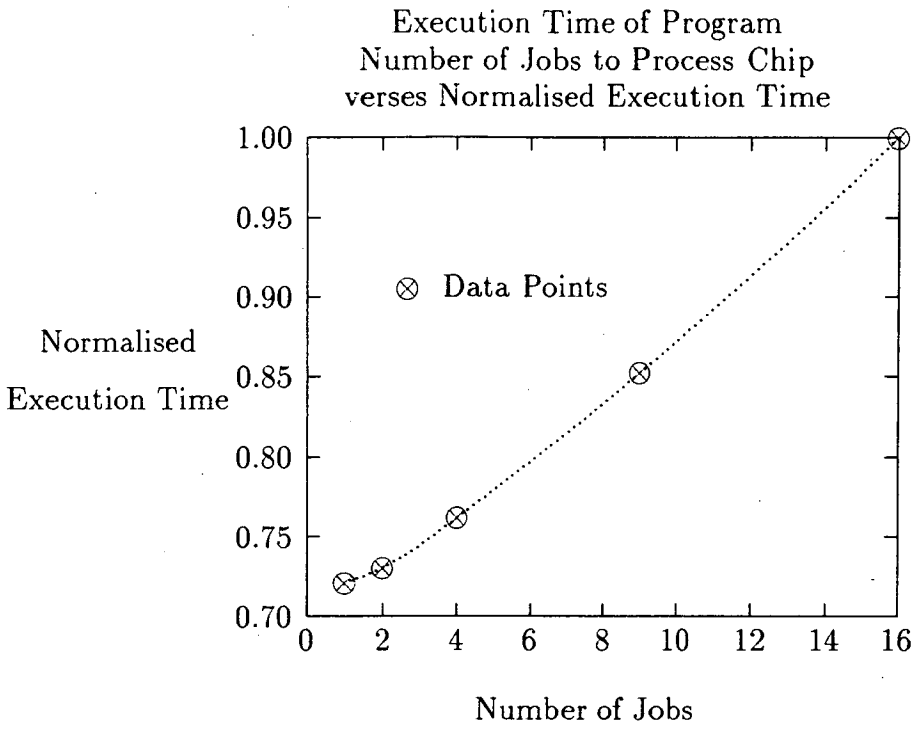


Figure 6–15: Graph of LocDes Execution Time

Execution Time minus Load Time (figure 6–16).

This graph indicates the actual time spend in applying LDRs, to generate the new layout i.e. the "work" time. It can be seen that work time decreases until a minimum at nine segments. This is due to the smaller spatial data structure that allows a faster region query and hence more rapid design rule checking.

The timing for sixteen segments is increased, which is a result of the increase in the total overlap region. LDRs are applied to geometry in the overlap regions that are subsequently discarded when the segments are clipped to form the overall result. In this example the same circuit was used in all runs, so that as the number of segments increases the amount of overlap as a function of the circuit area increases. Consequently these results should

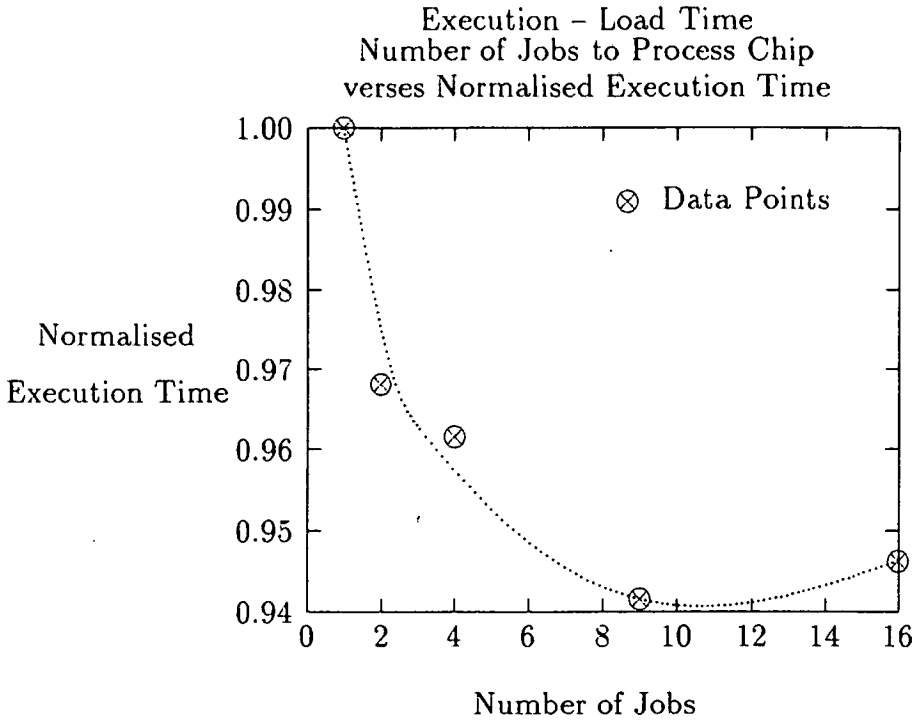


Figure 6–16: Graph of Execution Time minus Load Time

not be taken as an indication that in general more than nine segments will cause a increase in work time. For large circuits the division into more than nine segments is essential and the proportion of overlapped segment boundaries to the total layout will be negligible.

6.6.1 Distributed Processing

To reduce the time taken to process large circuits the LocDes program includes a facility to use multiple hosts to process a single circuit layout. This could be used for any layout but is of most use where a circuit layout is sufficiently large that it would normally have to be divided into segments. Figure 6–17 shows the graph execution time per segment. The normalised execution time for each segment is shown. This indicates that if these segments are processed in parallel the best times that could be expected, assuming a separate host for each job, is the time taken by the slowest segment. From the timings in figure 6–17 it can be

seen that, with sixteen hosts the best possible execution time for this particular circuit layout is 10% of the time taken with one host.

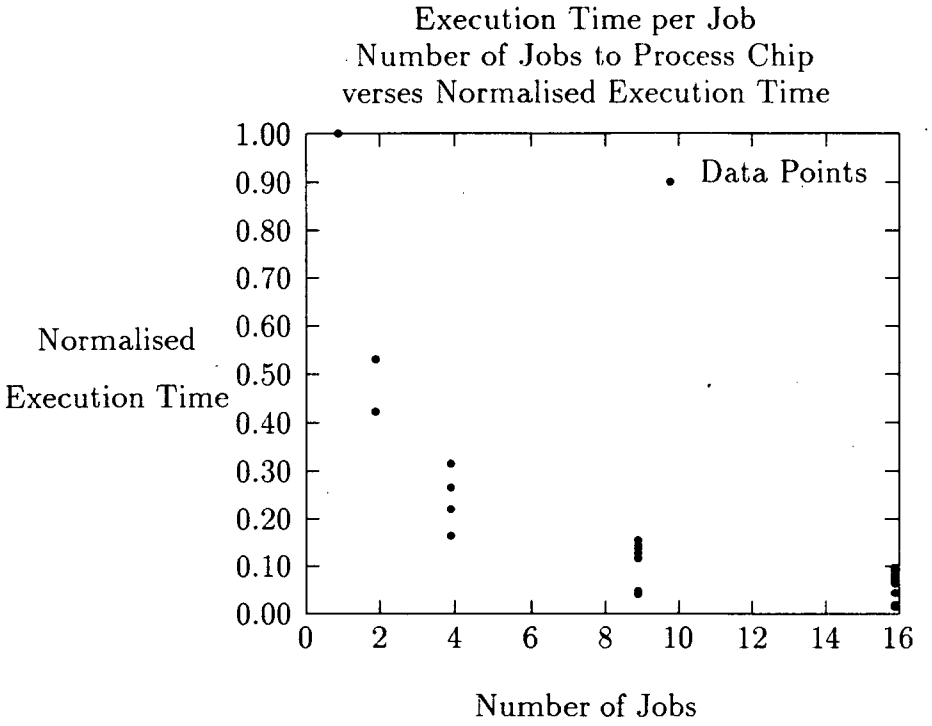


Figure 6–17: Graph of Execution Time Per Job in LocDes.

The program operates in parallel by generating segment coordinates that are used to produce a list of commands. The commands are of the form.

```
LocDes -X nnn -Y nnn -x nnn -y nnn [other options] design-name
```

The flags `-XYxy` give the segment coordinates of the design that are to be processed. The `[other options]` determine which local design rules are to be used. These are originally specified by the user in the command line of the program and copied to the parallel command list.

Once the list of commands has been generated they are sent to a remote host to be executed. Only one command is sent to each host. If there are more segments than available hosts, the program waits until a remote host has finished processing its segment then sends it another. Thus it is possible to use as many remote hosts as there are segments or use a limited number of hosts efficiently each processing only one segment at a given time.

The results generated by the remote hosts are sent back to the controlling host where they are merged to form the final result (figure 6–18). The response times using one host per segment are given in figure 6–19.

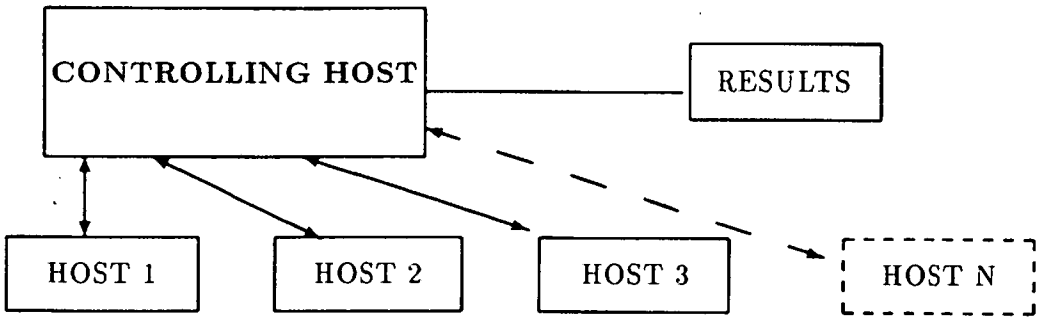


Figure 6–18: Data Flow in Parallel operation of LocDes.

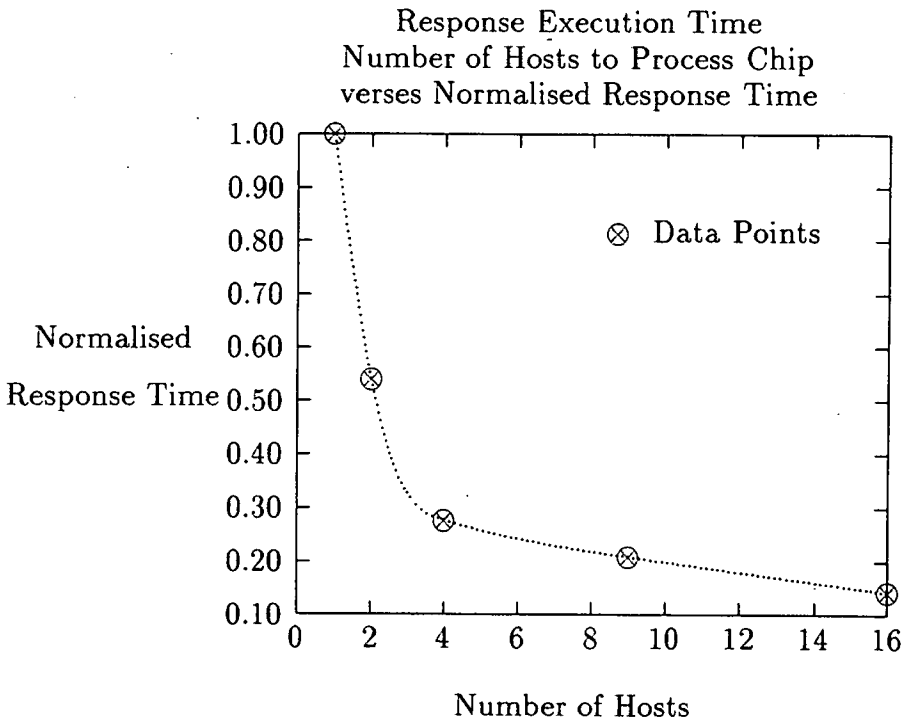


Figure 6–19: Response Time of Parallel LocDes.

Interprocess Communication

The LocDes Program allows two possible interprocess communication strategies,

- **Rsh – Remote Shell.**

  The rsh(section 1C Unix manual) unix command is used to connect to a specified host and execute a specified command. Rsh copies its standard input to the remote command and the standard output of the remote command is copied to its standard output.

  In order that a number of connections via rsh can be made to remote hosts, a socketpair(section 2 Unix manual) is generated and the process is forked using fork(2). This has the disadvantage that a large number of processes are generated on the local machine, 3 for each connection to a remote machine.

- **Rexec – Return Stream to Remote Command.**

  The rexec(section 3N Unix manual) unix command returns a stream (socket) to a specified command on a given remote host. This stream is connected to the standard input and standard output of the remote command.

  Rexec does not require a fork in the local process and does not generate any local processes. This makes it more efficient than rsh. The main difficulty encountered using rexec is that it requires a daemon, normally the inetd(section 8C Unix manual) to be running on the remote host. The Internet services daemon listens for connections to the services listed in its configuration file (usually /etc/inetd.conf). In order for rexec to work it must be listed in this file. Rexec is not always present; it is often commented out, as it is a possible security hole (see Unix manual section 8C, rexecd).

  While it should not be a problem on most systems there is a reluctance by some system managers to provide this service.

Rexec is the most efficient method of executing the program on multiple hosts. It does not require the paging in and out of processes to communicate between hosts and there is also no need to re-communicate data from the child processes to the parent process. As there is a perceived security problem with rexec the

rsh method was provided for the times when a system administrator could not be persuaded to provide the rexec service.

6.7 Conclusions

This chapter has introduced the LocDes program and the some of the algorithms it uses to automatically apply LDRs to circuit layout. LDRs can be applied to selected circuit geometry by using the X windows version of the program. This allows a circuit designer to take account of features of the design that may not respond well to automatic modification. For example, where the timing of circuit is critical the clock line should not be modified since this may change the timing characteristics.

Where there are no critical circuit components the program can be used in a batch mode to iteratively apply a set of LDRs to the whole layout. Because very large circuits may take too long to be processed in serial, the program has been designed so that it can, if required, process a design in parallel, using either networked computers or a multiple CPU mainframe.

Chapter 7

Design Rule Checking.

This chapter explains the method and algorithms used by the LocDes program to check that layout modifications do not violated any of the IC layout rules. The data structure used by these algorithms to obtain efficient region searches is given.

7.1 Introduction

In order to produce working circuits any layout output generated by the LocDes program must have an equivalent netlist to the circuit input and obey all the process design rules. The program only checks that changes performed by itself conform to the design rules. It does not check the validity of the initial layout, assuming it to be correct. Complete design rule checking is more efficiently done by a program designed specifically for that task. The design rule checking done by the LocDes program is a subset that is sufficient to deal with the changes in layout attempted by the program.

7.2 Track Width LDRs.

The track width LDRs increase the size of existing track segments. To ensure that an increase in geometry size does not cause a design rule violation the algorithm of figure 7–1 is used. There are two inputs to the procedure. The new segment to be checked and the segment layer. The interaction of every layer with the new segment is examined in turn to determine if any rules are violated. This is done by generating a test box (figure 7–2(a)) for every layer that has an LDR or GDR interacting with the segment layer. The dimensions of this test box

```
Procedure DesignRuleCheck Box Layer
        begin
        for TestLayer = all layers do {
            if GDR or LDR exists then {
              DesRul = GDR[TestLayer][Layer]+LDR[TestLayer][Layer].
              TestBox = Box + DesRul.
              if TestLayer = Layer then
                if any different node TestLayer Box touches TestBox
                  then return FAILED. /* Design Rules Violated */
              else if  TestLayer Box touches TestBox
                  then return FAILED. /* Design Rules Violated */
            }
          }
          return SUCCESS
        end
```

Figure 7–1: Design Rule Checking Algorithm for Track Width LDRs.

are dependent on the design rules for the two interacting layers. Any geometry of layer that has a different electrical node identifier from the segment and falls within or touches the area defined by the test box causes a design rule violation. Where the geometry has the same node identifier it is part of the same track as

the segment. It is acceptable for such geometry to be closer than the design rule distance or even overlap. For test boxes generated for the interaction of layers that are a different type from the segment layer, any geometry of that layer that falls within or touches the test box area causes a violation. In this case even geometry that forms the same electrical node causes design rule violations.

Some aspects of this algorithm are shown in figure 7–2.

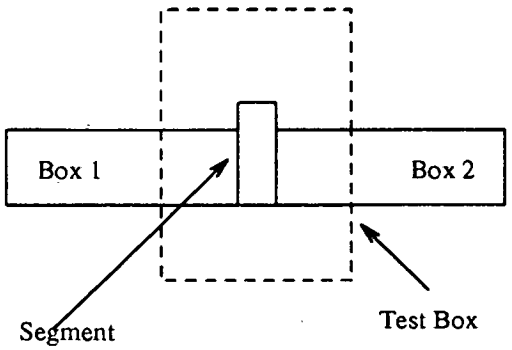
- Figure 7–2(a) shows the situation where the geometry found in the test box, generated for segment layer to segment layer interaction, is of the same layer type as the segment and forms the same node. There is no design rule violation.

- Figure 7–2(b) shows a similar situation to (a) except that the test box overlaps a piece of geometry (Box 3) of the same layer as the segment. Box 3 is not the same node as the segment so this is a design rule violation.

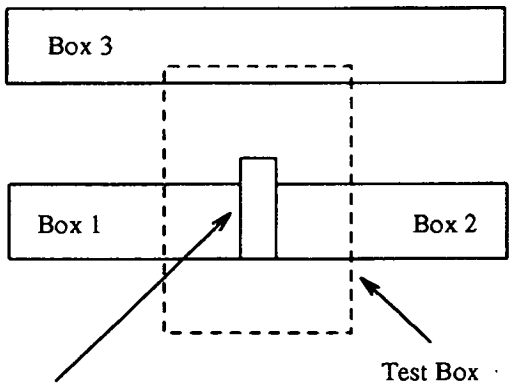
- Figure 7–2(c) again shows a situation similar to (a) where some of the geometry in the test box, is of the same layer type as the segment. Since this geometry forms the same node it does not cause a design rule violation. The other piece of geometry, labelled Box 3, is of a different layer type. It is not possible using the existing test box to determine whether this will cause a violation. This can only be done by generating a test box that represents its layer type and the segment's layer type interaction space. The possible outcomes of this are shown in (d) and (e).

- Figure 7–2(d) represents one of the two possible outcomes from (c). A new test box is generated for segment layer and Box 3 layer interaction. Here there is no design rule violation as box 3 falls outside the new test box.

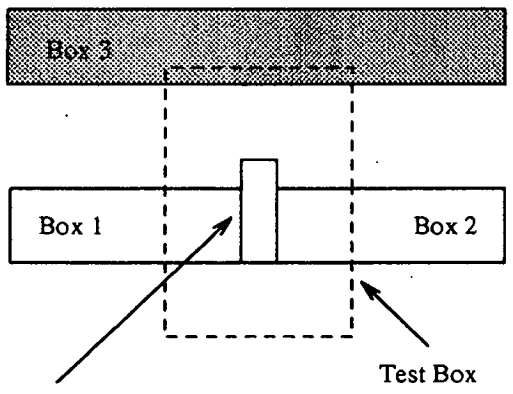
- Figure 7–2(e) represents the other possibility where the design rule space required overlaps the geometry of Box 3 and is therefore a design rule violation.



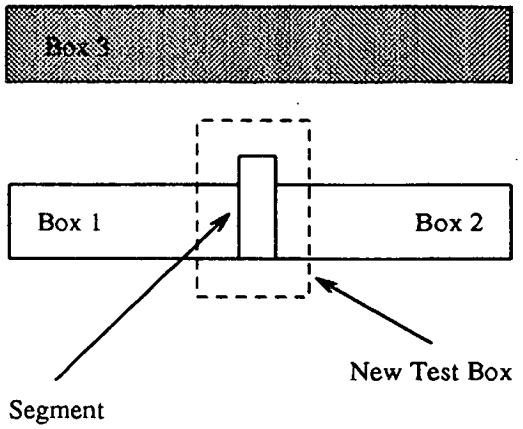
(a) Same Layer and Node
No Rule Violation



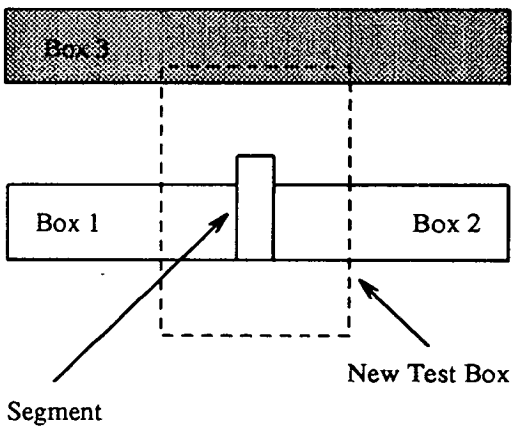
(b) Same Layer Different Node
Rule Violation



(c) Wrong Test Box for Box 3
No Rule Violation



(d) No "Geometry" in New Test Box
No Rule Violation



(e) Different Node Geometry Found
Rule Violation

Figure 7–2: Design Rule Checking in Track Width LDR.

7.3 Contact and Contact Overlap LDRs.

Contact overlap LDRs, like track width LDRs, involve the expansion of existing geometry and use an algorithm similar to the track width algorithm(figure 7-1). The algorithm differs in that geometry of both the layers forming the contact can be closer than the minimum design rule separation, where this geometry is the same electrical node as the contact (figure 7-3).

```
Procedure DesignRuleCheckContact Box Layer Layer2
        begin
        for TestLayer = all layers do {
            if GDR or LDR exists then {
              DesRul = GDR[TestLayer][Layer]+LDR[TestLayer][Layer].
              TestBox = Box + DesRul.
              if TestLayer = Layer OR  Layer2 then
                if any different node TestLayer Box touches TestBox
                  then return FAILED. /* Design Rules Violated */
              else if  TestLayer Box touches TestBox
                  then return FAILED. /* Design Rules Violated */
            }
          }
          return SUCCESS
        end
```

Figure 7-3: Design Rule Checking Algorithm for Contact Overlap.

Contact LDRs are identical to contact overlap LDRs except that the contact layer itself is expanded. Contacts are different from other geometry in that there are design rules regulating the separation of contacts that form the same electrical node. The algorithm could be easily extended to handle this with an extra conditional statement. However, it proved more efficient to label all contacts and

vias with a false but unique electrical node identifier[1]. This enables the algorithm of figure 7–3 to be used.

7.4 Displacement Extensions.

The conditions for an acceptable segment displacement are that the segment must not violate any of the design rules or change the netlist (circuit function). The standard design rule algorithm used (figure 7–1) for track width LDRs and contact LDRs is not sufficient, as it does not check for the sort of "damage" that can be done by a displaced segment. The additional conditions that need to be checked are:

- **Violation of Contact Overlap.**

  The overlap of a contact/via can be damaged by displacement of a track that forms the overlap as shown in figure 7–4. This does not necessarily violate any of the minimum separation design rules and so is not picked up by the standard checking routine.

  In order to ensure that overlaps are not damaged by displacement of a track forming overlap it is necessary to mark regions that form overlap. This is done by generating new symbolic layers, one for each layer that can be displaced. A symbolic layer is required for each displaceable layer. Process design rules may specify different overlaps for each layer forming the contact. The new layers are formed from the contact and via layers by over-sizing the contact/via by the amount required to generated the minimum overlap (figure 7–5). These symbolic layers are updated when a contact/via are changed in the same way as the actual overlap.

---

[1]Where the electrical node of a contact/via is required it can be determined from the metal geometry found within the area of the contact

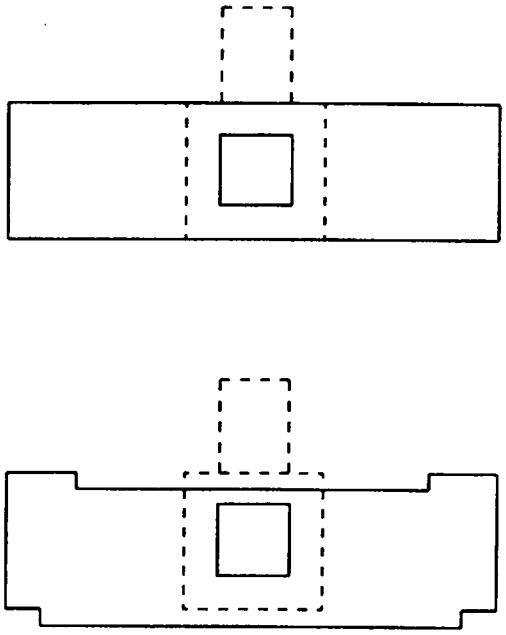
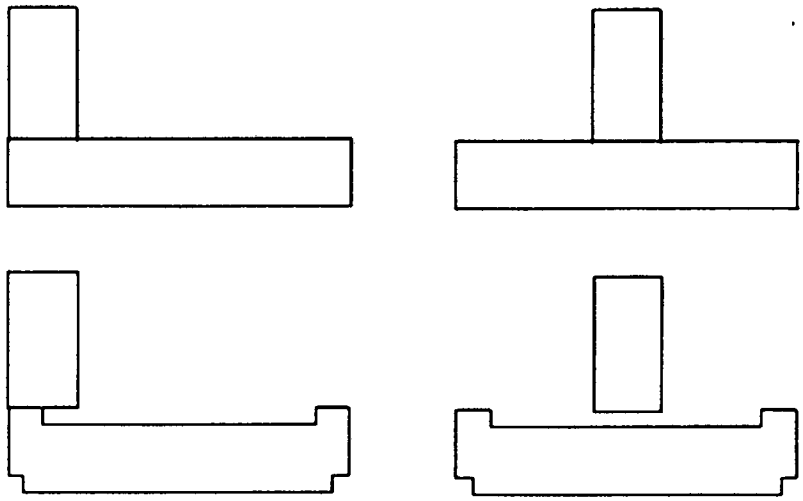


Figure 7–4: Contact Overlap Violation by Displacement

```
Procedure FixedPoints
        begin
          Initialise Binary tree for FixedMetal1 layer.
          FixedContactPoints Contacts Metal1 FixedMetal1.
          FixedContactPoints Vias Metal1 FixedMetal1.
          Initialise Binary tree for FixedMetal2 layer.
          FixedContactPoints Vias Metal2 FixedMetal2.
          Initialise Binary tree for FixedPoly layer.
          FixedContactPoints Contact Poly FixedPoly.
        end
Procedure FixedContactPoints ContactType Layer FixedLayer
        begin
          for all boxes in ContactType do {
            Generate minimum OverlapBox of Layer for ContactType.
            Add OverlapBox to Binary tree FixedLayer.
          }
        end
```

Figure 7–5: Fixed Layer Generation Algorithm.



(a) Wire Thinning by Displacement

(b) Wire Break by Displacement

Figure 7–6: Potential Errors from Track Displacement

- **Wire Thinning and Breaks.**

  The displacement of a track can cause thinning at the track corner as shown in figure 7–6(a). A more serious and related problem can occur at a branch point of a wire. The track can be displaced as shown in figure 7–6(b) causing a break in the connectivity and an almost certain circuit fault. Neither of these conditions necessarily violates the minimum separation of layers and is not detected by the standard checking routine.

The extra design rule checking required for displacement of track segments is shown in figure 7–7. Here the opposite, from the direction of displacement, side of the segment is tested to determine if it is in contact with any other box of the same layer type as itself. If it is then that displacement would cause a thinning or breaking of a wire and is therefore rejected. A similar test is performed using the appropriate fixed layer to determine if an overlap will be damaged. If the opposite side of the segment touches any geometry of the fixed layer the displacement is rejected.

The **BoxNotConnected** procedure is performed before the new displaced segment is generated and if it returns successfully the displaced segment is gen-

```
Procedure BoxNotConnected Direction Layer Segment
        begin
          OpDirection = Opposite to Direction. /* e.g. +x give -x */
          if  OpDirection Side of Segment touches any other boxes of Layer.
            then return FAIL
          if Layer = METAL then {
            if OpDirection Side of Segment touches any FixedMetal1 layer.
              then return FAIL
          }
          if Layer = METAL2 then {
            if OpDirection Side of Segment touches any FixedMetal2 layer.
              then return FAIL
          }
          if Layer = POLY then {
            if OpDirection Side of Segment touches any FixedPoly layer.
              then return FAIL
          }
          return SUCCESS
        end
```

Figure 7–7: Extra Design Rule Checking for Segment Displacement Algorithm.

erated. If the **DesignRuleCheck** procedure of figure 7–1 returns successfully a displacement is allowable.

7.5 Spatial Data Structure

The execution time of the program is heavily dependent on the efficiency of the design rule checking. This in turn is dependent on the internal organisation of the layout geometry. A spatial data structure capable of representing IC layout in a space efficient manner with a fast method for identifying neighbouring geometry is essential. Any data structure representation must provide:

- **Space Efficiency.**
  The amount of data to be manipulated depends on the circuit layout. Since the layout hierarchy must be flattened, a small CIF file can result in a very large amount of layout data. It is essential that this be represented in as small an amount of memory as possible.

- **Fast Region search.**
  The program makes a large number of region searches to determine the position of neighbouring geometry for design rule checking. The amount of geometry within the region of design rule interaction is very small in comparison to the total chip area. The execution time of the program is greatly affected by the efficiency of the search routines.

- **Fast Data Structure Generation.**
  Obviously it would be preferable if the initialisation of the data structure from the CIF layout could be fast, but since this operation only occurs once it is not as critical as the speed of operations on the structure. What is more important under the heading of Data Structure Generation is the speed at which boxes of layout geometry can be deleted and inserted when the size and position of boxes is changed by the program.

- **Simple Algorithm.**
  While this is not essential it is desirable to reduce the programming effort required and simplify the debugging.

7.5.1 Common Spatial Data Structures.

There are a number of well known methods of representing mask-level physical layout. These are important in many applications, with the most notable being circuit compaction. The most common data structures used are;

- **Painted Quad Tree.**

  The painted quad tree exhaustively splits the layout plane into quadrants until every quadrant is either completely "painted" or empty. This requires a large number of tree nodes. The memory requirement for this structure is too great.

- **Bisector List Quad Tree.**

  The bisector list quad tree stores the rectangles that intersect quadrant edges in X and Y bisector lists. This brings about great complexity of the data structure and increased execution time. The structure is, however, memory efficient.

- **Multiple Storage Quad Tree.**

  This quad tree does not use bisector lists; instead it references each rectangle within a quadrant by an index. The rectangles themselves are stored in an array and can be referenced by the index.

7.5.2 The Adaptive Multiple Storage Binary Tree

The spatial data structure used is the Multiple Storage(MS) binary tree which is a variant of the MS-quad tree [85]. It is more properly described as an adaptive multiple storage binary tree and is identical to the MS-quad tree, except that each node points to two further node/leaf structures rather than four. This simplifies the algorithm, and has no significant effect on execution time or memory requirement.

The MS-tree stores the layout geometry (rectangles) in a tree structure, each node represents an area and can either point to two sub-areas or contain an array

of references to all the geometry in that area, in which case it is a leaf cell. If a new item is added to a leaf cell and the reference array is already full, the area is split and all items in the existing array are repositioned in the appropriate sub-area leaf arrays.

If geometry straddles more than one area defined by a leaf cell, each leaf stores a copy of the reference to it. This does not use as much memory as might be thought at first sight, as only the index or pointer to the geometry is stored. The geometry itself is stored in a dynamic array and referenced by an index. In fact the MS-quad tree has about the same memory usage as the bisector list quad tree [86], a simpler data structure and faster execution time[87].

Each layer is stored in a separate dynamic array and an associated binary tree references by indices the data stored in the array. An example of a mask layout is given in figure 7–8 with the resulting data structure. This shows the multiple storage of rectangles e.g. 3, 4, 10, etc. A large example would give a more complicated data structure figure 7–9.

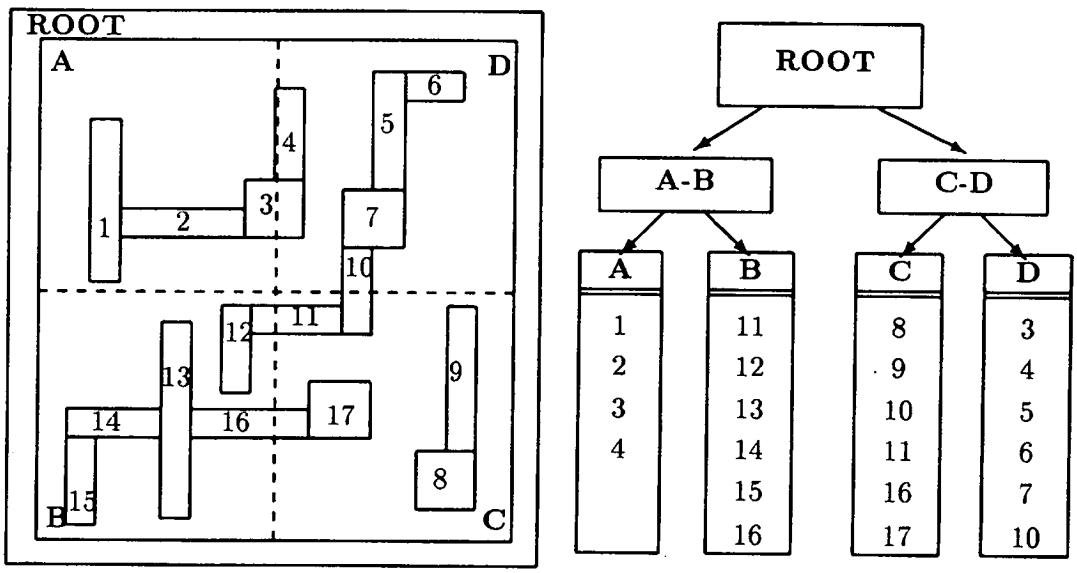


Figure 7–8: Mask Layout and Data Structure in MS Binary Tree.

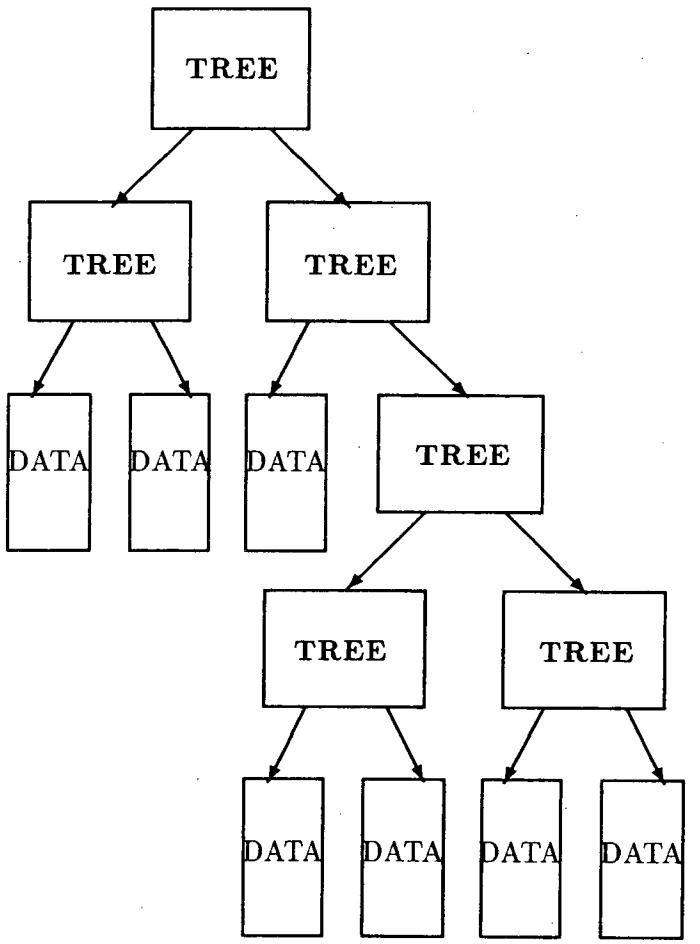


Figure 7–9: Data Structure of MS Binary Tree.

Binary Tree Elements.

The MS binary tree is made up from a number of linked data structures of the form of figure 7–10.

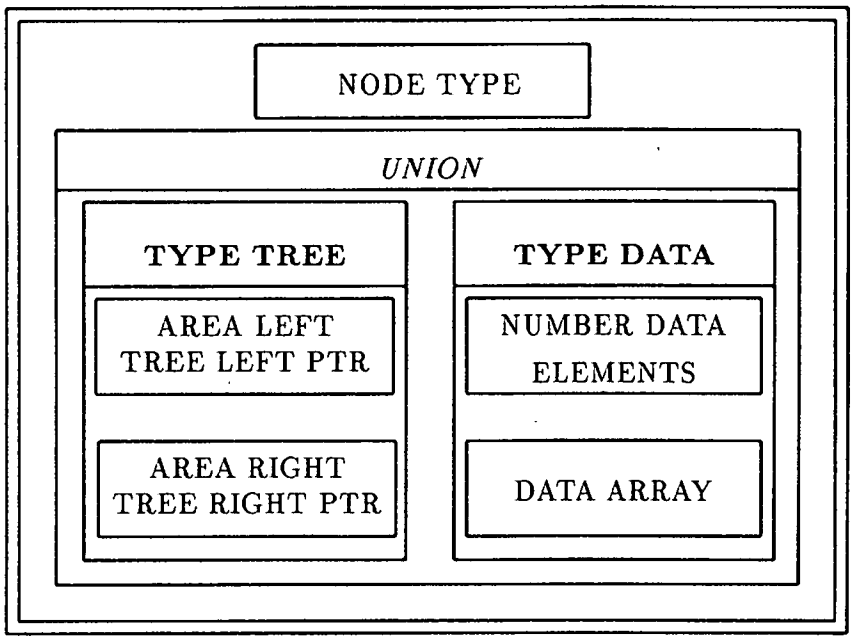


Figure 7–10: MS Binary Tree Building Block

- **Bintree Structure.**

  The Bintree structure consists of a union that can be interpreted as a Tree or a Data node depending on the variable node type that indicates whether the union is a Tree or Data node.

  - **Tree node.**

    A Tree node contains pointers to two further Bintree nodes. These can of course be interpreted as either Tree or Data type nodes. The area of the layout geometry contained in these branches is also given (Area Left, Area Right).

  - **Data node.**

    A Data node contains the data array and a variable that holds the

number of active box indexes stored in the array. The maximum number of elements this array can hold is defined by a variable `MAXBOX`, the value of this is set to maximise execution time (section 7.5.2).

Optimum Size of Index Array

The optimum size of the index array within the Data nodes of the binary tree was determined experimentally, by modifying the source code and re-compiling for different values of the variable `MAXBOX`. The results are shown in figure 7–11, they indicate that the optimum size of the array is nine elements.

The application in which the MS data structure is used determines, to a large extent, the index array size. The fast region search is optimised for small sizes of index array, while the initialisation of the data structure is optimised for large array sizes. The LocDes program executes a very large number of region searches, consequently the optimum size of the array is small.

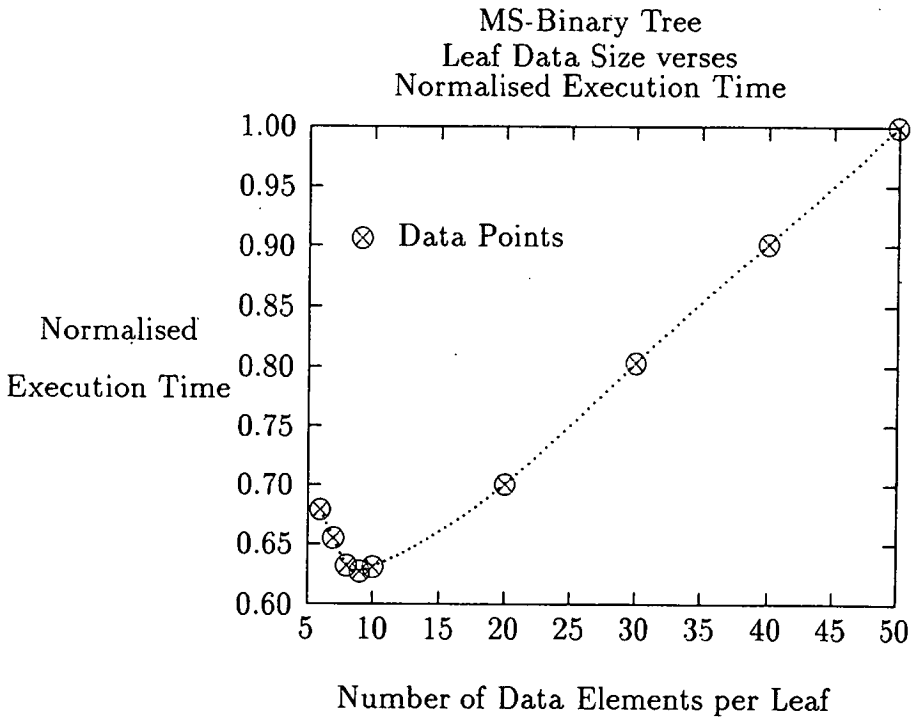


Figure 7–11: Reference Array Size Optimisation for MS Binary Tree.

7.6 Geometry Search Algorithm.

```
Procedure FindGeometry Area Node Layer
    begin
      if Node = Data then {
        for Elements in Data Array do {
          Index = DataArray[Element].
          if Area of Box BoxArray[Layer][Index] touches Area
            then Add Index to TouchList.
        }
        return
      }
      if Node = Tree then {
        if AreaLeft touches Area then FindGeometry Area NodeLeft Layer
        if AreaRight touches Area then FindGeometry Area NodeRight Layer
        return
      }
    end
```

Figure 7–12: Search Algorithm.

The algorithm used to search for geometry within the MS binary tree is given in figure 7–12. The algorithm is recursive, the initial entry arguments are,

- **Area** The area to be searched. This is usually an area defined by layout geometry and the design rules being tested.
- **BinTree** The root of the MS binary tree.
- **Layer** The geometry layer on which the search is conducted.

If the Bintree union is defined as a Tree node, that is it contains pointers to two other Tree nodes, and if the Area to be searched is in the Right Area (figure 7–10)

the search procedure (figure 7-12) is recursively called with NodeRight assigned to the Node argument. Since the seearch Area may overlap the Left Area as well as the Right Area, when the recursive call returns, a similar call is made using NodeLeft if Area overlaps or touches AreaLeft.

If the structure Node is a Data node, that is a leaf cell that holds an array of indices, all the layout geometry reference by indices in the array are tested. If the area occupied by the geometry overlaps or touches the test Area, the index is added to a list of touched boxes.

When this routine returns from the initial call the results of the search are contained in a list. This list may contain zero elements in which case no geometry was found or it may contain a large number of references. Some of these references may be duplicated where layout geometry has been stored in more than one leaf cell of the MS binary tree.

7.7 Conclusions

The design rule checking required to verify that layout modifications are possible, and that they do not violate any of the design rules is the most CPU intensive part of the LocDes program. In order for the program to be of practical use this must be done efficiently. The algorithms that have presented in this chapter to apply design rule checks, after layout has been modified, use an adaptive multiple storage binary tree data structure to efficiently perform the many local region searches that are required to perform the design rule checking. The data structure was optimised to meet the requirements of the region search and tree construction performed by the design rule checking algorithms.

Chapter 8

Results

The LocDes program was used to generate new layout for two classes of layout, routing and random logic. This chapter presents some simulation results from the application of contact, track width and displacement LDRs to this layout.

8.1 Application of LDRs to Routing.

In many ICs the routing between circuit blocks takes a significant proportion of the overall circuit area. The exact proportion varies from chip to chip, but is often quoted as being as much as 80%[76, 88] of chip area. At least one metal layer is normally used for routing. This is to provide low resistance connections. In many modern CMOS processes there are two or more metal layers available, that are used extensively to route power, ground and signals about the chip.

Metallisation can be one of the greatest sources of yield loss in IC production[78]. Since, routing covers such a large fraction of the chip area and makes extensive use of the metal layers, it follows that a significant fraction of yield loss occurs as a direct result of defects causing faults in the routing.

Local design rules can be used to reduce this yield loss by reducing the layout's sensitivity to defects. A small routing example is shown in figure 8–1; this is part of a larger layout produced by SOLO 1400[89] to ES2 2 micron design rules. A

small example is used here because of the time required to compute the change in yield of larger layouts, though LDRs can be applied to layouts of any size.

8.1.1 Track Displacement LDR

Track displacement LDRs allowing up to 6 displacements were applied to the routing on both metal layers, The resulting layout is shown in figure 8–2. [1]

Effects on Metal 1

A Monte Carlo yield simulation of metal 1 extra material defects over a range of defect sizes was performed on the original and transformed layouts. The effect of track displacement can been seen from the difference between the two simulations, given in figure 8–3. This shows that the number of faults caused by smaller metal 1 defects is reduced, and that there is a corresponding increase in faults from larger defects. These results are typical of LDR displacement, an explanation of this behaviour is given in section 5.3.2.

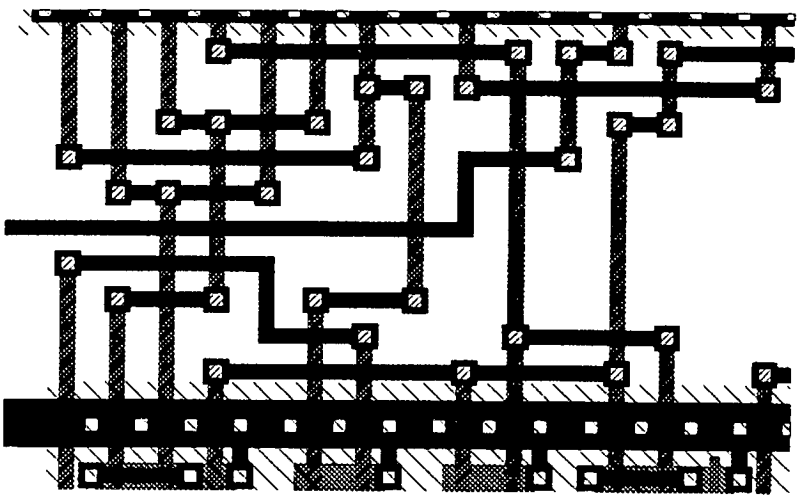


Figure 8–1: Routing Example Original Layout.

[1] In figures 8–1 to 8–5 metal 1 runs horizontally and metal 2 vertically.

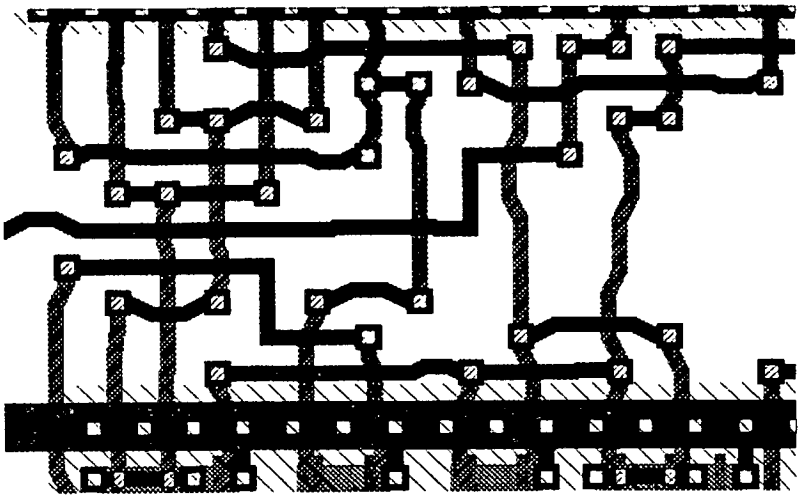


Figure 8–2: Routing Example After Displacement LDRs.

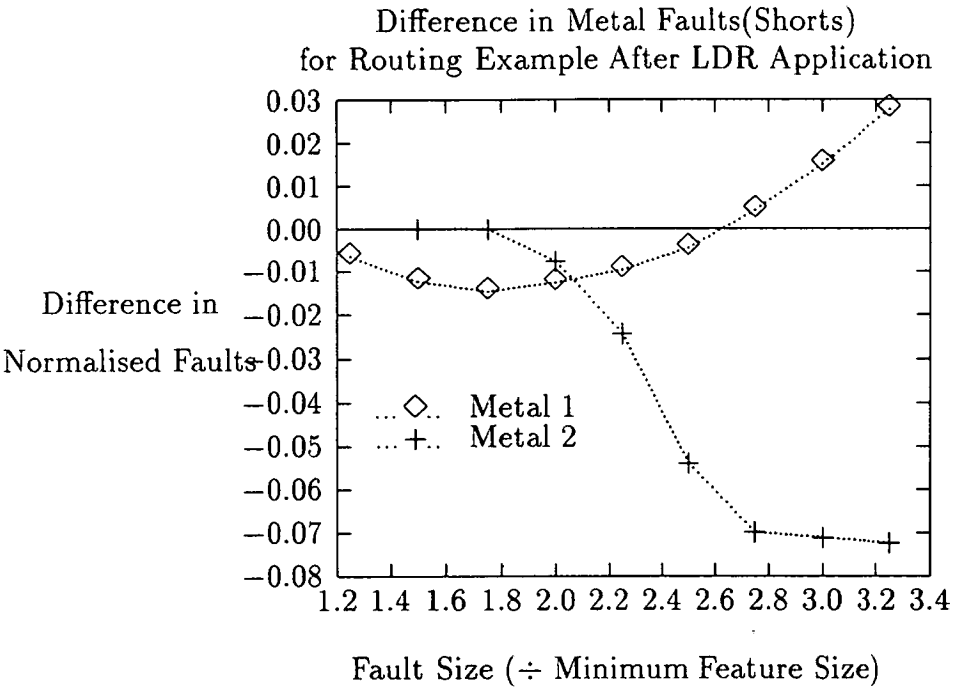


Figure 8–3: Routing Example Results: Difference in Normalised Faults for Metal Layers with Displacement LDRs

Effects on Metal 2

A similar simulation was performed for metal 2; this gives results that differ from those found for metal 1. The incidence of faults that are less than 1.8 times the minimum fault causing defect size remains unchanged. This is due to the fact that the only metal 2 geometry that can be shorted by these small defects takes the form of contact overlap. This geometry is "fixed" and cannot be displaced by the LDR. As a consequence the number of faults generated by the smaller defects is unaffected by the displacement LDR. However the larger defects, greater than 1.8 times the minimum defect size, do show a reduction in the number of faults that they generate.

Difference in Fault Size Distribution

The results for faults caused by missing material defects (opens), for both metal layers were effectively unchanged within the accuracy of the simulation. This indicates that these layout changes could be made for any fabrication process with an associated increase in yield or at least no change in yield. The magnitude of this yield increase in the metallisation layers as a result of the LDR application is dependent on the number of defects and their size distribution. If we assume this distribution to be proportional to $\frac{1}{Defect\ Size^3}$[4] and apply this as a weighting factor to the simulation results, the fault size distribution for the routing example is generated. Figure 8–4 shows the difference in the normalised fault size distribution of metal 1 and 2 as a result of the displacement LDR. The change in yield as a result of the track displacement LDR is proportional to the area under this curve; a yield improvement is a negative area.

8.1.2 Contact/Via Increase LDR

Any non-trivial routing problem involves signals crossing the path of other signals. This requires the use of more than one routing layer and the means to transfer

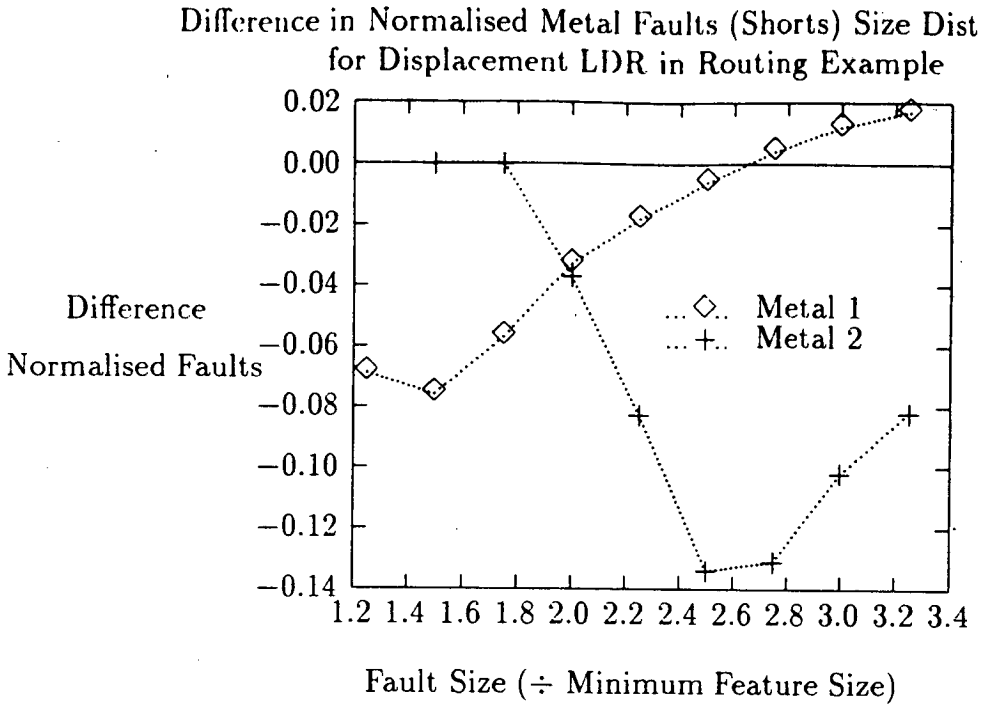


Figure 8–4: Routing Example Results: Difference in Fault Size Distribution of Metal Layers After Displacement LDRs

signals to different routing layers. In a metal 1 – metal 2 routing strategy vias[2] are used to accomplish the task. These vias are subject to yield loss and also increase interconnect resistance, so many routing programs[90, 76] attempt to minimise the use of vias to improve both the yield and performance. The yield of those vias that remain may be improved in some fabrication processes by applying a via increase LDR.

A Via increase LDR was applied to the circuit of figure 8–1. The resulting layout after 6 iterations of the LDR is shown in figure 8–5.

Effects on Metal 1

The effect of this layout change on the number of extra material defects causing faults in the metal 1 layer is given in figure 8–6. This graph shows, for comparison,

[2]Metal 1 to metal 2 contacts are often called vias.

the previously generated results for track displacement and also results generated from a layout, shown in figure 8–7, that has had both track displacement and contact increase LDRs applied to it.

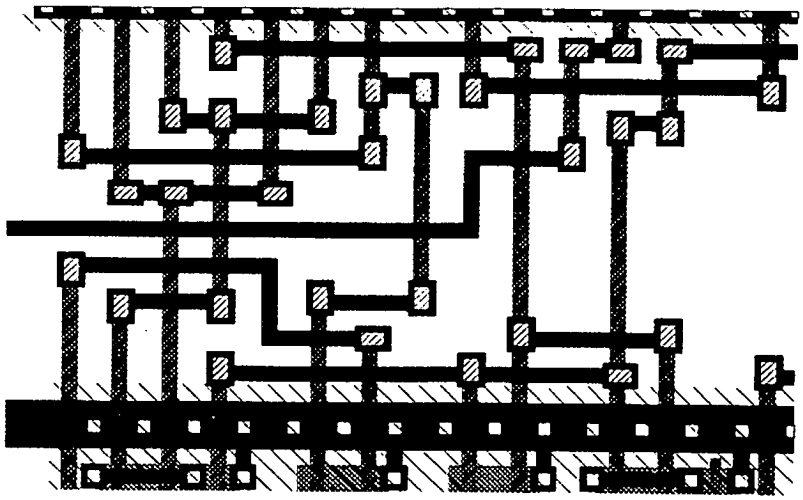


Figure 8–5: Routing Example After Via LDRs.

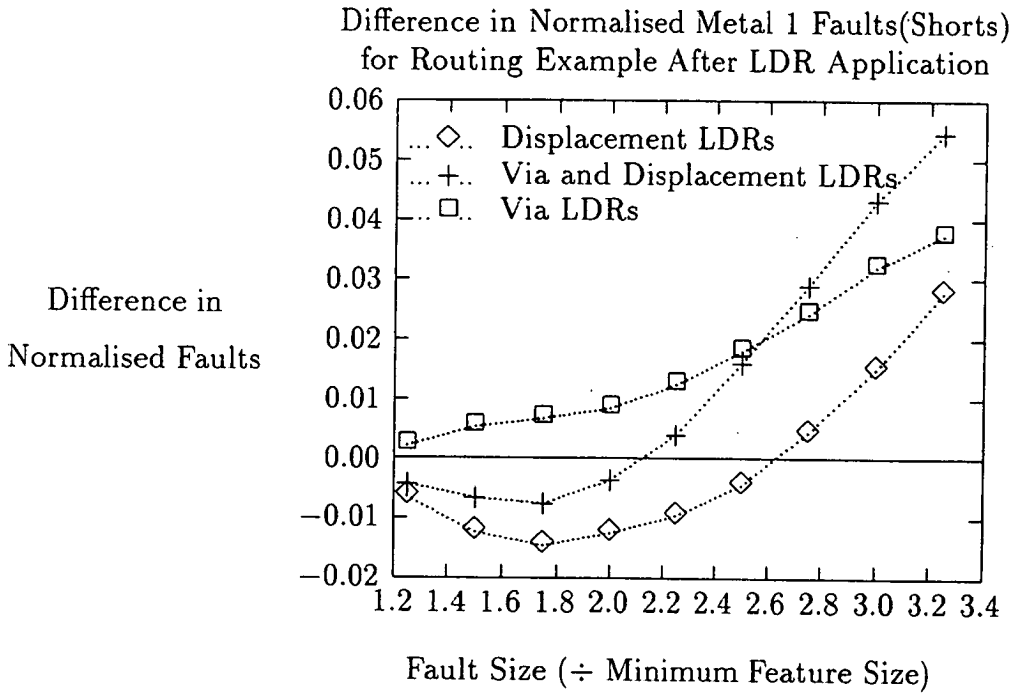


Figure 8–6: Routing Example Results: Metal 1 Layer After Via and Displacement LDRs

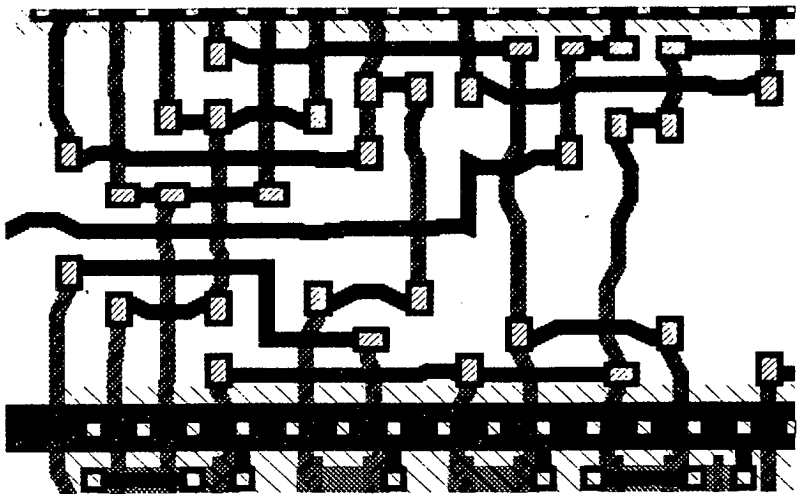


Figure 8–7: Routing Example After Displacement and Via LDRs.

The simulation results for 6 iterations of via increase LDRs to the routing example show that the number of extra material faults increases at all fault sizes. This is due to the larger contact overlap that increases the probability of a defect causing a fault between this overlap and neighbouring geometry. Where displacement and contact LDRs are combined on the same layout the gains from the track displacement offset the contact overlap increase losses. This can be seen more clearly where a defect size distribution ($\frac{1}{Defect\ Size^3}$) is applied to the results to give the fault size distribution. The change in this distribution is shown in figure 8–8. The area under the curve is proportional to the yield change to the metal 1 layer as a result of the LDR application. It can be seen that the yield of the metal 1 layer is reduced by the contact LDR.

Effects on Metal 2

Figure 8–9 gives similar results for the second layer of metal. These results are worse than those produced for the metal 1 layer as can be seen from a direct comparison of the change in fault size distribution for both metal layers shown in figure 8–10 and figure 8–8.

These results are significantly poorer than the metal 1 results because most metal 2 is distant from neighbouring metal 2, by at least 1.8 of the minimum

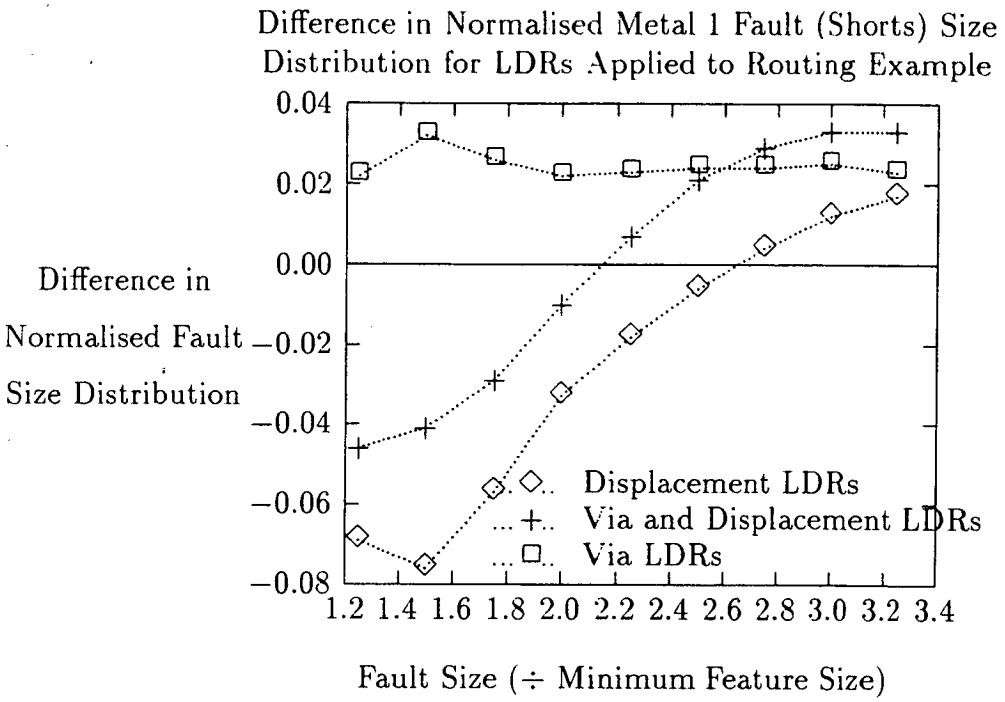


Figure 8–8: Routing Example Results: Fault Size Distribution Metal 1 Layer After Via and Displacement LDRs

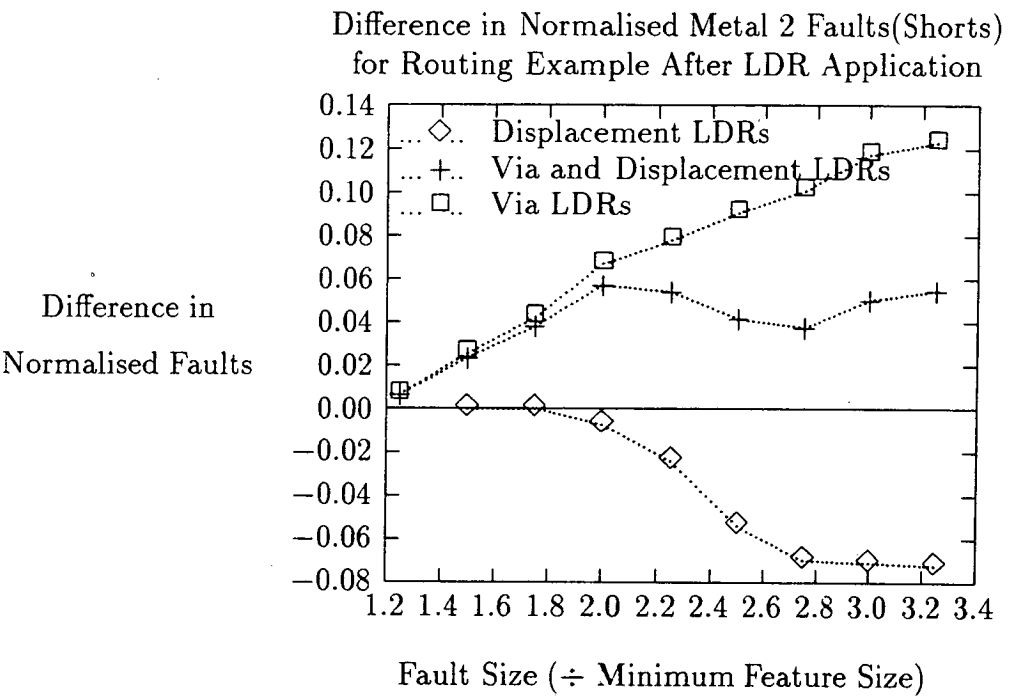


Figure 8–9: Routing Example Results: Metal 2 Layer After Via and Displacement LDRs

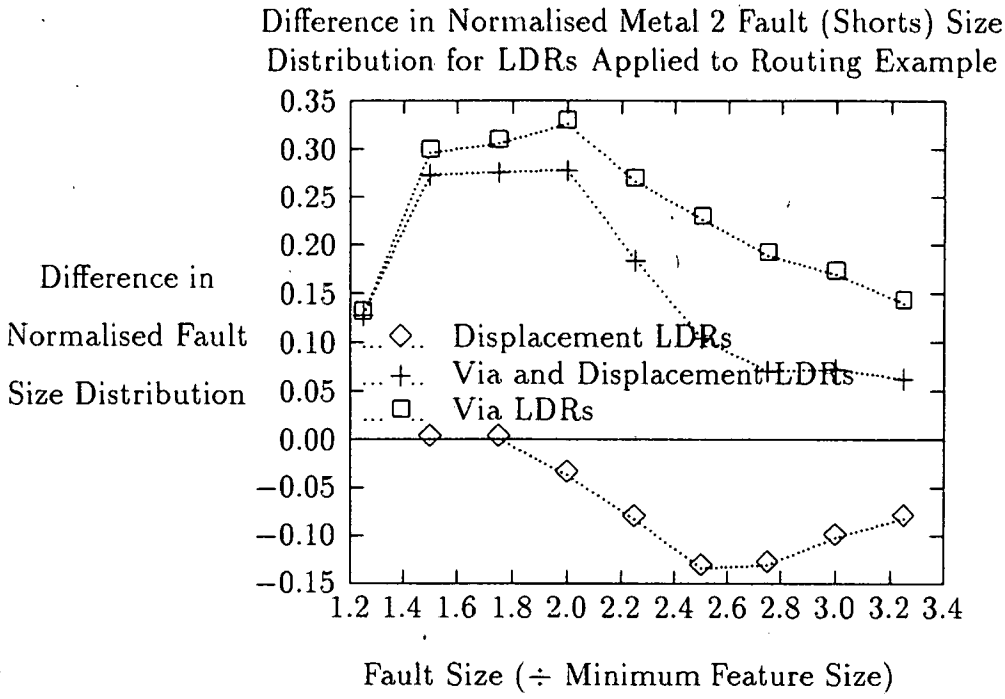


Figure 8–10: Routing Example Results: Fault Size Distribution Metal 2 Layer After Via and Displacement LDRs

separation, except for contact overlaps. These overlaps have been increased to accommodate the new larger vias causing the amount of near minimum separated metal 2 to be greatly increased. This in turn increases the number of small defects that result in faults. The increased susceptibility to small defects has a marked effect on the total number of faults generated because of the relatively large number of small defects present in a defect distribution of the form normally found in a fabrication process.

It is of course possible to use fewer iterations of the contact increase LDR in which case the results for extra material defects on the metal layers are slightly better. The graphs in figures 8–11 and 8–12 give results for 1, 2 and 3 iterations of the LDR which compare well with the results of 6 iterations.

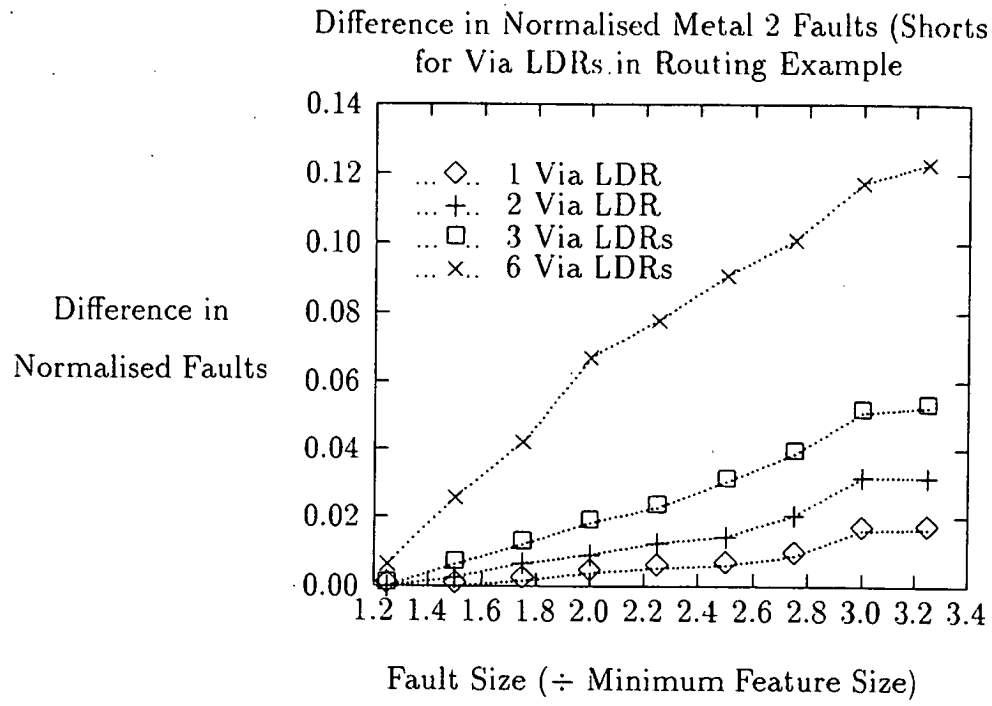


Figure 8–11: Routing Example Results: Metal 2 Layer After Contact LDRs

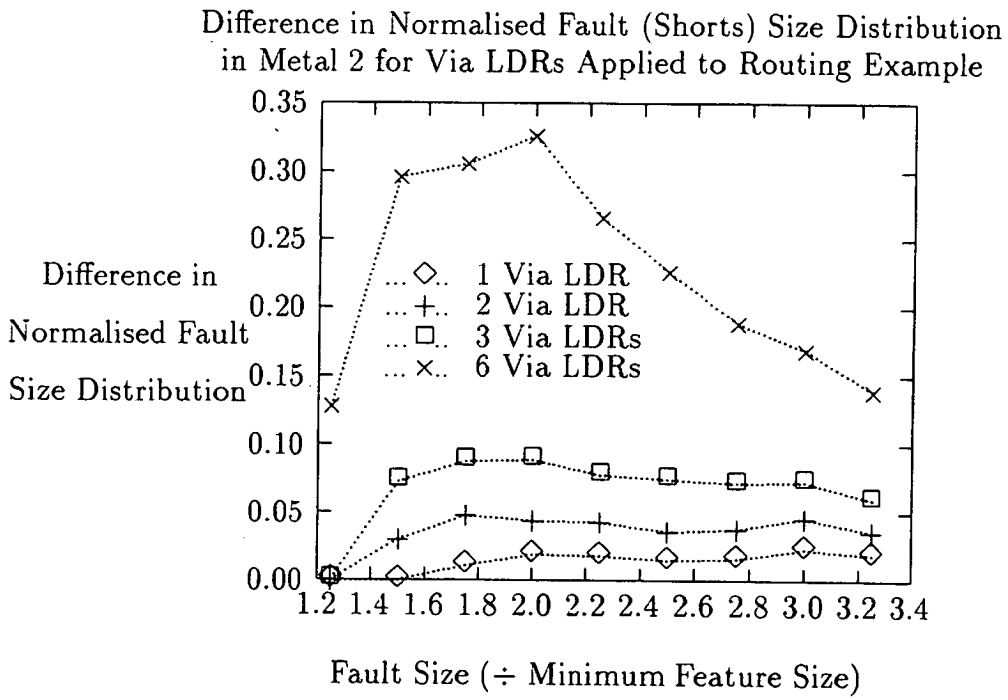


Figure 8–12: Routing Example Results: Fault Size Distribution Metal 2 Layer After Contact LDRs

Effects on Via Layer

If changes in the metal layers were the only effect of contact increase LDRs, they would be of no practical use. However the yield of contacts can sometimes be improved sufficiently, by increasing their size, to more than offset the yield loss caused by the larger overlap. This is of course process dependent.

The improvement in the yield of the larger vias can be simulated in a similar way to the metal layers[3]. The routing example layout was simulated before and after the contact LDRs were applied. The difference between the original layout and layout produced after 1, 2, 3, and 6 iterations of the contact LDR are shown in figure 8–13. These results in comparison with the results for metal 1 and 2 presented earlier indicate a significantly greater reduction in the number of faults. This is due to the relatively large changes in contact size that occur as a result of the contact LDR. By applying a defect distribution to these results the change in the fault size distribution is obtained, this is shown in figure 8–14.

Percentage Change in Faults

The results of the application of LDRs to the routing example can also be expressed as a change in the number of faults. The faults found after the LDRs were applied, expressed as a percentage of the original faults for the layer, are given in table 8–1. The displacement LDR on the metal 1 layer has resulted in a reduction of approximately 3% of the original faults and the same rule applied to the metal 2 layer has resulted in a greater reduction (approximately 11%).

While the metal 2 shows a greater relative decrease in the number of faults, metal 1 has a greater absolute reduction in faults from the application of the displacement LDR. This is because the sensitivity of this layout to metal 1 faults is greater than to metal 2 faults. This can be seen from figure 8–15 that shows

---

[3]A fault occurs where a defect completely covers the via.

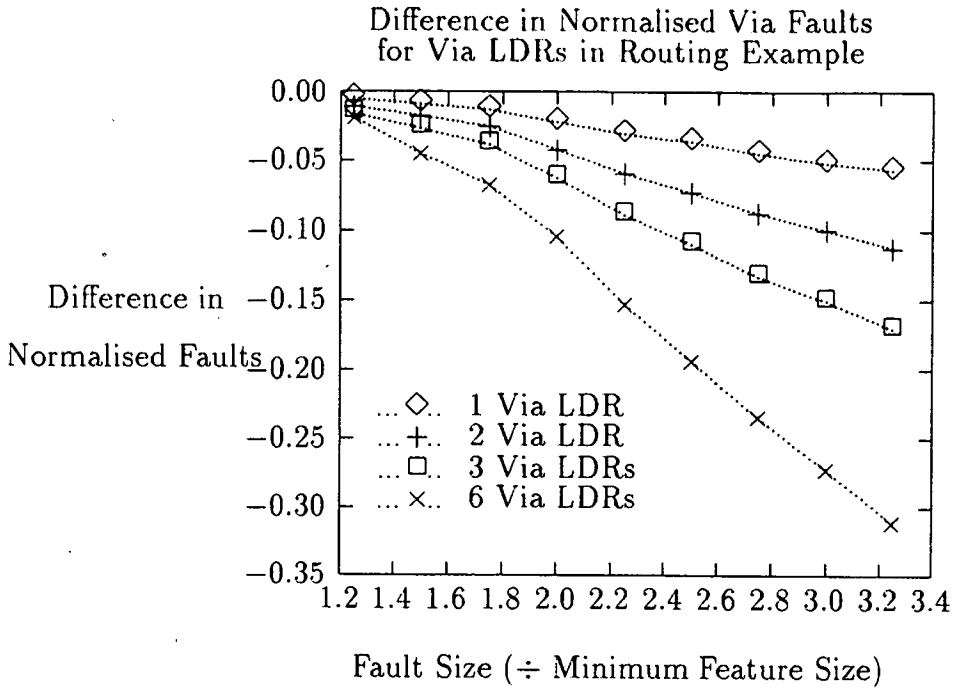


Figure 8–13: Routing Example Results: Via Faults After Via Increase LDR Application

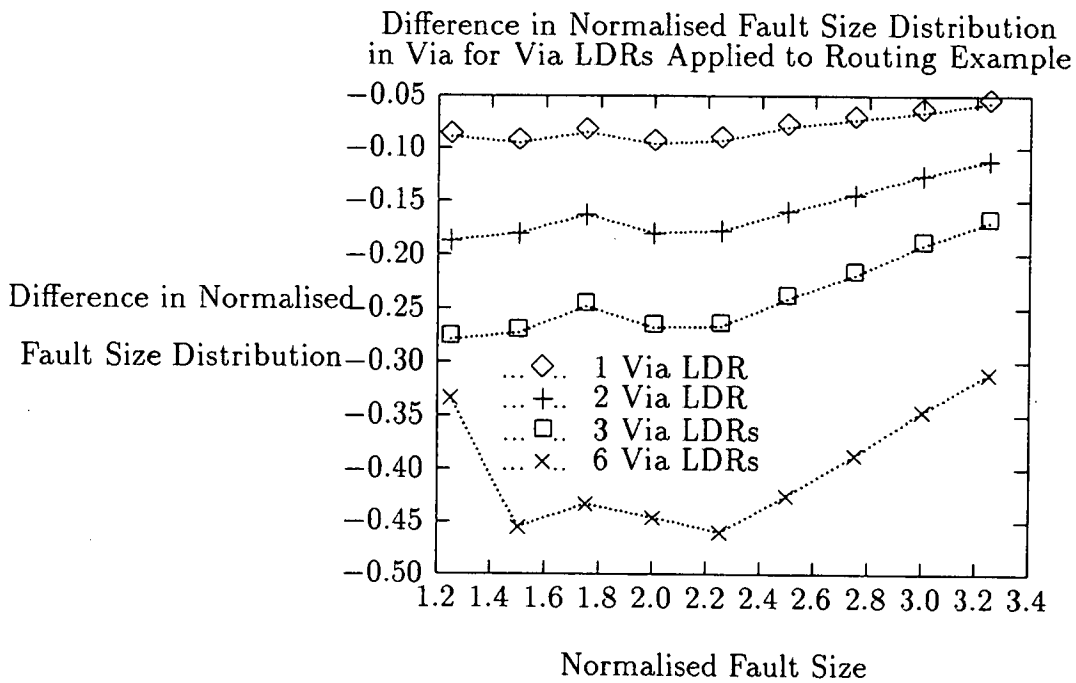


Figure 8–14: Routing Example Results: Fault Size Distribution for Via Faults After Via LDR Application

the fault size distribution of metal 1 and metal 2 faults from simulation results where identical defect size distributions were applied to both layers of the routing examples. The defect sensitivity is proportional to the area under the curve and in this case metal 1 is 5 times more sensitive to defects than metal 2. This implies that for identical defect distributions in both metal layers, a 3% reduction in metal 1 faults is equivalent to a 15% reduction in metal 2 faults.

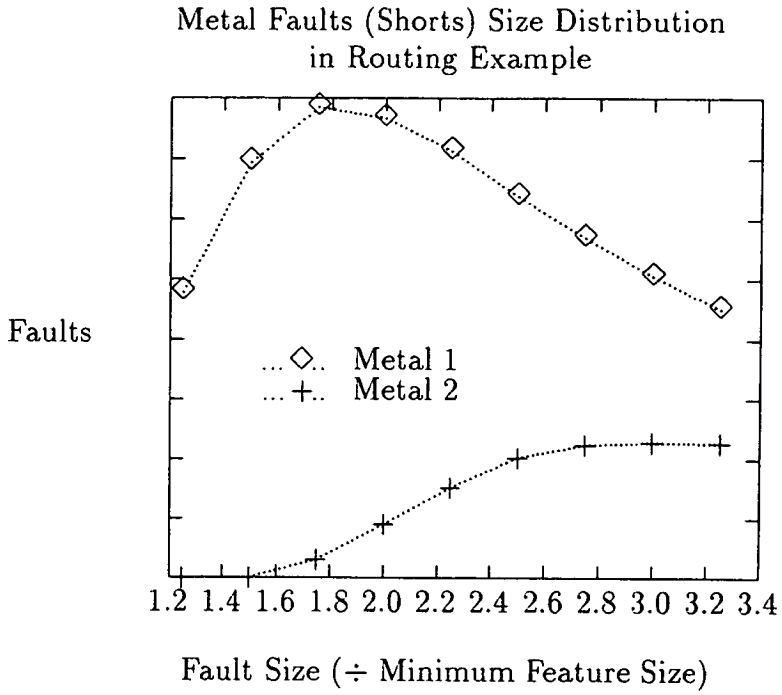


Figure 8–15: Routing Example Results: Fault Size Distribution of Metal Layers

8.2 Application of LDRs to Layout Cells

While routing is an important part of an IC layout the cells that are connected by this routing are equally important. These cells can be divided into a spectrum stretching from very regular layout like RAM to irregular layout such as random logic. Most cell layouts fall somewhere between these two extremes. Regular layout such as RAM provides little scope for LDR application since often a sub-cell is laid out with great care and then replicated in a string or array of sub-cells.

Faults After LDR Application As a Percentage of Original Layer Faults		
Results for Metal 1		
LDRs	Iterations	Faults (%)
Displacement	6	96.9
Displacement and Via Increase	6 6	100.1
Via Increase	6	103.1
Results for Metal 2		
LDRs	Iterations	Faults (%)
Displacement	6	88.8
Displacement and Via Increase	6 6	128.6
Via Increase	1	102.4
Via Increase	2	106.3
Via Increase	3	111.9
Via Increase	6	140.3
Results for Via		
LDRs	Iterations	Faults (%)
Via Increase	1	89.4
Via Increase	2	79.3
Via Increase	3	68.8
Via Increase	6	48.0

Table 8–1: Routing Example: Percentage Change in Faults from LDR Application

Irregular structures are in general more suitable for use with LDRs, particularly where an automatic layout tool has been used to generate the layout.

8.2.1 Track Displacement LDR

A track displacement LDR was applied to the metal layer of the cell given in figure 8–16, which is part of an adder circuit , generated using the STICKS [69] circuit compacter. Layout for 1, 2, ..., 6 iterations of track displacement was produced using the LocDes program. The resulting layout for 6 iterations is given in figure 8–17. All six layouts were simulated using the Monte Carlo simulator, the results of these simulations are given in figure 8–18, as the fault size versus the percentage change in fault occurrence.

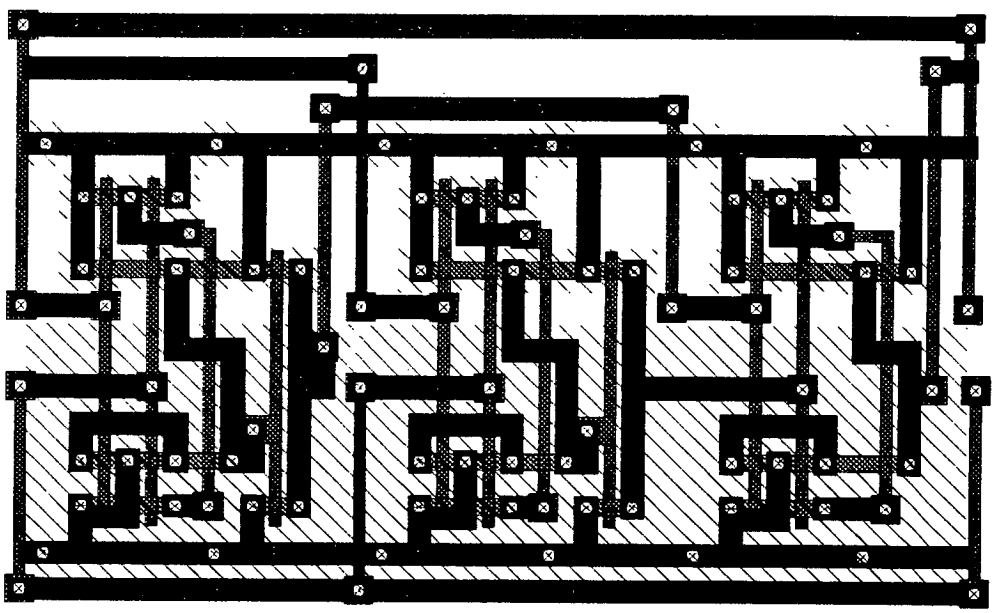


Figure 8–16: Random Logic Cell

The results of the simulations combined with a defect size distribution for shorts of $\frac{1}{(Defect\ Size)^3}$ [4] give a measure of the reduction in all metal 1 short faults for a cell with track displacement LDRs; this is shown in figure 8–19. These results indicated that the number of faults per unit area can be reduced

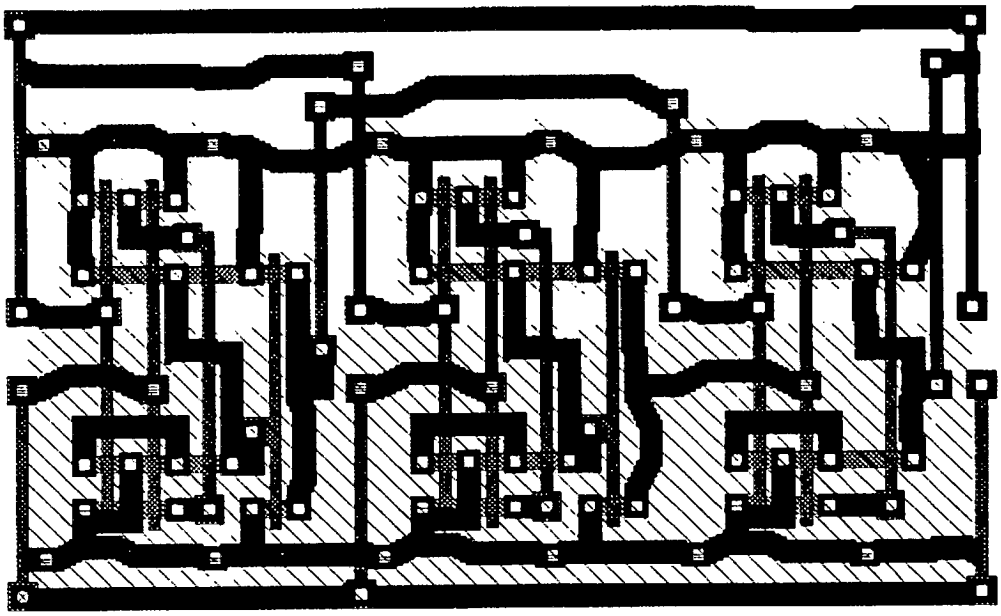


Figure 8–17: Random Logic Cell After 6 Track Displacement LDR Iterations

to less than 90% of the original value by using more than two iterations of the track displacement LDR on the layout of figure 8–16.

8.2.2 Change in Track Width

A track width increase LDR was applied to the layout of figure 8–16; the resulting layout is given in figure 8–20 The rule can be expressed as: Metal 1 is increased up to 1.3× minimum metal 1 width so long as 1.5× minimum separation is kept between unrelated metal 1 and no GDRs, involving metal 1 with any layer, are violated.

The results of a Monte Carlo simulation of the metal layer for both extra material defects (shorts) and missing material defects (breaks), showing the difference in the normalised faults as a result of the LDR are given in figure 8–21. The graph indicates that the probability of a spot defect causing a break can be reduced by applying this LDR, and the probability of shorts being formed by a defect is increased by a lesser amount. The data combined with the process defect size and distribution for metal 1, is used to determine if the LDR circuit

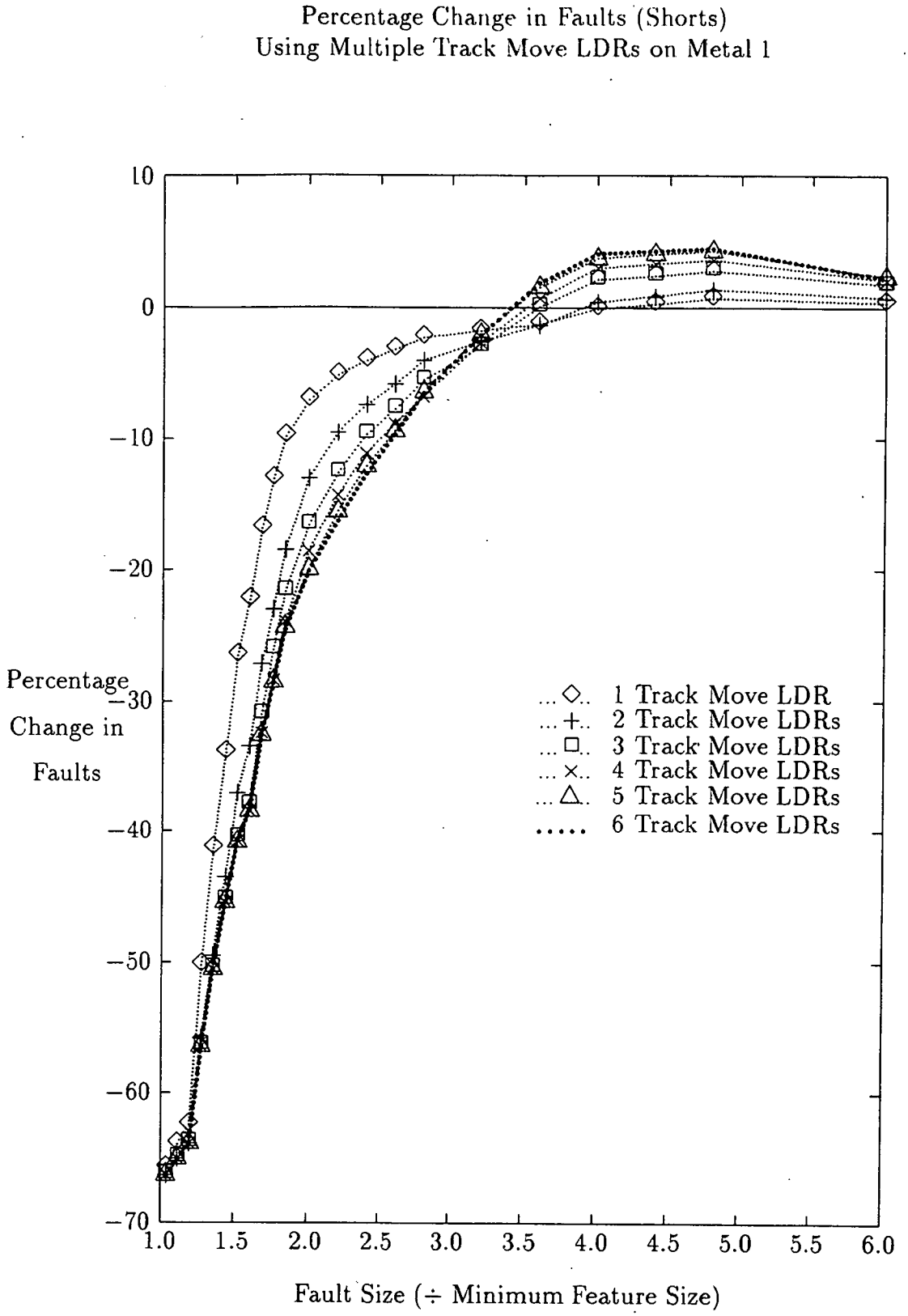


Figure 8–18: Percentage Change in Defects from Track Move LDRs

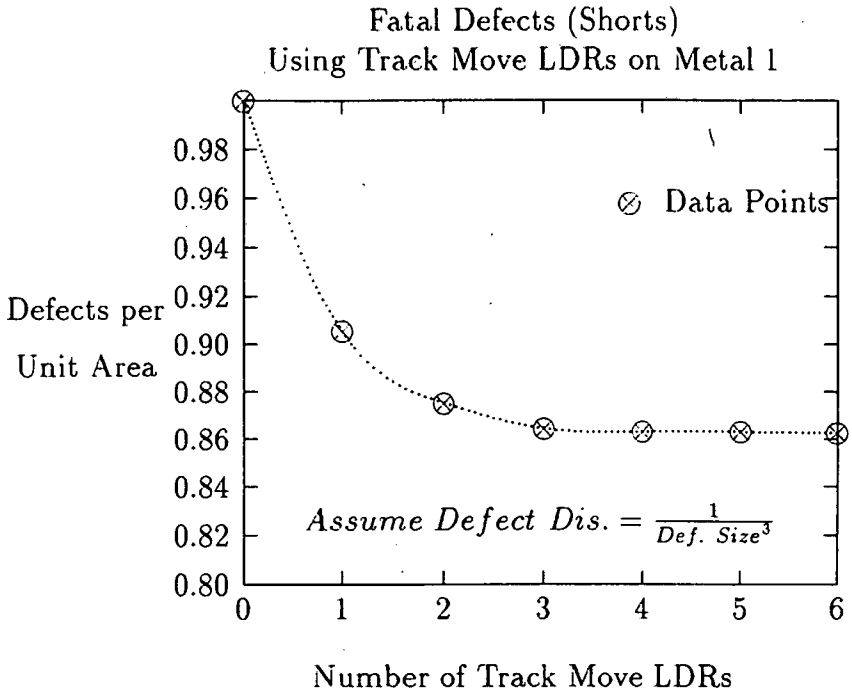


Figure 8–19: All Short Fatal Defects Using Track Displacement LDRs

has a higher yield when fabricated with that process. For example, where the defect size distributions of missing and extra material breaks are the same and proportional to $\frac{1}{Defect\ Size^3}$, the difference in the normalised fault size distribution is that shown in figure 8–22.

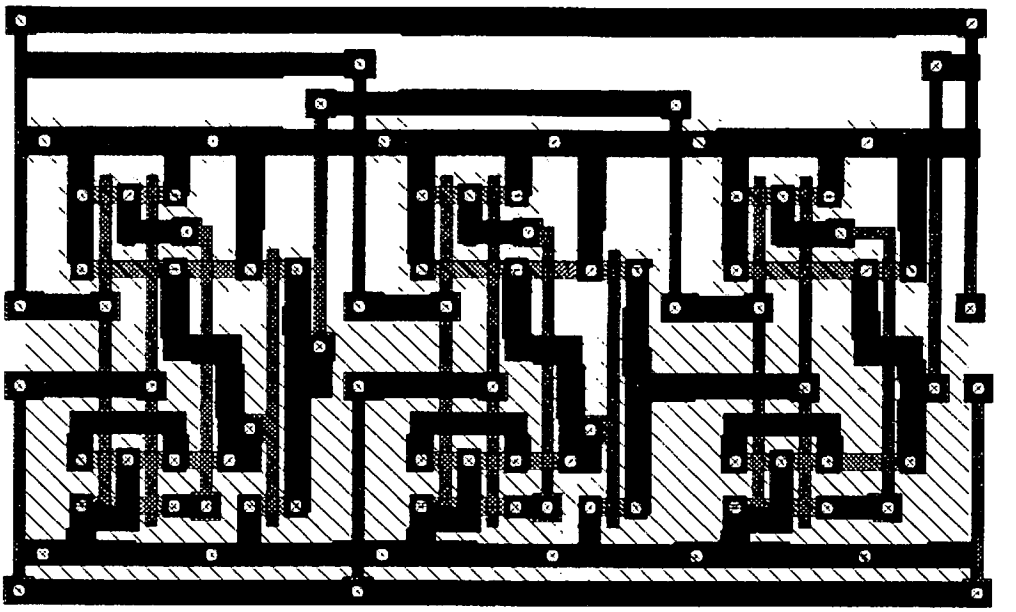


Figure 8–20: Random Logic Cell After Track Width Increase LDR

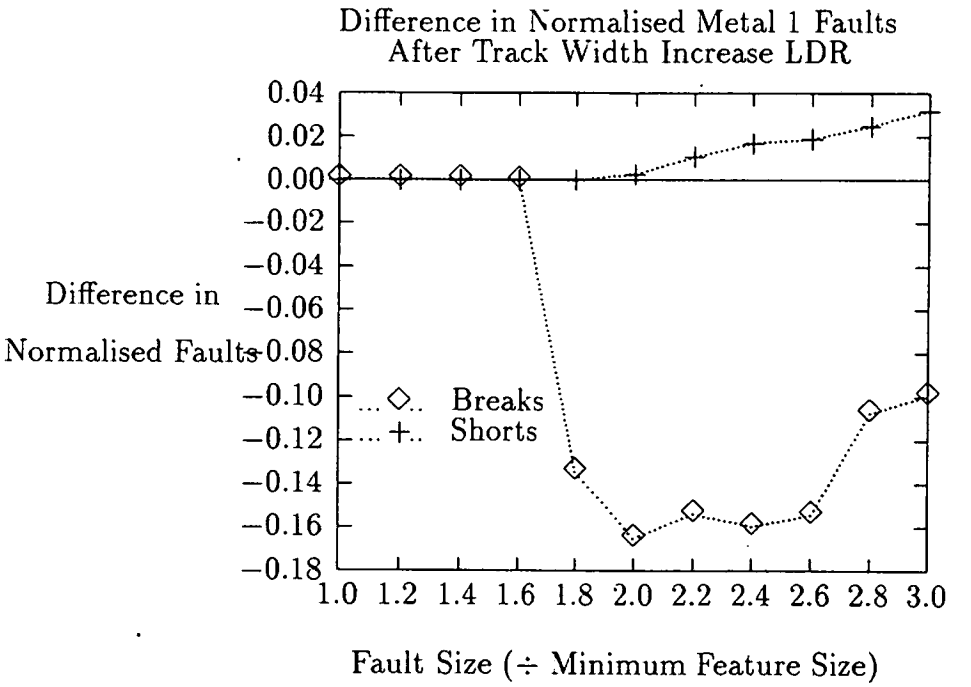


Figure 8–21: Random Logic: Difference in Normalised Faults After Track Width Increase LDR

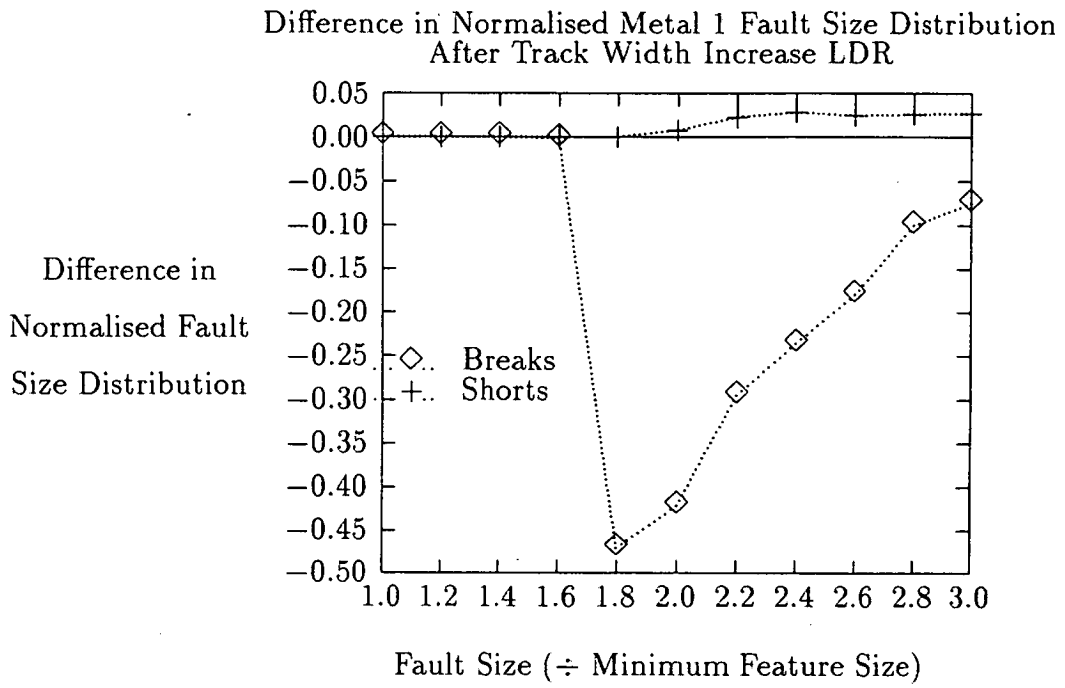


Figure 8–22: Random Logic: Difference in Normalised Fault Size Distribution After Track Width Increase LDR

8.3 Using LDRs

The results presented here suggest that LDRs have the potential to increase the defect sensitivity as well as reduce it. Great care will be required when applying LDRs that increase the defect sensitivity to one defect type and reduce the sensitivity to another type. Before such a LDR is applied to a device the ratio of the two or more fault mechanisms must known, or at least an estimate must be available. Unless it is very clear that a LDR will result in a reduction of the overall defect sensitivity, (i.e., well outside the error margin of the defect estimates) the application of a LDR is not recommended. It is of course often possible to adjust a LDR that does not meet this requirement to be more strict so that it can be applied, but in fewer instances. For example the contact LDRs that were applied to the layouts in this chapter used the GDR separations to define acceptable contact overlap separation from neighbouring geometry. By defining a more strict LDR that required 1.5× the minimum separation the increase in sensitivity to metal extra material defects would not be so great, but the number of instances in which the rule can be applied will also be reduced.

8.4 Conclusions

The results of applying LDRs using the LocDes program have been analysed using the Monte Carlo yield simulator presented in chapter 3. These results indicate that it is indeed possible to reduce the defect sensitivity of layout using small layout modifications. The clearest results are those for displacement of tracks, since the slight increase in track length is offset by the increase in track width at the point of displacement, so that there is no measurable change in the susceptible of the layout to track breaks. Consequently the reduction of the sensitivity to shorting defects will result in yield improvements in any process where this defect type is present.

The results for track width increase and contact/via size increase indicate that they can reduce the defect sensitivity to particular defect types, but they can also increase susceptibility to other defects. These LDRs can only be successfully used where the relative distribution of defect types is such that the overall defect sensitivity is reduced after the application of track width or contact size LDRs.

Chapter 9

Discussion and Conclusions

This chapter summarises the work of this thesis. The usefulness of a tool such as LocDes in the IC design process is discussed. Some future improvements and further developments of layout modification to improve circuit yield are explored.

9.1 Introduction

This thesis has presented a technique for improving the yield of integrated circuits by modifying the circuit layout to make better use of the silicon area. The rules that govern the layout modifications have been called Local Design Rules(LDR). This work includes the implementation of a program, LocDes, that automatically applies a pre-defined set of LDRs to circuit layout. The LDRs can also be applied individually to selected circuit components using a X windows version of the program. Large layouts can be processed in parallel by LocDes.

9.2 Local Design Rules

LDRs are complementary to the normal layout design rules, which in this thesis have been termed Global Design Rules(GDR). The application of a LDR must not result in the violation of any GDR and any layout modification suggested by a LDR that violates any GDR is rejected. Another restriction on their use is that layout changes should not increase circuit area. This enables them to be applied to parts (cells) of an existing design without the need to modify the whole design.

Three types of LDR have been presented,

- **Contact Increase**

  This LDR increases the size of contact structures, that is the contact cut itself and any associated overlap layers. It is also possible to increase the overlap without changing the contact cut layer.

- **Track Width**

  The width of geometry forming tracks is increased in width by this LDR, the effect of which is to reduce the susceptibility to track breaks.

- **Track Displacement**

  This LDR moves geometry forming tracks away from close neighbouring tracks to reduce the probability of shorts between then.

9.2.1 Performance

Layout changes can have consequences for the circuit performance. Since LDR layout modifications are small any corresponding performance degradation is also likely to be small, at least in a digital circuit. Analog circuits can be much more sensitive to changes in capacitance and resistance. If even a small change in performance is unacceptable it can be avoided by ensuring that those LDRs that impair performance, are not applied on the critical path within the circuit. This

will of course require that the critical path be known, to allow a strategy of LDR application to be adopted. The only way LDRs can be applied in this case, at present, is to manually select the geometry to which LDRs are applied. This, in some circumstances, may not prove too difficult since the critical path may be contained in a limited number of cells, e.g. ALU in a microprocessor. The LDRs can be applied to a layout with the critical cells removed; these cells can then be re-inserted into the design when the LDR procedure has been completed. Where the critical path is less localised, the geometry to which LDRs are applied must be individually selected. This can be achieved using the X interface to LocDes. This procedure could prove to be time consuming and therefore may not be justified for devices that are intended for low to medium volumes of production.

9.2.2 Reliability

Equally important is the impact of layout modifications on the reliability of the circuit. This should be taken into account when LDRs are being formulated. In general larger contacts and wider tracks will improve reliability, as the rate of electromigration will be reduced since it is proportional to current density[91]. Displaced tracks could suffer electromigration problems if there was significant current crowding at the point of displacement. Since GDRs must always be maintained where any LDR is applied, displacements must be within the specifications defined by the GDRs. The GDRs will normally be expressed so as to avoid the possibility of dangerously high current densities. Consequently displacements that do not violate the GDRs should not result in any increased risk, assuming that the specifications as defined by the GDRs are an accurate reflection of the fabrication process.

9.3 LDRs and the Fabrication Process

LDRs are only of use where there is yield loss by a mechanism that can be affected by layout manipulation. Many sources of yield loss are independent of the layout e.g., processing faults as a result of operator error or equipment malfunction. Also yield loss from very large defects and surface scratches cannot be affected by the small layout modifications made by LDRs.

LDRs can reduce yield loss from small random spot defects. As discussed in chapter 2 spot defects can account for a significant percentage of total yield loss (60-80%[11, 15, 16]). While defect densities are continually improving, the trend to smaller physical size of circuit elements and larger circuit area will ensure spot defects will continue to cause a sizable proportion of total yield loss in the foreseeable future.

It should also be remembered that not all fabrication processes are state of the art and that individual processes can have unique sources of yield loss. This may make LDRs such as those for contact/via increase useful in some processes and not in others. Also the use of LDRs may make it possible to start useful production earlier in the life a new fabrication process. Where a new process has a very low yield, tailoring the layout so that it is optimal for the particular process problems at a given stage in the process development could increase the number of useful devices fabricated.

9.4 Suitable Circuits

The concept of LDRs arose because it was observed that layouts, particularly those produced by automated layout systems, were far from optimal. It was observed that in a typical layout it was possible to have larger contacts and wider wires in some areas of the layout. However, if all the contact and wires were

increased uniformly the circuit area would also increase. By restricting changes to those circuit elements that can be adjusted without the need to reposition neighbouring circuit elements the circuit area is kept constant but has a higher yield, as the layout modifications are used to decrease the defect sensitivity.

It follows from this that LDRs will be of most use where there are a large number of instances, in a circuit layout, of local non-optimal layout. That is, where there is more space than the minimum separation specified by the GDRs. This extra space can be used to modify the surrounding geometry in such a way as to reduce the defect sensitivity and hence increase the yield. While in general these conditions arise most often in layout generated by automated methods, there are many situations in which layout generated by hand is suitable for LDRs. This is particularly true of routing where nearly every contact/via can be increased in one direction, and usually two, by expanding along the direction of the track. There is also usually scope to apply track displacement and/or width increases.

It is important that the redundant space is distributed evenly within the circuit layout. Layout that is composed of tightly compacted circuit blocks in a sea of empty silicon cannot make full use of the available space. A completely different strategy is required to deal with this type of "available" space; this is discussed further in section 9.10.

9.5 The LocDes Program

The task of applying LDRs to a circuit layout is not well suited to a human designer. Making a large number of small layout modification within strict rules, both LDR and GDR, would be a tedious process prone to error. It is an operation best done by a computer.

This work involved the development of a computer program that is capable of increasing the yield of integrated circuits by automatically manipulating the circuit layout using a pre-defined set of local design rules. LocDes can perform

layout modifications in a few minutes that would taken many hours to do manually. The layout changes that can be performed by the program are limited to those already mentioned namely;

- Track displacement
- Track width increases
- Contact/Via size and overlap increases

The program can be run using a X windows graphics user interface that allows LDRs to be applied to individual pieces of circuit geometry selected by a mouse click. There is also a batch version of the program that applies LDRs in a preselected order to the whole of the input circuit layout. The batch version contains an option to run the program in parallel on either a multiple processor mainframe (e.g. Sequent Symmetry) or on networked workstations (e.g. Sun workstations). The parallel option automatically splits the layout into a suitable number of rectangular segments and creates, using the segments, a queue of jobs to be executed. These jobs are executed on the available workstations or CPUs(one job per workstation/CPU) until the all the segments are processed, at which time, the results are merged to give the final layout.

The results of layout changes produced by LocDes have been analysed using a Monte Carlo yield simulator. This simulator was developed to measure the change in the number of track shorts and breaks as a result of applying LDRs. The results, from comparison of circuit layouts before and after LDR application, suggest that significant improvements can be made in the yield of individual process steps. For example, track displacement can result in a 3–15% reduction in the number of shorts between metal tracks. The effect of other LDRs is more difficult to determine since they involve interaction between more than one defect type. However, for suitable fabrication processes, such as those that suffer from poor contact yield and/or track breaks, yield improvements would be expected.

9.5.1 Using LocDes

The LocDes program can be used to increase the yield of integrated circuits by adjusting the layout to reduce the defect sensitivity. Only layouts that have been crafted with great care will have no scope for yield improvement e.g. RAM cells. However, the program should not be used on all layouts since the yield improvement that can be achieved may have little impact on the cost of the final product. Therefore, the cost of applying LDRs cannot always be justified.

The software is most suited to large area digital ICs where large volume production is expected and yield loss is primarily, or extensively, caused by metallisation and interconnect faults, such as shorts, breaks and/or bad contact/via yield.

The LocDes program can be applied to circuit layout as a final operation prior to normal mask making procedures[1]. This has the advantage that it is a one time operation that can be effectively hidden from the circuit designer. The disadvantage of making layout changes at such a late stage is that performance errors may be introduced. Since the layout changes are small, only those devices that are "close" to their specification limits will require checks(circuit simulations) to determine if the layout is still acceptable. How "close" a circuit can be to the performance specifications before simulation of the changed layout needs to be undertaken is open to question, and will require further research.

An alternative to applying LocDes as a final layout adjustment is to use LocDes throughout the layout process. This would involve using LocDes to adjust the layout of modules and cells, as the design is built up. The advantage of this is that, the effects of layout changes on performance can be modelled as a "natural" part of the design process. That is, LDRs would be applied to circuit layout before

---

[1]This may involve polygon shrinking and/or bloating of selected layers, to ensure that they will be fabricated at the drawn dimension

circuit feature extraction and simulation, so that no extra performance modelling would be required.

9.5.2 Who Will Use LocDes ?

The use of LocDes in practice will be limited by the fact that circuit designers are conservative and are unlikely to want to add an additional complexity to their work. This attitude will be difficult to change, particularly, while circuit designers believe that yield is a problem for process engineers and all that is require of them is to follow the layout design rules supplied by the IC manufacturer.

This problem is due, in some measure, to the fact that IC manufacturers have traditionally concealed yield information. For example, not only are the final yields confidential, but design rules do not contain annotations detailing which layout option would give the best results or which layers have the highest yield. Circuit designers have never had sufficient information to make an informed decision on the yield of their layout.

In view of this resistance, LocDes or a similar type of program will probably only be used in two circumstances:

1. When a device has been designed that has a very poor yield. This may be either because it makes use of a new process or it is an unusually large/complex device. In such circumstances effort will be expended to increase the yield. Previously, this could only be done by improving the fabrication process. However, it is now possible, using LocDes, to change the layout and so make it less sensitive to the defects that are still present, even after improving the fabrication process.

2. The other possibility is that a decision to adopt a strategy of design for quality will be made by management and thrust onto circuit designers. Such a decision would be consistent with the trend in industry towards quality control, e.g. Motorola's 6 sigma policy[92].

In the first circumstance, of a low yield device, the cost associated with using LocDes will be small in comparison to the costs of process improvements and the potential losses that may occur if the devices cannot be manufactured at a reasonable cost in sufficient quantity. Alternatively, where the decision is made by management, as part of a general strategy to improve product quality, the cost will become part of the budgeted investment. Whether the investment would be more wisely spend on improvements in the fabrication process is dependent, not only on the process yield and sources of yield loss, but also on the type and volume of the circuits produced.

9.5.3 Introducing Yield to Circuit Designers

The cost of adopting a yield improvement system as a standard part of the design process, could be made more acceptable if it were part of the design software. Resistance to its use could be reduced by allowing designers to use it when required, much like a design rule checker. The main functions[93] of such a system would be to;

- Predict yield
- Indicate regions of poor yield
- Suggest remedies to poor layout
- Automatically adjust layout to maximise yield

This approach has the advantage that it gently introduces the circuit designer to yield improvement. By giving a yield prediction, in some arbitrary units, and indicating poorer layout regions, feedback is provided, encouraging good layout styles. At the same time full control can be maintained by the designer over how much the software is allowed to do, so that suggested layout improvements need only be accepted when required. When the designer becomes more confident with the system, the automatic layout adjustment will be used more frequently.

9.6 Future Improvements to LocDes

The main improvements to the work described here would involve re-programming to enable more flexible changes to the layout and changes to integrate the software with current design approaches.

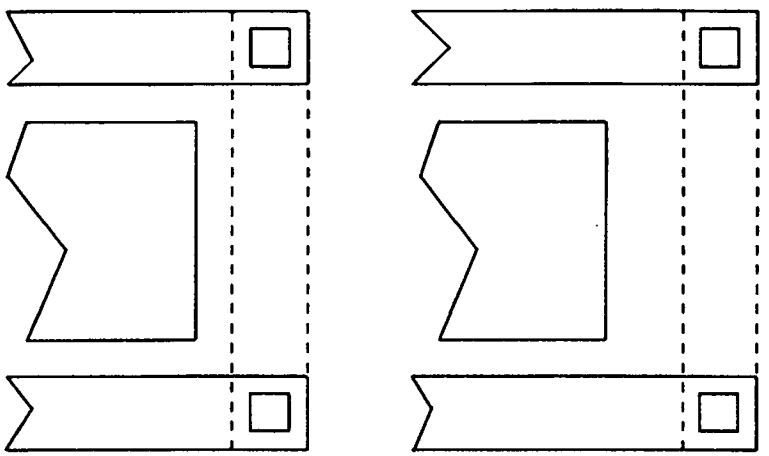
9.6.1 Great Flexibility

At present the layout is represented internally as a series of boxes. While these are each labelled with a node number, there is no attempt to recognise collections of these boxes as circuit features or objects[94]. As a consequence layout manipulations are limited to adjustments to single layout boxes (e.g. a contact increase generates new overlap geometry rather than using the existing geometry). This simplified the original implementation but greatly restricted the possible layout manipulations. Future extensions to the existing program would have to concentrate on a better representation of circuit features. This means that contacts, wires and transistors should be represented by data structures that allow circuit components to be treated as single objects rather than a collection of boxes. Redesign can then be best achieved using the methodology presented by Booch[94], which the author was unfamiliar with when this work was originally undertaken.

9.6.2 More Effective Layout Adjustment

The LDRs presented in this work have been restricted to simple geometry manipulation. This is due to the difficulty of interpreting and manipulating a circuit layout from a CIF representation. More complex LDRs that involve geometry manipulation which impacts on more than one layout box simultaneously are possible e.g., stretching layout wires in length to displace contacts and/or associated track (figure 9–1). Such LDRs are difficult to program using a CIF input. A

higher level representation, where a single symbol represents a track (and possibly contacts on the track) rather than a collection of layout boxes, would greatly ease this programming task. While it is possible to generate such a high level representation from CIF it is a demanding operation that can require human intervention[95].



(a) Original Layout

(a) More Complex Track Displacement LDR

Figure 9–1: More Complex Track Displacement LDR

It would be better to approach the problem of yield optimisation nearer the start of the design cycle rather than the end, so that a high level description of the circuit is available in a form more suitable for comprehensive layout manipulation. Such a high level description would in many instances be available, since many digital circuits are designed at a much higher level of abstraction than even transistors and wires; only later are they converted to a mask level representation, such as CIF.

9.6.3 CIF File Size

One poor feature of the LocDes program is the expansion of the CIF design file, when the hierarchy is removed. A human designer could easily determine instances where the hierarchy can be retained without restricting the use of LDRs.

The same task is difficult to accomplish automatically. It would be possible to improve the performance of the program in this regard. For example, where only track width and contact LDRs are applied, the LDRs only add geometry to the layout, they do not remove or reposition it. This makes it possible to represent the final design as a combination of one cell that contains all the LDR changes and the original hierarchical representation. This has not been implemented in the current version of LocDes mainly because track displacement LDRs cannot be represented this way, since the original geometry is repositioned. It is possible that contact LDRs may be of particular interest in some fabrication processes (see page 98) in which case a version of LocDes that had this facility would be useful.

9.6.4 Making LocDes More Usable

In order to have yield improvement of layout used in main-stream design, the software should be an integral part of the design suit. Changes would not only have to be made to the LocDes program but also the environment in which it was being used.

The major differences in the design environment would be the required support for two different types of cells; the original layout the one to which LDRs have been applied. Layout cells would not only have to be labelled as LDR version or the original, but since LDRs are sensitive to the surrounding layout, a number of different versions of the "same" cell with different LDRs applied must also co-exist. Further more, when layout is changed, automatic re-application of LDRs to the cell and the surrounding geometry should take place. This will require a method of deriving a dependency list to minimise the amount of re-computation.

There would also need to be a facility to freeze parts of a design once simulations have been successfully completed. This is to ensure that layout changes are not automatically made, that can affect the performance specifications of a layout that has already been passed as within specification.

In order to interface with such a layout system, LocDes would be required to apply LDRs intelligently to small layout fragments. These fragments can not be assumed to be rectangular cells. Consequently, LocDes would require at least two further modes of operation, since, at present LDRs are applied automatically within the rectangular boundary of a cell.

1. **GDR Envelope**

   In order that a layout cell can be directly substituted for an equivalent cell, to which LDRs have been applied, the GDR (global design rule) envelopes of the two cells should be identical. That is LDRs should not adjust geometry where the change could cause a design rule violation with adjacent, but unspecified, layout surrounding the cell. For example, metal tracks could be displaced so that they were nearer to the cell boundary. If they are moved closer, to the cell boundary, than the maximum separation of metal with any other layer, then there is a possibility that this could cause a design rule violation, not present when using the original cell. The displacement would be disallowed if this were the case.

2. **Environment Sensitive**

   Enforcing LDR application within the GDR envelope, will in some circumstances be overly restrictive, particularly with very small cells where no changes could be made. There are other circumstances in which the GDR envelope will not be strict enough, e.g. where cells have overlapping geometry. In order to deal with both these cases another mode of LDR application is required that will take into account the surroundings of the layout when applying LDRs. This mode would not be difficult to implement, since all that is required is for the cell geometry to be labelled distinctly from the surrounding geometry, so that LDRs can be applied only to it. The LocDes program would automatically ensure that there were no violations with the other, un-adjusted, geometry.

The effort needed to create a system that supports LDRs for hand-crafted

layout is considerable. The layout system would require extensive use of revision control procedures and automatic "re-compilation" where changes have been made that affect other parts of the layout. Before embarking on the production of such a system, the demand from the user community would need to be established, in order to justify the required expenditure.

9.7 Yield Simulation

While undertaking this work it has been found that yield simulation using the Monte Carlo method is very expensive in computer time. A large number of of defects over a range of defect sizes must be placed on the design and analysed. This procedure severely limits the size of layout that can be simulated in a reasonable time.

There are a number of more efficient methods of yield prediction than the Monte Carlo method. One of the simplest methods that could be used to generate a reasonably efficient and accurate yield predictor is the virtual layout method[44], where a simplified layout is generated to represent the circuit. The virtual layout is generated using polygon operations[46] such under/over size and logic operations (AND NOT etc.). Using a virtual layout has the disadvantage that details of the actual layout are lost, so that "problem" areas are difficult to identify.

Ideally the critical area would be calculated from the circuit layout and each polygon labelled with it's contribution to the total critical area. This can be done using polygon over/under sizing and a modified version of the plane sweep algorithm presented by Luther[96] for boolean operations on mask layers. This procedure would be slower than using a virtual layout, but would be more useful in identifying and observing the effects of LDR application.

Since these methods are both heavily based on polygon manipulation they could share a large amount of common code. It is therefore feasible to consider

implementing a yield predictor that can be operated in two modes. One that gives a quick solution, based on a virtual layout, with little detail and a second mode that provides details of the total contribution of each component (polygon, contact, wire, transistor etc.) This second mode should also be able to highlight circuit features that exhibit poor yield.

9.8 Results

Local design rules have been applied to a number of circuit layouts. Simulation of these layouts suggests that the defect sensitivity reduction is highly dependent on the actual circuit layouts. This is not surprising since some layout can be left unchanged by attempts to apply LDRs.

9.8.1 Track Displacement

The results presented in chapter 8 indicate that a reduction of between 3-15% in the number of extra material metallisation faults can be achieved using track displacement LDRs of metal layers. It is clear from an inspection of these layouts (figures 8–2 and 8–17) that there is further scope for improvement particularly if LDRs to displace contacts and tracks were developed.

9.8.2 Track Width

The results for track width LDRs are more difficult to assess than the displacement LDRs. While it is clear that it is possible to reduce the number of missing material faults (breaks) there is also an increase in the number of extra material faults (shorts). The ratio of the change in breaks to shorts can be manipulated by adjusting the LDR. By specifying that a larger separation between neighbouring geometry must be present before a track increase can be applied, the ratio

of break reduction to short increase is improved. This will of course reduce the number of instances in which the track width LDR can be applied.

Another factor that determines the ratio of extra and missing material faults changed by the track width LDR is the ratio of the defects that cause the faults. Modern MOS processes often suffer more from shorts than breaks. Track widths are determined more by the current densities they carry rather than the limits of the lithographic process[97]. For fabrication processes in which there are significantly more shorts than breaks, the only suitable changes in track widths are to tracks distant from other layout geometry. Since only a small proportion of tracks will be sufficiently distant to apply the LDR, the yield improvement will be small and therefore using a LDR may not be justified. In a process that has a more equal distribution of defect types or a larger number of breaks than shorts, track width LDRs can make a useful contribution to the overall yield.

9.8.3 Contact Increase

The results from contact size increase LDRs indicate that the extra overlap generated to accommodate the larger contact size can have an important impact on the defect sensitivity of the overlap layers. Just as is the case with track width LDRs increasing contact size can result in a decrease in overall yield for some processes. However, where the increase in yield from the larger contact cuts more than offsets the overlap yield loss contact increase LDRs can be used successfully.

9.9 Yield Improvement from LDRs

The yield improvement obtained from the application of a given LDR has been expressed as a percentage change in yield for that process step. E.g. in chapter 8 metal 1 routing yield was improved by 3%. The total yield of a device is the product of the individual process steps i.e.,

$$Y_{total} = Y_n \, Y_{(n-1)} \, Y_{(n-2)} \ldots . \, Y_2 \, Y_1 \tag{9.1}$$

Where each of the Y_n terms represents the yield of a process step. After a LDR has been applied, the yield of the process step is $(1 - \Delta Y_n)Y_n$ where $\Delta Y_n \times 100$ is the percentage change in the yield of step n. Thus the equation for total yield after LDRs have been applied is,

$$Y_{total} = (1 - \Delta Y_n)Y_n\,(1 - \Delta Y_{n-1})Y_{(n-1)} \ldots\ (1 - \Delta Y_2)Y_2\,(1 - \Delta Y_1)Y_1 \qquad (9.2)$$

Thus, when applying LDRs, it can be contended that resources should be expended on those process steps which contribute most to the total yield loss.

9.10 Future Work

The use of LDRs to improve yield of circuits is non-optimal; because of the complex nature of layout it is not possible to distill the optimum yield improvement strategy down to a small number of simple rules. It is proposed that a more optimum yield can be achieved where there is a closer link between IC layout generation and yield prediction. This implies that, where there are no other constraints, layout decisions are based on the resulting yield of the various layout options.

There are a number of stand-alone tools for yield prediction of circuit layout[48, 47, 42], but these do not contribute directly to improved yield. These tools are not able to report on the yield of individual layout components so that alternatives can be compared. They can, however, be used to test the results of yield improvement strategies, for example LDRs.

Future work could consist of the design and implementation of a hierarchical, constraint-based circuit-compactor of symbolic layout. This could be used to investigate and compare strategies for obtaining IC layout with minimum critical

area [2]. This will require an efficient method of calculating the critical area of the layout and also changes in critical area as a result of small changes, as different layout options are explored.

State of the art constraint based compactors, that attempt only area, rather than area and critical area, minimisation are able to generate layout with an area close to that achieved by hand-crafting layout[76]. Using even limited physical layout manipulation it is possible to reduce the defect sensitivity of a circuit layout. For example, the results presented here suggest that the sensitivity of metal layers could be reduced by as much as 15%. By combining a constraint based compactor with more sophisticated symbolic geometry manipulation, circuit layout with yields greater than the equivalent hand-crafted layouts becomes a possibility.

9.10.1 Sources of Yield Improvement

The ability to calculate critical area and hence yield can be used to obtain yield improvement of large IC circuits. The yield of the cells and interconnect defining the circuit can be used to determine the circuit yield. By exploiting the inherent hierarchy and symbolic nature of the layout, the yield of the whole system of interconnected cells can be optimised. This can be achieved by an iterative procedure of system, cell and layout geometry manipulation, either manually, where an obvious improvement can be made, or automatically, using the predicted yield as feedback. Where this process results in changed layout constraints, "re-compaction" to minimise critical area will have to be undertaken as detailed below.

---

[2]The critical area[5] is a measure of sensitivity to defects. It is dependent on the defect size distribution of the IC fabrication process and the dimensions and proximity of layout geometry

- **System Level**

  A symbolic routing routing layer is proposed to permit the layout of a cell to change at any stage in the design cycle. Where there is sufficient available space this symbolic routing would be limited to selected physical layers to allow partial or complete through routing of the cell. While this may increase the critical area and size of an individual cell, it can result in a reduction in both the critical and physical area of the complete system (figure 9–2(b)).

  Similarly, a cell can be re-shaped to abut with a neighbouring cell's ports, or conform to an available space. Such a cell may be longer and possibly thinner than the optimum cell layout and might have a larger critical and physical area. This is acceptable where the yield of the system as a whole is improved (figure 9–2(c)).

- **Cell Level**

  Where there is incomplete use of the surrounding area, a cell would be stretched to fill this space. Stretching of internal cells and/or "re-compaction" to minimise the critical area of the layout within this new larger area would result in a critical area less than or equal to the original value obtained for the minimum cell size (figure 9–2(d)).

- **Layout Geometry Level**

  The optimum yield of the individual cells would be achieved by minimising the critical area of the layout geometry within the constraints imposed by the cell boundaries and any overlapping geometry from other cells (e.g. through routing). A range of layout changes that often give reductions in critical area would be attempted. Changes would be accepted where they prove to cause a reduction in critical area. Such changes include,

  - Displacement of tracks and geometry away from closest neighbouring geometry
  - Increased track width

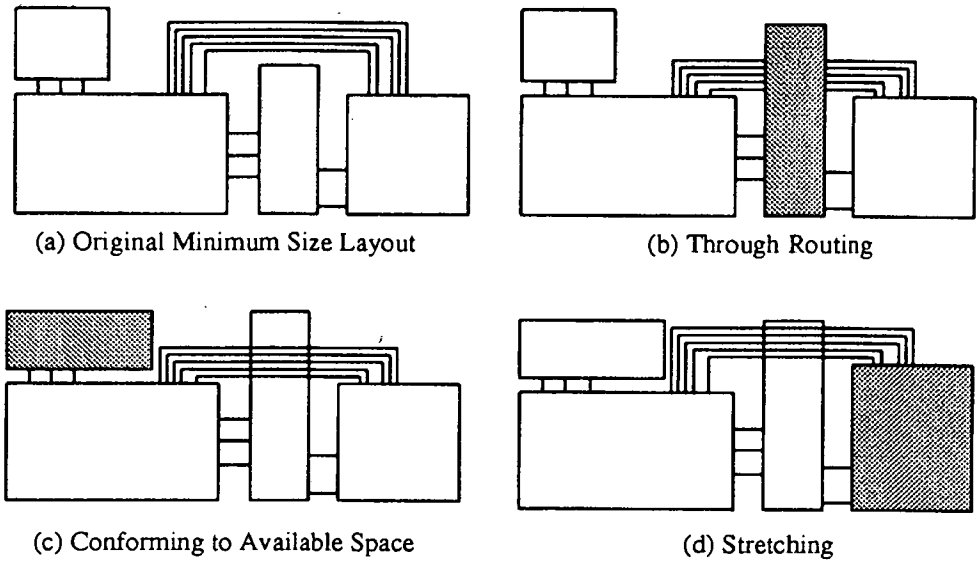


Figure 9–2: Application of Yield Improvement Techniques to an IC

- Shortening track lengths by straightening
- Increased contact size
- Increased number of contacts
- Redistribution of tracks in routing channels to obtain maximum benefit from track displacement and other layout changes.

9.10.2 Goal

The goal of this work would be to develop a CAD tool capable of producing circuit layout with a yield comparable to the best hand-crafted layout. It is intended that this tool would be largely automatic but guidance from the circuit designer, particularly in areas where the designer has a better "understanding" of the goals and specification requirements, would be included as an integral part of the design flow.

This proposed tool would have to take input in the form of a hierarchy of symbolic layout cells from a variety of sources,

- input directly by the designer through a graphics interface.
- existing tools that generate symbolic layout from netlist[95].
- circuit routers[90].

There should also be the ability to include existing layout, so that high yielding circuits, already developed for a given fabrication process, can be included within a large system. This will permit the use of specialist circuit parts such as RAM and CPUs.

This tool should not stand alone but should interface to existing CAD software, especially circuit simulation and routing software. Indeed using CAD tools that generate symbolic layout from physical layout[98], it will be possible to use the existing CAD environment with yield optimisation as the final stage before going to production.

9.11 Conclusions

This thesis has suggested that the layout of integrated circuits is non-optimal. Results have been presented that indicate that it is possible to create a more optimal layout by using a simple set of rules, known as Local Design Rules. These rules are used to improve the yield of a circuit without having an impact on the circuit performance or the area of the layout. Local design rules to displace circuit tracks, increase track width and to increase contact/via area have been presented along with algorithms for their implementation.

Any yield improvement strategy should be automated, if at all possible. It is unlikely that any procedure that delays the production of a new device will be accepted in a commercial environment. Therefore, it is essential that the operations required to optimise layout do so rapidly causing a minimum delay i.e., the procedure must be automated. A program LocDes that automatically applies local design rules to IC layout has been developed. This program can

be used interactively, through an X window interface so that individual items of layout can be selected, or in a batch mode, where LDRs are applied to the whole layout. The application of LDRs to large layouts can take a number of hours; very large layouts may take days. It has been shown that these times can be greatly reduced by using the parallel option in the LocDes program, cutting the response time to a fraction of the single CPU execution time.

Restricting the optimisation of layout to a few well defined changes, such as those specified by a LDR, ensures that the task does not become too computationally expensive. However, this method is unlikely to generate the optimal layout, mainly because some potentially beneficial layout options are ignored or not tested. The best that can be achieved is an improvement to the existing layout. Methods to give a near optimal solution to this problem have been outlined and discussed in section 9.10.

The effectiveness of LDRs is layout dependent; this is intuitively obvious and easily confirmed by comparing the resulting yield improvement in different cells. This is true even of cells that form part of the same design. The reduction in defect sensitivity to extra material defects (shorts) obtained by track displacement LDRs for metal layers can range from 0–15% though a figure of 3–5% would be more normal. The change in defect sensitivity obtained from track width and contact LDRs is more difficult to determine. This is due to the fact that these rules affect more than one defect mechanism and these mechanisms are highly process dependent. For example, contact size can in many instances be increased for the majority of contacts by 50%. This increase requires a corresponding increase in contact overlap. The combined effect of these changes depends on the relative proportions of defects for the contact itself and the layer that make up the overlap. Similarly, track width increase also reduces track separation so that the ratio of extra and missing material defects is required to calculate the overall defect sensitivity change.

As mentioned above, LDRs are process sensitive; it is not possible to develop or improve the yield of IC layout without some reference to the fabrication pro-

cess that is to be used to produce the device. This means that information not currently distributed by IC manufacturers is required to implement either LDRs or any other form of yield improvement strategy. It can be shown that it is in the manufacturers interest to supply this information, since it will effectively result in an increase in production capacity. The actual defect densities themselves may not be required but the relative defect densities of different layers and inter-layer interactions are essential.

The yield improvement obtained by applying LDRs to circuit layout can be used either to improve the production yield of the device or alternatively to develop a larger chip with greater functionality and similar yield to the old device. It is has been shown that a layout based yield improvement strategy can be of use to both the circuit designer and process engineer by extending the capabilities of existing technologies.

References

[1] Brian Hendrerson (Motorola Process Development Engineer). Personal communication.

[2] George Bell (IBM Greenock). Personal communication.

[3] B. T. Murphy. Cost-size optima of monolithic integrated circuits. *Proc. IEEE.*, 52, Dec 1964.

[4] C. H. Stapper. Modeling of integrated circuit defect sensitivities. *IBM J. Res. Develop.*, 27(6):549–557, Nov 1983.

[5] A. V. Ferris-Prabhu. Computation of the critical area in semiconductor yield theory. In *Europ. Conf. Electronic Design Automation*, pages 171–173, Warwick, 1974.

[6] C. Stapper. Personal Communication. In *IEEE Workshop on Defect and Fault Tolerance in VLSI Systems*, Institut National Polytechnique de Grenoble, Nov 1990.

[7] F. M. Klassen. Design and performance of micron-size devices. *Solid State Electron.*, 21:565–571, 1978.

[8] R. D. Rung. Determining IC layout rules for cost minimization. *IEEE J. Solid State Circuits*, SC-16(1), Feb 1981.

[9] A. C. Ipri. Integrated circuit process and design rule evaluation techniques. *RCA Review*, 38, Sep 1977.

[10] A. Gupta and J. W. Lathrop. Yield analysis of large integrated-circuit chips. *IEEE J. of Solid State Circuits*, SC-7(5):389–398, Oct 1972.

[11] C. H. Stapper. Integrated circuit yield statistics. *Proc. of the IEEE*, 71(4), Apr 1983.

[12] C. Alcorn. Kerf test structure designs for process and device characterization. *Solid State Tech.*, 28(5):229–235, May 1985.

[13] T. Yanagawa. Influence of epitaxial mounts on the yield of integrated circuits. *Proc. IEEE*, 57:1621–1628, 1969.

[14] D. S. Perloff et al. Microelectronic test chips in integrated circuit manufacture. *Solid State Technol.*, 22:205–211, Sept 1981.

[15] T. E. Mangir. Sources of failures and yield improvement for VLSI and restructurable interconnects for RVLSI and WSI. *Proc. IEEE*, 72(6):690–708 part 1, Jun 1984.

[16] T. E. Mangir. Impact and limitations of interconnect technology on VLSI and restructurable VLSI design. *Proc ICCD 83: VLSI in Computers*, Oct 1983.

[17] C. H. Stapper. LSI yield modeling and process monitoring. *IBM J. Res. Develop.*, 20(3):228–234, May 1981.

[18] C. H. Stapper. Yield model for productivity optimization of VLSI memory chips with redundancy and partially good product. *IBM Journal of Research and Development*, 24(3):398–409, May 1980.

[19] C. H. Stapper. A simple method for modeling VLSI yields. *Solid State Electronics*, 25(6), 1982.

[20] D. M. H. Walker. *Yield simulation for integrated circuits.* The Kluwer International Series in engineering and computer science, VLSI, computer architecture and digital signal processing. Kluwer, Boston; Lancaster:, c1987.

[21] R. Willians and M. Beguwala. Reliablity concerns for small geometry MOSFETS. *Solid State Tech*, 24(3):65–71, Mar 1981.

[22] S. Hofstein and F. Hieman. The silicon insulated-gate field-effect transistor. *Proc of IEEE*, 51(5):511–516, Mar 1963.

[23] A. Kovchavtsev and A. Frantsuzov. Defect density in thermally grown silicon oxide with thickness 30–800 angstroms. *Mikroektronika*, 8(5):330–333, Sep–Oct 1979.

[24] L. Jastrzebski. Origin and control of material defects in silicon VLSI technologies:An overview. *IEEE Journal of Solid-State Circuits*, SC17(2):105–107, April 1982.

[25] C. H. Stapper. Yield model for fault clusters within integrated circuits. *IBM Journal of Research and Development*, 28(5), Sep 1984.

[26] In *IEEE Workshop on Defect and Fault Tolerance in VLSI Systems*, Institut National Polytechnique de Grenoble, Nov 1990.

[27] I. Koren and Z. Koren. Employing the unified negative binomial distribution for yield analysis of empirical data. In *IEEE Workshop on Defect and Fault Tolerance in VLSI Systems*, Institut National Polytechnique de Grenoble, Nov 1990.

[28] R. B. Seeds. Yield, economic, and logistic models for complex digital arrays. *IEEE International Convention Record*, pages 60–61, Mar 1967.

[29] J. E. Price. A new look at yield of integrated circuits. *Proc. IEEE(Letters)*, 58(8):1290–1291, Aug 1970.

[30] A. P. Turley. LSI yield projections based upon test pattern results: An application to multilevel metal structures. *IEEE Trans. on Parts, Hybrids, and Packaging*, PHP-10(4), Dec 1984.

[31] P. R. Pukite and C. L. Berman. Defect cluster analysis for wafer scale integration. *IEEE Trans. on Semiconductor Manufacturing*, 3(3), 1990.

[32] C. H. Stapper. Discussion. In *IEEE Workshop on Defect and Fault Tolerance in VLSI Systems*, Institut National Polytechnique de Grenoble, Nov 1990.

[33] A. V. Ferris-Prabhu. Role of defect size distribution in yield modeling. *IEEE Transactions on Electron Devices*, ED-32(9):1727–1736, Sep. 1985.

[34] W. Maly. Systematic characterization of physical defects for fault analysis of MOS IC cells. In *In IEEE International Test Conference Proceedings*, pages 390–399. University of Oxford, Oct 1984.

[35] A. C. Ipri. Evaluation of CMOS transistor related design rules. *RCA Review*, 41, Dec 1980.

[36] C. Groves, M. A Caux, and M. Pigeon. An automated system for CMOS layout rule generation and analysis. *Proc. Custom Integrated Circuit Conference*, pages 425–429, 1984.

[37] A. J. Walton, R. J. Holwill, and J. M. Robertson. On-chip switching for DC parametric testing. *Electronics Letters*, 21(10):422–423, May 1985.

[38] E. J. Sprogis. Defect diagnostic matrix – a defect learning vehicle for submicron technologies. In *International Conference on Microelectronic Test Structures*, Essex Junction, Vermont, 1988. IBM General Technology Division.

[39] A. K. Lewis. A fault tolerent test chip for gathering wafer defect data. In *International Workshop on Designing for Yield*, University of Oxford, Jun 1987.

[40] R. Razdan. A statistical design rule developer. *IEEE Trans. Comp. Aided Design*, CAD-5(4), Oct 1986.

[41] S. R. Nassif. FABRICS II: A statistical based IC fabrication process simulator. *IEEE Trans. Comp. Aided Design*, CAD-3(1), Jan 1984.

[42] H. Walker. VLASIC: A catastrophic fault yield simulator for integrated circuits. *IEEE Trans. Comp. Aided Design*, CAD-5:541–556, Oct 1986.

[43] D. M. H. Walker and D. S. Nydick. DVLASIC: catastrophic fault simulation in a distributed processing envioronment. *IEEE Trans Comp. Aided Design*, CAD-9:655–664, Jun 1990.

[44] W. Maly. Modeling of lithography related yield losses for CAD of VLSI circuits. *IEEE TRANS. on Computer Aided Design*, CAD-4(4):166–177, Jul 1985.

[45] W. Maly. Yield estimation model for VLSI artwork evaluation. *Electronics Letters*, 19(6):226–227, Mar 1983.

[46] S. Gandemer. Critical area and critical levels calculation in IC yield modeling. *IEEE Trans. Electron Devices*, 35:158–166, Feb 1988.

[47] I. Chen and A. J. Strojwas. Realistic yield simulation for VLSIC structural failures. *IEEE Trans. Comp. Aided Design*, CAD-6(6):965–978, Nov 1987.

[48] M. Lorenzetti and P. McGill. McYield: A CAD tool for functional yield projections for VLSI. In *IEEE Workshop on Defect and Fault Tolerance in VLSI Systems*, Institut National Polytechnique de Grenoble, Nov 1990.

[49] A. V. Ferris-Prabhu. Modeling of critical area in yield forecasts. *IEEE Journal of Solid State Circuits*, SC-20(4):874–880, Aug. 1985.

[50] J. Ousterhouse, G. Hamachi, R. Mayo, W. Scott, and G. Taylor. Magic: a VLSI layout system. In *Proceedings of the 21st ACM/IEEE Design Automation Conf.*, pages 152–159, June 1988.

[51] W. R. Moore. A review of fault- tolerant techniques for the enhancement of integrated circuit yield. *Proc. IEEE*, 74(5):684–698, May 1986.

[52] J. M. Wilkinson. Design and manufacture of a wafer scale memory product. In *International Workshop on Designing for Yield*, University of Oxford, Jun 1987.

[53] K. Yamashita, S. Ikehara, M. Nagashima, and T. Tatematsu. A 600 MBit random access wafer scale memory. In *IEEE Workshop on Defect and Fault Tolerance in VLSI Systems*, Institut National Polytechnique de Grenoble, Nov 1990.

[54] R. J. Petritz. Current status of LSI technology. *IEEE Solid State Circuits*, SC 2(4):130–147, Dec 1967.

[55] J. W. Lathrop. A discretionary wiring system as the interface between design automation and semiconductor array manufacture. *Proc. IEEE*, 55:1988–1997, 1967.

[56] B. J. Donlan et al. Computer-aided design and fabrication for wafer scale integration. *VLSI Des.*, pages 34–42, Apr 1985.

[57] B. J. Donlan et al. Wafer scale integration using discretionary microtransmission interconnections. In C. R. Jesshope and W. R. Moore., editors, *Wafer Scale Integration.* Adam Hilger, Bristol, UK, 1986.

[58] R. T. Smith et al. Lazer programmable redundancy and yield improvement in a 64k DRAM. *IEEE Solid State Circuits*, SC - 16(5):506–513, Oct 1981.

[59] O. Minato et al. A high speed hi-CMOSII 4k static RAM. *IEEE J. Solid State Circuits*, SC-16(5):449–453, Oct 1981.

[60] O. Minato et al. A hi-CMOSII 8k x 8 bit static RAM. *IEEE J. Solid State Circuits*, SC-17(5):793–798, Oct 1982.

[61] T. Mano et al. A redundancy circuit for a fault tolerant 256k MOS RAM. *IEEE J. Solid State Electron.*, SC- 1(4):726–731, Aug 1982.

[62] J. R. Lineback. Vertical shorts add row to memory. *Electronics*, pages 50–52, Sept 8,.

[63] J. P. Coullahan et al. Redundant circuit personalization at the module level. *IBM Tech. Discl. Bull.*, 23(9):4168–4169, Feb 1981.

[64] C. Kuo et al. An 80ns 32K EEPROM using the FETMOS cell. *IEEE J. Solid-State Circuits*, pages 821–827, Oct 1982.

[65] T. Goto and N. Manatu. How japanese manufacturers achieve a high IC reliability. *Electronics*, 53(6):140–147, Mar 1980.

[66] C. Stapper. Fault tolerance galore. In *IEEE Workshop on Defect and Fault Tolerance in VLSI Systems*, Institut National Polytechnique de Grenoble, Nov 1990.

[67] D. T. Fitzpatrick. *MEXTRA: A Manhattan Goemetry Extractor.* Electronic Research Lab Memo M82/42, UC Berkeley, 1982.

[68] Carver Mead and Lynn Conway. *Introduction to VLSI Systems.* Addison-Wesley, Reading, Massachusetts, 1980.

[69] L. Walczowski, W. Waller, and E. Wells. *CHIPWISE: An integrated toolset for full custom VLSI.* The ECAD Laboratory, The University of Kent, Canterbury, Kent.

[70] W. Maly. Double-bridge test structure for the evaluation of type , size and density of spot defects. In *International Workshop on Designing for Yield.* University of Oxford, Jun 1987.

[71] P. Gill. Using snake patterns to monitor defects and enhance VLSI device yields (part 1). *Microcontamination*, pages 23–31, Feb 1989.

[72] A. C. Ipri. Impact of design rule reduction on size, yield, and cost of integrated circuits. *Solid State Tech*, Feb 1979.

[73] A. C. Ipri. Evaluation of CMOS transistor related design rules. *RCA Review*, 41, Dec 1980.

[74] R.D. Rung. Determining IC layout rules for cost minimisation. *IEEE J. Solid-State Circuits*, pages 35–43, Feb 1981.

[75] L. W. Nagel. *SPICE 2: A Computer program to simulate semiconductor circuits.* Electronic Research Lab Memo M510, UC Berkeley, May 1975.

[76] D. D. Hill. *Algorithms and Techniques for VLSI layout synthesis.* Kluwer, Boston; Lancaster:, c1989.

[77] J. P. Fishburn and A. Dunlop. TILOS: A posynomial programming approach to transistor sizing. In *Proceedings of IEEE International Conference on Computer Aided Design-85.* Santa Clara Ca., 1985.

[78] D. Guedj, J.P. Porcher, and M. Rivier. Yield simulation of CMOS memory and logic products. In *IEEE Workshop on Defect and Fault Tolerance in VLSI Systems*, Institut National Polytechnique de Grenoble, Nov 1990.

[79] Silicon Compilers Incorporated. *GENESIL System Users Manual.* 2045 Hamilton Avenue, San Jose, CA 95125.

[80] Silicon Compilers Incorporated. *GENECAL Users Guide.* 2045 Hamilton Avenue, San Jose, CA 95125.

[81] S. M. Trimberger. *An Introduction to CAD for VLSI.* Kluwer, Boston; Lancaster:, 1987.

[82] H. S. Blair. Fast algorithms for LSI artwork analysis. *Journal of Design Automation and Fault-Tolerant Computing*, pages 179–209, 1978.

[83] K. Yoshida. Layout verification. In T. Ohtsuki, editor, *Layout Design and Verification.* Elsevier Science Publishers, B.V North-Holland, 1986.

[84] A. Weinand E. Gamma and R. Marty. Design and implementation of a seamless object-oriented application framework. *Structured Programming*, 10(6), Jun 1989.

[85] R. L. Brown. Multiple storage quad trees: A simpler fast alternative to bi-sector quad trees. *IEEE Trans Comp. Aided Design*, CAD-5:413–419, 1986.

[86] G. Kadem. The quad-CIF tree: a data structure for hierarchical online algorithms. In *19th Design Automation Conference*, pages 352–357, 1982.

[87] P. Hsiao and W. Feng. Using a multiple storage quad tree on a hierarchical VLSI compaction scheme. *IEEE Trans. Comp. Aided Design*, CAD-9(5):522–536, May 1990.

[88] K. Ueda, R. Rasai, and T. Sudo. Layout strategy, standardization and CAD tools. In T. Ohtsuki, editor, *Layout Design and Verification.* Elsevier Science Publishers, B.V North-Holland, 1986.

[89] ES2 Publications Unit. *Solo 1400 Reference Manual.* ES2, Mount Lane, Bracknell, Berkshire, UK.

[90] J. Reed, A. Sangvovanni-Vincentelli, and M. Santomauro. A new symbolic router: YACR2. *IEEE Trans Comp. Aided Design*, CAD-10:208–219, July 1991.

[91] S. M. Sze. *VLSI Technology.* McGraw-Hill Book Company, New York, 1983.

[92] *The Motorola Guide to Statistical Process Control For Continuous Improvement Towards Six Sigma Quality (BR392/D.* Motorola Inc. Semiconductor Products Sector, Phoenix, Arizona.

[93] Gerard Blair. Personal communication.

[94] Grady Booch. *Object Oriented Design with Applications.* Benjamin /Cummings Publishing Company Inc., Redwood City, Calif., 1991.

[95] R. Costa, F.Curatelli, D. D. Caviglia, and G. M. Bisio. Symbolic generation of constrained random logic cells. *IEEE Trans Comp. Aided Design*, CAD-10:220–231, Feb 1991.

[96] U. Lauther. An O(Nlog N) algorithm for boolian mask operations. In *18th Design Automation Conference*, pages 555–560, 1981.

[97] Wojciech Maly. Personal communication.

[98] B. Lin and A. R. Newton. A circuit disassembly technique for synthesizing symbolic layouts from mask descriptions. *IEEE Trans Comp. Aided Design*, CAD-9:959–969, Sep 1990.

Appendix I

Yield Improvement with Local Design Rules

G. Allan, A. Walton, and R. Holwill.

Yield improvement with local design rules.

IEEE Workshop on Defect and Fault Tolerance in VLSI Systems,

Institut National Polytechnique de Grenoble, Nov 1990.

Yield Improvement with Local Design Rules.

Gerard A. Allan A. J. Walton R. J. Holwill

Edinburgh Microfabrication Facility
University of Edinburgh
Fax no 031-662 1081
Email gaa@uk.ac.ed.emf
Aug 29, 1990

1 Introduction

The yield of IC devices is a major factor in determining their commercial success. This has led to circuit designers actively designing for high yield. An example of this is the use of redundancy in RAMs where fusible elements are used to disable defective rows and insert replacements [13]. Since this early use of redundancy, involving the trade off of silicon area for yield, a range of procedures to increase yield through restructurable interconnect have been developed [10] and this type of approach has led to a commercial WSI device [18].

At the same time as these developments, silicon area has also been traded for ease of layout [11]. This paper introduces local design rules, which reclaim some of this lost area and use it to enhance the yield of a product.

2 Local Design Rules

The layout of an integrated circuits is bound by a single set of design rules. These rules, determine the minimum size and spacing of all layers of the circuit geometry in an attempt to maximise the yield, performance and reliability. There are occasions when specialist circuit parts, such as RAM, use a different set of design rules that have been optimised for that particular circuit block. However, normally the design rules are applied over the whole of layout area and will therefore be referred to as Global Design Rules (GDR).

The GDRs have been optimised to give good layout from a single global set of rules but are not necessarily optimised for the local layout conditions. For example, the conductor width and contact size can have a significant effect on the yield of a circuit [6, 7, 12]. Under certain local conditions, where there has been incomplete use of the surrounding area, it would be advantageous to adjust the metal width.

contact size [1] to greater than the global norm. Wider metal tracks and larger contacts have a higher yield, if they are sufficiently distant from other geometry. The use of automated layout generation and compaction produces designs that are less dense than traditional hand-crafted layout and give greater scope for yield enhancement through the use of Local Design Rules (LDR).

The global rules determine minimum size and spacing and hold over the whole design, but these rules should not be used to determine the maximum feature size of circuit components. While it is valid to attempt an initial layout with minimum sizes to reduce the circuit area, once that minimum area has been defined best use of any redundant area should be made.

There are a number of potential layout changes that can be made,

- Increased contact size.
- Increased contact overlap.
- Increased metal track width.
- Increased polysilicon and active track width where they do not form active devices.
- Increased substrate contact size.

It is proposed that the yield of circuits fabricated using layout generated from a GDR set can be increased by searching for instances of local non-optimal layout, applying a LDR set and adjusting the layout to meet the new design rules.

Performing the above procedure by hand is too time consuming and prone to error. To address this problem a Rule Adjustment In Local Environment (RAILE) program acting as a post-processor of CIF format layout has been developed. This uses the GDR layout to produce an enhanced circuit layout (Fig 1), with the same function and performance specifications but with a higher yield.

While it is intended that the yield of the resulting layout will be greater than the initial GDR layout, this can only be guaranteed if the fabrication process is understood well enough to to ensure that the LDRs are an accurate reflection of the relative yield of the layout options under test. That is, no changes are made to the layout except those that have been shown to give a higher yield.

3 Defining the Local Design Rules.

Local design rules are used to determine where changes in layout generated from GDR sets should be performed. LDRs do not define minimum widths and separations, since this is already done by the GDRs. Their purpose is to define maximum feature sizes,

[1] Larger contacts, particularly those increased in both the X and Y direction, may require an increase in the overlap of the contact. This avoids problems that may be created with over etch of large contacts in a process optimised for the minimum contact and via size.

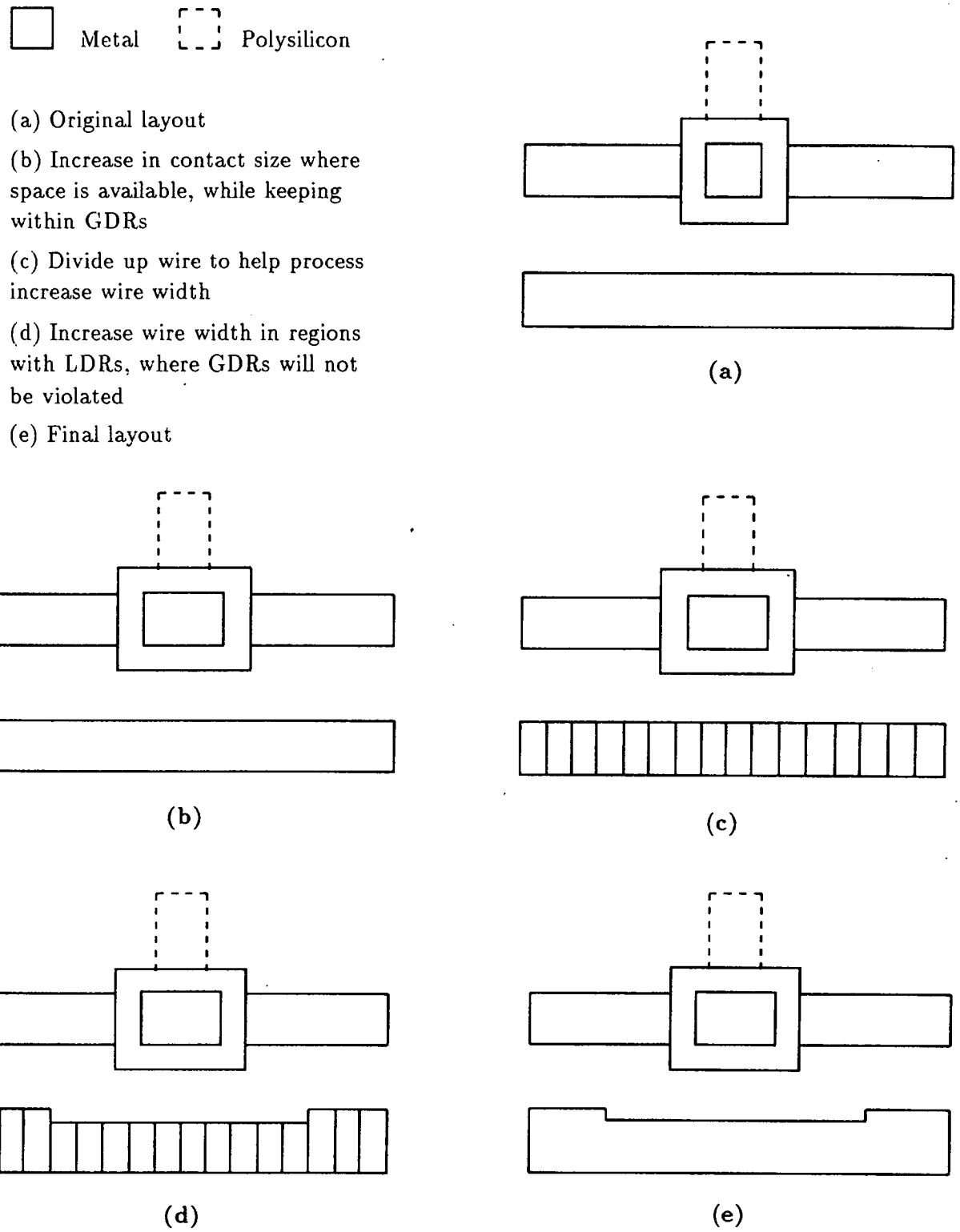


Figure 1: Example of Local Design Rule Changes to Contact and Wire.

in relation to the available space around the geometry under consideration. The LDRs are derived from a comparison of the yields of a number of layout options.

The complete definition of the LDRs requires a greater knowledge of the process than the GDRs, since the problem is no longer a "simple" matter of finding one rule set to maximise yield of a regular test structures. Rules to determine the optimum layout in a variety of situations need to be determined. This in practical terms means that it must be known where an increase in track width, contact size with a consequent reduction in track separation will result in an increase in yield. The information can be found using test structures as is the normal procedure for GDR generation. This will require a greater range of test structures, but there may be sufficient information to make a first approximation of LDRs from the results of test structures designed to determine GDRs.

3.1 LDRs from Defect Size and Distribution.

Yield simulation with defect size and distribution data [9, 4, 15, 3, 16] can, for example, be used to determine where a change in conductor width will increase the overall yield. The number of fatal defects that are the result of shorts between nodes cannot be reduced but the number caused by breaks can be reduced by widening the conductors. This will of course increase the number of shorts(fig. 5), but if the overall number of fatal defects can be decreased then there will be a corresponding increase in yield.

A number of LDRs can be generated from the results of the simulation, depending on the sophistication of the simulator available.

4 Performance

It is important that any changes as a result of the application of LDRs to a layout do not cause a degradation in performance of the circuit. For this reason no changes that affect the size of active devices should be made. The effects of changes in track width, contact and substrate contact size must be considered.

Increased contact size.
Increased contact size will result in a decreased contact resistance and increased overlap capacitance. The effect of overlap capacitance can be neglected because it is effectively shorted by the contact, the RC product will therefore be reduced in line with any reduction in the total contact resistance [17].

Increased substrate contact size.
Increased substrate contact size can take place where the contact is sufficiently distant from active devices. Since they are not used to pass signals there will be no effect on performance of the circuit.

Increased metal track width.
Increased track width will cause an increase the capacitance of the circuit but

will also decrease the the resistance of the circuit. The RC product will be reduced because the fringe capacitance, which is a significant fraction of the total, is not increased. Capacitance between tracks on the same conductor level will not be larger than that allowed for in the GDRs.

There is a possibility of greater cross talk where the capacitance between crossovers is increased. To avoid this, overlapping geometry, not forming a single electrical node, should not be altered.

Increased polysilicon width

The polysilicon layer is similar to the metal layers with the exception that care must be taken to ensure that polysilicon forming transistor gate is unchanged.

4.1 Reliability

The reliability of a component is also a performance feature, there is little point in increasing yield at the price of the component reliability. The reliability of the circuits optimised using LDRs can also be improved. This requires that when the LDRs are calculated reliability issues are taken into account and that no layout change is made that would cause yield or reliability to be reduced.

There are occasions when high reliability is of greater importance than yield and LDR sets that increase reliability at the expense of yield can be developed.

5 The RAILE Program

The program takes the original layout, in CIF format, and processes it using a set of LDRs. It is basically a design rule checker that attempts small changes in layout based on the LDRs, accepting only those changes that do not violate any of the global or other local design rules.

The program uses a heuristic algorithm, that makes adjustments iteratively, such that "large" increases in conductor width will occur in a number of steps with attempts to change other neighbouring geometry before further increasing in the original conductor width. This is to ensure that there is an even application of LDRs. While this will not necessarily give the optimal layout. it will produce a good approximation to it in a reasonable time.

5.1 Spatial Data Structure

The execution time of the program is heavily dependent on the internal organisation of the layout geometry. A spatial data structure capable of representing IC layout in a space efficient manor with a fast method for identifying neighbouring geometry is essential.

- **Space Efficiency.**
  The amount of data to be manipulated depends on the circuit layout. Since the

layout hierarchy must be flattened, a small CIF file can result in a very large amount of layout data.

- **Fast Region search.**
  The program makes a large number of region searches to determine the position of neighbouring geometry. The amount of geometry within the region of design rule interaction is very small in comparison to the total. It is important that these are found efficiently.

5.1.1 The Adaptive Multiple Storage Binary Tree

There are a number of well known methods of representing mask-level physical layout. These are important in many applications, with the most notable being circuit compaction.

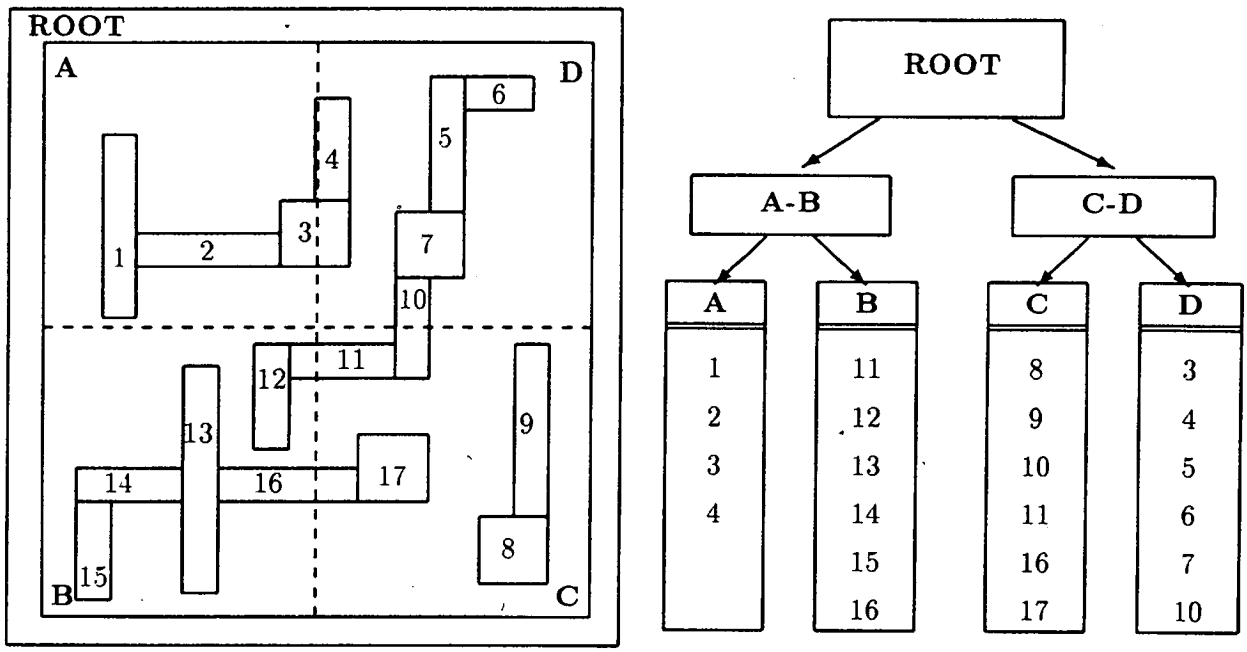


Figure 2: Mask Layout and Data Structure

The spatial data structure chosen was the Multiple Storage(MS) binary tree which is a variant of the MS-quad tree [1]. It is more properly described as an adaptive multiple storage binary tree and is identical to the MS-quad tree, except that each node points to two further node/leaf structures rather than four. This simplifies the algorithm. and has no significant effect on execution time or memory requirement.

The MS-tree stores the layout geometry (rectangles) in a tree structure, each node represents an area and can either point to two sub-areas or contain an array of references to all the geometry in that area, in which case it is a leaf cell. If a new item is added to a leaf cell and the reference array is already full, the area is split and all items in the existing array are repositioned in the appropriate sub-area leaf arrays.

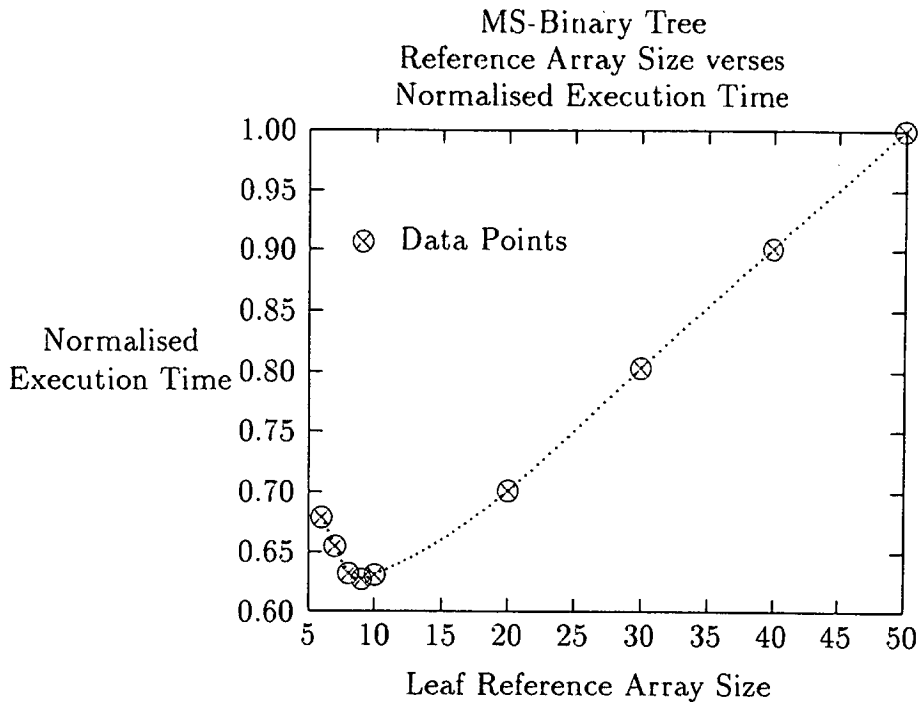


Figure 3: Reference Array Optimisation

If geometry straddles more than one area defined by a leaf cell, each leaf stores a copy of the reference to it. This does not uses as much memory as might be thought at first sight, as only the index or pointer to the geometry is stored. The geometry itself is stored in a dynamic array and referenced by an index. In fact the MS-quad tree has about the same memory usage as the bisector list quad tree [8], a simpler data structure and faster execution time [5].

Each physical mask layer is stored in a separate binary tree and an associated dynamic array referenced by indexes stored in that tree. An example of a mask layout is given in fig. 2 with the resulting data structure.

The size of the reference array has a marked affect on the efficiency of the program. The optimum value was was determined experimentally (fig. 3).

5.2 Large Circuit Layouts

Even using the most efficient data structures there will always be circuit designs that are too large to be processed in one piece. If circuit layout exceeds a specified limit, the program divides the layout into a number of segments. The layout can be processed in 1, 2, 4, 9, 16, 25 .. , segments depending on the size of design. Each segment overlaps with its neighbours to at least twice the maximum design rule operating distance, to ensure global and local design rules are not violated. All the individual segments are processed, the overlaps deleted and the results appended to a single output file.

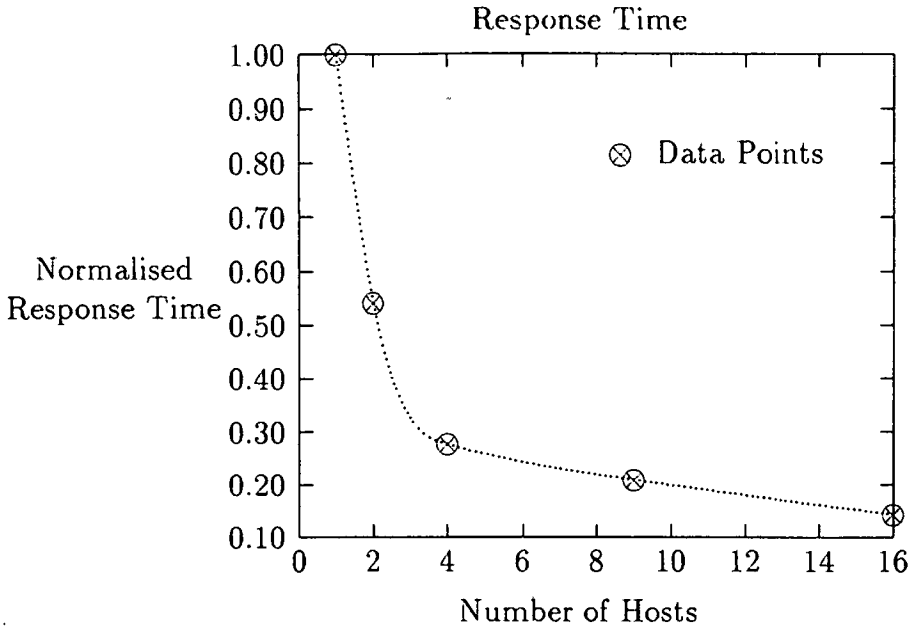


Figure 4: Response Time with Multiple Hosts

5.2.1 Distributed Processing

To reduce the time taken by the program to process large circuits the program includes a facility to use multiple hosts to process a single circuit layout.

Segments are assigned to remote hosts to be processed. Only one segment is assigned to each host. If there are more segments than available hosts, the controlling host waits until a remote host has finished processing its segment, before sending another one. Thus it is possible to use as many remote hosts as there are segments or use a limited number of hosts efficiently each processing only one segment at a time.

The results generated by the remote hosts are sent back to the controlling host where they are merged to form the final result. Fig. 4 shows the timing results from 1, 4, 9, and 16 hosts, for a circuit, where the circuit was segmented to match the number of hosts. These results are non-linear reflecting the extra processing that is done to extract the flattened area of interest from the hierarchical CIF layout. For maximum efficiency the segments should be large and for minimum response time the number of segments should match the number of hosts.

6 Results

The effectiveness of LDRs is highly process and to some degree circuit dependent. They are best suited for fabrication processes that have,

- yield loss due to poor contacts,
- yield loss from metal breaks,

- volume production of part.

The LDR approach is ideally suited to circuits that are, digital in nature, with inefficient layout eg. from automated cell layout and circuit compacters.

Although not implemented, LDRs could, where there is sufficient room, provide for the movement of tracks to increase their separation. This is a more complicated task and is a natural extension to the present work.

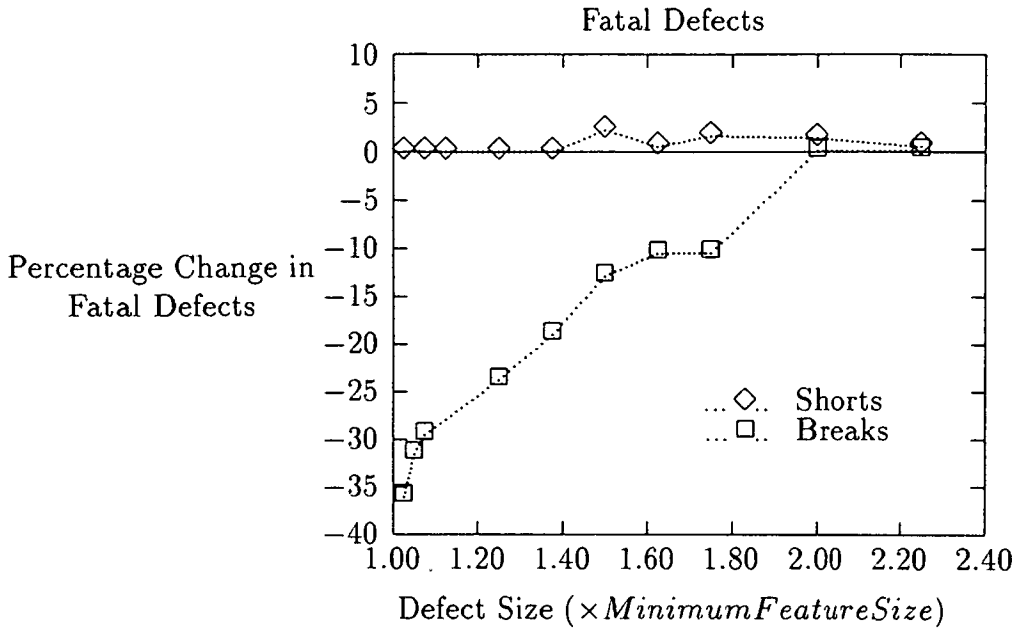


Figure 5: Percentage Change in Fatal Defects for Metal 1 LDR

6.1 Change in Defect Sensitivity

The result of a Monte Carlo simulation of yield of metal 1 stage for a dynamic shift register cell, generated using the STICKS [14] circuit compacter give the results shown in fig.5. This gives the percentage change in the number of fatal defects after application of a LDR. The rule can be expressed as: Metal 1 is increased up to 1.5× minimum metal 1 width so long as 1.5× minimum separation is kept between unrelated metal 1 and no GDRs, involving metal 1 with any layer, are violated.

The graph indicates that the probability of a spot defect causing a break can be reduced by applying this LDR. The data combined with the process defect size and distribution for metal 1, is used to determine if the LDR circuit has a higher yield when fabricated with that process.

Monte Carlo simulations are CPU intensive, a more efficient approach would be to use an analytical yield simulator such as RYE [2].

Contact Changes Per Iteration (%)					
Mixed Logic					
	Number of Changes in $\pm X/Y$				
Iteration	0	1	2	3	4
1st	4.3	19.5	31.6	31.0	13.6
2nd	12.9	36.3	28.9	19.6	2.3
3rd	22.5	38.3	22.8	15.1	1.3
4th	29.6	40.5	22.2	6.8	1.0
5th	35.4	40.8	17.7	6.1	0.0
PLA					
	Number of Changes in $\pm X/Y$				
Iteration	0	1	2	3	4
1st	0.0	15.1	30.2	41.5	13.2
2nd	1.9	19.8	35.8	32.1	10.4
3rd	2.8	23.6	36.8	26.4	10.4
4th	2.8	27.4	39.6	19.8	10.4
5th	6.6	25.5	41.5	16.0	10.4

Table 1: Percentage of contact changes in each iteration of a contact LDR

6.2 Change in Contact Size

The LDRs for contacts allow for bigger contacts. As an example table 1 gives the results of application of five iterations of a single LDR to a mixed logic circuit and a PLA. Each iteration attempts to increase the contacts size in four directions (ie. $+/-$X and $+/-$Y) by 1/6th of the minimum contact size. The LDR can be stated as: Contacts are increased in size, so long as there is no violation of any GDRs including those governing contact overlap (fig. 1(a), (b)). More complex LDRs involving contact size and overlap can be evolved to accommodate problems associated with over etch of large contacts cuts.

The the table lists the percentage of total contacts that were increased in a number of directions per iteration. The results show that a large percentage of contacts can be increased in size without violating GDRs, and that this percentage varies with the type of circuit. As in sec. 6.1 above, the usefulness of these result depends on the fabrication process.

7 Conclusions

It has been shown that the yield of IC layout is non-optimum. The yield can be increase by more effective use of device area through the application of local design

rules to layout generated from the normal "global" design rules. These rules, which are dependent on the fabrication process, can be found using test structures and the results of yield simulation with defect size and distribution data.

Local design rules will be of most use in processes that have suffer from poor contacts and conductor breaks, though extension to the present work (Sec. 6) my bring about increases in yield from other types of fault.

The proposed use of LDRs has indicated that its application can increase the yield of circuits which makes it a valuable addition to the circuit designer's toolkit.

References

[1] R. L. Brown. Multiple storage quad trees: A simpler fast alternative to bisector quad trees. *IEEE Trans Comp. Aided Design*, CAD-5:413–419, 1986.

[2] I. Chen and A. J. Strojwas. Realistic yield simulation for VLSIC structural failures. *IEEE Trans. Comp. Aided Design*, CAD-6(6):965–978, Nov 1987.

[3] S. Gandemer. Critical area and critical levels calculation in IC yield modeling. *IEEE Trans. Electron Devices*, 35:158–166, Feb 1988.

[4] P. Gill. Using snake patterns to monitor defects and enhance VLSI device yields. *Microcontamination*, pages 23–31 pt. I (Feb), pages 33–38 pt. II (Mar), 1989.

[5] P. Hsiao and W. Feng. Using a multiple storage quad tree on a hierarchical VLSI compaction scheme. *IEEE Trans. Comp. Aided Design*, CAD-9(5):522–536, May 1990.

[6] A. C. Ipri. Impact of design rule reduction on size, yield, and cost of integrated circuits. *Solid State Tech*, Feb 1979.

[7] A. C. Ipri. Evaluation of CMOS transistor related design rules. *RCA Review*, 41, Dec 1980.

[8] G. Kadem. The quad-CIF tree: a data structure for hierarchical online algorithims. In *19th Design Automation Conference*, pages 352–357, 1982.

[9] W. Maly. Double-bridge test structure for the evaluation of type , size and density of spot defects. In *International Workshop on Designing for Yield.* University of Oxford, Jun 1987.

[10] T. E. Mangir. Sources of failures and yield improvement for VLSI and restructurable interconnects for RVLSI and WSI. *Proc. IEEE*, 72(6):690–708 pt I (June), 1687–1692 pt. II (Dec), 1984.

[11] W. R. Moore. A review of fault-tolerant techniquies for the enhancement of integrated circuit yield. *Proc. IEEE*, 74(5):684–698, May 1986.

[12] R.D. Rung. Determining IC layout rules for cost minimisation. *IEEE J. Solid-State Circuits*, pages 35–43, Feb 1981.

[13] R. T. Smith et al. Lazer programmable redundancy and yield improvement in a 64k DRAM. *IEEE Solid State Circuits*, SC - 16(5):506–513, Oct 1981.

[14] L. Walczowski, W. Waller, and E. Wells. *CHIPWISE: An integrated toolset for full custom VLSI.* The ECAD Laboratory, The University of Kent, Canterbury, Kent.

[15] D. M. H. Walker. *Yield simulation for integrated circuits.* The Kluwer International Series in engineering and computer science, VLSI, computer architecture and digital signal processing. Kluwer, Boston; Lancaster:, c1987.

[16] H. Walker. VLASIC: A catasrophic fault yield simulator for integrated circuits. *IEEE Trans. Comp. Aided Design*, CAD-5:541–556, Oct 1986.

[17] A. J. Walton. Contact resistance of silicon-polysilicon interconnection for different current-flow geometries. *Electronics Letters*, 21:13–14, Jan 1985.

[18] J. M. Wilkinson. Design and manufacture of a wafer scale memery product. In *International Workshop on Designing for Yield*, University of Oxford, Jun 1987.

Yield Improvement by Track Displacement

G. A. Allan, A. J. Walton, and R. J. Holwill.

Yield improvement by track displacement.

In T. E. Mangir, editor, *Defect and Fault Tolerence in VLSI Systems III.*

Plenum Publishing Corp., New York, To be published.

YIELD IMPROVEMENT BY TRACK DISPLACEMENT

Gerard A. Allan A. J. Walton R. J. Holwill

Edinburgh Microfabrication Facility

University of Edinburgh, Kings Buildings. Edinburgh, EH9 3JL, UK.

1 Introduction

This paper presents an algorithm for track displacement based on the Local Design Rule(LDR) concept[1, 2, 3]. LDRs are IC layout rules that define the optimum feature size and spacing in relation to the surrounding geometry and are used to increase the yield of ICs. The yield of circuits fabricated using layouts generated from a Global Design Rule(GDR)[1] set can be increased by searching for instances of local non-optimal layout and adjusting the circuit layout to produce a reduction in the defect sensitivity. In practice this done with a set of LDRs that define a more optimum value of layout geometry size and/or separation in relation to the local layout conditions.

The LDRs do not contradict the GDRs. The GDRs determine minimum size and spacing and hold over the whole design, but they should not be used to determine the maximum feature size or spacing of circuit components by always using the minimum value. While it is valid to attempt an initial layout with minimum sizes to reduce the circuit area, once the area has been defined the best use of any redundant space should be made to reduce the defect sensitivity of the layout.

1.1 Potential Layout Changes.

There are a number of potential layout changes that can be made, the most important for modern MOS fabrication processes are,

Track displacement (fig. 1(b)).
Two metal or other layer tracks can be moved further apart from each other, thus reducing the probability of a short forming between the tracks [4, 5].

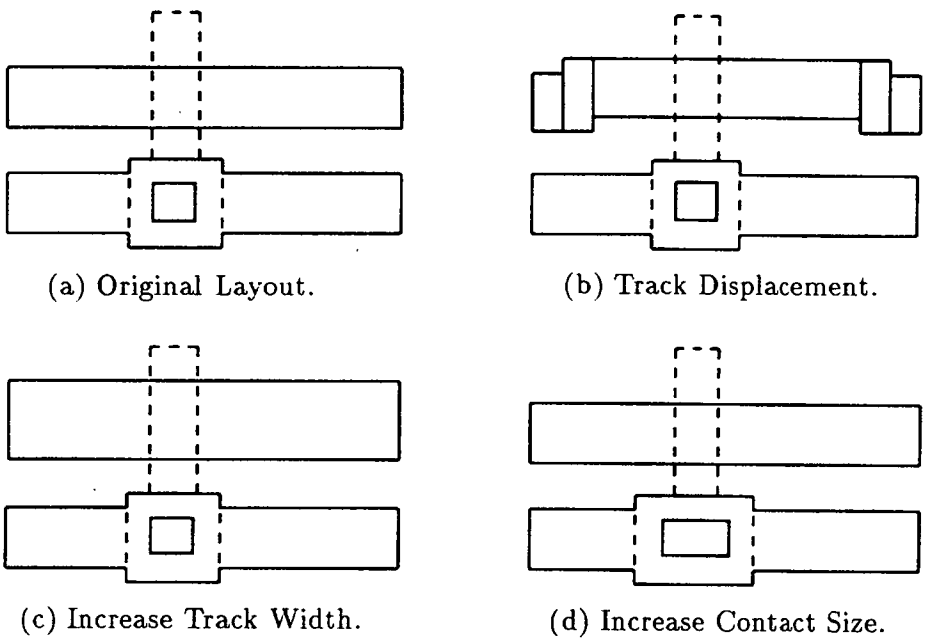
[1]These are just the ordinary layout design rules. In this paper they are referred to as GDRs to distinguish them from LDRs.

Increased track width (fig. 1(c)).

A metal or other layer track can be increased in width. A spot defect on this wider track will be less likely to cause a complete track break [4, 5].

Increased contact size or number of contacts (fig. 1(d)).

A contact can be increased in size[2], since larger contacts have a higher yield [6, 7, 8]. The contact overlap must be adjusted to fit the larger contact to avoid violation of the GDRs. It is also possible to generated another contact where there is sufficient space so that if one contact fails the other can still provide a connection.



(a) Original Layout. (b) Track Displacement.

(c) Increase Track Width. (d) Increase Contact Size.

Figure 1: Potential Layout Changes to Enhance Yield.

2 Track Displacement.

In a mature process a serious cause of yield loss can be shorts between metal tracks. This loss can be reduced, sometimes by a significant amount, by displacing tracks away from surrounding geometry where space permits.

The probability of shorts between same layer geometry such as metal tracks can be reduced by displacing the tracks to increase their separation. Figure 2 shows a simple layout that was simulated using a Monte Carlo yield simulator[2], using a defect size distribution of $\frac{1}{Defect\ Size^3}$[9]. The position of the metal track was varied

[2]Larger contacts, particularly those increased in both the X and Y direction, may require an increase in the overlap of the contact beyond that normally required by the GDRs. This would avoid problems that are created with over etch of large contacts in some processes optimised for the minimum contact and via size.

by increasing the separation. The results of this simulation are shown in figure 3. It can be seen that the defect sensitivity is at a minimum where the track is at the midpoint between the surrounding geometry.

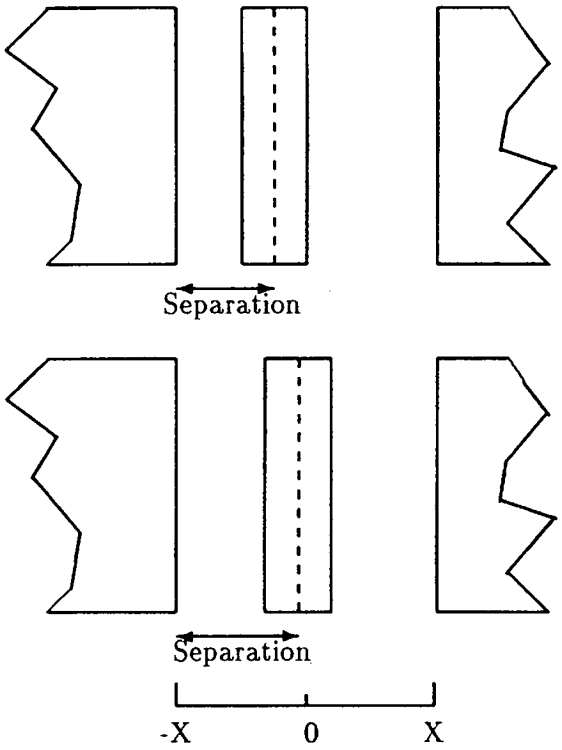


Figure 2: Test for Optimum Separation of Geometry.

Local design rules for track displacement make use of this result. Tracks are displaced away from their close[3] neighbouring same layer geometry until they are no longer close or they are equidistant from the surrounding same layer geometry. Tracks are only displaced away from geometry that is of the same layer and does not form the same electrical node as the track. Where geometry forms the same node a defect would not cause a short since the geometry is already electrically connected.

3 Performance

It is important that any changes that result from the application of LDRs to a layout do not cause a degradation in performance of the circuit. The track displacement results in a longer path, with an increase in RC. This is due to the small step in the track used to implement the displacement (fig. 4). This will not necessarily cause a problem for metal tracks since the increase is small and, except for very long lines, the main resistive components in interconnect are contacts to other layers. Any reduction in performance will be very small and will only be apparent if the critical

[3]The definition of *close* is dependent on the defect size distribution for the process used to fabricate the circuit. This would normally be 2 or 3 times the minimum layer separation.

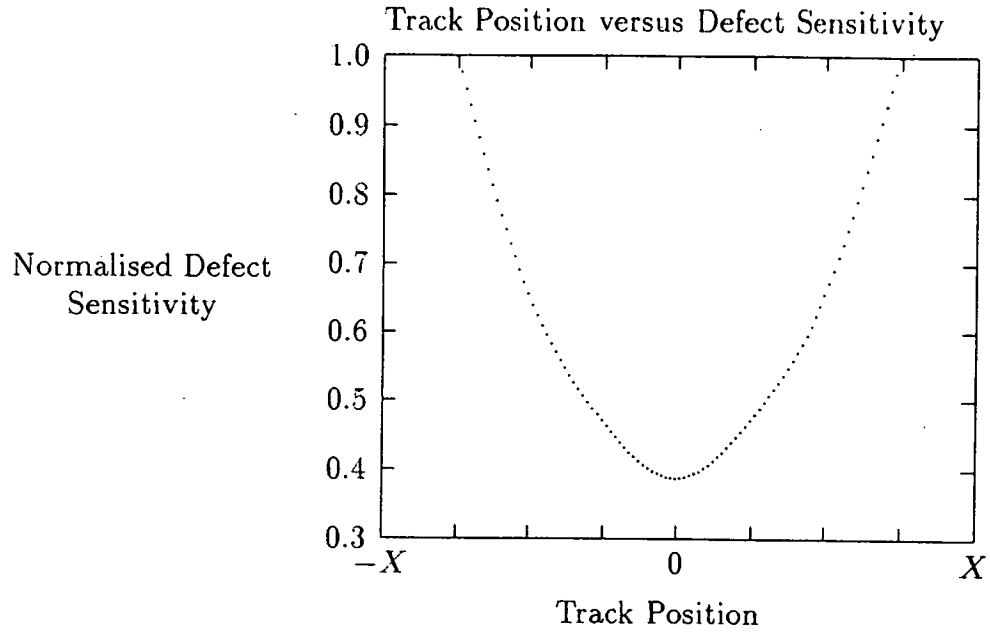


Figure 3: Optimum Geometry Separation Results.

path is affected. If the critical path is known then performance loss can be avoided altogether by ensuring that no significant changes in track length are made on this path.

For the more resistive polysilicon tracks greater care should be taken. It may be best to avoid changes altogether unless significant yield improvements can be made.

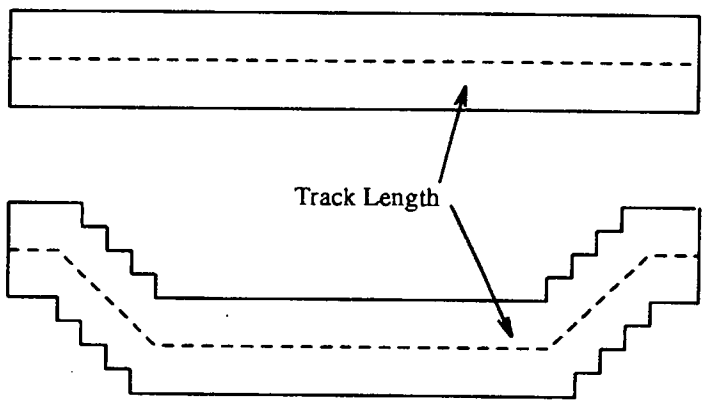


Figure 4: Increased Track Length with Displacement.

4 Applying Local Design Rules.

While it is possible to apply a track displacement LDR by hand, the process would be tedious and error prone. A program, LocDes[2], has been developed that can apply

several different types of LDR to circuit layout. The following (simplified) algorithm is used by this program to apply a track displacement LDR.

4.1 Track Displacement Algorithm

The program assumes that the original layout has been passed by a design rule checker so that rule checking of adjusted layout is limited to the interaction of the adjusted geometry and the immediate environment.

Figure 6(a) gives an example of a metal track being displaced using the algorithm in figure 5. The track is first split into segments. Where there are crossovers

```
Procedure DisplaceLDR Track
    begin
      Split Track into Segments. /* small segments and crossovers */
      for all Segments do {
        Dir = FindMoveDirection(Segment).
        if Dir NOT NULL then {
          if DesignRulesOK(Segment,Dir)
          AND LegalMove(Segment,Dir) then Mark Segment Dir.
        }
        if Segment is Crossover AND is Marked then
          Split Up Segment to give normal Segment on each end (See fig. 6(d)).
      }
      for Direction Up and Down do {
        Make Current Block
        for all Segments do {
          if Segment Marked Direction then Add to Current Block.
          else if Current Block NOT Empty then {
            Add Current block to ListofBlocks.
            Make new Current Block.
          }
        }
      }
      for all Blocks do {
        if Block >= minimum movable size then {
          for all Segments in Block do {
            if first OR last Segment in Block
              then Extent Segment by Displacement LDR in Mark Direction.
            else Displace Segment by Displacement LDR in Mark Direction.
          }
        }
      }
      Merge Segments into Track (see fig. 6(e)).
    end
```

Figure 5: Track Displacement LDR Algorithm

with design rules determining minimum edge separation, larger segments equal to the width of the crossover track plus two edge separations and two associated segments are formed as shown in fig. 6(b). This is the minimum length of track that can be displaced at the crossover without rule violation.

Each of the segments is examined in turn to determine if they can be moved. Normally a segment would be marked for a move when there is more space available on one side of the segment than the other, with the proviso that the displaced segment does not violated any design rules. If a "crossover" segment is marked for displacement, it is divided into three. The sub-segments at either end are the same size as the normal segments with middle part being equal to the remainder of the "crossover". segment defined above (fig. 6(c)). This is to ensure that during displacement the track width is not extended at the crossover point, as this would result in a large segment of the track increased in width. This procedure need only be implemented if there are GDRs associated with edge separation.

The marked segments are then collected into groups of continuous lengths of track that can be moved in the same direction. All of these blocks that are greater than a minimum length are displaced. This minimum, normally greater than the GDR minimum track width, ensures that displacements of very short lengths of track that provide no yield improvement are excluded. The displacement is performed by extending the first and last segments of the block in the displacement direction and displacing, by the same amount, the remaining block segments (fig. 6(d)). The resulting segments are then merged together to give the final layout (fig. 6(e)).

4.1.1 Design Rule Checking

To ensure that any change in geometry size or position does not cause a design rule violation the algorithm of fig. 7 is used. There are two inputs to the procedure. A box representing the new segment to be checked and the segment layer, these were generated from the segment and the displacement direction. The interaction of every layer with the new segment is examined in turn to determine if any rules are violated. This is done by generating a test box for every layer that has a GDR interacting with the segment layer. The dimensions of this test box are dependent on the design rules for the two interacting layers. Any geometry of the segment layer type that has a different electrical node identifier from the segment and falls within or touches the area defined by the test box causes a design rule violation. Where the geometry has the same node identifier it is obviously part of the same track as the segment. Consequently, it is acceptable for such geometry to be closer than the design rule distance or even overlap. For test boxes generated for the interaction of layers that are a different type from the segment layer, any geometry of that layer that falls within or touches the test box area causes a violation. In this case even geometry that forms the same electrical node causes design rule violations.

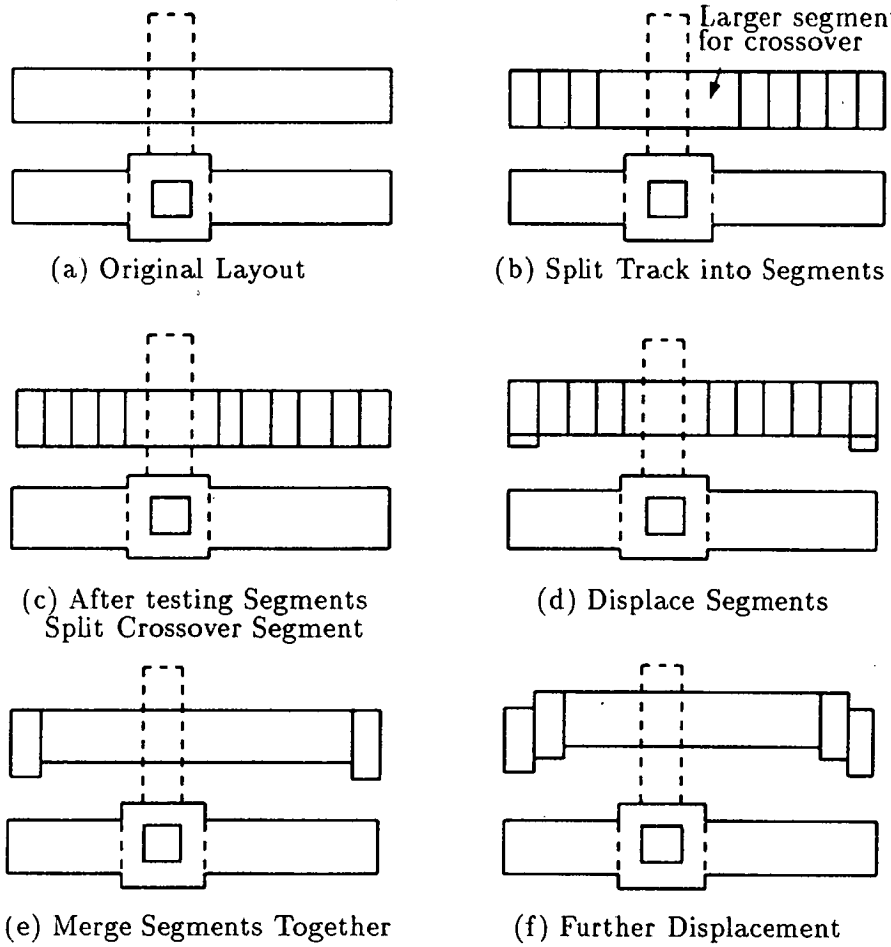


Figure 6: Application of Track Displacement LDR.

```
Procedure DesignRuleCheck Box Layer
          begin
          for TestLayer = all layers do {
              if GDR exists then {
                DesRul = GDR[TestLayer][Layer].
                TestBox = Box + DesRul.
                if TestLayer = Layer then
                  if any different node TestLayer Box touches TestBox
                    then return FAILED. /* Design Rules Violated */
                else if  TestLayer Box touches TestBox
                      then return FAILED. /* Design Rules Violated */
              }
            }
            return SUCCESS
          end
```

Figure 7: Design Rule Checking Algorithm.

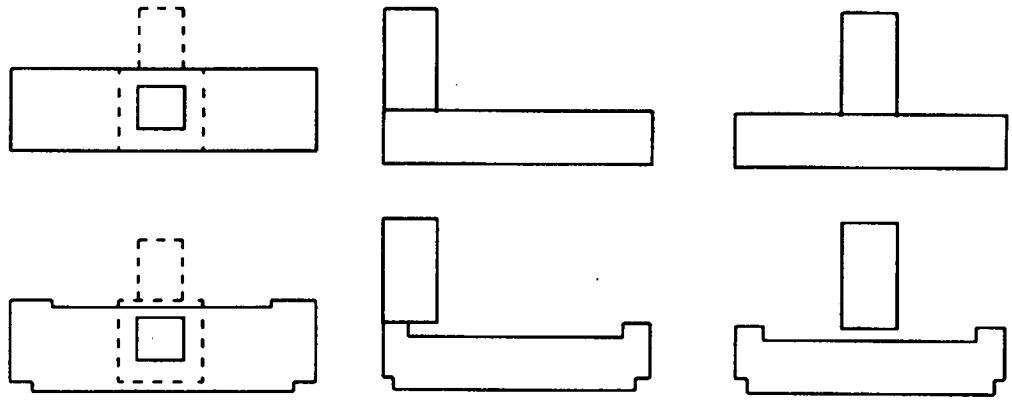


Figure 8: Potential Errors from Track Displacement

4.1.2 Legal Displacements.

The conditions for an acceptable segment displacement are that the segment must not violate any of the design rules or change the netlist (circuit function). The standard design rule algorithm used (fig. 7) is not sufficient as it does not check for the sort of "damage" that could be done by a displaced segment. The additional conditions that need to be checked for are,

- **Violation of Contact Overlap.**
  The overlap of a contact/via can be damaged by displacement of a track that forms the overlap as shown in fig. 8(a). This does not necessarily violate any of the minimum separation design rules and so is not picked up by the standard checking routine.

- **Wire Thinning and Breaks.**
  The displacement of a track can cause thinning at the track corner as shown in fig. 8(b). A more serious and related problem can occur at a branch point of a wire. The track can be displaced as shown in fig. 8(c) causing a break in the connectivity and an almost certain circuit fault. Neither of these condition necessarily violates the design rules as defined by the algorithm of figure 7.

To determine whether a displacement is legal, the side opposite from the direction of segment displacement is checked to ensure that it is not in contact with any other box of the same layer type as itself. If it does touch, then that displacement would cause a thinning or breaking of a wire and is therefore rejected. A similar test is performed to determine whether the overlap would be damaged. If the segment touches any of same layer geometry inside the area occupied by the contact overlap the displacement is rejected. The overlap area is defined by the GDRs.

5 Track Displacement Applied to a Layout Cell

A track displacement LDR was applied to the metal layer of the cell given in figure 9, which is part of an adder circuit, generated using the STICKS [10] circuit compacter. Layout with for 1, 2, ..., 6 iterations of track displacement were produced using the LocDes program. The resulting layout for 6 iterations is given in figure 10. All six layouts were simulated using a Monte Carlo simulator, the results of these simulations are given in figure 11, as the fault size versus the percentage change in fault occurrence. The increase in faults greater than approximately 3.5 $\times$ the minimum defect size is due to the fact that when a track is displaced away from layout geometry it is also displaced towards other geometry, increasing the probability of a short from a larger defect.

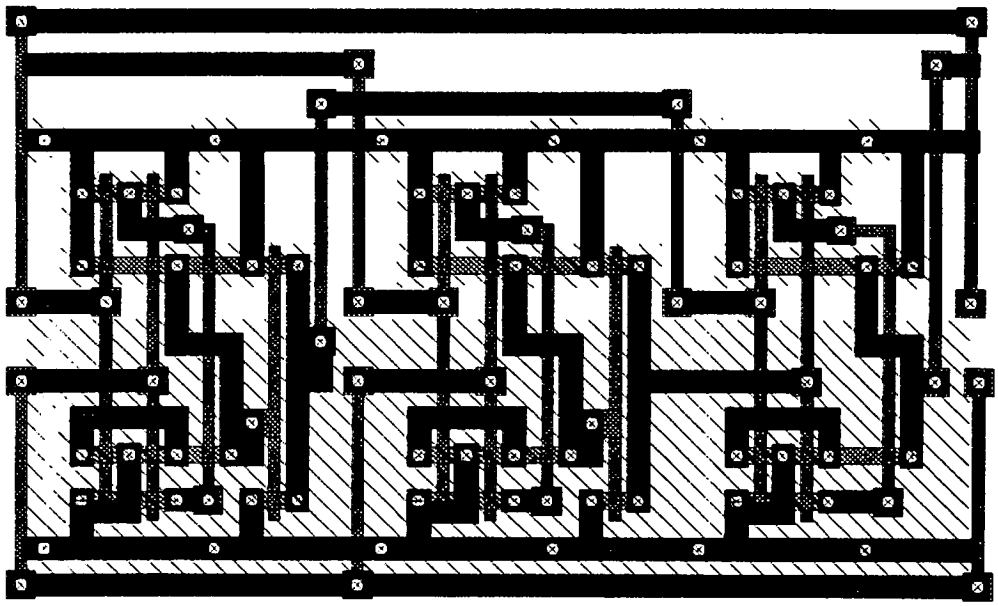


Figure 9: Random Logic Cell

The results of the simulations combined with a defect size distribution for shorts of $\frac{1}{(Defect\ Size)^3}$ [9] give a measure of the reduction in all metal 1 short faults for a cell with track displacement LDRs, this is shown in figure 12. These results indicated that the number of faults per unit area can be reduced to less than 90% of the original value by using more than two iterations of the track displacement LDR on the layout of figure 9.

6 Conclusions

It has been suggested that the yield of IC layout is locally non-optimum and that the yield can be increase by more effective use of device area. We have argued that a more optimum position for tracks is achieved where the track is equidistant from neighbouring geometry. An algorithm to displace tracks to this more optimum

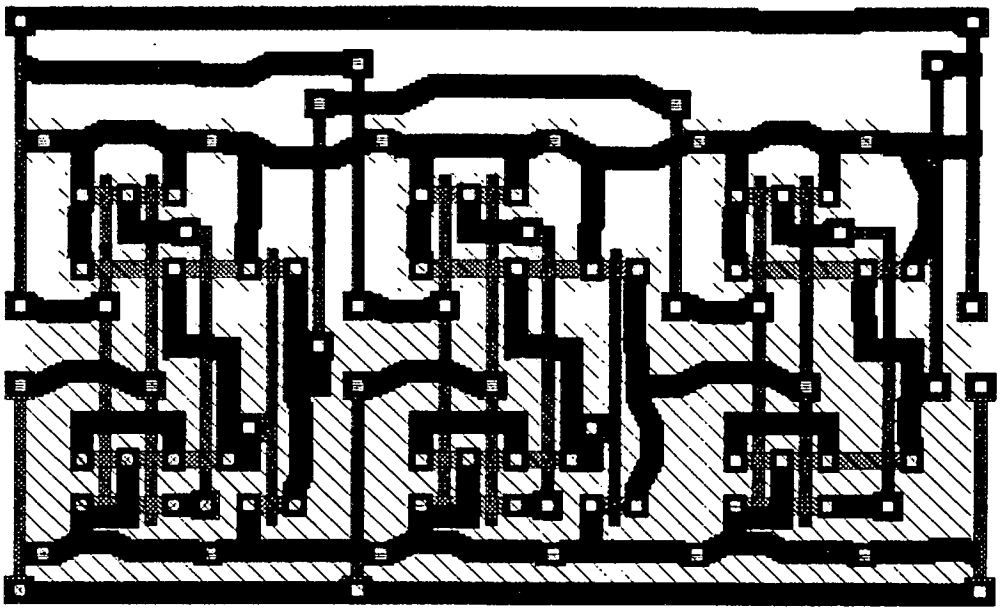


Figure 10: Random Logic Cell After 6 Track Displacement LDR Iterations

position has been presented. Results from simulation of a layout cell to which the track displacement algorithm has been applied suggest that a reduction in the number of metal faults(shorts) in the order of 10% can be obtained for at least some layouts.

References

[1] G. Allan, A. Walton, and R. Holwill. Yield improvement with local design rules. In *IEEE Workshop on Defect and Fault Tolerance in VLSI Systems*, Institut National Polytechnique de Grenoble, Nov 1990.

[2] Gerard A. Allan. *Phd Thesis: Yield Improvement with Local Design Rules in VLSI.* University of Edinburgh, 1991.

[3] G. A. Allan, A. J. Walton, and R. J. Holwill. A novel yield improvement technique for IC layout. *IEEE Trans Comp. Aided Design*, To be published.

[4] W. Maly. Yield estimation model for VLSI artwork evaluation. *Electronics Letters*, 19(6):226–227, Mar 1983.

[5] W. Maly. Modeling of lithography related yield losses for CAD of VLSI circuits. *IEEE TRANS. on Computer Aided Design*, CAD-4(4):166–177, Jul 1985.

[6] A. C. Ipri. Impact of design rule reduction on size, yield, and cost of integrated circuits. *Solid State Tech*, Feb 1979.

[7] A. C. Ipri. Evaluation of CMOS transistor related design rules. *RCA Review*, 41, Dec 1980.

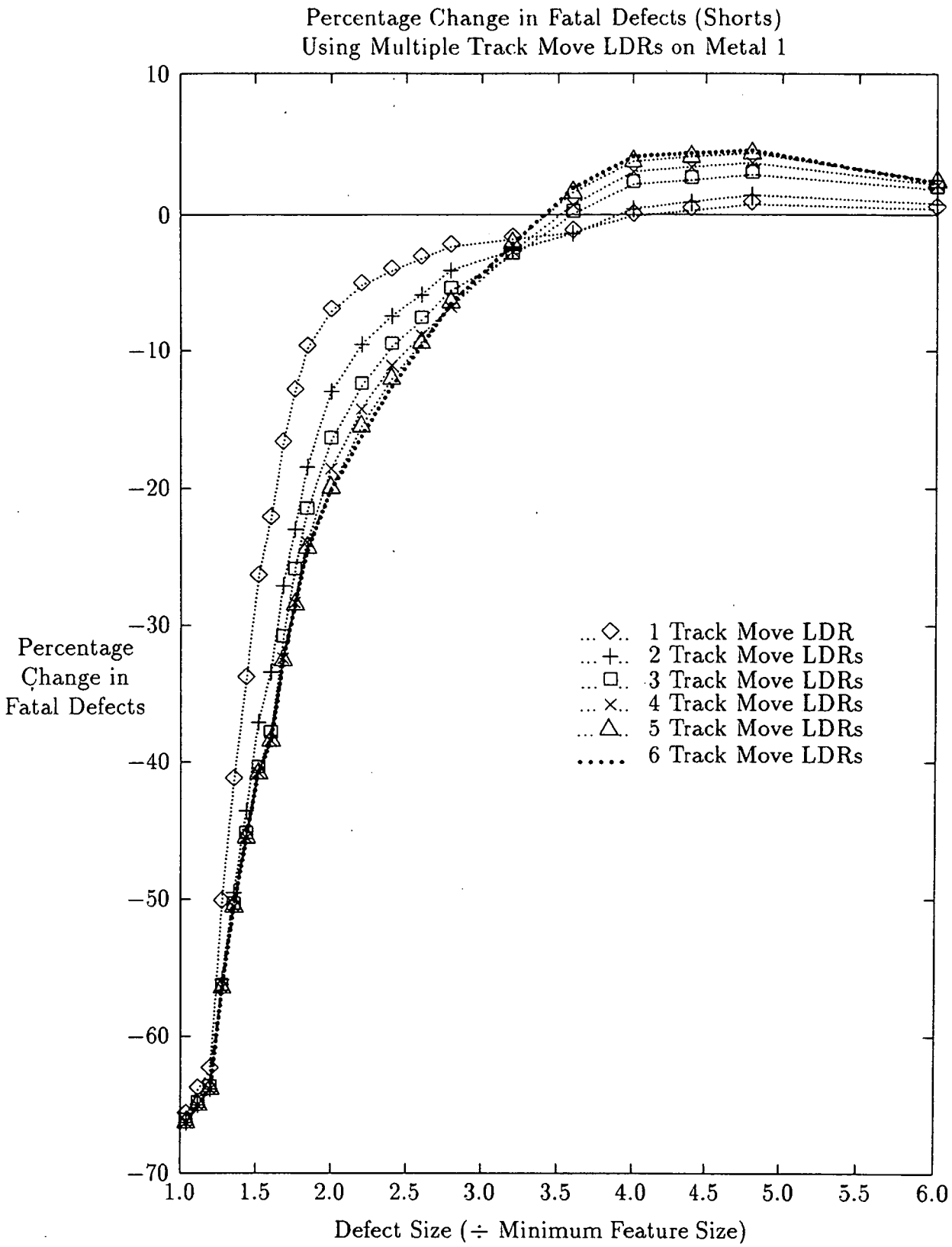


Figure 11: Percentage Change in Defects from Track Move LDRs

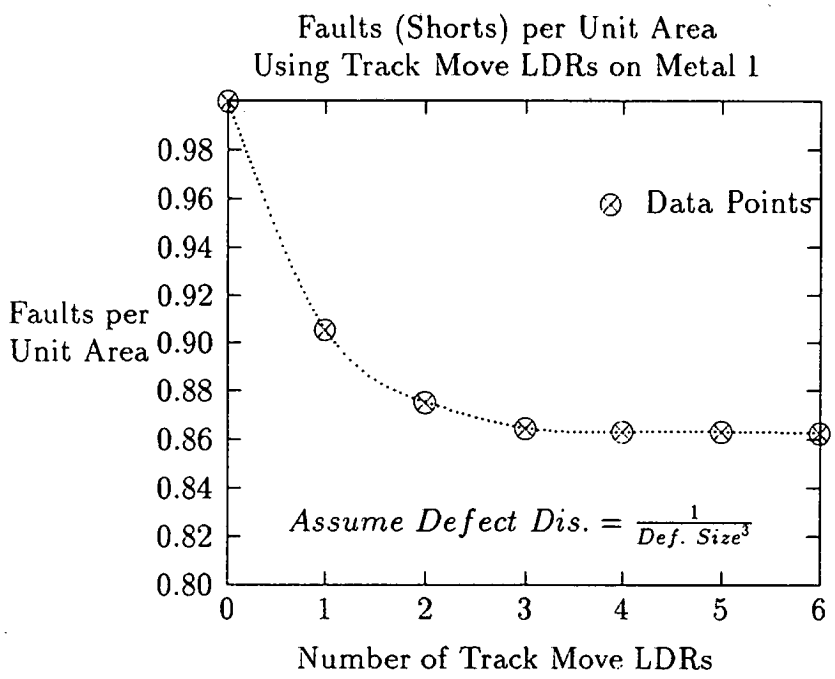


Figure 12: All Short Fatal Defects Using Track Displacement LDRs

[8] R.D. Rung. Determining IC layout rules for cost minimisation. *IEEE J. Solid-State Circuits*, pages 35–43, Feb 1981.

[9] C. H. Stapper. Modeling of integrated circuit defect sensitivities. *IBM J. Res. Develop.*, 27(6):549–557, Nov 1983.

[10] L. Walczowski, W. Waller, and E. Wells. *CHIPWISE: An integrated toolset for full custom VLSI.* The ECAD Laboratory, The University of Kent, Canterbury, Kent.

A Yield Improvement Technique for IC Layout using Local Design Rules

G. A. Allan, A. J. Walton, and R. J. Holwill.

A yield improvement technique for IC layout using local design rules.

IEEE Trans Comp. Aided Design, To be published.

A Yield Improvement Technique for IC Layout using Local Design Rules

Gerard A. Allan A. J. Walton R. J. Holwill

Edinburgh Microfabrication Facility
University of Edinburgh
Fax no 031-662 1081
Email gaa@uk.ac.ed.ee
Aug 24, 1991

Abstract

This paper introduces the concept of Local Design Rules. These are IC layout rules that define the optimum feature size and spacing in relation to the surrounding geometry and are used to increase the yield of ICs. The impact of these rules on the performance and reliability of ICs is discussed. Algorithms that enable the automatic application of track displacement, track width and contact size local design rules to IC layout are presented. Simulation results are provided for some layout examples.
Keywords: Local design rules, yield improvement, CAD, layout.

1 Introduction

The layout of integrated circuits is bound by a set of design rules. These rules, determine the minimum size and spacing of all layers of the circuit geometry in an attempt to maximize the yield, performance and reliability. There are occasions when specialist circuit parts, such as RAM, use a different set of design rules that have been optimized for that particular circuit type. However, normally the design rules are applied over the whole of layout area and will therefore be referred to as Global Design Rules (GDR).

The GDRs have been optimized to give good layout from a single global set of rules but are not necessarily optimized for the local,layout conditions. For example, the conductor width and contact size can have a significant effect on the yield of a circuit [?, ?, ?]. Under certain local conditions, where there has been incomplete use of the surrounding area, it would be advantageous to adjust the conductor width and contact size [1] to greater than the global norm. Wider tracks and larger contacts

[1] Larger contacts, particularly those increased in both the X and Y direction, may require an increase in the overlap of the contact. This avoids problems that may be created with over etch of large contacts in a process optimized for the minimum contact and via size.

have a higher yield, if they are sufficiently distant from other geometry. It is also possible to displace track away from neighboring geometry to increase the separation and thus the possibility of shorts. The Local Design Rules (LDR) define the best values of track width, contact size/overlap in relation to the local layout conditions.

The global rules determine minimum size and spacing and hold over the whole design, but these rules should not be used to determine the maximum feature size of circuit components. While it is valid to attempt an initial layout with minimum sizes to reduce the circuit area, once the area has been defined the best use of any redundant space should be made.

There are a number of potential layout changes that can be made,

- Track displacement.
- Increased contact size or number of contacts.
- Increased contact overlap.
- Increased track width.

It is suggested that the yield of circuits fabricated using layout generated from a GDR set can be increased by searching for instances of local non-optimal layout, applying a LDR set and adjusting the layout to meet the new design rules. The use of automated layout generation and compaction produces designs that are less dense than traditional hand-crafted layout and give greater scope for yield enhancement with LDRs.

Applying LDRs by hand is too time consuming and prone to error. To address this problem a program (LocDes) acting as a post-processor of Caltech Intermediate Format (CIF) layout has been developed. This uses the GDR layout to produce an enhanced circuit layout, with the same function and performance specifications but with a higher yield. A more efficient approach would be to use a higher level of abstraction than provided by CIF, this would require the issue of yield to be addressed earlier in the design cycle[?].

While it is intended that the yield of the resulting layout will be greater than the initial GDR layout, this can only be guaranteed if the fabrication process is understood well enough to ensure that the LDRs are an accurate reflection of the relative yield of the layout options under test. That is, no changes are made to the layout except where there is good evidence to suggest that a higher yield can be obtained.

2 Defining the Local Design Rules.

Local design rules are used to determine where changes in layout generated from GDR sets should be performed. LDRs do not define minimum widths and separations, since this is already done by the GDRs. Their purpose is to define maximum feature sizes, in relation to the available space around the geometry under consideration. The LDRs are derived from a comparison of the yields of a number of layout options.

The complete definition of the LDRs requires a greater knowledge of the process than the GDRs, since the problem is no longer a "simple" matter of finding one rule set to maximize yield of regular test structures. Rules to determine the optimum layout in a variety of situations need to be determined. This in practical terms means that it must be known where an increase in track width or contact size will result in an increase in yield, since associated with these adjustments is a reduction in track separation. The information required can be found using test structures as is the normal procedure for GDR generation. This will require a greater range of test structures, but there may be sufficient information to make a first approximation of LDRs from the results of test structures designed to determine GDRs.

2.1 LDRs from Defect Size and Distribution.

Yield simulation with defect size and distribution data[?, ?, ?] obtained from test structures[?, ?] can, for example, be used to determine where a change in conductor width will increase the overall yield. The number of faults that are the result of shorts between nodes is increased by widening the conductors but the number caused by breaks is reduced. If the overall number of faults can be decreased then there will be a corresponding increase in yield.

A number of LDRs can be derived from the results of the simulation, depending on the sophistication of the simulator available.

3 Performance

It is important that any changes that result from the application of LDRs to a layout do not cause a degradation in performance of the circuit. For this reason no changes that affect the size of active devices should be made. The effects of changes in track width, separation, contact and substrate contact size must be considered.

Increased contact size.

Increased contact/via size will result in a decreased contact resistance and increased overlap capacitance. The effect of overlap capacitance can be neglected because it is effectively shorted by the contact; the RC product will therefore be reduced in line with any reduction in the total contact resistance [?].

Increased substrate contact size can take place where the contact is sufficiently distant from active devices. Since they are not used to pass signals there will be no effect on performance of the circuit.

In many GDRs it is recommended that a number of small contacts are preferred to one large contact. Once contacts reach an appropriate size they can be split to accommodate this rule.

Track displacement.

The displacement of tracks in general results in a longer track, with an increase in RC. This should not be a problem for metal tracks since the increase is small

and, except for very long lines, the main resistive components in interconnect are the contacts to other layers. However, for the more resistive polysilicon tracks care should be taken with critical circuit parts.

Increased track width.

Increased track width will cause an increase the capacitance of the circuit but will also decrease the resistance of the circuit. The RC product will be reduced because the fringe capacitance, which, for narrow tracks is a significant fraction of the total, is not increased. Capacitance between tracks on the same conductor level will not be larger than that allowed for in the GDRs.

There is a possibility of greater cross talk where the capacitance between crossovers is increased. This can be avoided by ensuring that overlapping geometry, not forming a single electrical node, remains unaltered.

If timing paths are critical this must be taken into consideration when layout adjustments are performed. For example fan out from a single inverter may require that the signal reaches the input to the next gates simultaneously. The interconnect forming these time critical paths will require special treatment.

3.1 Reliability

The reliability of a component is also of importance. There is little point in increasing yield at the price of the component reliability. The reliability of the circuits optimized using LDRs can also be improved. This requires that when the LDRs are calculated reliability issues are taken into account and that no layout change is made that would cause yield or reliability to be reduced.

There are occasions when high reliability is of greater importance than yield and LDR sets that increase reliability at the expense of yield can be developed.

4 The LocDes Program

The program takes the original layout, in CIF format, and processes it using a set of LDRs. It is basically a design rule checker that attempts small changes in layout based on the LDRs, accepting only those changes that do not violate any of the global or other local design rules.

The program uses a heuristic algorithm, that makes adjustments iteratively, such that changes in layout geometry occur in a number of small steps with attempts to change other neighboring geometry before any further change. This is to ensure that there is an even application of LDRs since a large number of small changes evenly applied give greater yield improvement than a small number of large changes. While this will not necessarily give the optimal layout, it will produce a good approximation to it in a reasonable time.

4.1 Algorithms

The following (simplified) algorithms[?, ?] are used by the program to adjust layout. The program assumes that the original layout has been passed by a design rule checker so that rule checking of adjusted layout is limited to the interaction of the adjusted geometry box and the immediate environment.

4.1.1 Increase Track Width

The widths of tracks can be increased where local and global design rules permit. This will reduce the probability that a track will break when fabricated; it will also increase the risk of shorts between adjoining tracks. Many [2] MOS processes will have greater yield loss from shorts than from breaks so care must be taken in defining track width LDRs.

A small example layout is shown in figure 1(a). The bottom metal track has a track width LDR applied to it. The track is split into segments (figure 1(b)), each segment is tested in turn. If there is space above or below the segment greater than that required for the GDR and the LDR separation a new wider segment is generated. All the design rules for the new larger segment are checked and if there are no violations the change in width is accepted (figure 1(c)). The resulting segments are then merged together (figure 1(d)). The algorithm used can be expressed as shown in figure 2.

4.1.2 Increase Contact Size

Yield loss from contacts can be a significant proportion of total yield loss. Using larger contacts where there is available space can reduce this loss. Where contacts are increased the overlap layers must also be adjusted to at least the minimum overlap. This will give rise to some yield loss due to an increase in the probability of shorts between these layers and the surrounding layout geometry.

An example contact layout is given in figure 3(a). An LDR adjustment has been applied to this contact to give figs 3(b) and 3(c). A contact can be increased in four directions ($\pm$ X and Y) with the directions being tested in turn to determine if an adjustment can be made. In order for a contact change to be acceptable each of the layers forming the contact must be checked to ensure that there is sufficient space to extend them without design rule violations. If all layers can be suitably modified then the contact and corresponding overlap layers are extended. Further attempts to increase the contact, can be attempted after the same procedure has been applied to the other contacts in the layout. The algorithm used is given in fig 4.

4.1.3 Track Displacement

In a mature process a serious cause of yield loss can be shorts between metal tracks. This loss can be reduced, sometimes by a significant amount, by displacing tracks

[2] No two fabrication processes are identical but there are features common to many mature production lines. Process that are at an early stage of development will often exhibit unique sources of yield loss

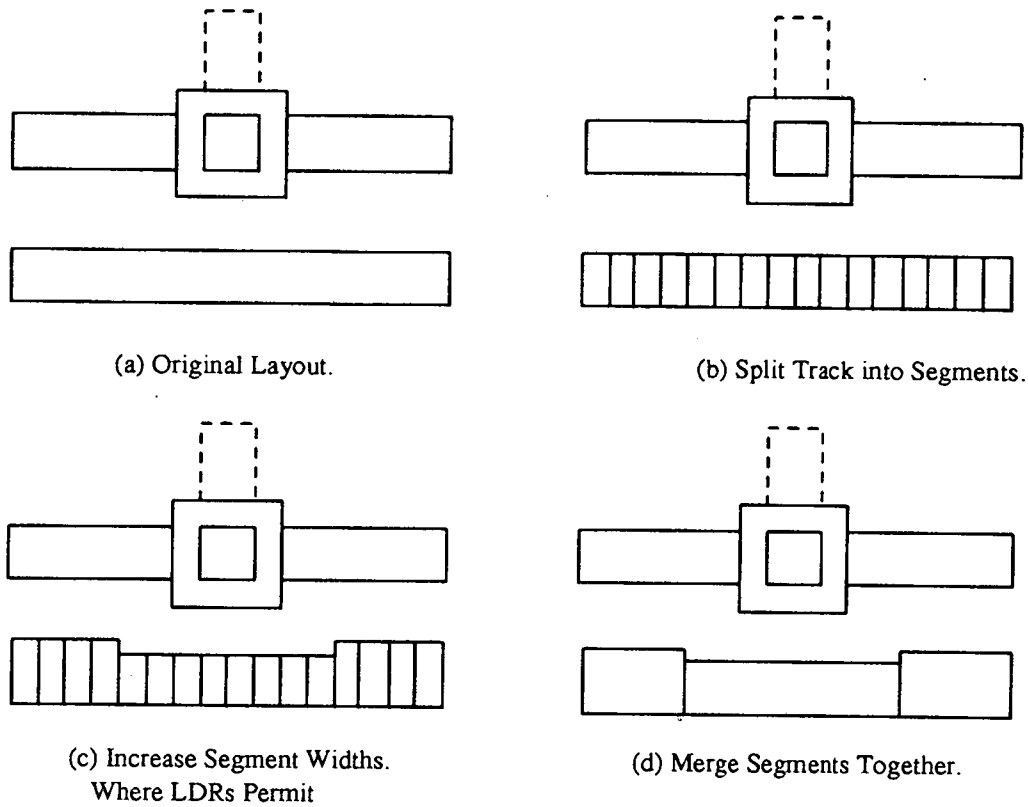
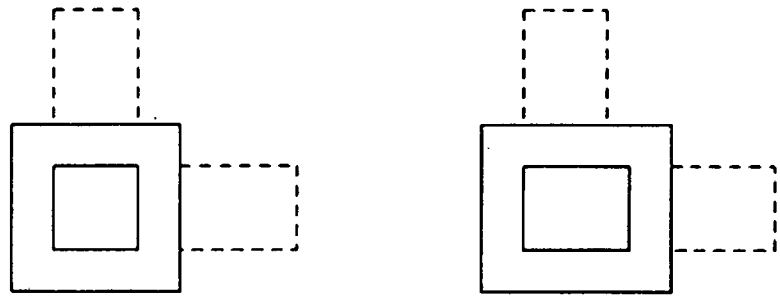


Figure 1: Application of Track Width LDR

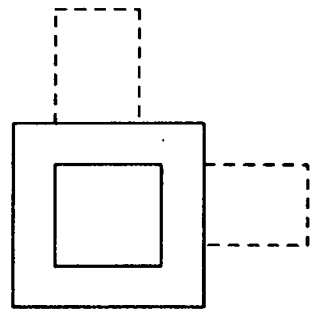
Procedure **WidthLDR** Track
begin
Split Track into Segments (see fig. 1(b)).
for all Segments **do** {
if (Space Above Segment >= $LDR + GDR$) **AND**
(DesignRuleCheck = OK) **then** Increase Segment Size (Top).
if (Space Below Segment >= $LDR + GDR$) **AND**
(DesignRuleCheck = OK) **then** Increase Segment Size (Bottom).
}
Merge Segments (see fig. 1(d)).
end

Figure 2: Algorithm for Track Width LDR



(a) Original Layout

(b) Increase in +X



(c) Increase in +Y

Figure 3: Application of Track Contact LDR

```
Procedure TryChangeContact Contact
          begin
            for all possible Directions do {/* +X -X +Y -Y. */
            Success=FALSE.
              for all Layers forming Contact do { /* Contact, Metal, Poly etc. */
                if Space In Direction for Layer >= LDR[Layer] + GDR[Layer])
                  AND (DesignRuleCheck = OK ) then Success=TRUE.
                else {
                  Success=FALSE.
                  Exit Layer for Loop. /* Failed */
                }
              }
              if Success= TRUE then Increase Contact in Direction.
            }
          end
```

Figure 4: Algorithm for Contact LDR

away from surrounding geometry where space permits.

Figure 5(a) gives an example of the top metal track being displaced using the algorithm of figure 6. The track is first split into segments. Where there are crossovers with design rules determining minimum edge separation, larger segments equal to the width of the crossover track plus two edge separations and two associated segments are formed as shown in figure 5(b). This is the minimum length of track that can be displaced at the crossover without rule violation.

The first and last segments are fixed, which ensures that external connections to the layout will remain correct. Each of the other segments is examined in turn to determine if they can be moved. Normally a segment would be marked for a move when there is more space available on one side of the segment than the other, with the proviso that the displaced segment does not violate any design rules. If a "crossover" segment is marked for displacement, it is divided into three. The sub-segments at either end are the same size as the normal segments with middle part being equal to the remainder of the "crossover". segment defined above (figure 5(c)). This is to ensure that during displacement the track width is not extended at the crossover point. This procedure is only implemented if there are GDRs associated with edge separation. The marked segments are then collected into groups of continuous lengths of track that can be moved in the same direction. All of the blocks that are greater than a minimum length are displaced. This minimum, normally greater than the GDR minimum track width, ensures that displacements of very short lengths of track that provide no yield improvement are excluded. The displacement is performed by extending the first and last segments of the block in the displacement direction and displacing, by the same amount, the remaining block segments (figure 5(d)). The resulting segments are then merged together to give the final layout (figure 5(e)).

4.2 Layout Data Structure

The execution time of these algorithms is heavily dependent on the internal organization of the layout geometry. A spatial data structure capable of representing IC layout in a space efficient manner with a fast method for identifying neighboring geometry is essential.

- **Space Efficiency.**
  The amount of data to be manipulated depends on the circuit layout. Since the layout hierarchy must be flattened, a small CIF file can result in a very large amount of layout data.

- **Fast Region search.**
  The program makes a large number of region searches to determine the position of neighboring geometry. The amount of geometry within the region of design rule interaction is very small in comparison to the total. It is important that this geometry be found efficiently.

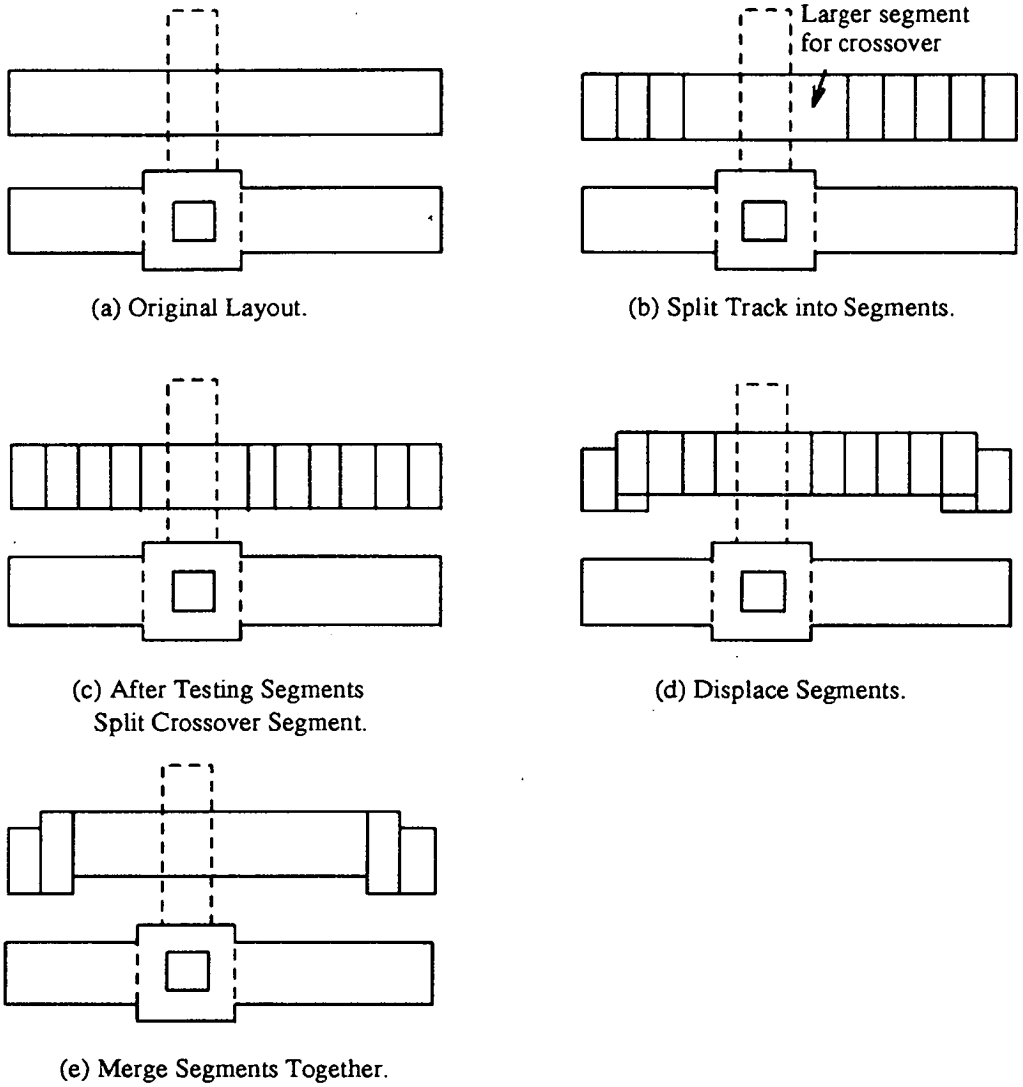


Figure 5: Application of Track Displacement LDR

```
Procedure DisplaceLDR Track
        begin
          Split Track into Segments. /* small segments and crossovers */
          for all Segments except first and last do {
            if CheckMoveUpOK then Mark Segment Up.
            else if CheckMoveDownOK  then  Mark Segment Down.
            if Segment is Crossover AND is Marked then
              Split Up Segment to give normal Segment on each end (See fig. 5(d)).
          }
          for all Segments except first and last do {
            if  Segment Mark Up then Add to Current Block.
            else if Current Block NOT Empty then Make new Current Block.
          }
          for all Segments except first and last do {
            if Segment Mark Down then Add to Current Block.
            else if Current Block NOT Empty then Make new Current Block.
          }
          for all Blocks do {
            if Block >= minimum movable size then {
              for all Segments in Block do {
                if first OR last Segment in Block
                  then Extent Segment by Displacement LDR in Mark Direction.
                  else Displace Segment by Displacement LDR in Mark Direction.
              }
            }
          }
          Merge Segments into Track (see fig. 5(e)).
        end
```

Figure 6: Algorithm for Track Displacement LDR

4.2.1 The Adaptive Multiple Storage Binary Tree

There are a number of well known methods of representing mask-level physical layout. These are important in many applications, with the most notable being circuit compaction.

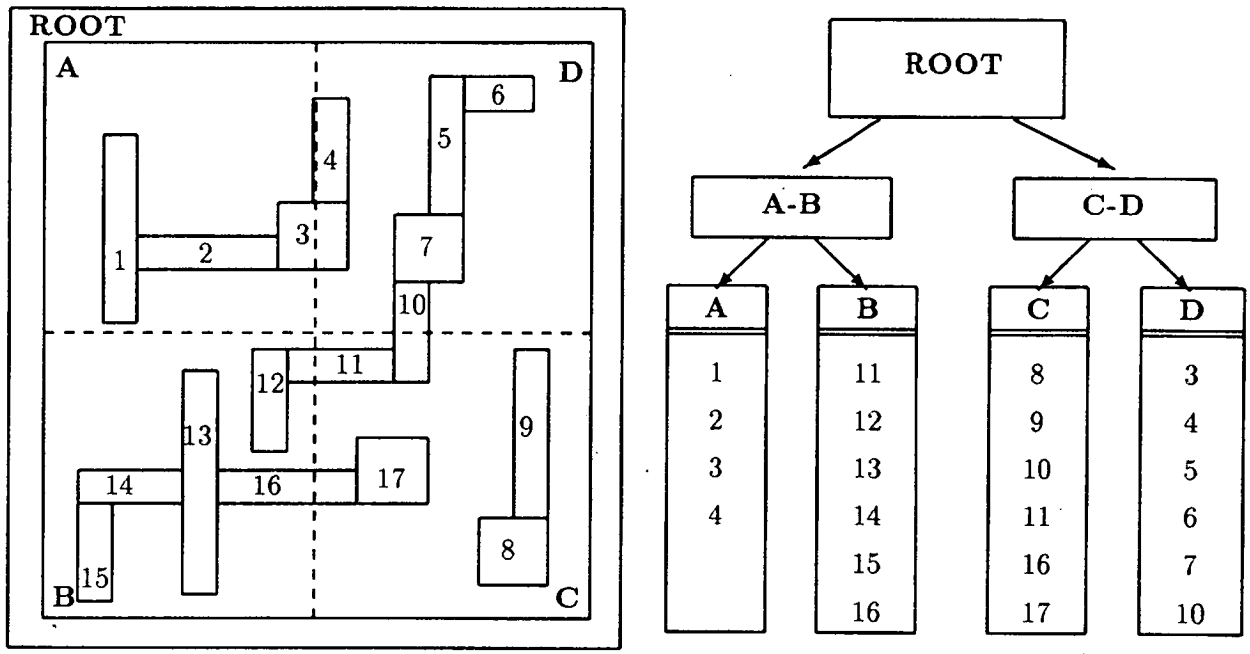


Figure 7: Mask Layout and Data Structure

The spatial data structure chosen was the Multiple Storage(MS) binary tree which is a variant of the MS-quad tree [?]. It is more properly described as an adaptive multiple storage binary tree and is identical to the MS-quad tree, except that each node points to two further node/leaf structures rather than four. This simplifies the algorithm, and has no significant effect on execution time or memory requirement.

The MS-tree stores the layout geometry (rectangles) in a tree structure. Each node represents an area and can either point to two sub-areas or contain an array of references to all the geometry in that area, in which case it is a leaf cell. If a new item is added to a leaf cell and the reference array is already full, the area is split and all items in the existing array are repositioned in the appropriate sub-area leaf arrays.

If geometry straddles more than one area defined by a leaf cell, each leaf stores a copy of the reference to it. This does not uses as much memory as might be thought at first sight, as only the index or pointer to the geometry is stored. The geometry itself is stored in a dynamic array and referenced by an index. In fact the MS-quad tree has about the same memory usage as the bisector list quad tree [?], a simpler data structure and faster execution time [?].

Each physical mask layer is stored in a separate dynamic array and an associated binary tree that references by indexes the data stored in the array. An example of a mask layout is given in figure 7 with the resulting data structure.

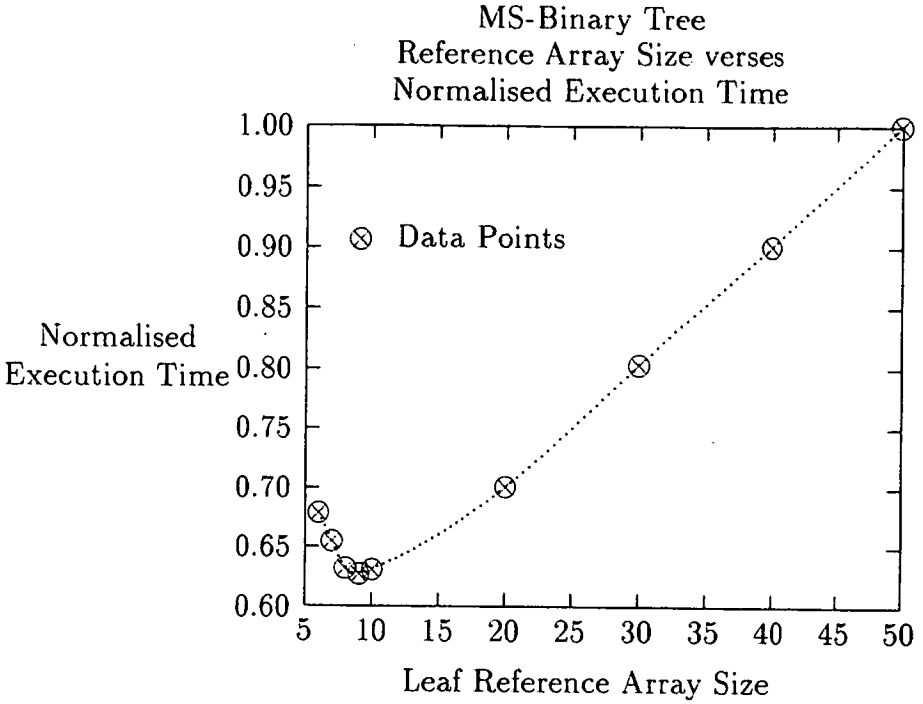


Figure 8: Reference Array Optimization

The size of the reference array has a marked affect on the efficiency of the program. The optimum value was determined experimentally (figure 8).

4.3 Large Circuit Layouts

Even using the most efficient data structures there will always be circuit designs that are too large to be processed in one piece. If circuit layout exceeds a specified limit, the program divides the layout into a number of segments. The layout can be processed in 1, 2, n^2 n=2, 3, ..., segments depending on the size of design. Each segment overlaps with its neighbors to at least twice the maximum design rule operating distance, to ensure global and local design rules are not violated. All the individual segments are processed, the overlaps deleted and the results appended to a single output file.

4.3.1 Distributed Processing

To reduce the time taken by the program to process large circuits the program includes a facility to use multiple hosts to process a single circuit layout.

Segments are assigned to remote hosts to be processed. Only one segment is assigned to each host. If there are more segments than available hosts, the controlling host waits until a remote host has finished processing its segment, before sending another one. Thus it is possible to use as many remote hosts as there are segments, or to use a limited number of hosts efficiently, each processing only one segment at a time. The results generated by the remote hosts are sent back to the controlling host

where they are merged to form the final result.

4.3.2 Execution Time

The LocDes program execution time is dependent on both the specific layout and the design rules. Average execution times gives per Sun 4/60 CPU second, 180 attempted contact changes or 1.5mm length of track checked for displacement or width increase. Since as many as four iterations are required for the best results, the total execution time can be a number of hours for large circuits.

4.3.3 User Interface

The LocDes program has been provided with X window user interface (figure 9), that allows the user to apply LDRs to individually selected pieces of circuit geometry. In this case LDRs have been applied to one track and a contact. It should be noted that for this track displacements the GDRs have not required an edge separation with the underlying polysilicon. The interface was built using ET++[?], an object oriented application framework, implemented in C++. This enabled a menu driven interface to be built in a very short time.

This type of tool can be used to apply LDRs to the layout of individual cells forming the database of a cell based automated layout tool. The application of LDRs can be controlled to avoid design rule violation across cell boundaries and can ensure that there will be no performance degradation of the resulting cell. This is done by applying LDRs selectively within the cell boundary keeping all geometry changes within the original GDR envelope and taking care that changes, along the critical path, do not result in a deterioration of performance.

5 Results

The effectiveness of LDRs is highly process and to some degree circuit dependent. They are best suited for fabrication processes that have,

- yield loss from shorts between tracks,
- yield loss due to poor contacts,
- yield loss from metal breaks.

The LDR approach is ideally suited to circuits that are, digital in nature, with inefficient layout eg. from automated cell layout and circuit compactors. Significant gains can be made where there is volume production of the circuit.

5.1 Change in Track Width

The result of a Monte Carlo simulation of yield of metal 1 stage for a dynamic shift register cell, generated using the STICKS [?] circuit compacter give the results

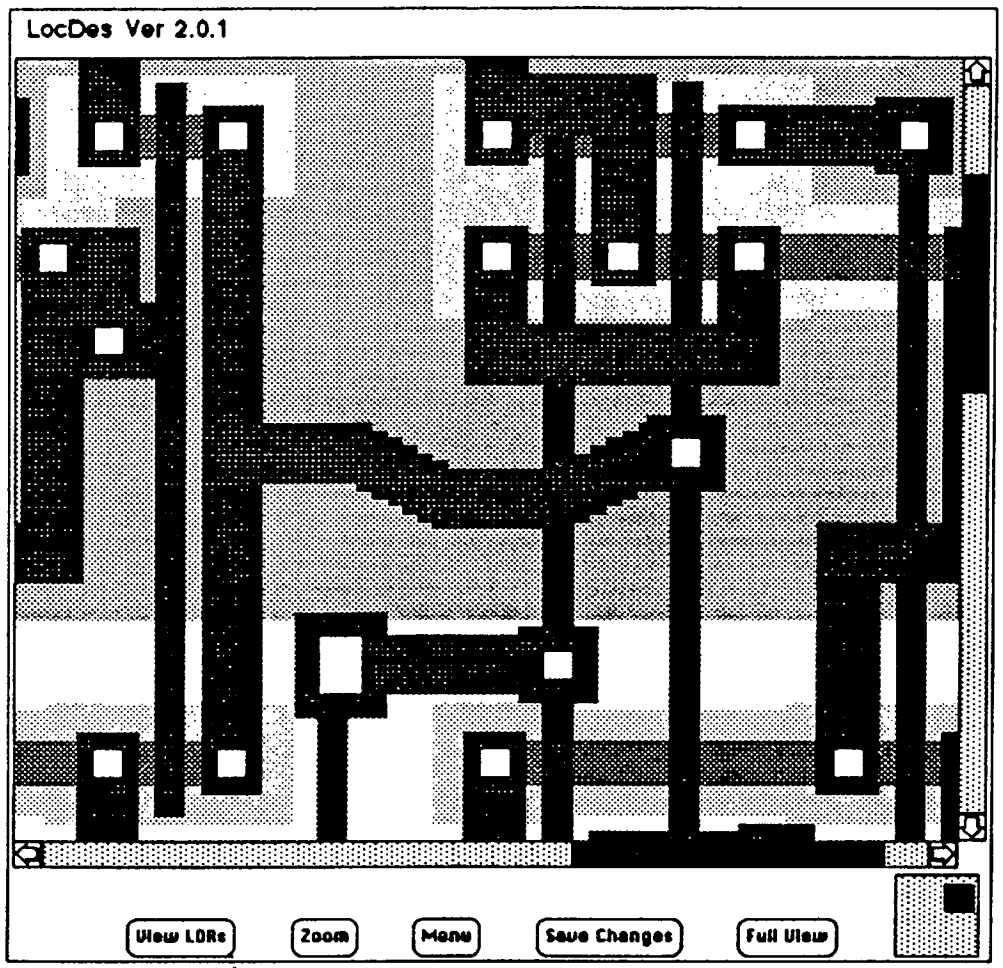


Figure 9: LocDes User Interface

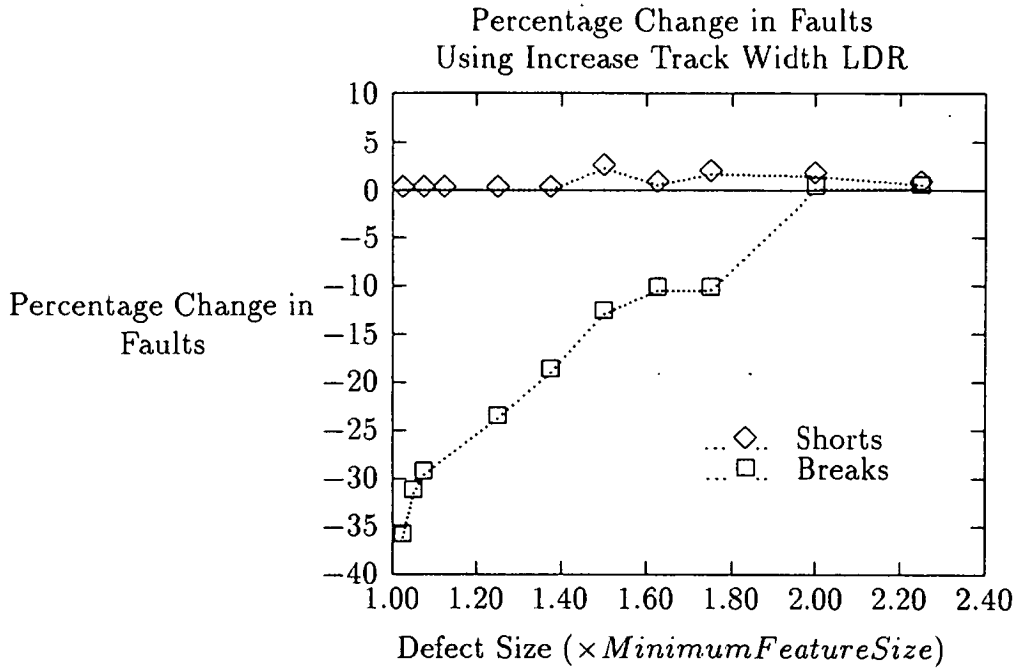


Figure 10: Percentage Change in Faults for Metal 1 LDR

shown in figure 10. This gives the percentage change in the number of faults after the application of a LDR. The rule can be expressed as: Metal 1 is increased up to 1.5× minimum metal 1 width so long as 1.5× minimum separation is kept between unrelated metal 1 and no GDRs, involving metal 1 with any layer, are violated.

The graph indicates that the probability of a spot defect causing a break can be reduced by applying this LDR. The data combined with the process defect size and distribution for metal 1, is used to determine if the LDR circuit will have a higher yield when fabricated with that process.

Monte Carlo simulations are CPU intensive, so a more efficient approach would be to use an analytical yield simulator such as RYE [?].

5.2 Change in Contact Size

The LDRs for contacts allow for bigger contacts. As an example table 1 gives the results of application of five iterations of a single LDR to a mixed logic circuit and a PLA. Each iteration attempts to increase the contacts size in four directions (ie. ±X and ±Y) by 1/6th of the minimum contact size. The LDR can be stated as: Contacts are increased in size, so long as there is no violation of any GDRs including those governing contact overlap. More complex LDRs involving contact size and overlap can be evolved to accommodate problems associated with over etch of large contacts cuts.

The table lists the percentage of total contacts that were increased in a number of directions per iteration. The results show that a large percentage of contacts can be increased in size without violating GDRs, and that this percentage varies with the

Contact Changes Per Iteration (%)					
Mixed Logic					
	Number of Changes in $\pm X/Y$				
Iteration	0	1	2	3	4
1st	4.3	19.5	31.6	31.0	13.6
2nd	12.9	36.3	28.9	19.6	2.3
3rd	22.5	38.3	22.8	15.1	1.3
4th	29.6	40.5	22.2	6.8	1.0
5th	35.4	40.8	17.7	6.1	0.0
PLA					
	Number of Changes in $\pm X/Y$				
Iteration	0	1	2	3	4
1st	0.0	15.1	30.2	41.5	13.2
2nd	1.9	19.8	35.8	32.1	10.4
3rd	2.8	23.6	36.8	26.4	10.4
4th	2.8	27.4	39.6	19.8	10.4
5th	6.6	25.5	41.5	16.0	10.4

Table 1: Percentage of contact changes in each iteration of a contact LDR

type of circuit. As in section 5.1 above, the usefulness of these result depends on the fabrication process.

5.3 Track Displacement LDR

The LDRs for track displacement allow tracks to use any available space to increase their separation. When a track is displaced, the moved track length can be tested to determine if a further movement of parts of that segment is possible. The nesting of track displacement can be continued to as many levels as required. Figure 11 gives Monte Carlo yield simulations results for a CMOS cell (figure **??**), with up to 6 levels of nesting.

The results of the simulation combined with a defect size distribution for shorts of $\frac{1}{(Defect\ Size)^3}$ [?] give a measure of the reduction in all metal 1 shorts for a cell with track displacement LDRs. Figure 12 shows that the number of defects per unit area can be reduced to less than 90% of the original value by using the layout in figure **??**.

6 Conclusions

It has been suggested that the yield of a typical IC layout is not necessarily optimum. The yield can be increased by more effective use of silicon area through the application of local design rules to layouts that have been generated from the normal "global" design rules. These rules, which are dependent on the fabrication process, can be found using test structures and the results of yield simulation with defect size and distribution data.

Local design rules can be used to increase the yield in processes that suffer from conductor shorts, contacts problems and conductor breaks, though extension to the present work may bring about increases in yield from other types of layout manipulation e.g. displacement of contacts.

The proposed use of LDRs has indicated that its application can increase the yield of circuits which makes it a valuable addition to the circuit designer's toolkit.

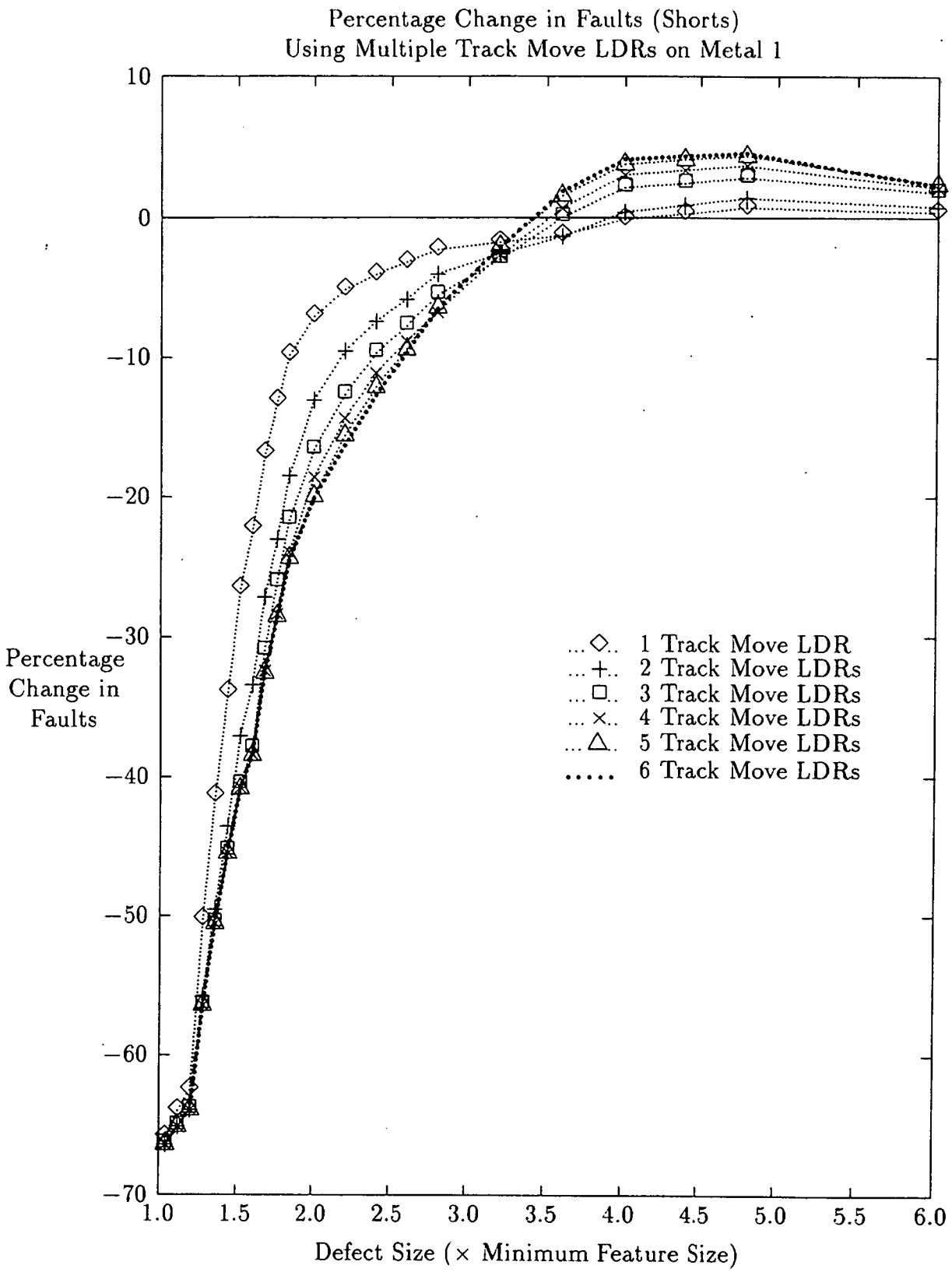


Figure 11: Percentage Change in Defects from Track Move LDRs

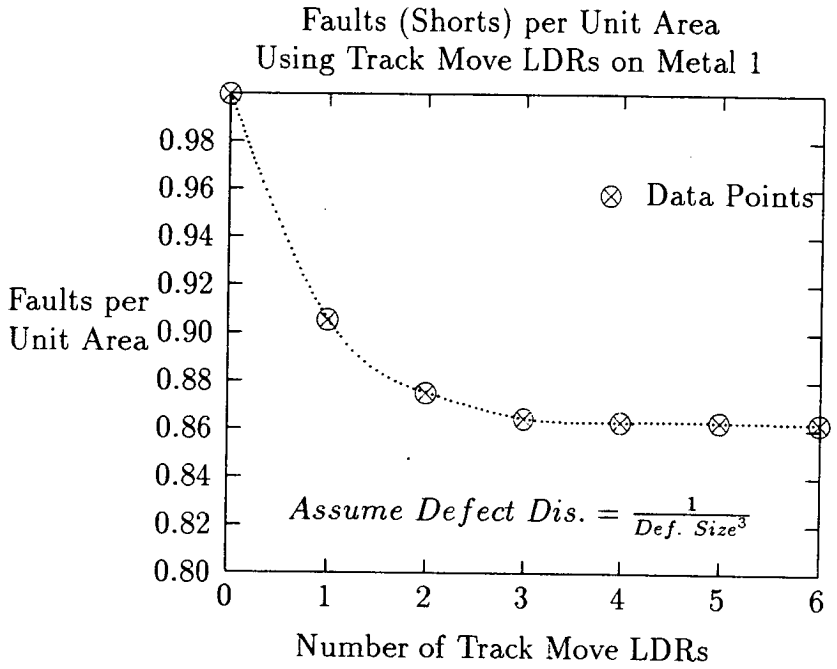


Figure 12: All Short Faults Using Track Displacement LDRs

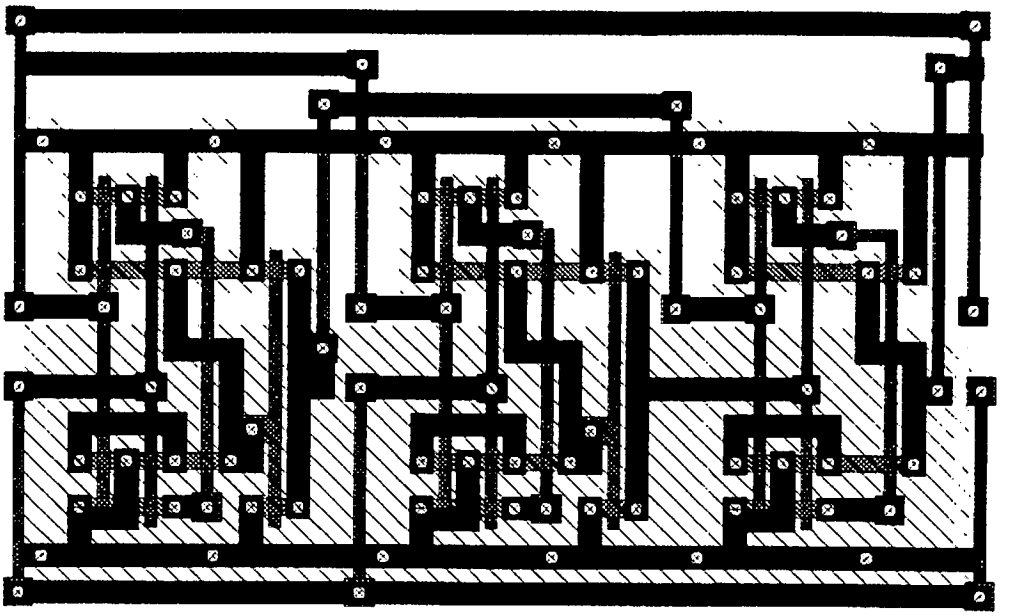


Figure 13: GDR CMOS Circuit

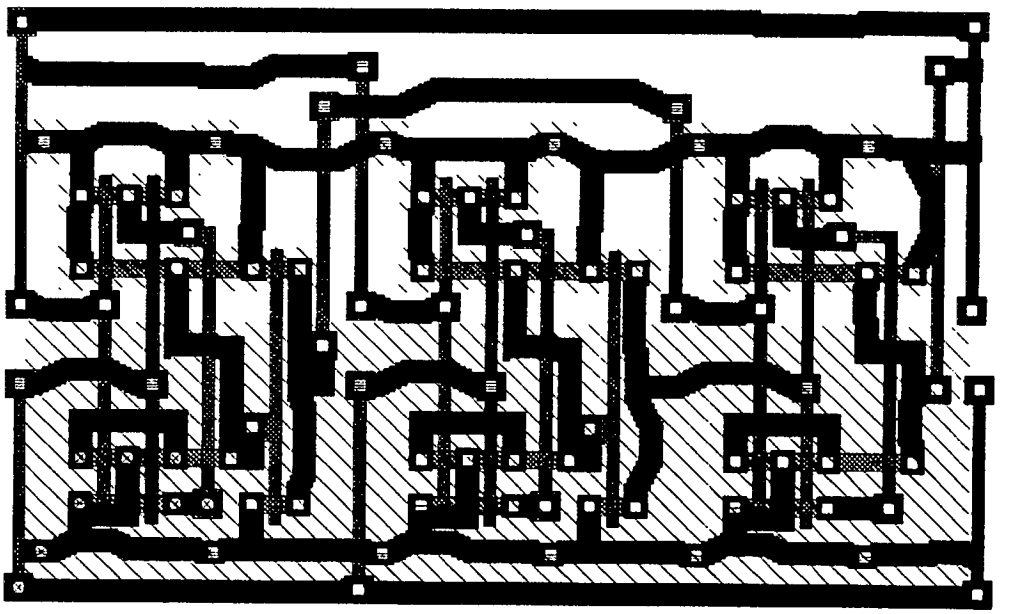


Figure 14: CMOS Circuit with Metal 1 Track Displacement LDR

References

[1] A. C. Ipri. Impact of design rule reduction on size, yield, and cost of integrated circuits. *Solid State Tech*, Feb 1979.

[2] A. C. Ipri. Evaluation of CMOS transistor related design rules. *RCA Review*, 41, Dec 1980.

[3] R.D. Rung. Determining IC layout rules for cost minimisation. *IEEE J. Solid-State Circuits*, pages 35–43, Feb 1981.

[4] A. Onozawa. Layout compaction with attractive and repulsive constraints. In *27th ACM/IEEE Design Automation Conference*, pages 369–374, 1990.

[5] D. M. H. Walker. *Yield simulation for integrated circuits.* The Kluwer International Series in engineering and computer science, VLSI, computer architecture and digital signal processing. Kluwer, Boston; Lancaster:, c1987.

[6] S. Gandemer. Critical area and critical levels calculation in IC yield modeling. *IEEE Trans. Electron Devices*, 35:158–166, Feb 1988.

[7] H. Walker. VLASIC: A catastrophic fault yield simulator for integrated circuits. *IEEE Trans. Comp. Aided Design*, CAD-5:541–556, Oct 1986.

[8] W. Maly. Double-bridge test structure for the evaluation of type , size and density of spot defects. In *International Workshop on Designing for Yield.* University of Oxford, Jun 1987.

[9] P. Gill. Using snake patterns to monitor defects and enhance VLSI device yields (part 1). *Microcontamination*, pages 23–31, Feb 1989.

[10] A. J. Walton, R. J. Holwill, and J. M. Robertson. Contact resistance of silicon-polysilicon interconnection for different current-flow geometries. *Electronics Letters*, 21:13–14, Jan 1985.

[11] Gerard A. Allan. *Phd Thesis: The Effect of Layout Modifications on the Yield of Integrated Circuits.* University of Edinburgh, 1991.

[12] G. A. Allan, A. J. Walton, and R. J. Holwill. Yield improvement by track displacement. In T. E. Mangir, editor, *Defect and Fault Tolerence in VLSI Systems III.* Plenum Publishing Corp., New York, To be published.

[13] R. L. Brown. Multiple storage quad trees: A simpler fast alternative to bisector quad trees. *IEEE Trans Comp. Aided Design*, CAD-5:413–419, 1986.

[14] G. Kadem. The quad-CIF tree: a data structure for hierarchical online algorithms. In *19th Design Automation Conference*, pages 352–357, 1982.

[15] P. Hsiao and W. Feng. Using a multiple storage quad tree on a hierarchical VLSI compaction scheme. *IEEE Trans. Comp. Aided Design*, CAD-9(5):522–536, May 1990.

[16] A. Weinand E. Gamma and R. Marty. Design and implementation of a seamless object-oriented application framework. *Structured Programming*, 10(6), Jun 1989.

[17] L. Walczowski, W. Waller, and E. Wells. *CHIPWISE: An integrated toolset for full custom ,VLSI.* The ECAD Laboratory, The University of Kent, Canterbury, Kent.

[18] I. Chen and A. J. Strojwas. Realistic yield simulation for VLSIC structural failures. *IEEE Trans. Comp. Aided Design*, CAD-6(6):965–978, Nov 1987.

[19] C. H. Stapper. Modeling of integrated circuit defect sensitivities. *IBM J. Res. Develop.*, 27(6):549–557, Nov 1983.