Efficient Critical Area Extraction for Photolithographically Defined Patterns on ICs

Mark P. C. Chia

A thesis submitted for the degree of Doctor of Philosophy
The University of Edinburgh
August 2002

Abstract

The IC industry is developing at a phenomenal rate where smaller and denser chips are being manufactured. The yield of the fabrication process is one of the key factors that determine the cost of a chip. The pattern transfered onto silicon is not a perfect representation of the mask layout, and for an SRAM cell this results in a difference of 3 % between the average number of faults calculated from the mask layout and the aerial image. This thesis investigates methods that are capable of better estimating the yield of an IC during their design phase which can efficiently and accurately estimate the critical area (CA) without the need to directly calculate the aerial image.

The initial attempt generates an equivalent set of parallel lines from the mask layout which is then used to estimate the CA after pattern transfer. To achieve this EYE, Depict and WorkBench were integrated with in-house software. Benchmarking on appropriate layouts resulted in estimates within 0.5 - 2.5 % of the aerial image compared with 1.5 - 3.5 % for the mask layout. However, for layouts which did not lend themselves to representation by equivalent parallel lines, this method resulted in estimates that were not as accurate as those obtained using the mask layout.

The second approach categorises CA curves into different groups based on physical characteristics of the layout. By identifying which group a curve belongs to, the appropriate mapping can be made to estimate the pattern transfer process. However, due to the large number of track combinations it proved too difficult to reliably classify layouts into an appropriate group. Another method proposed determines a track length and position using a combination of AND and OR operations with shifting algorithms. The limitation of this approach was that it was not robust and only proved to work with certain layout types.

The fourth method used a one dimensional algorithm to categorise layouts. The estimated CA was within 0.2 % of the aerial image as compared to the mask layout CA of 2.2 %. The disadvantage of this method is that it can only classify parallel tracks. The next approach built upon the above method and can categorise a layout in two dimensions, not being limited to parallel tracks. A variety of designs were used as benchmarks, and for these layouts this method resulted in estimates that were within 0 - 10.7 % of the aerial image compared with 0.5 - 13.4 % for the mask layout.

Declaration of originality

I hereby declare that the research recorded in this thesis and the thesis itself was composed and originated entirely by myself in the Department of Electronics and Electrical Engineering at The University of Edinburgh.

Mark P. C. Chia

Acknowledgements

I would like to thank my supervisor Prof. Anthony Walton for his invaluable help, advice and support throughout this work. My thanks also goes to Dr. Gerard Allan for his help and advice.

I would also like to thank everyone at the Electrical Engineering Department who have helped, my friends for their help, humour and distractions. Mention should also go to my family who have always supported me. Lastly but not least, I would like to thank my parents for their help, patience and encouragement.

Contents

1. Introduction **1**

1.1. Introduction 1

1.2. Objectives 5

1.3. Thesis Structure 5

2. IC Manufacturing **8**

2.1. Introduction 8

2.2. Crystal Growth 9

2.2.1. Bulk Growth 9

2.2.2. Epixital Growth 11

2.3. Film Formation 11

2.3.1. Oxidation 11

2.3.2. Interconnect 14

2.3.2.1. Aluminium Metallisation 16

2.3.2.2. Copper Metallisation 16

2.4. Doping 17

2.4.1. Diffusion 17

2.4.2. Ion Implantation 18

2.4.3. Comparison of Diffusion and Ion Implantation 19

2.5. Pattern Transfer 20

2.5.1. Lithography 20

2.5.2. Etching 26

2.6. Process Control 29

2.7. Summary 30

3. Yield Concepts **31**

3.1. Introduction 31

3.2. Systematic and Functional Yield 32

3.3. Random Defects 33

3.3.1. Types of Defects 33

3.3.2. Defect Size Distribution 34

3.3.3. Defect Spatial Distribution 37

3.4. Critical Area 38

3.5. Yield Models 40

3.6. Yield Enhancement 43

3.7. Critical Area and Yield Calculation Tools 45

3.7.1. Statistical Approach 45

3.7.1.1. VLSI Layout Simulation for Integrated Circuits (VLASIC) 45

3.7.2. Virtual Artwork Approach 46

3.7.2.1. Virtual Mask Layout 46

3.7.3. Grid Approach 47

3.7.3.1. YMAP 47

3.7.4. Geometric Approach 48

3.7.4.1. Critical Area Prediction by Pattern Recognition 48

3.7.4.2. Critical Area Estimator (CREST) 48

3.7.4.3. Accurate, Fast, Flexible Computation of Critical Area (AFFCCA) 49

3.7.4.4. Multi-Layer Critical Area Computation 49

3.7.4.5. Systematic Extraction of Critical Areas 50

3.7.4.6. Edinburgh Yield Estimator (EYE) 50

3.7.4.7. Edinburgh Yield Estimator by Sampling (EYES) 52

3.7.5. Mathematical Approach 53

3.7.5.1. Voronoi Diagrams 53

3.8. Summary 54

4. Extraction of Photolithographic Critical Area (EPCA) 55

4.1. Introduction 55

4.2. EPCA System 56

4.2.1. Depict 57

4.2.2. Edinburgh Yield Estimator (EYE) 59

4.2.3. Workbench (TWB) 60

4.2.4. Photolithographic Critical Area Mapper (ATOM) 62

4.2.4.1. Calibration Table Generation 63

4.2.4.2. Equivalent Parallel Lines Generation 63

4.3. Results 66

4.4. Summary 71

5. Categorisation and Analysis of Mask Layouts (CALMA) 73

5.1. Introduction 73

5.2. Initial Categorisation Attempts 73

5.2.1. Intepretation of Physical Characteristics in a Layouts 73

5.2.2. Critical Area Analysis on Parallel Tracks 77

5.2.2.1. The Boolean Method 77

5.2.2.2. The Cross Section (1-Dimensional) Method 81

5.3. Categorisation Method 86
5.3.1. Tools for Categorising and Analysis of Mask Layout (CALMAT) 87
5.3.1.1. Main Program (Calma) 87
5.3.1.2. Bitmap Generation (Calma-cif2tl and Calma-tl2cif) . . 89
5.3.1.3. Address Symmetry Calculation (Calma-symmetry) . . . 90
5.3.1.4. Critical Area Calculation Module (Calma-runeye) . . . 94
5.3.1.5. Database Generation (Calma-database) 95
5.3.1.6. Plot Generator (Calma-plot) 96
5.4. Summary 99

6. Calculation of Critical Area Using CALMA **100**
6.1. Introduction 100
6.2. Database Generation 101
6.3. Results 106
6.3.1. Region Size of 1 106
6.3.2. Region Size of 2 110
6.3.3. Routing Track Examples 116
6.3.4. Program Execution Time 122
6.4. Summary 123

7. Conclusion and Future Work **126**
7.1. Improvements to Calma 127
7.1.1. Automatic Database Generation 127
7.1.2. Further Analysis of Addresses 128
7.1.3. Improved Address Algorithm 128
7.1.4. Hierarchical Input Files 129
7.1.5. Generic Mask Layouts 130
7.2. Improvements to EPCA 131
7.3. Advantages of Calma 132

Bibliography **134**

A. User Manual for Programs **141**

A.1. ATOM . . . 141
A.2. CALMAT . . . 142
A.2.1. Calma . . . 142
A.2.2. Calma-plot . . . 144
A.2.3. Calma-database . . . 144
A.3. Miscellaneous Programs . . . 145
A.3.1. Cif2tl . . . 145
A.3.2. Tl2cif . . . 146
A.3.3. Mktrack . . . 147
A.3.4. Scalecif . . . 148

B. Reference Manual for Programs **149**

B.1. ATOM . . . 149
B.2. CALMAT . . . 150
B.2.1. Calma . . . 151
B.2.2. Calma-cif2tl . . . 153
B.2.3. Calma-tl2cif . . . 154
B.2.4. Calma-runeye . . . 155
B.2.5. Calma-symmetry . . . 156
B.2.6. Calma-plot . . . 156
B.2.7. Calma-database . . . 158
B.3. Miscellaneous Programs . . . 159
B.3.1. Cif2tl . . . 159
B.3.2. Tl2cif . . . 159
B.3.3. Mktrack . . . 160
B.3.4. Scalecif . . . 161

C. Further ATOM Results 163

D. Full Calma Results 166

D.1. Type 2 Tracks 167

D.2. Type 3 Tracks 171

D.3. Type 4 Tracks 174

D.4. Routing Track Examples 180

E. Original Publications 186

List of Figures

1.1. Technology node trend based on the DRAM half pitch. (After ITRS [3].) 3

1.2. A typical production ramp curve. (After ITRS [3].) 4

2.1. IC manufacturing process. 9

2.2. Schematic of the setup used to grow silicon crystals by the CZ process. (After Sze [4].) 10

2.3. Schematic of the setup used to grow silicon crystals by the FZ process. (After Mahajan [5]) 11

2.4. Schematic of a typical sputtering systems. (After Smith [10].) 15

2.5. Schematic of a low pressure CVD system. (After Sze [4].) 15

2.6. The effect of positive and negative resists in pattern transfer. (After Mahajan [5].) 21

2.7. Two special anisotropy cases. (a) Isotropic (wet etch). (b) Anisotropic (dry etch). 27

2.8. Schematic of a typical RIE system. (After Stokes [19].) 28

3.1. The different defect types. (a) Extra material. (b) Missing material. (c) Oxide pinhole. 34

3.2. General form of a defect size distribution. 36

3.3. Different CA types on a layout. (a) Extra material. (b) Missing material. (c) Oxide pinhole. 40

4.1. A brief overview of the inputs and output from the EPCA system. . . . 57

4.2. (a) Mask layout and (b) simulated aerial image of the SRAM cell. 59

4.3. FPM generated using EYE for Metal 1 for an SRAM cell. (a) Using the mask layout and (b) aerial image simulated from Depict. 60

4.4. Module layout and experiment tree generated for an experiment. 61

4.5. Run table in TWB used to modify the control factors in the experiment. 62

4.6. Schematic diagram of the steps carried out to obtain the estimated photorealistic CA curve of an IC. 63

4.7. A schematic diagram of the two dimensional matrix used in the EPL algorithm. 64

4.8. A diagram showing the calculation of the critical area values for the first three defect sizes. 65

4.9. Classification of a mask layout using the EPL procedure. (a) Three different sets of parallel lines obtained (Region A, B and C). (b) The CA curve for the mask layout obtained from the total of the individual CA curves for Region A, B and C. 66

4.10. Critical area as a function of defect size for the mask layout, Depict and ATOM for different defect size ranges. 68

4.11. Cumulative effect of the different curves used in the creation of the CA curve generated for the SRAM cell for different defect size ranges. 68

4.12. Fault probability curve as a function of defect size for the mask layout, Depict and ATOM for different defect size ranges. 69

4.13. Mask layout from the AMD 2901 chip used to calculate the fault probability curves. (a) The bounding box highlights the area for which the FP curves are generated from the layout. (b) Generated FP curves. 70

5.1. Some random locations on the AMD chip that EYES selected for sample extraction. 74

5.2. Four mask layouts obtained from the corresponding samples where the shaded areas represent the tracks. (a) Sample number 47. (b) Sample number 53. (c) Sample number 106. (d) Sample number 113. 75

5.3. The six different groups of fault probability curves. (a) Group 1. (b) Group 2. (c) Group 3. (d) Group 4. (e) Group 5. (f) Group 6. 76

5.4. Boolean operators on geometric shapes. (a) AND. (b) OR. 78

5.5. Shifting operators to geometric shapes. (a) Vertical shift. (b) Horizontal shift. 78

5.6. Two possible track layouts giving similar results when using the AND operator to determine the number of tracks. (a) Length wise placement of tracks. (b) Stacked placement of tracks. 79

5.7. Three possible track layouts. 80

5.8. Obtaining the unit cross section value of a layout. (a) Unit cross sectional area taken at the left of layout. (b) Classification of the area. (c) Unit cross sectional area taken in the middle of the layout. (d) Classification of the area. 82

5.9. Classification of the track types in a layout. (a) Three regions used to classify the layout. (b) Classification of Region A. (c) Classification of Region B. (d) Classification of Region C. 83

5.10. Determining the left end and middle identifiers in a track type group. (a) Region A group for which the left end and middle identifiers are to be determined. (b) The end identifier for Region A. (c) The middle identifier for Region A. 83

5.11. CA calculation using Depict on track type 1011 for the end and middle identifiers. (a) CA calculation for the end identifier. (b) CA calculation for the middle identifier. 84

5.12. Example of the method employed. (a) Mask layout. (b) Mask layout lengthened for the calculation of critical area for a defect size of 1μm . . 85

5.13. Schematic overview of the inputs and outputs used by the categorisation tools (CALMAT). 87

5.14. Detailed schematic diagram of the interactions between the modules of CALMAT. 88

5.15. Process used in bitmap generation on two parallel tracks. (a) Two parallel tracks. (b) Bounding box generated by Calma-cif2tl. (c) Expanded bounding box based on region size. (d) Bitmap generated by Calma-tl2cif using the EYE "number" function. 90

5.16. A visual representation of address calculation for a region size of 1 on two parallel tracks. (a) Two parallel tracks. (b) Bitmap generated for the parallel tracks and a boxed bit which is to have its region address determined. (c) Surrounding bits for a region size of 1. (d) Determination of the region address starting from top left and moving in a clockwise direction. (e) Region address for the bit. 91

5.17. Diagram illustrating bit address calculation for a region size of 2 on two parallel tracks. (a) Two parallel tracks. (b) Bitmap generated for the parallel tracks and a boxed bit which is to have its bit address determined. (c) Surrounding bits for a region size of 1 and 2. (d) Determination of the region address (size = 1) starting from top left and moving in a clockwise direction. (e) Determination of the region address (size = 2) starting from top left and moving in a clockwise direction. (f) Region address (size = 1) for the bit. (g) Region address (size = 2) for the bit. (h) Bit address from the combination of region addresses. (i) Bit address represented in decimal. 92

5.18. Some symmetry types in a layout. (a) Vertical symmetry. (b) Horizontal symmetry. (c) Diagonal symmetry. 93

5.19. A layout illustrating symmetry in the x or y axis only. (a) Three parallel tracks where the bit address is calculated at points x and y. (b) Symmetry in the y axis for points x. (c) Symmetry in the x axis for points y. . . . 94

5.20. Generation of CA result file from two parallel tracks using Calma-runeye. (a) Two parallel tracks. (b) Extract of the result file. 95

5.21. Schematic diagram of database generation from individual data files . . 96

5.22. An example extract from the two result files generated for two parallel lines. (a) CAR file. (b) FPR file. 97

5.23. Example results from Calma-plot. (a) Comparison of critical area curves. (b) Comparison of FP curves. 98

6.1. Different track arrangements used in database generation for set 1 (2 parallel tracks). (a) Type 2-1. (b) Type 2-2. (c) Type 2-3. (d) Type 2-4. 101

6.2. Different track arrangements used in database generation for set 2 (3 parallel tracks). (a) Type 3-1. (b) Type 3-2. (c) Type 3-3. 102

6.3. Different track arrangements used in database generation for set 3 (4 parallel tracks). (a) Type 4-1. (b) Type 4-2. (c) Type 4-3. (d) Type 4-4. (e) Type 4-5. (f) Type 4-6. 103

6.4. Determining the bounding region for the address 31:774 in the type 2-1 track. (a) Location of address on the layout. (b) Bounding region required for the address. 103

6.5. Modifications made to specify the bounding region for address 31:774. . 104

6.6. Enlarging the bounding region for the address 31:774 in the type2-1 track using (a) Location of similar addresses on the layout. (b) Bounding region required for the addresses. 105

6.7. Modifications necessary for the enlarged bounding region for address 31:774 (6 unit length). 105

6.8. Relationships for track type 2-1 as a function of defect size. (a) CA error. (b) Fault probability. 108

6.9. Relationships for track type 3-1 as a function of defect size. (a) CA error. (b) Fault probability. 109

6.10. Relationships for track type 2-1 as a function of defect size. (a) CA error. (b) Fault probability. 111

6.11. Relationships for track type 3-1 as a function of defect size. (a) CA error. (b) Fault probability. 113

6.12. Relationships for track type 4-1 as a function of defect size. (a) CA error. (b) Fault probability. 115

6.13. Six different track layouts extracted from an AMD 2901 chip used to evaluate Calma. 117

6.14. Relationships for track type A as a function of defect size. (a) CA error. (b) Fault probability. 118

6.15. Relationships for track type B as a function of defect size. (a) CA error. (b) Fault probability. 118

6.16. Relationships for track type C as a function of defect size. (a) CA error. (b) Fault probability. 119

6.17. Relationships for track type D as a function of defect size. (a) CA error. (b) Fault probability. 120

6.18. Relationships for track type E as a function of defect size. (a) CA error. (b) Fault probability. 120

6.19. Relationships for track type F as a function of defect size. (a) CA error. (b) Fault probability. 121

7.1. Alternate address generation with fewer bit points. (a) Two parallel lines. (b) Address calculated for the hashed areas which will determine the address for the black area. (c) Bitmap of the layout showing the overlap of the two calculated areas. 129

7.2. A schematic diagram for modifications to Calma to handle hierarchical input files. 130

7.3. Bitmap and address generation for two paralel lines (region size of 1). (a) Two parallel tracks used. (b) Enclosed bit for which its address is to be determined. (c) Bounding region using the current module. (d) Bounding region using the new module. (e) Bitmap address from the current module. (f) Bitmap address from the new module. 131

7.4. Two track corners of varying separations. 132

C.1. Fault probability curve as a function of defect size for layout 1. 164

C.2. Fault probability curve as a function of defect size for layout 2. 165

D.1. Relationships for track type 2-1 as a function of defect size. (a) CA error. (b) Fault probability. 167

D.2. Relationships for track type 2-2 as a function of defect size. (a) CA error. (b) Fault probability. 168

D.3. Relationships for track type 2-3 as a function of defect size. (a) CA error. (b) Fault probability. 169

D.4. Relationships for track type 2-4 as a function of defect size. (a) CA error. (b) Fault probability. 170

D.5. Relationships for track type 3-1 as a function of defect size. (a) CA error. (b) Fault probability. 171

D.6. Relationships for track type 3-2 as a function of defect size. (a) CA error. (b) Fault probability. 172

D.7. Relationships for track type 3-3 as a function of defect size. (a) CA error. (b) Fault probability. 173

D.8. Relationships for track type 4-1 as a function of defect size. (a) CA error. (b) Fault probability. 174

D.9. Relationships for track type 4-2 as a function of defect size. (a) CA error. (b) Fault probability. 175

D.10. Relationships for track type 4-3 as a function of defect size. (a) CA error. (b) Fault probability. 176

D.11. Relationships for track type 4-4 as a function of defect size. (a) CA error. (b) Fault probability. 177

D.12. Relationships for track type 4-5 as a function of defect size. (a) CA error. (b) Fault probability. 178

D.13. Relationships for track type 4-6 as a function of defect size. (a) CA error. (b) Fault probability. . . . 179

D.14. Relationships for track type A as a function of defect size. (a) CA error. (b) Fault probability. . . . 180

D.15. Relationships for track type B as a function of defect size. (a) CA error. (b) Fault probability. . . . 181

D.16. Relationships for track type C as a function of defect size. (a) CA error. (b) Fault probability. . . . 182

D.17. Relationships for track type D as a function of defect size. (a) CA error. (b) Fault probability. . . . 183

D.18. Relationships for track type E as a function of defect size. (a) CA error. (b) Fault probability. . . . 184

D.19. Relationships for track type F as a function of defect size. (a) CA error. (b) Fault probability. . . . 185

List of Tables

2.1. Application of different SiO_2 thickness in IC fabrication 12

2.2. Optical sources and associated wavelength. 22

4.1. Average number of faults and the associated errors compared with Depict for the SRAM cell. 70

4.2. Average number of faults and the associated errors compared with Depict for the AMD 2901 layout. 71

5.1. Physical characteristics and the curve groups for two samples. 77

5.2. Length of the two tracks obtained from the three track types. 81

5.3. Table illustrating the calculation of CA for the different track types. . . 85

5.4. Total CA for the layout and the associated errors. 86

6.1. Average number of faults and associated errors for track type 2-1. 110

6.2. Average number of faults and associated errors for track type 3-1. 110

6.3. Average number of defects and associated errors for track type 2-1. . . . 112

6.4. Average number of faults and associated errors for track type 3-1. 114

6.5. Average number of faults and associated errors for track type 4-1. 116

6.6. Average number of faults and associated errors for track type A. 118

6.7. Average number of faults and associated errors for track type B. 119

6.8. Average number of faults and associated errors for track type C. 119

6.9. Average number of faults and associated errors for track type D. 120

6.10. Average number of faults and associated errors for track type E. 121

6.11. Average number of faults and associated errors for track type F. 121

6.12. Associated layout areas and program execution times. 122

6.13. Theoretical and actual number of addresses required for database generation (⋆ estimated value). 124

7.2. Error ranges for various Calma settings. 127

C.1. Average number of faults and the associated errors for layout 1. 164

C.2. Average number of faults and the associated errors for layout 2. 165

D.1. Average number of faults and associated errors for track type 2-1. 167

D.2. Average number of faults and associated errors for track type 2-2. 168

D.3. Average number of faults and associated errors for track type 2-3. 169

D.4. Average number of faults and associated errors for track type 2-4. 170

D.5. Average number of faults and associated errors for track type 3-1. 171

D.6. Average number of faults and associated errors fortrack type 3-2. 172

D.7. Average number of faults and associated errors for track type 3-3. 173

D.8. Average number of faults and associated errors for track type 4-1. 174

D.9. Average number of faults and associated errors for track type 4-2. 175

D.10.Average number of faults and associated errors for track type 4-3. 176

D.11.Average number of faults and associated errors for track type 4-4. 177

D.12.Average number of faults and associated errors for track type 4-5. 178

D.13.Average number of faults and associated errors for track type 4-6. 179

D.14.Average number of faults and associated errors for track type A. 180

D.15.Average number of faults and associated errors for track type B. 181

D.16.Average number of faults and associated errors for track type C. 182

D.17.Average number of faults and associated errors for track type D. 183

D.18.Average number of faults and associated errors for track type E. 184

D.19.Average number of faults and associated errors for track type F. 185

1. Introduction

1.1. Introduction

The semiconductor industry can trace its roots back to the 1940s. The invention of the transistor at Bell Telephone Laboratories in 1947 by William Shockley, Walter Brattain and John Bardeen was the forerunner to the integrated circuit (IC). It was Jack Kilby from Texas Intruments that invented the integrated circuit [1] in early 1958. He designed and built a phase-shift oscillator consisting of a transistor, a capacitor and various resistors on a single germanium wafer. Another researcher, Robert Noyce of Fairchild Semiconductor Corporation worked on the interconnections [2] within an IC which was fabricated shortly after Kilby's announcement. Both Kilby and Noyce are regarded as the coinventors of the integrated circuit and in 1989 they were awarded the first Charles Stark Draper Prize, the engineering equivalent of the Nobel Prize. Thus it was the advent of the triode, transistor and integrated circuit that herald an unprecedented change in the world of entertainment, communication, finance and information processing among others.

The advantage integrated circuits have over discrete component systems are in its size, speed and power consumption. The transistors and wires on an integrated circuit are in the region of micrometers compared to the millimeter or centimeter scales of discrete components. Signals can be transmitted more quickly within the chip than between chips due to the smaller components and wires. In addition, logic operations within the chip require less power due to the reduced parasitic capacitances and resistances. The advantages of integrated circuits translate into benefits at the system level - smaller

physical size, lower power consumption and reduced costs. Given two similar products, consumers tend to prefer the one which occupies the least space and is especially true for portable applications (e.g. mobile phones, handheld computers, laptops). The lower power consumption of integrated circuits enables the designer to reduce the power supply with cheaper low rated units. With less power consumption comes less heat generation which can save costs in fans and heat dissipation technology. Finally, the reduction in the number of components will inevitably lead to the lowering of systen cost.

Integrated circuit manufacturing is also a versatile process. Most manufacturing processes tend to be directly coupled with the product they are manufacturing. Take for example an assembly line designed to build motorcycles. In order to build a different motorcycle there will have to be significant changes made in the equipment. The same is not true for integrated circuit manufacturing. Given that there are different manufacturing processes for various technology types, an assembly line can make any circuit of a particular type simply by changing the mask set. Thus, a CMOS manufacturing plant can produce chips for either a handheld PC or a car by changing the relevant masks. Integrated circuit manufacturing is a key technology for two reasons. The first is that any combination of interconnect and transistors can be built in a fabrication line by the modification of masks and secondly, any circuit can be designed from a combination of transistors and wires. However, no manufacturing process is perfect and with the possibility that some of the chips on the wafer may not function correctly each chip is individually tested and those that pass the test are then packaged.

Over the past thirty years, the number of transistors per chip has doubled about once a year. In the 1960s Gordon Moore predicted that the number of transistors that could be manufactured on a chip would grow exponentially. This prediction now known as Moore's Law has been remarkably accurate. There have been several generations of integrated circuits - SSI (small scale integration), MSI (medium scale integration), LSI (large scale integration), VLSI (very large scale integration) and ULSI (ultra large scale

integration). The traditional way to characterise these technology generations has been the size of the critical dimension (CD) which is the narrowest line patterned on the wafer. For example, in CMOS integrated circuits, the critical dimension is usually the width of the polysilicon gate that defines the transistor's length.

The International Technology Roadmap for Semiconductors (ITRS) [3] details the semiconductor predicted technology requirements for the industry over a period of 15 years and this document is created by experts from around the world. The Roadmap utilises the Dynamic Random Access Memory (DRAM) half pitch to designate the current technology node although alternative measures such as a microprocessor unit (MPU) half pitch can be employed. The technology node can be considered a modern approach in characterising the integrated circuit technology generation. Figure 1.1 illustrates the current technology node trend.

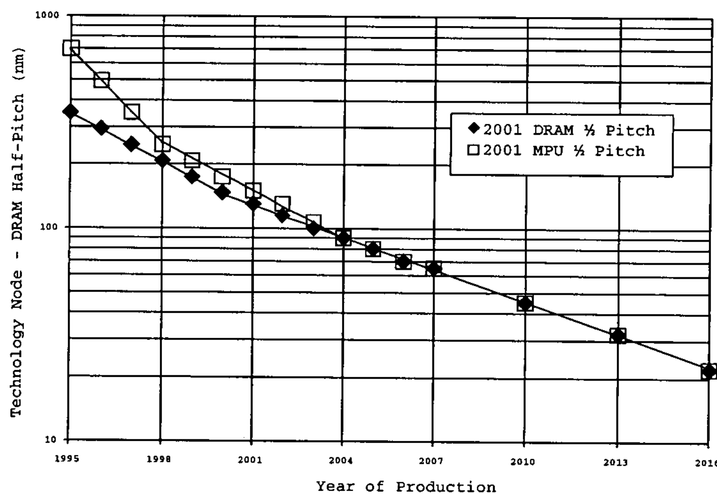


Figure 1.1.: Technology node trend based on the DRAM half pitch. (After ITRS [3].)

The ITRS has identifed two areas that are crucial for the semiconductor industry which are enhancing performance of chips and cost effective manufacturing solutions. With the 90 nm technology node being predicted in 2004, some of these issues remain unresolved. At present numerous solutions are required but unavailable to address the technology requirements of the 90 nm technology node.

One of the key economic driving forces in the semiconductor industry is yield and reliability. The success of an integrated circuit depends not only on new technologies introduced or improved functionality but also on manufacturing costs. Generally, for the success of an integrated circuit, market demand has to be greater than the development and fabrication costs. As the manufacturing process matures, the fabrication costs of an integrated circuit decrease due to improved yield and increased volume. This is known as product ramp up and is illustrated by the production ramp curve shown in figure 1.2. The ITRS defines the production time as when two companies reach a volume of 10k parts per month within three months of each other.

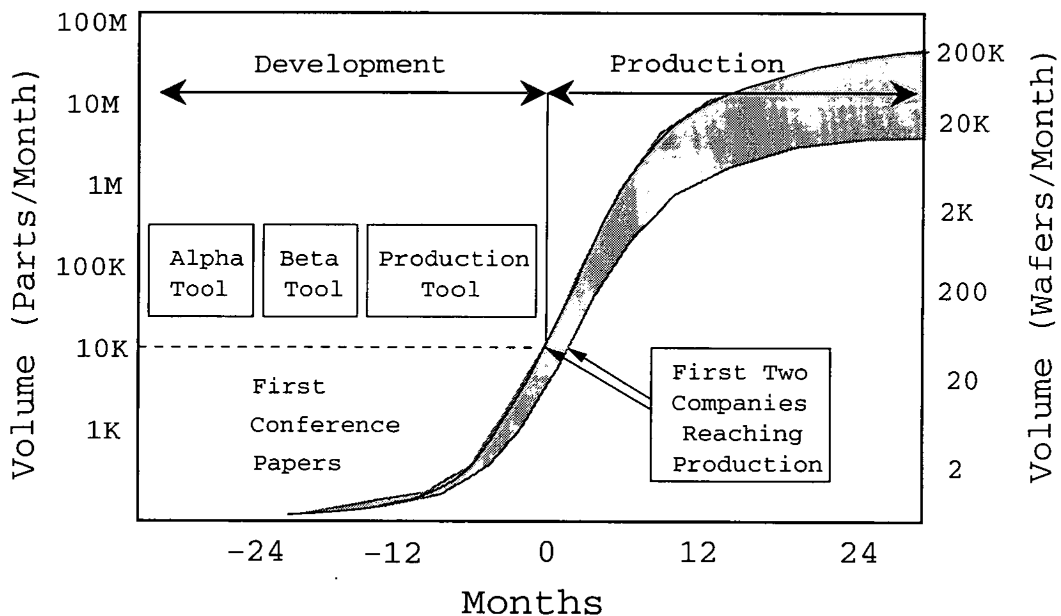


Figure 1.2.: A typical production ramp curve. (After ITRS [3].)

Thus considerable research in yield and reliability has been carried out which can be divided into two distinct but interelated categories, namely software and hardware enhancements. Hardware enhancements would for example include better lithography and defect detection equipment whereas examples of software enhancements would be better mask layout compaction algorithms and yield prediction tools. This thesis concentrates on the software aspect of yield enhancement by aiming to provide better estimations of

critical area (CA) in a mask layout. Critical area is a measure of the susceptibility of features on a chip which will be affected by defects and is defined as the area in which a defect must fall in order to cause a fault.

1.2. Objectives

The primary objective of this thesis is to investigate the feasibility of estimating the critical area of a layout after pattern transfer without the need to directly calculate the aerial image. This can be broken down into two parts. The first was to design methods that are capable of estimating the critical area of a layout after pattern transfer. The second was to evaluate the performance of these methods and to determine whether the results obtained are more accurate than those obtained using the mask layout.

1.3. Thesis Structure

This section outlines the chapters found in the thesis.

Chapter 2: IC Manufacturing

This chapter describes the processing steps invloved in IC manufacturing. The importance of oxidation in silicon technology and doping techniques are discussed. In addition, the move to deep submicron technology has led to copper replacing aluminium for interconnects and next generation lithography solutions being investigated. Finally, the concept of process control and its application to IC manufacturing is discussed.

Chapter 3: Yield Concepts

This chapter discusses yield concepts and forms the foundation on which this thesis is based. The difference between systematic and parametric yield in IC manufacturing

is described with the defect distribution models used in yield prediction. In addition, the concept of critical area is introduced which is then used as a substitute for chip area in yield models. Finally yield enhancement techniques, critical area tools and yield prediction tools are described.

Chapter 4: Extraction of Photolithographic Critical Area

The chapter discusses the initial approach used to estimate the critical area of a layout after pattern transfer. This method generates an equivalent set of parallel lines from a mask layout. To do this various commercial tools were integrated with in-house software. A database was then generated based on the difference between the critical area of the parallel tracks before and after pattern transfer. From the database and equivalent parallel lines, the critical area of the layout after pattern transfer can be estimated. In addition, the performance of this approach is evaluated and analysed.

Chapter 5: Categorisation and Analysis of Mask Layouts (CALMA)

This chapter presents methods that employ mask information to estimate the critical area of a layout after pattern transfer. The premise of this approach is that essential information of surrounding tracks can be used to categorise a layout. Four methods were proposed from which the two dimensional categorisation technique (Calma) was used. Once a layout has been categorised, using a similar database as that in chapter 4, the estimated critical area of a layout after pattern transfer can be determined.

Chapter 6: Calculation of Critical Area Using CALMA

The chapter presents the results obtained from using Calma on different layouts. A database was first created from parallel tracks. Using similar tracks as the database, the performance of Calma was benchmarked. Next, the robustness of Calma was evaluated

using tracks selected at random from an AMD 2901 chip to better represent a typical layout. In addition, the results obtained from using the mask layout was compared with Calma.

Chapter 7: Conclusion and Future Work

This chapter summarises the work carried out in this thesis. It discusses the limitations of the methods and highlights further improvements that can be made. In addition, it also describes the advantages of using Calma for estimating the critical area of a layout after pattern transfer and the possibility of integration with EYES for larger layouts.

2. IC Manufacturing

2.1. Introduction

Integrated circuit technology has come a long way since the vacuum tubes of the 1950's. At present it is possible to put tens of millions of transistors on a single chip and this number is increasing yearly. These advances have been mainly due to better facilities at wafer manufacturing sites, in particular improved lithography and reduction in defectivity.

IC development comprises of three main stages which are design, fabrication and testing. The design stage is where the design rules which specify minimum feature size and IC functionality are developed. The fabrication stage involves the manufacturing of the IC from a mask layout onto a silicon wafer. This stage is composed of many separate steps which include lithography, oxidisation, layer deposition, etching and implantation. Finally, the testing stage is where the manufactured IC is verified for conformance to the specifications laid down in the design stage.

This chapter describes the various steps necessary to fabricate an IC. Wafers can be processed as single wafers or as a batch where a number of wafers are processed simultaneously. Modern technology is tending towards single wafer processing because of better process control. A simplified flow diagram of the IC manufacturing process is

illustrated in figure 2.1 where the layer deposition - lithography - etch loop is repeated a number of times depending upon the process technology.

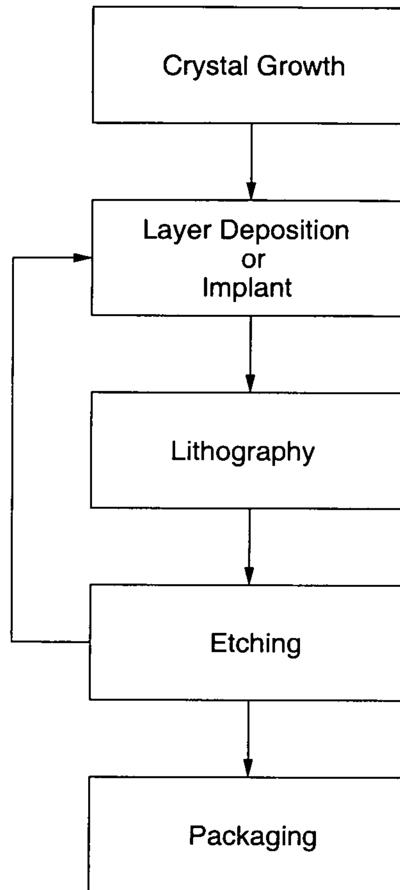


Figure 2.1.: IC manufacturing process.

2.2. Crystal Growth

2.2.1. Bulk Growth

The two methods employed to grow silicon (Si) ingots used for fabricating integrated circuits are Czochralski (CZ) and Float Zone (FZ). The CZ method is the most common method and is a relatively simple process which retains high crystal purity. A quartz

crucible is used in which electronic grade silicon (EGS) is melted and a seed crystal, with the required crystal orientation, is then lowered into the molten Si and partially allowed to melt. The seed is then extracted at a controlled rate as the molten Si solidifies around it as illustrated in figure 2.2.

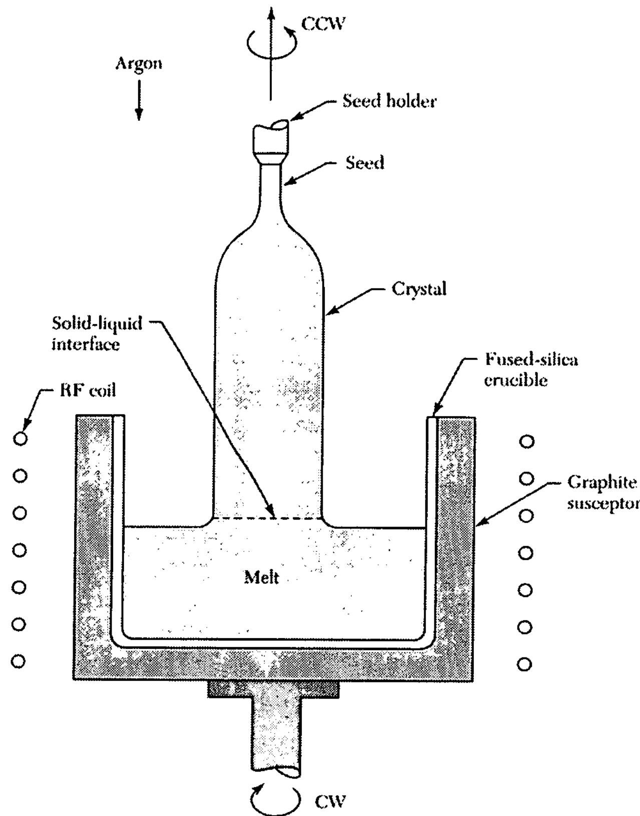


Figure 2.2.: Schematic of the setup used to grow silicon crystals by the CZ process. (After Sze [4].)

One disadvantage of this method is the presence of impurities in the crystal due to the reaction of Si with the quartz crucible and the heating elements used to melt the Si. This technique however is capable of producing large diameter bulk crystals. In the FZ process, a polysilicon rod is suspended vertically in a growth chamber and is maintained in position by supporting balls. Both the rod and seed are heated through induction and a "bottleneck" is formed within the molten zone. It is the controlled solidification of the melt on the seed that converts the polysilicon rod into a single crystal as shown in

figure 2.3. As the molten zone is not in contact with a crucible, the FZ process produces crystals of a higher purity than CZ.

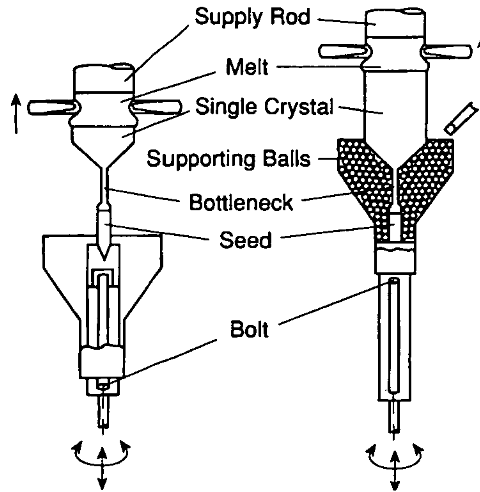


Figure 2.3.: Schematic of the setup used to grow silicon crystals by the FZ process. (After Mahajan [5])

2.2.2. Epixital Growth

The process of growing structures on a substrate wafer which acts as a seed crystal is known as epitaxy. This is different from bulk crystal growth methods as the epitaxial layer can be formed at substantially lower temperatures. There are two different types of epitaxy, homoepitaxy and heteroepitaxy which can be used for the fabrication of bipolar transistors, epitaxial wafers for CMOS, lasers, light-emiting diodes and detectors.

2.3. Film Formation

2.3.1. Oxidation

The ability to produce high quality SiO_2 of varying thickness has played an important part in the maturation of silicon integrated cirucit technology. There are various uses

of thermal oxide in the fabrication of integrated circuits which include

1. Gate oxide and capacitor dielectrics.
2. Isolation.
3. Etch, implant and dopant diffusion masks.
4. Passivation of silicon surfaces.

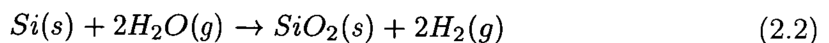
Table 2.1 details the uses for different thickness of SiO_2 in IC manufacturing.

SiO_2 thickness (nm)	Use in IC fabrication
6 - 10	Tunneling oxide
5 - 50	Capacitor dielectric, gate oxide
200 - 500	Surface passivation oxide, masking oxide
300 - 1000	Field oxide

Table 2.1.: Application of different SiO_2 thickness in IC fabrication

Silicon dioxide films are formed at high temperatures through the use of oxygen or water vapour (typically burnt hydrogen). The following reactions describe the formation of SiO_2 using oxygen and water vapour.

$$Si(s) + O_2(g) \rightarrow SiO_2(s) \tag{2.1}$$



$$Si(s) + 2H_2O(g) \rightarrow SiO_2(s) + 2H_2(g) \tag{2.2}$$

The kinetics of thermal oxidation of silicon can be studied based on the model formulated by Deal and Grove [6] which assumes that an initial layer of oxide is already present on

the silicon surface. Three steps are associated with the oxidation process. The first is the transport of the oxidant from the gas phase to the vicinity of the gas-oxide interface. Next the diffusion of the oxidant towards the silicon and finally the interaction of the oxidant at the Si/SiO_2 interface to form the oxide. The equation which models the oxide thickness as a function of time is

$$x_o = \frac{A}{2}(\sqrt{1 + \frac{t+\tau}{A^2/4B}} - 1) \tag{2.3}$$

$$A = \frac{2D}{\kappa} \tag{2.4}$$

$$B = \frac{2DC_0}{C_1} \tag{2.5}$$

$$\tau \equiv \frac{(x_i^2 + Ax_i)}{B} \tag{2.6}$$

where D is the diffusion coefficient of the oxidising species, C_0 is the equilibrium bulk concentration of the species at the oxidation temperature, C_1 is the number of molecules of the oxidising species in a unit volume of oxide, κ is the surface reaction rate constant for oxidation and x_i is the initial thickness of oxide. For short oxidation times $(t+\tau) \ll A^2/4B$

$$x_o = \frac{B}{A}(t+\tau) \tag{2.7}$$

which is the linear oxidation law. Conversely for long oxidation times $t \gg A^2/4B$

$$x_o^2 = B(t + \tau) \tag{2.8}$$

which is the parabolic oxidation law. The calculation of the different rate constants B and B/A can be determined from the measurement of the oxide thickness with respect to the oxidation time. There are many factors that affect the oxidation rates of silicon, the most significant being orientation of the silicon surface, dopants [7], pressure [8] and presence of halogen impurities [9] in the gas phase.

2.3.2. Interconnect

Interconect is a key element of IC technology where up to 8 levels of interconnect is common in modern processes. Interconnects are typically formed using either aluminium or copper on more modern processes. In order to isolate the different interconnect layers, an inter layer dielectric (ILD) is placed between the layers. With device speeds in the GHz range, ILD is an important factor in determining signal propagation time where low dielectric constant materials are necessary as opposed to high dielectric constant materials used for fabricating DRAMs.

The most common method used to deposit metals for IC technology is sputtering. This is a process where atoms are dislodged from a solid due to the exchange in momentum of colliding high energy ions and is shown in figure 2.4. The metal target is placed on the negative electrode and the wafer is placed on the positive one. Ions from the plasma collide with the metal target releasing metal atoms which leave in all directions. In order to increase the number of atoms deposited on the wafer surface a collimator is sometimes employed to direct the motion of sputtered atoms.

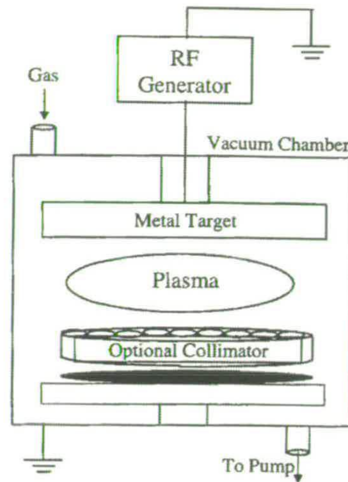


Figure 2.4.: Schematic of a typical sputtering systems. (After Smith [10].)

Another metallisation technique, chemical vapor deposition (CVD) used for depositing thin conducting films such as tungsten is shown in figure 2.5. Some of the advantages of this method are the conformal nature of the deposit, ability to deposit films on a large number of wafers and the simplicity of the equipment. The wafers are placed in a reactor where WF_6 is used as the tungsten source gas. A furnace is used to heat the reactor and this is known as a "hot wall" system or alternatively for a "cold wall" system inductive heating is employed.

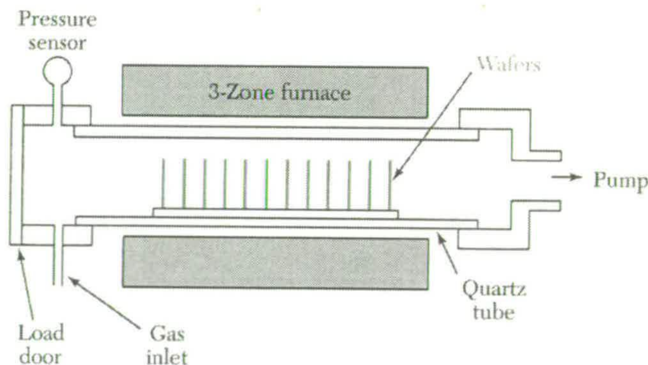


Figure 2.5.: Schematic of a low pressure CVD system. (After Sze [4].)

2.3.2.1. Aluminium Metallisation

Aluminium is widely used for metallisation in integrated circuits and can be deposited using physical vapour deposition (PVD) or CVD techniques. However using aluminium in shallow junctions gives rise to spiking and suffers from electromigration. Spiking occurs when the Al deposited causes Si to dissolve into the aluminium. One method to prevent this is to introduce a barrier metal layer such as titanium nitride (TiN). The mass transport in thin film conductors when subjected to high direct current densities is known as electromigration. The movement of metal ions gives rise to local mass accumulation or depletion and leads to the formation of hillocks and voids respectively. If a hillock grows too large it is capable of causing shorts in the circuit while a void could lead to open circuits in a conductor which will affect the yield. The electromigration resistance can be improved by alloying.

2.3.2.2. Copper Metallisation

The search for metals more resistant to electromigration has led to copper replacing aluminium for ULSI technology [11]. The advantages of copper over aluminium are the superior scratch resistance, electromigration characteristics and electrical conductivity. Copper can be deposited by electrochemical deposition (ECD), PVD and CVD methods.

The disadvantages of using copper are its tendency to corrode under standard chip manufacturing conditions, lack of stable self passivation oxide and poor adhesion to dielectric materials. The fabrication of multilevel copper interconnects is perfomed by patterning the dielectric first and then filling copper into the trenches. This is known as the damascene process. Typically once the trenches have been formed, a diffusion barrier layer (TaN) and a copper seed layer are deposited using PVD which are then filled using ECD. In order to remove excess copper from the wafer surface, chemical mechanical polishing (CMP) is used. The process consists of moving the wafer surface against a pad where a slurry is placed between the two surfaces. Particles and chemicals

within the slurry causes damage to the wafer surface which dislodges material from high points on the surface.

2.4. Doping

2.4.1. Diffusion

Diffusion in semiconductors is the movement of dopant atoms in the crystal lattice via vacancies or interstitals. By controlling the diffusion process it is possible to move dopant atoms deposited near the surface to the desired depth in the wafer and produce the appropriate impurity profile. The flux of dopant atoms is directly proportional to their concentration gradient [12] and is given in the following equation which is know as Fick's first law of diffusion.

$$J = -D\frac{\partial C(x,t)}{\partial x} \tag{2.9}$$

where C is the concentration of atoms as a function of position and time (number of atoms per unit volume), J is the flux of atoms expressed in number of atoms passing perpendicular to a reference surface of unit area in unit time and D is the diffusion coefficient. Assuming that the mass is conserved and that the diffusion coefficient is independent of concentration the following relationship is obtained.

$$\frac{\partial C(x,t)}{\partial t} = D\frac{\partial^2 C(x,t)}{\partial x^2} \tag{2.10}$$

which is Fick's second law of diffusion. This has been solved for some common situations such as constant surface concentration and constant total dopant [13]. In addition, the diffusion coefficient D is a function of temperature, concentration and external fields [13]. In each case the appropriate modifications to Fick's laws have to be made in order to solve the equations.

When diffusion is employed in IC manufacturing, a two step process is commonly used. The first step is known as predeposition where masked regions of a wafer are either implanted or exposed to a diffusant whose surface concentration is constant. Next, the wafer is subjected to a high temperature to drive the dopant into the semiconductor.

2.4.2. Ion Implantation

Ion implantation is a process that involves the introduction of energetic ions into a substrate. The majority of device applications require implantation energies between 1 keV and 1 MeV resulting in ion distributions with average depths ranging from 10 nm to 10 μm. Ion implantation is used to selectively dope regions of a wafer and has largely replaced diffusion for selective doping. A simple model [14] that describes the concentration profile of the implanted ions in an amorphous solid is given by

$$N(x) = \frac{Q}{\sqrt{2\pi}\Delta R} \exp[-\frac{1}{2}(\frac{x - R}{\Delta R})^2] \tag{2.11}$$

where $N(x)$ is the impurity concentration, Q is the dose (ions/cm^2), x is the distance from the surface (cm), R is the projected range (cm) and ΔR is the straggle (cm).

Selective doping using ion implantation is carried out using masks to localise implants to certain regions of a wafer. The mask materials should have characteristics such as a

high stopping power so that thin layers can block incoming ions, compatibility with photolithographic techniques and easy removal after implantation. Examples of materials used as masks are SiO_2, Si_3N_4, polysilicon, metal films, photoresist and polymides.

Once a wafer has undergone ion implantation it has to be annealed to remove the ion implantation induced damage in the crystal and to activate the dopant. Annealing can be performed using furnaces, lasers and rapid thermal annealing (RTA) equipment. Furnace annealing is a simple method however one major drawback is the long annealing times which causes the profiles of the implanted impurities to change. One solution is to employ laser annealing to shorten the time however the implanted material may either melt or regrow by solid phase epitaxy. Rapid thermal annealing is able to overcome the shorfalls of furnace and laser annealing. The process uses a large area of incoherent energy sources that emit radiant light which heats the wafers in an inert atmosphere. This allows for very rapid uniform heating and cooling. Various heat sources can be used such as arc lamps, slotted graphite sheets and tungsten-halogen lamps.

2.4.3. Comparison of Diffusion and Ion Implantation

The advantages of ion implantation over diffusion are

1. Dopants can be introduced in a controlled manner at specific locations.
2. The accelerating voltage controls the penetration depth of ions into a substrate. Hence, by varying the accelerating voltage the junction depth can be controlled.
3. Lower processing temperature.
4. The wide range of ion doses 10^{12} ions cm^{-2} to 10^{18} ions cm^{-2} can be implanted.
5. Ion implantation is a nonequilibrium process which means that it is possible to add more dopant to a substrate even though it has reached its solubility limits.
6. Ion beams can be generated that are monoenergetic and highly pure.

There are disadvantages in using ion implantation.

1. The lattice is damaged, hence it is necessary to anneal the wafer.
2. The implanted profile may change during annealing.
3. Ion implantation requires expensive sophisticated equipment which requires skilled professionals.

However the use of automation and fine degree of control of the process is possible with ion implantation. Thus the advantages of ion implantation far outweigh its disadvantages and as a result it is widely used in the semiconductor industry.

2.5. Pattern Transfer

The design for an integrated ciruit comprises of geometrical patterns for the circuit elements and connections. The reproduction of these patterns on the surface of a semiconductor is achieved through lithography and etching.

2.5.1. Lithography

The process of lithography is the masking of a photo sensitive film (resist) deposited on the semiconductor surface and exposed to a suitable radiation (typically light). This is followed by removal of the exposed or unexposed regions of the film. There are two types of resists which are photosensitive polymeric materials, positive and negative. The exposed areas of a resist are removed in a positive resist leaving a positive image in the resist. Conversely, the unexposed areas of a resist are removed in a negative resist leaving a negative image in the resist. The effects of the different types of resists is illustrated in figure 2.6. Negative resists are seldom used in advanced submicron

processes being superceeded by positive resists. There are a number of properties of a resist which are common to all lithographic techniques [15]. These include contrast, sensitivity, optical density, etching resistance, purity, solubility, resolution, adhesion, spectral response, ease of processing and toxicity.

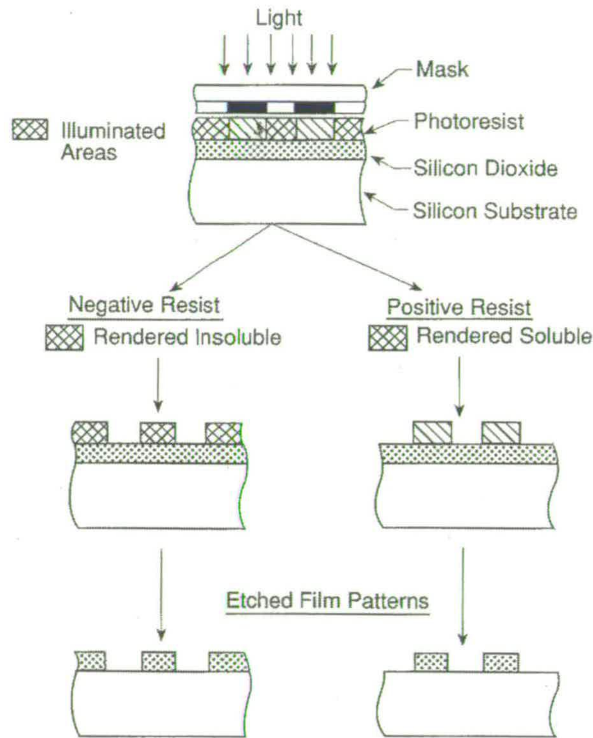


Figure 2.6.: The effect of positive and negative resists in pattern transfer. (After Mahajan [5].)

Masks are used to transfer the circuit layout onto the semiconductor surface and are fabricated by depositing a thin chromium film on a glass plate followed by a layer of a photosensitive resist. Next the pattern is transferred onto the film by exposing the resist which is then developed. The chromium is etched away where the developed resist has been removed and the resulting patterned glass plate is called a reticle. For high density circuits, the masks are produced using electron beam lithography in which the image is scanned onto a suitable resist by an electron beam.

Traditionally, soda lime plates were used as reticles but due to the difference in the thermal exapansion coeffient as compared to silicon, quartz plates are now used. In general, a number of masks are required to complete the fabrication of integrated circuits. Thus the different masks have to be aligned correctly with respect to each other using appropriate alignment marks for the correct positioning and functioning of circuit elements. The errors in positioning one pattern with another is known as an overlay or registration error.

There are three different radiation sources under which lithography can be carried out. These are either optical, X-rays or electrons/ion beams. Currently optical sources are used in the semiconductor industry and cover a wavelength range spanning deep and near ultraviolet. Table 2.2 details some of the common optical sources used and associated wavelengths.

Choices	Illumination source	Wavelength (nm)
G-line	Hg Lamp	436
I-line	Hg Lamp	365
DUV	Excimer Laser (KrF)	248
DUV	Excimer Laser (ArF)	193
DUV	Excimer Laser (F_2)	157

Table 2.2.: Optical sources and associated wavelength.

The simplest method in obtaining a pattern on the resist is to hold the mask in contact with the resist and expose it to light of the appropriate wavelength. This technique is known as contact printing. The presence of particles on the surface, spikes in epixital layers and defects caused by adhesion to the mask and resist limits the usefulness of this technique. In addition, the mask quality degrades after each use and makes it difficult to achieve good reproducibility. Hence, this technique is not used for producing high density submicron circuits.

To reduce damage of the mask, a separation between the mask and resist is desirable. Proximity printing achieves this while positioning the mask and resist parallel to each other. The wave nature of light affects the minimum line width W_m that can be satisfactorily printed and is given by

$$W_m \cong \sqrt{d\lambda} \tag{2.12}$$

where λ is the wavelength of the light and d is the mask wafer separation. It should be noted that there will be diffraction effects in contact printing as well however at a given wavelength the value of W_m is smaller.

To improve on the resolution of proximity printing and reduce mask damage with contact printing, projection printing has been developed. The principle of projection printing involves the demagnification of a mask which is then projected onto a resist through the use of mirrors and lenses. As the mask is demagnified in this method, it is capable of having dimensions which are larger than on the wafer. This allows the easier fabrication, inspection and correction of the mask. A disadvantage is that the optical system used in projection printing gives rise to abberation errors that can distort the image. However, through technological and optical advancements the available optical systems have reduced many of these errors resulting in the predominant use of projection lithography. The ability of a projection system to form separate images of closely spaced objects is known as resolution (S_r) and is given by

$$S_r = k_1 \frac{\lambda}{NA} \tag{2.13}$$

where λ is the wavelength of the light, NA is the numerical aperture of the objective lens and k_1 is a process dependent factor. Using a larger NA will enable better resolution but the drawback of this is a smaller depth of focus (δ). An optical image degrades when the system is defocussed. The depth of focus is the amount of defocussing that can be tolerated and is given by

$$\delta = k_2 \frac{\lambda}{(NA)^2} \tag{2.14}$$

where k_2 is another process dependent factor. The depth of focus has to be sufficiently large to cover the thickness of the resist and surface topography during lithography. One way to improve resolution while keeping the depth of focus high is to use shorter wavelength light.

There are a variety of resolution enhancement techniques (RET) available such as phase-shift masks (PSM) [16] and optical proximity correction (OPC) [17] that strive to improve the resolution of any given generation of IC technology. Conventional photomasks are often known as binary masks because the image is created from transparent or opaque materials. Phase-shift masks improve the resolution through constructive and destructive interference in addition to amplitude modulation. This can be achieved by replacing some of the quartz on the reticle by a phase-shifting material or by substituting the chrome with a material that partially transmits light.

It is well known in lithography that the width of a patterned line is dependent upon the location of shapes in the vicinity while corners of squares can be rounded to form circles. Optical proximity correction attempts to address the distortion effects that occur by modifying the mask prior to pattern transfer. For example, by adding serifs to the corners of a square, the pattern transfered onto the wafer will result in a shape that better represents a square.

In addition, the increasing diameter of silicon wafers make it very difficult to design a projection system that can simultaneously expose the whole wafer. A solution is to step the wafer with precise mechanical movements to a new position and then expose it. The system where the repetition of stepping, aligning and exposing is known as a step and repeat exposure system or stepper for short.

Although the usefullness of optical lithography can be extended through RET, new methods are necessary to meet the requirements of deep submicron and nanometer ICs. Some next generation lithography (NGL) methods are X-ray lithography, electron beam lithography, extreme UV lithography and ion beam lithography.

With X-ray lithography (XRL), line widths of less than 100 nm can be printed. The X-ray wavelength is about 1 nm and printing is through a $1\times$ mask in close proximity to the wafer. This type of printing is crucial as there are no suitable mirrors or lenses which can project or form images. The diffraction effects can be ignored in X-ray lithography for separations in the order of a few micrometers. Mask substrates have to be thin and made of a low atomic number material such as silicon while the pattern is made from high atomic number material such as tantalum, tungsten or gold.

Extreme UV (EUV) lithography is capable of printing line widths up to 30 nm while still maintaing high wafer throughput. The EUV wavelength is 10 - 14 nm and is obtained from laser produced plasma or synchrotron radiation. The difference between this technology and optical projection lithography is that the radiation is reflected by a mask through a $4\times$ reduction lens and imaged onto the wafer.

Electron beam lithography offers several advantages over optical lithography such as better resolution due to the smaller wavelength, maskless patterning of the wafer, greater depth of focus and highly automated and precisely controlled operation. It is also capable of printing line widths of 100 nm or less. In order to expose the resist coated substrate, the resist is divided into a grid of addressable locations. The smallest elementary area to be exposed is known as a pixel and it must receive a certain amount

of energy so that the exposed resist is chemically different from the unexposed resist.

In addition, the time required to write a pattern on a wafer is important in designing electron beam lithography systems. Improvements in electron sources and optics have led to the development of alternative beam shapes such as Gaussian round beams, fixed shaped beams and variable shape aperture beams. Registration between the substrate and the desired pattern can be obtained by analysing the back scattered electrons from the surface topography.

One new approach for electron beam technology is electron beam projection lithography. SCALPEL [18] is one such system which uses $4\times$ projection reduction and combines the advantages of electron beam lithography with the high throughput of optical projection lithography. The resolution of electron beam lithography is limited by electron scattering.

Ion beam lithography is capable of higher resolution than optical, X-ray or electron beam lithographic techniques. The reason for this is that ions scatter less than electrons due to their heavier mass. Two types of ion beam lithography systems are available which are a scanning focussed beam system and a mask beam system. One disadvantage of ion beams is that they may suffer from random space charge effects which broadens the ion beam.

2.5.2. Etching

The removal of unwanted material is known as etching. There are two types of etching which are dry and wet. The three parameters that define an etch process are selectivity (S), anisotropy (A) and uniformity (U) and are given by

$$S = \frac{E_f}{E_s} \qquad (2.15)$$

where E_f is the film etch rate and E_s is the substrate etch rate.

$$A = \frac{vertical\, etch\, rate}{lateral\, etch\, rate} \tag{2.16}$$

$$U = \frac{Max - Min}{Max + Min} \tag{2.17}$$

where Max and Min can correspond to either the etch rate or film thickness depending on which is being measured. Two special cases of anisotropy are isotropic ($A = 1$) and anisotropic ($A = \infty$) which are shown in figure 2.7.

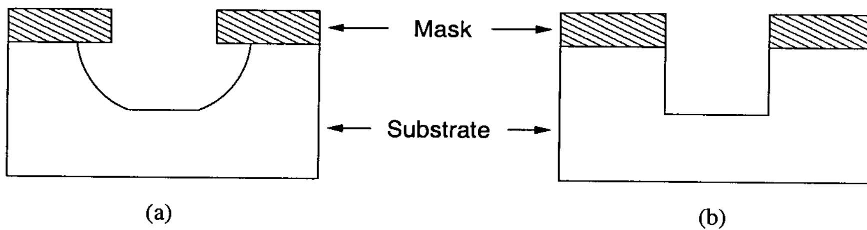


Figure 2.7.: Two special anisotropy cases. (a) Isotropic (wet etch). (b) Anisotropic (dry etch).

In wet etching, an aqueous solution is used whereas in dry etching a gaseous species is employed. In wet etching the etchant is in contact with a variety of materials. Thus to maintain the dimensions of the original pattern the etchant has to be highly selective. Some of the problems of wet etching are that etched materials reveal ragged edge forms, bubbles can grow during etching and act as localized masks preventing proper film removal. In addition, wet etching is isotropic which causes difficulty in controlling the line width as shown in figure 2.7 (a).

Dry etching is typically introduced when feature dimensions fall below 5 μm. The fabrication of high resolution patterns requires a technique for material removal that is highly anisotropic, very selective in etching and provides reasonable rates for material removal. There are a variety of dry etching methods available. Reactive ion etching (RIE) is the most commonly used dry etching technique in fabricating semiconductors and is illustrated in figure 2.8. The wafer is kept in contact with one of the internal electrodes while the pressure of the chamber is kept low. Plasma is then produced between the electrodes by radio frequency gas discharge. Due to the large difference in mobility between electrons and ions, an ion rich layer called an ion sheath is formed on the surfaces. When the wafer and electrode are negatively charged, a large electric potential difference occurs across the ion sheath. This causes the ions to accelarate perpendicularly to the surface of the wafer which increases the etching reaction.

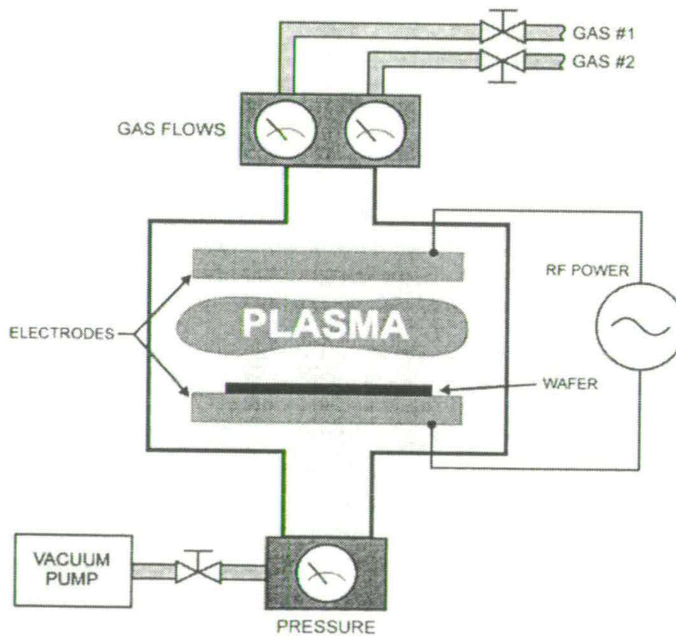


Figure 2.8.: Schematic of a typical RIE system. (After Stokes [19].)

2.6. Process Control

In integrated circuit manufacturing a typical wafer can undergo hundreds of processing steps using different machines. In order to maintain a high yield, each processing step has to be carefully controlled and monitored. The use of process control to monitor the different inputs and outputs of the process and make changes which maintain yield are essential. In order to design a process control system, an accurate model that describes the process has to be formulated which is capable of determining the relationship between the inputs and outputs. Physical and process modelling are examples of two different modelling approaches. A physical model is based on the actual physics involved in a process from which mathematical models are formulated. In process modelling, mathematical models are formulated to describe the process.

There are a variety of control strategies such as run to run control [20] or statistical process control [21]. In statistical process strategies, statistical methods are applied to the input and output data to form control charts. From these charts a process can be monitored and controlled such that it is within set performance limits. In run to run strategies, algorithms are employed whereby the inputs are adjusted after each process based on measurements made on the key output factors. Statistical process control has traditionally been used in industry, however there is a trend towards more dynamic control using run to run strategies.

For these strategies to be effective, process inputs and outputs have to be monitored. Traditionally, wafers were examined ex situ after a processing stage. A sample of the wafers would be selected and moved to metrology stations to determine the reliability and performance of a process recipe. However with advances in metrology equipment it is possible to examine wafers in situ which allow for all the wafers to be examined. This decreases the possibility of introducing defects as wafers are moved from one station to the next and provide process controllers with the necessary feedback information.

In addition, cluster tools are becoming more common in wafer manufacturing plants

[22] as they allow for automation and process control. A cluster tool is a manufacturing system with integrated processing modules linked mechanically which can handle multiple wafers simultaneously.

2.7. Summary

This chapter has briefly described the different processes that are required to manufacture an integrated circuit. Defects will affect the overall yield of the manufacturing process and it is desirable to reduce the number of defects. However, not all the defects can be reduced as some are inherent in the processing stages while others are dependant upon random factors.

As smaller circuit geometries are being designed, fabrication methods have to improve to keep up with the current trend. One of the key processes in a fabrication cycle is pattern transfer as it is responsible for determining the resolution of the critical dimension on a wafer. Lithography is currently reaching its limits for optical sources where 157 nm is the state of the art. Therefore the use of next generation lithography techniques and improvements to optical projection lithography have to be utilised for the fabrication of geometries of 90 nm or less.

3. Yield Concepts

3.1. Introduction

Yield is an important factor in determing the cost of an IC as the number of chips on a wafer that are non-functional is a key element in determining the price for the consumer. It is crucial to be able to predict the yield of the manufacturing process before fabrication as this allows corrective action to be undertaken before manufacturing starts e.g. changing process conditions, design rules or by adding redundancy to the circuit.

The design, fabrication and testing stages in IC development have traditionally been addressed independently. However, this is no longer the case where interaction between the design and fabrication stages play an important role in maximising IC yield. Design engineers take into consideration the IC area, perfomance and design rules when creating the mask layout. The design rules are formulated such that global disturbances, variations in line width and misalignments have minimal impact on the IC yield. For process engineers defect monitoring is important and it is crucial that information regarding defects in processes is obtained as fast as possible. This is especially the case in the early stages of a product development to enable engineers to speedily implement changes necessary to increase yield. Currently, the data obtained from defect monitoring is based on physical and electrical defects. The wafer has to be examined physically

to observe the size and location of defects and then tested electrically (functionally) to determine if a defect causes the failure of an IC. In addition, test engineers create a range of tests on which the manufactured IC has to pass based on design specifications. It is important to identify the cause of any failure and the information gathered fedback to the relevant design or process engineer for the necessary changes. This chapter will discuss the basic fundamentals of yield such as defects, critical area, yield models, yield enhancement methods and yield prediction tools.

3.2. Systematic and Functional Yield

IC manufacturing involves a multitude of different processing steps. Even with improved manufacturing techniques, defects will be introduced which affect the overall yield. In general there are two different sources of yield loss in manufacturing, the global process faults and local process faults. Global (or engineering) manufacturing faults include lithographic misalignments, temperature and processing time. It has been shown [23] that this form of fault primarily affects the operation parameters of an integrated circuit which either results in less tight specifications or in the extreme failure. Local process faults such as random spot defects will either be benign or cause a failure of an IC. Based on these two forms of yield loss the terms parametric (systematic) yield and functional (random defect) yield was developed. The yield for the ith processing step for an IC can be represented using Y_i and is given by the equation

$$Y_i = Y_{si} Y_{fi} \tag{3.1}$$

where Y_{si} is the systematic (parametric) yield for step i and Y_{fi} is the functional (random defect) yield for step i. The total yield for the chip which is represented using Y_{chip} after n processing steps is given by

$$Y_{chip} = \prod_{i=1}^{n} Y_i \qquad (3.2)$$

$$Y_{chip} = Y_{schip} \prod_{i=1}^{n} Y_{fi} \qquad (3.3)$$

where Y_{schip} is the product of the individual systematic yields for each processing step (Y_{si}) and is also known as the chip gross yield. The value of Y_{schip} can be considered a constant and effectively unity in a mature well controlled manufacturing process.

3.3. Random Defects

3.3.1. Types of Defects

There are a variety of defects [24, 25, 26] that can occur on a wafer that can affect functional yield. The defects that are considered here are spot defects which only affects the region surrounding their location.

The main cause of extra and missing material defects is particulates on the mask or wafer surface. The particles manifest themselves during lithography by giving rise to unexposed photoresist areas or resist pinholes which will lead to unwanted material or unwanted etching on a layer. Another name for these defect types is photo or lithography defects [24]. These defects are typically modelled as circular with a given defect radius.

Extra material defects give rise to electical shorts on the circuit. This occurs when a defect lands touching two separate electical nodes as shown in figure 3.1 (a) where the defect is represented as a black circle. Similarly, missing material defects can cause an open circuit. This is caused when a defect lands on a electrical node effectively acting as an insulator as shown in figure 3.1 (b) where this defect is represented as a white circle with dashed black outline.

Oxide pinhole defects are caused from crystal defects, contamination and nitride cracking. This type of defect is modelled as points for missing oxide and can cause the formation of new vias. For example, if the defect occurs between two overlapping separate electrical nodes then the nodes will be shorted together as shown in figure 3.1 (c).

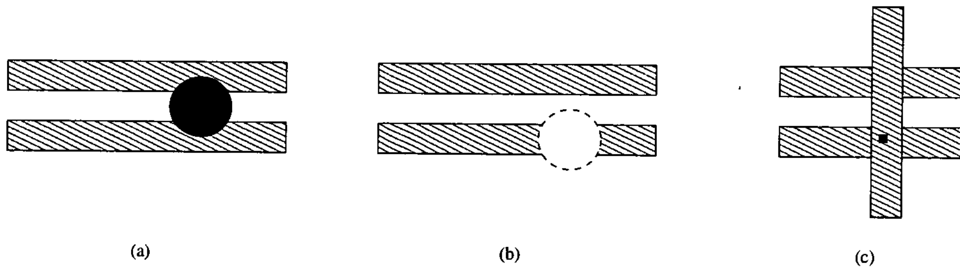


Figure 3.1.: The different defect types. (a) Extra material. (b) Missing material. (c) Oxide pinhole.

Crystal defects or contamination in the crystal lattice at the diffusion junction give rise to junction leakage defects. These can be modelled as points where there is large current leakage to the substrate across the junction but for the purposes of this thesis these are considered to be engineering related defects.

3.3.2. Defect Size Distribution

The exact determination of defect size distribution is a difficult task as completely rigorous visual inspection methods are not feasible and the use of yield data to infer the defect distribution is limited to the assumptions made in a particular yield model. The situation is further compounded as there is no defect distribution model capable of describing every measured defect data set and thus a variety of models have been proposed.

Ferris-Prabhu showed that it is possible to formulate a hypothethical model [27] for the defect distribution using a few simple assumptions. The model assumes that the defect

distribution reaches a maximum at size x_m and decreases monotonically to zero on both sides of the maximum. Using a power law equation, the defect size distribution is given by

$$D(x) = \frac{cx^q}{x_m^{q+1}} \qquad 0 \leq x \leq x_m \tag{3.4}$$

$$D(x) = \frac{cx_m^{p-1}}{x^p} \qquad x_m \leq x \leq \infty \tag{3.5}$$

$$c = \frac{(q+1)(p-1)}{(q+p)} \qquad p \neq 1 \tag{3.6}$$

where p and q are positive values. Hence by using different values of p and q different defect distributions can be modelled and their impact on yield analysed. Conversely, it is also possible to fit experimental data to a proposed defect distribution model. The disadvantage in attempting to correlate a model to the observed data is that although very good correlation might be obtained there is no indication if a better model exists for the data set.

Stapper [28] used specially designed defect monitors that are able to determine if a short or open has occured on a circuit to determine the value of p by substituting $q = 1$. He found that values of p between 2.5 and 3.07 gave good correlations. However, using the sum of least squares technique it was determined that $p = 3.02$ gave optimal results. Figure 3.2 illustrates the general form of this distribution. It should be noted that the value x_m represents the minimum resolvable defect size by the lithographical process. As circuit designs are required to have a minimum critical dimension which in this case

is x_m it can be deduced that defects less than this value will not affect the functionality of the circuit[1]. Therefore it has been widely accepted that the defect size distribution varies proportionally to $\frac{1}{x^3}$.

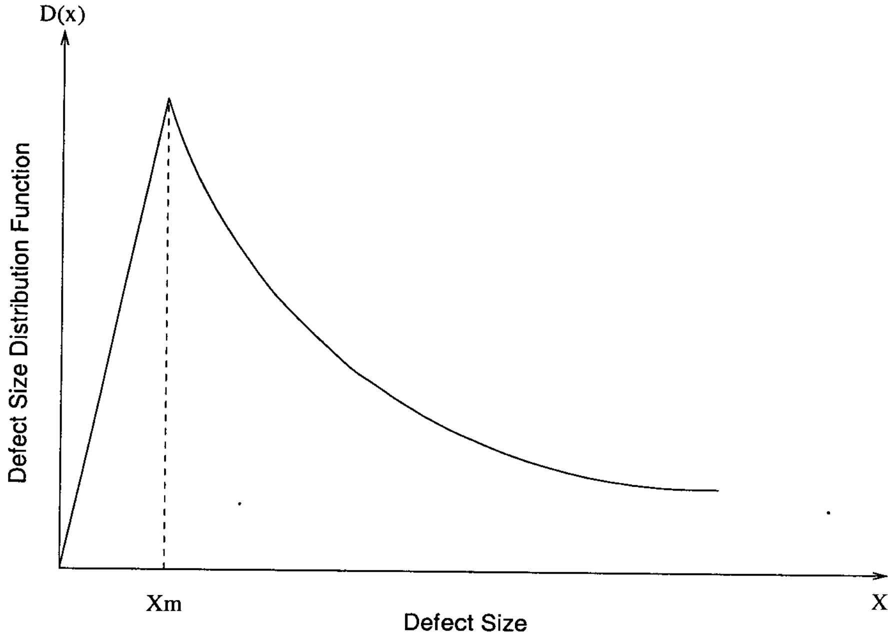


Figure 3.2.: General form of a defect size distribution.

An alternative defect distribution has been proposed by Maly [29] to model the lithographical effects. In this model the defect size is related to the radius of the defect on the mask and the line registration error. The model also proposes two independent sources of defects, particulates in the clean room environment and mechanical contact between mask and wafer (contact printing). This latter source of defects is no longer relevant for modern technologies. The Rayleigh distribution is used to model the probability density function of a single defect source and is given by

$$f_r(x,\alpha) = \frac{x}{\alpha^2} e^{\left(-\frac{x^2}{2\alpha^2}\right)} \quad x \geq 0 \tag{3.7}$$

[1]Note that the reliability of an IC is affected by defects smaller than x_m.

where α is the distribution parameter. The equation describing both defect sources is given by

$$f(R) = \beta_2[\beta_1 f_r(R - m_1, \alpha_1) + (1 - \beta_1) f_r(R - m_2, \alpha_2)] + (1 - \beta_2) f_r(R - m_3, \alpha_3) \quad (3.8)$$

where m_1, m_2, α_1 and α_2 are used to characterise the particulates. Parameters m_3 and α_3 are used to characterise mechanical mask contact while β_1 and β_2 are the weights for the particulate and mechanical defect sources. In modern processes where there is no contact printing, $\beta_2 = 1$.

Another defect distribution model has been proposed by Stamenkovic [30] due to disagreements between the observed defect distribution and the $\frac{1}{x^3}$ distribution widely used. This distribution model is based on the Gamma function and is given by

$$h(x) = \frac{x^{\alpha-1} e^{(-\frac{x}{\beta})}}{\Gamma(\alpha)\beta^{\alpha}} \qquad x \geq 0 \quad (3.9)$$

where α and β are fitting parameters that can be obtained from the experimental data.

3.3.3. Defect Spatial Distribution

Defects have a tendency to cluster within a wafer, between wafers and between lots due to variances in particles found in the environment of clean rooms and chemicals used in manufacturing. The negative binomial distribution [31] is the most common spatial distribution used to model this effect and is given by

$$P(x) = \begin{pmatrix} \alpha + x - 1 \\ \alpha - 1 \end{pmatrix} \frac{(\frac{\lambda}{\alpha})^x}{(1 + \frac{\lambda}{\alpha})^{x+\alpha}} \tag{3.10}$$

where x is the number of defects per wafer, λ is the expected number of defects and α is the clustering coefficient.

Radial defect density distributions [32, 33] have been observed on wafers where defects tend to be more prominent on the wafer edges than in other parts of the wafer. There are a variety of radial distributions available such as a parabola, a second order polynomial, a line, two variable piecewise linear function and two zones. Taking for example the two zone distribution, a wafer is divided into a circular inner and outer zone where defects are assumed to be random within each zone.

In addition, it has been noted that defects on a wafer have a tendency to form clusters [34]. The implication of this is that the presence of a defect would increase the probability of another defect being present in the surrounding area. The difficulty arises from what consitutes a cluster, how many defects would be considered a cluster and how far apart do defects have to be to be considered in a cluster.

3.4. Critical Area

Defects introduced onto a chip as a result of the manufacturing process play an important role in the calculation of IC yield. However, not every defect causes a chip to fail as some defects are so small that they have no effect on the functionality of the chip or land in a non-critical area. Defects that cause the failure of chips are known as fatal or killer defects. Thus it is important to classify critical area on a chip in order to better understand the conditions under which a fault would occur. There are different types of critical areas [35], such as extra material, missing material and pinhole critical areas.

These critical area types correspond to the defect types mentioned in section 3.3.1. For example, the critical area of a chip for a circular defect of diameter x is defined as the area in which its center must fall in order to cause a fault. The expected value of critical area, A_c is calculated as follows

$$A_c = \int_0^\infty A(x)f(x)dx \tag{3.11}$$

where $A(x)$ is the critical area for defects of size x and $f(x)$ is the defect size probability density function.

Extra material critical area occurs when a defect which lands on a layout causes a short circuit between tracks and is illustrated in figure 3.3 (a). To simplify the representation of critical area in the figure, square defects are used. The critical area is highlighted by the black region while the defect is represented by a white square. Missing material critical area is encountered when a defect that lands on a layout causes an open circuit between tracks and is shown in figure 3.3 (b). Pinhole critical area on the other hand does not have any associated defect size i.e. a zero defect size. These defects occur between two conducting layers separated by a dielectric which causes the two layers to be electrically connected as schematically shown in figure 3.3 (c). The presence of the defect in this figure is used to visually point out the location of the defect and is not used to represent its size. In this thesis, the methods presented to estimate CA deal only with extra material critical area which is the most significant type of failure mechanism and is used as a proof of concept.

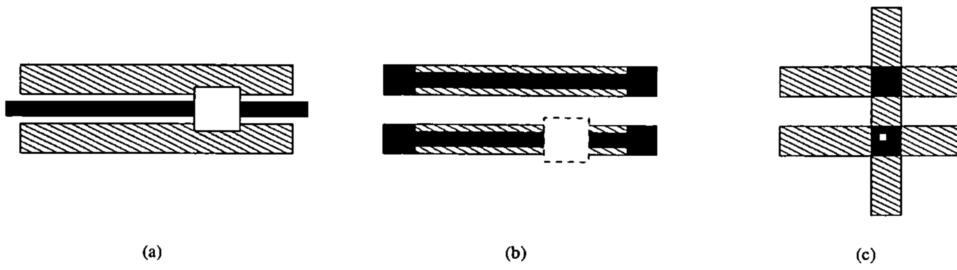


Figure 3.3.: Different CA types on a layout. (a) Extra material. (b) Missing material. (c) Oxide pinhole.

It can be deduced that it is desirable to minimise the overall critical area of a layout and the following section describes the determination of IC yield using the calculated critical area.

3.5. Yield Models

To predict the yield it is important to understand the factors that determine its value. These include the critical area, the number, size, location and distribution of defects and in formulating a model must be considered. Traditionally, the Poisson model is the simplest model used to predict semiconductor yield and for a single processing step is given by

$$Y = Y_o e^{-\lambda} \tag{3.12}$$

$$\lambda = A_{chip} D \tag{3.13}$$

where Y_o is the systematic yield factor, A_{chip} is the chip area and D is the defect density. The product of $A_{chip}D$ which is represented by λ is the average number of faults per chip. From this equation it can be observed that as the chip area increases the yield of a process decreases.

The use of chip area in yield calculation has been replaced with the chip critical area. Hence the original Poisson distribution is modified to take into account the critical area for a single processing step and is given by

$$Y = Y_o e^{-\int_{x_{min}}^{\infty} A_c(x) f(x) dx} \tag{3.14}$$

where Y_o is the systematic yield factor, $A_c(x)$ the critical area for defect sizes x and $f(x)$ is the defect density function. Another yield model was proposed by Murphy [36] and is represented for a single processing step by

$$Y = [\frac{1 - e^{-\lambda}}{\lambda}]^2 \tag{3.15}$$

$$\lambda = A_{ca} D \tag{3.16}$$

where A_{ca} is the critical area of the chip and D is the defect density. The most commonly used model is the negative binomial distribution and the yield for a single processing step can be modelled as

$$Y = Y_o(1 + \frac{\lambda}{\alpha})^{-\alpha} \tag{3.17}$$

$$\lambda = A_{ca}D \tag{3.18}$$

where Y is the yield of the die, Y_o is the systematic yield factor, D represents the defect density, A_{ca} the critical area and α is the clustering parameter. For large ICs it was observed by Stapper [37] that the negative binomial yield model more closely modelled the actual yield whereas the Poisson model tended to be pessimistic in its prediction [36].

There are other model types based on the Poisson model which do not use chip area or critical area in yield prediction. These models use circuit design parameters such as number of transistors and feature size to calculate the value of λ which represents the average number of faults in the Poisson model. Some of the alternative yield models [38] are as follows

$$\lambda = N_{tr}D_dF_s^pD \tag{3.19}$$

$$\lambda = N_{tr}D_dF_s^pD^{\frac{d_1}{D_d}} \tag{3.20}$$

where N_{tr} is the number of transistors, D_d is the design density, F_s is the feature size, D is the defect density, d_1 is the power factor for the defect density and p is the power factor for the feature size.

3.6. Yield Enhancement

There are a variety of yield enhancement techniques and these can be divided into two groups, hardware and software. Hardware techniques involve using advanced manufacturing techniques along with process monitoring tools at each manufacturing process step. Based on information from the monitoring tools, the manufacturing process can be varied accordingly to maximise yield. Software enhancement techniques such as layout redundancy and compaction concentrate on modifying the mask layout before fabrication. Due to the enormous scope of research in yield enhancement methodology, it is not possible to comprehensively describe all the different techniques. This section will describe some of the more common software based enhancement techniques such as redundancy, layout compaction and channel routing.

One yield enhancement technique involves designing a layout with redundant elements. If a defect were to land rendering one of these elements non-functional the layout could be fixed by using another redundant element by reconfiguring the circuit. For example in a memory chip one approach is to use an extra row or column for signal propagation which can be used in the event of a defect rendering another row or column non-functional.

In the past redundancy has been used exclusively for memory chips [39, 40], however there is a trend towards using it in logic circuits [41, 42] and microprocessors [43]. In order to optimise redundancy it is necessary to be able to predict the yield of a redundant circuit accurately. The traditional [44] yield prediction techniques are unable to predict the yield for cirucits other than memory chips. Hence, a new yield model was proposed by Khare [45] which is capable of predicting yield for general circuits. The model calculates the yield by distinguishing between the configurable and non configurable elements in a circuit.

Layout compaction tools attempt to minimise the chip area while maintaining design rule correct layouts. These programs are capable of generating the mask layout from

symbolic layouts or from layouts generated by other layout generation tools. Compactors generate what is known as a critical path [46, 47] using a constraint graph along which none of the elements are able to be relocated. However any elements not along this path are capable of being relocated and hence there is the possibility of modifying the layout for increasing yield or performance.

There are a number of issues that these tools have to address with respect to yield. The minimisation of chip area implies that elements may be placed as close as possible to each other within design rules specifications. The close proximity of the compacted elements will give rise to larger critical areas for shorts. Similarly, the compactors can sometimes stretch wire lines which will result in larger critical areas for open circuits.

Therefore compaction algorithms have to take yield into account when optimising a layout. Chiluvuri [48] proposed a new graph constraint algorithm which is capable of improving yield. This is achieved by increasing the distance between non critical elements which decreases the critical area of shorts while also increasing the widths of wire lines to reduce the critical area of open circuits. Another algorithm which involves the minimization of circuit critical area is presented by Bourai [49].

Channel routing tools attempt to route all the nets on a chip with as few tracks as possible while achieving 100% connectivity in the smallest possible chip area. In multilayer channel routing [50] each channel layer is assigned to either horizontal or verticle wire lines. The ability to move certain wire lines from one layer to another has the possibility of increasing yield. For example, if a horizontal wire in a highly dense region is move to another layer which is less dense the critical area for shorts will be reduced.

The DTR algorithm [51] is capable of minimising the critical area but is constrained to only horizontal routing layers. An alternative algorithm proposed by Kuo [52] addresses this limitation. This algorithm employs net burying and net floating techniques in order to minimise chip critical area. In net burying a wire line is moved down to an alternative layer while in net floating a wire line is moved up to an alternative layer.

3.7. Critical Area and Yield Calculation Tools

Due to the importance of critical area and yield in IC manufacturing, tools for their calculation have become more prominent as they are capable of providing useful feedback to the design and process engineers. The tools developed are based on either the deterministic or statistical calculation techniques in determining critical area and yield. Statistical techniques have been based on the Monte Carlo method whereas a variety of deterministic approaches are available such as virtual artwork, grid analysis, geometric methods and mathematical modelling.

In deterministic approaches, the critical area is obtained from analysis carried out on the mask layout whereas statistical calculation techniques involve generating a large number of defects and placing them on the layout to observe their effects. Generally tools that employ statistical calculation techniques tend to take a longer computational time than the deterministic calculation tools. However, statistical tools are capable of mapping defect faults on multiple layers of a layout and provide more detailed information regarding the nature and location of faults.

3.7.1. Statistical Approach

3.7.1.1. VLSI Layout Simulation for Integrated Circuits (VLASIC)

VLASIC [53] is a Monte Carlo simulation based yield prediction tool developed at Carnegie Mellon University. It functions by generating and placing defects on the chip sample and analysing the resulting layout for circuit faults. Random number generators are used to generate and place the defects on the layout based upon the defect size and spatial distribution functions. By choosing the appropriate number of defects the accuracy of the method can be predicticed within a given confidence level.

The output of the random number generators is a list of defect sizes and locations on the die sample. VLASIC then examines the defects based on fault analysis procedures

which uses the layout geometry and the neighboring area surrounding the defect. Thus, a detailed list of failure nodes and mechanisms can be obtained and the susceptibility of different regions on the layout to faults can be quantified. It is capable of identifying a variety of faults caused by defects such as extra material, missing material, oxide pinhole and junction leakage.

The major disadvantage of VLASIC is the computational time required to generate and analyse the random defects on the layout. To generate 123,808 defects for a 4 μm NMOS RAM cell required 189 CPU minutes on a VAX-11/785. Although computing power has increased over the years so has the size of chips which limits the practical use of VLASIC to small layouts. However a parallel version of VLASIC [54] has been developed to address this problem.

3.7.2. Virtual Artwork Approach

3.7.2.1. Virtual Mask Layout

Maly [29] developed a technique which is capable of extracting the critical area of large layouts by creating a simplified representation of the layout. The virtual mask layout is based on creating a statistical representation of a layout using parameters such as track width, total length of tracks and average separation between tracks. This is achieved by using polygon manipulation such as bloat and shrink operations on the original mask layout.

For example, given a layout with 1, 2 and 3 μm track widths. To obtain the statistics on 3 μm width tracks, the layout is shrunk by 2 μm which causes all tracks less than 3 μm to vanish. The remaining tracks are then bloated by the same amount. Subtracting this new layout from the original leaves tracks of 2 μm or less. Thus by subsequent polygon manipulations and layout substraction the statistics for the layout can be obtained.

Using the statistics obtained for the different parameters a simplified version of the layout can be generated which consists of parallel tracks of varying widths and separations. The critical area of the simplifed layout can be calculated from analytical equations formulated for the virtual layout. However due to the simplification made, layout detail is lost and defect sensitive regions on the layout would not be identifiable. Hence results are less accurate and informative than those obtained from VLASIC.

3.7.3. Grid Approach

3.7.3.1. YMAP

YMAP, developed by Wagner [55] is another tool capable of calculating layout critical area. It is based on deterministic methods which uses algorithms to partition a layout into grids. An error bound can then be calculated for the critical area based on the grid size used. The algorithms use the critical radius (minimal size of a defect causing a circuit fault) at each point on the grid to obtain the critical area of the layout. Hence, the smaller the grid size the more accurate the estimation of critical area. However, one drawback of this technique is that smaller grid sizes result in increased computation time.

The first algorithm is based on searching every grid point for the critical radius and is of $O(n^2)$ complexity. A second alogrithm which improves upon the first only requires an exhaustive search for the first grid point. It then uses a ring theorem to identify the regions where the critical radius of the remaining grid points might be located and is of $O(n^{1.5})$ complexity.

The values obtained from YMAP can be used in two ways. One is used to determine the yield of the layout and the other is to draw a "topographic map" of the layout where the intensity of the color at a point is proportional to the defect sensitivity. YMAP can be used as a visual feedback tool to a design engineer to obtain fast and precise yield prediction of the layout.

3.7.4. Geometric Approach

3.7.4.1. Critical Area Prediction by Pattern Recognition

Mattick [56] proposed a technique to calculate critical area of a rectilinear layout using pattern recognition. By identifying the different shapes that make up a layout the total critical area can be obtained from the partial critical area of the individual shapes. The layout is initially classified into partitioning rectangles by superimposing an elemental grid which is equal to the minimum feature resolution.

The partitioning rectangles are then categorised according to corner cell counts which is the number of adjacent grid cells composed of dissimilar material. The ordered set of the corner cell counts uniquely identify a partitioning rectangle. It has been shown that there are only 16 different patterns in rectilinear layouts where each pattern consists of a particular combination of partitioning rectangles. Thus mathematical equations to calculate critical area can be obtained for each different partitioning rectangle which is then used to obtain the total critical area of the layout.

3.7.4.2. Critical Area Estimator (CREST)

CREST, developed by Nag [57] is a tool capable of using layout hierarchy to calculate critical area and is implemented using polygon operations. The calculation of extra material critical area is obtained by bloating the polygons by the defect size. The resulting intersections between the bloated polygons represent the required critical area. The algorithm employed for polygon manipulations is of $O(nlog(n))$ complexity. This technique works by identifying a root pattern cell which is a cell surrounded by the interacting regions of other cells. From this cell the leaf pattern cells which are a subset of the root cell can be obtained. Thus by calculating the critical area of the root pattern cells, leaf pattern cells and unique cells in a layout the total critical area of the layout can be obtained.

CREST was tested on an SRAM layout with 4,500 transistors. Using a flat critical extraction method the computation time was 1,585 seconds, however with a hierarchical extraction method this was reduced to 127 seconds. Thus computational time can be reduced using a hierarchical extraction technique.

3.7.4.3. Accurate, Fast, Flexible Computation of Critical Area (AFFCCA)

AFFCCA [58] is another critical area calculation tool based on polygon manipulation. It was developed to analyse layouts with circular defects and lithography deformed layouts. The critical area for extra material defects is calculated in a similar manner as that in the CREST tool. However to simulate a circular defect, the corners of the bloated shapes are rounded by a rounding radius. As AFFCCA uses a scan line algorithm it is of $O(nlog(n))$ complexity.

In order to estimate the critical area of a lithography deformed layout, extreme cases of deformations are assumed which are over etching and under etching. For under etching the layout was bloated by the oversize factor and for over etching the layout was shrunk by the undersize factor. Tests carried out on a 512 bit SRAM cell showed that AFFCCA required 1,117 seconds to calculate the critical area.

3.7.4.4. Multi-Layer Critical Area Computation

Xue [59] developed a multi layer critical area calculation tool for orthogonal layouts. The technique involves using the corner stitching data structure with polygon manipulation and is of $O(n)$ complexity. The concept of the corner stitching method is to stitch rectangles at their corners. For every rectagular object, horizontal lines are drawn on its upper and lower side until they reach another object. Thus a layout can be divided into occupied and unoccupied rectangles which are known as solid and space tiles respectively. Each mask layer is stored in a corner stitching plane with interactions

between the planes noted (e.g. formation of a transistor through an overlap of a poly layer and a diffusion layer).

For a given defect size a bounding region is created around a solid tile by expanding the edges of the rectangle by the defect size. Any other solid tiles in this region are then flagged and the critical area calculated by expanding the rectangles by half the defect size. These critical areas are then inserted into the extra corner stitching plane. The process is then repeated for each solid tile on the layout. The total critical area is the sum of the solid tiles in the extra corner stitching plane. This technique required 75 seconds to calculate the extra material critical area of a chip with 8,854 transistors on a HP750 workstation.

3.7.4.5. Systematic Extraction of Critical Areas

SCA [60] is another polygon manipulation based critical area calculation tool and as such is of $O(nlog(n))$ complexity. It uses a scan line approach to calculate critical area on multiple layers of a layout. Two scans of the layout in the horizontal and vertical direction are used to identify the susceptible sites (critical region) in the layout.

The extra material critical area of a susceptible site is obtained by analysing the intersection of the polygons (either to the left and right or top and bottom for a vertical or horizontal site respectively) after bloating by the defect size. The overall critical area of the layout is obtained from adding the individual critical area for the critical regions. SCA is capable of identifying the critical regions on a layout which are susceptible to shorts and opens.

3.7.4.6. Edinburgh Yield Estimator (EYE)

One tool which uses a deterministic method to calculate CA and yield is the Edinburgh Yield Estimator (EYE) [61] and is of $O(nlog(n))$ complexity. EYE uses a scan line

based approach where the layout is divided into polygons. Polygon operations such as those described for CREST are carried out to calculate the extra material critical area. Similarly for missing material critical areas, shrink operators are used on the polygons.

In addition to square defects, arbitrary shaped defects can be used to calculate critical area. This is achieved by placing the defect shape at each vertex of a polygon. A new polygon is then formed from the line segments such that the new line segments are tangential to the defect shapes. Similarly, the intersections of the new polygons formed will give the critical area for these defects.

The critical area for soft faults and hard faults can be obtained from EYE. Hard faults cause functional failures in the circuit (shorting of two conductors or open circuit along a track). In constrast, soft faults do not affect the functionality of the circuit. However the physical distance between conductors (extra material) or thickness of a conductor (missing material) can be affected which might give rise to faults under prolonged operation.

EYE is capable of extracting the critical area of ICs using either a flat or hierarchical [62] based approach and is not limited to Manhathan based layouts. Due to the increased complexity of ICs and using the Alliance dlxm 32bit processor as an example, a flat approach takes seven hours [61] to calculate CA on a SUN Sparc 20. The hierarchical approach is capable of reducing CA extraction times for the Alliance processor on the same workstation by a factor of three [61].

In addition, EYE has also been integrated with Cadence so that critical area and fault probability maps (FPM) can be obtained from mask layouts and this gives mask designers useful visual feedback. On the FPM, the darker the region the higher the probability of a defect causing a fault in the circuit.

3.7.4.7. Edinburgh Yield Estimator by Sampling (EYES)

With larger and more complex chips such as the STACS CPU (a superscalar 128 bit very long instruction word processor) EYE takes seventeen hours [63] to calculate the CA on a SUN Sparc 20. Hence, quicker methods are necessary especially with ever increasing number of transistors being placed on chips. A sampling based approach was implemented using EYES (EYE-Sampling) which reduced extraction times by a factor of twenty [63].

The foundation of EYES is based on survey sampling where population properties can be estimated by using a number of random samples from the population. In survey sampling, the error bounds are determined on the variance on the population and is independent of the population size. Mathematical formulas can be used to determine the mean (μ), variance (σ^2) and bound on the error (δ) of estimation in the population. These formulas are given by

$$\mu = \bar{y} = \frac{\sum_{i=1}^{n} y_i}{n} \tag{3.21}$$

$$\sigma^2 = \frac{s^2}{n}(\frac{N-n}{N}) \tag{3.22}$$

$$\delta = 2\sigma \tag{3.23}$$

where $\bar{y}$ is the sample average, N is the population size, n is the number of samples and s^2 is the sample variance.

In addition, a more accurate technique is stratified random sampling. The population is initially classified into different non overlapping areas known as strata. From each strata a random sample is obtained and mathematical formulas can then be used to calculate the population parameters and are given by

$$\mu = \bar{y}_s = \frac{1}{N}\sum_{i=1}^{L} N_i \bar{y}_i \tag{3.24}$$

$$\sigma^2 = \frac{1}{N^2}\sum_{i=1}^{L} N_i^2 \sigma_i^2 \tag{3.25}$$

where L is the number of strata, N_i is the population size of the ith of L strata and σ_i^2 is the variance of the ith of L strata.

3.7.5. Mathematical Approach

3.7.5.1. Voronoi Diagrams

Papadopoulou [64, 65] proposed a method for calculating critical area using voronoi diagrams. There are numerous uses for voronoi diagrams in practise such as generating aesthetic patterns, modelling biological processes and calculating mobile phone antenna placements. However, by changing the metric in which voronoi diagrams are calculated from Euclidean to L_∞ they are able to be used for analysing mask layouts.

In order to calculate the critical area of polygons, a voronoi diagram which partitions the layout into regions called voronoi cells has to be generated. A voronoi cell of a polygon P is the locus of points closer to P than any other polygon. Using the diagram, a

second order voronoi diagram can be generated which provides information about the second nearest polygon. Thus the total critical area of the layout is the sum of the critical areas within each second order voronoi cell. This technique was employed using a scan line algorithm with $O(nlog(n))$ complexity. Both the extra material and missing material critical areas can be calculated using voronoi diagrams.

3.8. Summary

This chapter has described the concepts of yield and the different parameters that affect yield. Defect density is difficult to quantify and hence is difficult to model accurately resulting in a variety of models where the $\frac{1}{x^3}$ distribution is most commonly used. A variety of yield models have been proposed which are based on chip area and layout parameters. Recently there has been a shift in methodology to using critical area instead of chip area to give better yield estimates in yield models. In addition various tools have been described which are capable of calculating the critical area and yield of a layout. These tools are based on five different methods where the two main techniques that are used for commercial IC layouts with arbitary geometries are statistical and geometric methods.

As not all the tools are capable of calculating the critical area on non-Manhattan based layouts it will follow that these tools will be unable to determine the critical area on a layout after pattern transfer. EYE/EYES was employed in this thesis as it is capable of fast and accurate CA calculations on large layouts and is not limited to Manhattan based layouts. At present the tools that can calculate the CA of non-Manhattan layouts require an external program to first obtain the pattern transfered before performing this task. This thesis will develop techniques which uses the original mask layout to directly predict the critical area of the layout after pattern transfer.

4. Extraction of Photolithographic Critical Area (EPCA)

4.1. Introduction

As described previously, the calculation of critical area plays a key role in the prediction of yield and hence it is essential that accurate and realistic calculations can be obtained. During the manufacturing process, lithography and etching affects the pattern transferred onto silicon, which in turn effects the critical area. One solution is to take into account the effect of photolithography by using a simulator with the mask layout and then running a critical area extraction tool on the resulting aerial image. This task can prove time consuming if different parameters in the photolithographic simulation have to be modified. Hence a combination of in-house and commercial software has been developed to create a system that enables the effects of input parameters of both the photolithography simulator and the critical area extraction tool to be examined. The major elements of this system, generated as part of this work is known as the Extraction of Photorealistic Critical Area (EPCA) are

- Avant! Workbench: An integrated design environment
- Avant! Depict: A photolithographic simulator

- Edinburgh Yield Estimator (EYE): A critical area extraction tool
- Photolithographic Critical Area Mapper (ATOM): A photolithography critical area mapping tool

The EPCA system has been built around Workbench (TWB) [66] which provides the facility to assemble modules to create a process recipe that can be used to design an experiment. Thus by creating modules for Depict, EYE and ATOM it is possible to integrate these tools within TWB. Currently there are two methods within EPCA for extracting the photorealistic critical area of ICs. One is by using Depict and the other by ATOM. Depict simulates the aerial image from the mask layout but this is not a feasible option for large ICs with many thousands of transistors due to the time required to obtain the result. Hence, one solution is to map the critical area curves generated from the mask layout to the critical area curves obtained after pattern transfer and this is the approach used by ATOM. This chapter will describe how the different elements of the EPCA are integrated to obtained the estimated critical area curves of a layout after pattern transfer.

4.2. EPCA System

The EPCA system is based on classifying a mask layout into parallel lines of varying spacing. This is similar to the virtual mask technique used in calculating the critical area of ICs by replacing the original layout by a simplified layout. In this case the simplified layout is a combination of parallel lines with different lengths and line spacings. A brief overview of the EPCA system is illustrated in figure 4.1. Inputs to EPCA include the database file for parallel lines and the IC mask layout for which the estimated CA curves are to be estimated. The outputs consists of two files, one of which consist of the CA curves of the layout generated by Depict (Depict Result) and the second which consist of the CA curves obtained using ATOM (ATOM Result). The following sections

will describe the EPCA modules in further detail. The user and reference manuals for EPCA are presented in appendices A and B respectively.

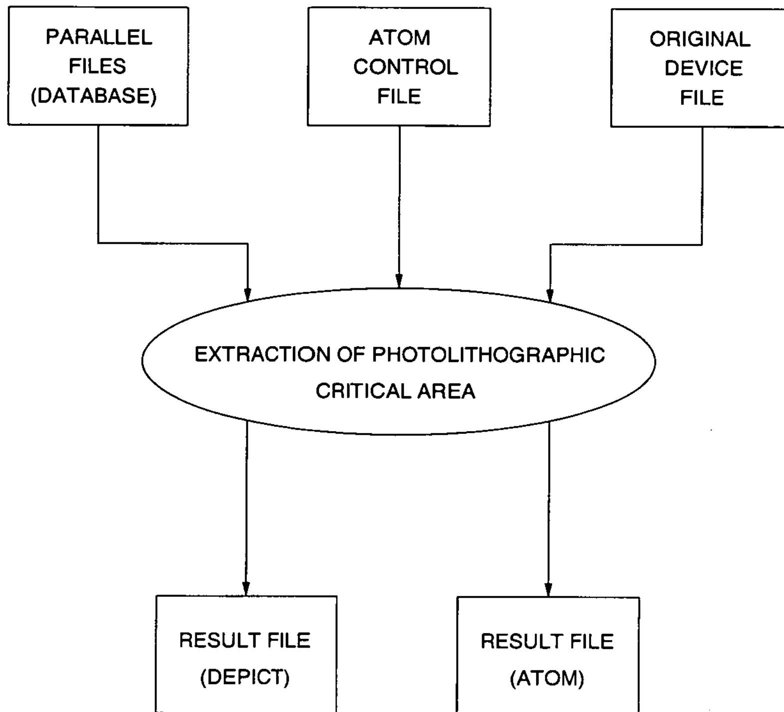


Figure 4.1.: A brief overview of the inputs and output from the EPCA system.

4.2.1. Depict

It is impractical for mask designers to manufacture their mask layouts to test for printability or to optimise their designs every time changes are made to the layout. Hence in order to save time and resources, photolithography simulators are available which allow mask designers to simulate the pattern transfer process. These simulators are capable of simulating a two dimensional (2D) aerial image which is then used to calculate a three dimensional (3D) image of the structure based on the processing steps involved.

A numerical simulation package for submicron photolithography has been described by Barouch [67] which is capable of analysing aerial image generation, exposure, post-

exposure bake and dissolution. SAMPLE-3D [68], a 3D optical photolithography simulator has been developed based on a new ray-string algorithm for dissolution etch-front advancement. Another simulator, Depict [69] which is available from Avanti! consists of a 2D aerial image module and a 3D Advanced Applications Module (AAM). These simulators enable designers to obtain accurate representations of a layout on the wafer. For example, research carried out by Flack [70] consisted of a series of photolithographic simulations using Depict and when compared to experimental results by overlaying the patterns obtained resulted in good agreement.

The Depict aerial image module computes the mask image on the wafer generated by an optical projection stepper. Aerial image simulation allows the designer to analyse the effects of defects, residual lens aberration and to verify phase shifting mask design. Mathematical models [71, 72, 73] are used in the calculation of the aerial image and some reasonable assumptions made in the model are

- Quasimonochromatic light [74]
- Kohler's illumination condition [75]
- Scalar aerial image formulation [74]

The Depict 3D AAM allows the designer to perform exposure and post-exposure bake and incorporates databases for exposure and development parameters for the most commonly used g, h and i-line photoresists.

In this thesis, Depict which is one of the modules employed in the EPCA system to calculate the two dimensional aerial image transferred onto the wafer for the parallel lines (database file) and for the IC mask layout. An example of the pattern transfer process using an SRAM cell is illustrated in figure 4.2. It can be observed that the difference between the mask layout and aerial image is quite significant and hence the effect of a defect landing on each layout will not be the same. The main differences

between the layouts are that track corners are rounded off and track widths are no longer equal throughout the length of the track.

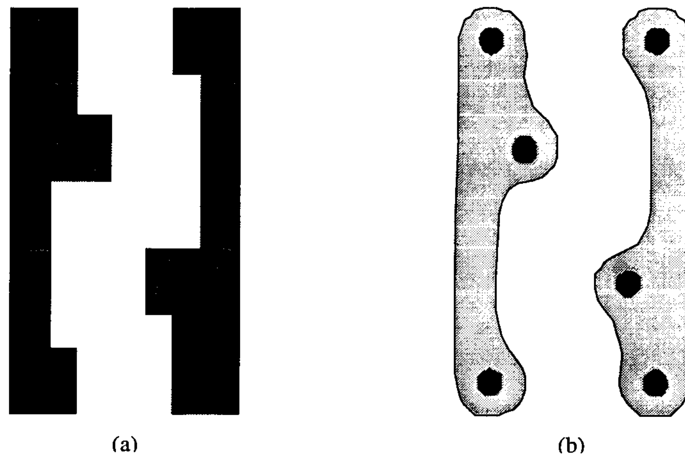


Figure 4.2.: (a) Mask layout and (b) simulated aerial image of the SRAM cell.

4.2.2. Edinburgh Yield Estimator (EYE)

EYE, the second module of EPCA, can be employed for the extraction of critical area and is capable of processing mask layouts with either Caltech Intermediate Format (CIF) or Graphical Design System (GDS) II formats. The purpose of EYE in EPCA is to calculate the critical area from both the mask layout file and the aerial image obtained from Depict.

As an example, figure 4.3 shows a fault probability map (FPM) generated for the SRAM cell shown in figure 4.2 with darker regions indicating where defects have the highest probability of causing a fault. It can be observed that there is a difference in the FPM from the two layouts. This can be seen in more detail in the lower half of the SRAM cell. The difference in the track shapes of the aerial image and mask layout has resulted in the FPM changing from a rectangular region to that of a trapezoidal region. In addition, the critical area of the layout changes from 8.82 to 8.72 when comparing the

mask layout with the aerial image and this results in an increased yield of 1 % for the aerial image.

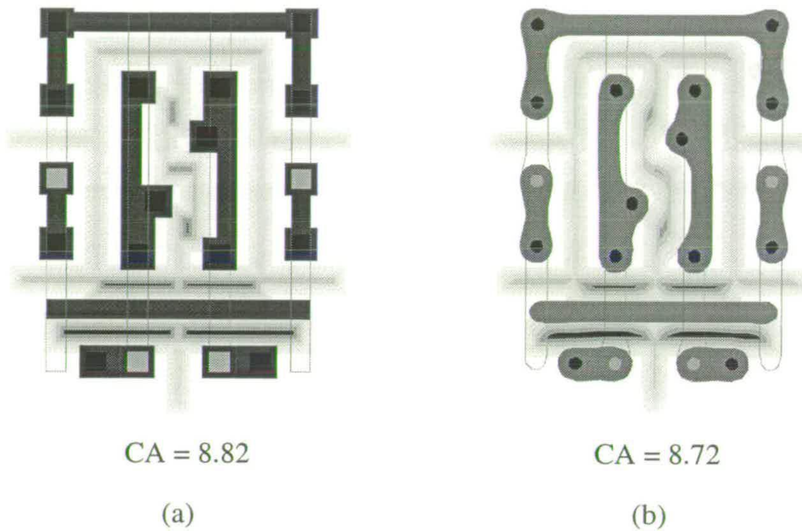


Figure 4.3.: FPM generated using EYE for Metal 1 for an SRAM cell. (a) Using the mask layout and (b) aerial image simulated from Depict.

4.2.3. Workbench (TWB)

Workbench, the third EPCA module integrates software tools to help automate the system for extracting critical area from a photolithographic simulation of a mask layout. In addition, TWB enables experiments to be assembled which can be used to investigate the effect of different input parameters on critical area. The experiment tree in Workbench consists of interlinked modules which utilise datafiles for both inputs and outputs. Thus, one of the features of Workbench is that external programs can be integrated with the in-built modules by ensuring the correct format for the inputs and outputs of each external program. The purpose of Workbench within EPCA is to automate the generation of the database files for parallel lines.

As an example, figure 4.4 shows the modules used for the extraction of critical area and the simulation tree built by TWB. The simulation tree represents the different

control factors being varied within the experiment and is represented by interconnecting modules where each branch of the tree is a different control factor used.

In this example there are five modules which are used in the simulation tree. The first module scales the CIF file to various technology sizes while the second module converts it into a file format that Depict utilises. The third module runs Depict while the fourth integrates the results into a CIF file. This is then used by the last module to run EYE which calculates the critical area.

Figure 4.5 shows the run table for the experiment where the two lower ticks indicate the outputs that are obtained while the topmost tick is the control factor used. In this case the minimum spacing (feature or defect size) was the control factor being varied while the cell name and critical area values are the outputs from the experiment.

Modules

- **Scalecif** : Automatically scales the CIF file to various technology sizes.
- **Cif2tl** : Reads in the CIF file and breaks up the various layers to a format that Depict understands.
- **Depict** : Runs the Depict simulator.
- **Tl2cif** : Merges the various layers of the photolithographic process back into a single CIF file.
- **Eye** : Runs the EYE tool.

Figure 4.4.: Module layout and experiment tree generated for an experiment.

Run Table

Design	Label	Name	ok 1	ok 2	ok 3	
✓	%MINS	minspac	0.4	0.6	0.8	
	%THRE	THRESI	0.3	0.3	0.3	
	%NA	NA	0.6	0.6	0.6	
	%LAMI	LAMBD	0.248	0.248	0.248	
	%POLE	POLE.R	0.6	0.6	0.6	
	Round	round$	0	0	0	
	Aspect	aspect$	1	1	1	
	Angle	angle$	0	0	0	
✓	Cell		SRAM	SRAM	SRAM	SRA
✓	CA		8.69686	8.60089	8.66837	8.69

Re-Read Results
Edit
View
DOE...
RSM...
Spread Sheet...
Print...
Help...
Close

Figure 4.5.: Run table in TWB used to modify the control factors in the experiment.

4.2.4. Photolithographic Critical Area Mapper (ATOM)

ATOM, the fourth module of EPCA is capable of mapping the critical area curve of a mask layout to a more realisitc curve which takes into account the effects of pattern transfer. It uses as input a control file which specifies the databases, input and output file names. The input file consists of the CA curve of the IC for which the CA curve after pattern transfer is to be determined. The databases consist of two files, one of which contains the CA curves of the orginal parallel lines (varying spacing) while the other consists of the CA curves of parallel lines (varying spacing) after pattern transfer. Figure 4.6 illustrates schematically the procedures implemented to calculate the final estimated photorealistic critical area curve. The two main procedures of ATOM are the generation of "Calibration Table" and "Equivalent Parallel Lines" and will be disscussed in the following sections.

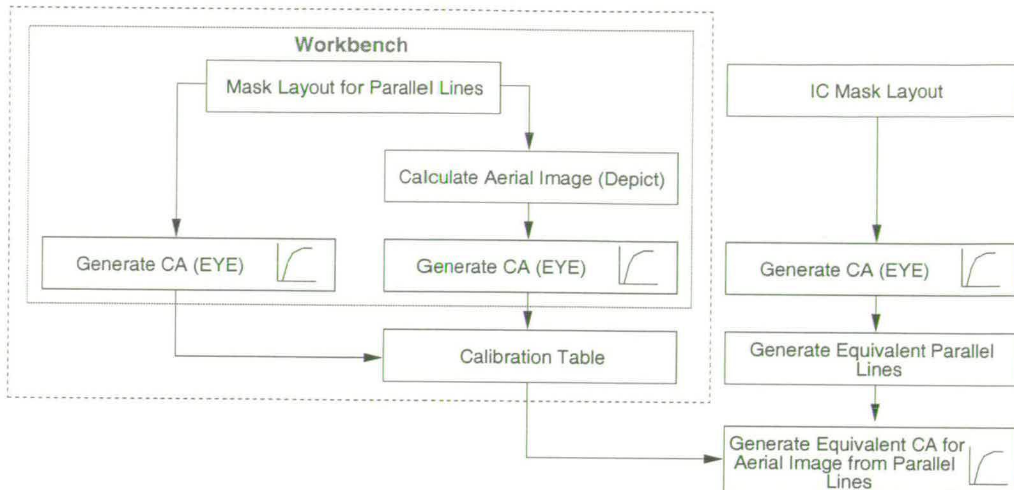


Figure 4.6.: Schematic diagram of the steps carried out to obtain the estimated photo-realistic CA curve of an IC.

4.2.4.1. Calibration Table Generation

The two datafiles specified as inputs to ATOM are used to create the calibration table. Each datafile consists of the CA values for the parallel lines (varying spacings) using different defect sizes. ATOM calculates the difference between the two datafile values to generate the calibration table. Hence, using this table a mapping can be obtained between the original parallel lines and that obtained after pattern transfer.

For example, if the CA value of a defect size of 0.5 μm is known for parallel lines with a spacing of 1.0 μm, the CA value of the same lines after pattern transfer can be obtained by summing the original value with the corresponding value in the calibration table.

4.2.4.2. Equivalent Parallel Lines Generation

If a mask layout can be represented by parallel lines, the estimated critical area after pattern transfer can be obtained from the calibration table. Hence a method is required that is capable of classifying a mask layout into equivalent parallel lines and this is

achieved using the Equivalent Parallel Lines (EPL) algorithm within ATOM. The EPL algorithm is recursive and uses the critical area values for the different defect sizes of the mask layout to generate the parallel lines that will result in the same critical area values. The algorithm employs a two dimensional matrix consisting of columns which represent the different parallel track separations while the rows represent the different defect sizes and is populated by critical area values (initially all set to zero). A schematic diagram of this matrix is illustrated in figure 4.7.

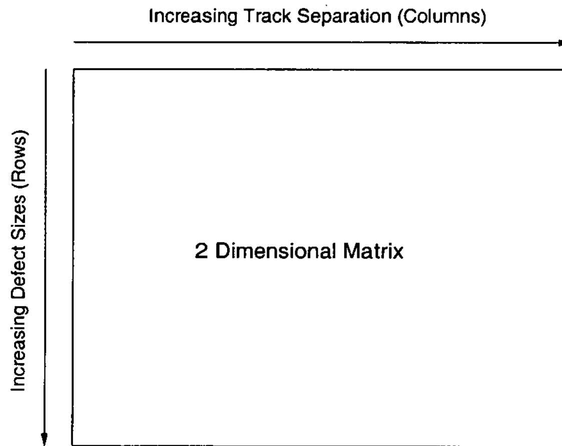


Figure 4.7.: A schematic diagram of the two dimensional matrix used in the EPL algorithm.

The number of rows, x depend upon the defect size range and increment while the number of columns, y depend upon the number of different parallel tracks with varying separations. The rows are then numbered from 1 to x while the columns are numbered from 1 to y. The algorithm starts from the top left corner (row 1 and column 1) of the matrix and moves through to the bottom right corner (row x and column y). The process of determining the critical area values is known as matrix propagation where the initial starting condition for the matrix is obtained from the critical area of the mask layout.

The critical area value calculated from the mask layout for the first defect size is then entered into the first column of the matrix and is the starting condition for the algorithm.

For the second defect size, the critical area in the first column for the second defect size can be determined and hence only the value of the second column has to be calculated. The difference between the critical area calculated from the mask layout and the critical area in the first column at the second defect size is used in the second column. Similarly for the third defect size, the difference between the critical area calculated from the mask layout and the critical area in the first two columns at third defect size is used in the third column. Figure 4.8 schematically illustrates the calculation of the critical area values for the first three defect sizes.

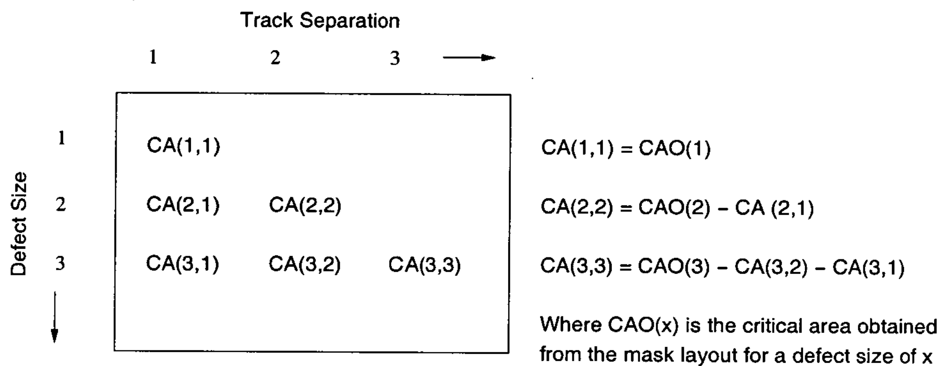


Figure 4.8.: A diagram showing the calculation of the critical area values for the first three defect sizes.

The remaining matrix values are calculated in a similar fashion. If at any defect size in the algorithm, the total critical area of the previous columns is larger than the critical area calculated from the mask layout which will result in a negative value being entered into the matrix, the value of the present column is set to zero. Modifications to the previous values in the matrix are then implemented which will recursively traverse back in the matrix compensating for over calculating the critical area. Once all the matrix values have been generated, it is merged with the calibration table to obtain the estimated critical area curve for the aerial image of the IC.

An example of how the EPL procedure is used is shown in figure 4.9 (a) which illustrates how a mask layout can be classified into three different parallel lines indicated by Region

A, B and C. The critical area curve of the mask layout can be obtained from the total of the three critical area curves as shown in figure 4.9 (b).

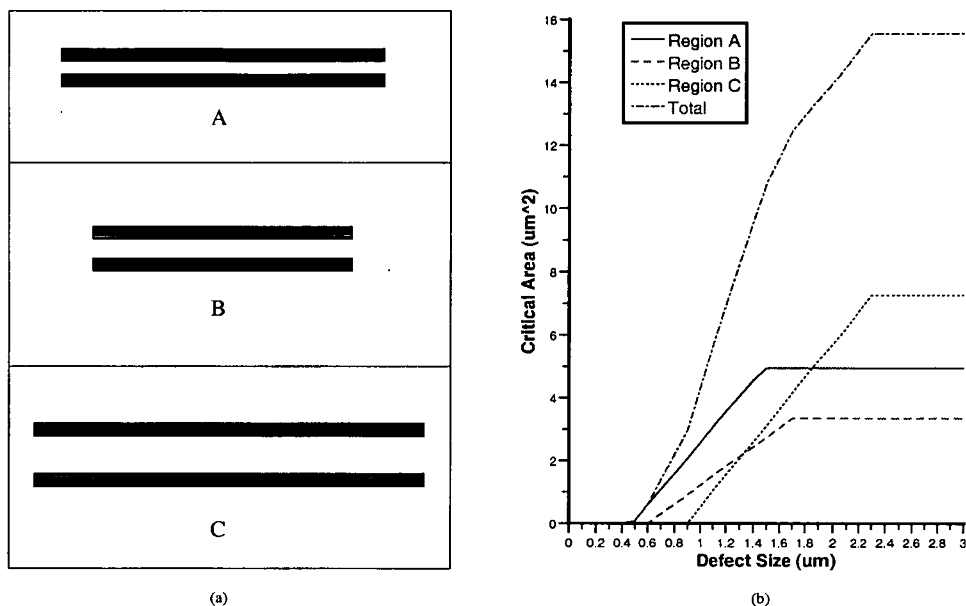


Figure 4.9.: Classification of a mask layout using the EPL procedure. (a) Three different sets of parallel lines obtained (Region A, B and C). (b) The CA curve for the mask layout obtained from the total of the individual CA curves for Region A, B and C.

4.3. Results

An SRAM cell was used to test the functionality of EPCA and more importantly the accuracy of ATOM. The SRAM cell has a small mask layout which enables Depict to calculate the critical area within a reasonable time. For the evaluation, a wavelength (λ) of 365 nm and numerical aperture (NA) of 0.6 were used. The feature size of the SRAM cell was 0.5 μm, which matches the parallel lines used to generate the database. The defect size used to calculate the critical area for the SRAM cell ranged between 0.1 μm and 3.0 μm at 0.1 μm intervals.

The results obtained from ATOM proved encouraging with the critical area curve generated from ATOM and consequently the Fault Probability (FP) curve providing a good estimate of the critical area curve obtained from Depict. The critical area curves are shown in figure 4.10 (a) and a magnified view for defect sizes ranged between 0.4 μm and 0.75 μm in figure 4.10 (b). The cumulative sum of the individual data curves obtained from ATOM for varying line spacings was used to form the critical area curve of the layout. In general, it should be noted that only a subset of these curves are used for a particular IC layout as not all masks will consist of all the available parallel lines of varying line spacing. The cumulative affect of these curves in estimating the critical area curve of the SRAM cell is illustrated in figure 4.11 (a) and a magnified view for defect sizes ranged between 0.4 μm and 1.0 μm in figure 4.11 (b).

It should be noted that, the cumulative curve for a spacing of 0.6 μm is the sum of the curves for 0.5 μm and that for 0.6 μm. Similarly, the cumulative curve for a spacing of 0.8 μm is the sum of the four individual curves consisting of spacings 0.5 μm, 0.6 μm, 0.7 μm and 0.8 μm. Thus it can be observed that as more individual curves for different spacings are added together, the cumulative critical area curve more closely resembles the critical area curve as estimated by ATOM in figure 4.10. Therefore for the SRAM cell, ATOM has used six equivalent parallel tracks of varying spacings and their associated critical area curves to estimate the critical area of the layout.

The close nature of the critical area curves makes it difficult to analyse the effects this has on the yield of the SRAM cell. Hence, in order to examine these differences, the FP curve was generated. This curve is the multiplication of the critical area curve with the defect distribution curve which has been taken as $\frac{1}{x^3}$, where x is the defect size. The area under this curve is the average number of faults (λ_{av}) and is proportional to the yield of the layout. The FP curves are shown in figure 4.12 (a) and the magnified view for defect sizes ranged between 0.6 μm and 1.8 μm in figure 4.12 (b).

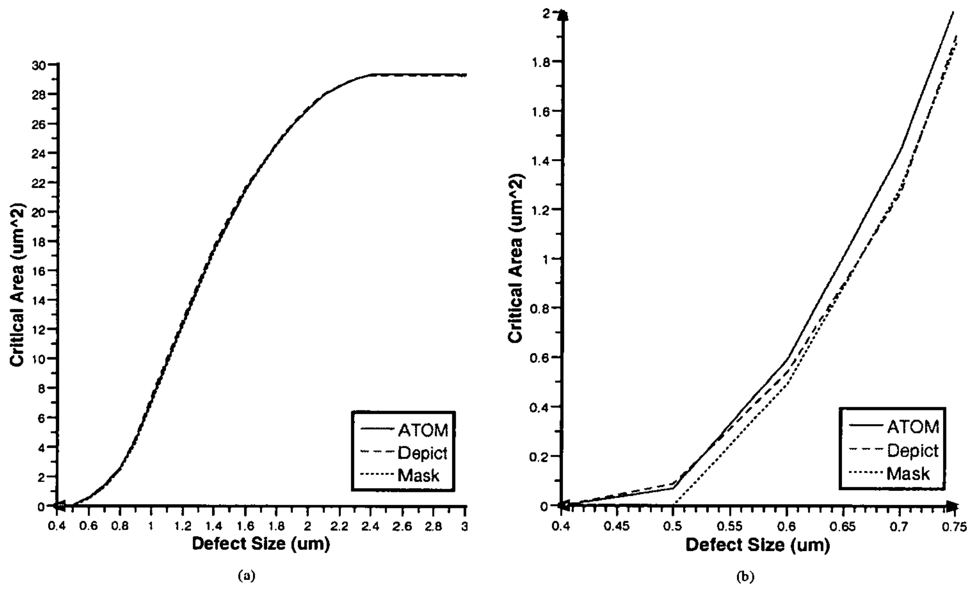


Figure 4.10.: Critical area as a function of defect size for the mask layout, Depict and ATOM for different defect size ranges.

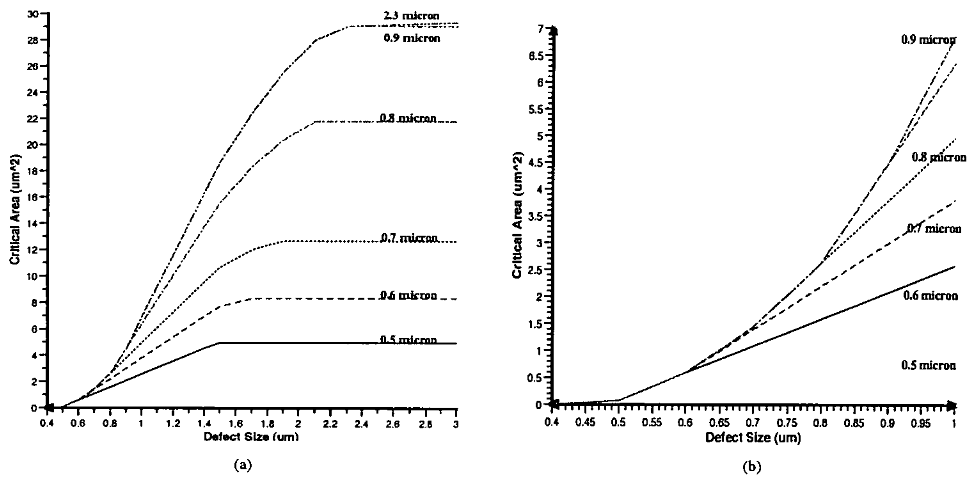


Figure 4.11.: Cumulative effect of the different curves used in the creation of the CA curve generated for the SRAM cell for different defect size ranges.

From figure 4.12, it can be observed that between defect sizes of 0.8 μm and 1.8 μm, ATOM underestimates the photolithographic process, whereas between 0.5 μm and 0.8 μm it overestimates the critical area. This limitation is a result of approximating a complex mask layout by simple structures (parallel lines in this case). The average number of faults and associated errors are shown in table 4.1 and it is interesting to note that the number of faults calculated by ATOM and Depict are within 0.5 % of each other which is more accurate for estimating yield than using the mask layout. However, it should be borne in mind that this result does not necessarily indicate this will be true in every case.

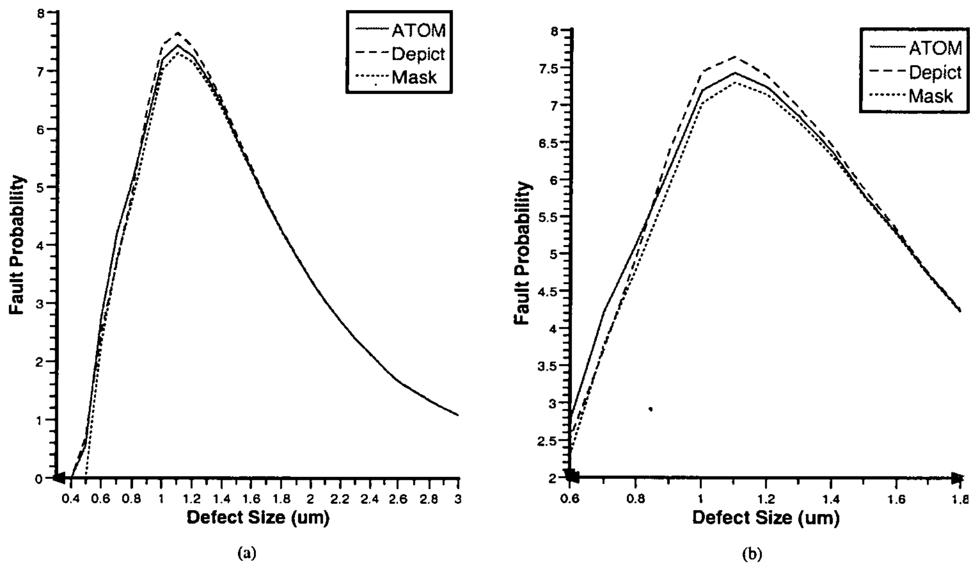


Figure 4.12.: Fault probability curve as a function of defect size for the mask layout, Depict and ATOM for different defect size ranges.

	Average number of faults, λ_{av} ($\times 10^{-8}$)	Difference in λ_{av} compared with Depict	Error in λ_{av} compared with Depict (%)
Mask layout	9.68	0.31	3.1
Estimated layout (ATOM)	9.94	0.05	0.5
Aerial Image (Depict)	9.99	-	-

Table 4.1.: Average number of faults and the associated errors compared with Depict for the SRAM cell.

In addition to the SRAM cell, an AMD 2901 chip (Alliance tutorial example) was also used to test ATOM by selecting a random area on the layout. This area was used by ATOM as the input file and is illustrated in figure 4.13 (a) by the bounding box drawn around the tracks. The fault probability curves calculated are shown in Figure 4.13 (b) and the average number of faults shown in table 4.2. The error in the average number of faults for ATOM is again seen to be more accurate for estimating yield than using the mask layout.

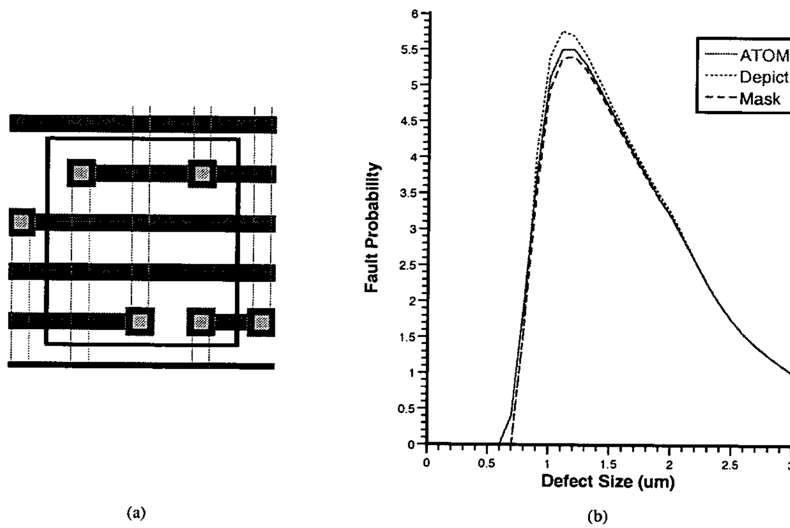


Figure 4.13.: Mask layout from the AMD 2901 chip used to calculate the fault probability curves. (a) The bounding box highlights the area for which the FP curves are generated from the layout. (b) Generated FP curves.

	Average number of faults, λ_{av} ($\times 10^{-8}$)	Difference in λ_{av} compared with Depict	Error in λ_{av} compared with Depict (%)
Mask layout	7.14	0.26	3.5
Estimated layout (ATOM)	7.30	0.10	1.4
Aerial Image (Depict)	7.40	-	-

Table 4.2.: Average number of faults and the associated errors compared with Depict for the AMD 2901 layout.

However, for layouts that do not lend themselves to representation by equivalent parallel lines, ATOM resulted in inaccurate estimates where better estimates were obtained from using the mask layout. To identify the probability of this occuring 7 random layout samples were selected from the AMD 2901 chip. Of these, two gave worse estimates than the mask layout being within 1.2 - 2.0 % of the aerial image compared with 2.9 - 3.9 % obtained from ATOM. The fault probability curves with the tables detailing the average number of faults and associated errors for the two sampled layouts can be found in Appendix C.

4.4. Summary

This chapter has demonstrated the use of Workbench to automate the extraction of photorealistic critical area for different mask layouts. The use of Depict is limited to small ICs because of the time required to obtain the aerial image in larger layouts. Hence, by employing ATOM with the Edinburgh Yield Estimator - Sampling (EYES), it can be used to calculate the critical area of large ICs. The main limitation of ATOM is that estimates are based on parallel tracks. Currently, ATOM has been tested by using parallel lines with a fixed track width of 0.5 μm with variable spacings and has been found to be capable of estimating the critical areas of mask layouts after pattern transfer by using an SRAM cell and AMD 2901 layout as examples. However, it should

be noted that ATOM is not capable of producing accurate results for all mask layouts due to the representation of complex structures by simple parallel lines.

5. Categorisation and Analysis of Mask Layouts (CALMA)

5.1. Introduction

The previous chapter discussed a method for estimating the critical area of a layout after pattern transfer using parallel tracks of varying separation. This method made no assumptions or attempt to obtain mask layout information and so valuable data was not utilised in this technique. Hence, an improved approach would be to use the layout information for the calculation of estimated critical area after pattern transfer. This chapter discusses a number of different approaches.

5.2. Initial Categorisation Attempts

5.2.1. Intepretation of Physical Characteristics in a Layouts

This method attempts to formulate a mapping between the physical characteristics of a layout and the estimated critical area curves after pattern transfer. It was theorised that for a given chip there may be a limited number of critical area curve groups. Here a curve group is defined as a collection of curves that possess similar characteristics although

their values might not be identical. A mask pattern transfer simulation program such as Depict can then be used on the different curve groups to obtain critical area curves after pattern transfer. Hence if a critical area curve from the original layout can be classified into a given group, its estimated critical area after pattern transfer can be obtained.

In order to test this method, physical characteristics of a layout have to be identified and large number of random layouts generated. EYES, a critical area analysis tool capable of sampling an IC was used on an AMD 2901 chip to obtain the random mask layouts and calculate critical area curves. Figure 5.1 schematically illustrates some random locations that EYES selected from the AMD chip to generate the different samples. A total of 512 samples were taken and figure 5.2 shows the corresponding mask layouts obtained from some of the samples.

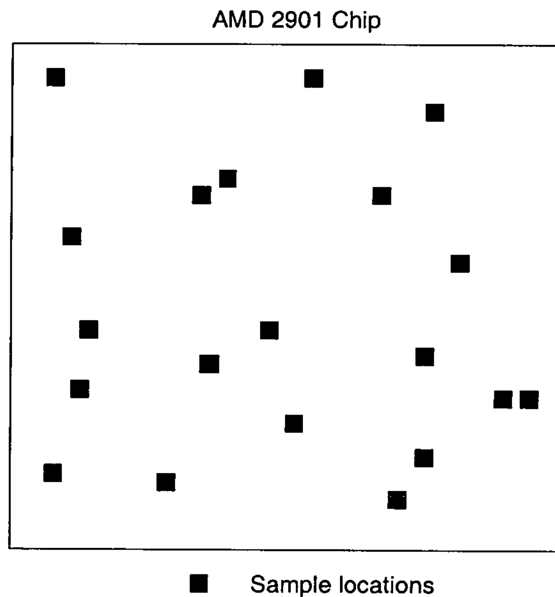


Figure 5.1.: Some random locations on the AMD chip that EYES selected for sample extraction.

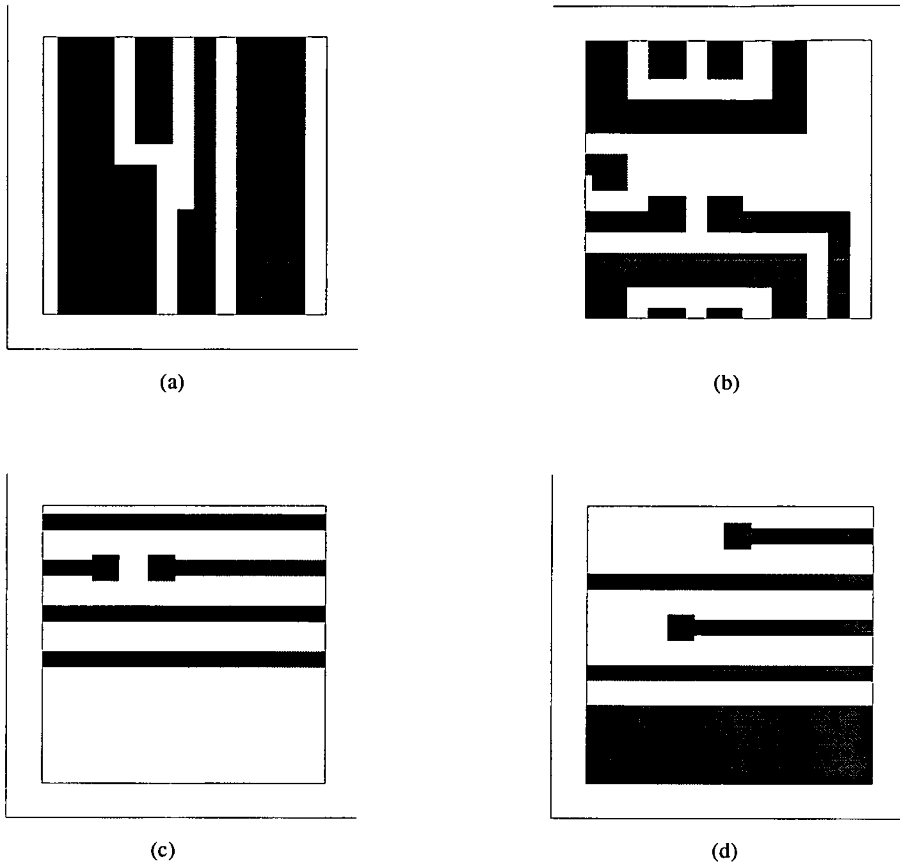


Figure 5.2.: Four mask layouts obtained from the corresponding samples where the shaded areas represent the tracks. (a) Sample number 47. (b) Sample number 53. (c) Sample number 106. (d) Sample number 113.

The critical area curves for the 512 samples was then calculated using EYES and transformed into fault probability curves. This was achieved by multiplying the critical area curves by the defect distribution ($\frac{1}{x^3}$). The next step was to select the physical characteristics that would be used to classify the layout. As a proof of concept, simple easily identifiable characteristics were used. The physical characteristics selected to classify the layout were

- Number of track intersections
- Number of horizontal lines
- Number of vertical lines

A visual inspection of the fault probability curves was carried out and it was possible to place the curves into roughly six different groups as illustrated in figure 5.3.

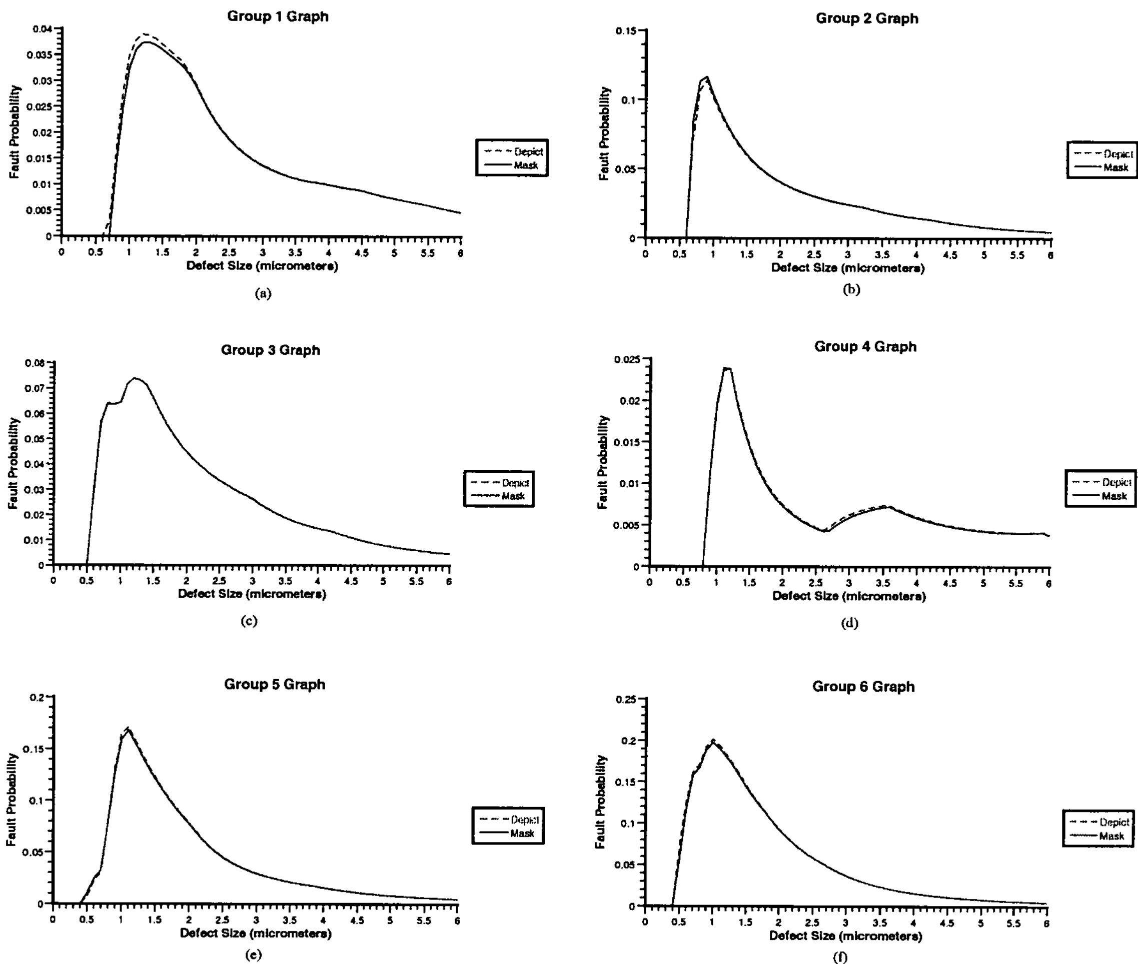


Figure 5.3.: The six different groups of fault probability curves. (a) Group 1. (b) Group 2. (c) Group 3. (d) Group 4. (e) Group 5. (f) Group 6.

However, upon a more thorough analysis it was observed that this method was not reliable as it would not be possible to always categorise certain layouts into the correct group. For example using sample numbers 106 and 113 in figure 5.2, the physical characteristics and curve groups for the samples can be obtained as shown in table 5.1. Although the two samples have similar physical characteristics, they belong to two different curve groups. One reason for this can be attributed to the different width, length and arrangement of tracks. Thus, the variety of physical characteristics proved

too problematic for a reliable mapping to be obtained and therefore an improved method of analysis is required.

Sample Number	Track Intersections	Horizontal Lines	Vertical Lines	Curve Group
106	0	5	0	1
113	0	5	0	5

Table 5.1.: Physical characteristics and the curve groups for two samples.

5.2.2. Critical Area Analysis on Parallel Tracks

The intepretation of mask layouts using physical characteristics proved impractical as there were too many physical variables involved and different layout groups. Hence, it was decided, at least initially, to limit the layout groups to the routing region of an IC. This layout type would consist only of parallel tracks with equal width and separation. Instead of using physical information of the layout, another classification technique was employed. The restriction placed on the layout (parallel tracks only) enabled the formulation of two classification techniques which map critical area curves to the estimated critical area curves after pattern transfer. The first is based on using boolean operators on the tracks to classify the layout while the second is based on categorising the layout using cross sections of tracks.

5.2.2.1. The Boolean Method

This method involved using boolean operators on geometric shapes which happen to be horizontal parallel tracks in this case. Some assumptions made on the layout is that the tracks are of uniform spacings and that the track widths are constant. It was theorised that it may be possible to determine the number of tracks, length of a track and location of a track relative to other tracks using these boolean operators with the critical area calculated. The AND operator is used to find the intersection of shapes whereas the

OR operator is used to obtain the union of shapes. A diagram of these operations can be seen in figure 5.4.

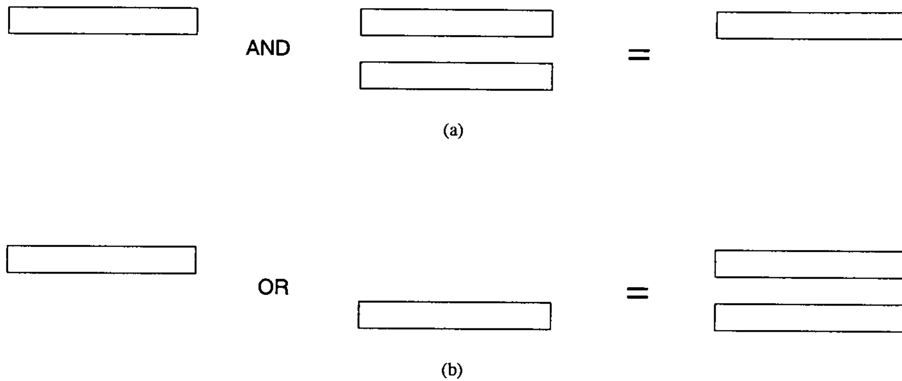


Figure 5.4.: Boolean operators on geometric shapes. (a) AND. (b) OR.

As illustrated in figure 5.4 (a) the result of the intersection of the two shapes is the topmost track whereas in figure 5.4 (b) the result of the union of the two shapes is the two parallel tracks. Shifting operators are also employed in this method. These operators move the original mask layout in the vertical or horizontal direction. A diagram of shifting operations is illustrated in figure 5.5.

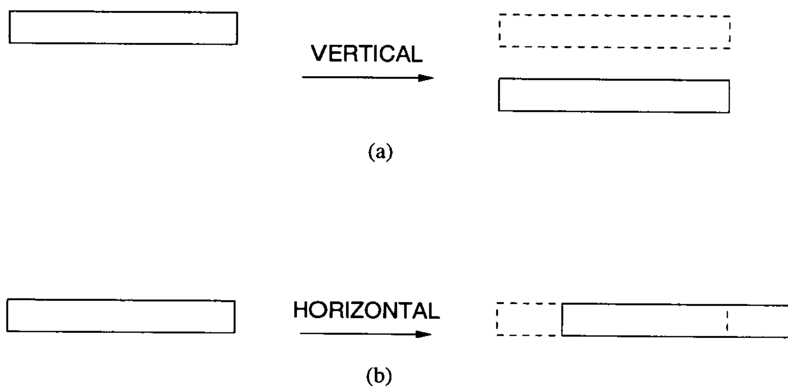


Figure 5.5.: Shifting operators to geometric shapes. (a) Vertical shift. (b) Horizontal shift.

In figure 5.5 a vertical shift down moves the track down while a horizontal shift to the right moves it to the right where the dashed outlines shows the original position of

track before shifting. The EYE program was employed in this method as it is capable of obtaining the resultant layout after boolean and shifting operations in addition to determining the critical area. It was found that the determination of the number of separate tracks was easier than the location of tracks on the layout.

Number of Tracks

In order to determine the number of tracks the layout was operated on itself using the logical AND operator. Using EYE to calculate the resulting number of intersections using the "Number" function gives the number of separate tracks in the layout. The disadvantage in using the AND operator is that it is unable to distinguish the relative placements of the tracks. Two possible arrangements of tracks that would result in the same answer from EYE are shown in figure 5.6. The tracks could be placed length wise one after another in a straight line or stacked one on top of the other to give parallel tracks. Therefore, it is necessary to be able to determine the length and location of individual tracks.

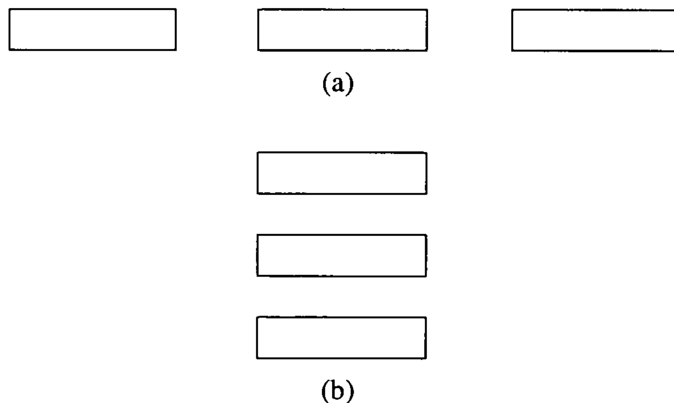


Figure 5.6.: Two possible track layouts giving similar results when using the AND operator to determine the number of tracks. (a) Length wise placement of tracks. (b) Stacked placement of tracks.

Length of Tracks

The determination of the individual lengths of the tracks was not feasible using only boolean and shifting operators on the original layout. One algorithm which was attempted had limited success with two parallel tracks. The algortihm is described as follows.

```
unit separation = track spacing + track width
defect size = track width
layout[1] = Original layout
for x = 2 to 4 do
layout[x] = Shift the original layout down by (x-1) unit separation and OR it with layout[1]
end
for x = 1 to 4 do
Result[x] = Critical area for layout[x]
end
Difference[1] = Result[2] - Result[1]
Difference[2] = Result[3] - Result[4]
Length[1] = (Difference[1])/(defect size) - 2 * (defect size)
Length[2] = (Difference[2])/(defect size) - 2 * (defect size)
```

The algorithm is able to calculate the two track lengths if the shorter track is located within the length of the longer track as shown in figure 5.7 (a) and (b). The three track types in figure 5.7 are used in a worked example where the longer track is labeled "track 1" while the shorter track is labeled "track 2". The defect size, track width and track spacing was set to 0.5 μm and the results obtained using the algorithm are detailed in table 5.2.

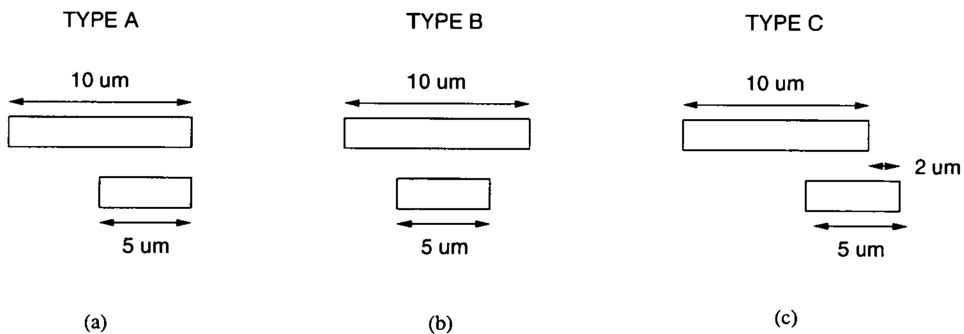


Figure 5.7.: Three possible track layouts.

Track Type	Length of track 1 (μm)	Length of track 2 (μm)	Success
A	10	5	Yes
B	10	5	Yes
C	12	3	No

Table 5.2.: Length of the two tracks obtained from the three track types.

For track type C in figure 5.7 (c) the algorithm overestimates the length of the longer track while underestimating the length of the shorter track. This error is given by the amount the shorter track extends past the edge of the longer track. Although the number of tracks could be determined, the individual length and placement of the tracks was still indeterminable. Thus, an alternative method which provided a better categorisation technique was required.

5.2.2.2. The Cross Section (1-Dimensional) Method

The premise behind this method is that the critical area of the entire layout can be calculated from the sum of the individual cross sectional critical areas. It is also known as a 1-dimensional method as the layout is categorised in one dimension. The method will be applied to horizontal parallel tracks which will first be divided into unit cross section areas where the horizontal length (unit length) of the area will be equal to the track width. Similar assumptions are made as with the boolean method which are that tracks are of uniform spacings and that the track widths are constant.

The presence or absence of a parallel track in the unit cross section is represented by a binary number (1 or 0) as can be observered in figure 5.8. The number of bits in the binary number equals the number of parallel tracks in the layout. This binary number is used to classify the type of tracks present at the cross sections as seen in figure 5.9. In addition, each track type consists of a "left end", "right end" and "middle" identifier. This identifier is used in determining if the track type is at the end or middle of a cross section group as illustrated in figure 5.10.

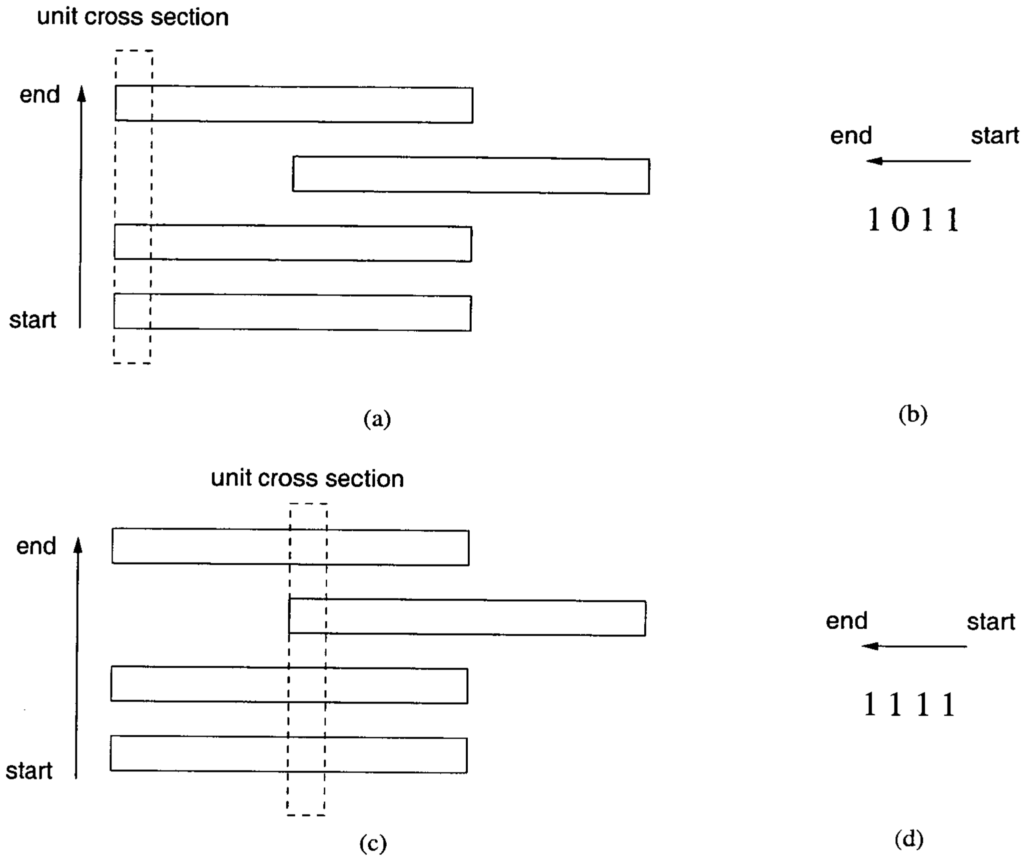


Figure 5.8.: Obtaining the unit cross section value of a layout. (a) Unit cross sectional area taken at the left of layout. (b) Classification of the area. (c) Unit cross sectional area taken in the middle of the layout. (d) Classification of the area.

Once the track types have been obtained, Depict is used to calculate the critical area curves for both the "ends" and "middle" identifiers in a unit cross section as illustrated in figure 5.11. Thus using the CA values calculated after pattern transfer for the track types and adding the individual values the estimated CA of the entire layout can be determined and will be further explained in the example that follows.

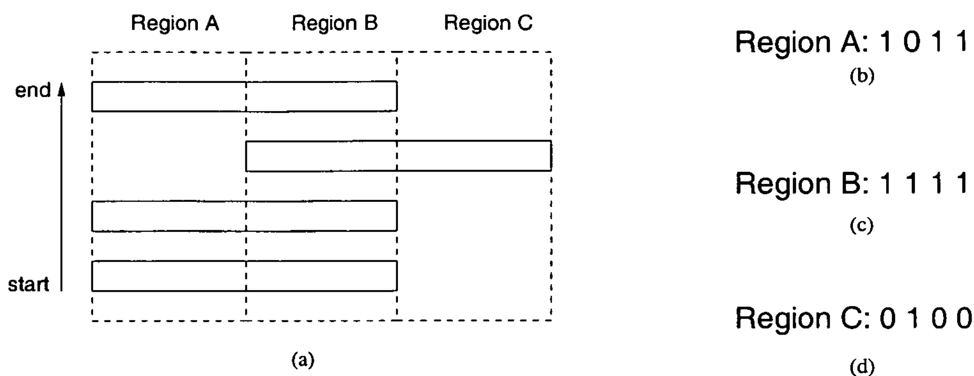


Figure 5.9.: Classification of the track types in a layout. (a) Three regions used to classify the layout. (b) Classification of Region A. (c) Classification of Region B. (d) Classification of Region C.

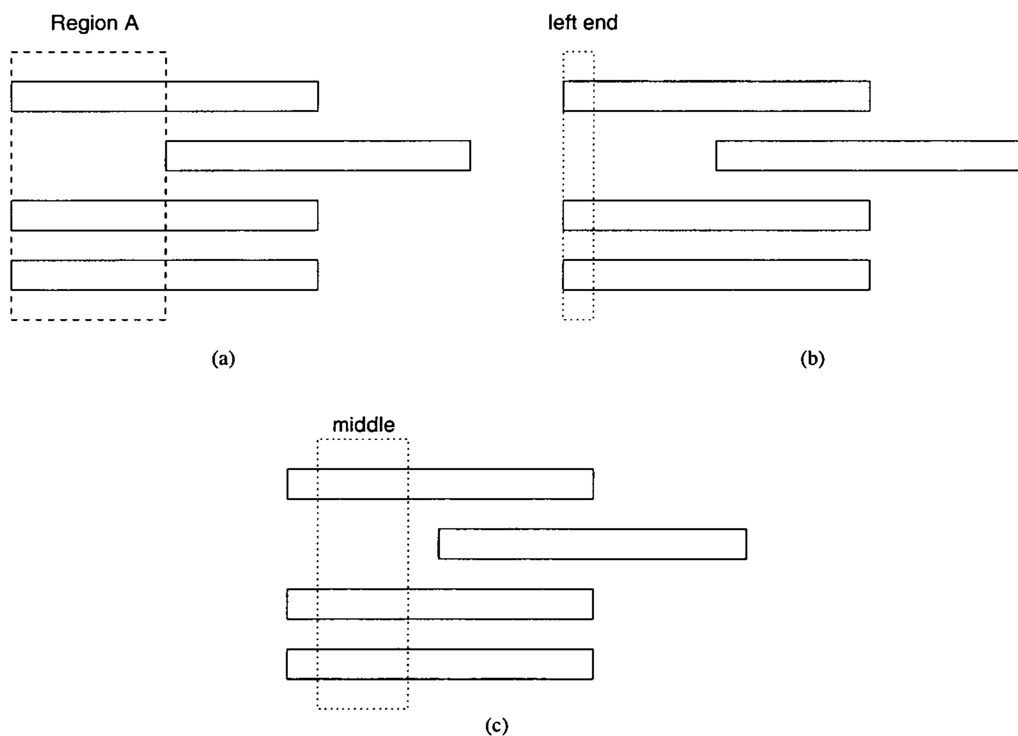


Figure 5.10.: Determining the left end and middle identifiers in a track type group. (a) Region A group for which the left end and middle identifiers are to be determined. (b) The end identifier for Region A. (c) The middle identifier for Region A.

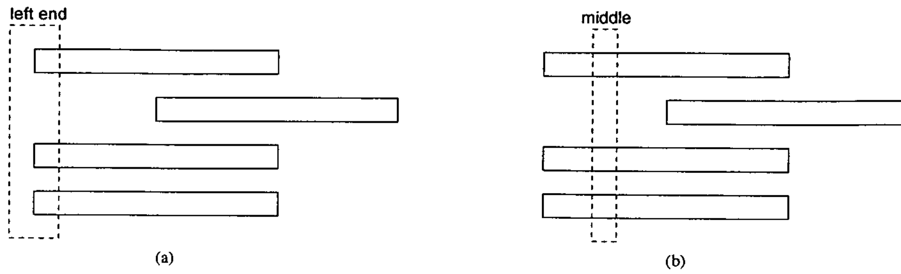


Figure 5.11.: CA calculation using Depict on track type 1011 for the end and middle identifiers. (a) CA calculation for the end identifier. (b) CA calculation for the middle identifier.

To illustrate the technique, an example layout will be employed which consist of three parallel tracks with given lengths as illustrated in figure 5.12. Starting from the left of the layout and moving to the right, the following track types are obtained 101, 111 and 010 respectively. In this example, a defect size and unit length of 1μm is used. Figure 5.12 (a) shows the breakdown of the various track types with the associated length (for middle and end identifiers). In figure 5.12 (b) the unit lengths for the track types have changed due to the increase in track lengths (illustrated by the dotted lines) which occurs when calculating the critical area of a track. Hence using the Depict data, the sum of the critical area for the middle and ends identifiers for all the track types will give the resulting critical area of the layout as detailed in table 5.3. A comparison between the CA obtained using this method and from Depict can be seen in table 5.4.

One drawback of this categorisation method is that it is limited to parallel tracks. A more general technique of track classification is required if the layout is not limited to parallel tracks and a techinque which addresses this is presented in the following section.

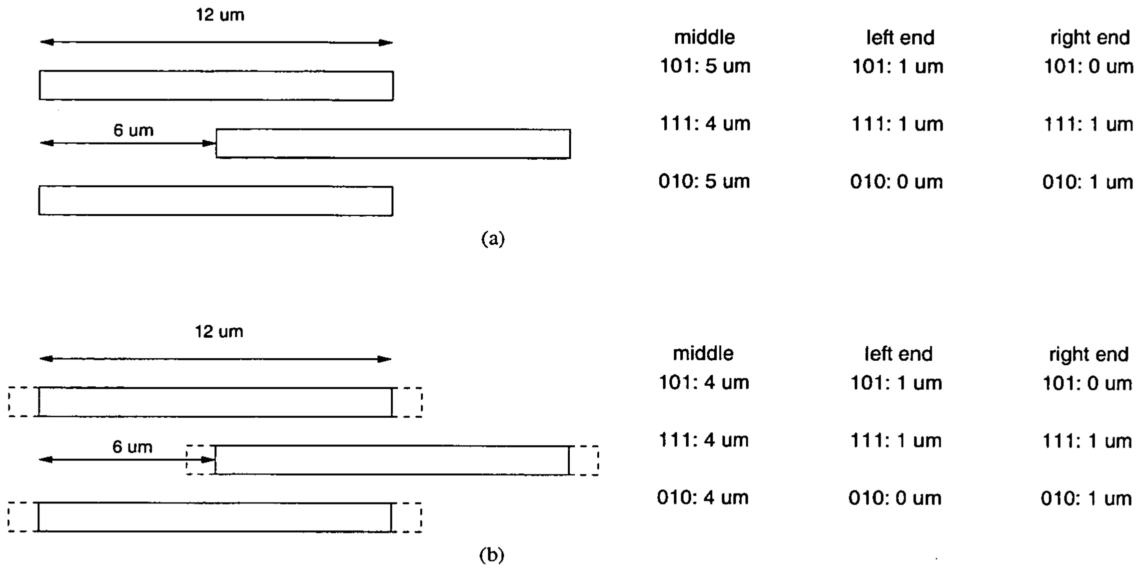


Figure 5.12.: Example of the method employed. (a) Mask layout. (b) Mask layout lengthened for the calculation of critical area for a defect size of $1\mu m$

Track Type	unit CA (μm^2)	Number of units	Total CA (μm^2)
101(middle)	0	4	0
101(left end)	0	1	0
101(right end)	0	0	0
111(middle)	1.95410	4	7.81640
111(left end)	3.90144	1	3.90144
111(right end)	3.90488	1	3.90488
010 (middle)	0	4	0
010 (left end)	0	0	0
010 (right end)	0	1	0

Table 5.3.: Table illustrating the calculation of CA for the different track types.

	Critical Area (μm^2)	Error (%)
Estimated	15.6227	-0.19
Mask	16	2.2
Aerial Image (Depict)	15.6527	-

Table 5.4.: Total CA for the layout and the associated errors.

5.3. Categorisation Method

The previous categorisation attempts were either somewhat impractical or too restrictive and could not be applied to general layout formats. One way to overcome this limitation is to employ a method that can categorise (or identify) the layout. By categorising the layout, information regarding the presence and type of tracks can be obtained. Hence, a collection of tools have been created that will assist in the categorisation and analysis of a mask layout and are collectively known as CALMAT which means "Tools for Categorising and Analysis of Mask Layouts".

A brief overview of the inputs and outputs used by CALMAT is shown in figure 5.13. Inputs to CALMAT include a control file which is used to specify the various program parameters, combined datafiles (database) and the mask layout of the device whose critical area is to be estimated. One of the outputs (result file) includes a file containing the estimated critical area after pattern transfer. In addition, an error graph can be produced from comparing results from the estimated and actual critical area values obtained after a photolithographic simulator is used on the mask layout. The following sections will describe the CALMAT modules in more detail. The user and reference manuals for CALMAT are presented in appendices A and B respectively.

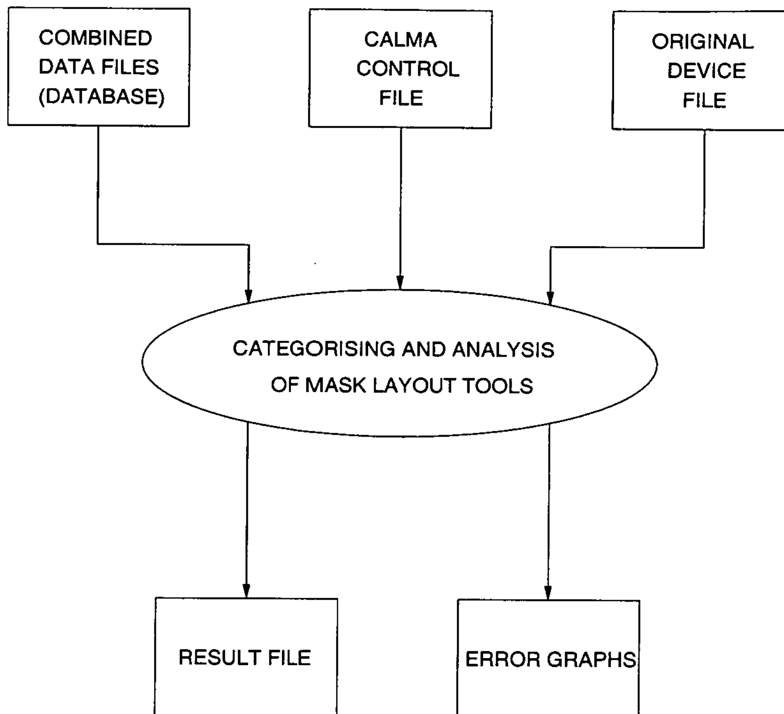


Figure 5.13.: Schematic overview of the inputs and outputs used by the categorisation tools (CALMAT).

5.3.1. Tools for Categorising and Analysis of Mask Layout (CALMAT)

5.3.1.1. Main Program (Calma)

Calma is the main program used within CALMAT and integrates and controls modules. A detailed schematic diagram of the interaction between the modules of CALMAT with their input and outputs is illustrated in figure 5.14. The rectangular boxes represent inputs and outputs while the modules are represented by ovals. It should be noted that the control file contains the necessary parameters for the various modules and user intervention is not required for any of the individual modules highlighted within the dotted box in figure 5.14.

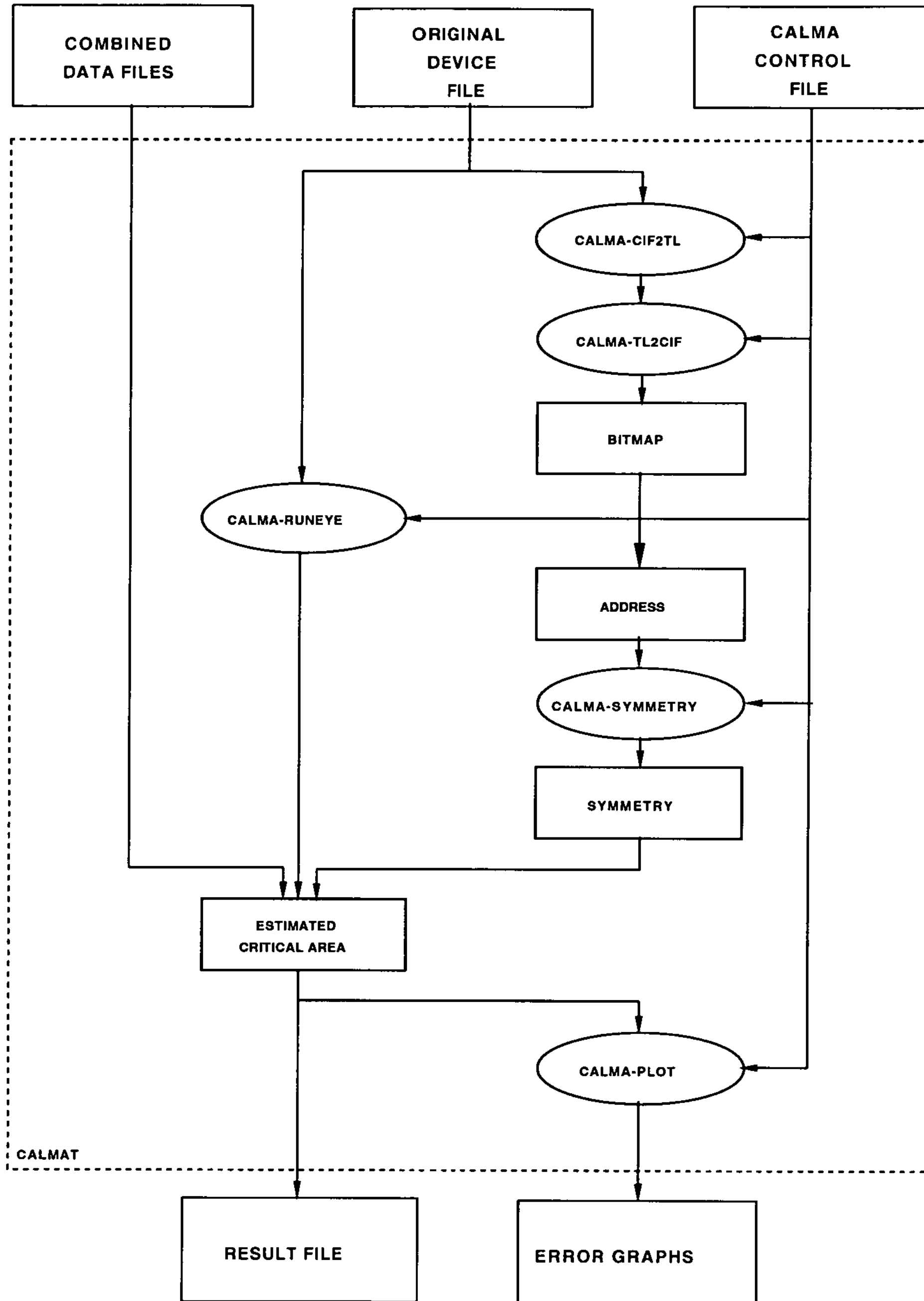


Figure 5.14.: Detailed schematic diagram of the interactions between the modules of CALMAT.

5.3.1.2. Bitmap Generation (Calma-cif2tl and Calma-tl2cif)

In order to categorise a mask layout it first has to be modified into a form that Calma is capable of processing. This is where bitmap generation plays a part and is essential in attempting to categorise a layout. Calma uses the bitmap file obtained to generate an internal picture of the mask layout which is then categorised. Two modules are employed for bitmap generation. The first, calma-cif2tl takes as input the mask layout of the device and calculates the bounding region of the layout which is then converted to a TL2 (Depict) file format. The second module, calma-tl2cif uses the TL2 file created and based on the region size specified calculates and generates the new bounding region for the device.

Calma then runs EYE to generate the bitmap file. The bitmap is calculated by using the "number" function of EYE for a unit size (bit) bounding box on the layout. The size of the bounding box is dependent upon the track width and spacing between tracks. For horizontal parallel tracks, the horizontal length of the box is equal to the track width while the vertical length is equal to the track width plus track spacing. If the bounding box is on a portion of the layout then a value of "1" is returned otherwise "0" is returned. An example to show the process used in bitmap generation with two parallel tracks is illustrated in figure 5.15. The unit size bounding box in figure 5.15 (c) results in a value of "1" being returned when forming the bitmap for the parallel tracks and takes into account the separation between tracks. Hence the two parallel tracks result in two rows of "1" in the bitmap as shown in figure 5.15 (d).

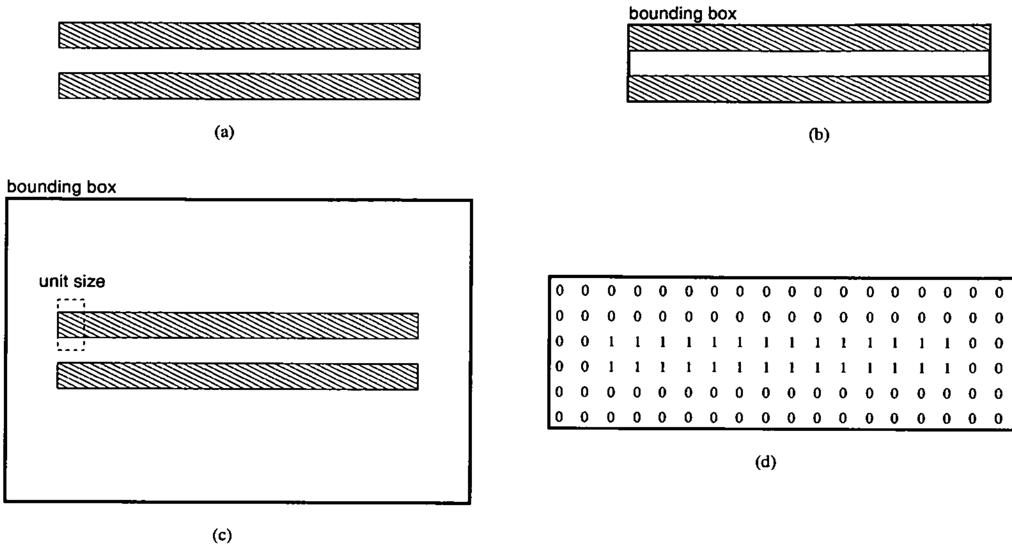


Figure 5.15.: Process used in bitmap generation on two parallel tracks. (a) Two parallel tracks. (b) Bounding box generated by Calma-cif2tl. (c) Expanded bounding box based on region size. (d) Bitmap generated by Calma-tl2cif using the EYE "number" function.

5.3.1.3. Address Symmetry Calculation (Calma-symmetry)

Once the bitmap of the mask layout has been generated, Calma uses this information to categorise the layout. This is when the Calma-symmetry module is used and it has two functions. The first is to categorise the individual bits (unit size boxes) obtained from the bitmap. Each bit is assigned an address based upon the values of the surrounding bits and this essentially identifies the type and relative location of the bit in the layout. The second function is to determine the similar addresses that would be obtained from symmetry in the layout.

In addition, the Region Size (RS) setting used in the Calma control file affects the bit address. Each bit address is composed of the same number of Region Address (RA) as the RS value used. For example, a bit address using a RS of 3 will be composed of three RAs, which are RA 1, RA 2 and RA 3. To determine the bits associated with a specific RA value, the distance (in bits) from the address bit is used. Thus any bits which is the

same distance away as the RA value are associated. As an example all bits a distance of two from the address bit will form the RA of 2. The RA is determined by starting from the top left corner of the associated bits and moving in a clockwise direction. Hence, the length of each RA increases as the RA value increases and is given by the formula $8 \times$ (RA number). Therefore a RA of 3 would have an address length of twenty four.

A visual representation of the categorisation process for a RS of one on two parallel tracks is illustrated in figure 5.16. For a RS greater than one, the bit address is represented by the individual RA separated by a colon (e.g. a bit address for RS 2 will have an address in the form of 00000001:0000000000000001). Therefore to simplify the representation of bit addresses, the binary value of the address is converted to decimal. A diagram illustrating the bit address calculation for a region size of 2 on two parallel tracks is illustrated in figure 5.17.

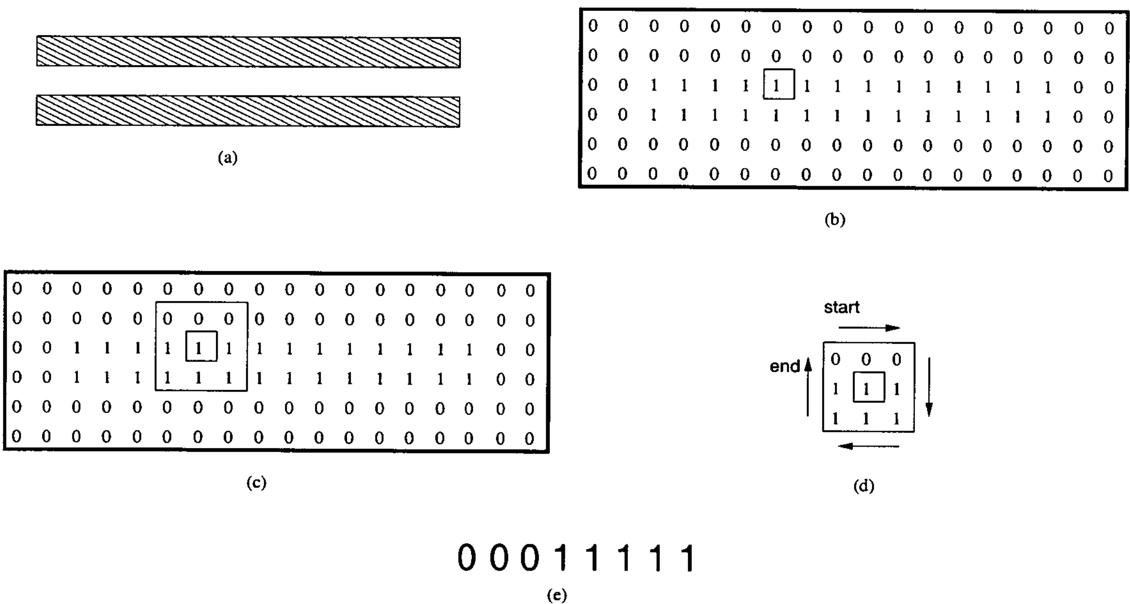


Figure 5.16.: A visual representation of address calculation for a region size of 1 on two parallel tracks. (a) Two parallel tracks. (b) Bitmap generated for the parallel tracks and a boxed bit which is to have its region address determined. (c) Surrounding bits for a region size of 1. (d) Determination of the region address starting from top left and moving in a clockwise direction. (e) Region address for the bit.

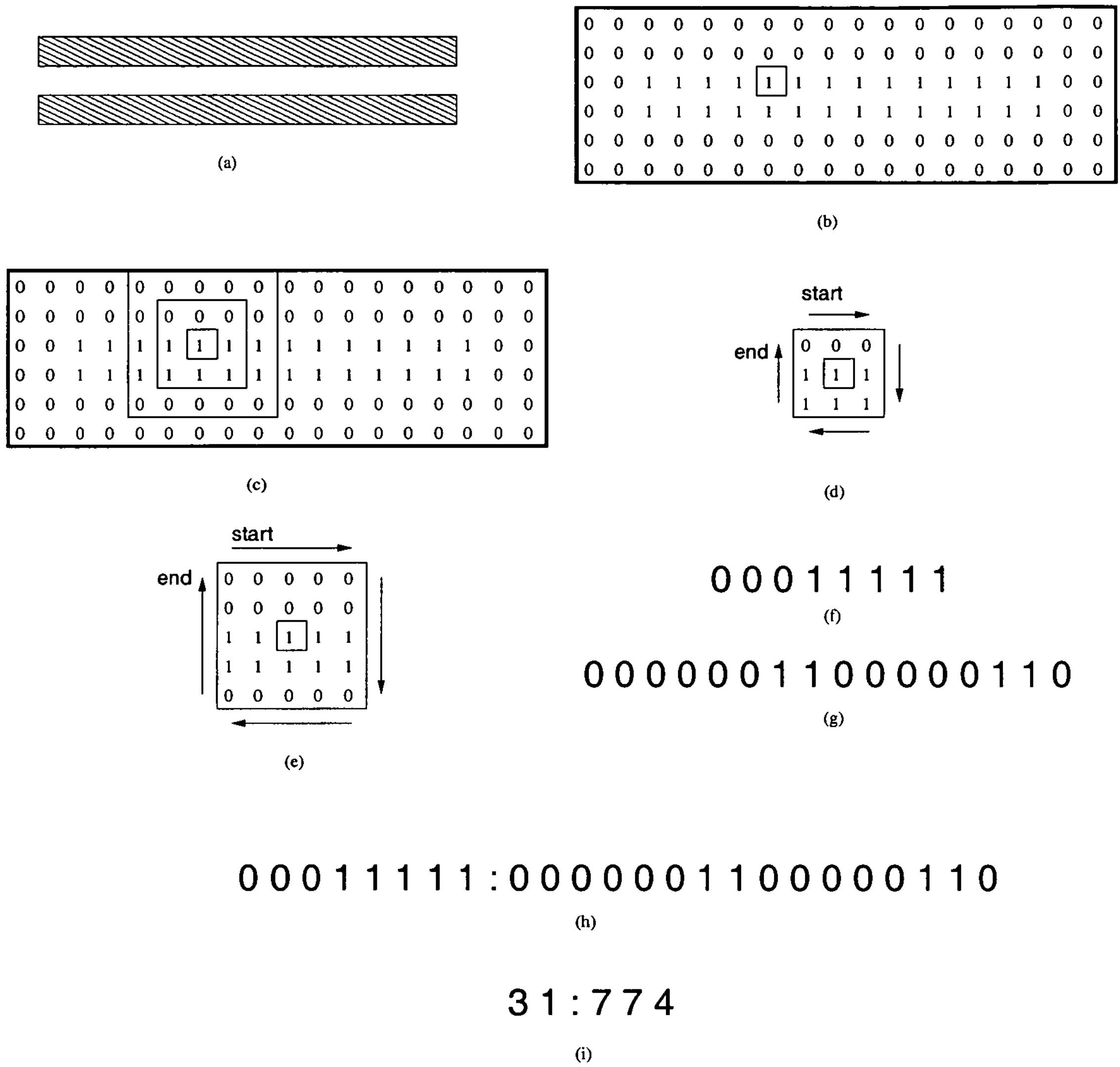


Figure 5.17.: Diagram illustrating bit address calculation for a region size of 2 on two parallel tracks. (a) Two parallel tracks. (b) Bitmap generated for the parallel tracks and a boxed bit which is to have its bit address determined. (c) Surrounding bits for a region size of 1 and 2. (d) Determination of the region address (size = 1) starting from top left and moving in a clockwise direction. (e) Determination of the region address (size = 2) starting from top left and moving in a clockwise direction. (f) Region address (size = 1) for the bit. (g) Region address (size = 2) for the bit. (h) Bit address from the combination of region addresses. (i) Bit address represented in decimal.

Calma-symmetry is also capable of determining symmetry in a layout which is used to reduce the number of bit addresses calculated. A layout can have different forms of symmetry such that some bit addresses would be identical if it was flipped or rotated along the x or y axis. Figure 5.18 illustrates some of the possible symmetry types in a layout where the line represents the plane of symmetry.

The effect of symmetry in a layout can be modelled by using a bit shifting algorithm which shifts each bit in an address clockwise cyclicly. From figure 5.18 it can be observed that there are four possible symmetry types in a layout. Therefore, the algorithm has to shift each bit address by a value which will enable it to obtain the four different bit patterns. The increased number of bits in larger RA values implies that the amount shifted has to be proportonal to the RA value and is given by 2 $\times$ (RA number).

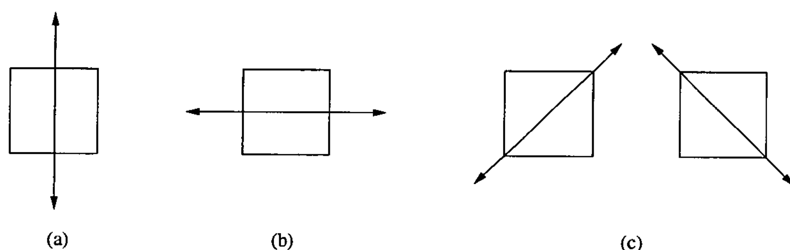


Figure 5.18.: Some symmetry types in a layout. (a) Vertical symmetry. (b) Horizontal symmetry. (c) Diagonal symmetry.

However it should be noted that not all layouts will result in the proper symmetry patterns if bit shifted through all the four combinations. For example, in figure 5.19 the layout is only symmetrical only about the x or y axis and hence a modification to the algorithm has to be made. This can be achieved by changing the amount shifted to 4 $\times$ (RA number). If the unmodified symmetry algorithm were used to calculate symmetry in figure 5.19, it would classify the two x points and the two y points identical. However it can be seen that only the x points are symmetrical as are only the y points and this is obtained from the modification made to the algorithm.

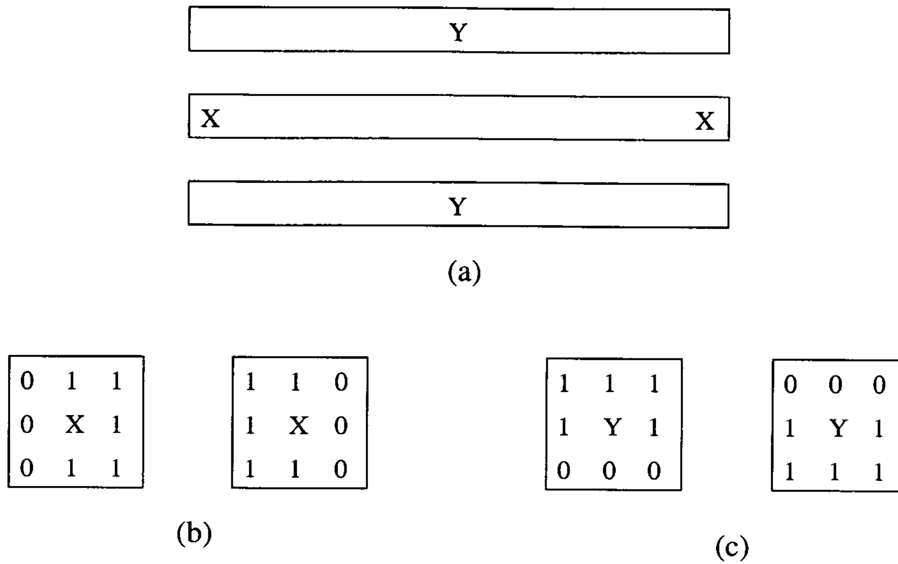
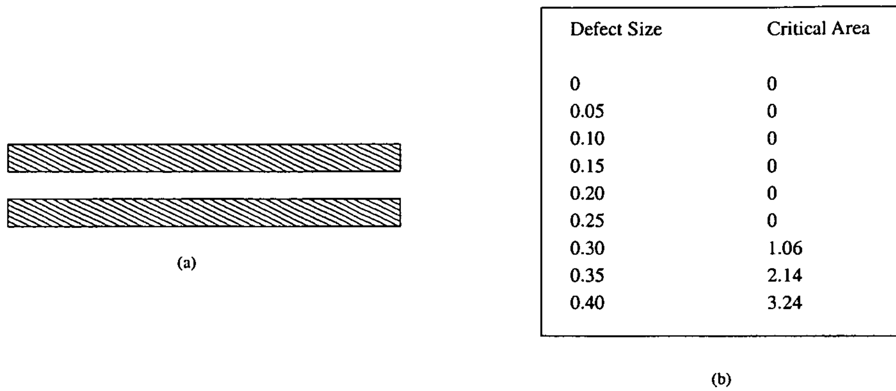


Figure 5.19.: A layout illustrating symmetry in the x or y axis only. (a) Three parallel tracks where the bit address is calculated at points x and y. (b) Symmetry in the y axis for points x. (c) Symmetry in the x axis for points y.

5.3.1.4. Critical Area Calculation Module (Calma-runeye)

This module (Calma-runeye) is one of the key parts of CALMAT and is utilised for all critical area calculations. However, it should be noted that Calma-runeye does not calculate the critical area of a layout but rather runs an external critical area extraction program. Therefore, by modifying the source code of Calma, different tools can be used for calculating critical area. EYE was used as the critical area extraction program as it is locally available. The results obtained from Calma-runeye is stored in a result file which allows other CALMAT modules to easily access the data. Figure 5.20 llustrates the generation of the result file from two parallel tracks.



Defect Size	Critical Area
0	0
0.05	0
0.10	0
0.15	0
0.20	0
0.25	0
0.30	1.06
0.35	2.14
0.40	3.24

(a) (b)

Figure 5.20.: Generation of CA result file from two parallel tracks using Calma-runeye. (a) Two parallel tracks. (b) Extract of the result file.

5.3.1.5. Database Generation (Calma-database)

The previous sections described the generation of unique addresses and critical area calculation for a layout. CALMAT operates by summing the critical area calculations for each address to determine the critical area for the entire layout. Hence if the critical area after pattern transfer for each address is obtained, the estimated critical area of the layout can be determined. One method to calculate the photolithographic critical area of an address is to use Depict and Calma-runeye. The resulting file (datafile) consist of the critical area calculations for the defect sizes used.

Hence, to better organise the individual datafiles, Calma-database is used to merge the files into a database. The advantage of generating a database is the reuseability of the address critical area values for different layouts. In addition, if an address is not present in the database, it is a simple procedure to append it using Calma-database. Figure 5.21 illustrates schematically the database generation from individual data files.

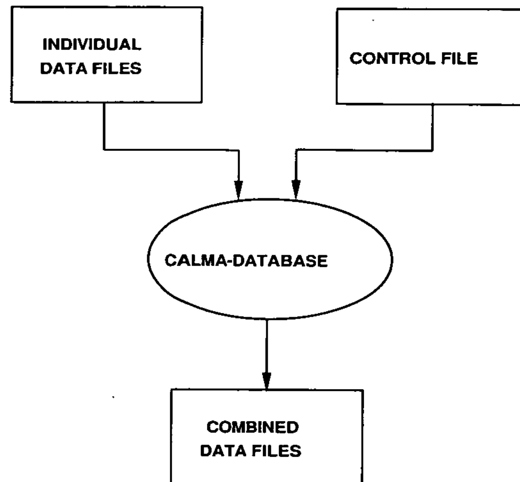


Figure 5.21.: Schematic diagram of database generation from individual data files

5.3.1.6. Plot Generator (Calma-plot)

The results obtained from CALMAT is stored in two text files. The first file, Critical Area Results (CAR) presents the critical area calculations and the asociated errors while the second file, Fault Probability Results (FPR) present the calculated fault probability values as obtained from the CAR file.

The contents of the CAR file consists of five columns, three of which are critical area calculations while the remaining two are error calculations. The first calculated critical area is obtained from the mask layout (labelled "Critical Area"), while the second is from the layout which has undergone pattern transfer (labelled "Critical Area (dep)"). The last critical area is the estimated value generated by CALMAT (labelled "Critical Area (est)").

The first error column labelled "Original Error" can be determined by calculating the percentage difference between the critical area values obtained from the mask layout and the critical area obtained after pattern transfer. Similarly, the second error column labelled "Estimated Error", is obtained from the percentage difference between the crit-

ical area value estimated from CALMAT and the value obtained after pattern transfer. An example extract of a CAR file is illustrated in figure 5.22 (a).

The contents of the FPR file consists of three columns which are the fault probability curves generated from the corresponding three critical area columns of the CAR file. The columns are labelled similarly with the change of "Critical Area" to "Fault Probability Curve". For example, the label "Critical Area (est)" for the estimated critical area changes to "Fault Probability Curve (est)" for the corresponding fault probability curve. An extract of the FPR file is shown in figure 5.22 (b). Therefore, to better visualise the results, Calma-plot uses the two result files and generates the corresponding plots for critical area and fault probability in postscript format. Two example plots generated from the CAR and FPR files are illustrated in figure 5.23.

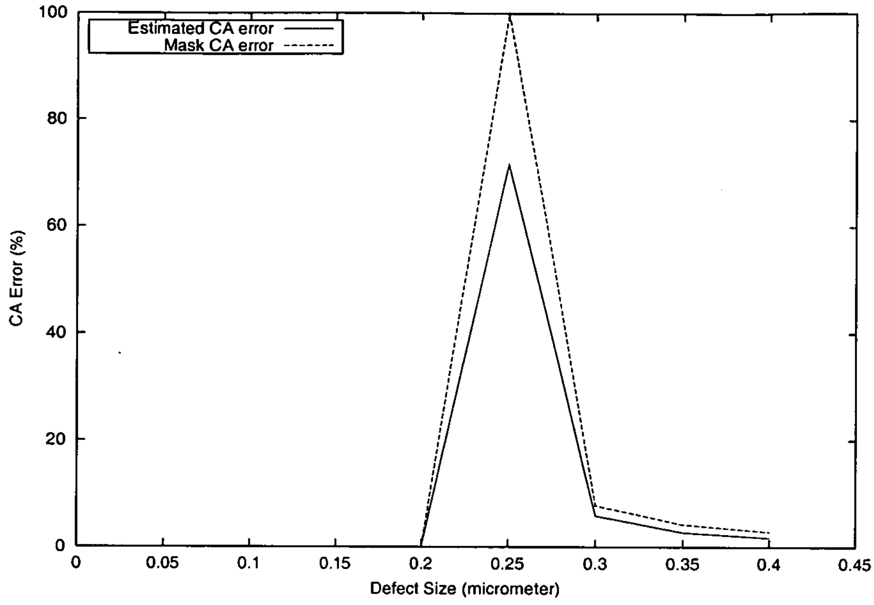
Defect Size	Critical Area	Critical Area (dep)	Critical Area (est)	Estimated Error (%)	Original Error (%)
0	0	0	0	0	0
0.05	0	0	0	0	0
0.10	0	0	0	0	0
0.15	0	0	0	0	0
0.20	0	0	0	0	0
0.25	0	0.01283	0.02202	71.65790	100
0.30	1.06	0.98391	1.04113	5.81547	7.73376
0.35	2.14	2.05353	2.10957	2.72887	4.21080
0.40	3.24	3.15082	3.20552	1.73595	2.83037

(a)

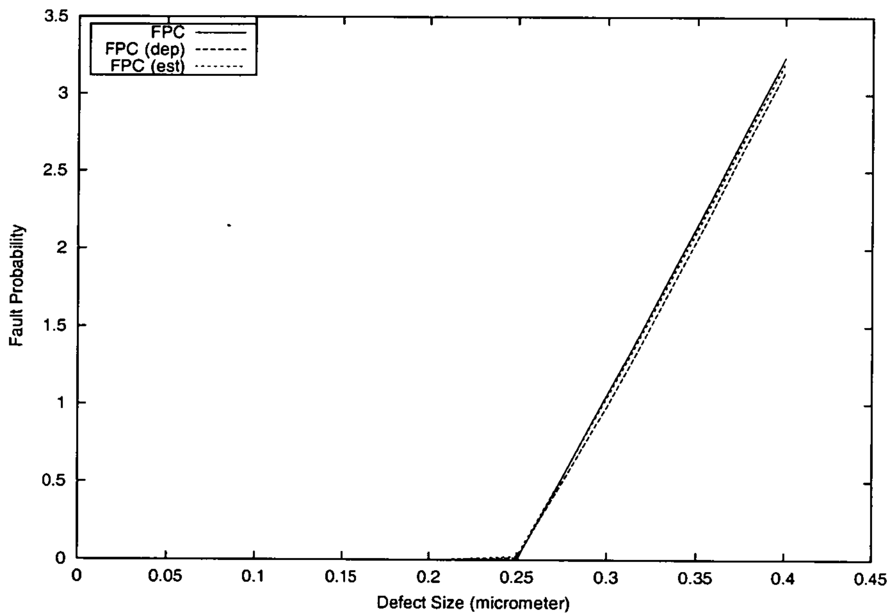
Defect Size	Fault Probability Curve	Fault Probability Curve (dep)	Fault Probability Curve (est)
0	0	0	0
0.05	0	0	0
0.10	0	0	0
0.15	0	0	0
0.20	0	0	0
0.25	0	0.82112	1.40928
0.30	39.25926	36.44111	38.56037
0.35	49.91253	47.89574	49.20280
0.40	50.625	49.23156	50.08625

(b)

Figure 5.22.: An example extract from the two result files generated for two parallel lines. (a) CAR file. (b) FPR file.



(a)



(b)

Figure 5.23.: Example results from Calma-plot. (a) Comparison of critical area curves. (b) Comparison of FP curves.

It can be observed from figure 5.23 (a) that the estimated (using CALMAT) critical area plot between defect sizes of 0.2 μm and 0.4 μm results in smaller percentage errors than using the mask layout. One consequence is that the estimated (using CALMAT) fault probability curve is located between the fault probability curves obtained from the mask layout and the layout after pattern transfer as shown in figure 5.23 (b). Thus it can be deduced that the estimated fault probability curve provides a more accurate result than using the fault probability curve of the mask layout.

5.4. Summary

This chapter discussed the various methods employed to categorise a layout. There were three initial attempts at categorising the mask layout. The first method was based on the physical characteristics of a layout while the other two methods utilised track layout information. The method that was finally employed was based upon utilising track layout information. Calma was designed with the idea that a layout could be represented as a bitmap consisting of "bits". By observing the neighbouring bits on a layout, sufficient information could be obtained regarding a particular bit and its effect on the layout. It should also be noted that Calma does not necessarily have to be limited to parallel tracks although these were used as an initial test of the effectiveness of Calma. The following chapter will present the results obtained from using Calma on various mask layouts consisting of parallel tracks.

6. Calculation of Critical Area Using CALMA

6.1. Introduction

The previous chapter presented a method for estimating critical area curves of a layout after pattern transfer. This chapter describes the usage of Calma to calculate CA on a variety of mask layouts. It consists of two main sections with the first describing the generation of the database used by Calma, while the second presents and discusses its performance.

The accuracy of the CA generated using Calma is affected by the Region Size (RS) setting which defines the bounding area used in categorising the layout region. Larger values of RS increases the size of the database storing greater detail which enables a more accurate estimation of CA to be made.

In order to evaluate the performance of Calma two different sets of layouts based on variations of parallel tracks have been used. The first set is similar to the arrangement of tracks employed to generate the database and is used to benchmark the accuracy of Calma. The second set is based on actual routing tracks extracted from an IC layout and are used to determine the accuracy that can be expected from Calma on typical layouts.

6.2. Database Generation

Before Calma can be used to estimate CA, a database must first be created for the required region size. Thus to determine the accuracy of Calma on parallel routing tracks, a database which categorises the parallel tracks must be generated. To do this three different sets consisting of 2, 3 or 4 parallel tracks have been used. In all the sets, the tracks have a width and separation of 0.5 μm and the resulting database enables the calculation of critical area for layouts with equal track width and spacing. Figure 6.1, figure 6.2 and figure 6.3 illustrates the combinations which have been used to generate the database for the 3 sets of tracks.

The first step in the database generation process is to categorise each layout. As the three sets of tracks all have uniform width and separation, a unit box of 0.5 μm $\times$ 1.0 μm (horizontal length $\times$ vertical length) has been used. Each mask layout is effectively overlayed with a rectangular grid consisting of unit boxes and these rectangles then define an address which categorises the layout based on the RS used.

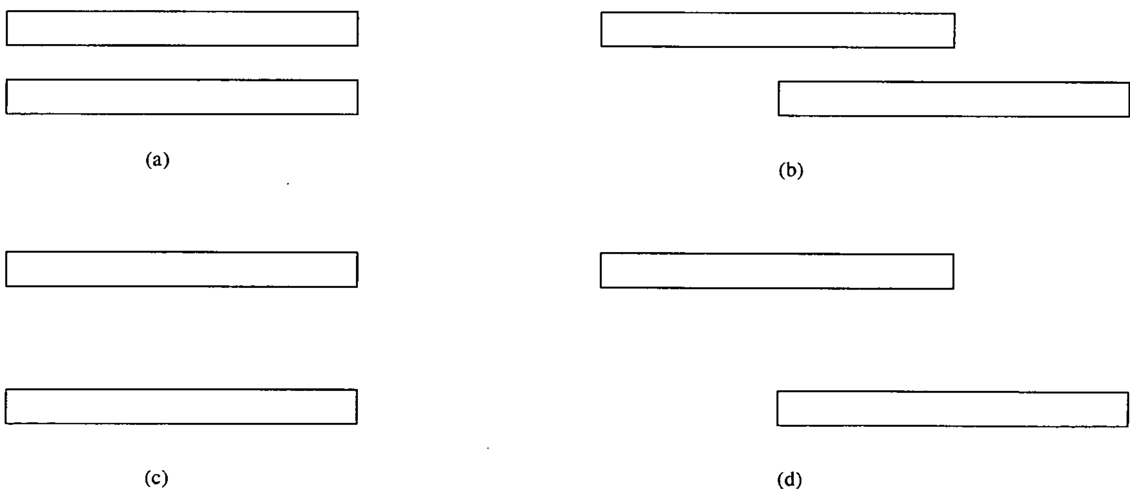


Figure 6.1.: Different track arrangements used in database generation for set 1 (2 parallel tracks). (a) Type 2-1. (b) Type 2-2. (c) Type 2-3. (d) Type 2-4.

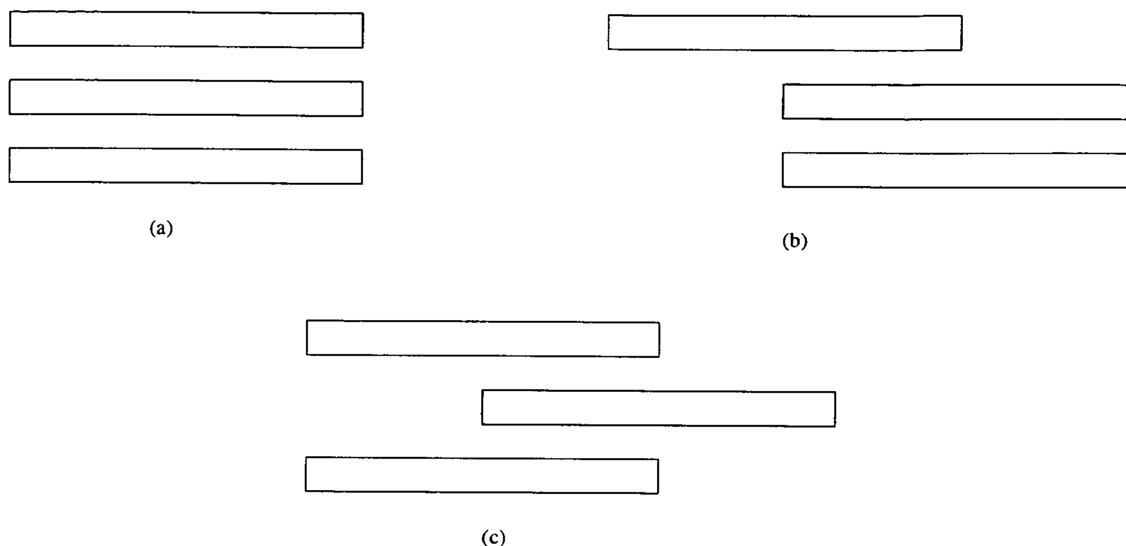


Figure 6.2.: Different track arrangements used in database generation for set 2 (3 parallel tracks). (a) Type 3-1. (b) Type 3-2. (c) Type 3-3.

Depict and Calma-runeye are then employed to obtain the critical area after pattern transfer for each of these addresses which is then entered into the database. The layout information with the bounding region of an address is stored in a CIF file. The bounding region limits Calma-runeye to only calculate the critical area within this area after Depict has determined the layout after pattern transfer. The process of using Depict and Calma-runeye is automated but unfortunately manual intervention is required to modify the CIF file to specify the bounding region for each address. For example, the bounding region of an address in the type 2-1 track for a region size of 2 is shown in figure 6.4 while the modifications made to the CIF file is illustrated in figure 6.5.

The CIF file is used to describe integrated circuits by providing a limited set of parameters that are employed to describe two dimensional shapes on different layers of a chip. In figure 6.5, the two parallel tracks are placed on the "metal" layer with the layer (L) parameter and described using the box (B) parameter. The modifications to the CIF file adds an extra "box" layer and is described by the box parameter line which specifies the bounding region for the address.

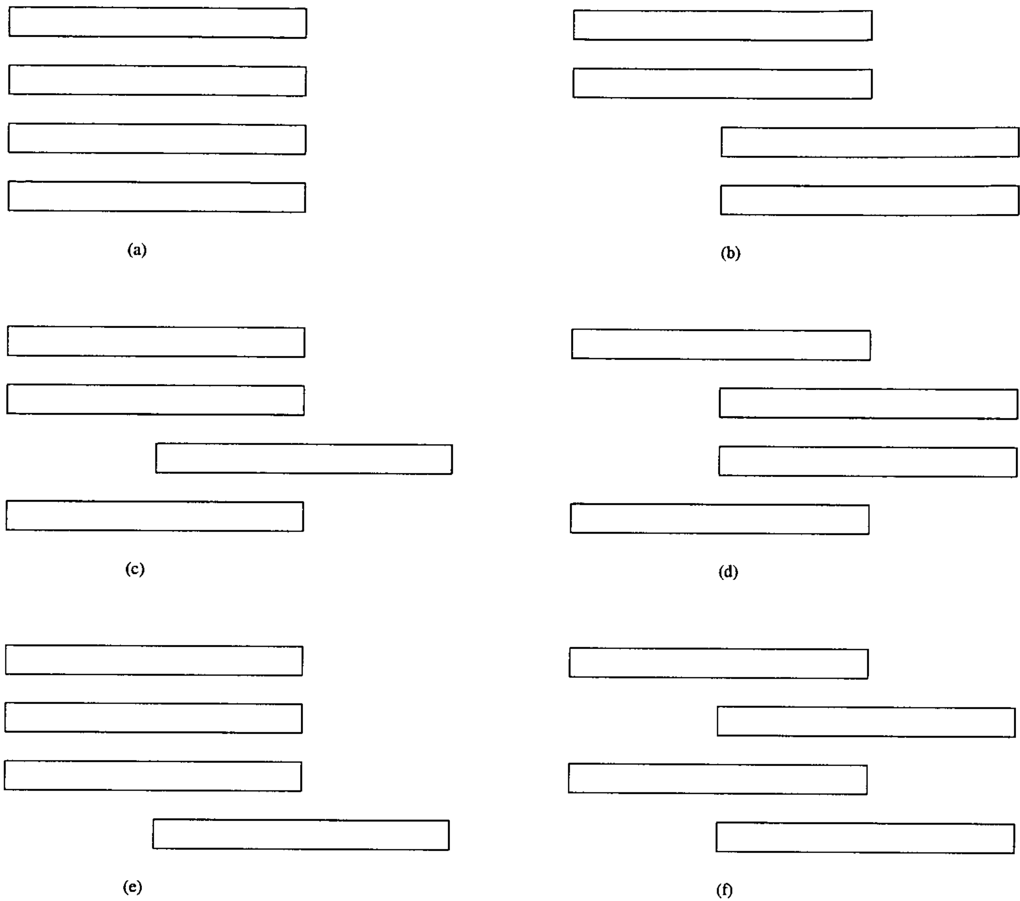


Figure 6.3.: Different track arrangements used in database generation for set 3 (4 parallel tracks). (a) Type 4-1. (b) Type 4-2. (c) Type 4-3. (d) Type 4-4. (e) Type 4-5. (f) Type 4-6.

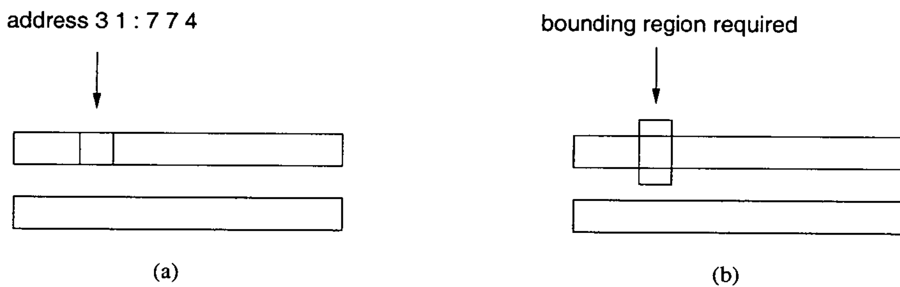


Figure 6.4.: Determining the bounding region for the address 31:774 in the type 2-1 track. (a) Location of address on the layout. (b) Bounding region required for the address.

```
(CIF file written by mktrack);
DS 1 1 1;               ← start of subroutine and scaling factor
9 cell;                 ← user specified information
L metal;                ← metal layer
B 1000 50 500,25;       ← bottom track
B 1000 50 500,125;      ← top track
L box;                  ← box layer (modifications made)
B 50 100 125,125 ;      ← box region (modifications made)
DF;                     ← end of subroutine
E                       ← end of file
```

Figure 6.5.: Modifications made to specify the bounding region for address 31:774.

The length of the bounding region in this example is a unit length as it corresponds to one address location. This process is then repeated for each address and the results used for building the database. For a RS of 1 there are 2^8 (256) potential address types and for RS of 2 there are $2^8 \times 2^{16}$ address types. This is a large number of addresses, but fortunately this procedure is only carried out once when generating the database. In addition, not all the addresses will be valid and depending on the layout symmetry, two or more addresses will generate similar results which effectively cuts the database size by a factor of two or more. The database can be further reduced in size by targetting specific types of tracks. For the database generated in this chapter, its size has been significantly reduced by targetting parallel tracks only. A RS of 1 required 41 address calculations while a RS of 2 required 252 address calculations.

In addition, two different sets of databases have been generated for each RS. The first set (unmodified database) was generated as outlined above. The second set (modified database) consisted of changing the bounding region for an address. For some addresses it is possible to expand the bounding region as the same address is located along side. Thus by calculating the critical area obtained over the increased bounding region for a given address an averaged value can be used in the database. As an example, figure

6.6 illustrates how the address of 31:774 for track type 2-1 can be determined by using a bounding region of 6 unit lengths and figure 6.7 shows the changes necessary to the CIF file. The only difference between this CIF file and the one in figure 6.5 is that the size of the bounding box has been increased.

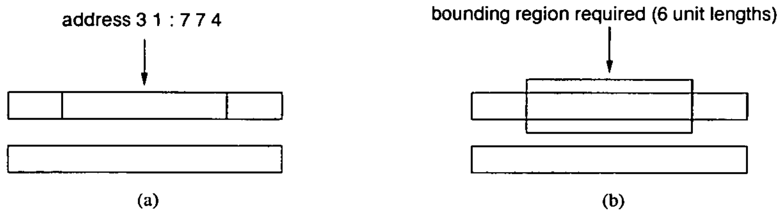


Figure 6.6.: Enlarging the bounding region for the address 31:774 in the type2-1 track using (a) Location of similar addresses on the layout. (b) Bounding region required for the addresses.

Employing Calma to obtain the estimated critical area curves after pattern transfer for the three set of tracks is a two step process. Firstly a database of addresses has to be generated and secondly the mask layout has to be processed by the Calma program. The following section discuss the results obtained from using tracks similar to the orginal database tracks and routing track examples.

```
(CIF file written by mktrack);
DS 1 1 1;              ← start of subroutine and scaling factor
9 cell;                ← user specified information
L metal;               ← metal layer
B 1000 50 500,25;      ← bottom track
B 1000 50 500,125;     ← top track
L box;                 ← box layer (modifications made)
B 300 100 250,125 ;    ← box region (modifications made)
DF;                    ← end of subroutine
E                      ← end of file
```

Figure 6.7.: Modifications necessary for the enlarged bounding region for address 31:774 (6 unit length).

6.3. Results

This section presents the results obtained from using Calma on two different groups of tracks. The first is based closely on the track types used to generate the symmetry database, the difference being that different track lengths are used instead of the ones used to generate the database. The second group type is based on actual routing tracks observed on an AMD 2901 Alliance chip (a small IC) and these are used to evaluate the performance of Calma. In order to concentrate on highlighting trends in the following sections, a selected number of layouts are used. The full results for all the layouts are presented in appendix D.

6.3.1. Region Size of 1

This section illustrates the limitation of using a RS setting of 1 in estimating the critical area of a layout after pattern transfer and the following results have been generated from track type 2-1. Figure 6.8 (a) shows the relationship between the CA calculated by Calma and the CA obtained from the aerial image calculated using Depict (CA error (%)) as a function of defect size. There are three curves plotted where the first two (region size 1 unmod and region size 1 mod) are obtained from using a RS of 1 for the unmodified and modified databases respectively (the modified database is one where the critical area of some of the addresses have been averaged). The remaining curve (mask layout) is the CA error (%) obtained from the mask layout.

Figure 6.8 (b) shows the fault probability (FP) curves obtained from multiplying the CA curves by the defect distribution which is taken as $\frac{1}{x^3}$. This graph consists of four curves where the first three are directly calculated from the curves presented in figure 6.8 (a) with the last (aerial image) being obtained from the CA of the aerial image calculated using Depict.

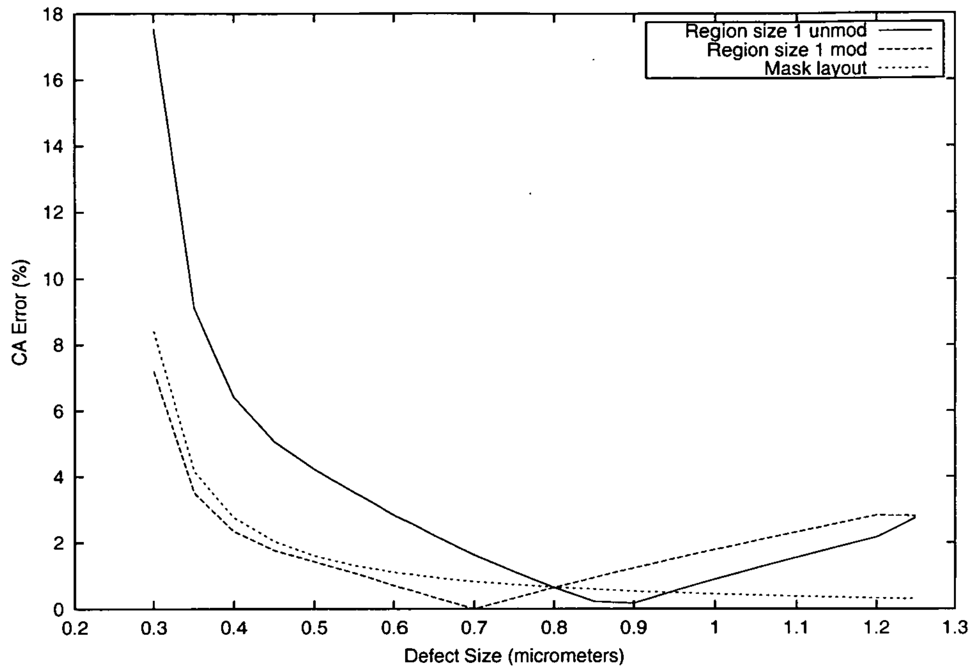
The integral of the FP curve between two defect sizes x_1 and x_2 is equal to the aver-

age number of faults within the defect range and is given by $\lambda_{x_1-x_2}$. The maximum Theoretical Defect Size (TDS) is defined as the maximum defect size that the results obtained from Calma are valid and is the region size value multiplied by the unit length. For example, using a region size of 2 on the data generated (unit length of 0.5 μm) in the previous section would give a TDS of 1 μm.

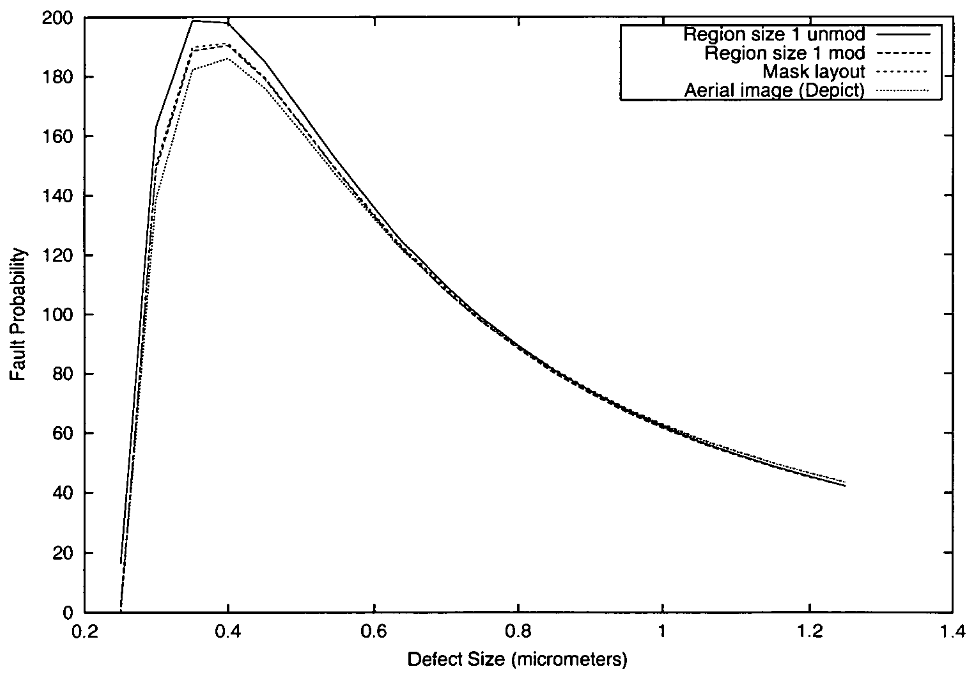
Table 6.1 presents the average number of faults for the defect ranges and associated errors compared with the value obtained from the aerial image. The first column presents the average number of faults up to 0.5 μm (TDS 1), while the second is up to 1.0 μm (TDS 2). The remaining two columns detail the associated error compared with the calculated value obtained from the aerial image.

From figure 6.8 it can be observed that the unmodified region size 1 result gives consistently higher errors than the mask layout result for defect sizes between 0.3 and 0.8 μm whereas the modified region size 1 result has lower errors for these defect sizes. As a consequence the modified region 1 result give a better estimate when using a RS of 1. This is reflected in the average number of faults where the error for the modified region 1 result is less than that obtained from the mask layout. It should be remembered that using a RS of 1 the method is only valid for defect size up to 0.5 μm (TDS 1) and for sizes larger than this the accuracy can not be guranteed. Figure 6.9 presents another set of results obtained from using track type 3-1 with table 6.2 detailing the average number of faults and associated errors for the defect ranges.

From figure 6.9 it can be observed that both the unmodified and modified region size 1 results gives consistently higher errors than the mask layout curves for defect sizes between 0.3 and 1.25 μm. The reason is that with three parallel tracks, some of the addresses would be represented as if they were from a two parallel track address due to the limitation of using a RS of 1 to categorise the layout. Thus the region size 1 results can not be used to estimate the critical areas of three or more parallel tracks. Here the limitation of using a RS of 1 is apparent and its application should be confined to only two parallel tracks.

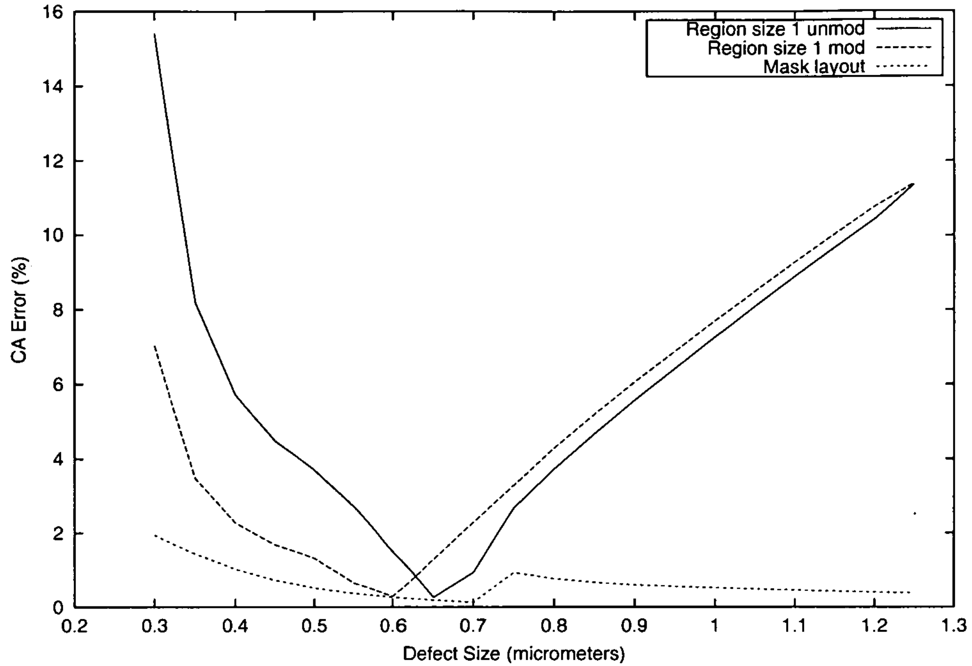


(a)

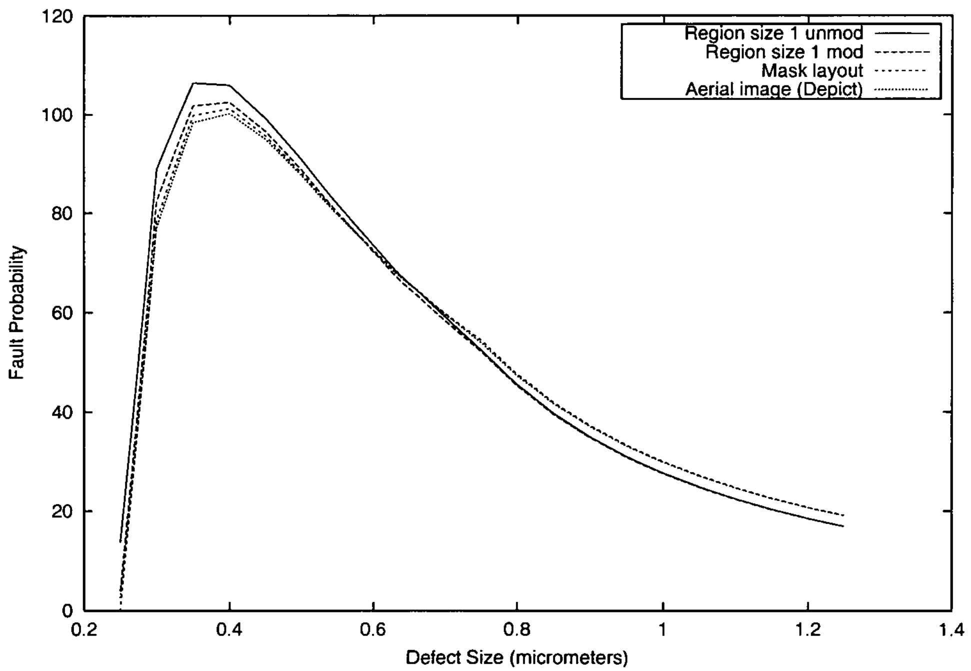


(b)

Figure 6.8.: Relationships for track type 2-1 as a function of defect size. (a) CA error. (b) Fault probability.



(a)



(b)

Figure 6.9.: Relationships for track type 3-1 as a function of defect size. (a) CA error. (b) Fault probability.

Model	Average number of faults, $\lambda_{0-0.5}$ (defect size $\leq$ 0.5 μm) ($\times10^{-8}$)	Average number of faults, $\lambda_{0-1.0}$ (defect size $\leq$ 1.0 μm) ($\times10^{-8}$)	Error in $\lambda_{0-0.5}$ compared with Depict (%)	Error in $\lambda_{0-1.0}$ compared with Depict (%)
Region size 1 unmod	38.05	92.94	11.3	5.6
Region size 1 mod	35.47	89.37	3.7	1.5
Mask layout	35.55	89.89	4.0	2.1
Aerial image (Depict)	34.19	88.01	-	-

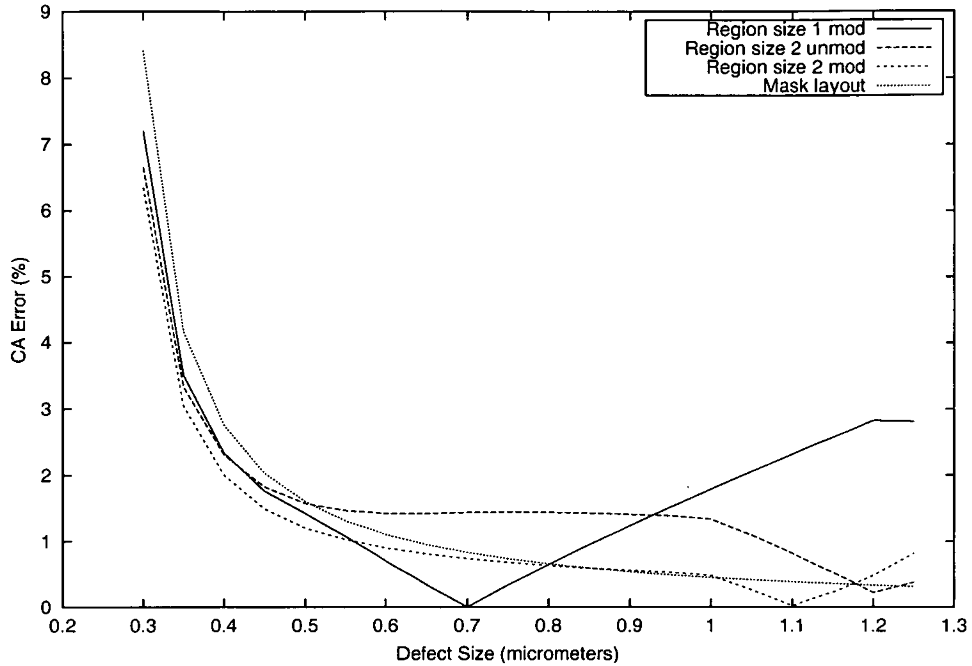
Table 6.1.: Average number of faults and associated errors for track type 2-1.

Model	Average number of faults, $\lambda_{0-0.5}$ (defect size $\leq$ 0.5 μm) ($\times10^{-8}$)	Average number of faults, $\lambda_{0-1.0}$ (defect size $\leq$ 1.0 μm) ($\times10^{-8}$)	Error in $\lambda_{0-0.5}$ compared with Depict (%)	Error in $\lambda_{0-1.0}$ compared with Depict (%)
Region size 1 unmod	20.71	49.46	11.0	4.0
Region size 1 mod	19.37	47.71	3.8	0.3
Mask layout	18.76	47.81	0.5	0.5
Aerial image (Depict)	18.66	47.57	-	-

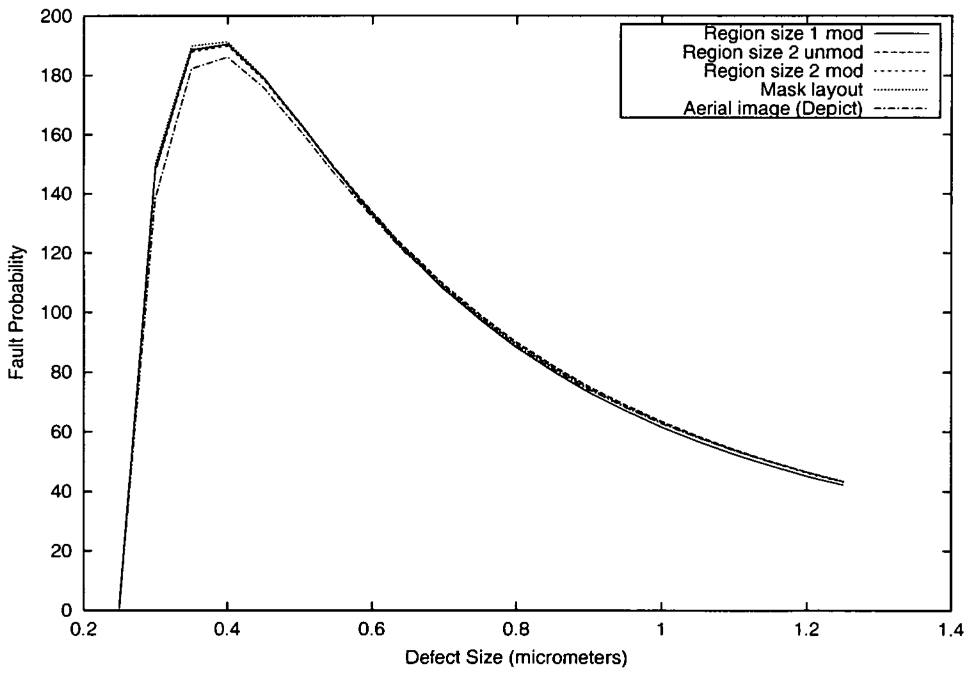
Table 6.2.: Average number of faults and associated errors for track type 3-1.

6.3.2. Region Size of 2

This section presents the use of an increased TDS of 1.0 μm which results from using a RS of 2. As with the previous section, two graphs and a table are used to present the results for track type 2-1. The difference seen in figure 6.10 is that there are two curves (region size 2 unmod and region size 2 mod) plotted using a RS of 2 for the unmodified and modified databases that were not present in the previous section. In addition, table 6.3 presents average number of faults and associated errors for the defect ranges. The purpose of using the "Region 1 mod" result is to compare the accuracy of using the two different region size values.



(a)



(b)

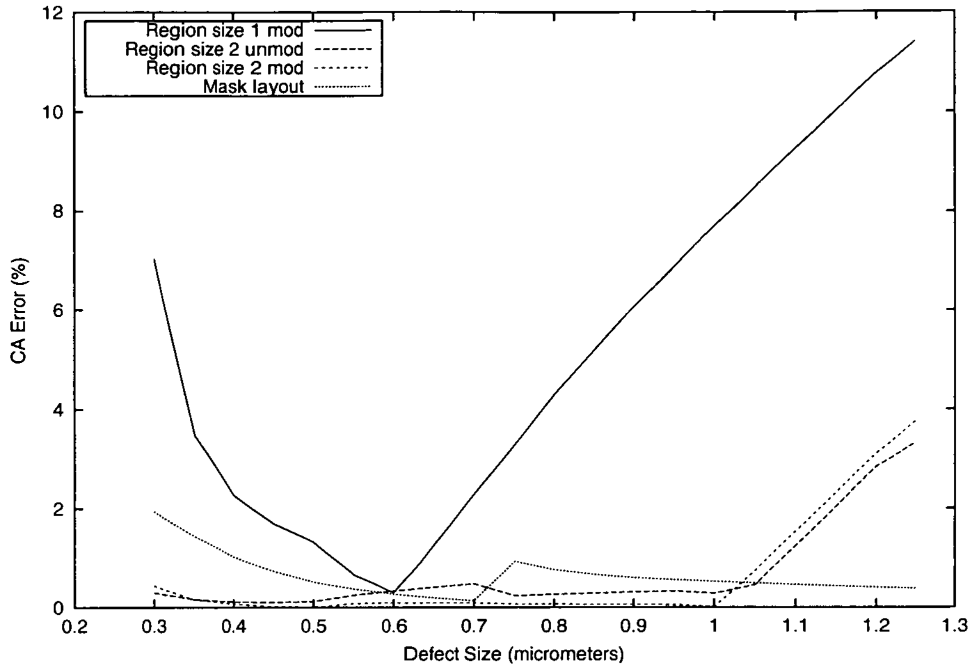
Figure 6.10.: Relationships for track type 2-1 as a function of defect size. (a) CA error. (b) Fault probability.

Model	Average number of faults, $\lambda_{0-0.5}$ (defect size $\leq$ 0.5 μm) ($\times 10^{-8}$)	Average number of faults, $\lambda_{0-1.0}$ (defect size $\leq$ 1.0 μm) ($\times 10^{-8}$)	Error in $\lambda_{0-0.5}$ compared with Depict (%)	Error in $\lambda_{0-1.0}$ compared with Depict (%)
Region size 1 mod	35.47	89.37	3.7	1.5
Region size 2 unmod	35.36	89.96	3.4	2.2
Region size 2 mod	35.26	89.52	3.2	1.7
Mask layout	35.55	89.89	4.0	2.1
Aerial image (Depict)	34.19	88.01	-	-

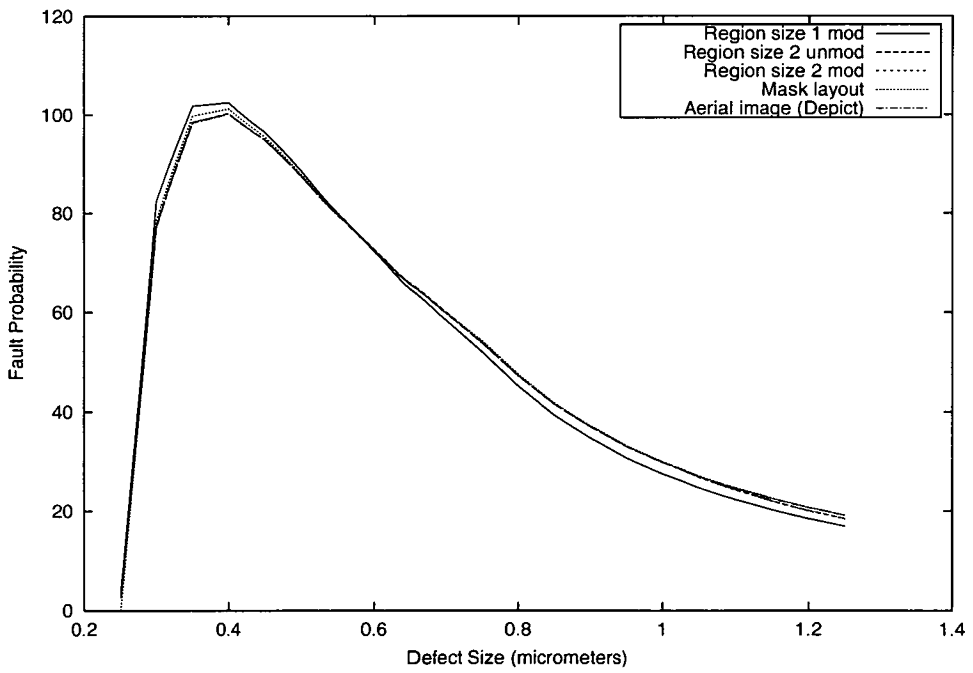
Table 6.3.: Average number of defects and associated errors for track type 2-1.

From figure 6.10 it can be observed that there are three regions of interest, from 0.3 - 0.5 μm, 0.5 - 0.8 μm and > 0.8 μm. For the first region, all three results give consistently lower errors than the mask layout. This implies that all three can be used to estimate the critical area in this defect range. In the second region, the modified region size 1 and 2 results give lower error values than the mask layout while the unmodified region size 2 result gives higher error values. One possibility for this is that the critical area values for certain addresses used is overestimating the critical area of the layout while using an average for these values produce more accurate results. In the final region, the modified region size 2 result has a smaller error value than the mask layout until 0.9 μm, after which there is a very small difference between the two values. Thus it can be observed that the modified region size 2 result gives better estimations than any of the other results as shown in table 6.3. Another set of curves obtained from using track type 3-1 is shown in figure 6.11 while table 6.4 presents the average number of faults and associated errors for the defect ranges.

As expected it can be seen that the modified region size 1 generally has higher errors than the other results only intersecting with the unmodified region size 2 curve at around 0.6 μm. The modified region size 2 result consistently has a lower error value than the mask layout for defect sizes between 0.3 and 1.0 μm.



(a)



(b)

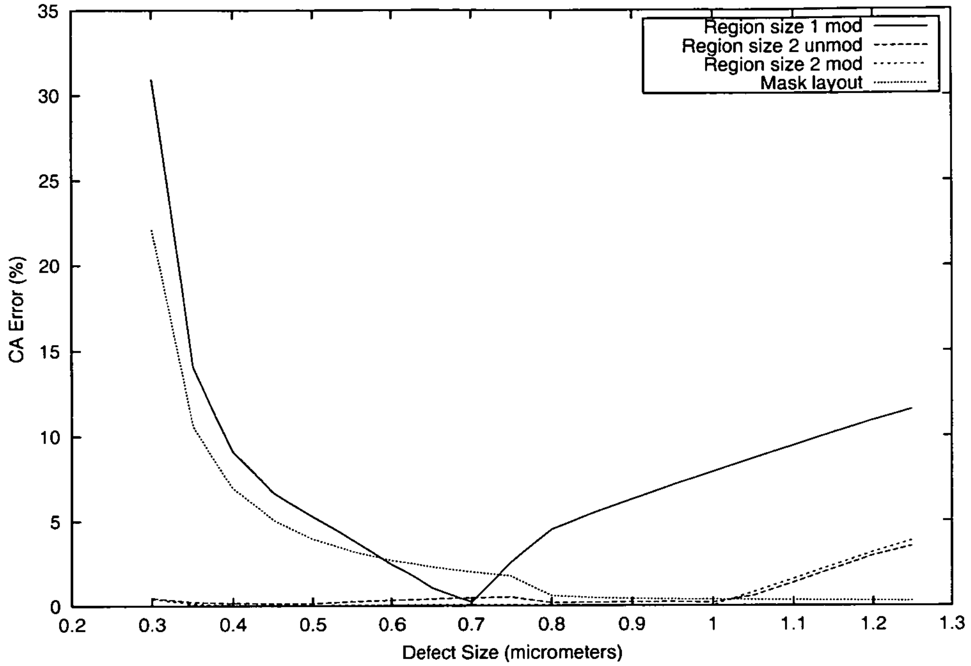
Figure 6.11.: Relationships for track type 3-1 as a function of defect size. (a) CA error. (b) Fault probability.

Model	Average number of faults, $\lambda_{0-0.5}$ (defect size $\leq$ 0.5 μm) ($\times 10^{-8}$)	Average number of faults, $\lambda_{0-1.0}$ (defect size $\leq$ 1.0 μm) ($\times 10^{-8}$)	Error in $\lambda_{0-0.5}$ compared with Depict (%)	Error in $\lambda_{0-1.0}$ compared with Depict (%)
Region size 1 mod	19.37	47.71	3.8	0.3
Region size 2 unmod	18.70	47.69	0.2	0.3
Region size 2 mod	18.70	47.63	0.2	0.1
Mask layout	18.76	47.81	0.5	0.5
Aerial image (Depict)	18.66	47.57	-	-

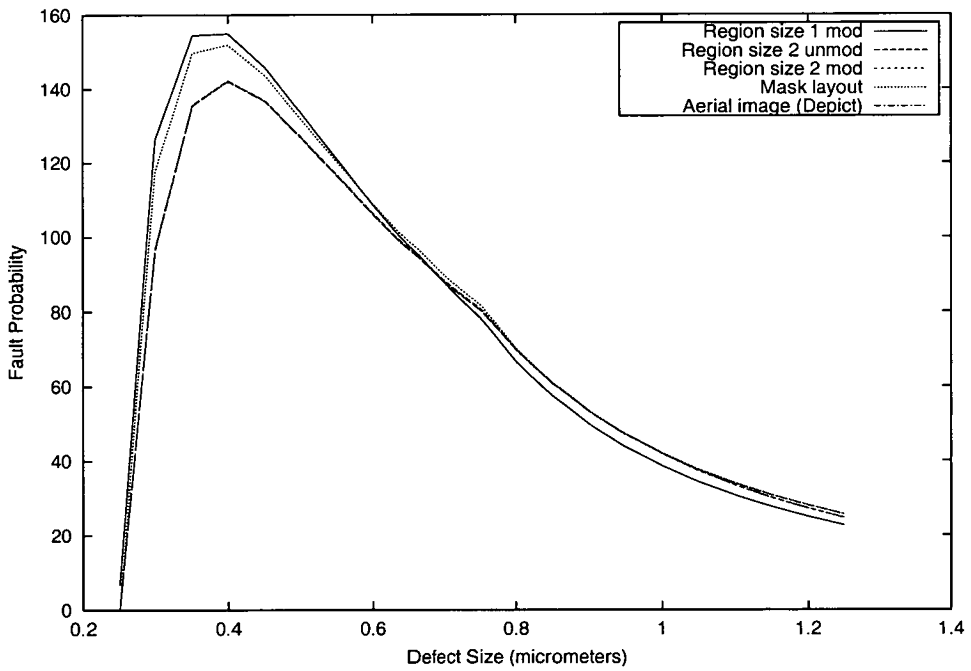
Table 6.4.: Average number of faults and associated errors for track type 3-1.

Between defect sizes of 0.6 and 0.7 μm, the unmodified region size 2 result has an larger error than the mask layout after which the error becomes smaller. This again can be attributed to not using averaged values for certain addresses. In addition, no simplifications of the tracks are made in categorising the layout as with the type 3-1 track in the previous section using a RS of 1. Therefore accurate estimates can be obtained from using the modified region size 2 result and this is shown in table 6.4. A final set of results obtained from using track type 4-1 is shown in figure 6.12 while table 6.5 presents the average number of faults and associated errors for the defect ranges.

Concentrating on the region size 2 results, it can be seen that both the unmodified and modified region size 2 results have lower error values than the mask layout for defect sizes between 0.3 and 1.0 μm. In this four parallel track layout, some of the addresses would be represented as if it were from a three parallel track address due to the limitation of using a RS of 2 to categorise the layout and is similar to the RS 1 case in the previous section. However, the limitation of using a RS of 2 going from 3 to 4 parallel tracks is not a considerable factor compared with using a RS of 1 going from 2 to 3 parallel tracks. Therefore using the modified region size 2 result to estimate the critical area generates an accurate value as shown in table 6.5.



(a)



(b)

Figure 6.12.: Relationships for track type 4-1 as a function of defect size. (a) CA error. (b) Fault probability.

Model	Average number of faults, $\lambda_{O-0.5}$ (defect size $\leq$ 0.5 μm) ($\times 10^{-8}$)	Average number of faults, $\lambda_{O-1.0}$ (defect size $\leq$ 1.0 μm) ($\times 10^{-8}$)	Error in $\lambda_{O-0.5}$ compared with Depict (%)	Error in $\lambda_{O-1.0}$ compared with Depict (%)
Region size 1 mod	29.40	71.63	15.2	5.7
Region size 2 unmod	25.58	67.96	0.2	0.3
Region size 2 mod	25.55	67.82	0.1	0.07
Mask layout	28.15	71.30	10.3	5.2
Aerial image (Depict)	25.52	67.77	-	-

Table 6.5.: Average number of faults and associated errors for track type 4-1.

6.3.3. Routing Track Examples

In the previous sections it was observed that accurate estimations of the critical area could be obtained from tracks that were very similar to those used to generate the original database. However, tracks on an actual IC are typically more complex than those presented in the previous section and so an AMD 2901 chip was used to test Calma on track layouts that were not used to generate the original database. Six different sample layouts were chosen and are illustrated in figure 6.13.

As the modified region size 2 result provided the highest accuracy among the different region size settings in the previous sections, this curve in conjunction with the mask layout curve were used to generate the CA and fault probabilty curves for the six layouts. The results obtained from running Calma on the six sample layouts are shown in figures 6.14 to 6.19 while tables 6.6 to 6.11 present the average number of faults and associated errors for the defect ranges.

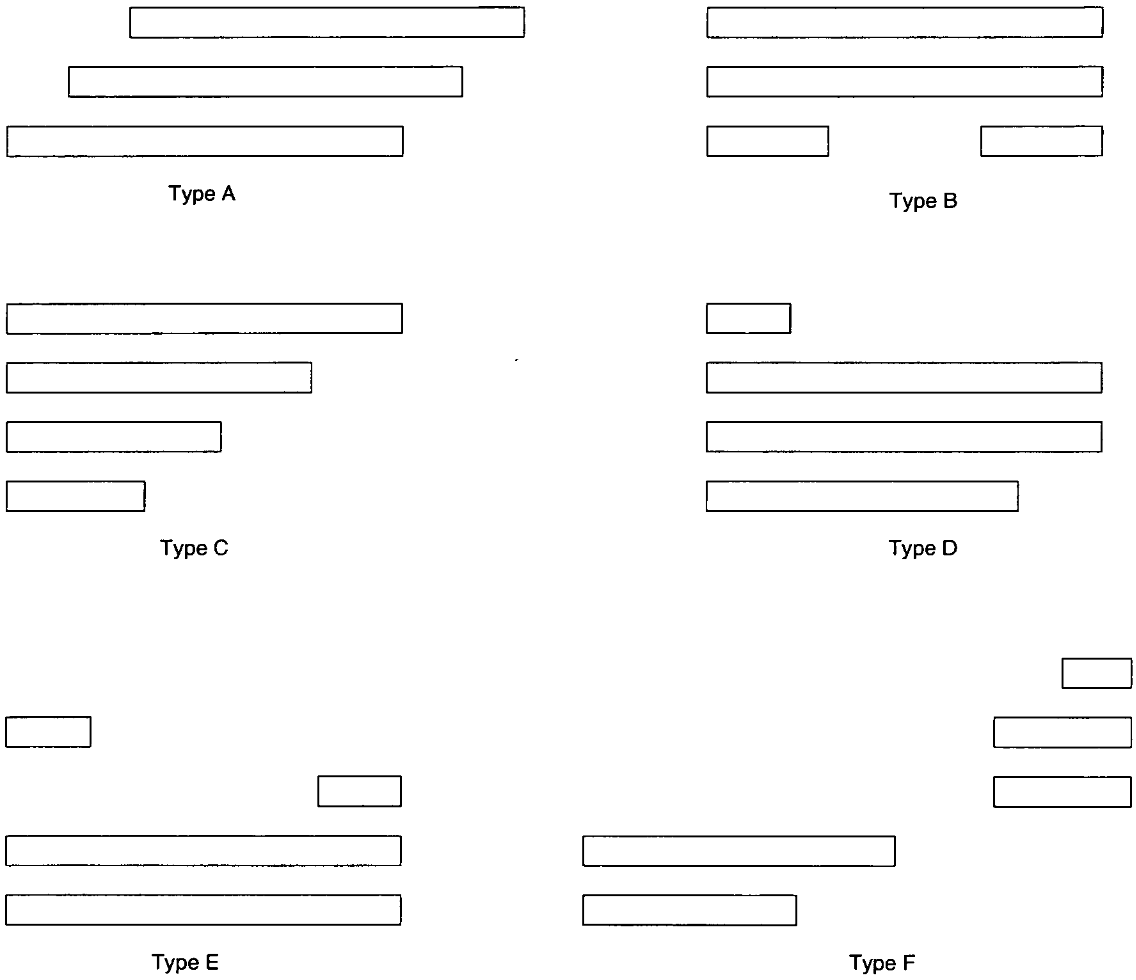
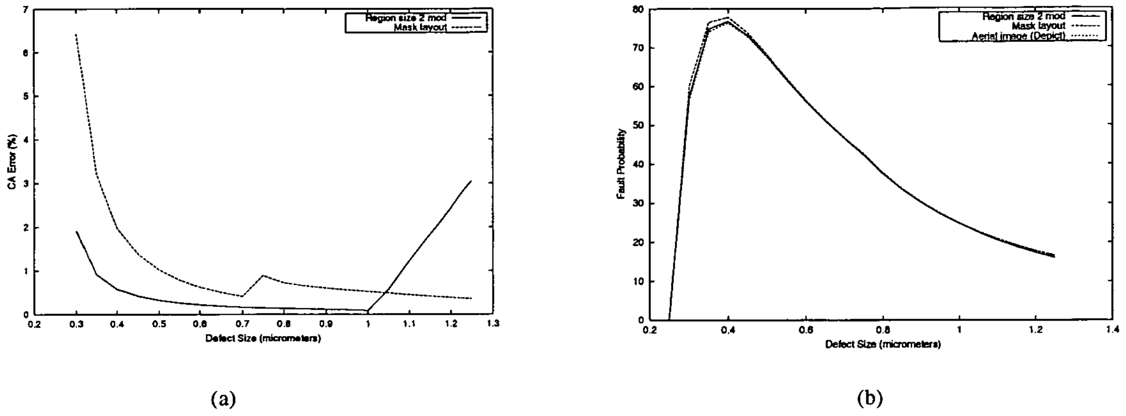


Figure 6.13.: Six different track layouts extracted from an AMD 2901 chip used to evaluate Calma.

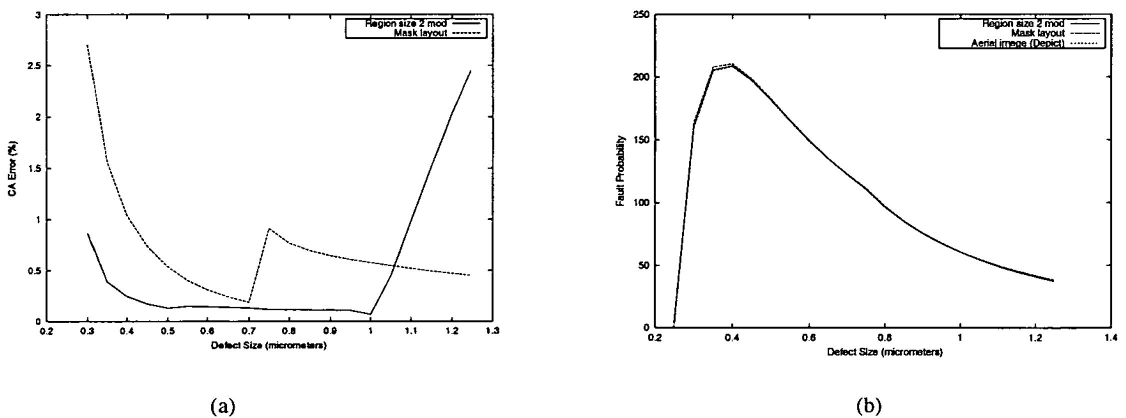


(a) (b)

Figure 6.14.: Relationships for track type A as a function of defect size. (a) CA error. (b) Fault probability.

Model	Average number of faults, $\lambda_{0-0.5}$ (defect size $\leq$ 0.5 μm) ($\times10^{-8}$)	Average number of faults, $\lambda_{0-1.0}$ (defect size $\leq$ 1.0 μm) ($\times10^{-8}$)	Error in $\lambda_{0-0.5}$ compared with Depict (%)	Error in $\lambda_{0-1.0}$ compared with Depict (%)
Region size 2 mod	14.13	36.85	0.9	0.5
Mask layout	14.40	37.23	2.8	1.5
Aerial image (Depict)	14.01	36.67	-	-

Table 6.6.: Average number of faults and associated errors for track type A.

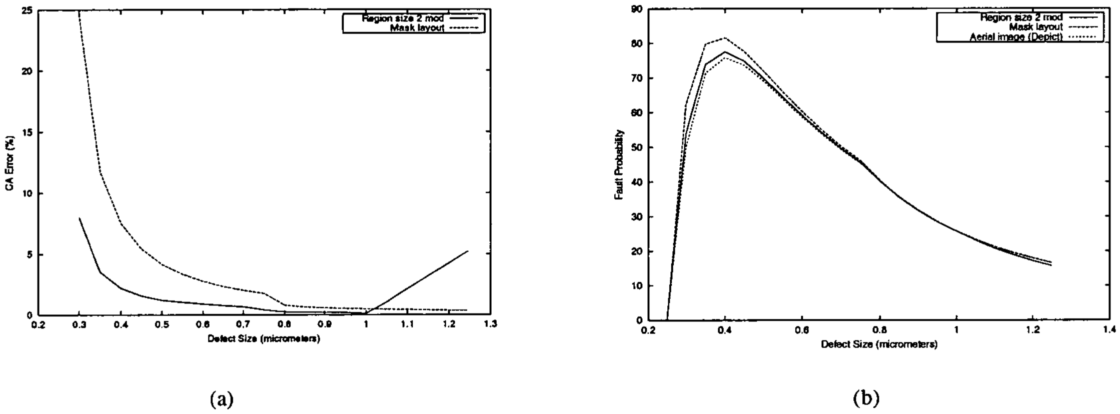


(a) (b)

Figure 6.15.: Relationships for track type B as a function of defect size. (a) CA error. (b) Fault probability.

Model	Average number of faults, $\lambda_{0-0.5}$ (defect size $\leq$ 0.5 μm) ($\times10^{-8}$)	Average number of faults, $\lambda_{0-1.0}$ (defect size $\leq$ 1.0 μm) ($\times10^{-8}$)	Error in $\lambda_{0-0.5}$ compared with Depict (%)	Error in $\lambda_{0-1.0}$ compared with Depict (%)
Region size 2 mod	38.81	98.17	0.4	0.2
Mask layout	39.05	98.62	1.0	0.7
Aerial image (Depict)	38.67	97.94	-	-

Table 6.7.: Average number of faults and associated errors for track type B.



(a) (b)

Figure 6.16.: Relationships for track type C as a function of defect size. (a) CA error. (b) Fault probability.

Model	Average number of faults, $\lambda_{0-0.5}$ (defect size $\leq$ 0.5 μm) ($\times10^{-8}$)	Average number of faults, $\lambda_{0-1.0}$ (defect size $\leq$ 1.0 μm) ($\times10^{-8}$)	Error in $\lambda_{0-0.5}$ compared with Depict (%)	Error in $\lambda_{0-1.0}$ compared with Depict (%)
Region size 2 mod	14.03	37.98	3.6	1.7
Mask layout	15.06	39.38	11.2	5.5
Aerial image (Depict)	13.54	37.34	-	-

Table 6.8.: Average number of faults and associated errors for track type C.

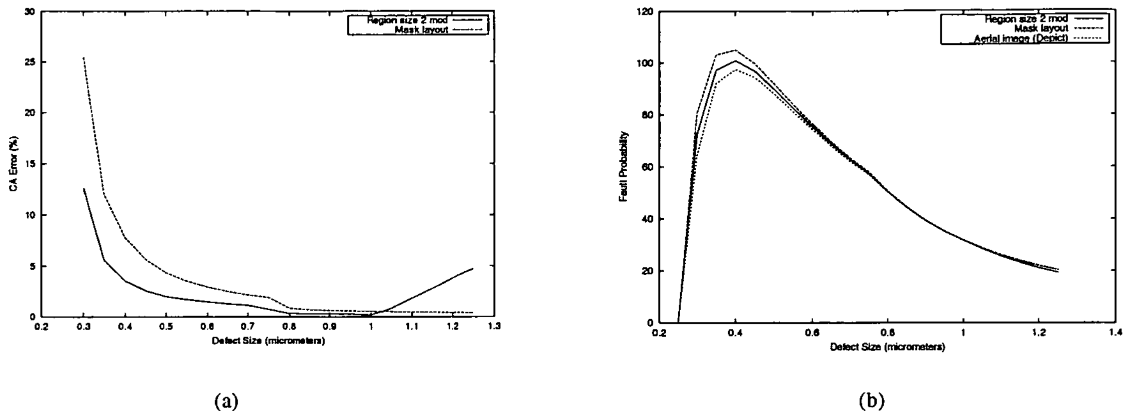


Figure 6.17.: Relationships for track type D as a function of defect size. (a) CA error. (b) Fault probability.

Model	Average number of faults, $\lambda_{0-0.5}$ (defect size $\leq$ 0.5 μm) ($\times 10^{-8}$)	Average number of faults, $\lambda_{0-1.0}$ (defect size $\leq$ 1.0 μm) ($\times 10^{-8}$)	Error in $\lambda_{0-0.5}$ compared with Depict (%)	Error in $\lambda_{0-1.0}$ compared with Depict (%)
Region size 2 mod	18.42	48.76	5.8	2.8
Mask layout	19.42	50.14	11.5	5.8
Aerial image (Depict)	17.41	47.41	-	-

Table 6.9.: Average number of faults and associated errors for track type D.

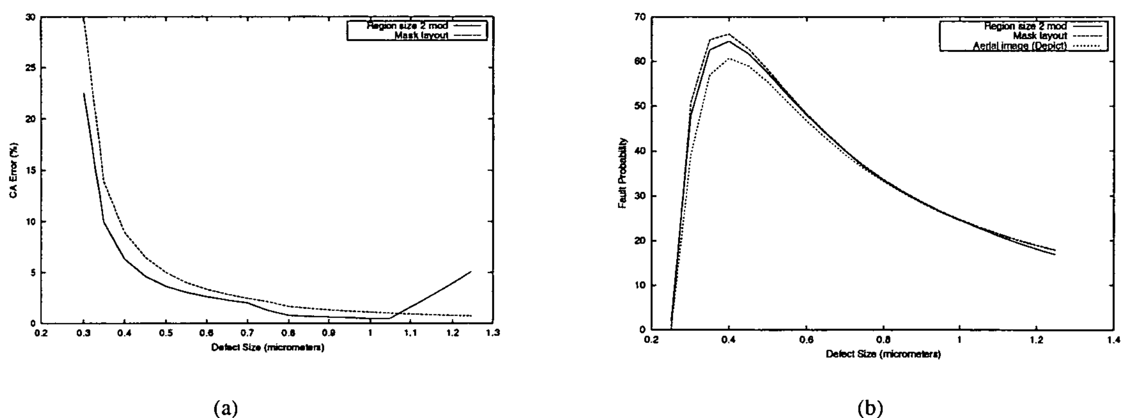


Figure 6.18.: Relationships for track type E as a function of defect size. (a) CA error. (b) Fault probability.

Model	Average number of faults, $\lambda_{0-0.5}$ (defect size $\leq$ 0.5 μm) ($\times 10^{-8}$)	Average number of faults, $\lambda_{0-1.0}$ (defect size $\leq$ 1.0 μm) ($\times 10^{-8}$)	Error in $\lambda_{0-0.5}$ compared with Depict (%)	Error in $\lambda_{0-1.0}$ compared with Depict (%)
Region size 2 mod	11.93	31.79	10.7	5.1
Mask layout	12.22	32.24	13.4	6.6
Aerial image (Depict)	10.78	30.24	-	-

Table 6.10.: Average number of faults and associated errors for track type E.

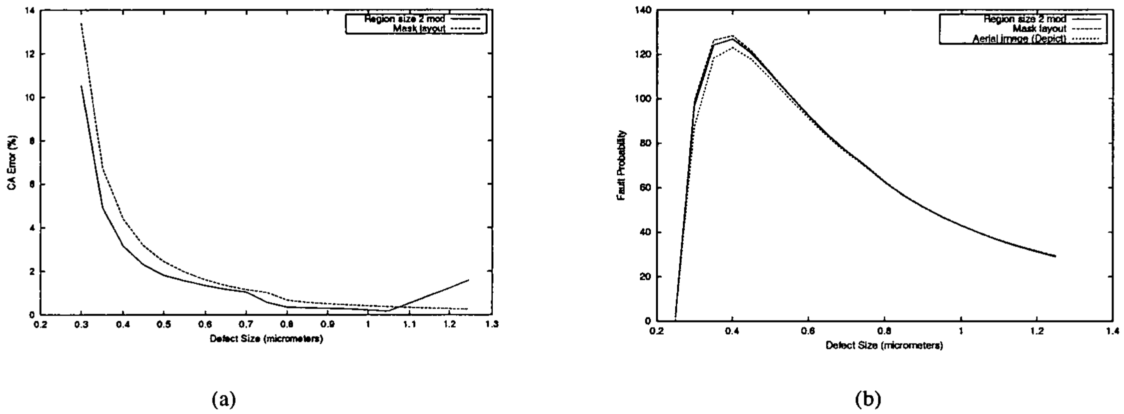


Figure 6.19.: Relationships for track type F as a function of defect size. (a) CA error. (b) Fault probability.

Model	Average number of faults, $\lambda_{0-0.5}$ (defect size $\leq$ 0.5 μm) ($\times 10^{-8}$)	Average number of faults, $\lambda_{0-1.0}$ (defect size $\leq$ 1.0 μm) ($\times 10^{-8}$)	Error in $\lambda_{0-0.5}$ compared with Depict (%)	Error in $\lambda_{0-1.0}$ compared with Depict (%)
Region size 2 mod	23.58	61.22	5.1	2.6
Mask layout	23.79	61.54	6.1	3.1
Aerial image (Depict)	22.43	59.68	-	-

Table 6.11.: Average number of faults and associated errors for track type F.

In all six cases the modified region size 2 result had smaller errors than the mask layout for defect sizes between 0.3 and 1.0 μm. It also provided better estimations for the average number of faults up to TDS 1 (0.5 μm) and TDS 2 (1.0 μm) as shown in tables

6.6 to 6.11. Therefore, from the results of the six layouts it can be deduced that using a RS of 2 with a modified database it is possible to estimate a randomly sampled layout with similar track widths and separations.

6.3.4. Program Execution Time

This section presents the execution times for Calma and compares this with Depict. Employing the tracks presented in the previous sections, the execution time for Calma and Depict was obtained using the Unix "time" command by averaging three measurements made on a Sun Ultra-60 with 768 MB of RAM. Table 6.12 presents the associated layout areas and program execution times.

Track type	Layout Area (μm^2)	Execution time for Depict (s)	Execution time for Calma (s)
Type 2-1	90	70.62	4.43
Type 3-1	25	11.67	3.19
Type 4-1	35	13.70	3.23
Type A	37.5	8.91	3.27
Type B	56.25	28.84	3.65
Type C	35	9.65	3.23
Type D	35	11.01	3.24
Type E	35	10.01	3.21
Type F	135	29.54	3.70

Table 6.12.: Associated layout areas and program execution times.

From the above table, it can be observed that Calma is between 3.0 and 15.9 times faster than Depict. The execution time for Depict is proportional to the layout area, increasing as the area increases. One exception is track type F where although the layout area is 2 times larger than type 2-1, the execution time is 2.4 times faster. The reason for this is due to the lower layout density of track type F compared to type 2-1 which may result in simplified calculations when determining the aerial image. In

addition, although the execution times for Calma are also dependent on layout area and density, these effects are less significant as shown by the faster execution times.

6.4. Summary

Calma is capable of estimating the critical area of layouts after pattern transfer with a better accuracy than is obtained directly from the mask layout. It is also approximately 3.0 - 15.9 times faster than running Depict on the mask layout where the execution times are roughly proportional to the layout area. From the results obtained, the following observations can be made.

- A region size of 1 has limited use and is only capable of estimating critical areas of two parallel tracks.
- A region size of 2 using a modified database resulted in better critical area estimations than those obtained from using a smaller region size and/or an unmodified database.
- Modified databases provide better estimates than unmodified databases where a typical error range in the layouts examined was is 0 - 5.1 % compared to an unmodified database of 0.3 - 5.3 %.
- The theoretical defect size limits the maximum defect size for which the method is valid.

It can be observed from these conclusions that Calma's limitation lies with providing sufficient track information to categorise an address (which is determined by the region size setting) and the database. This limitation is user specified in the control file when running Calma. Due to the limited categorisation information obtained from a region size 1 address, it is difficult to place any real certainty on the results obtained and it

is recommended that region size 2 or above be used to estimate the pattern transfer effects on a mask layout.

Another factor that affects the accuracy of the estimation is the database type used. As mentioned previously, for some addresses an averaged value can be used to better represent a particular address type in the modified database. Although this is only possible for a small number of selected addresses, increased accuracy can be observed in the results. In addition, it should also be noted that the more detailed the catergorisation the more database information is required. The database should contain sufficient information for all the addresses used in categorising the layout. Table 6.13 details the theoretical and actual number of addresses employed to generate the database.

Region size	Theoretical number of addresses	Actual number of addresses used	Theoretical defect size (μm)
1	2^8	41	0.5
2	$2^8 \times 2^{16}$	252	1.0
3	$2^8 \times 2^{16} \times 2^{24}$	2017*	1.5
4	$2^8 \times 2^{16} \times 2^{24} \times 2^{32}$	14904*	2.0

Table 6.13.: Theoretical and actual number of addresses required for database generation ($\star$ estimated value).

From the above table it can be observed that using a larger region size results in an increased theoretical defect size but also requires a larger number of addresses to create the database. The database for region size 3 and 4 have not been generated and so the number of addresses required were estimated. This was performed by employing the function $Ae^{(\mathbf{B}\times region\,size)}$ to approximate the actual number of addresses needed where the values of $A = 5$ and $B = 2$ were used. Although large region sizes are desirable its practical use is limited by the time required to create these databases. For modern manufacturing processes, using a region size of 4 should suffice as it would not be unreasonable to expect the majority of defects to be less than 2.0 μm. Thus, a

balance would have to be struck between the accuracy and detail required in a layout to the size of the database generated.

7. Conclusion and Future Work

It has been shown that in IC manufacturing, the critical area and yield of the process is affected by pattern transfer. Hence, it is desirable that the critical area of the resulting mask layout after pattern transfer can be determined accurately and efficiently. This thesis has described two different methods (EPCA and Calma) for the estimation of critical area from an IC after pattern transfer. The two methods were based on different approaches. The first, EPCA used parallel lines to build a database which was then used to estimate the critical area of a mask layout. There was no limitation placed on the input mask layout however the usage of parallel lines in the estimation would result in inaccuracies for some layouts. The second method, Calma which is based on layout categorisation is not limited to using parallel lines for the estimation and could be adapted to more complex mask layouts. Thus as a proof of concept Calma was tested using parallel lines with different parameter settings. The region size was chosen as either 1 or 2 and two different databases were used, modified and unmodified (the modified database is one where the critical area of some of the addresses have been averaged). Using a modified database with a region size setting of 2 has been shown to produce the best critical area estimates out of the four different combinations. However for completeness, table 7.2 illustrates the error ranges obtained from employing the modified and unmodified databases with two region size settings in Calma.

Model	Error range in average number of faults, $\lambda_{0-0.5}$ up to defect size of 0.5 μm (TDS 1) compared with Depict (%)	Error range in average number of faults, $\lambda_{0-1.0}$ up to defect size of 1.0 μm (TDS 2) compared with Depict (%)
Region size 1 unmod	8.1 - 23.4	0.5 - 9.6
Region size 1 mod	2.5 - 15.2	0.05 - 9.4
Region size 2 unmod	0.07 - 10.8	0.3 - 5.3
Region size 2 mod	0 - 10.7	0 - 5.1
Mask layout	0.5 - 13.4	0.5 - 11.8
Aerial Image (Depict)	-	-

Table 7.2.: Error ranges for various Calma settings.

Using the error ranges as a rough estimate to the accuracy of the various Calma settings, it can be observed that the modified region size 2 settings produced the best results. This validates the results from chapter 6 from which using a modified database with a region size setting of 2 gave the most accurate estimations. The unmodified database for region size 2 gives better error ranges than the mask layout and can be used where a high level of accuracy is not needed, otherwise it is preferable to use the modified database for such calculations as its error range is even better than the unmodified database.

7.1. Improvements to Calma

7.1.1. Automatic Database Generation

One key improvement in Calma would be an automatic database generation module. It should generate and calculate the critical area from a layout for a specific address not

present in the database. At present manual modifications have to be made to CIF files to generate the different addresses necessary for the database. The number of addresses generated was relatively small for parallel lines of a region size of 1 and 2. However, larger region sizes and more complicated database entries will be more difficult and prone to human error. This module will assist in automating Calma which can then be utilised with little or no user intervention.

7.1.2. Further Analysis of Addresses

Once a layout has been categorised by Calma, the number of different region addresses that form the layout is internally stored. This information can be employed for statistical analysis of the layout. The number and type of a certain layout feature can be calculated if the appropriate region address for the feature is available. For example, a simple layout of two parallel tracks can be broken down into the track edges or middle. In addition, error analysis can be carried out on the different features. By specifying the error bound for each region address feature, a range of estimated critical area values can be obtained. This can also be used to inform a designer of the feature that is most susceptible to defects.

7.1.3. Improved Address Algorithm

At present the Calma address generation algorithm searches each individual address in a bitmap and generates the address for each location. This procedure is relatively time consuming and is of $O(xy)$ complexity, where x is the number of bits along the x axis and y the number of bits along the y axis of the layout. This value is dependent upon the minimum grid sized used to partition the layout into individual bits. However, it should be possible to utilise some of the region address information obtained on one row of a bitmap to constitute the region address of an adjacent row without having to use the address generation algorithm. Hence if a more detailed region size value is specified,

then fewer addresses in the layout bitmap have to be generated. An example of how an improved address algorithm can function is illustrated in figure 7.1. By calculating the region address for a region size of 1 for the two hashed area (bits), it can be seen that the region address of the black bit can be constructed from these two region addresses. Thus instead of having to calculate the region address of 3 bits, only 2 bits are required.

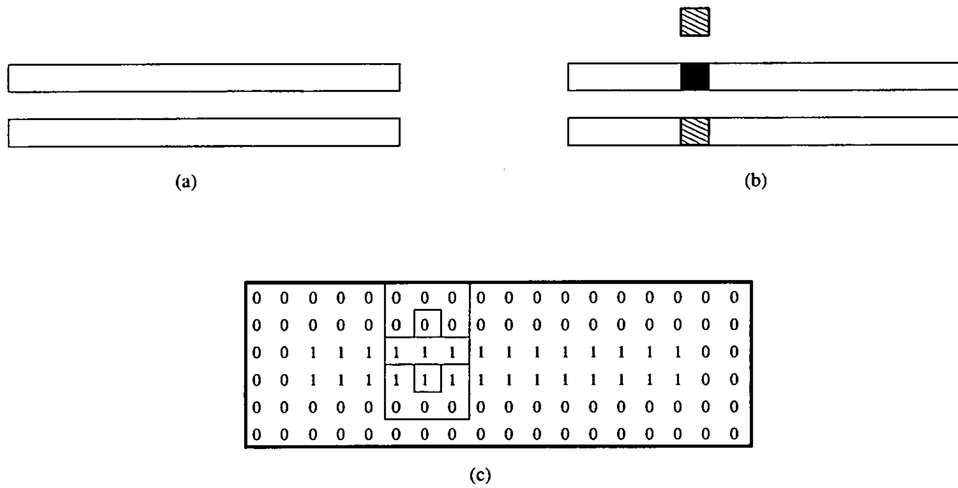


Figure 7.1.: Alternate address generation with fewer bit points. (a) Two parallel lines. (b) Address calculated for the hashed areas which will determine the address for the black area. (c) Bitmap of the layout showing the overlap of the two calculated areas.

7.1.4. Hierarchical Input Files

The generation of the bitmap addresses from a CIF file is an important step in the categorisation method. As chip density and size increases, CIF file sizes increase accordingly and most if not all CIF files for large ICs are of a hierarchical nature. At present Calma is only capable of processing CIF files of a flattened layout. Hence, for Calma to be used on larger layouts it is important for it to be able to process CIF files of a hierarchical nature. By partitioning a layout into smaller more managable layouts, less memory will be required to store the bit addresses and there is a possibility of parallelising Calma. Thus employing a layout parser which breaks a layout into the individual cells enables

the bit addresses for each of these cells to be processed independently. Next by integrating all the individual bit addresses the total critical area of a layout can be obtained. A schematic diagram of the layout parser and its interaction with Calma is shown in figure 7.2.

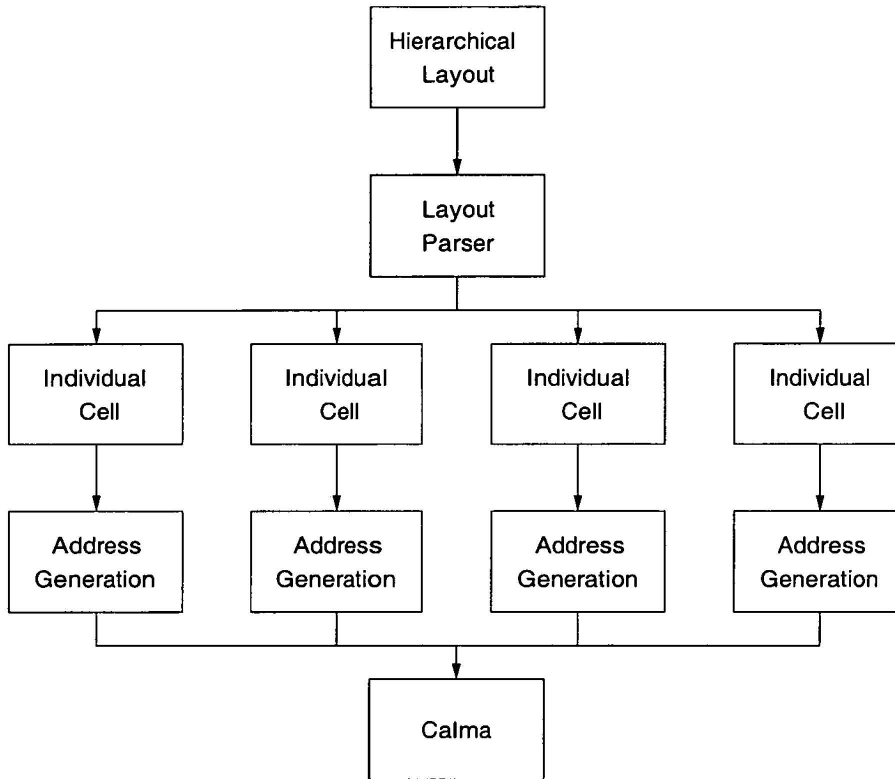


Figure 7.2.: A schematic diagram for modifications to Calma to handle hierarchical input files.

7.1.5. Generic Mask Layouts

At present Calma has been tested with parallel lines but it is envisaged that it would be able to calculate the estimated critical area of more generic mask layouts. Currently, the bitmap generation of the mask layout has been designed specifically for parallel tracks. However, it is a simple procedure to modify the bitmap generation module with no changes necessary for the address generation module. However, this change will affect

the region size setting as layout area covered by the old RS value will be different to the new RS value. Figure 7.3 illustrates the generation of the current and proposed bitmap for two parallel lines.

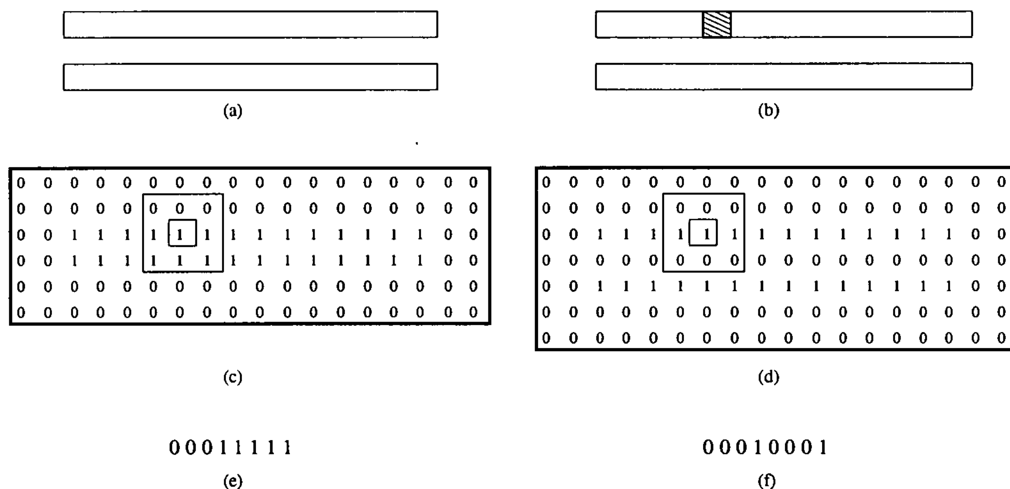


Figure 7.3.: Bitmap and address generation for two paralel lines (region size of 1). (a) Two parallel tracks used. (b) Enclosed bit for which its address is to be determined. (c) Bounding region using the current module. (d) Bounding region using the new module. (e) Bitmap address from the current module. (f) Bitmap address from the new module.

7.2. Improvements to EPCA

One limitation of EPCA has been the parallel lines in the database from which the estimated CA of a layout is calculated. This limitation is due to the transformation of a mask layout into the equivalent parallel lines used to calculated the CA. One improvement will be to use an additional database which would consist of track corners of varying separations as illustrated in figure 7.4.

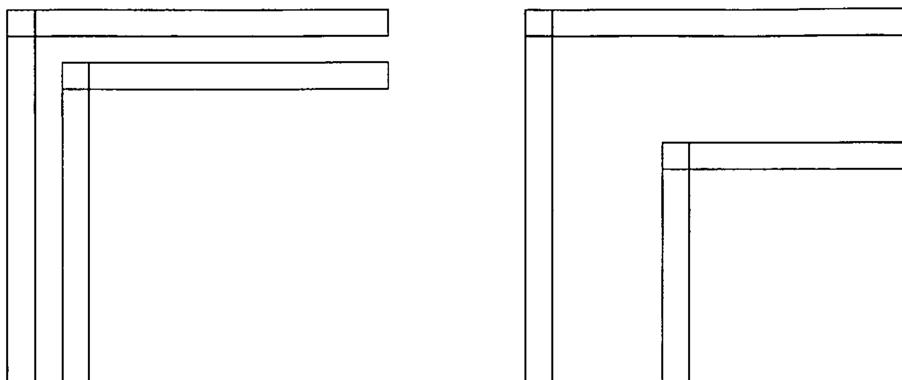


Figure 7.4.: Two track corners of varying separations.

The difficulty involved is the integration of this database with the exsisting database of parallel lines. One solution is to use the Calma categorisation method to determine the percentage of track corners in the layout and using a weighted percentage from each of the two databases. In addition, it is hoped that this technique may be expanded to include databases of different track characteristic for the calculation of estimated CA. However, further testing will be required to test the feasibilty of the proposed technique.

7.3. Advantages of Calma

Calma has been shown to be capable of accurately determining the critical area of a layout after pattern transfer. However its effectiveness is also limited by the database file generated where the more detail the file the more accurate Calma. Although Calma utilises a pattern transfer simulator to generate the databases, it is only performed once and any other layout with similar characteristics may then be evaluated by Calma.

It is possible to use a pattern transfer simulator on a mask layout and then use a critical area extraction program to obtain the critical area of a layout after pattern transfer. However, this setup is not practical as some layouts are large and the time required to run the simulator can be considerable. In addition, if minor changes are made to the

mask layout the simulator will have to be run again to obtain the new pattern. This is not the case with Calma as if a modification is made to the mask, the region addresses will have to be recalculated but the database will still be useable.

Once improvements have been made to Calma, it will also be capable of generating feature statistics of a layout. This can be useful as manual modifications to the database file can be made to gauge the effectiveness of newer process methods. For example, a new process method for track corners has been developed where the the pattern on the wafer is identical to the mask layout. This change can be made in the database and the new estimated critical area obtained for this improved process change.

In addition, with IC techonology rapidly advancing, larger layouts are becoming the norm and it might be impractical to use these layouts (even hierarchical) as inputs to Calma. Hence, one solution is to integrate Calma with EYES. The sampling technique used in EYES will enable the calculation of estimated critical area for larger ICs.

Bibliography

[1] J. S. Kilby. *Miniaturized Electronic Circuits Integrated Circuit.* US Patent Number 3138743, 1959.

[2] R. N. Noyce. *Semiconductor Device-and-Lead Structure Integrated Circuit.* US Patent Number 2981877, 1959.

[3] *The International Technology Roadmap for Semiconductors.* Semiconductor Industry Association, 2001.

[4] S. M. Sze. *Semiconductor Devices: Physics and Technology.* Wiley, 2002.

[5] S. Mahajan and K. S. Sree Harsha. *Principles of Growth and Processing of Semiconductors.* McGraw-Hill Series in Materials Science and Engineering, 1999.

[6] B. E. Deal and A. S. Grove. General relationship for the thermal oxidation of silicon. In *Journal of Applied Physics*, page 3770, 1965.

[7] C. Ho et al. In *Journal of Electrochem. Soc.*, page 1146, 1978.

[8] J. R. Ligenza. In *Journal of Phys. Chem.*, page 2011, 1961.

[9] L. E. Katz. *Oxidation, VLSI Technology.* McGraw-Hill, 1983.

[10] T. H. Smith et al. Run by run advanced process control of metal sputter deposition. In *IEEE Trans. on Semiconductor Manufacturing*, page 276, May 1998.

[11] S. P. Muraka and S. W. Hymes. Critical reviews in solid state and material science. page 87, 1995.

[12] A. Fick. In *Ann. Phys.*, page 59, 1855.

[13] J. C. C. Tsai. *Diffusion, VLSI Technology.* McGraw-Hill, 1983.

[14] S. K. Ghandi. *VLSI Fabrication Principles.* John Wiley and Sons, 1983.

[15] L. F. Thompson et al. Introduction to microlithography. In *ACS Symposium*, 1983.

[16] N. S. Viswanathan M. D. Levinson and R. A. Simpson. Improving resolution in photolithography with a phase-shifting mask. In *IEEE Trans. Electron Devices*, page 1828, 1982.

[17] A. Nakae et al. Improvement in optical proximity correction by optimizing second illumination source. In *Jpn. J. Appl. Physics*, page 6396, 1996.

[18] J. A. Liddle et al. The scattering with angular limitation in projection electron-beam lithography (scalpel) system. In *Jpn. J. Appl. Phys.*, page 6663, 1995.

[19] D. Stokes and G. S. May. Real-time control of reactive ion etching using neural networks. In *IEEE Trans. on Semiconductor Manufacturing*, page 469, Nov 2000.

[20] Z. Ning et al. Comparative analysis of run-to-run control algorithms in the semiconductor manufacturing industry. In *IEEE/SEMI Advanced Semiconductor Manufacturing Conf.*, pages 375–381, 1996.

[21] L. A. Doty. *Statistical Process Control.* New York: Industrial Press, 1991.

[22] *Applied materials, Novellus, LAM research lead cluster tool market.* CMP Media Inc., 1998.

[23] W. Maly, A. Strojwas, and S. Directory. Fabrication based statisticl design of monolithic ics. In *IEEE International Symposioum on Circuits and Systems*, pages 135–138, Apr 1981.

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

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

[26] 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.

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

[28] C. H. Stapper. Modelling of defects in integrated circuit photolithographic patterns. *IBM J. Res. Develop.*, 28(4):461–474, July 1984.

[29] 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.

[30] Z. Stamenkovic and N. Stojadinovic. New defect size distribution function for estimation of chip critical area in integrated circuit yield models. *Electronics Letters*, 28(6):528–530, Mar 1992.

[31] C. H. Stapper. The effects of wafer to wafer defect density variations on integrated circuit defect and fault distributions. *IBM J. Res. Develop.*, 28(4):87–97, July 1984.

[32] T. Yanagawa. Influence of epitaxial mounds on the yield of integrated circuits. In *IEEE Proc.*, pages 1621–1628, Sep 1969.

[33] W. E. Ham. Yield-area analysis: Part i – a diagnostic tool for fundamental integrated circuit process. volume 39, Princeton, N.J, Jun 1978. RCA Laboratories.

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

[35] 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 Academic Publishers, Boston; Lancaster, 1987.

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

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

[38] P. Simon Y. Fei and W. Maly. New yield models for dsm manufacturing. *IEDM*, 36(5):1–4, 2000.

[39] I. Fukushi et al. A 256 kb ecl ram with redundancy. In *ISSCC Dig. Tech. Papers*, pages 134–135, Feb 1988.

[40] S. Kikeda et al. Optimized redundancy selection based on failure related yield model for a 64-mb dram and beyond. In *IEEE Solid-State Circuits*, pages 1550–1555, Nov 1991.

[41] Y. Takafuji et al. A poly-si tft monolithic lc data driver with redundancy. In *ISSCC Dig. Tech. Papers*, pages 118–119, Feb 1992.

[42] G. H. Teepe and W. L. Engl. A bipolar correlator with redundancy. In *IEEE Solid-State Circuits*, pages 1190–1195, Dec 1987.

[43] B. Joshi et al. A bicmos 50 mhz cache controller for superscalar microprocessors. In *ISSCC Dig. Tech. Papers*, pages 110–111, Feb 1992.

[44] R. C. Varshney T. L. Michalka and J. D. Meindl. A discussion of yield modeling with defect clustering, circuit repair and circuit redundancy. In *IEEE Transaction on Semiconductor Manufacturing*, pages 116–127, Aug 1990.

[45] J. K. Khare, D. B. I. Feltham, and W. Maly. Accurate estimation of defect-related yield loss in reconfigurable VLSI. *IEEE Solid State Circuits*, SC-28(2):146–156, Feb 1993.

[46] D. G. Boyer. Symbolic layout compaction review. In *IEEE Proc. DAC*, pages 383–389, 1988.

[47] J. L. Burns and R. Newton. Sparcs: A new constraint-based ic symbolic layout spacer. In *IEEE Proc. Custom Integrated Circuits*, pages 534–539, 1986.

[48] V. K. R. Chiluvuri and I. Koren. Layout-synthesis techniques for yield enhancement. *IEEE Trans. on Semiconductor Manufacturing*, 8(2):178–187, May 1995.

[49] Y. Bourai and C.-J. R. Shi. Layout compaction for yield optimization via critical area minimization. In *Proc. Design, Automation and Test in Europe*, pages 122–125, 2000.

[50] T. Yoshimura and E. S. Kuh. Efficient algorithms for channel routing. *IEEE Trans Comp. Aided Design*, CAD-1(1):25–35, Jan 1982.

[51] A. Pitaksanonkul, S. Thanawastien, C. Lursinsap, and J.A. Gandhi. DTR: A defect-tolerant routing algorithm. In *26th ACM/IEEE Design Automation Conference*, pages 795–798, 1989.

[52] S. Kuo. YOR: A yield-optimizing routing algorithm by minimizing critical areas and vias. *IEEE Trans Comp. Aided Design*, 12(9):1303–1311, Sep 1993.

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

[54] 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.

[55] I. A. Wagner and I. Koren. An interactive vlsi cad tool for yield estimaton. In *IEEE Transactions on Semiconductor Manufacturing*, pages 130–138, 1995.

[56] J. H. N. Mattick, R. W. Kelsall, and R. E. Miles. Improved critical area prediction by application of pattern recognition techniques. *Microelectronics Reliability*, 36(11–12):5–18, Nov–Dec 1996.

[57] P. K. Nag and W. Maly. Hierarchical extraction of critical area for shorts in very large scale ICs. In *IEEE Workshop on Defect and Fault Tolerance in VLSI Systems*, pages 19–27, Lafayette, Louisiana, Nov 1995.

[58] W. Maly, I. Bubel, T. Waas, P. K. Nag, D. Schmitt-Landsiedel, and S. Griep. AFFCCA: A tool for critical area analysis with circular defects and lithography deformed layout. In *IEEE Workshop on Defect and Fault Tolerance in VLSI Systems*, pages 10–18, Lafayette, Louisiana, Nov 1995.

[59] H. Xue, C. Di, and J.A.G. Jess. A net-oriented method for realistic fault analysis. In *IEEE ACM Int Conf Comput Aided Des*, pages 78–83, 1993.

[60] J. Pineda de Gyvez and J. A. G. Jess. Systematic extraction of critical areas from ic layouts. In *IEEE Workshop on Defect and Fault Tolerance in VLSI Systems*, pages 47–62, 1990.

[61] G. A. Allan. *EYE: User Manual.* University of Edinburgh, Electrical Engineering Dept.,The King's Buildings,Edinburgh, EH9 3JL Scotland,UK, 1995.

[62] G. A. Allan and A. J. Walton. Hierarchical critical area extraction with the EYE tool. In *IEEE Workshop on Defect and Fault Tolerance in VLSI Systems*, pages 28–36, Lafayette, Louisiana, Nov 1995.

[63] G. A. Allan and A. J. Walton. Sampling based yield prediction for ULSI. In *SPIE Symposium on Microelectronic Manufacture*, pages 198–209, Austin, Texas, Oct 1996.

[64] E. Papadopoulou and D. T. Lee. Critical area computation via voronoi diagrams. In *IEEE Trans. on Computer Aided Deisgn of Integrated Circuits and System*, pages 463–474, Apr 1999.

[65] E. Papadopoulou. Critical area computation for missing material defects in vlsi circuits. In *IEEE Trans. on Computer Aided Deisgn of Integrated Circuits and System*, pages 583–597, May 2001.

[66] *TMA Workbench User's Manual.* Technology Modeling Associates, Inc, 3950 Fabian Way, Palo Alto, CA 94303-4605, 1997.

[67] E. Barouch, J. W. Cahn, U. Hollerback, and S. A. Orszag. Numerical simulation of submicron photolithographic processing. *Scientific Computing*, 6(3):229–250, 1991.

[68] K. K. H. Toh and A. R. Neureuther. Three-dimensional simulation of optical lithography. In *SPIE - The International Society for Optical Engineering*, pages 356–367, 1991.

[69] *DEPICT Reference Manual.* Technology Modeling Associates, Inc, 3950 Fabian Way, Palo Alto, CA 94303-4605, 1997.

[70] W. W. Flack et al. Advanced simulation techniques for thick photoresist lithography. In *SPIE-Int. Soc. Opt. Eng. Proceedings of Spie - the International Society for Optical Engineering*, pages 789–804, Santa Clara, California, 1997.

[71] M. S. C. Yeung. Modeling aerial images in two and three dimensions. In *Kodak Microelectronics Seminar*, pages 115–126, 1985.

[72] H. P. Urbach and D. A. Bernard. Modeling latent image formation in photolithography, using the helmholtz equation. In *SPIE Symposium of Optical/Laser Microlithography III*, pages 278–293, 1989.

[73] D. A. Bernard, J. Li, J. C. Rey, K. Rouz, and V. Axelrad. Efficient computational techniques for aerial imaging simulation. In *SPIE Symposium of Optical/Laser Microlithography IX*, pages 273–287, Mar 1996.

[74] Tom Stevenson (University of Edinburgh). Personal communication.

[75] R. S. Longhurst. *Geometrical and Physical Optics.* Longman Scientific and Technical, 1990.

A. User Manual for Programs

This appendix describes the user manuals for vaious programs written to support the work reported in this thesis. All the programs were written in C++ and was compiled using gcc on a Solaris operating system. No responsibility is accepted for the use of the programs, in part or whole, outwith the work presented herein.

A.1. ATOM

The atom program calculates the estimated critical area of a layout that has undergone pattern transfer using the critical area curve generated from the mask layout. It requires a database of critical area for the various parallel tracks as a function of defect size which is then used to estimate the aerial image critical area. It requires a control file to be specified as input to the program.

As an example, given an input file (input) which consists of two columns; the first of which is the defect sizes (0.1 - 3.0 μm at 0.1 μm increments) and the remaining column the critical area calculations obtained using EYE on the mask layout. Next, a data file (datafile) which contains the critical areas after pattern transfer for 10 equivalent parallel tracks is also available. To estimate the resulting critical area curve after pattern transfer for the mask layout, the following atom command is required.

```
>atom atom.input
```

where "atom.input" is the control file and consists of the following lines

```
output      ← output file name
input       ← input file name
datafile    ← data file name
10          ← number of equivalent parallel tracks used in the data file
30          ← number of defect points in the data file
```

The resulting output file created for the estimated critical area as a function of defect size is "output".

A.2. CALMAT

The following is the user manual for the various programs used in CALMAT. It should be noted that the main program Calma calls the various subprograms and the user only has to know the usage format of Calma. The exception to this is the calma-database program where user intervention is required to create the databases. In addition, calma-plot is also included as it can be used to plot simple graphs using the appropriate files and may prove useful when used outwith of Calma.

A.2.1. Calma

Calma calculates the estimated critical area of a mask layout that has undergone pattern transfer by using the CIF file of the mask layout. It is also capable of comparing the estimated critical area with the critical area of the aerial image. This requires the mask and aerial image CIF files of a layout. In actuality, the aerial image critical area (or aerial image CIF file) of a layout will not be available and calma is then used to

estimate this critical area. Calma also requires a database of critical area addresses which is used to estimate the critical area after pattern transfer. It requires a control file to be specified as input to the program.

Take for example an input CIF file (input.cif) which consists of tracks with a width and separation of 0.5 μm. Two databases are used, each consisting of a positive (typeA_pos and typeB_pos) and negative (typeA_neg and typeB_neg) database where the first column of each database is the defect size (0 - 1.25 μm at 0.05 μm increments) while the remaining columns contain the critical area after pattern transfer for the addresses specified. Next the region size data file is specified and is set to create (write) the file for a region size of 1. The bitmap file (bitmapfile) is used to store the bitmap created from the mask layout with the given bit size (0.5 μm). If this file exists then the bitmap for the layout is not calculated and the available file used. To estimate the critical area after pattern transfer for the mask layout, the following calma command is required.

```
>calma calma.input
```

where "calma.input" is the control file and consists of the following lines

```
output                     ← output file name
bitmapfile 0.5             ← bitmap file name and bit size
input 0.5 0.5              ← input file name with the track width and separation
region1.data 1 write       ← region size data file, region size and mode
0 1.25 0.05                ← defect start and stop size with the increment
2                          ← number of databases
typeA_pos typeA_neg 10 1   ← the first positive and negative database names with
                             the number of addresses and multiplier
typeB_pos typeB_neg 7 1    ← the second positive and negative database names with
                             the number of addresses and multiplier
```

The resulting output file generated detailing the mask layout and estimated critical areas is "output".

A.2.2. Calma-plot

The calma-plot program runs the external program "gnuplot" on a data file which contains a given number of columns which generates various curves. These columns are used as the x and y axes in the curves generated and is specified in a control file and the output is in the form of a postscript file

For example, given a data file (datafile) which consists of 3 columns, the first of which is the defect sizes and the remaining two the critical area calculations obtained from Depict and the mask layout. In order to create two critical area curves labelled "Depict" and "Mask" as a function of defect size from columns 2 and 3 respectively where column 1 is used as the x-axis requires the use of the following calma-plot command.

```
>calma-plot calma-plot.input
```

where "calma-plot.input" is the control file and consists of the following lines

```
output.ps                      ← output plot name
datafile                       ← input data file name
3                              ← number of columns in data file
defect_size critical_area      ← x-axis label and y-axis label
2                              ← number of output curves
1 2 Depict                     ← first curve labelled ''Depict''
1 3 Mask                       ← second curve labelled ''Mask''
```

The resulting postscript file created is "output.ps".

A.2.3. Calma-database

The program calma-database collates the results of individual data files into a single database. The individual data files consists of two columns such as the defect sizes

and critical areas for a given mask layout. It is also capable of labelling the individual critical area columns in the database and requires a control file to be specified as input to the program.

For example, given two data files (filel.data and file2.data) which consists of 2 columns, the first of which is the defect sizes (0.1 - 3.0 μm at 0.1 μm increments) and the remaining column the critical area calculated. In order to create a merged database from the two individual files, which will be labelled "typel" and "type2" requires the use of the following calma-plot command.

```
>calma-database calma-database.input
```

where "calma-database.input" is the control file and consists of the following lines

```
output.database        ← merged database
0.1 3 0.1              ← input data file name
2                      ← number of data files
filel.data typel 1     ← first file name and label used in database
file2.data type2 1     ← second file name and label used in database
```

The resulting merged database file generated is "output.database".

A.3. Miscellaneous Programs

A.3.1. Cif2tl

The cif2tl program converts between the CIF file format used to represent mask layouts and the TL2 format used by Depict. It is only capable on working with a flat layout and any hierarchial layout would have to be flattened before running cif2tl. The output

from the program are individual TL2 files for each layer. The only parameter cif2tl accepts is the input CIF file name.

For example, to execute the program on a layout (input.cif) which contains tracks placed on the "metal1" and "metal2" layer the command is:

```
>cif2tl input.cif
```

The resulting files are the tl2 files with the original file name appended with either "metal1.tl2" or "metal2.tl2" and can be shown using the Unix "ls" command:

```
>ls
input.cif.metal1.tl2
input.cif.metal2.tl2
```

A.3.2. Tl2cif

The tl2cif program converts between the TL2 file format used by Depict and the widely used CIF format used to represent mask layouts. It requires a control file to be specified as input to the program.

As an example, the two TL2 layer files (for a "metal1" and "metal2" layer) as shown with the "ls" command requires the use of the following tl2cif command.

```
>ls
tl2cif.input input.cif.metal1.tl2 input.cif.metal2.tl2
>tl2cif tl2cif.input
```

where "tl2cif.input" is the control file and consists of the following lines

```
output.cif                              ← output file name
cell                                    ← user specified cell name
2                                       ← number of input TL2 layers
input.cif.metal1.tl2       metal1       ← first TL2 layer file name and layer name
input.cif.metal2.tl2       metal2       ← second TL2 layer file name and layer name
```

The the resulting output CIF file created is "output.cif".

A.3.3. Mktrack

The mktrack program assists in the generation of parallel or corner tracks of fixed width and separation. The resulting output file is a flattened CIF file. It requires a control file to be specified as input to the program.

For example, to create 5 parallel tracks with a separation and width of 0.5 μm on a "metal" layer requires the use of the following mktrack command.

```
>mktrack mktrack.input
```

where "mktrack.input" is the control file and consists of the following lines

```
parallel      ← track type required
output.cif    ← output file name
cell          ← user specified cell name
metal         ← layer name
5             ← number of tracks
0.5           ← width of track
0.5           ← track separation
```

The resulting output CIF file is "output.cif".

A.3.4. Scalecif

The scalecif program assists in the scaling of a CIF file. Using the input and output technology sizes, scalecif modifies the scaling factor of a cell. At present it is only capable of modifying the first cell in a flattened layout. It requires a control file to be specified as input to the program.

For example, to scale a CIF file (input.cif) from a 2.0 μm technology size to 3.0 μm requires the use of the following scalecif command.

```
>scalecif scalecif.input
```

where "scalecif.input" is the control file and consists of the following lines

```
$a dummy line        ← a dummy line
input.cif            ← input file name
output.cif           ← output file name
2                    ← input technology size
3                    ← output technology size
```

The resulting output CIF file is "output.cif".

B. Reference Manual for Programs

This appendix describes the reference manuals for vaious programs written to support the work reported in this thesis. All the programs were written in C++ and was compiled using gcc on a Solaris operating system. No responsibility is accepted for the use of the programs, in part or whole, outwith the work presented herein.

B.1. ATOM

Synopsis

```
atom [control file]
Control file format :
     [Name of output file]
     [Name of input file]
     [Name of data file]
     [Number of columns]
     [Number of rows]
```

Description

Atom calculates the estimated critical area of a layout that has undergone pattern transfer using the critical area curve generated from the mask layout. It requires a

database of critical area for the various parallel tracks as a function of defect size which is used to estimate the aerial image critical area. The program relies on control file parameters described below.

Control file parameters

Name of output file: The file name for the results generated.

Name of input file: The file name of the critical area curve generated from the mask layout.

Name of data file: The database file name which contains the critical area of parallel tracks after pattern transfer.

Number of columns: The number of equivalent parallel tracks used in the database.

Number of rows: The number of defect points used to generate the critical area curve.

B.2. CALMAT

The following is the manual for the various programs used in CALMAT. It should be noted that the main program Calma calls the various subprograms and the user only has to know the usage format of Calma. The exception to this is the calma-database program where user intervention is required to create the databases. However, for the completeness the remaining subprograms and their operation information is detailed below.

B.2.1. Calma

SYNOPSIS

```
calma [control file]
Control file format :
      [Output file name / stdout]
      [Bitmap file name] [unit bitmap length]
      [Input cif file name (without .cif)] [track width] [track separation]
      [Region data file] [Region size] [Region file setting (read/write)]
      [defect start size] [defect stop size] [defect increment size]
      [Number of data files]
      [Data file name (pos)] [Data file name (neg)] [Number of columns] [unit length]
```

DESCRIPTION

Calma calculates the estimated critical area of a mask layout that has undergone pattern transfer by using the CIF file of the mask layout. It is also capable of comparing the estimated critical area with the critical area of the aerial image. This requires the mask and aerial image CIF files of a layout. In actuality, the aerial image critical area (or aerial image CIF file) of a layout will not be available and calma is then used to estimate this critical area. Calma also requires a database of critical area addresses which is used to estimate the critical area after pattern transfer. The program relies on control file parameters described below.

CONTROL FILE PARAMETERS

Output file name / stdout: The file name for the results generated. If a file name of "stdout" is used then the results are displayed on screen.

Bitmap file name: The bitmap file name.

Unit bitmap length: The unit size of each bit in the bitmap file (μm).

Input cif file name (without .cif): The input CIF file name without ".cif".

Track width: The width of the track in the CIF file (μm).

Track separation: The separation between tracks in the CIF file (μm).

Region data file: The region data file name.

Region size: The size of the region used to categorise the layout.

Region file setting (read/write): The setting used on the region file. If set for "read", the program will read the region and symmetrical data from the file. If set for "write", the program will write the region and symmetrical data to the file.

Defect start size: The start size of the defects (μm).

Defect stop size: The stop size of the defects (μm).

Defect increment: The increment between defect sizes.

Number of data files: The number of data files .

Data file name (pos): The data file name for positive addresses.

Data file name (neg): The data file name for negative addresses.

Number of columns: The number of columns (excluding the defect size column) in the data file.

Unit length: The unit size of the critical area in the data file (μm).

B.2.2. Calma-cif2tl

SYNOPSIS

```
calma-cif2tl [track width] [track separation] [region size] [output cif file]
             [bitmap file] [tl2cif config file] [input cif file]
```

DESCRIPTION

Calma-cif2tl converts between the CIF file format used to represent mask layouts and the TL2 format used by Depict. It is only capable on working with a flat layout and any hierarchial layout would have to be flattened before running calma-cif2tl. The program relies on the program parameters described below.

PROGRAM PARAMETERS

Track width: The width of the track in the CIF file (μm).

Track separation: The separation between tracks in the CIF file (μm).

Region size: The size of the region used to categorise the layout.

Output cif file: The CIF file name where the .cif suffix will be appended to the file name.

Bitmap file: The bitmap of the input CIF file.

Tl2cif config file: The config file name used by the calma-tl2cif subprogram.

Input cif file: The CIF file name.

B.2.3. Calma-tl2cif

SYNOPSIS

```
calma-tl2cif [control file]
Control file format :
     [Name of bitmap file]
     [Name of cif file]
     [Name of topmost cell]
     [track width] [track separation]
     [x min] [x max] [y min] [y max]
     [Number of layers]
     [Name of layer file] [Name of layer]
```

DESCRIPTION

Calma-tl2cif converts between the TL2 file format used by Depict and the widely used CIF format used to represent mask layouts. The program relies on control file parameters described below.

CONTROL FILE PARAMETERS

Name of bitmap file: The output bitmap file name.

Name of cif file: The input CIF file name.

Name of topmost cell: The topmost cell name in the CIF file.

Track width: The width of the track in the CIF file (μm).

Track separation: The separation between tracks in the CIF file (μm).

X min: The minimum x coordinate in the CIF file.

X max: The maximum x coordinate in the CIF file.

Y min: The minimum y coordinate in the CIF file.

Y max: The maximum y coordinate in the CIF file.

Number of layers: The number of layers in the CIF file.

Name of layer file: The TL2 file name for the CIF layer.

Name of layer: The layer name used in the CIF file.

B.2.4. Calma-runeye

Synopsis

```
calma-runeye [defect start size] [defect stop size] [defect increment]
             [bounding layer (none = 0)] [input cif file] [output data file]
```

Description

Calma-runeye runs the EYE program used to calculate the critical area of a mask layout. The result of the critical areas calculated for the defect sizes are stored in a data file. The program relies on the program parameters described below.

Program parameters

Defect start size: The start size of the defects (μm).

Defect stop size: The stop size of the defects (μm).

Defect increment: The increment between defect sizes.

Bounding layer: The layer name used as a bounding box for the CIF file. Using a layer of "0" would mean that no bounding layer is specified.

Input cif file: The input CIF file name.

Output data file: The output data file name.

B.2.5. Calma-symmetry

Synopsis

```
calma-symmetry [region size] [output data file]
```

Description

Calma-symmetry calculates the symmetrical addresses of each address in a given region size. The results of the calculation are stored in an external data file. The program relies on the program parameters described below.

Program parameters

Region size: The size of the region used to categorise the layout.

Ouput data file: The output data file name.

B.2.6. Calma-plot

Synopsis

```
calma-plot [control file]
Control file format :
     [Output plot file]
     [Input data file]
     [Number of columns in data file]
     [x axis label] [y axis label]
     [Number of graphs]
     [x axis column] [y axis column] [graph label]
```

Description

Calma-plot runs the external program "gnuplot" on a data file which contains a given number of columns which generates various curves. These columns are used as the x and y axes in the curves generated and is specified in a control file and the output is in the form of a postscript file. The program relies on control file parameters described below.

Control file parameters

Output plot file: The output postscript file name.

Input data file: The input data file name.

Number of columns in data file: The number of columns in the input data file.

X axis label: The name of the x axis in the graphs.

Y axis label: The name of the y axis in the graphs.

Number of graphs: The number of graphs plotted.

X axis column: The column number in the input data file for the x axis of the graph.

Y axis column: The column number in the input data file for the y axis of the graph.

Graph label: The name of the graph plotted.

B.2.7. Calma-database

Synopsis

```
calma-database [control file]
Control file format :
    [Name of output file]
    [defect start] [defect stop] [defect increment]
    [Number of data files]
    [Name of data file] [Name of data type] [multiplier]
```

Description

Calma-database collates the results of individual data files into a single database. The individual data files consists of two columns such as the defect sizes and critical areas for a given mask layout. It is also capable of labelling the individual critical area columns. The program relies on control file parameters described below.

Control file parameters

Name of output file: The output file name of the database generated.

Defect start: The start size of the defects (μm).

Defect stop: The stop size of the defects (μm).

Defect increment: The increment between defect sizes.

Number of data files: The number of data files that will be collated into the database.

Name of data file: The data file name.

Name of data type: The name of the data obtained from the data file that will be collated in the database.

Multiplier: The value used to divide the critical area for each defect size before collation into the database.

B.3. Miscellaneous Programs

B.3.1. Cif2tl

Synopsis

```
cif2tl [CIF file]
```

Description

Cif2tl converts between the CIF file format used to represent mask layouts and the TL2 format used by Depict. It is only capable on working with a flat layout and any hierarchial layout would have to be flattened before running cif2tl. The output from the program are individual TL2 files for each layer. The only parameter cif2tl accepts is the input CIF file name.

Control file parameters

CIF file: The input CIF file name.

B.3.2. Tl2cif

Synopsis

```
tl2cif [control file]
Control file format :
     [Name of output file]
     [Name of topmost cell]
     [Number of layers]
     [Name of layer file] [Name of layer]
```

Description

Tl2cif converts between the TL2 file format used by Depict and the widely used CIF format used to represent mask layouts. The program relies on control file parameters described below.

Control file parameters

Name of output file: The generated CIF file name.

Name of topmost cell: The topmost cell name in the CIF file.

Number of layers: The number of layers in the CIF file.

Name of layer file: The TL2 file name for the CIF layer.

Name of layer: The layer name used in the CIF file.

B.3.3. Mktrack

Synopsis

```
mktrack [control file]
Control file format :
      [Type of track]
      [Name of output file]
      [Name of cell]
      [Name of layer]
      [Number of tracks]
      [Width of track (micrometers)]
      [Spacing between tracks (micrometers)]
```

Description

Mktrack assists in the generation of parallel or corner tracks of fixed width and separation. The resulting output file is a flattened CIF file. The program relies on control file parameters described below.

Control file parameters

Type of track: The track type generated which is either "parallel" or "corner".

Name of output file: The generated CIF file name.

Name of cell: The cell name in the CIF file.

Name of layer: The layer name used in the CIF file.

Number of tracks: The number of tracks generated.

Width of track (micrometers): The track width in micrometers.

Spacing between tracks (micrometers): The track separation in micrometers.

B.3.4. Scalecif

Synopsis

```
scalecif [control file]
Control file format :
    $Control file for scalecif
    [Name of input file]
    [Name of output file]
    [Size of input technology (micrometers)]
    [Size of output technology (micrometers)]
```

DESCRIPTION

Scalecif assists in the scaling of a CIF file. Using the input and output technology sizes, scalecif modifies the scaling factor of a cell. At present it is only capable of modifying the first cell in a flattened layout. The program relies on control file parameters described below.

CONTROL FILE PARAMETERS

Name of input file: The input CIF file name.

Name of output file: The generated scaled CIF file name.

Size of input technology (micrometers): The technology size for the input CIF file.

Size of output technology (micrometers): The technology size for the scaled CIF file.

C. Further ATOM Results

This appendix presents the results from ATOM on two randomly sampled layouts (layout 1 and layout 2) obtained from the AMD 2901 chip. Figures C.1 and C.2 present the fault probability curves while tables C.1 and C.2 detail the average number of faults and associated errors.

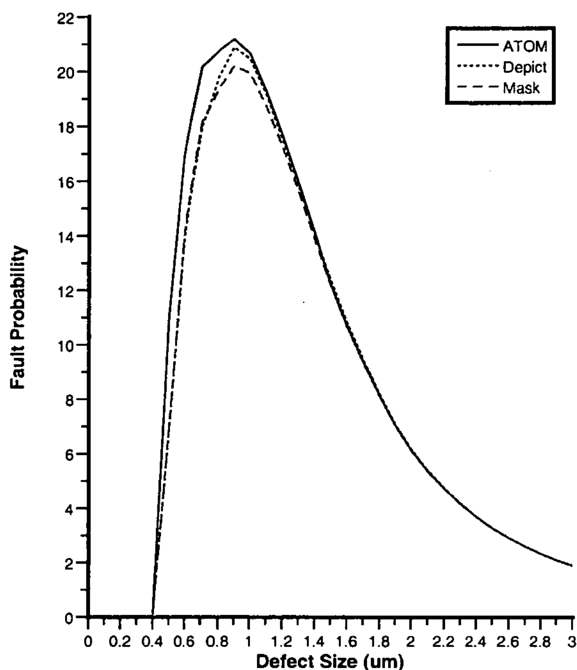


Figure C.1.: Fault probability curve as a function of defect size for layout 1.

	Average number of faults, λ_{av} ($\times 10^{-8}$)	Difference in λ_{av} compared with Depict	Error in λ_{av} compared with Depict (%)
Mask layout	25.0	0.3	1.2
Estimated layout (ATOM)	26.3	1.0	3.9
Aerial Image (Depict)	25.3	-	-

Table C.1.: Average number of faults and the associated errors for layout 1.

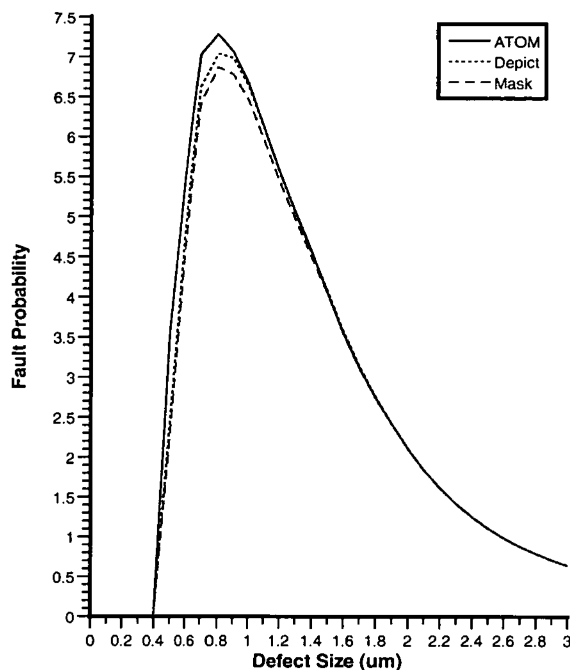


Figure C.2.: Fault probability curve as a function of defect size for layout 2.

	Average number of faults, λ_{av} ($\times 10^{-8}$)	Difference in λ_{av} compared with Depict	Error in λ_{av} compared with Depict (%)
Mask layout	8.29	0.17	2.0
Estimated layout (ATOM)	8.71	0.25	2.9
Aerial Image (Depict)	8.46	-	-

Table C.2.: Average number of faults and the associated errors for layout 2.

D. Full Calma Results

This appendix presents the results obtained from track types 2, 3, 4 and the routing tracks sampled from an AMD 2901 chip. Figures D.1 to D.19 illustrate the critical area error and fault probability curves while tables D.1 to D.19 detail the average number of faults and associated errors.

D.1. Type 2 Tracks

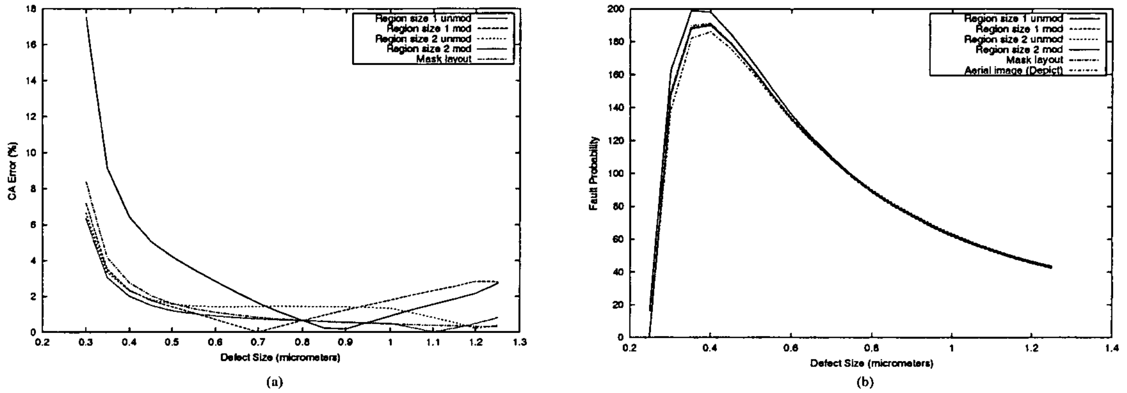


Figure D.1.: Relationships for track type 2-1 as a function of defect size. (a) CA error. (b) Fault probability.

Model	Average number of faults, $\lambda_{0-0.5}$ (defect size $\leq$ 0.5 μm) ($\times 10^{-8}$)	Average number of faults, $\lambda_{0-1.0}$ (defect size $\leq$ 1.0 μm) ($\times 10^{-8}$)	Error in $\lambda_{0-0.5}$ compared with Depict (%)	Error in $\lambda_{0-1.0}$ compared with Depict (%)
Region size 1 unmod	38.05	92.94	11.3	5.6
Region size 1 mod	35.47	89.37	3.7	1.5
Region size 2 unmod	35.36	89.96	3.4	2.2
Region size 2 mod	35.26	89.52	3.2	1.7
Mask layout	35.55	89.89	4.0	2.1
Aerial image (Depict)	34.19	88.01	-	-

Table D.1.: Average number of faults and associated errors for track type 2-1.

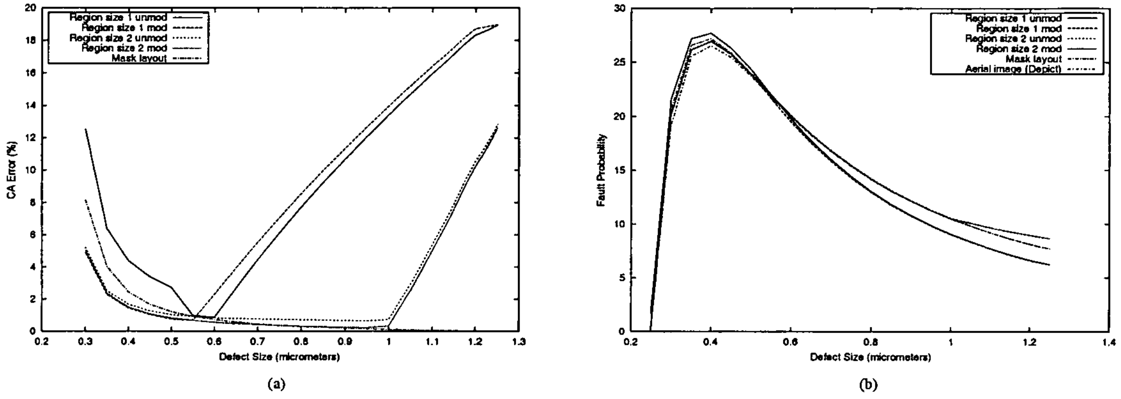


Figure D.2.: Relationships for track type 2-2 as a function of defect size. (a) CA error. (b) Fault probability.

Model	Average number of faults, $\lambda_{0-0.5}$ (defect size $\leq$ 0.5 μm) ($\times 10^{-8}$)	Average number of faults, $\lambda_{0-1.0}$ (defect size $\leq$ 1.0 μm) ($\times 10^{-8}$)	Error in $\lambda_{0-0.5}$ compared with Depict (%)	Error in $\lambda_{0-1.0}$ compared with Depict (%)
Region size 1 unmod	5.23	13.22	8.1	0.5
Region size 1 mod	4.96	12.85	2.5	2.4
Region size 2 unmod	4.97	13.35	2.7	1.4
Region size 2 mod	4.96	13.31	2.5	1.1
Mask layout	5.02	13.39	3.7	1.7
Aerial image (Depict)	4.84	13.16	-	-

Table D.2.: Average number of faults and associated errors for track type 2-2.

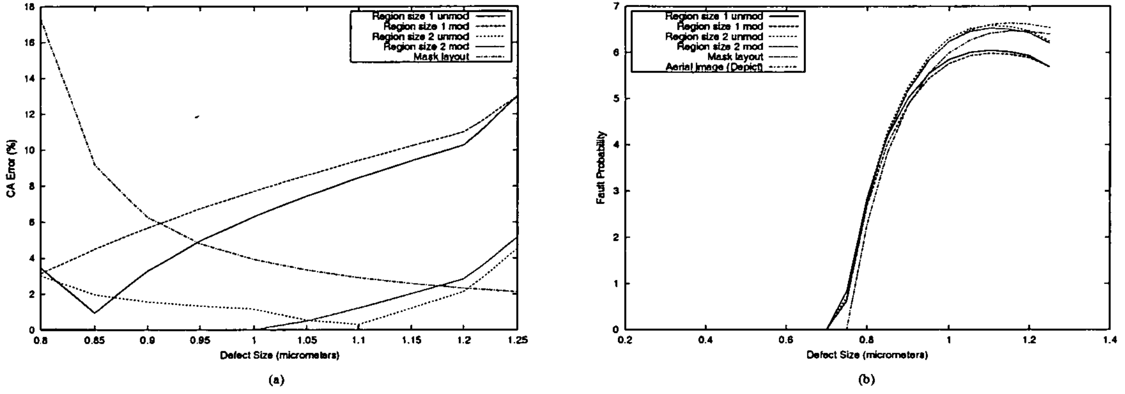


Figure D.3.: Relationships for track type 2-3 as a function of defect size. (a) CA error. (b) Fault probability.

Model	Average number of faults, $\lambda_{0-0.5}$ (defect size $\leq$ 0.5 μm) ($\times 10^{-8}$)	Average number of faults, $\lambda_{0-1.0}$ (defect size $\leq$ 1.0 μm) ($\times 10^{-8}$)	Error in $\lambda_{0-0.5}$ compared with Depict (%)	Error in $\lambda_{0-1.0}$ compared with Depict (%)
Region size 1 unmod	0	0.92	-	1.1
Region size 1 mod	0	0.88	-	5.4
Region size 2 unmod	0	0.95	-	2.2
Region size 2 mod	0	0.93	-	0
Mask layout	0	0.82	-	11.8
Aerial image (Depict)	0	0.93	-	-

Table D.3.: Average number of faults and associated errors for track type 2-3.

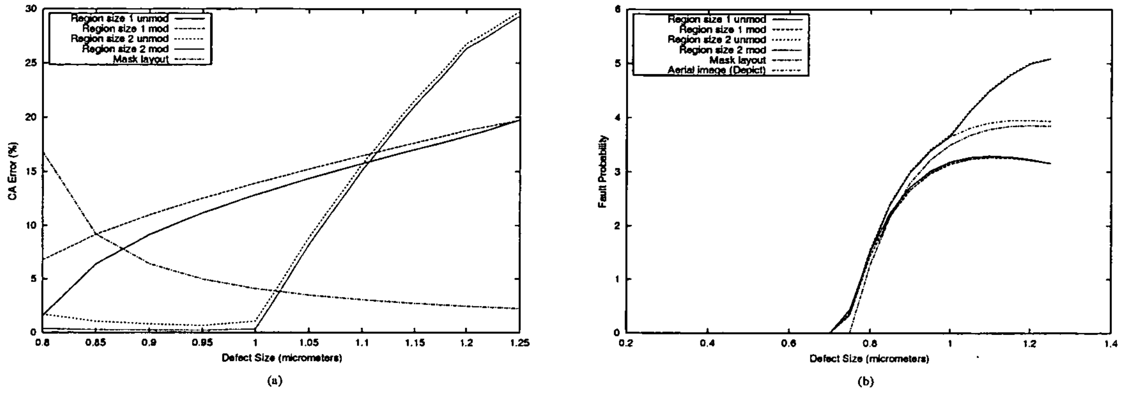


Figure D.4.: Relationships for track type 2-4 as a function of defect size. (a) CA error. (b) Fault probability.

Model	Average number of faults, $\lambda_{0-0.5}$ (defect size $\leq$ 0.5 μm) ($\times 10^{-8}$)	Average number of faults, $\lambda_{0-1.0}$ (defect size $\leq$ 1.0 μm) ($\times 10^{-8}$)	Error in $\lambda_{0-0.5}$ compared with Depict (%)	Error in $\lambda_{0-1.0}$ compared with Depict (%)
Region size 1 unmod	0	0.50	-	5.7
Region size 1 mod	0	0.48	-	9.4
Region size 2 unmod	0	0.54	-	1.9
Region size 2 mod	0	0.53	-	0
Mask layout	0	0.47	-	11.3
Aerial image (Depict)	0	0.53	-	-

Table D.4.: Average number of faults and associated errors for track type 2-4.

D.2. Type 3 Tracks

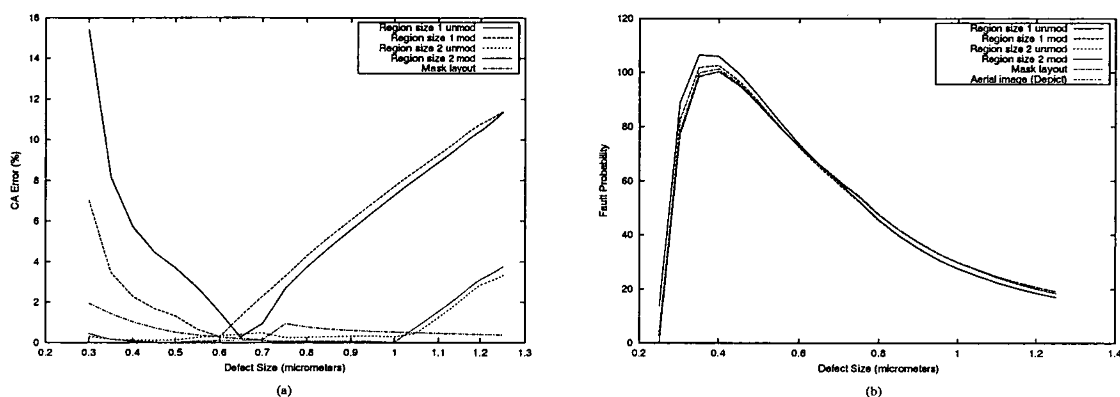


Figure D.5.: Relationships for track type 3-1 as a function of defect size. (a) CA error. (b) Fault probability.

Model	Average number of faults, $\lambda_{0-0.5}$ (defect size $\leq$ 0.5 μm) ($\times 10^{-8}$)	Average number of faults, $\lambda_{0-1.0}$ (defect size $\leq$ 1.0 μm) ($\times 10^{-8}$)	Error in $\lambda_{0-0.5}$ compared with Depict (%)	Error in $\lambda_{0-1.0}$ compared with Depict (%)
Region size 1 unmod	20.71	49.46	11.0	4.0
Region size 1 mod	19.37	47.71	3.8	0.3
Region size 2 unmod	18.70	47.69	0.2	0.3
Region size 2 mod	18.70	47.63	0.2	0.1
Mask layout	18.76	47.81	0.5	0.5
Aerial image (Depict)	18.66	47.57	-	-

Table D.5.: Average number of faults and associated errors for track type 3-1.

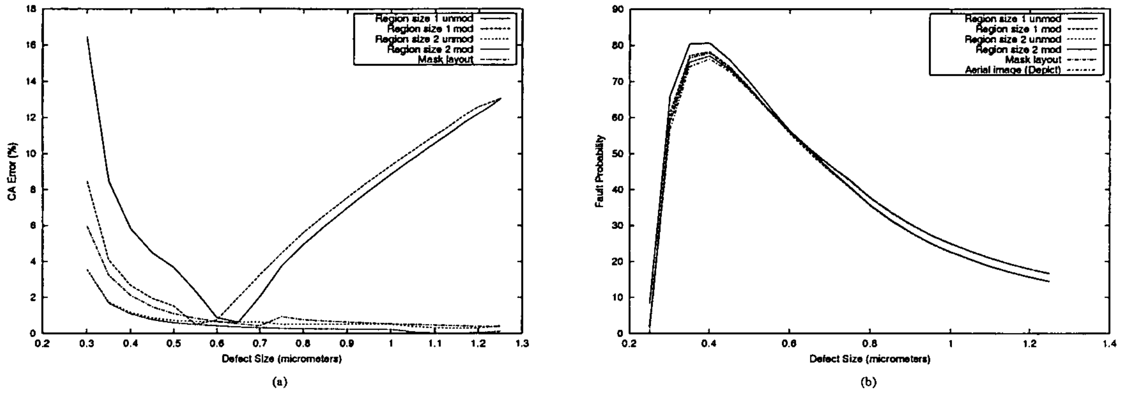


Figure D.6.: Relationships for track type 3-2 as a function of defect size. (a) CA error. (b) Fault probability.

Model	Average number of faults, $\lambda_{0-0.5}$ (defect size $\leq$ 0.5 μm) ($\times 10^{-8}$)	Average number of faults, $\lambda_{0-1.0}$ (defect size $\leq$ 1.0 μm) ($\times 10^{-8}$)	Error in $\lambda_{0-0.5}$ compared with Depict (%)	Error in $\lambda_{0-1.0}$ compared with Depict (%)
Region size 1 unmod	15.56	37.89	10.7	3.2
Region size 1 mod	14.66	36.70	4.3	0.05
Region size 2 unmod	14.32	37.11	1.9	1.1
Region size 2 mod	14.31	37.05	1.8	0.9
Mask layout	14.40	37.23	2.4	1.4
Aerial image (Depict)	14.06	36.72	-	-

Table D.6.: Average number of faults and associated errors fortrack type 3-2.

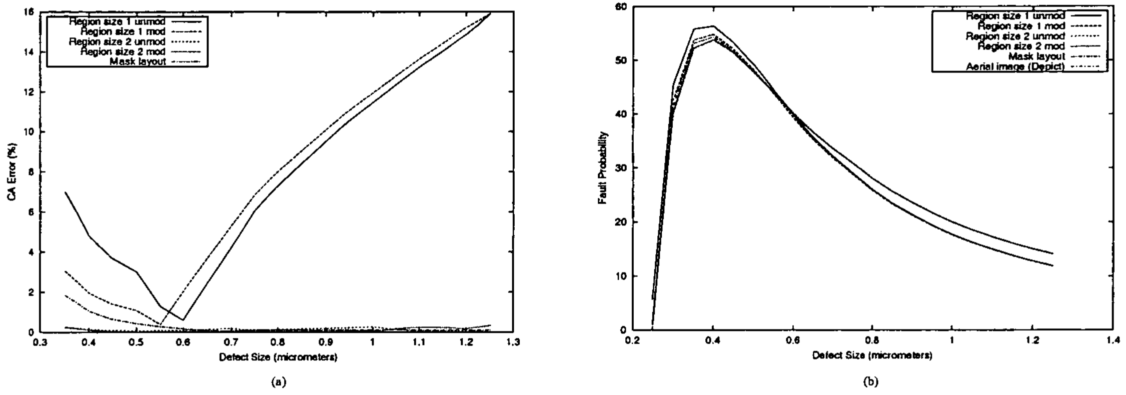


Figure D.7.: Relationships for track type 3-3 as a function of defect size. (a) CA error. (b) Fault probability.

Model	Average number of faults, $\lambda_{0-0.5}$ (defect size $\leq$ 0.5 μm) ($\times 10^{-8}$)	Average number of faults, $\lambda_{0-1.0}$ (defect size $\leq$ 1.0 μm) ($\times 10^{-8}$)	Error in $\lambda_{0-0.5}$ compared with Depict (%)	Error in $\lambda_{0-1.0}$ compared with Depict (%)
Region size 1 unmod	10.84	26.87	9.2	1.3
Region size 1 mod	10.24	26.08	3.1	1.7
Region size 2 unmod	9.96	26.57	0.3	0.2
Region size 2 mod	9.96	26.54	0.3	0.08
Mask layout	10.04	26.65	1.1	0.5
Aerial image (Depict)	9.93	26.52	-	-

Table D.7.: Average number of faults and associated errors for track type 3-3.

D.3. Type 4 Tracks

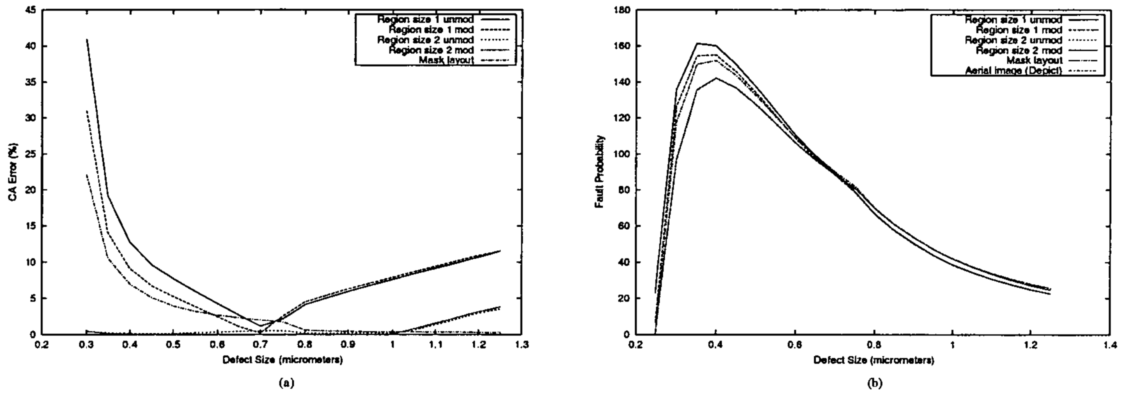


Figure D.8.: Relationships for track type 4-1 as a function of defect size. (a) CA error. (b) Fault probability.

Model	Average number of faults, $\lambda_{0-0.5}$ (defect size $\leq$ 0.5 μm) ($\times 10^{-8}$)	Average number of faults, $\lambda_{0-1.0}$ (defect size $\leq$ 1.0 μm) ($\times 10^{-8}$)	Error in $\lambda_{0-0.5}$ compared with Depict (%)	Error in $\lambda_{0-1.0}$ compared with Depict (%)
Region size 1 unmod	31.50	74.29	23.4	9.6
Region size 1 mod	29.40	71.63	15.2	5.7
Region size 2 unmod	25.58	67.96	0.2	0.3
Region size 2 mod	25.55	67.82	0.1	0.07
Mask layout	28.15	71.30	10.3	5.2
Aerial image (Depict)	25.52	67.77	-	-

Table D.8.: Average number of faults and associated errors for track type 4-1.

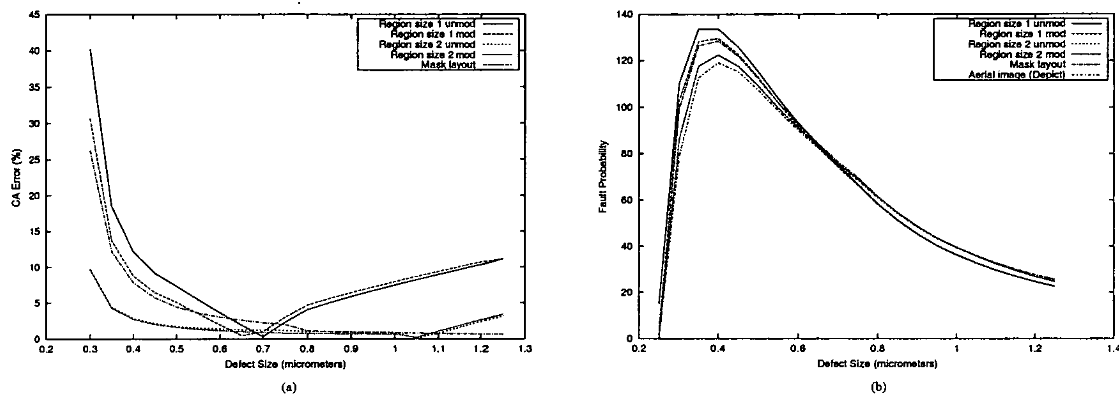


Figure D.9.: Relationships for track type 4-2 as a function of defect size. (a) CA error. (b) Fault probability.

Model	Average number of faults, $\lambda_{0-0.5}$ (defect size $\leq$ 0.5 μm) ($\times 10^{-8}$)	Average number of faults, $\lambda_{0-1.0}$ (defect size $\leq$ 1.0 μm) ($\times 10^{-8}$)	Error in $\lambda_{0-0.5}$ compared with Depict (%)	Error in $\lambda_{0-1.0}$ compared with Depict (%)
Region size 1 unmod	25.89	62.55	21.7	8.5
Region size 1 mod	24.36	60.55	14.5	5.0
Region size 2 unmod	22.26	59.10	4.7	2.5
Region size 2 mod	22.24	58.98	4.6	2.3
Mask layout	23.79	61.07	11.8	6.0
Aerial image (Depict)	21.27	57.64	-	-

Table D.9.: Average number of faults and associated errors for track type 4-2.

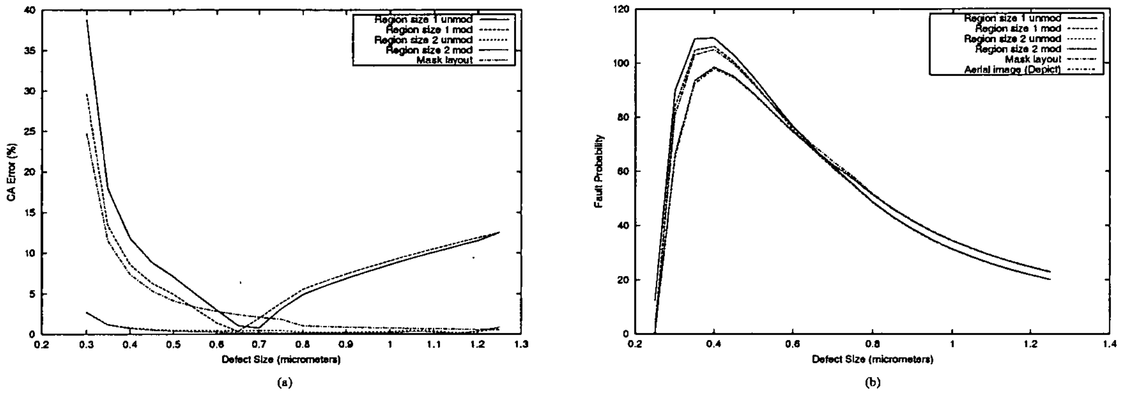


Figure D.10.: Relationships for track type 4-3 as a function of defect size. (a) CA error. (b) Fault probability.

Model	Average number of faults, $\lambda_{0-0.5}$ (defect size $\leq$ 0.5 μm) ($\times 10^{-8}$)	Average number of faults, $\lambda_{0-1.0}$ (defect size $\leq$ 1.0 μm) ($\times 10^{-8}$)	Error in $\lambda_{0-0.5}$ compared with Depict (%)	Error in $\lambda_{0-1.0}$ compared with Depict (%)
Region size 1 unmod	21.17	51.54	21.1	7.7
Region size 1 mod	19.94	49.93	14.1	4.4
Region size 2 unmod	17.71	48.19	1.3	0.7
Region size 2 mod	17.69	48.12	1.2	0.6
Mask layout	19.42	50.49	11.1	5.5
Aerial image (Depict)	17.48	47.84	-	-

Table D.10.: Average number of faults and associated errors for track type 4-3.

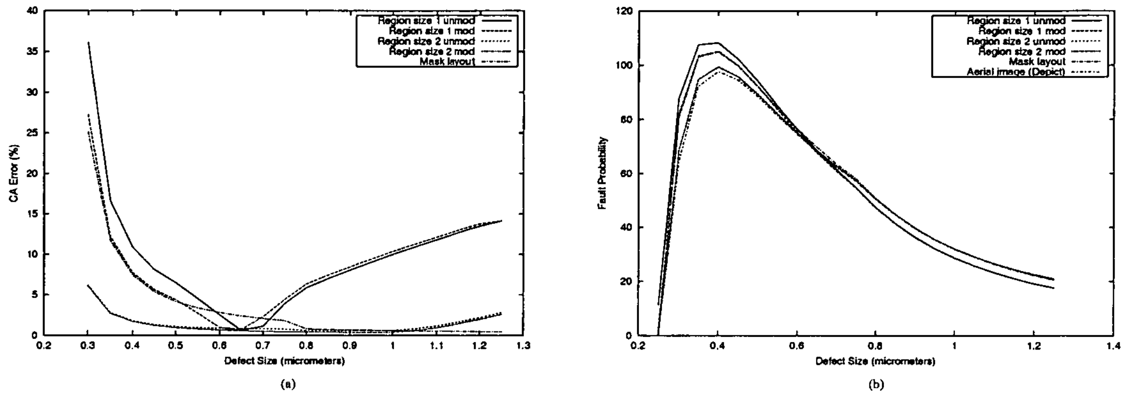


Figure D.11.: Relationships for track type 4-4 as a function of defect size. (a) CA error. (b) Fault probability.

Model	Average number of faults, $\lambda_{0-0.5}$ (defect size $\leq$ 0.5 μm) ($\times 10^{-8}$)	Average number of faults, $\lambda_{0-1.0}$ (defect size $\leq$ 1.0 μm) ($\times 10^{-8}$)	Error in $\lambda_{0-0.5}$ compared with Depict (%)	Error in $\lambda_{0-1.0}$ compared with Depict (%)
Region size 1 unmod	20.86	50.71	19.5	6.8
Region size 1 mod	19.67	49.16	12.7	3.5
Region size 2 unmod	17.98	48.26	3.0	1.6
Region size 2 mod	17.96	48.17	2.9	1.4
Mask layout	19.42	50.14	11.2	5.6
Aerial image (Depict)	17.46	47.50	-	-

Table D.11.: Average number of faults and associated errors for track type 4-4.

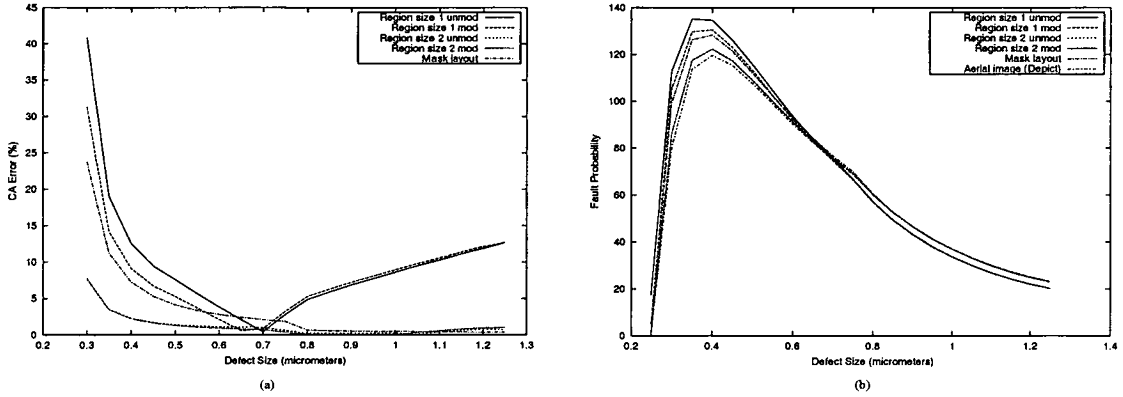


Figure D.12.: Relationships for track type 4-5 as a function of defect size. (a) CA error. (b) Fault probability.

Model	Average number of faults, $\lambda_{0-0.5}$ (defect size $\leq$ 0.5 μm) ($\times 10^{-8}$)	Average number of faults, $\lambda_{0-1.0}$ (defect size $\leq$ 1.0 μm) ($\times 10^{-8}$)	Error in $\lambda_{0-0.5}$ compared with Depict (%)	Error in $\lambda_{0-1.0}$ compared with Depict (%)
Region size 1 unmod	26.34	62.72	22.7	8.9
Region size 1 mod	24.70	60.61	15.1	5.3
Region size 2 unmod	22.27	58.71	3.8	2.0
Region size 2 mod	22.26	58.63	3.7	1.8
Mask layout	23.79	60.72	10.9	5.5
Aerial image (Depict)	21.46	57.58	-	-

Table D.12.: Average number of faults and associated errors for track type 4-5.

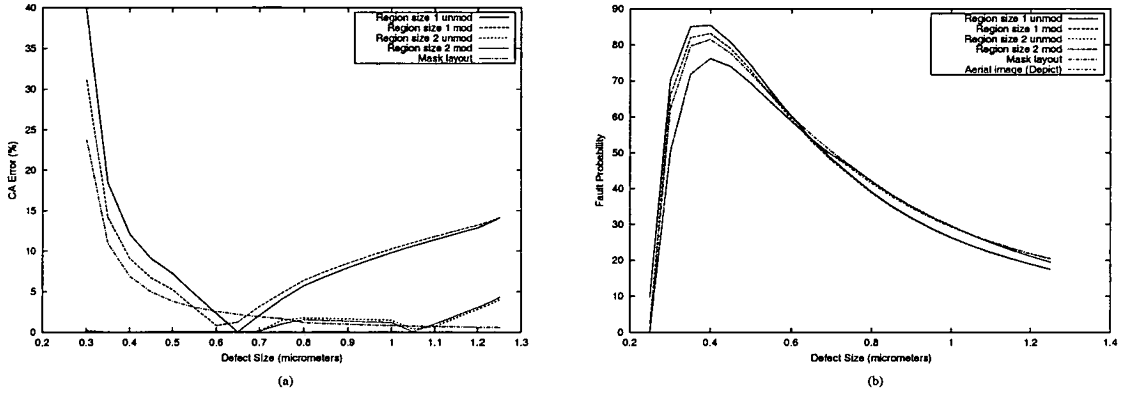


Figure D.13.: Relationships for track type 4-6 as a function of defect size. (a) CA error. (b) Fault probability.

Model	Average number of faults, $\lambda_{0-0.5}$ (defect size $\leq$ 0.5 μm) ($\times 10^{-8}$)	Average number of faults, $\lambda_{0-1.0}$ (defect size $\leq$ 1.0 μm) ($\times 10^{-8}$)	Error in $\lambda_{0-0.5}$ compared with Depict (%)	Error in $\lambda_{0-1.0}$ compared with Depict (%)
Region size 1 unmod	16.57	40.69	21.6	7.2
Region size 1 mod	15.64	39.47	14.7	4.0
Region size 2 unmod	13.64	38.12	0.07	0.4
Region size 2 mod	13.63	38.05	0	0.3
Mask layout	15.06	39.91	10.5	5.2
Aerial image (Depict)	13.63	37.95	-	-

Table D.13.: Average number of faults and associated errors for track type 4-6.

D.4. Routing Track Examples

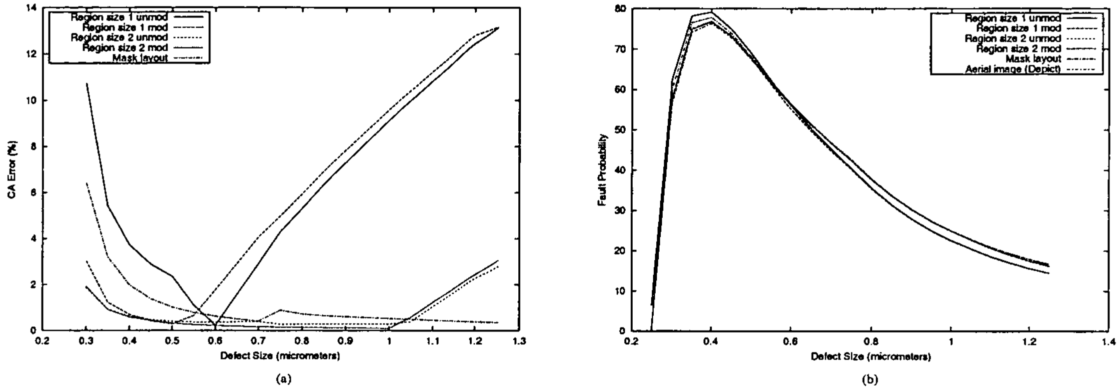


Figure D.14.: Relationships for track type A as a function of defect size. (a) CA error. (b) Fault probability.

Model	Average number of faults, $\lambda_{0-0.5}$ (defect size $\leq$ 0.5 μm) ($\times 10^{-8}$)	Average number of faults, $\lambda_{0-1.0}$ (defect size $\leq$ 1.0 μm) ($\times 10^{-8}$)	Error in $\lambda_{0-0.5}$ compared with Depict (%)	Error in $\lambda_{0-1.0}$ compared with Depict (%)
Region size 1 unmod	15.05	37.19	7.4	1.4
Region size 1 mod	14.18	36.05	1.2	1.7
Region size 2 unmod	14.14	36.88	0.9	0.6
Region size 2 mod	14.13	36.85	0.9	0.5
Mask layout	14.40	37.23	2.8	1.5
Aerial image (Depict)	14.01	36.67	-	-

Table D.14.: Average number of faults and associated errors for track type A.

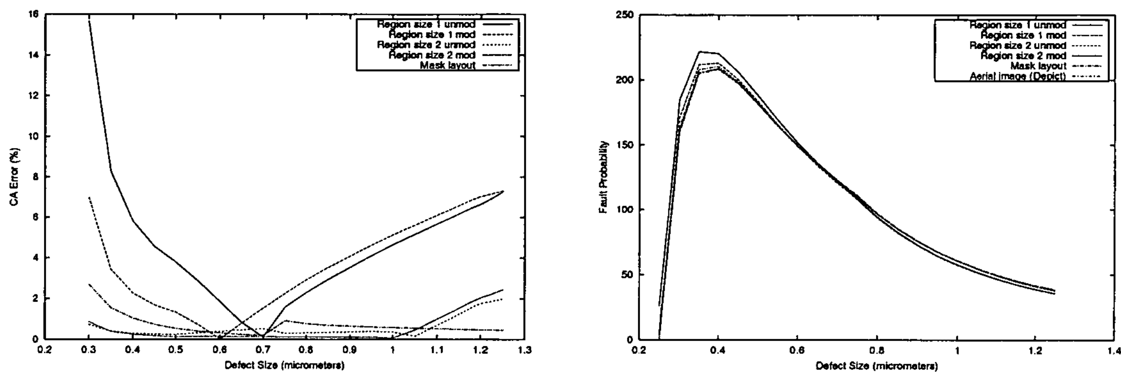


Figure D.15.: Relationships for track type B as a function of defect size. (a) CA error. (b) Fault probability.

Model	Average number of faults, $\lambda_{0-0.5}$ (defect size $\leq$ 0.5 μm) ($\times 10^{-8}$)	Average number of faults, $\lambda_{0-1.0}$ (defect size $\leq$ 1.0 μm) ($\times 10^{-8}$)	Error in $\lambda_{0-0.5}$ compared with Depict (%)	Error in $\lambda_{0-1.0}$ compared with Depict (%)
Region size 1 unmod	42.94	102.39	11.0	4.5
Region size 1 mod	40.08	98.67	3.6	0.7
Region size 2 unmod	38.82	98.31	0.4	0.4
Region size 2 mod	38.81	98.17	0.4	0.2
Mask layout	39.05	98.62	1.0	0.7
Aerial image (Depict)	38.67	97.94	-	-

Table D.15.: Average number of faults and associated errors for track type B.

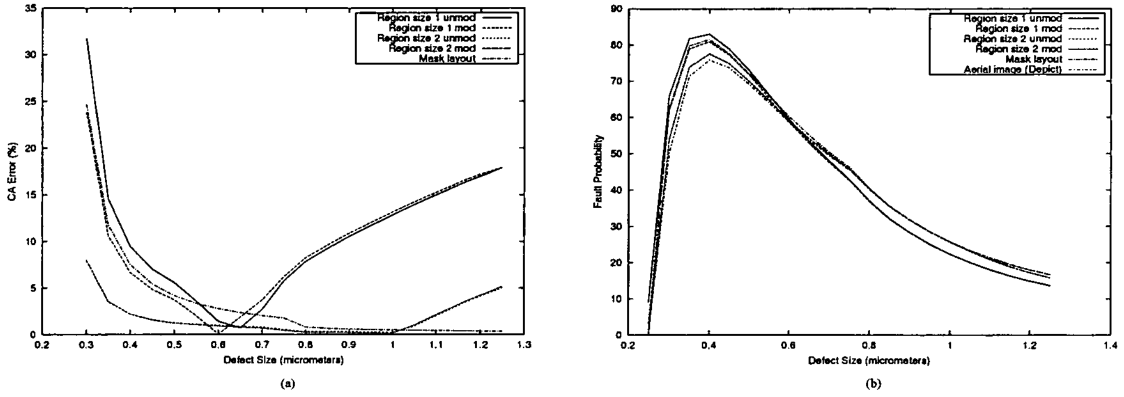


Figure D.16.: Relationships for track type C as a function of defect size. (a) CA error. (b) Fault probability.

Model	Average number of faults, $\lambda_{0-0.5}$ (defect size $\leq$ 0.5 μm) ($\times 10^{-8}$)	Average number of faults, $\lambda_{0-1.0}$ (defect size $\leq$ 1.0 μm) ($\times 10^{-8}$)	Error in $\lambda_{0-0.5}$ compared with Depict (%)	Error in $\lambda_{0-1.0}$ compared with Depict (%)
Region size 1 unmod	15.92	39.19	17.6	5.0
Region size 1 mod	15.10	38.13	11.5	2.1
Region size 2 unmod	14.03	38.00	3.6	1.8
Region size 2 mod	14.03	37.98	3.6	1.7
Mask layout	15.06	39.38	11.2	5.5
Aerial image (Depict)	13.54	37.34	-	-

Table D.16.: Average number of faults and associated errors for track type C.

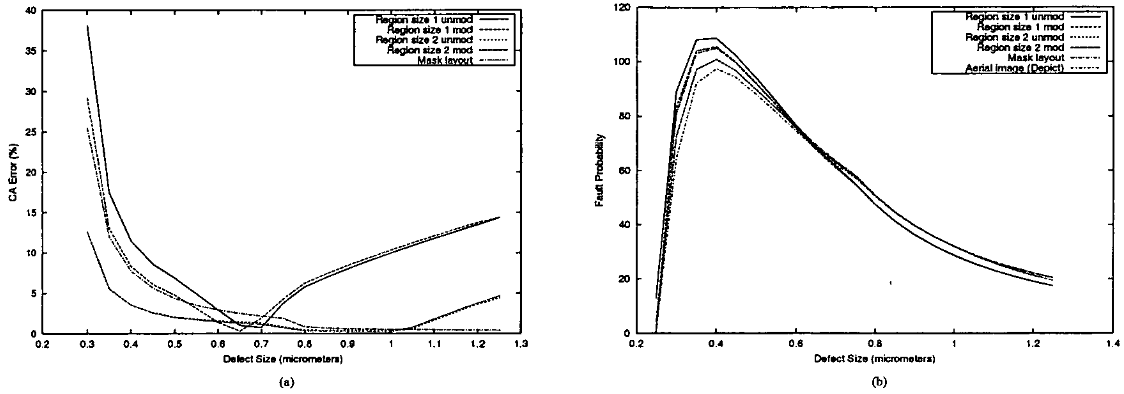


Figure D.17.: Relationships for track type D as a function of defect size. (a) CA error. (b) Fault probability.

Model	Average number of faults, $\lambda_{0-0.5}$ (defect size $\leq$ 0.5 μm) ($\times 10^{-8}$)	Average number of faults, $\lambda_{0-1.0}$ (defect size $\leq$ 1.0 μm) ($\times 10^{-8}$)	Error in $\lambda_{0-0.5}$ compared with Depict (%)	Error in $\lambda_{0-1.0}$ compared with Depict (%)
Region size 1 unmod	21.04	50.95	20.9	7.5
Region size 1 mod	19.85	49.40	14.0	4.2
Region size 2 unmod	18.42	48.80	5.8	2.9
Region size 2 mod	18.42	48.76	5.8	2.8
Mask layout	19.42	50.14	11.5	5.8
Aerial image (Depict)	17.41	47.41	-	-

Table D.17.: Average number of faults and associated errors for track type D.

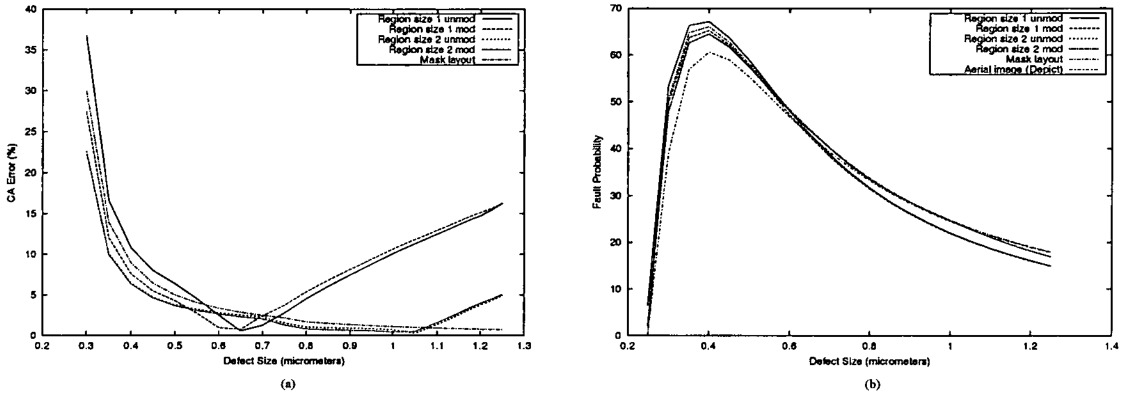


Figure D.18.: Relationships for track type E as a function of defect size. (a) CA error. (b) Fault probability.

Model	Average number of faults, $\lambda_{0-0.5}$ (defect size $\leq$ 0.5 μm) ($\times10^{-8}$)	Average number of faults, $\lambda_{0-1.0}$ (defect size $\leq$ 1.0 μm) ($\times10^{-8}$)	Error in $\lambda_{0-0.5}$ compared with Depict (%)	Error in $\lambda_{0-1.0}$ compared with Depict (%)
Region size 1 unmod	12.86	32.22	19.3	6.5
Region size 1 mod	12.17	31.28	12.9	3.4
Region size 2 unmod	11.94	31.84	10.8	5.3
Region size 2 mod	11.93	31.79	10.7	5.1
Mask layout	12.22	32.24	13.4	6.6
Aerial image (Depict)	10.78	30.24	-	-

Table D.18.: Average number of faults and associated errors for track type E.

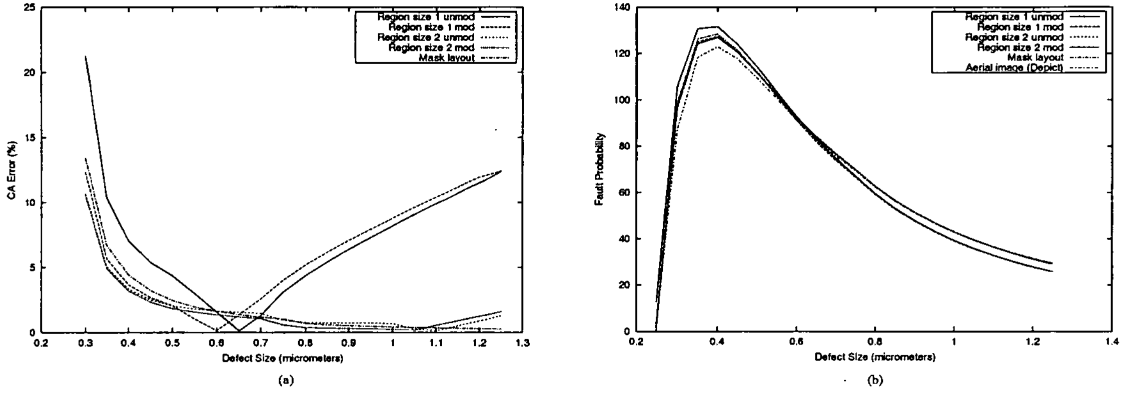


Figure D.19.: Relationships for track type F as a function of defect size. (a) CA error. (b) Fault probability.

Model	Average number of faults, $\lambda_{0-0.5}$ (defect size $\leq$ 0.5 μm) ($\times 10^{-8}$)	Average number of faults, $\lambda_{0-1.0}$ (defect size $\leq$ 1.0 μm) ($\times 10^{-8}$)	Error in $\lambda_{0-0.5}$ compared with Depict (%)	Error in $\lambda_{0-1.0}$ compared with Depict (%)
Region size 1 unmod	25.26	62.21	12.6	4.2
Region size 1 mod	23.77	60.19	6.0	0.9
Region size 2 unmod	23.61	61.37	5.3	2.8
Region size 2 mod	23.58	61.22	5.1	2.6
Mask layout	23.79	61.54	6.1	3.1
Aerial image (Depict)	22.43	59.68	-	-

Table D.19.: Average number of faults and associated errors for track type F.

E. Original Publications

The work described in this thesis has been reported in the following publications:

- [1]M. P. C. Chia and G. A. Allan and A. J. Walton, "Photolithography expert system for improved estimation of IC critical area", *SPIE Symposium on Microelectronic Manufacturing*, Santa Clara, California, pages 74-81, Sep 1998

[1]Reprinted in this appendix

Photolithography expert system for improved estimation of IC critical area

Mark P. C. Chia and Gerard A. Allan and Anthony J. Walton
Department of Electronics and Electrical Engineering
University of Edinburgh
Scotland EH9 3JL, United Kingdom

ABSTRACT

In the manufacturing of Integrated Circuits (IC) the yield of the process is an important factor in estimating the cost. The calculation of critical area plays a key role in helping to determine the expected yield. This paper will demonstrate the use of a system, that will evolve into an expert system which can estimate the affects of photolithography on critical area. The system transforms critical area curves generated from the mask layout to more realistic curves related to the pattern on the wafer. It has been observed that this gives improved critical area estimates.

Keywords: Critical Area, Photolithography, Yield

1. INTRODUCTION

IC technology has come a long way since the vacuum tubes of the 1950s. At present it is possible to put millions of transistors on a single chip and this value is constantly increasing. These advances have been mainly due to better facilities at wafer manufacturing sites, in particular improved lithography and reduction in defectivity. Defects introduced onto a chip as a result of the manufacturing process play an important role in the calculation of IC yield. These can cause circuit failures if they land on a critical point of the chip and produce a fault. However, not every defect causes a chip to fail, as some defects are so small that they have no effect on the functionality of the chip or land in a non-critical area. Thus it is important to classify the critical area on a chip in order to better understand the conditions under which a fault would occur. There are different types of critical area[1] such as extra material, missing material and pinhole critical areas. The critical area of a chip for a circular defect of diameter x is defined as the area in which its center must fall in order to cause a fault. The expected value of critical area, A_c is calculated as follows

$$A_c = \int_0^\infty A(x) f(x) dx ,$$

where $A(x)$ is the critical area for defects of size x and $f(x)$ is the defect size probability density function[2].

Further author information :-
M.P.C.C. (correspondence): Email: Mark.Chia@ee.ed.ac.uk; Telephone: +44-131-6505610
G.A.A.: Email: Gerard.Allan@ee.ed.ac.uk
A.J.W.: Email: Anthony.Walton@ee.ed.ac.uk

The critical area, the number, size, location and distribution of defects, effects the yield of an IC. In formulating a model, the above mentioned factors should be considered. There are a number of models (such as Murphy[3] and Seeds[4]) used to estimate the yield of the manufacturing process. The commonly used models, an exponential yield model and a negative binomial yield model are represented as follows. Using the negative binomial distribution, the yield for a single processing step can be modeled as

$$Y = Y_o(1 + \frac{dA_c}{\alpha})^{-\alpha},$$

where Y is the yield of the die, Y_o the gross yield factor, d the average number of defects per unit area, A_c the critical area and α the clustering parameter. Similarly, using the exponential distribution the yield for a single processing step can be modeled as

$$Y = Y_o e^{-\int_{x_{min}}^{\infty} A_c(x) f(x) dx},$$

where Y is the yield of the die, Y_o the gross yield factor, $A_c(x)$ the critical area for defect sizes x and $f(x)$ the defect density function. For large ICs Stapper[5] observed that the negative binomial yield model more closely matched the actual yield whereas the exponential model tended to be pessimistic[3] in its prediction.

As described above, the calculation of critical area plays a key role in the prediction of yield and hence accurate and realistic calculations are essential. During the manufacturing process, photolithography effects the pattern transferred onto silicon, which in turn effects the critical area. One solution to take into account the affect of photolithography is to run a photolithography simulator on the original mask layout and then run a critical area extraction tool on the modified layout♦. This task can prove time consuming and tedious if different factors in the photolithographic simulation have to be modified. Hence a combination of in-house and commercial software has been used to create an automated system that enables the affects of input parameters of both the photolithography simulator and the critical area extraction tool to be examined. The major elements of this system, known as the Automated Extraction of Photorealistic Critical Area (AEPCA) are:

- Avant! Workbench[6]: An integrated design environment
- Avant! Depict[7]: A photolithographic simulator
- Edinburgh Yield Estimator[8] (EYE): A critical area extraction tool
- Aerial image To Original mask Mapper (ATOM): A photolithography critical area mapping tool

The system (AEPCA) developed has been built around Workbench (TWB) which uses modules to create a process recipe for an experiment. Thus by designing modules for Depict, EYE and ATOM it is possible to integrate these tools within TWB. Currently there are two methods within AEPCA capable of extracting the photorealistic critical area of ICs, one is by using Depict and the other by ATOM. The method using Depict calculates a simulated photorealistic aerial image from the mask layout but is not feasible for large ICs with thousands of transistors. Hence, one solution is to map

♦ Some critical area extraction tools only work on Manhattan layouts and hence can not be used.

the critical area curves generated from the original mask layout to more realistic curves related to the pattern on the wafer and one such tool capable of this is ATOM which will be described in Sect. 2.

This paper will discuss the integration of various commercial tools and in-house software to extract the critical area from a modified photolithographic pattern. It will also present an alternative method to approximate the critical area curve for larger layouts using ATOM. Section 2 will briefly describe the various software tools used and explain how these can be integrated within AEPCA. The section will also describe ATOM and will use an SRAM to gauge ATOM effectiveness as compared to Depict. Next, Sect. 3 will analyse the results obtained and compare the new critical area curves generated from ATOM with the critical area curves generated from Depict. Finally Sect. 4 will conclude the paper and outline future enchantments to ATOM.

2. TOOLS INVOLVED AND IMPLEMENTATION

This section will discuss the tools involved in building AEPCA. It will also describe the control factors used for the SRAM cell when comparing the accuracy of ATOM with Depict.

2.1 AEPCA

2.1.1 Depict

Depict is a photolithography simulator tool available from Avant!. Its aerial image module is capable of performing photolithographic simulations of an IC by taking as input the original mask layout and calculating the two dimensional aerial image transferred onto the wafer. An example of this process can be demonstrated by using an SRAM cell. The original mask layout of an SRAM cell is illustrated in Fig. 1(a) and the simulated aerial image presented in Fig. 1(b) for lambda λ = 365 nm. The difference between the two is quite significant and hence the effect of a defect landing on the two layouts will not be the same.

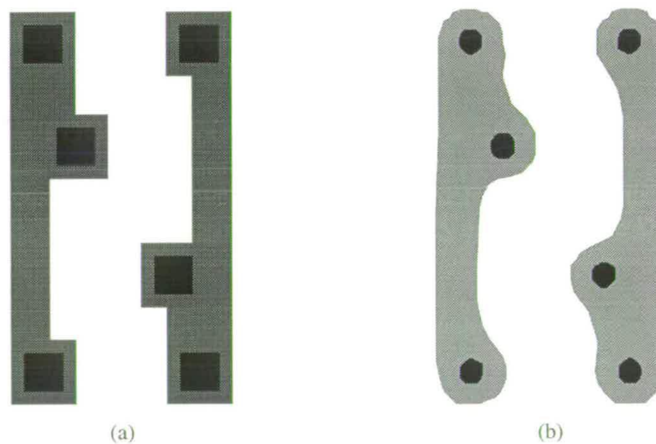
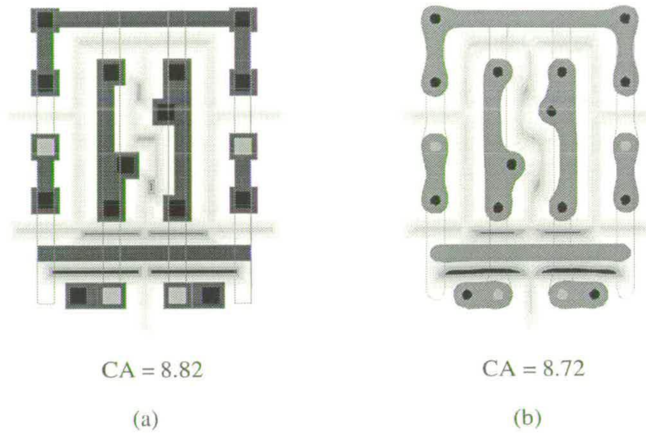


Figure 1. (a) Original mask layout and (b) simulated mask layout for a zoomed in view of the SRAM cell with λ = 365 nm.

2.1.2 EYE

EYE can be employed for the extraction of critical area and is capable of processing mask layouts written in either CIF or GDS II formats. Figure 2 shows a Fault Probability Map (FPM) generated for the SRAM cell shown in Fig. 1 with darker regions indicating where defects have the highest probability of causing a fault. It can be observed that there is a difference in both the FPM and effective critical areas extracted from the two layouts.



CA = 8.82 (a) CA = 8.72 (b)

Figure 2. FPM generated using EYE for Metal 1 for an SRAM cell. (a) Using the original mask layout and (b) aerial image simulated from Depict.

2.1.3 ATOM

ATOM is capable of mapping the critical area curve of an IC to more realistic curves based on photolithography. It uses as input a control file which specifies the data, input and output file names. The input file is the critical area curve of the Device Under Test (DUT). The other two data files are the test structures (in this case, parallel lines of varying spacing) before and after photolithography.

The program creates the CA table from the input file and the photorealistic table from the data file of the photorealistic test structures. The CA table is formed by mapping the critical area curve of the DUT onto equivalent critical area curves of parallel lines. It is similar to the Virtual Mask[9] technique used in calculating the critical area of ICs by replacing the original layout by a simplified layout. In this case, the simplified layout is a combination of parallel lines with different lengths and line spacings. The photorealistic table is obtained by forming cubic splines through the critical area curves. Next, the mapping table is formed from the combination of the CA table and the data file of the original mask layout. Finally, the photorealistic table and mapping table are combined to form the final output file which is the estimated photorealistic critical area curve. Figure 3 illustrates schematically the creation of the different tables formed within ATOM before calculating the final critical area curve.

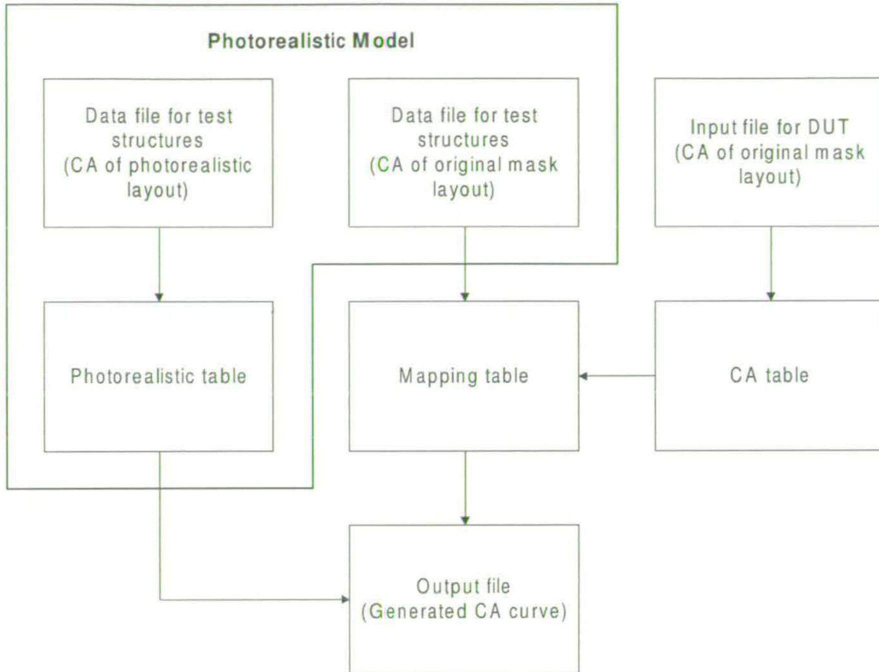


Figure 3. Schematic diagram of ATOM outlining the generation of the different tables and output file from the input critical area curves.

2.1.4 Workbench

Workbench can be used to integrate Depict, EYE and ATOM to automate the system for extracting critical area from a photolithographic simulation of a mask layout. In addition, TWB allows experiments to be designed which can be used to investigate the affects of different input parameters on critical area. Figure 4 shows the modules used for the extraction of critical area and the experiment tree built by TWB. In this case the minimum spacing (feature size) was the parameter being varied. Figure 5 shows the run table used to create the experiment shown in Fig. 4. The ticks indicate the parameters that have been varied in the simulations.

Modules

- **Scalecif** : Automatically scales the CIF file to various technology sizes.
- **Cif2tl** : Reads in the CIF file and breaks up the various layers to a format that Depict understands.
- **Depict** : Runs the Depict simulator.
- **Tl2cif** : Merges the various layers of the photolithographic process back into a single CIF file.
- **Eye** : Runs the EYE tool.

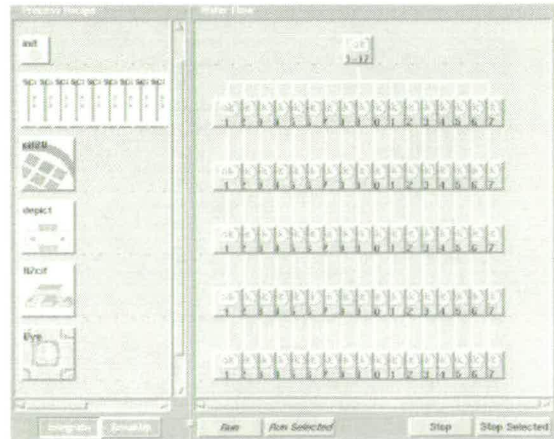


Figure 4. Module layout and experiment tree generated for an experiment.

Figure 5. Run table in TWB used to modify the control factors in the experiment.

2.2 Implementation

An SRAM cell was used to test the functionality of AEPCA and more importantly the accuracy of ATOM. The SRAM cell has a small mask layout which enables Depict to calculate the critical area within a reasonable amount of time. For the experiment, an optical stepper wavelength, $\lambda = 365$ nm and numerical aperture, NA = 0.6 was used. The feature size of the SRAM cell was 0.5 μm, which matches the test structures used to generate the photorealistic critical area curves. The defect size was taken from 0.1-3.0 μm at 0.1 μm intervals.

3. RESULTS

The results obtained from ATOM proved very encouraging. It was observed that the critical area curve generated from ATOM and consequently the fault probability curve provided a good estimate to the critical area curve obtained from Depict. The critical area curves can be observed in Fig. 6(a) and a zoomed in view for defect sizes of 0.40-0.75 μm in Fig. 6(b). The close nature of the critical area curves makes it difficult to analyse the effects this has on the yield of the SRAM cell. Hence, in order to outline these differences, the fault probability curve was generated. This curve is the multiplication of the critical area curve with the defect distribution curve which has been taken as x^{-3}, where x is the defect size. The fault probability curves can be seen in Fig. 7(a) and the zoomed in view for defect sizes of 0.60-1.80 μm in Fig 7(b). From Fig. 7, it can be observed that between defect sizes of 0.80-1.80 μm, ATOM underestimates the photolithographic process, whereas between 0.50-0.80 μm it overestimates the critical area. This limitation is a result of approximating a complex mask layout using simple structures (parallel lines in this case). The total area under the fault probability curves can be seen in Tab. 1 and it is interesting to note that the area under the curves generated by ATOM and Depict are within 0.5 % of each other.

In addition, the cumulative sum of the photorealistic data curves for varying line spacings was used to form the critical area curve for ATOM. In general, it should be noted that only a subset of these curves are used for a particular IC layout. The cumulative* affect of these curves in estimating the critical area curve of the SRAM cell is illustrated in Fig. 8(a) and a zoomed in view for defect sizes of 0.40-1.00 μm in Fig. 8(b). It can be observed that as more photorealistic curves are added together, the cumulative critical area curve more closely resembles the critical area curve in Fig. 6.

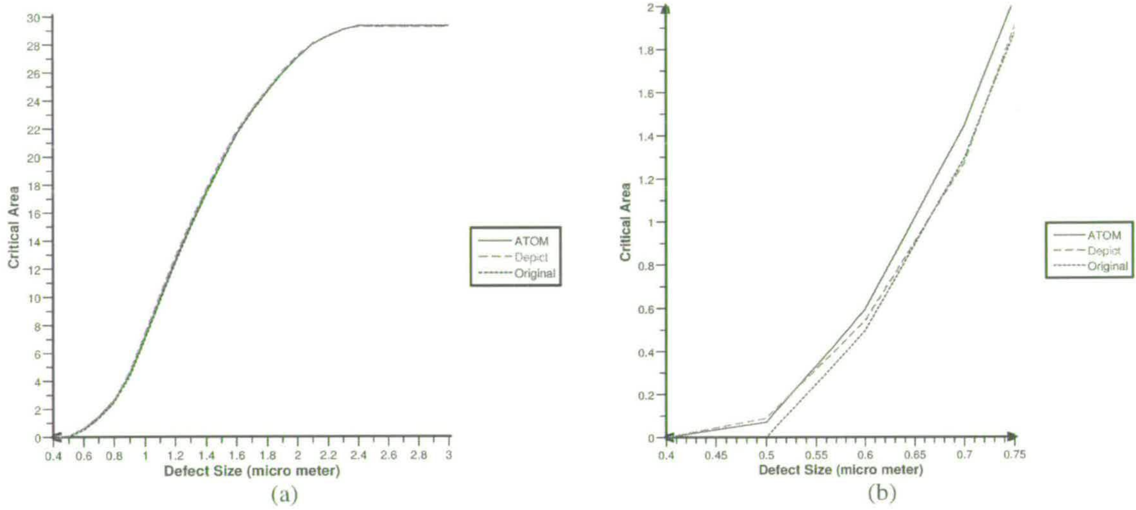


Figure 6. Critical area as a function of defect size for the original mask layout, Depict and ATOM. (a) Normal view and (b) zoomed in view.

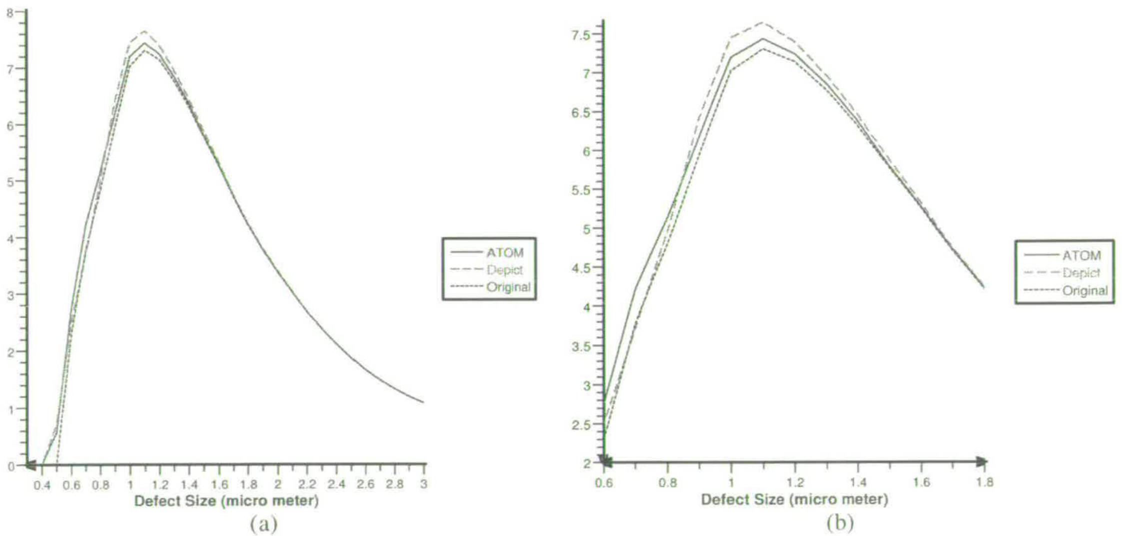


Figure 7. Fault probability curve as a function of defect size for the original mask layout, Depict and ATOM. (a) Normal view and (b) zoomed in view.

* The curve for a spacing = 0.6 μm includes the spline calculated for 0.5 μm and that for 0.6 μm.

Table 1. Calculation of area under the fault probability curves and the associated error.

	Area	Difference	Percentage Error (%)
Depict	9.99	-	-
ATOM	9.94	0.05	0.50
Original	9.68	0.31	3.10

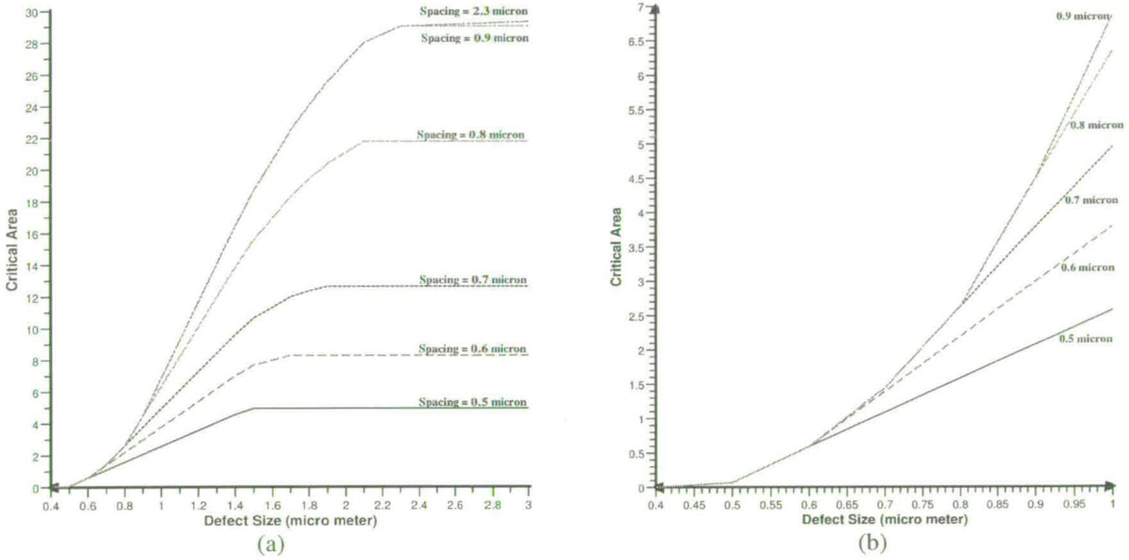


Figure 8. Cumulative effect of the different splines used in the creation of the CA curve generated for the SRAM cell. (a) Normal view and (b) zoomed in view.

4. CONCLUSION

This paper has demonstrated the use of Workbench to automate the extraction of photorealistic critical area for different mask layouts. The use of Depict is limited to small ICs due to resource limitations in larger layouts. ATOM has been found to be capable of estimating the photolithographic process by using an SRAM cell as an example. Hence, by employing ATOM with the Edinburgh Yield Estimator – Sampling[10] (EYES), it can be used to calculate the critical area of large ICs. One limitation of ATOM is that estimates are based on simple test structures. Hence to increase accuracy, more detailed input and data files have to be utilised. Currently, ATOM has been tested by using a fixed track width of 0.5 μm with variable spacings. It is envisaged that in future, ATOM will not be limited to fixed track widths and will be able to generate critical area curves for different feature sizes and optical stepper wavelengths.

REFERENCES

1. 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 Academic Publishers, Boston;Lancaster, 1987.
2. C. H. Stapper. Modeling of defects in integrated circuit photolithographic patterns. *IBM J. Res. Develop.*, 28(4):461-474, July 1984.
3. B. T. Murphy. Cost-size optima of monolithic integrated circuits. *Proc. IEEE.*, 52, Dec 1964.
4. R. B. Seeds. Yield, economic and logistic models for complex digital arrays. *IEEE International Convention Record*, pages 60-61, March 1967.
5. C. H. Stapper. Modeling of integrated circuit defect sensitivities. *IBM J. Res. Develop.*, 27(6):549-557, Nov 1983.
6. *TMA Workbench User's Manual*. Technology Modeling Associates, Inc, 3950 Fabian Way, Palo Alto, CA 94303-4605, 1997.
7. *DEPICT Reference Manual*. Technology Modeling Associates, Inc, 3950 Fabian Way, Palo Alto, CA 94303-4605, 1997.
8. G. A. Allan. *EYE: User Manual*. University of Edinburgh, Electronics and Electrical Engineering Dept., Kings Buildings, Edinburgh EH9 3JL Scotland, UK, 1995.
9. 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.
10. G. A. Allan and A. J. Walton. Sampling based yield prediction for ULSI. *In SPIE Symposium on Microelectronic Manufacture*, pages 198-209, Austin, Texas, Oct 1996.