UNIVERSITY OF SOUTHAMPTON

DEPARTMENT OF ELECTRONICS AND COMPUTER SCIENCE

Electron Beam Proximity Effect Correction

Chee Seng EA

January, 2000

A thesis submitted for the title of
Doctor of Philosophy.

UNIVERSITY OF SOUTHAMPTON

Electron Beam Proximity Effect Correction

by

Chee Seng EA

A thesis submitted for the degree of
Doctor of Philosophy.

Department of Electronics and Computer Science,
University of Southampton

January, 2000

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING AND APPLIED SCIENCE

DEPARTMENT OF ELECTRONICS AND COMPUTER SCIENCE

Doctor of Philosophy

Electron Beam Proximity Effect Correction

by Chee Seng EA

Scattering of incident electrons in a resist-substrate structure causes undesired energy to be deposited in the area surrounding a written shape. Any shape that falls within this area can suffer significant variation from its intended dimensions in the developed resist image. This effect is well known as the *proximity effect.* To correct for the effect, the written pattern is modified in a way that the developed resist image is closer to its desired shape. This modification is known *as proximity effect correction* (PEC), and is usually carried out as a software based pre-processing step before mask production.

The principal goal of this work is to produce an electron beam PEC system that is faster, more accurate, and requires less computational resources than those currently available.

The correction system described in this thesis is derived from the pattern area density proximity effect correction (PADPEC) method of Murai *et al.* [F. Murai, H.Yoda, S. Okazaki, N. Saitou and Y. Sakitani, Journal Of Vacuum Science & Technology **B10**, 3072(1992)], one of the fastest published algorithms. Each individual source of error in the processing chain is analysed and attacked, and a number of algorithmic enhancements introduced (corner rounding error reduction and shape fracturing); the outcome is a significant improvement in accuracy and data throughput of the process. The new system is called the Enhanced PADPEC, or EPADPEC.

Compared to PADPEC, EPADPEC reduces line width errors by factors ranging from 4 to 11, at a cost of doubling the calculation time. However, for the *same* accuracy requirement, EPADPEC is around five times faster than PADPEC.

In parallel with the development of the EPADPEC system, because experimental work on an electron beam lithography (EBL) machine is costly and slow, a suite of support programs was developed: a GDSII browser, a shape processor, a proximity effect simulator and an EBL machine write time simulator. These allowed the bulk of the experimental and development work to be entirely software based; physical experimentation is used only to calibrate and validate the simulator, and to verify a handful of key results.

Finally, the thesis presents a feasibility study of the idea of implementing proximity correction as an embedded hardware/software system, to be integrated into the EBL machine. The design goal is to accelerate the correction calculation to the point that it is no longer rate limiting and can be performed in-line. The study suggests that such an approach is likely to succeed.

Contents

Contents 3

Chapter 1 Introduction 17

1.1 Objectives 19

1.2 Thesis organisation 20

Chapter 2 The architecture and application of an electron beam lithography (EBL) system 21

2.1 Overview of an EBL system 22

2.2 Electron beam writing strategies. 23

2.3 Applications 25

2.4 Potentially rival technologies 29

Chapter 3 Electron beam proximity effect 32

3.1 The nature of the proximity effect 33

3.2 Profile modelling 35

3.2.1 Inaccuracies in the Gaussian model EDF 38

3.2.2 Determination of the EDF parameters 41

3.3 Proximity effect avoidance 43

Chapter 4 Proximity effect correction (PEC) 45

4.1 Overview of PEC techniques 46

4.2 GHOST (A dose equalisation scheme for proximity effect correction) 47

4.2.1 Performance indicators of GHOST 49

4.2.2 Advantages and disadvantages of GHOST 50

4.3 Shape correction 51

4.3.1 Performance indicators of PYRAMID 53

4.3.2 Advantages and disadvantages of PYRAMID 54

4.4 Dose correction 54

4.4.1 Self-consistent method 55

4.4.1.1 Performance indicators of the self-consistent method 56

4.4.1.2 Advantages and disadvantages of the self-consistent method 58

4.4.2 The hierarchical approach 59
4.4.2.1 Performance indicators of the hierarchical approach 59
4.4.2.2 Advantages and disadvantages of computer aided proximity correction (CAPROX) 62
4.4.3 Optimisation approach 62
4.4.3.1 Performance indicators of the optimisation approach 64
4.4.3.2 Advantages and disadvantages of the optimisation approach 65
4.4.4 The Fourier pre-compensation (de-convolution) proximity effect correction (FPPEC) method 66
4.4.4.1 Performance indicators of FPPEC 67
4.4.4.2 Advantages and disadvantages of FPPEC 68
4.4.5 Neural network approach 69
4.4.5.1 Performance indicators of the neural network approach 69
4.4.5.2 Advantages and disadvantages of the neural network approach 70
4.4.6 The representative figure proximity effect correction (RFPEC) method 70
4.4.6.1 Performance indicators of RFPEC 72
4.4.6.2 Advantages and disadvantages of RFPEC 75
4.4.7 The pattern area density proximity effect correction(PADPEC) method 75
4.4.7.1 Derivation of the optimum incident dose formula 78
4.4.7.2 Performance indicators of PADPEC 79
4.4.7.3 Advantages and disadvantages of PADPEC 80
4.5 Comparison of various PEC schemes 81
4.6 Rationale for selecting PADPEC for improvement in this research 88

Chapter 5 The enhanced pattern area density proximity effect correction algorithm 89

5.1 Sources of correction error in the PADPEC technique. 90
5.1.1 Interpolation 91
5.1.2 Assuming a constant incident dose per shape 92
5.1.3 Smoothing 95
5.1.4 Neglect of forward scattering 97
5.1.5 Incident dose formula 98
5.1.6 Conclusions 101
5.2 Overview of differences between the EPADPEC and PADPEC schemes 102

5.2.1 Theoretical background 104
5.2.1.1 The iterative algorithm 104
5.2.1.2 Framing 107
5.2.1.3 Interpolation and quantization strategies 108
5.2.2 Algorithm implementation 112
5.2.2.1 Modifications to the software version of EPADPEC for a Gaussian spot, step and write EBL machine 114

Chapter 6 Algorithm evaluation 117

6.1 Test pattern selection and PEC settings 118
6.2 Correction accuracy 122
6.2.1 Figures of merit 122
6.2.2 Comparison 123
6.2.2.1 Simulation results 123
6.2.2.2 Experimental results 136
6.3 Timing 140
6.3.1 Method of time measurement 141
6.3.2 Comparison 142
6.4 Effect of quantizing the incident dose 147
6.5 Conclusions 149

Chapter 7 Further improving the EPADPEC algorithm 152

7.1 Corner correction 153
7.1.1 Survey of existing corner correction algorithms 153
7.1.2 Description of the corner correction algorithm 154
7.1.3 Algorithm implementation 157
7.1.4 Estimation of the optimum correction parameter values 158
7.1.5 Computation overhead and correction accuracy improvement of the corner correction scheme 162
7.1.5.1 Conclusions 166
7.2 Shape fracturing 167
7.2.1 Survey of existing shape fracturing algorithms 167
7.2.2 Description and implementation of the physical fracturing algorithm (PFA) . 169

7.2.3 Speed and correction accuracy comparison between PFA and the sub-field fracturing algorithm (SFA) 171

7.2.4 Conclusions 175

Chapter 8 Feasibility study of performing EPADPEC in real time 176

8.1 Bottlenecks in the data flow of the existing EBL system 177

8.2 Computational time requirement projection of EPADPEC for real time correction on future semiconductor devices using future EBL machines 180

8.3 Ways to achieve the required speed up factor 184

8.3.1 Faster computers 185

8.3.2 Dedicated hardware 186

8.4 Conclusions 192

Chapter 9 Conclusions 193

9.1 Contributions 194

9.2 Further work 194

9.2.1 Hardware/software co-design 194

9.2.2 Correction of small shapes whose width is smaller than the frame width (4α) 195

9.2.3 Extension for non-Manhattan shapes 195

9.2.4 Adaptation of EPADPEC to projection type EBL machines 195

9.2.5 Formatter for other EBL machine formats 196

9.2.6 Resist calibration 196

9.2.7 Cost and deliverables 196

Appendix A Publication 199

Appendix B The electron beam data preparation system (EDAPS) 211

B.1 The PEC module 213

B.2 GDSII layout browser 215

B.2.1 Algorithm implementation 215

B.3 Proximity effect simulator 217

B.3.1 Theoretical background 218

B.3.2 Algorithm implementation 219
B.3.3 Calibration using scanning electron microscope photographs of developed resist image 221
B.4 EBL machine write time simulator 223

Appendix C The user manual of the electron beam data preparation system (EDAPS) 225

C.1 Introduction 226
C.1.1 Conventions used in this manual 229
C.1.2 Getting started 229
C.2 Standard menu for various viewers 230
C.2.1 The File menu 231
C.2.1.1 The Open command 231
C.2.1.2 The Close command 231
C.2.1.3 The Save As command 231
C.2.1.4 The Print command 231
C.2.1.5 The Print Preview command 232
C.2.1.6 The Print Setup command 233
C.2.1.7 The Exit command 234
C.2.2 The View menu 234
C.2.2.1 The Toolbar command 234
C.2.2.2 The Status Bar command 235
C.2.2.3 The Aerial Viewer command 235
C.2.2.4 The Grid command 235
C.2.2.5 The Zoom By Area command 235
C.2.3 The Settings menu 235
C.2.3.1 The Grid Size property page 235
C.2.3.2 The Mapping Mode property page 236
C.2.4 The Window menu 237
C.2.4.1 The New Window command 237
C.2.4.2 The Cascade command 237
C.2.4.3 The Tile command 237
C.2.4.4 The Arrange Icons command 237
C.3 GDSII layout browser 237

C.3.1 The View menu ... 239
C.3.1.1 The Fill Drawing Objects command ... 239
C.3.1.2 The Select Layers' Visibility command ... 239
C.3.2 The Settings menu ... 240
C.3.2.1 The EBL property page ... 240
C.3.2.2 The EDF property page ... 243
C.3.3 The Tools menu ... 244
C.4 The shape processor viewer ... 246
C.4.1 The Tools menu ... 246
C.4.1.1 The PADPEC command ... 248
C.4.1.2 The No PEC command ... 251
C.5 The exposure pattern viewer ... 251
C.5.1 The View menu ... 252
C.5.2 The Tools menu ... 252
C.5.2.1 The Exposure Simulator command ... 253
C.5.2.2 The Threshold Developer Simulator command ... 254
C.5.2.3 The Resist Characteristic Curve Developer Simulator command ... 255
C.5.2.4 The EBL Machine Write Time Simulator command ... 256
C.5.3 The Formatter menu ... 256
C.6 Viewers of the proximity effect simulator ... 256
C.6.1 The View menu ... 256
C.6.2 The exposure simulator viewer ... 257
C.6.2.1 The Display Settings menu ... 258
C.6.3 The threshold developer simulator viewer ... 259
C.6.3.1 The Tool menu ... 259
C.6.4 The resist characteristic curve developer simulator viewer ... 260

Appendix D Calculation of the spot radius and the peak charge density of the correction beam used in GHOST, for an energy density function (EDF) with n Gaussian terms ... 261

Appendix E Proof of desirable mathematical features of a Gaussian function for PEC ... 263

E.1 Separable property ... 263

E.2 Normalization of a Gaussian function 264
E.3 Integration 265
E.4 Convolution with an infinite line or rectangle 266
E.5 Convolution with another Gaussian function 268

Glossary and acronyms 271

References 274

List of Figures

Figure 1-1 Image projection system in optical lithography. 17

Figure 2-1 A simplified block diagram of an EBL system. 22

Figure 2-2 Formation of shape primitives (shaded shapes) in a VSB system. 24

Figure 2-3 Interference with a phase shift mask. 28

Figure 2-4 Image projection system in X-ray proximity lithography. 29

Figure 2-5 Image projection system in extreme ultraviolet lithography. 31

Figure 3-1 Monte Carlo electron trajectories. 33

Figure 3-2 Simulated and experimental line spread EDDs. 36

Figure 3-3 Experimental and analytical point spread EDDs. 39

Figure 3-4 Comparison between the double Gaussian and the improved EDF. 40

Figure 4-1 General technique of GHOST. 48

Figure 4-2 Variation of efficiency with the number of workstations. 53

Figure 4-3 Division of a chip area into smaller areas. 57

Figure 4-4 Line width variation versus incident dose. 61

Figure 4-5 Tower pattern. 65

Figure 4-6 Shape framing procedure. 67

Figure 4-7 CD deviation versus incident dose [110]. 68

Figure 4-8 Correction procedures of the RFPEC system. 72

Figure 4-9 Comparison between experimental and simulation results. 74

Figure 4-10 Variation of λ with incident dose tolerance for various CD values. 80

Figure 5-1 Definition of edge error. 90

Figure 5-2 (a) Test pattern for the relative error analysis of section 5.1. (b) Geometric frames of reference of the pattern, mesh and sub-field grids. 91

Figure 5-3 Variation in edge error for edge 1, with $M = 0.2\beta$ and $S = 0.02\beta$, as ϕ_M and ϕ_S move from 0 to 2π. 93

Figure 5-4 Variation of edge errors due to changing the values of M and S. 94

Figure 5-5 Approximation of *erf(r)* by a piece wise linear function, *P(r)*. 95

Figure 5-6 Relative errors associated with varying the order of the convolution matrix... 97

Figure 5-7 Simulated resultant normalized deposited energy density distribution - detail. 98

Figure 5-8 Simulated resultant normalized deposited energy density distribution. 100

Figure 5-9 Correction procedures in the conventional PADPEC and the EPADPEC schemes. 103
Figure 5-10 Variations of the maximum error in D_I with number of iterations. 107
Figure 5-11 Normalised convolution, *P(x)* between a normalised Gaussian distribution and an isolated semi-infinite shape. 108
Figure 5-12 The interior hole problem. 109
Figure 5-13 The applied and ideal incident dose distribution of the isolated large one-dimensional pattern. 110
Figure 5-14 Elimination of the interior hole problem. 111
Figure 5-15 The uniform upper-level quantization strategy. 112
Figure 5-16 EPADPEC data flow. 113
Figure 5-17 Shape definition in the exel format. 115
Figure 6-1 KasugaTP for comparing various PADPEC systems. 118
Figure 6-2 RauTP for comparing LPEC, CAPROX and EPADPEC. 119
Figure 6-3 Cross section of a basic building block of the complementary-metal-oxide-silicon integrated circuit. 120
Figure 6-4 Simulated resultant normalized deposited energy distributions at edges 2 and 3 of the test pattern. 124
Figure 6-5 Line width profile of the monitor shape without any PEC. 125
Figure 6-6 Line width variations of the monitor shape for KasugaTP corrected by MPADPEC, KPADPEC and EPADPEC. 125
Figure 6-7 Line width variations of the monitor shape for KasugaTP corrected by EPADPEC. 126
Figure 6-8 Variation of the line width errors with *M* and *S* for EPADPEC corrected KasugaTP. 129
Figure 6-9 Variation of the maximum line width errors with *M* and *S* for (a) MPADPEC and (b) KPADPEC corrected KasugaTP. 130
Figure 6-10 Variation of the line width errors and ζ with the physical widths covered by *A* for EPADPEC corrected KasugaTP. 131
Figure 6-11 The simulated developed resist image (shaded regions) of EPADPEC corrected RauTP. 132
Figure 6-12 The simulated developed resist image (shaded regions) of LPEC corrected RauTP. 133

Figure 6-13 The simulated developed resist image (shaded regions) of CAPROX with Manhattan fracturing corrected RauTP. 134
Figure 6-14 The simulated developed resist image (shaded regions) of CAPROX with physical fracturing corrected RauTP. 135
Figure 6-15 Experimental and analytical line spread EDDs........ 137
Figure 6-16 Scanning electron microscope photographs of RauTP with and without PEC. 138
Figure 6-17 Scanning electron microscope photographs of RauTP with PEC........ 139
Figure 6-18 The data flow of a typical electron beam data preparation system........ 140
Figure 6-19 Variations of t_{PEC} of EPADPEC with the square of the order of A for layer 11 of MP16. 142
Figure 6-20 (a) Variation of t_{PEC} of EPADPEC with M and S for layer 11 of MP16. (b) Variation of the sub-rectangle and sub-frame count after shape fracturing with S for layer 11 of MP16. 143
Figure 6-21 Variations of t_{PEC} of MPADPEC with M and S for layer 11 of MP16. 144
Figure 6-22 Variations of t_{PEC} of KPADPEC with M and S for layer 11 of MP16. 145
Figure 6-23 Speed comparison among various PEC schemes........ 146
Figure 6-24 Variation of the maximum line width errors with the number of quantization levels for EPADPEC corrected KasugaTP. 148
Figure 6-25 Variation of the average line width errors with the number of quantization levels for EPADPEC corrected KasugaTP. 149
Figure 7-1 Corner rounding of a 12.8×12.8 μm^2 isolated square. 153
Figure 7-2 Mask patterns, showing serif and jog modifications. 154
Figure 7-3 Six distinct types of corners for rectangles. 155
Figure 7-4 Frames and central rectangles formed by EPADPEC without the corner correction algorithm........ 155
Figure 7-5 Type 1 corner correction. 156
Figure 7-6 Inner 90° corner correction. 157
Figure 7-7 The corner rounding at a type 4 corner of two 2.56 × 2.56 μm^2 isolated squares. 158
Figure 7-8 Test pattern for evaluating the optimum parameter values of the corner correction scheme. 159
Figure 7-9 Simulation windows at (a) inner 90° and (b) outer 90° corners. 160

Figure 7-10 Variation of the improvement factor on the absolute area difference at corners A and B (Figure 7-8) with l_2. 161

Figure 7-11 Variation of the improvement factor on the absolute area difference at the eight type 1 corners (Figure 7-8) with l_1 and w_1. 161

Figure 7-12 Shape corner fidelity metrics. 162

Figure 7-13 Simulated developed resist images (shaded regions) of the test pattern (Figure 7-8) corrected by EPADPEC. 163

Figure 7-14 The incident dose ranges of central rectangles, frames and corner squares. 164

Figure 7-15 The simulated developed resist image (shaded regions) of RauTP corrected by EPADPEC with the corner correction. 165

Figure 7-16 The computation time overhead factor of the corner correction method for various layers of MP16. 166

Figure 7-17 Kratschmer's shape partitioning scheme. 168

Figure 7-18 Procedures of PFA. 170

Figure 7-19 Variation of the maximum line width error of KasugaTP with S of SFA, and the isodose line count and the minimum fractured size of PFA. 172

Figure 7-20 Variation of the rectangle count after shape fracturing with S of SFA and the isodose line count and the minimum fractured size of PFA. 173

Figure 7-21 Variation of t_{PEC} with the sub-field size of SFA and the isodose line count and the minimum fractured size of PFA. 174

Figure 7-22 The relative speed and rectangle count between SFA and PFA for various average shape sizes. 175

Figure 8-1 Existing data flow diagram of the EBL system. 177

Figure 8-2 Estimated computation times of various components in the EBL system on layer 11 (polysilicon gate layer) of MP16. 178

Figure 8-3 Data flow of the proposed EBL system. 179

Figure 8-4 Data flow diagram of the data processing hardware in the proposed EBL system. 180

Figure 8-5 Timing components of EPADPEC. 181

Figure 8-6 Breakdown of t_{PEC} of EPADPEC. 183

Figure 8-7 Projected computational times of EPADPEC components relative to t_{Write}. 184

Figure 8-8 Block diagram of a hardware/software co-design implementation of EPADPEC. 187

Figure 8-9 A pipelined and parallel hardware implementation of the interpolation and quantization procedures. 188
Figure 8-10 Selection of the incident dose values of the four nearest mesh sites from the nine registers. 189
Figure 8-11 A pipelined and parallel hardware implementation of the evaluation of the λ values of shapes. 191
Figure B-1 The overall architecture of EDAPS. 212
Figure B-2 The architecture of the PEC module. 214
Figure B-3 Hierarchy chart of C++ classes used to model the GDSII elements. 216
Figure B-4 Data elements. 217
Figure B-5 The architecture of the proximity effect simulator. 218
Figure B-6 Data structure for storing the developed resist image in the threshold developer simulator. 220
Figure B-7 Quantization of a resist characteristic curve. 221
Figure B-8 Scanning electron microscope photograph of RauTP corrected with EPADPEC. 222
Figure B-9 Experimental and simulated EBMF 10.5 EBL machine write times. 224
Figure C-1 The overall architecture of EDAPS. 228
Figure C-2 Main user interface of EDAPS. 229
Figure C-3 Toolbar of EDAPS. 230
Figure C-4 Standard Windows 95 Print dialog box. 232
Figure C-5 Print Preview window. 233
Figure C-6 Standard Windows 95 Print Setup dialog box. 234
Figure C-7 Grid Size property page. 236
Figure C-8 Mapping Mode property page. 236
Figure C-9 User interface of the GDSII layout browser. 238
Figure C-10 Cell Selection dialog box. 239
Figure C-11 Layer Visibility Selection dialog box. 240
Figure C-12 EBL Machine property page. 241
Figure C-13 Leica EBMF 10.5 option dialog box showing the default values. 242
Figure C-14 Non-Gaussian spot, step and write EBL machines options dialog box with the default parameter values. 243
Figure C-15 EDF property page with the default parameter values. 244
Figure C-16 User interface of the shape processor viewer. 246

Figure C-17 Exposure Pattern Viewer Display Settings dialog box 248
Figure C-18 PADPEC Settings dialog box 249
Figure C-19 No PEC dialog box 251
Figure C-20 User interface of the exposure pattern viewer 252
Figure C-21 Exposure Simulator Settings dialog box. 253
Figure C-22 Threshold Developer Simulator Settings *wizard*-type dialog box. 254
Figure C-23 Resist Characteristic Curve Developer Simulator Settings dialog box 255
Figure C-24 User interface of the exposure simulator viewer. 258
Figure C-25 Exposure Simulator Display Settings dialog box 258
Figure C-26 User interface of the threshold developer simulator viewer 259
Figure C-27 User interface of the resist characteristic curve developer simulator viewer. 260

Acknowledgements

I would like to express my profound thanks to Professor Andrew Brown for his consistent guidance, great patience, encouragement and advice throughout this research.

I am very grateful to Dr. Paul Routley for fabricating the test patterns for this research work, and his help and patience in answering my endless questions on the electron beam lithography (EBL) machine in the Department of Electronics and Computer Science, University of Southampton. Thanks are due to Mr. Ian McNally for supplying real circuit patterns and to Professor Henri Kemhadjian for allowing us to use the EBL machine.

I would also like to thanks Mr. John Clark and Dr. Mark Zwolinski for their help related to the GDSII format specifications.

Thanks are due to Dr. Zaher Baidas and Dr. Alan Williams for their invaluable help and patience and diligence in answering my endless requests.

Thanks to all other members of the Electronics Systems Design Group at the University of Southampton.

Finally, I would like to thank to my family for their unconditional support and love.

Chapter 1

Introduction

The process of pattern definition on a wafer surface by photon exposure (optical lithography) is similar to a photographic process. A pattern is defined on a mask that is analogous to a monochrome photographic negative. The wafer surface is coated with a resist and is exposed selectively by photons projected through the mask and de-magnifying lenses (Figure 1-1). The photon exposure modifies the resist, leaving it more soluble (positive tone resists) or less soluble (negative tone resists) in a solvent. After developing the wafer in the solvent, the projected mask pattern is formed on the wafer surface.

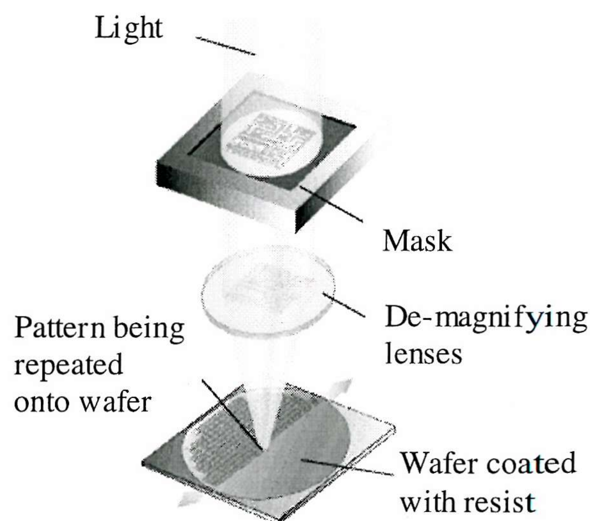


Figure 1-1 Image projection system in optical lithography.

Ever since the invention of the first solid state transistor, the semiconductor industry has been reducing the physical size of circuit components and hence increasing their speed and complexity. Throughout its history, the industry has maintained its growth by achieving an annual 25-30% fabrication cost reduction per function. Decreasing feature size remains the largest contribution to productivity growth. Statistics show

that the critical dimension (CD) or minimum feature size of dynamic random access memory (DRAM) decreases in size by roughly a factor of two every six years [1]. On the other hand, the number of shapes (polygons) increases by roughly a factor of three every three years, with a corresponding increase in pattern data volume. Current advanced microprocessors are produced using 0.25-micron technology. Taiwan Semiconductor Manufacturing has just started producing commercial devices using 0.18-micron technology [2], while CDs of present day experimental devices have already reached 100nm and below. Optical lithography is little use below 0.1 μm (see section 2.3). These current trends in integrated circuit technology are rapidly approaching the point at which electron beam lithography (EBL) with its high-resolution capability will play a pivotal role in device fabrication.

In theory, the resolution of electron optical systems can approach 0.1nm [3]. However, the ultimate resolution of an EBL system is set by the interaction range of incident electrons with a resist-substrate structure and the development mechanisms of the resist. Scattering of the incident electrons in the resist-substrate structure causes undesired energy to be deposited in the area surrounding a written shape. Any shape that falls within this area can suffer significant variation from its intended dimensions in the developed resist image. This effect is well known as the *proximity effect*. To correct for the effect, the written pattern is modified in such a way that the developed resist image is closer to its desired shape. This modification is known *as proximity effect correction* (PEC). In the past three decades, many PEC methods have been developed. However, all the techniques suffer from one of two following problems:

1. They are not general and accurate enough to correct a wide range of circuit patterns;
2. They are computationally too intensive to allow correction in a reasonable amount of time. Ideally, the throughput of the electron beam machine should not be rate-limited by the PEC software.

PEC is a well-established step in the data processing chain from a computer aided design (CAD) workstation to an electron beam machine. A CAD system usually codes pattern data in a relatively high level format such as GDSII [4] stream format which is the industry standard for interchanging integrated circuit designs between CAD systems. The format supports multi-vertex polygons, repetitive structures and

hierarchy. To drive an EBL machine directly, the data needs to be converted into a low-level format that is native to the machine. Besides hierarchy unwrapping in the format conversion, the data might need significant extra processing such as overlap removal, biasing, tone reversal and PEC (PEC is the most numerically intensive step). The processing is normally done off-line by a piece of converter software running on a conventional computing engine. The output data is usually many GByte in length, and is extremely cumbersome to manipulate. As chip sizes increase and CDs decrease, the size of this final data set will continue to increase, to the point that its manipulation will threaten the overall throughput of the system.

1.1 Objectives

The main objective of this research is to produce a method, superior in terms of correction speed and accuracy to commercial and published PEC systems. A secondary aim is to produce a feasibility study of the possibility of *accelerating* the improved PEC method so that it does not rate limit the throughput of an EBL machine. By incorporating sufficient computing capability into the machine itself, the data processing can be done in real-time. This will allow the high-level pattern data to be input directly to the machine, thereby removing the current need of intermediate data files of many GByte.

Specifically, the work presented encompasses:

1. The development of a PEC scheme, enhanced in terms of correction speed and accuracy.
2. Quantitative performance comparisons between this improved PEC method and commercial PEC systems.
3. The development of an electron beam data preparation system (EDAPS) that incorporates the enhanced PEC module and auxiliary tools. These tools include a proximity effect simulator, an EBL machine write time simulator and a GDSII layout browser.
4. A feasibility study of performing the improved PEC technique in real time.

1.2 Thesis organisation

Chapter 2 gives a brief account of the architecture of an EBL machine. After pointing out different writing strategies employed in EBL machines, applications of EBL and the potentially rival technologies are outlined. Chapter 3 first describes the nature of the proximity effect, followed by mathematical modelling of the proximity effect in EBL. Finally, techniques for minimising the proximity effect are discussed.

Chapter 4 reviews existing PEC techniques before comparing them in terms of their computing requirements and correction accuracy. The last section of the chapter rationalises the decision of selecting the pattern area density proximity effect correction (PADPEC) scheme for the basis of this research.

Having explained and analysed sources of correction error in PADPEC, Chapter 5 details the theoretical background and the algorithmic implementation of the Enhanced PADPEC, (*EPADPEC*) method. Chapter 6 evaluates and compares the performance of EPADPEC with other commercial and published PEC systems in terms of correction speed and accuracy. Chapter 7 describes further improvements to EPADPEC that enhance the correction accuracy at the expense of correction speed.

Chapter 8 provides a feasibility study of performing EPADPEC in real time. The last chapter summarizes the contributions of this research and discusses potential further work arising from the results of this research. Appendix A contains a journal paper published, describing some of this work. Appendix B details the architecture of the electron beam data preparation system (EDAPS), while Appendix C contains the User Manual of the system.

Chapter 2

The architecture and application of an electron beam lithography (EBL) system

This chapter begins by giving a brief account of the architecture of an EBL machine. After pointing out different writing strategies employed in EBL machines, the applications of EBL and the potentially rival technologies are stated.

2.1 Overview of an EBL system

A simplified block diagram of an EBL system is shown in Figure 2-1. The system consists of the following components:

- A column containing an electron gun for creating a beam of electrons and a set of electromagnetic lenses to form the electron beam.
- A stage for holding and removing a sample (a substrate coated with resist) under the beam area.
- A computer that controls the operation of the system.

By varying the fields of the deflection lenses, the beam can be steered around, tracing out a pattern on the sample.

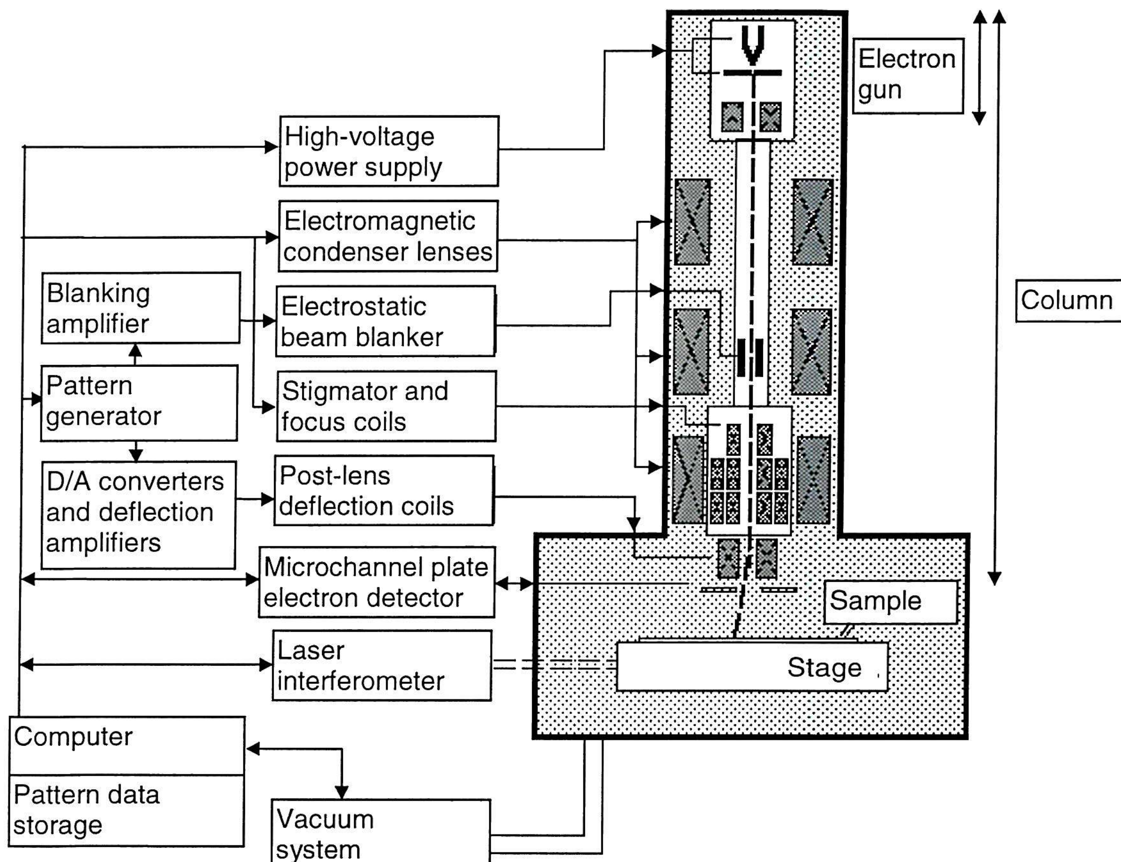


Figure 2-1 A simplified block diagram of an EBL system.

A column is typically composed of:

1. An electron gun for creating a beam of electrons.
2. Two or more electromagnetic lenses for focusing and de-magnifying the beam.
3. A set of coils or plates for deflecting the beam.

4. An electrostatic blanker for turning the beam on and off.
5. A stigmator for correcting any astigmatism in the beam.
6. Apertures for helping to define the beam.
7. Alignment systems for centring the beam in the column.
8. An electron detector for assisting with focusing and locating marks on the sample.

Conventionally, the optical axis (Z) is parallel to the electron beam, while X and Y are parallel to the plane of the sample.

Underneath the column is a chamber containing a stage for moving the sample around and facilities for loading and unloading it. Associated with the chamber is a vacuum system needed to maintain an appropriate vacuum level throughout the machine and also during the load and unload cycles. A set of control electronics supplies power and signals to the various parts of the machine. Finally, the system is controlled by a computer, which may be anything from a personal computer to a mainframe. The computer handles such diverse functions as setting up an exposure job, loading and unloading the sample, aligning and focusing the electron beam, and sending pattern data to the pattern generator.

In many ways, the system is similar to an electron microscope or a cathode-ray tube found in television sets and computer screens. What makes an EBL system significantly more complicated than either an electron microscope or a television set is in the scale, the accuracy, the precision, and the automation of operation.

2.2 Electron beam writing strategies.

The vital parameters defining the writing strategy of an EBL machine are beam shape, beam scanning and stage movement technologies [5]. The beam is either raster scanned or vector scanned. The former method scans the entire wafer, whereas the latter scans only the written regions. Stage movement is either continuous (on the fly) or step-and-write. The latter scheme stops the stage for writing and then moves it to the next region to be exposed.

There are three types of beam shape technologies: Gaussian spot, variable shape beam (VSB) and cell projection.

The first beam technology focuses the beam to a desired spot size and rasters the beam to form shapes. The beam charge intensity profile resembles a Gaussian distribution. Gaussian spot EBL systems have the highest resolution capability but the lowest throughput. For a Gaussian spot EBL machine employing a step-and-write stage movement technology, the machine writes shapes by pausing the focused beam at each written exel (exposure pixel) position before deflecting the beam to the next written exel position.

A VSB system forms the beam itself into shape primitives (rectangles and triangles) using two rectangular apertures. The upper aperture forms two sides of a rectangle, and the overlap of the lower aperture defines the length and width of the rectangle (Figure 2-2(a)). Deflectors between the apertures determine the overlap. When one of the apertures is rotated 45° with respect to the other, deflecting the beam along the diagonal of the lower aperture forms a triangular shape (Figure 2-2(b)). The system writes complex shapes using combinations of these shape primitives.

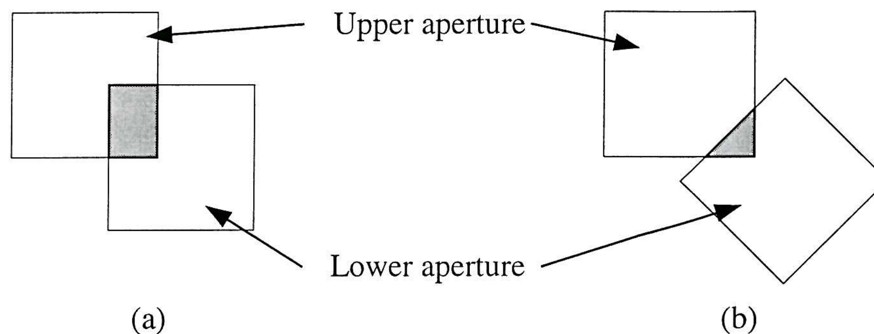


Figure 2-2 Formation of shape primitives (shaded shapes) in a VSB system.

Using complex shaping apertures, a cell projection EBL machine [6, 7, 8] writes an entire device cell in a single shot instead of writing single shape primitives or dots. This machine provides the highest throughput for highly repetitive patterns. Table 2-1 shows the writing strategies of existing commercial EBL machines.

Supplier	Model	Beam Shape			Scan		Stage	
		Spot	VSB	Cell	Raster	Vector	On-fly	Step/Write
Etec	MEBES 5000	X			X		X	
	MEBES 4500	X			X		X	
Hitachi	HL-800M		X	X		X	X	X
	HL-800D		X	X		X	X	X
JEOL	JBX-7000MVII		X			X		X
	JBX-6000FS	X				X		X
	JBX-5000LS		X			X		X
	JBX-8600DV		X			X		X
Leica	ZBA31		X			X	X	
	ZBA32		X			X	X	
	VB-6	X				X		X
	WePrint200		X	X		X	X	
Lepton	EBES4	X				X	X	

Table 2-1 EBL systems.

2.3 Applications

Optical lithography is a *projection process*, whereas EBL is a *direct write* process. The direct write process is similar to the projection process except patterns are written directly onto the wafer surface without the need of any masks. Direct write techniques are sequential in that the entire pattern is traced out, one exposure element at a time, whereas projection processes are parallel in the sense that the whole pattern is exposed at the same time. Thus, projection methods have a much higher throughput than direct write approaches (Table 2-2). Because of its much higher throughput, refractive optical lithography (optical lithography that uses de-magnifying lenses) is much cheaper than EBL and is currently the mainstream approach for volume production in the industry.

Technology	Beam shape	Resolution (nm)	Throughput (6 inch wafers per hour)
Refractive optical lithography using light with 248nm wavelength (projection process)	-	180	40-80
EBL (direct write process)	Gaussian (VB-6)	<30	<1
	VSB (JBX-8600DV)	100	<10
	Cell (HL-800D)	300	≤15

Table 2-2 Capabilities of existing lithography technologies.

Presently, EBL plays an essential role in the semiconductor industry, where it has three niche markets:

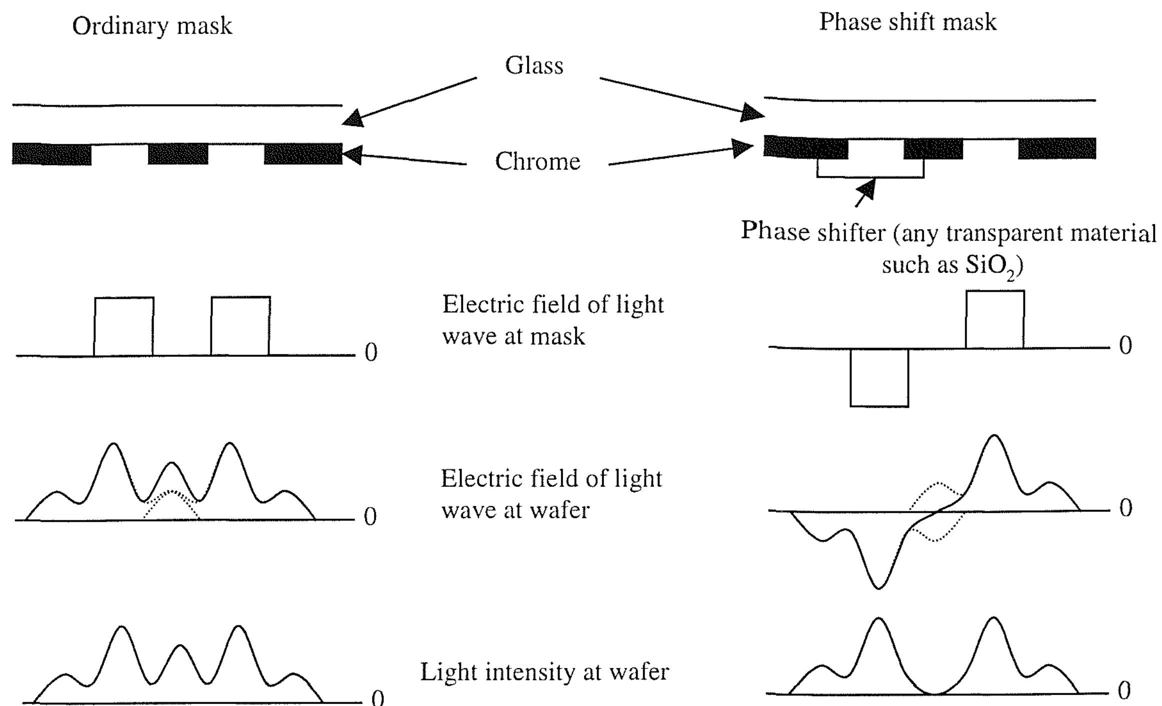
- Prototyping of research devices.
- Mask making.
- Small volume production of advanced semiconductor devices.

Owing to its high-resolution capability (<30 nm against 180 nm for refractive optical lithography), EBL is the preferred process to make research devices to study the scaling limits of integrated circuits, quantum effects and other novel physical phenomena at very small dimensions. These devices usually have minimum feature sizes of 100 nm or less. A typical application is the study of the Aharanov-Bohm effect [9-11], where electrons moving along two different paths about a micron in length can interfere destructively or constructively, depending on the strength of an applied magnetic field.

EBL is a well-established technique for mask making, typically the chrome-on-glass masks used by optical lithography tools. It is the preferred technique because of its ability to meet stringent line width control and pattern placement specifications. The masks are mainly used for fabricating integrated circuits, although other applications such as flat panel displays and disk drive heads also use such masks.

The final application is in fabrication of low volume specialized products, such as optical waveguides and Gallium Arsenide integrated circuits. Hewlett Packard and Texas Instruments have been using EBL in production of Gallium Arsenide devices for the past twelve and seven years respectively. Here, the resolution of EBL has proven essential for meeting the critical dimension specifications of the devices that are perhaps one or two generations ahead of mainstream optical lithography techniques. The use of EBL is justified for low volume device production when the mask cost (a few thousand dollars, required by the refractive optical lithography) becomes excessive.

Since no materials exist that are optically transparent at wavelength significantly shorter than the wavelength of deep ultraviolet light (193nm), optical lenses needed for refractive optical lithography cannot be advanced to smaller wavelengths. For light with lower wavelengths, mirrors are used in the image projection system. Using resolution enhancement techniques, refractive optical lithography could be extended below 193nm to about 0.1 um. These techniques include optical proximity effect correction and phase shift masks. Optical proximity effect correction is the modification of mask patterns in such a way that the transferred patterns are closer to the desired pattern dimensions. The effect refers to pattern dimension deviations incurred in lithography, whether or not they are entirely optical in nature. In a phase shift mask, certain regions are covered with a shifter that alters the phase of the incident light by 180° [12]. Destructive interference between light waves diffracted from adjacent apertures minimizes the light intensity between the apertures on a wafer, and hence improves the resolution of the lithography tool (Figure 2-3). Although these techniques may be useful for fabricating isolated transistor gates or other simple sparse patterns, they do not work for patterns with any degree of complexity.



For an ordinary mask, constructive interference between light waves diffracted from adjacent apertures increases the light intensity between the apertures on a wafer surface. On the contrary, destructive interference minimizes the light intensity between the apertures in a phase shift mask.

Figure 2-3 Interference with a phase shift mask.

Refractive optical lithography is fast approaching its physical resolution limit as feature sizes approach 100nm. In the 1997 edition of The National Technology Roadmap For Semiconductors, the Semiconductor Industry Association predicts that refractive optical lithography will fall into disuse after 2006 when 100nm-technology generation is in use. Below 100nm, possible successor technologies are electron beam, ion beam, extreme ultraviolet and X-ray proximity lithography.

In X-ray proximity lithography, a 1x mask is held in close proximity (hence the name proximity lithography), typically about 10-40 μm to the wafer (Figure 2-4). The mask is called a 1x mask because the mask dimensions are not reduced in the projected image. The masks usually have feature dimensions ranging from 0.25 μm to less than 0.1 μm. Fabricating these masks will need placement accuracy and line width control of 20 nm or better. The manufacture of these masks is an emerging market in the mask industry. If X-ray proximity lithography ever becomes the successor technique for volume production, it will have a dramatic effect on EBL machine development

since the requirements of resolution, throughput, and accuracy, while technologically feasible, are far beyond the capabilities of any single machine available today.

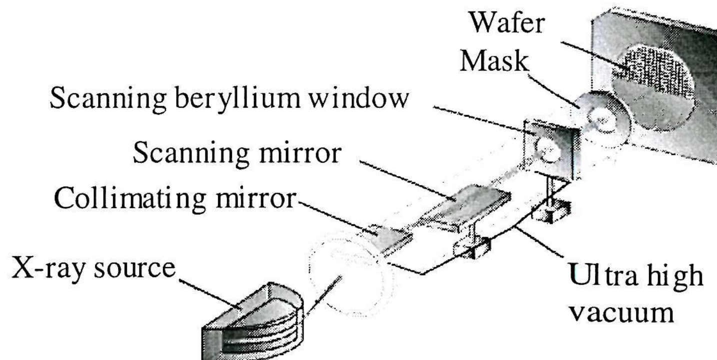


Figure 2-4 Image projection system in X-ray proximity lithography.

2.4 Potentially rival technologies

For optical mask making, there are optical mask writers that are based on multiple individually controlled round laser beams. A current state-of-the-art laser beam system, ALTA 3500, is capable of fabricating masks with a minimum feature size down to 0.25 μm. Although optical mask writers are generally cheaper than EBL machines, the latter tools are technologically more advanced and have higher resolution capability. Thus, EBL will continue to be the most attractive option for manufacturing advanced masks, such as 1× X-ray, phase shift and optical proximity effect correction masks.

Ion beam lithography is in many ways similar to EBL. Both use charged particle beams to write patterns. The complexity, resolution, throughput and cost of ion beam systems are comparable with EBL systems. Compared with EBL, the main advantages of ion beam lithography are:

1. It has a negligible proximity effect, as ions tend to lose energy in a resist by generating low energy secondary electrons with short ranges of about 5 nm [13]. It is relatively easier to control the depth of ion penetration in a resist-substrate structure.
2. Ion sensitive resists are much more efficient than electron sensitive resists [14]. Because of the higher energy deposited per volume, ion beam bombardment

typically needs an incident dose (charge per unit area) of about two orders of magnitude smaller than the dose required in EBL for a similar resist exposure. The exact resist sensitivity depends on the ion species and charge.

3. When the appropriate ion species are available, it is possible for in situ doping and in situ material removal by ion beam assisted etching.

On the other hand, ion beam lithography suffers from the following disadvantages:

1. Relative to EBL, the technique has not reached the same advanced stage of development.
2. Possible damage to partially complete chip devices from ion bombardment.
3. The technique does not work on all thicknesses of resist.

For volume production, current research efforts are on increasing the low throughput of EBL. One approach is to have multiple-beams writing simultaneously [15]. Each beam is generated by a separate miniature electron optical column [16]. Another approach is to change the paradigm of EBL from direct write to projection process. A research group at AT&T Bell laboratory is developing a projection EBL system called SCALPEL. Instead of being absorbed, the unwanted electrons are scattered away and filtered out by various lenses and apertures in the system [17, 18]. The main hurdles in developing a fully functional electron projection system are the mutual repulsion of the charged electrons and the local heating on the mask due to absorption of electrons. The former problem leads to difficulties with a high flux (intensity) projection system, whereas the latter problem causes distortion in the projected image.

X-ray proximity lithography is another front runner technology to replace refractive optical lithography as a high volume production tool. Because the wavelength of the X-rays used is usually about 1 nm, so the intrinsic diffraction limits of X-ray proximity lithography are very low and minimum feature sizes down to 30nm have been demonstrated [19]. Unfortunately, suitable X-ray sources are very expensive and are energetically inefficient at the required wavelength. The present choice of X-ray source is a cumbersome synchrotron storage ring. As mask patterns are not reduced in the projected images, this requires stringent placement accuracy and line width control on the mask-making tool. Success of X-ray proximity lithography will hinge on the ability to manufacture 1× masks to specifications, cost and cycle time.

Extreme ultraviolet lithography systems use light of very short wavelength (about 13 nm) in the X-ray spectrum. The systems employ mirrors to de-magnify the mask patterns as shown in Figure 2-5. The mirror systems are complicated and the many reflections attenuate the light significantly, leading to long exposure times.

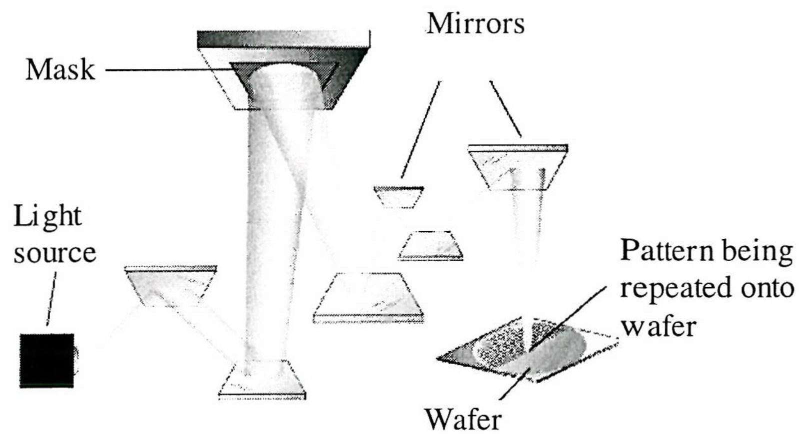


Figure 2-5 Image projection system in extreme ultraviolet lithography.

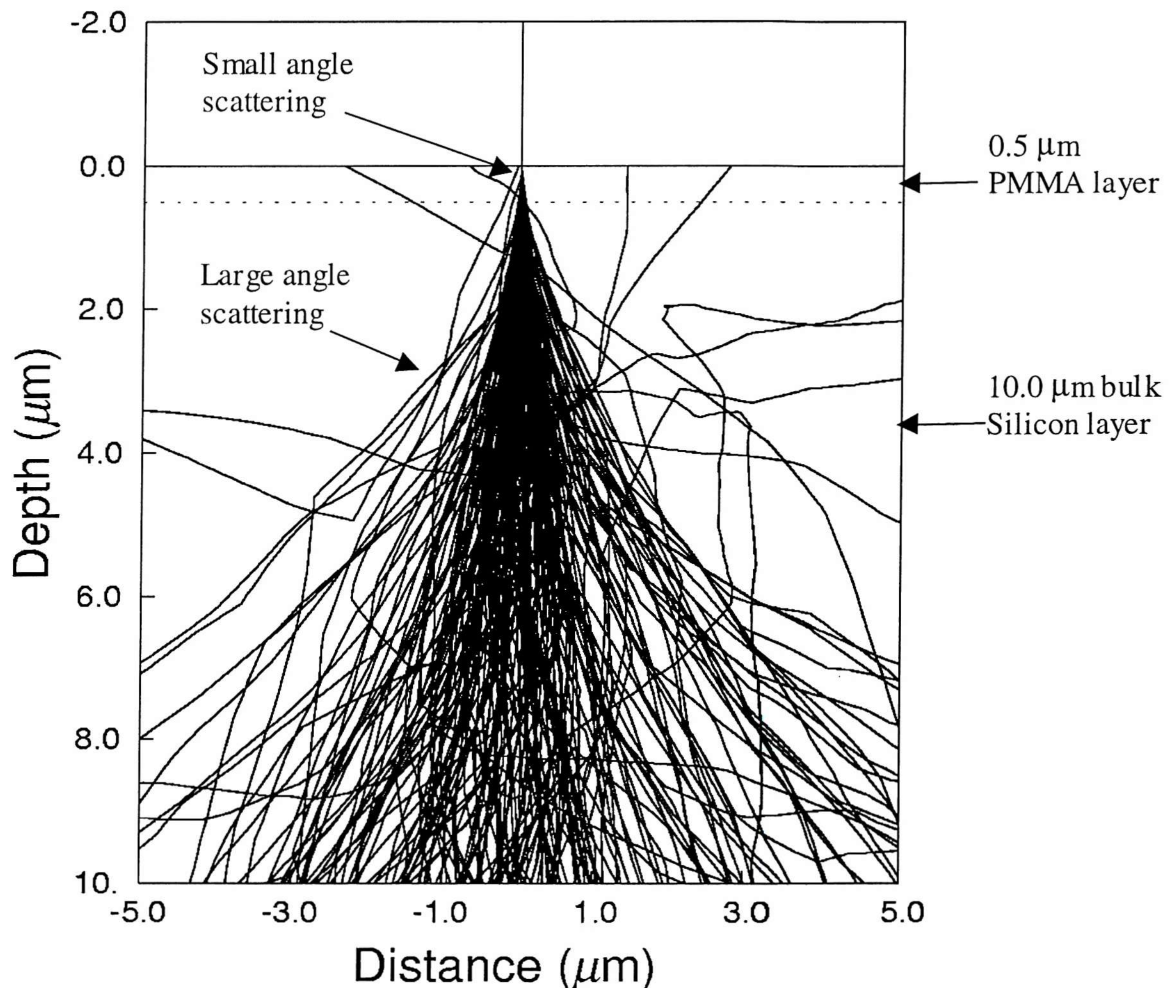
Chapter 3

Electron beam proximity effect

This chapter starts by explaining the physical nature of the proximity effect. An account of the mathematical models that describe the resist development process and the electron scattering in the resist and the substrate (section 3.2) are given. The last section (section 3.3) discusses techniques that minimise the proximity effect.

3.1 The nature of the proximity effect

As incident electrons travel through the resist, they experience many small angle scattering events (forward scattering). Penetrating through the resist into the substrate, some of them undergo a large angle scattering (back scattering) and travel back to the resist layer. During the scattering process, the electrons dissipate energy in the resist as they are continuously slowing down, producing a cascade of low energy electrons called secondary electrons. Figure 3-1 shows some computer simulations of electron scattering in typical samples consisting of a layer of polymethyl methacrylate (PMMA) resist on top of a silicon substrate.



Electron trajectories of 200 electrons, generated by a Monte Carlo simulation program (CASINO), in a typical sample composed of a layer of PMMA resist on top of a Silicon substrate at a beam acceleration voltage of 100kV.

Figure 3-1 Monte Carlo electron trajectories.

Forward scattering can result in a significantly larger beam diameter at the bottom than at the top of a resist. The increase in effective beam diameter (in nanometres) at the bottom of the resist is given empirically by:

$$d = 0.9\left(\frac{t}{V}\right)^{1.5}$$

where t and V are the resist thickness in nanometres and the beam acceleration voltage in kilovolts respectively [20]. According to this formula, using the thinnest possible resist and the highest available beam acceleration voltage minimises forward scattering.

The back-scattered electrons cause the proximity effect. A small fraction of secondary electrons may have significant energies, of the order of 1keV. These fast secondary electrons can contribute to the proximity effect in the range of a few tenths of a micrometre. Electron scattering ranges in the resist and the substrate define the region over which the back-scattered electrons expose the resist. The electron scattering range depends on:

- the beam energy (which is linearly proportional to the beam acceleration voltage);
- the substrate atomic number;
- the resist and the substrate film thickness.

As the beam energy increases, the energy loss per unit path length and the scattering cross-sections decrease[1] [21, 22]. Therefore, the lateral transport of the forward-scattered electrons decreases. However, the lateral extent of the back scattered electrons increases due to the longer electron range. As the substrate atomic number increases, the electron reflection coefficient increases, which in turn increases the back-scattered contribution.

---

[1] These are the predictions of the continuous energy loss law of Bethe and the screened Rutherford formula that describe the electron energy loss per unit path length and the electron scattering cross-sections respectively.

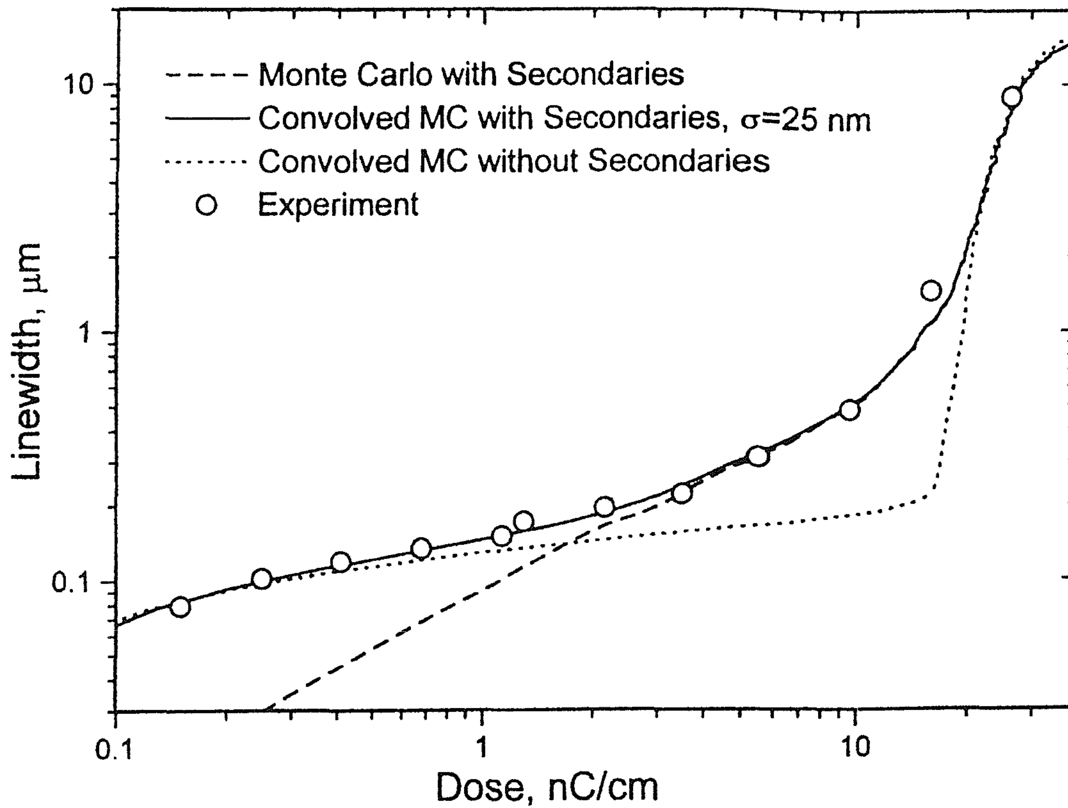
For a thicker resist film, the cumulative effect of the small angle collisions by the forward-scattered electrons is greater. Thus, the area exposed by the forward-scattered electrons at the resist-substrate interface is larger in thick films than in thin films. The physically thinnest possible substrates (i.e. of the order of about 60nm) have been shown theoretically and experimentally [23, 24] to minimise the contribution from the back scattered electrons. However, these thin substrates are fragile and difficult to handle in practice.

3.2 Profile modelling

A resist profile is defined as the remaining resist thickness distribution on a substrate after the resist is developed in a solvent as a function of a given incident dose distribution. The resist has been exposed by an electron beam prior to the developing process. The electron exposure modifies the resist, leaving it more soluble (positive resist) or less soluble (negative resist) in the solvent. In order to predict a resist profile, both the exposure and development models are needed. The energy dissipated per unit volume per electron in a resist is the fundamental quantity needed to determine the resist profile. This energy density distribution (EDD) in the resist due to point source electrons has been obtained for conditions of interest to EBL using the following methods:

- Monte Carlo simulations [25];
- analytic models [25];
- experiments [26].

Marrian *et al.* [27] and Fretwell *et al.* [28] demonstrate that incorporating the effect of fast secondary electrons in Monte Carlo simulations improves the agreement between the simulated data and experimental data in the transition region as shown in Figure 3-2.



Comparisons of simulated line spread functions, with and without secondary electrons, and experimental data from 50kV exposure on a 50nm thick SAL-601 resist. The line-spread function gives the deposited energy as a function of developed line-width for an incident charge per unit length. The beam has a Gaussian charge density distribution with a standard deviation (σ) of 25nm. These results are taken from [27].

Figure 3-2 Simulated and experimental line spread EDDs.

Unfortunately, the obtained EDDs are available only as complicated analytical formulae or in numerical form. Thus, these EDDs cannot be used easily in PEC algorithms. To overcome this difficulty, Chang [29] suggested fitting the results to a simple analytical function. The energy density function (EDF) contains two Gaussian terms, a narrow one for forward scattering and a wide one for backward scattering:

$$f(r) = \frac{1}{\pi(1+\eta)}\left[\frac{1}{\alpha^2}\exp\left(\frac{-r^2}{\alpha^2}\right) + \frac{\eta}{\beta^2}\exp\left(\frac{-r^2}{\beta^2}\right)\right] \quad \textbf{(3-1)}$$

where r is the distance from the point of incidence, α and β (called the *characteristic lengths*) reflect the beam broadening effects of the forward and backward scattering processes, η is the deposited energy ratio of back scattered electrons to forward scattered electrons. α and β are equal to ($\sqrt{2} \times$ the standard deviation) of the

respective Gaussian terms in equation (3-1). Equation (3-1) has been normalised such that:

$$2\pi\int_0^\infty rf(r)dr \equiv 1$$

(see Appendix E.2 for the proof). This exposure model describes the deposited energy density distribution at some resist depth for a point-like incident electron beam. The model is linear and space invariant. When a resist is exposed with an incident dose distribution, $d_I(r)$, the model describes the resultant deposited energy density distribution, $e_R(r)$, on the resist mathematically as a convolution of $d_I(r)$ with the EDF, *i.e.*

$$e_R(r) = k\int_{-\infty}^{\infty} f(r-\lambda)d_I(\lambda)d\lambda \qquad \textbf{(3-2)}$$

where k is a constant charge to energy conversion factor. The value of k depends on the specific resist and the beam voltage used. In practice, the finite electron beam size of an EBL system is taken into account by convolving the EDF with the beam charge density distribution.

For a Gaussian spot, step and write EBL machine, each exel is treated as a point source and is represented by a delta function, $\delta(r)$. Since

$$\int_{-\infty}^{\infty} f(r-\lambda)d_I(r)\delta(r)dr = f(r-\lambda)d_I(r)$$

so the integral in equation (3-2) becomes a summation (*i.e.* discrete convolution) as follows:

$$e_R(r) = k\sum_{-\infty}^{\infty} f(r-\lambda)d_I(\lambda)$$

If a resist has a high contrast, then detailed simulations of the development process are not needed. The development contours can be identified as contours of equal energy density. A profile will develop as long as it has received an energy density

greater than a certain critical value. This is called the threshold energy density model and the critical value is called the resist threshold energy density, E_T. The model is used extensively in PEC algorithms.

In the rest of this thesis:

- The quantity *normalized deposited energy density* is defined as the deposited energy density divided by the resist threshold energy density.
- The quantity *normalized incident dose* is defined as the incident dose divided by D_0, where D_0 is the constant incident dose required to achieve the resist threshold energy density value for an infinite written region, i.e.

$$kD_0 \int_{-\infty}^{\infty} f(r-\lambda)d\lambda = E_T$$

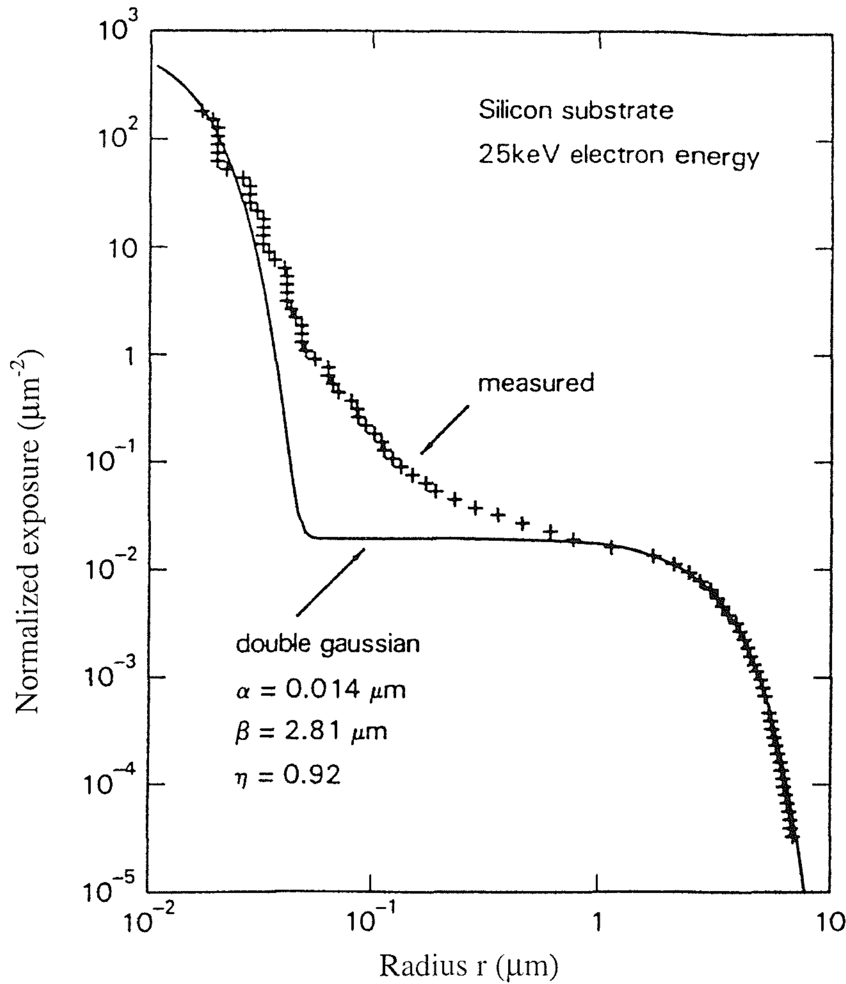
- The quantity *normalized exposure* is defined as D_0 divided by the incident charge.

3.2.1 Inaccuracies in the Gaussian model EDF

Experimental results on silicon at 25kV beam acceleration voltage obtained by Rishton *et al.* [26] show that the double Gaussian EDF fails to predict the deposited energy density accurately in the sub-micron range. There is a large discrepancy (more than an order of magnitude) in the range from 0.04 to 0.1μm as shown in Figure 3-3. These deviations may be due to:

- scattering in the resist;
- high energy secondary electrons;
- tails in the beam distribution.

It was also reported that the back-scattered electron distribution on Gallium Arsenide follows more closely an exponential distribution than a Gaussian. As the energy dissipation per unit path length is greater in Gallium Arsenide (which has a higher atomic weight than Silicon), the distribution of the energy dissipation follows more of an absorption (exponential) law than a random scattering (Gaussian) law. Other recent results [30, 31] also demonstrate that the double Gaussian model is inadequate for patterns with CD values less than 0.1μm. Using a hybrid Monte Carlo simulation, Gentili *et al.* [32] showed that the actual EDF is a sum of complex Gaussian-like terms.



r is the distance of a measurement / evaluation point from the beam point of incidence. The results are taken from [26].

Figure 3-3 Experimental and analytical point spread EDDs.

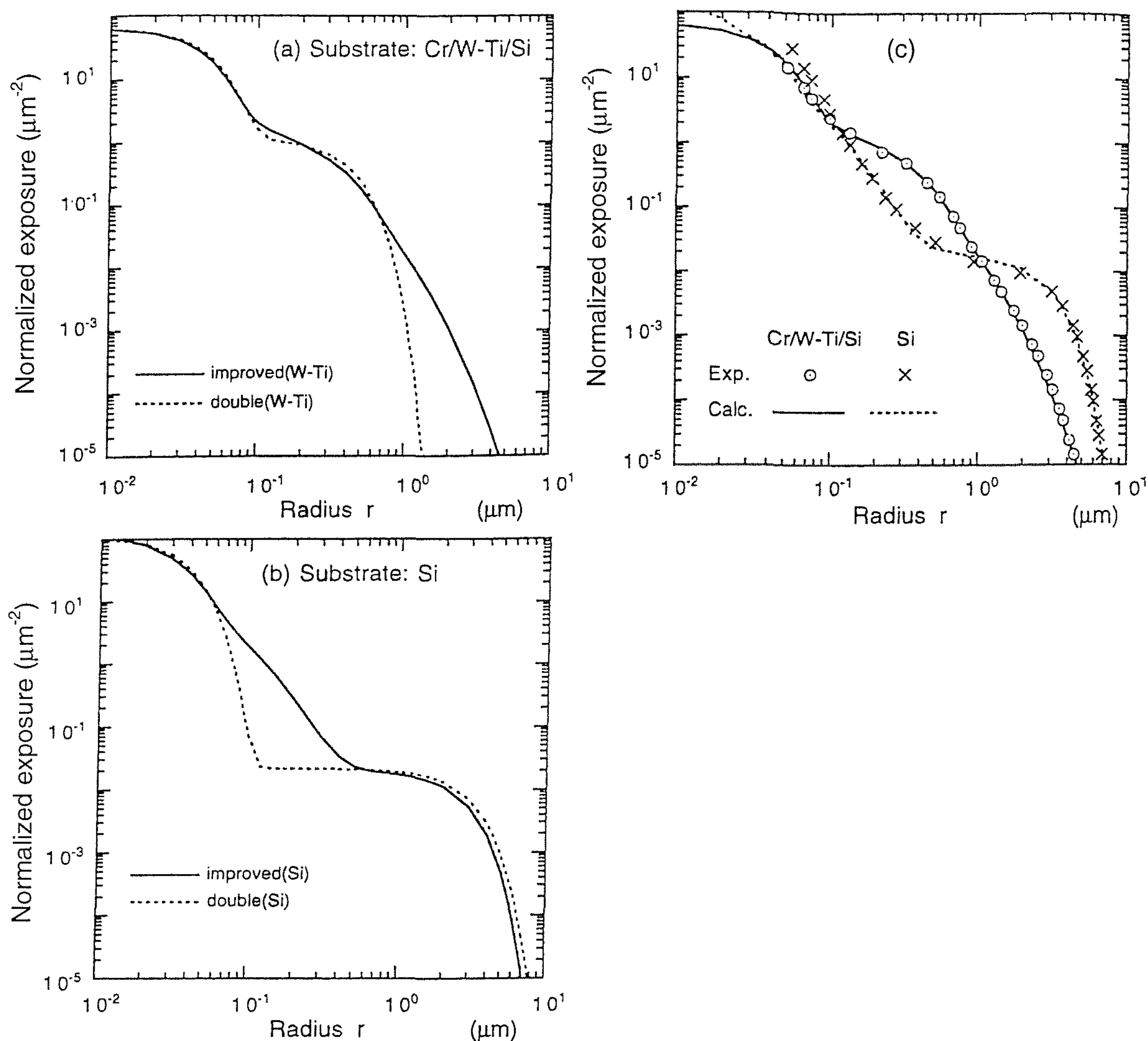
To improve the approximation accuracy, Rosenfield *et al.* [31] introduced a third Gaussian term to the double Gaussian EDF and verified experimentally the superiority of the triple Gaussian EDF over the double Gaussian EDF in approximating the actual EDD. Yu *et al.* [33] went a step further by using a quadruple Gaussian EDF.

Aya *et al.* [34] improved the double Gaussian model by including an extra term as follows:

$$f(r) = \frac{1}{\pi[1+\eta_1(1+\eta'_2)]}\left[\frac{1}{\alpha^2}\exp\left(\frac{-r^2}{\alpha^2}\right)+\eta_1\left\{\frac{1}{\beta^2}\exp\left(\frac{-r^2}{\beta^2}\right)+\frac{\eta'_2}{24\gamma^2}\exp\left(-\sqrt{\frac{r}{\gamma}}\right)\right\}\right]$$

(3-3)

Figure 3-4 shows this improved EDF fits much better with experimental results as compared with the double Gaussian model. However, equation (3-3) fails to produce



(a) a 0.05µm Chromium/ 0.5µm Tungsten (W-Ti) / Silicon resist-substrate structure referred as Cr / W - Ti / Si in the diagram; (b) a bulk Silicon substrate referred as Si in the diagram. The curves in (c) show excellent agreement between the experiment EDD data (Exp.) and the EDD (calc.) predicted by the improved EDF. The results are taken from [34].

Figure 3-4 Comparison between the double Gaussian and the improved EDF.

a closed form expression when convolved with a Gaussian distribution function[1]. Thus, it is mathematically less convenient than EDF with only Gaussian terms when it

[1] In a Gaussian spot EBL machine, the beam charge intensity distribution is a Gaussian (see section 2.2).

is used in PEC algorithms. Furthermore, a Gaussian term has the following desirable mathematical features [35]:

1. It readily factors into its x and y components/variables, i.e. its variables are *separable*. Thus, a multi-dimensional integral of a Gaussian term is reduced into a product of one-dimensional integrals of each of its components, simplifying the calculation considerably.
2. It integrates easily to error functions.
3. Convolution between two Gaussian terms produces another Gaussian term with $\sigma = \sqrt{\sigma_1 + \sigma_2}$, where σ_1 and σ_2 are the characteristic lengths of the two convolved Gaussian terms.

Appendix E contains the proof of these properties.

3.2.2 Determination of the EDF parameters

When curve fitting the analytic n Gaussian expression to either Monte Carlo or experimental EDD data, a merit function is used to assess the agreement between the data and the model. A standard non-linear technique based on the Levenberg-Marquardt [36] algorithm adjusts the parameters of the model to achieve a minimum in the merit function. For a chi-square merit function, the technique minimises the following sum:

$$\chi^2 = \sum_{i=0}^{\infty} \left[\frac{E_i - F_i}{\sigma_i} \right]^2$$

where E_i is the value of the Monte Carlo or experimental EDD at point i, and F_i is the value predicted by the adopted analytic model at point i. σ_i is the expected error in E_i and is usually assumed to be proportional to E_i. This merit function indicates the mathematical quality of the curve fitting.

However, Rosenfield *et al.* [37] noted that the best mathematical fit might not necessarily give the best results in the implementation of a PEC system. As alternatives, the LOG and ENERGY merit functions are introduced. The LOG technique minimises the following sum:

$$\sum_{i=0}^{\infty}\left|\log(E_i)-\log(F_i)\right|$$

whereas the ENERGY technique minimises the sum of energy deviations:

$$\sum_{i=0}^{\infty}\left|(E_i-F_i)r\Delta r\right|$$

where r is the distance from the point of incidence and Δr is the evaluation point interval. The former method generally gives all points equal weighting when performing the curve fitting, whereas the latter technique tends to fit well at high values of E_i and at larger distances. Several studies [37, 38] have proved that different curve fitting algorithms can result in significantly different extracted EDF parameters.

The analytic Gaussian expression can be in either a normalised form, $f(r)$ or a simpler non-normalised form, $f'(r)$ as follows:

$$f(r)=\frac{1}{\pi\sum_{i=1}^{n}\eta_i}\sum_{i=1}^{n}\frac{\eta_i}{\sigma_i^2}e^{-\left(\frac{r}{\sigma_i}\right)^2} \quad \textbf{(3-4)}$$

$$f'(r)=K\sum_{i=1}^{n}X_i e^{-\left(\frac{r}{\sigma_i}\right)^2} \quad \textbf{(3-5)}$$

where K is a constant, σ_i is the characteristic length of each term, $\eta_1 = 1$, and η_i and X_i define the contribution of each term. Fitting the EDD data using the non-normalised curve fitting technique with different starting r values results in significant variations in the extracted EDF parameters. On the other hand, the normalised curve fitting technique does not produce significant variations in the extracted EDF parameters. It also reduces notably the variations between the EDF parameters extracted from the experimental EDD data and the EDF parameters obtained from the Monte Carlo EDD data [37]. These results imply that the normalised curve fitting technique is more robust.

Other methods [39-43] exist for extracting the double Gaussian EDF parameters without the need for determining the actual EDD. These techniques use specific formulae to compute the parameters from measurements of certain exposed patterns.

3.3 Proximity effect avoidance

Many different techniques have been proposed and implemented to minimise the proximity effect. For patterns with fairly uniform density and line-width, one can simply adjust the overall dose until the developed patterns have the proper size. This method typically works well for isolated transistor gate structures.

Multilevel resist schemes reduce the forward scattering effect and result in steep developed resist slopes, but do not eliminate back scattered electrons completely. In order to reduce the proximity effect significantly, the bottom layer of the resist must be thick enough to absorb almost completely the electrons back scattered from the substrate. As an unwanted consequence, alignment marks buried under this resist layer are difficult to detect since electrons back scattered from the marks are also strongly absorbed. The extra complexity introduced by the use of multilevel resists increases processing cost considerably.

As mentioned in section 3.1, higher beam voltages (≥50 kV) decrease the forward scattering range, though in some cases, this can increase the back-scattering effect. When writing on very thin membranes such as X-ray masks, higher voltages reduce the back-scattered contribution as well since most of the incident electrons pass completely through the membrane [44]. However, Umbach *et al.* [45] showed that the deposited energy ratio of back-scattered to forward scattered contribution, η for gold/chromium/silicon x-ray mask structures would saturate at about 0.31 even if the beam voltages were increased substantially. Hence, increasing the beam voltage infinitely cannot eliminate completely backward scattering in the mask structures.

On the contrary, lower beam voltages reduce the electron back-scattering range. As the beam voltage continues to decrease, the value of the electron range will eventually become saturated. The saturated value is mainly determined by the beam diameter. When the electron range is smaller than CD, the use of very low voltage beams gets

rid of the proximity effect [46, 47, 48]. However, the penalty for this is that the thickness of a single layer resist must also be less than CD so that the electrons can expose the entire film thickness[1] [20]. Because electrons are hard to focus into a small spot and are more sensitive to stray magnetic and electrostatic fields, so low voltage systems have poorer alignment accuracy and their electron-optical design is much more complicated. Despite these difficulties, this is the current approach in optical mask-making, employing a 10 kV beam to expose a 0.3 μm thick resist with CD = 1 μm on a 5× mask. In more advanced studies, McCord *et al.* [49] used a 1.5 kV beam to write test patterns with CD = 0.15 μm on a 70 nm thick resist, while Houli *et al.* [50] employed a 2kV beam to fabricate both sparse and dense lines that are 30 nm wide on a 45 nm thick PMMA resist on top of a Galium Arsenide substrate. In conjunction with the low voltage beam approach, placing a layer of material with a high atomic number, such as tungsten, underneath the resist increases the useful range of electron energy. This has the effect of further limiting the electron back-scattering range.

---

[1] Proper exposure requires that the electron range in a resist film must be at least 1.5 times larger than the film thickness.

Chapter 4

Proximity effect correction (PEC)

This chapter gives a survey of existing proximity effect correction schemes with emphasis on their performance, strengths and weaknesses. After a comparison of their relative performances in section 4.5, the last section rationalizes the approach adapted by this research.

4.1 Overview of PEC techniques

Since the proximity effect is predictable, so it is amenable to correction. In solving the proximity effect correction problem, one generally specifies $e_R(r)$ in equation (3-2) to be equal to the desired deposited energy density distribution and tries to solve for $d_I(r)$. Gerber [51] derived a closed form solution of the equation for the double Gaussian EDF. His results imply that the correction problem is *ill-posed*[1] and the ill-posed nature is due to forward scattering. If we ignore the forward scattering by representing it as a delta function in the EDF, then the correction problem is well-posed. GHOST (a dose equalisation scheme for proximity effect correction) [52], PROXECCO (proximity effect correction by de-convolution) [53], RFPEC (representative figure proximity effect correction) [54] and PADPEC (pattern area density proximity effect correction) [55] use this approach to simplify the correction calculation.

All the correction schemes fall into one of the following classes:

1. GHOST correction, a proximity effect correction technique based on dose equalisation;
2. shape correction;
3. dose correction.

---

[1] A problem is well-posed if and only if:

- It has one and only one solution.
- And a small change in the data produces only a small change in the solution.

Otherwise the problem is said to be ill-posed.

4.2 GHOST (A dose equalisation scheme for proximity effect correction)

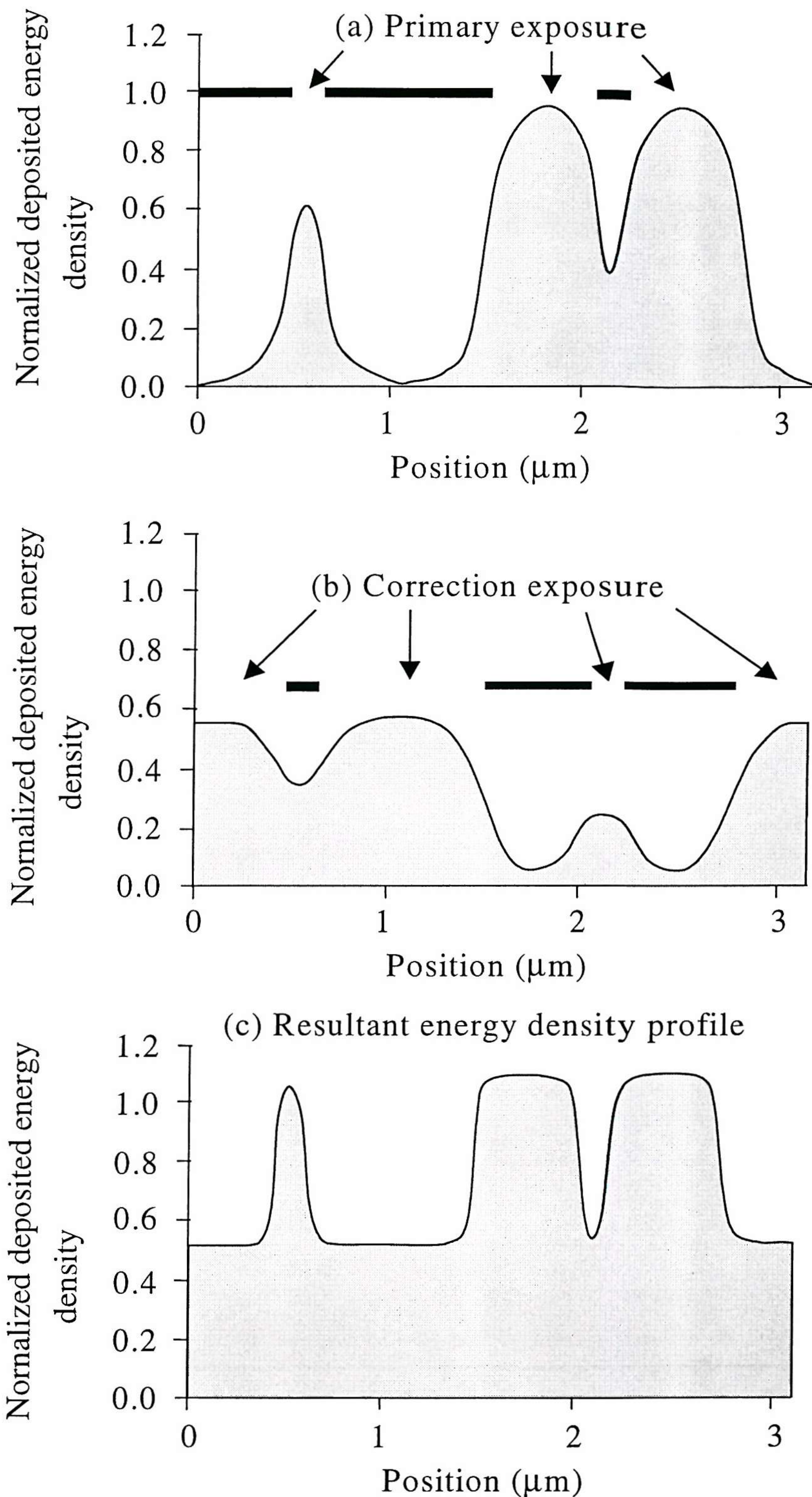
GHOST or dose equalisation proximity effect correction [52] works by equalising the deposited back scattered electron energy density at all points in a circuit pattern, with a defocused and attenuated beam. This scheme exposes a pattern and the reversed image of the pattern using electron beams (primary exposure beams) with a peak charge density of Q_e and correction beams with a peak charge density of Q_c respectively. The result is a *uniform* EDD at nominally unwritten regions as shown in Figure 4-1.

There are three basic assumptions that must be true for GHOST to work. They are:

- The main (forward scattering) Gaussian peak must be localised to a single exel;
- The defocused correction beams imitate successfully the EDD of the back scattered electrons of the exposure beams;
- The difference in the deposited energy density between nominally written and unwritten regions after GHOST correction must be sufficient to allow development of a pinhole-free resist pattern.

The correction beam approximates the EDD of the back scattered electrons of the exposure beams by fulfilling the following two conditions:

1. The peak energy density of the correction beam is equal to the peak back scattered electron energy density of the exposure beam.
2. The total energy deposited by the back-scattered electrons of the exposure beam is equal to the total energy deposited by the correction beam.



The pattern consists of one-dimensional lines and spaces. The vertical axis indicates the normalized energy density deposited on the resist, whereas the horizontal axis shows the horizontal location in the resist. (a) Normalized deposited energy density profile due to primary exposure. (b) Normalized deposited energy density distribution due to correction exposure. (c) Resultant normalized deposited energy density profile.

Figure 4-1 General technique of GHOST.

T.K. Leen [56] proved that GHOST gives an approximation to the exact solution of $d_I(r)$ to equation (3-2) which is convolved with the forward scattering (first) term of the double Gaussian EDF.

For a double Gaussian EDF and an electron beam with a Gaussian charge density distribution, the correction beam that satisfies the above conditions, has a *spot radius*[1], σ_c and Q_c as follows:

$$\sigma_c = \frac{\beta}{(1+\eta)^{\frac{1}{4}}} \qquad \textbf{(4-1)}$$

$$Q_c = Q_e \frac{\eta}{(1+\eta\ )} \qquad \textbf{(4-2)}$$

Appendix D contains the calculation of σ_c and Q_c for an EDF with n Gaussian terms and the derivation of the above equations.

With the invention of projection EBL machines (SCALPEL) in early 90s, GHOST is receiving greater interest as an inverse image of the pattern is readily available because of the nature of the image formation process in the system. Thus, it is possible to get the GHOST correction beam from the system with little extra effort. Liddle *et al.* [57, 58] investigated the feasibility of implementing GHOST in a projection EBL machine. The system produced a larger correction beam diameter than the ideal diameter. Successful implementation of GHOST in a projection EBL machine still remains to be demonstrated.

4.2.1 Performance indicators of GHOST

Using a 100kV EBL system and GHOST correction, Gesley *et al.* [59] demonstrated that patterns with a CD (critical dimension) value of 0.15μm can be written on X-ray masks with a line-width control of ±5%. The mask structure is made from a 0.3μm

---

[1] A spot radius is the distance from the beam incident point, at which the value of the charge density falls to 1/e of its peak value. It is similar to the definition of the characteristic length (section 3.2).

thick PMMA resist layer on top of a 0.3μm thick tungsten X-ray absorber layer which in turn rests on top of a 2.5μm thick silicon membrane. Without any PEC, the achievable CD with the same line-width control is 0.55 μm. For the same mask structure but with a 0.6μm thick PMMA and a 0.6μm thick tungsten layer, the achievable CDs with a line-width control of ±5% are 0.25μm and 1.6μm for cases with GHOST correction and without any PEC respectively. These higher CD values are due to a larger electron forward scattering range in the thicker resist and a higher back scattered electron contribution in the thicker tungsten layer.

Several studies [56, 60, 61] show that the allowable tolerances for correction dose (peak charge density), correction beam diameter and overlay accuracy between the patterning (primary) and the correction exposure are wide. The deposited energy density is found to be more sensitive to variations in the correction dose than variations in the correction beam diameter. The tolerance on overlay accuracy depends linearly on β and η [62]. The bigger β and the smaller η values give greater tolerance, implying the application of GHOST to higher beam energy EBL systems and multi-layer resists is much easier.

Equations (4-1) and (4-2) do not give the optimum values for the GHOST parameters [59, 61]. This is partly due to the inaccuracy of the model used in EDF. Peckerar *et al.* [63] showed that using an inaccurate EDF to determine GHOST parameters can result in significant errors in the resultant EDD. Kostelak *et al.* [61] and Gesley *et al.* [59] obtained the optimum values experimentally by varying the GHOST parameters on a test pattern. However, these optimum values are dependent on the chosen test pattern. To accommodate wider range of patterns, Muray *et al.* [64] used several test patterns in calculating the optimum GHOST parameters. The optimum values minimise the sum of line width deviation of those test patterns.

4.2.2 Advantages and disadvantages of GHOST

Clearly, GHOST is computationally trivial and the correction beam parameters, σ_c and Q_c (as calculated using equations (4-1) and (4-2)), are independent of written shapes.

The principal drawback of this technique is the need for an additional correction dose exposure per wafer, reducing significantly the throughput of an EBL machine. Projection EBL machines (SCALPEL) do not suffer from this problem as the primary and correction exposure is carried out simultaneously. GHOST also reduces resist side wall angle, resist thickness and developer latitude [65]. Using high contrast resists alleviate these problems. The loss in the developer latitude can be recovered with the use of a more dilute developer. Owen *et al.* [52] pointed out that GHOST reduces significantly the line width control (CD uniformity) of isolated lines. The degradation of CD uniformity by GHOST is also demonstrated by Cha *et al.* [66]. They also showed that CD linearity increases with resist of higher resist contrast. Kostelak *et al.* [61] demonstrated that using a correction dose greater than 40% of the patterning dose causes a serious pinhole problem in AZ-2400-17 resists. For a low contrast negative resist, GHOST creates a *scumming* problem in which some resist remains on unexposed regions in the resist profile.

As GHOST makes an implicit assumption that an infinite area is written, so there are errors at the boundary regions of the pattern layout due to the absence of back scattered energy from outside the layout. These errors can be eliminated by exposing an outer frame region surrounding the layout with the GHOST correction beam, at the expense of decreasing the throughput. This scheme does not compensate for the forward scattering effect.

4.3 Shape correction

A shape correction scheme modifies shape dimensions such that the influence of the proximity effect is to yield developed shapes with the desired dimensions in the resist. This technique is of special interest for electron projection systems that do not have the capability to change incident doses for each shape. By defining the value, $e_R(i)$ as the desired average deposited energy density in the ith shape, the proximity equation (3-2) becomes:

$$e_R(i) = \frac{kD}{A_i^d} \sum_{j=1}^{N} \int_{A_i^d} \int_{A_j^w} f(r) dA_j^w dA_i^d \qquad \textbf{(4-3)}$$

where D is the constant incident dose assigned to all shapes and N is the shape count. A_i is the area of the ith shape and its superscript values, d and w denote the desired and written area of the shape respectively. There are two ways to define the variables of such a system of equations (equation (4-3)) [67]:

1. All the dimensions of a shape are treated as variables. The system becomes *underdetermined* as the number of variables ($N \times$ the number of variables per shape) exceeds the equation count (N).
2. Only one variable per shape is allowed. For example the variable can be defined as the magnification factor, $(k_j)^2$ between the desired and written dimensions of the jth shape: $k_j^2 A_j^w = A_j^d$. Unfortunately, k_j is embedded in the limits of the 4-dimensional integral in equation (4-3), resulting in a system of coupled non-linear equations.

Evaluation of the required shape modifications needs a lengthy and costly computation.

In 1991, a group of researchers from Cornell University [68, 69] invented a hierarchical rule-based shape correction scheme called PYRAMID. This technique performs PEC in two parts:

- local correction;
- global correction.

The former part takes care of the proximity effect from adjacent shapes, whereas the latter corrects the proximity effect from the rest of shapes that are far away from the shape under correction. Since its invention, PYRAMID has undergone several modifications [70, 71, 72] to allow correction for a wider range of circuit patterns. Cook *et al.* [73] described the correction procedures involved in the latest version of PYRAMID.

PYRAMID corrects shapes on an individual basis, but the correction of a shape affects how other previously corrected shapes should have been corrected. This is called a *recursive effect*. The solution typically takes less than 4 iterations for convergence [73]. Lee *et al.* [74] suggested two ways to reduce this recursive effect. However, these methods do not guarantee improvement in the correction accuracy for arbitrary patterns.

4.3.1 Performance indicators of PYRAMID

Lee *et al.*[69] demonstrated that PYRAMID provides rapid correction for circuit patterns with a CD value down to 0.1μm. The corrected patterns have a line width control of ±10%. Correction of 131072 rectangles takes 203.2 seconds and 90.3 seconds on SPARC2 and HP700 workstations respectively.

In 1999, Lee *et al.*[75] developed a parallel version of PYRAMID that partitions a pattern into equal segments and distributes the segments to workstations in a network. This method of reducing the correction computation time is also applicable to other PEC schemes. The efficiency of the parallel program is defined as the ratio of speed up factor over the serial program to the number of workstations. Figure 4-2 shows the variation of the efficiency with the number of Sun Sparc10 workstations in a network for PYRAMID corrected large (≥200×200 μm^2) test pattern. The efficiency decreases as the number of workstations increases, indicating that it is not possible to increase the speed up factor indefinitely by ever increasing the number of workstations in the network. For small test patterns (50×50 μm^2), the computation time of the parallel program is higher than the serial program due to the relatively high communication overhead of the former program.

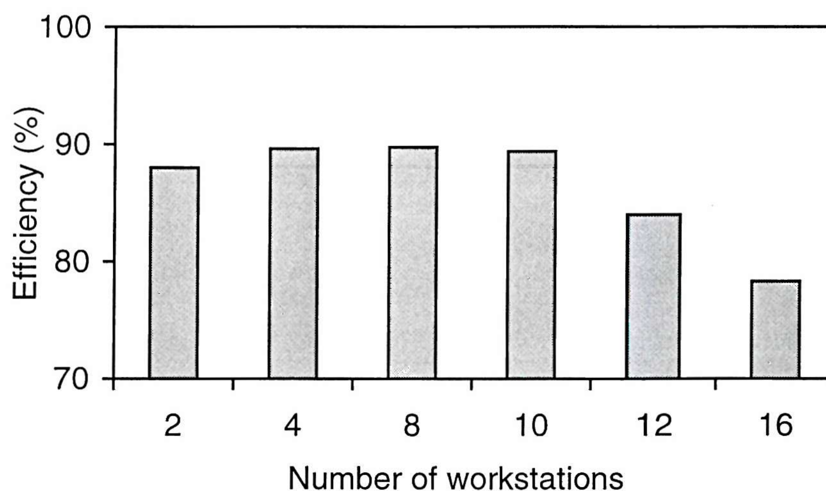


Figure 4-2 Variation of efficiency with the number of workstations.

Cook *et al.* [76] analysed the dependencies of the correction accuracy on the PYRAMID parameters. They concluded that the differences in the correction accuracy are not significant enough to play a role in choosing the parameter values.

4.3.2 Advantages and disadvantages of PYRAMID

As PYRAMID pre-computes all the correction calculations and stores them in look-up tables, the correction can be done rapidly. It is easily extended to correct patterns on heterogeneous substrates [72].

As PYRAMID is based on *ad-hoc* rules rather than a mathematical model, it is not general enough for a wide range of circuit patterns. Furthermore, shape adjustment reduces edge contrast (defined as the slope of a developed resist profile at shape edges [77]). When patterns contain both isolated and dense features, PYRAMID produces lower process latitude than dose correction schemes since the isolated features are under-dosed while the dense features are overdosed.

Since the magnitudes of the required shape changes are small fractions of the CD value, so fine address spacing is essential and even the minimum size shapes must be composed of several scans of a small beam. Thus, this method is not applicable to patterns with a CD value close to the smallest beam size.

4.4 Dose correction

A dose correction scheme adjusts incident doses in each shape so that the influence of the proximity effect is to yield developed shapes with the desired dimensions in the resist. This correction scheme is the most popular among all PEC techniques. Unlike the GHOST and the shape correction methods which are applicable to all types of EBL machines, an efficient implementation of a dose correction scheme on an EBL machine requires the machine to have the ability to vary the incident dose on a shape/exel basis (dose *modulation*).

By selectively multiply exposing a pattern, a dose correction method can be implemented on an EBL machine without dose modulation capability (*e.g.* a

projection EBL machine) [78, 79]. However, this approach reduces significantly the throughput of the EBL machine and is prone to alignment errors between passes.

4.4.1 Self-consistent method

The proximity equation (3-2) can be expressed in a matrix form:

$$E_R = FD_I \qquad \textbf{(4-4)}$$

where E_R and D_I are column vectors, and F is an adjacency matrix derived from the point spread EDF. Each corresponding element in E_R and D_I contains the deposited energy density at an exel centre and the incident dose applied to the entire exel site respectively. Each element, F_{ij}, represents the fraction of energy deposited at the centre of exel j from a unitary dose applied to the entire site of exel i. Thus, the correction problem reduces to solving equation (4-4), involving the inversion of F.

There are other ways of deriving the self-consistent system of linear equations described by equation (4-4). M. Parikh [80] defined the values in E_R as the desired average deposited energy density in each shape. Hence, calculating the values of F matrix elements involve double integrals of the EDF. This method disregards nominally unwritten regions. On the other hand, Song *et al.* [81] set the values in E_R to be either the desired deposited energy density at a shape corner or the desired average deposited energy density along a shape edge or in a shape, depending on how the shape is fractured.

Alternatively, values in E_R can represent the desired deposited energy density at various sample points in a pattern. Equation (4-4) as defined in the first paragraph is just a special case of this approach, in which the sample points are the centre points of each exel site. Because each shape can have only one incident dose value, if there are n (n>1) sample points per shape, then we have an over-determined system of linear equations. Such an over-determined system can yield only an approximate least-square solution. Harafuji *et al.* [82] invented a fast iterative method for solving such over-determined systems and Lu *et al.* [83] refined this technique. The iterative method uses the following termination condition:

$$\frac{d_I^{(n+1)}(i) - d_I^{(n)}(i)}{d_I^{(n)}(i)} \leq \varepsilon_{crit}$$

where $d_I^{(n)}(i)$ is the incident dose applied to exel i and the superscripts denote the number of iterations. For $\varepsilon_{crit} = 8 \times 10^{-3}$, the technique takes less than 5 iterations for convergence.

For a pattern consisting of alternating lines and spaces (a line/space pattern), such as those found in memory cells of DRAM, we can reduce the 2-dimensional integral to 1-dimensional integral when computing the element values in F [84]. Using polar co-ordinates instead of Cartesian co-ordinates, Fabrizio *et al.* [85] and Grella *et al.* [86] lowered the 4-dimensional integral (Parikh's method) to a 2-dimensional integral for concentric ring shapes found in *Fresnel zone plates* (diffractive optical elements). These simplifications reduce the computational cost significantly.

4.4.1.1 Performance indicators of the self-consistent method

As the electron back-scattering range is a few micrometres, and circuit pattern (chip) dimensions are usually on the order of millimetres, so F is a highly banded matrix. When an infinite region is exposed with a constant incident dose, the exposure outside a 4β (β is the electron back-scattered characteristic length.) wide square contributes less than 1% of the total back-scattered electron energy density deposited at the centre of the square, which is negligible. Thus, we can ignore those elements in F, corresponding to exels outside the square of side length 4β centred on an evaluation exel. The total number, T of non-zero elements in F is roughly equal to the product of the exel count in the square and the size of E_R, N:

$$T \approx 16\left(\frac{\beta}{\Delta}\right)^2 N$$

where Δ is the exel spacing. Inverting such a banded matrix needs about $N \times T$ arithmetic operations [87]. Hence, in inverting F, the required number of operations is proportional to

$$P = 16\left(\frac{\beta}{\Delta}\right)^2 N^2 \qquad \textbf{(4-5)}$$

For a 10 x 10 mm^2 chip size, Δ =100 nm and β = 10 μm[1], N and P are equal to 1.00×10^{10} and 1.60×10^{25} respectively. Using a one tera-FLOPS (10^{12} floating point operations per second) supercomputer, this would take a prohibitive 5.07×10^{5} years to compute the inversion.

To reduce the intensity of computation requirement, M. Parikh [88] introduced a zoning algorithm that partitions a chip area into a number of smaller regions of dimension $s \times s$, such as the square labelled ABCD in Figure 4-3. When evaluating the incident dose distribution in each small area, the algorithm uses shapes within the corresponding larger square (such as abcd) in order to take into account the influence from shapes in the border regions.

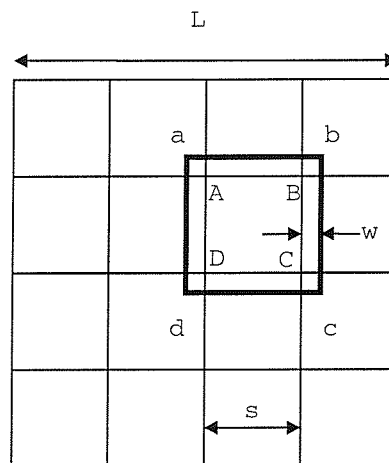


Figure 4-3 Division of a chip area into smaller areas.

If the chip dimensions are $L \times L$, then the number of small regions such as ABCD within the pattern is $(L/s)^2$ and the number of exels within abcd is $((s+2w)/\Delta)^2$. Thus, using equation (4-5) the number of arithmetic operations needed for the matrix inversion is:

$$P = 16\left(\frac{\beta}{\Delta}\right)^2\left(\frac{s+2w}{\Delta}\right)^4\left(\frac{L}{s}\right)^2$$

By differentiating this expression with respect to s, we find that P is a minimum when $s = 2w$. For $w = 2\beta$, the maximum reduction factor of this partitioning algorithm is therefore equal to:

[1] This is a typical value for a Silicon substrate at a beam acceleration voltage of 50kV.

$$\frac{16\left(\frac{\beta}{\Delta}\right)^2 N^2}{16\left(\frac{\beta}{\Delta}\right)^2\left(\frac{8\beta}{\Delta}\right)^4\left(\frac{L}{4\beta}\right)^2} = \frac{N^2}{\left(\frac{8\beta}{\Delta}\right)^4\left(\frac{L}{4\beta}\right)^2}$$

$$= \frac{N^2}{256\left(\frac{\beta}{\Delta}\right)^2\left(\frac{L}{\Delta}\right)^2}$$

$$= \frac{N}{256\left(\frac{\beta}{\Delta}\right)^2} \text{ as } N = \left(\frac{L}{\Delta}\right)^2$$

For $w = 2\beta$ and the same parameters as in the first paragraph of this section, the one tera-FLOPS supercomputer will take 1.30×10^2 years for the matrix inversion.

Parikh's method reduces the computation time needed for the matrix inversion by a factor equal to the inverse of the square of the fraction of written exels in the pattern, without using the zoning algorithm. Using the zoning algorithm, the reduction factor is equal to the inverse of the fraction of written exels in the pattern.

Using a 50kV EBL system and the self-consistent method, Nakajima *et al.* [89] fabricated a 4 Gbit DRAM with a line width control of ±5%. The DRAM has a minimum feature size of 0.15 μm.

4.4.1.2 Advantages and disadvantages of the self-consistent method

This PEC technique compensates for the forward scattering effect.

Unfortunately, the method is computationally intensive and requires huge storage spaces. As mentioned in section 4.1, the proximity effect correction problem is ill-posed. Thus, such large matrices can be ill-conditioned, making a solution impossible. If unwritten regions are included in equation (4-4), its inversion usually leads to negative incident doses which cannot be realised in practice. The higher the ratio of the desired deposited energy density in nominally written to unwritten regions, the greater the chances of getting negative incident doses in the solution.

Because of the intensive computational nature of the conventional matrix inversion techniques, other approaches have been extensively researched. These include a hierarchical approach, Fourier pre-compensation proximity effect correction (FPPEC), neural network approaches, RFPEC and PADPEC dose correction methods. These all attempt to overcome the problem at the expense of correction accuracy.

4.4.2 The hierarchical approach

One possible way to bring the computation time of the self-consistent method down to a reasonable level is to exploit the hierarchical structure of a circuit pattern. Hierarchy processing algorithms are based on cell-bound operators. Unfortunately, PEC is not confined to a cell. Therefore, an abutting shape of a neighbouring cell may partially destroy the PEC result of a given cell. Consequently, the cell boundaries need recalculation. The hierarchical structure most convenient and natural from the designers' point of view is frequently not the best from the standpoint of attacking the PEC problem.

Commercially available PEC software, CAPROX (Computer Aided PROXimity correction) [90, 91], reorganises the original hierarchy structure for optimum hierarchical PEC. CAPROX uses a set of rules for the hierarchy reorganisation. These rules aim to produce a hierarchy structure that minimises recalculations in cell boundaries and the memory requirements of CAPROX. CAPROX is developed by research groups from Sigma-C GmbH, Germany and OKI Electric Industry Corporation, Japan.

Misaka *et al.* [92, 93] created a two level hierarchical PEC system. The system employs an inner-outer frame technique to account for the proximity effect between shapes in the two hierarchy levels. Their PEC system uses more than one sample point per shape, resulting in an over-determined system of linear equations which is solved by Harafuji's algorithm (see section 4.4.1).

4.4.2.1 Performance indicators of the hierarchical approach

For a 64 Mbit DRAM device, Misaka's method reduces the final data volume to between 1/15 and 1/100 of the data volume of a hierarchically-flattened PEC system.

The correction time for each layer is less than 4 hours (layer A = 1.00 hour, layer B = 3.85 hours, layer C = 2.00 hours, layer D = 0.92 hours) using a general-purpose computer capable of executing 10 MIPS (Mega-Instructions Per Second) and a vector processor with a peak value of 500 MFLOPS. The corrected DRAM device has a line width accuracy of 0.04μm and a nominal CD value of 0.4μm.

CAPROX defines the *hierarchy factor* as the ratio of the area occupied by the flattened layout to the area occupied by cells in the hierarchy tree. The hierarchy factor is a measure of the degree of hierarchy. Table 4-1 and Table 4-2 show CAPROX correction times on a SPARCstation 20 for different layers of DRAM devices and logic devices respectively. From Table 4-2, CAPROX is about 15.8 and 20.5 faster than the conventional self-consistent PEC method (*flat* PEC) for logic devices with low and high degrees of hierarchy respectively.

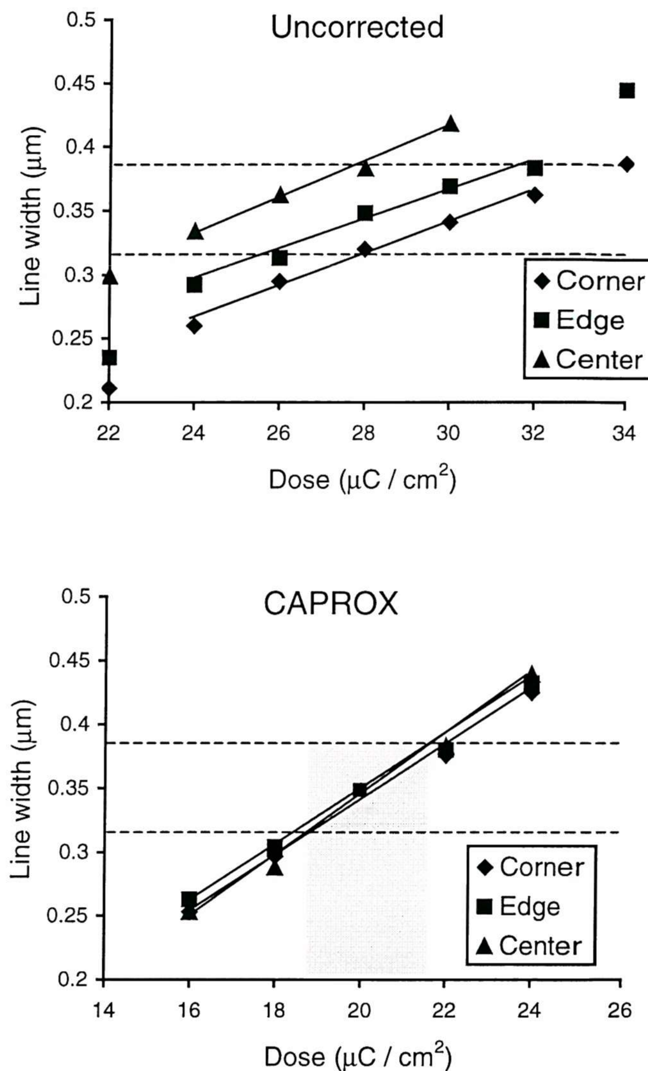
		64 Mbit DRAM	**256 Mbit DRAM**	**256 Mbit DRAM**
Layer		Word Line	Bit Line	Active
Initial Layout	Hierarchy Factor	90000	2943	385
	File Size (MBytes)	0.014	0.150	0.530
After Correction	Hierarchy Factor	38000	946	66
	Time	19min	4h 4min	6h 31min
	File Size (MBytes)	0.6	29.0	20.0

Table 4-1 Layout characteristics of different layers of DRAM devices [94].

		Initial CAD layout	**CAPROX**	**'flat' PEC**
Deep Hierarchical Logic Device	Hierarchy Factor	46.45	2.74	1.00
	File Size (MBytes)	0.65	10.20	26.70
	Time (h:min:s)		0:25:35	6:35:05
Low Hierarchical Logic Device	Hierarchy Factor	2.32	1.16	1.00
	File Size (MBytes)	8.60	137.00	500.00
	Time (h:min:s)		9:46:42	200:00:00

Table 4-2 Layout characteristics of logic devices [95].

Figure 4-4 shows that only for an incident dose of $28\mu C/cm^2$, the line width variation of the active layer of a 256 Mbit DRAM device is within 10% for the uncorrected case. On the other hand, the CAPROX corrected case has a large process window that is the shaded region in Figure 4-4 for the same line width control. Using the optimum incident dose, the line width variation of the corrected active layer is roughly ±1% from the nominal CD value of 0.35μm.



The DRAM device is fabricated on a silicon substrate coated with a 0.4 μm thick SAL601 (negative resist) using a 20 kV AEBLE150 EBL machine [94].

Figure 4-4 Line width variation versus incident dose.

4.4.2.2 Advantages and disadvantages of computer aided proximity correction (CAPROX)

Because CAPROX is derived from the self-consistent method, so CAPROX inherits its advantages and disadvantages, except that CAPROX is much faster and needs less memory space. As the number of repetitive patterns (i.e. hierarchy factor) in a layout decreases, the speed up factor of CAPROX over flat PEC reduces. Hence, CAPROX works well for memory devices, but it is less efficient for random patterns.

4.4.3 Optimisation approach

As mentioned in section 4.1, the proximity equation (3-2) can be ill posed, resulting in adjacency matrix singularity. One possible way of dealing with this problem is to recast equation (4-4) as an optimisation problem. Carroll [96] appears to be the first person to apply this approach in PEC.

To convert the PEC problem into an optimisation problem, we start by forming a cost function:

$$C(D_I) = (FD_I - E_R)^2 \qquad \textbf{(4-6)}$$

The above equation can be viewed as a quadratic function of F. Obviously, when $FD_I = E_R$, $C(D_I)$ is zero and we have a solution to the problem. Even when no solution is possible, this approach gives an optimum solution for incident doses of individual exels in a least-square sense. By viewing it this way, the problem reduces to minimizing some quadratic function that has at most a single well-defined minimum.

Having defined the cost function, a gradient descent method can be employed to solve the optimisation problem. Here we simply assign some starting value for D_I, take the gradient of the cost function, and vary D_I by moving along the direction of the gradient vector until the cost function reaches a minimum. In each iteration, the incident dose adjustment at i^{th} exel is given by the following equation expressed in discrete notation:

$$\Delta D_{I_i} = -(\sum_{j=1}^{N}\left[F_{ij}D_{I_j}\right] - E_{R_i}) \qquad \textbf{(4-7)}$$

where N is the total number of exels. This is the basis of the approach described by Groves [97]. However, this method does not eliminate the problem of negative incident doses in the solution. To overcome this problem, we can simply reset D_I to zero if it goes negative. Marrian *et al.* [98] term this algorithm as local area dose correction (LADC). Although the algorithm is simple, it is hard to quantify the achieved optimisation in a mathematical sense.

Another way of constraining the solution to positive numbers is to include a regularizer in equation (4-6). This regularizer can be any function that is small when D_I is positive and large when it approaches negative values. Marrian *et al.* [99] chose the informational entropy as the regularizer:

$$S(D_I) = \sum_{i=1}^{N}\left(\frac{D_{I_i}}{D_T}\ln\left(\frac{D_{I_i}}{D_T}\right)\right) \qquad \textbf{(4-8)}$$

where D_T is the sum of all incident doses. Using the method of Lagrange multipliers, the new cost function for the constrained optimisation becomes

$$C'(D_I) = C(D_I) - \psi S(D_I) \qquad \textbf{(4-9)}$$

where ψ is a constant. This is called the maximum entropy (ME) approach and uses the gradient descent algorithm to solve the above equation iteratively.

Rau *et al.* [100] converted the PEC problem into a non-linear, convex and constrained optimisation problem by assigning a weighting factor to each exel and defining a convex error function as the cost function. The error function incorporates a multiple Gaussian EDF and a resist characteristic function that relates incident doses to remaining resist thickness after the resist development process. They implemented the non-negativity constraint of D_I by projecting D_I to the closest set of non-negative points at the end of each iteration. In evaluating the cost function, they convolved D_I with cascaded uniform filters [101, 102] to approximate the convolution between D_I and the multiple Gaussian EDF. Their algorithm is a hybrid of conjugate gradient and gradient projection algorithms.

4.4.3.1 Performance indicators of the optimisation approach

For the test pattern used in [100], Rau's method takes roughly 150 iterations for the solution to converge. On the other hand, Marrian *et al.* [103] found ME produced a converged solution after about 100 iterations for their test patterns. They also discovered empirically that ψ should be set as small as possible with values ranging from 0.002 to 0.02 being appropriate to their test patterns. The LADC approach needs fewer iterations and is computationally less intensive.

Marrian *et al.* evaluated the correction accuracy of LADC and ME using a quality factor, R. For each exel, if the resultant deposited energy density in the exel after PEC is less than 90% or greater than 10% of the desired deposited energy density for a nominal written or an unwritten exel respectively, then R is increased by one divided by the total number of exels; otherwise the value of R remains unchanged. Table 4-3 shows R values of the space/line/space feature in the tower pattern (Figure 4-5). R is expressed as a function of pad width of the tower pattern. Electron forward scattering ranges are 10nm and 50nm for beam acceleration voltages of 50kV and 20kV respectively [104]. For the 20kV beam voltage case, in which the forward scattering effect is significant, the results indicate that ME is superior to LADC in compensating for the forward scattering effect.

	Pad Width (μm)	Uncorrected (%)	LADC (%)	ME (%)
50kV	>3.30	17	15	13
	1.25	22	10	10
	0.63	22	7	7
	0.31	26	4	4
	0.13	21	2	2
20kV	>0.30	24	20	15
	0.13	23	19	16

Table 4-3 The quality factor (R) values for the space/line/space feature through the tower pattern.

The correction error in Rau's approach does not increase significantly because of the quantization error in digitising D_I into a finite number of levels supported by a practical EBL machine. However, the correction accuracy of this method depends strongly on the forward scattering characteristic length (α) which is difficult to

determine precisely. A 20% error in α can increase the correction error by a factor of about 3 [100].

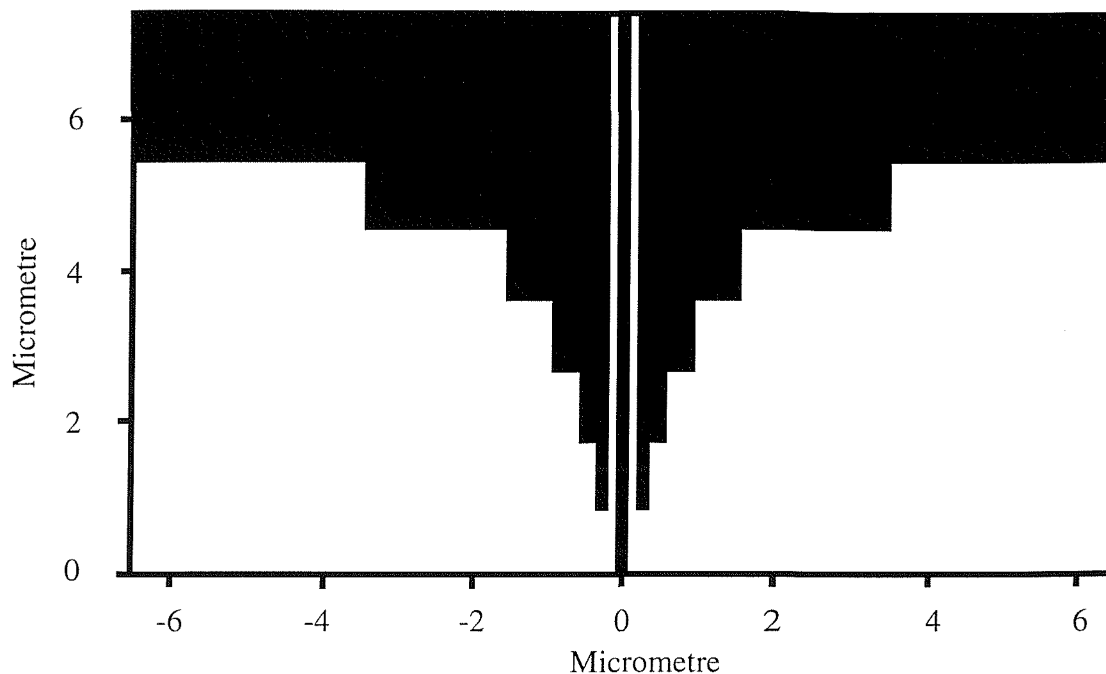


Figure 4-5 Tower pattern.

The ME approach is computationally intensive: it takes several hours to correct a 40,000 exel pattern on a personal computer with a 33 MHz 486 processor. For correcting realistic data, the technique is obviously impractical even using supercomputers. However, Marrian *et al.* [103] estimated that a parallel computer with ten thousand processors could process 10^8 exels in 3 hours. Each processor is a 8-bit processor and runs at 100 MHz. In [105], the design of a ME neural net co-processor is proposed that can reduce the computation time from the previous estimate of 3 hours to 62 minutes.

4.4.3.2 Advantages and disadvantages of the optimisation approach

Although optimisation techniques can constrain a solution to non-negative values and deal with the adjacency matrix singularity problems, they are computationally very expensive.

4.4.4 The Fourier pre-compensation (de-convolution) proximity effect correction (FPPEC) method

Kern [106] seems to be the first person to recognise that EBL is a form of incoherent imaging and hence classical methods in image processing can be used to correct the proximity effect. In this technique, an incident dose distribution, $d_I(r)$ is Fourier pre-compensated as follows:

$$d_I(r) = F^{-1}\left(\frac{F(e_R(r))}{F(f(r))}\right)$$

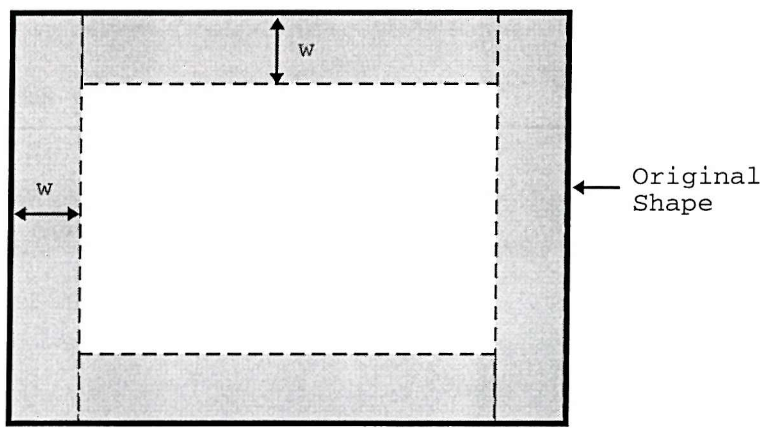
where $F^{-1}(x)$ and $F(x)$ are the inverse and forward Fourier transforms of x. The Fourier transforms are computed using the fast Fourier transform (FFT) method. In essence, FPPEC performs matrix inversion of equation (4-4). However, the solution contains too many details, resulting from rapid oscillations, and this tends to increase the database undesirably. Filtering out high frequency components in $f(r)$ or using a Walsh transform or Haar thinning algorithm alleviates the problem, at the expense of solution accuracy. Chow *et al.* [107, 108] describe this technique in detail.

In 1993, a group of researchers from Germany [109, 110] made several improvements to Kern's method and produced a commercial PEC package, PROXECCO (PROXimity Effect Correction by de-Convolution). They split the calculation of $d_I(r)$ into a correction related and a pattern related step which are the first and second terms respectively in the following expression:

$$d_I(r) = F^{-1}\left[F(e_R(r)) \times \left(\frac{h}{F(f(r))} - C\right)\right] + C \times e_R(r) \qquad \textbf{(4-10)}$$

where C is a constant and h is a filter transfer function. The filter output has a value of C for high frequencies and has identical values to its input for low frequencies. The high frequency components in $F(f(r))$ are due solely to the forward scattering term in $f(r)$. Since *F(delta function)* is constant in the frequency domain, so the filtering effectively converts the electron forward scattered component of $f(r)$ into a delta function. The filtering eliminates the rapid oscillations in the solution mentioned previously. Because the operand of the first term in equation (4-10) lacks high frequency components, the Fourier transforms in the correction step can be evaluated

using a coarse grid without sacrificing the correction accuracy significantly. This improvement enables PROXECCO to perform the correction much faster than Kern's method, which needs a fine grid to preserve pattern resolution. PROXECCO corrects forward scattering using a framing procedure (frame correction) [109]. In the procedure, each shape is partitioned into a peripheral part (called a frame) and a central rectangle as shown in Figure 4-6. PROXECCO assigns higher incident doses to frames than the central rectangle. (The theoretical background of the frame procedure will be described later in section 5.2.1.2.)



The shaded regions in the diagram are the frames.

Figure 4-6 Shape framing procedure.

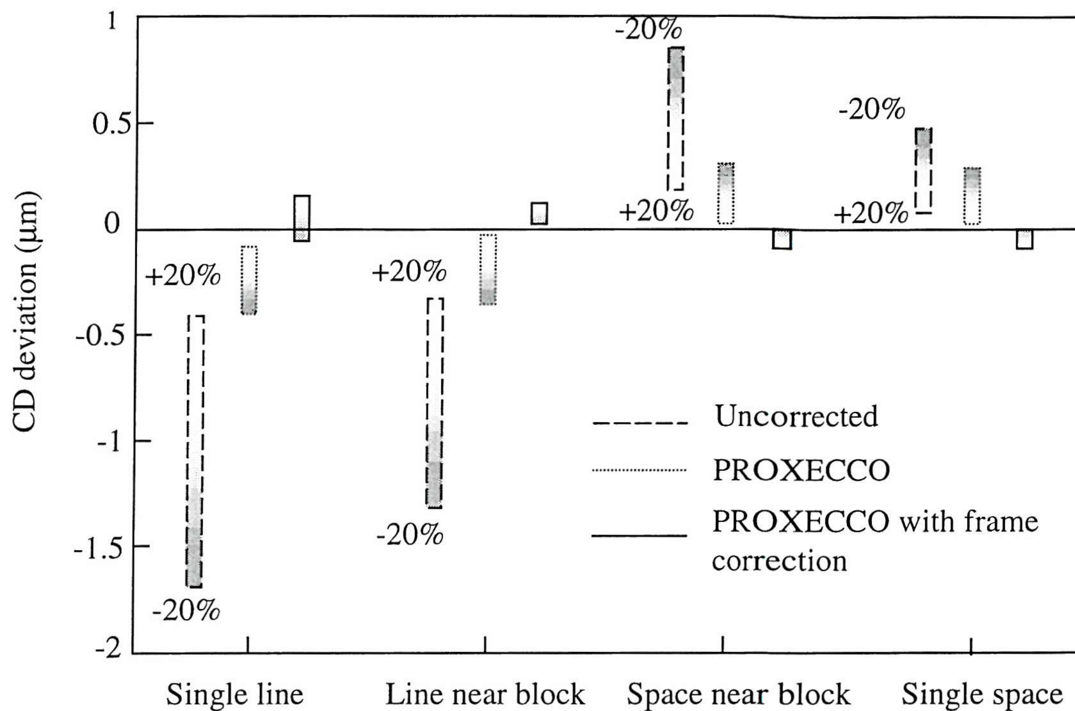
4.4.4.1 Performance indicators of FPPEC

Experimental results (Figure 4-7) on an ASIC (Application Specific Integration Circuit) pattern with a CD value of 1.2 μm show PROXECCO with the frame correction has good tolerances of incident dose deviations. PROXECCO with the frame correction takes about an hour to correct the pattern, which has approximately one million polygons (shapes). Simulation results from [109] demonstrate that PROXECCO gives an error of 40nm to patterns with a CD of 1μm.

The most time consuming steps are the Fourier transform operations in FPPEC. Because FFTs are used, the number of operations, P, for each transform is proportional to

$$P = N \log_2 N$$

where N is the total number of exels or the sample points for the conventional FPPEC method and PROXECCO respectively. Since FPPEC involves two forward FFTs and one inverse FFT, so the total number of operations is of the order of $3N\log_2 N$ which is much smaller than $4(\beta N / \Delta)^2$ (equation (4-5)) required by the conventional matrix inversion techniques.



The incident dose ranges from -20% to +20% of its ideal value.

Figure 4-7 CD deviation versus incident dose [110].

4.4.4.2 Advantages and disadvantages of FPPEC

This technique takes full advantage of fast special hardware that can be used to implement the FFT algorithm. If the ratio (E_u / E_w) of the desired deposited energy density at unwritten to written exels is very small, then FPPEC might produce unphysical solutions with negative incident doses. The cause of these negative values is that it is physically impossible to get E_u in a real pattern.

Although FPPEC appears to be computationally straight forward, its practical implementation is not so. For example, pattern fracturing will be needed to bring the memory requirement down to a reasonable size. Unless it is done carefully, the partitioning can produce errors when reconstructing the corrected pattern.

4.4.5 Neural network approach

For each exel, the solution of the proximity equation (3-2) can be expressed as a convolution between the desired deposited energy density distribution, $e_R(r)$ and a correction *kernel*, $g(r)$:

$$d_I(r) = e_R(r) * g(r)$$

The correction kernel of each exel differs slightly from one to another. In their PEC schemes, Lynch *et al.* [111], R. C. Frye [112] and P. Jedrasik [113] assumed that all the correction kernels are the same. Lynch *et al.* computed the kernel analytically from an approximate solution of the self-consistent correction scheme, while the last two authors used neural networks to generate the kernel. A set of pre-corrected patterns is used to *train* the neural network, in which the weights of the kernel are adjusted adaptively using the gradient descent method. The sizes of the training patterns are comparable to the characteristic length of electron back scattering, β. Frye's training patterns are corrected using an optimisation approach, whereas Jedrasik used the FPPEC method to correct the training patterns.

4.4.5.1 Performance indicators of the neural network approach

For the conventional matrix inversion, the computational cost is the product of the exel count in a circuit pattern and the *cost* of convolution (equation (4-5)). On the other hand, the cost of the neural network approach is the sum of the cost of performing the conventional matrix inversion on the training, generating the kernel and the convolution cost. Since the training pattern sizes are much smaller than the circuit pattern sizes, so the convolution cost is the dominant component. The range of the kernel is comparable with the electron scattering range, so the convolution cost is roughly equal for both schemes. Thus, the neural network approach is approximately faster by a factor equal to the exel count, which is extremely large. Frye claimed a speed up factor of 10,000, directly attributable to this approach, while Jedrasik estimated his system is about 3000 times faster than the conventional matrix inversion method.

Compared with the optimisation scheme, Frye estimated the neural network is typically accurate to within roughly 1% on average and 5% at worst. Using a Leica EBMF 10.5 EBL machine and the neural network approach, Cummings *et al.* [114] have successfully written a test pattern with a CD value of 1 μm on a Silicon substrate covered with a 0.5 μm SAL601-ER7 resist. The CD deviations of the corrected pattern were at or below 50 nm, while the deviations of the uncorrected pattern is 0.1 μm.

4.4.5.2 Advantages and disadvantages of the neural network approach

This technique is much faster and needs less memory space than the conventional matrix inversion approach. However, these gains are achieved at the expense of correction accuracy. The correction accuracy of the method depends on the training patterns and the PEC scheme used to generate the patterns.

4.4.6 The representative figure proximity effect correction (RFPEC) method

In 1991 [54], a research group from the ULSI Research Centre, Toshiba Corporation presented an idea of representing shapes in a $2\Delta \times 2\Delta$ area with a rectangle. 2Δ is smaller than the electron back scattering characteristic length, β, and the representative rectangle has the same area and centre of gravity as the original shape. The forward scattering characteristic length is assumed to be negligibly small compared with the CD value.

The RFPEC system computes an optimum incident dose distribution, $d_I(r)$ using a modified Pavkovich's formula [115] as follows:

$$d_I(r) = \frac{D}{1 + 2\eta U(r)}$$
$$U(r) = \frac{1}{\pi} \int_{\text{written shapes}} e^{-\frac{(r-r')^2}{\beta^2}} dr' \qquad \textbf{(4-11)}$$

where D is a constant and its value depends on the specific resist and the beam voltage used. $U(r)$ is evaluated using the representative rectangles of the actual circuit pattern. The modified Pavkovich's formula aims to equalise the average deposited energy density in written and unwritten exels $((E_u + E_w) / 2)$, instead of equalising E_w as in the original formula.

Figure 4-8 depicts the correction procedures involved in RFPEC [116]. The following is a brief precis of the correction procedure as reported in [116]:

1. RFPEC evaluates the representative rectangle in each $2\Delta_r \times 2\Delta_r$ region (called a reference region). $2\Delta_r$ is roughly equal to β.
2. Using equation (4-11), RFPEC computes the optimum incident dose at the centre of each smaller $2\Delta_e \times 2\Delta_e$ region (called an element region). $2\Delta_e$ is smaller than β (typically $2\Delta_e \leq 0.556\beta$, the maximum allowable value for a CD control of ±10% [121]).
3. RFPEC fractures the original shapes along the boundaries of the element regions into *sub-shapes*. Finally, it assigns the calculated incident dose of each element region to sub-shapes in the region.

The research group applies the representative figure idea to the GHOST scheme to increase the throughput of an EBL machine. Instead of exposing an inverse pattern image with a correction beam, their system, called Phantom, exposes the representative rectangles of the inverse pattern image. Phantom reduces the correction exposure time in GHOST by a factor of 13 [117, 118].

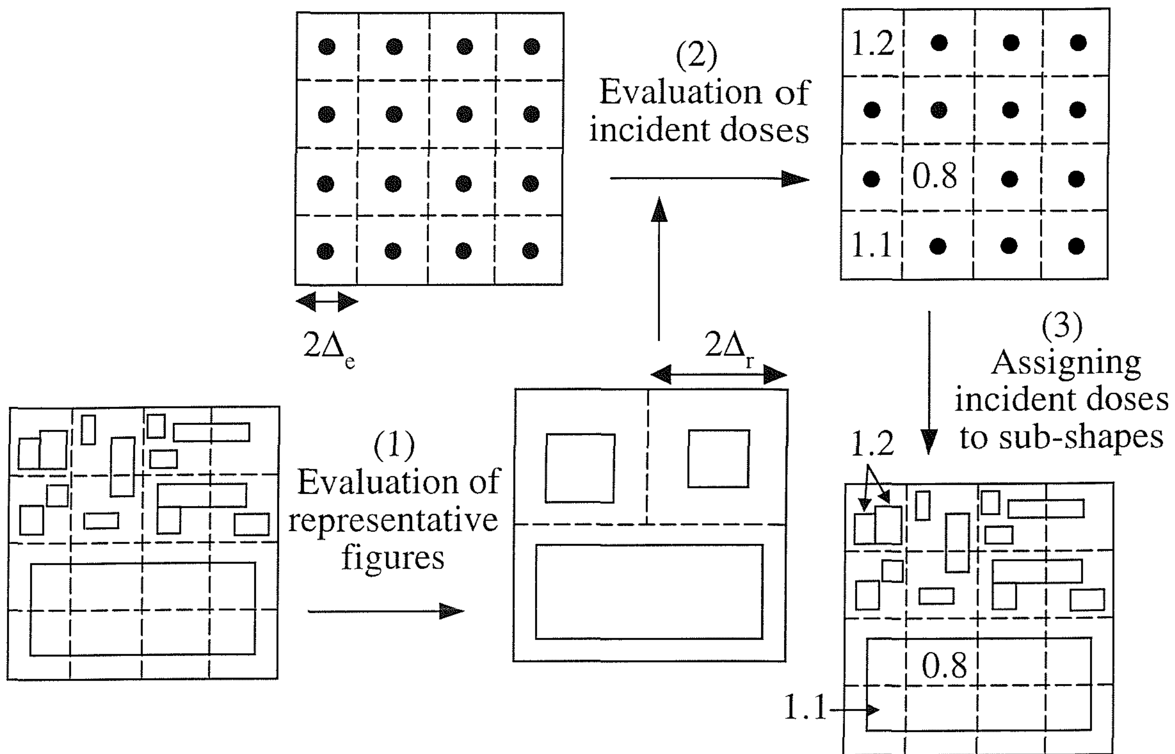


Figure 4-8 Correction procedures of the RFPEC system.

4.4.6.1 Performance indicators of RFPEC.

There are four sources of error in RFPEC:

- *E error*[1] which is due to a uniform incident dose applied to shapes within each element region;
- *Intrinsic error* which is the difference between the *U(r)* values computed using the representative rectangles and the values calculated using the original circuit pattern;
- Error in Pavkovich's formula;
- *F error*, which is caused by fixing the evaluation point of the optimum incident dose at the centre of each element region, instead of at the centre of gravity of the original shapes in the region.

[1] The E error is known as edge error in [119-121]. Here, the term *E error* is used because the term *edge error* is used to describe a different quantity in Chapter 5.

Table 4-4 shows the accuracy of Pavkovich's formula decreases as η increases. T. Abe [119] demonstrated that the E error exceeds the F error.

η	Maximum dose error (%)	Maximum deposited energy density error (%)	Edge deposited energy density error (%)
0.75	2.60	2.10	0.75
1.00	3.50	2.80	0.37
1.25	4.50	3.60	0.20
1.50	6.20	4.20	0.86

Table 4-4 Comparison between Pavkovich's approximate solution and the numerical solution for the case of an isolated edge for increasing values of η [115].

Abe *et al.* [119, 120] estimated the maximum intrinsic error and the maximum E error as:

$$Error_{\text{intrinsic}} = \pm\frac{2}{3}\lambda\eta\left(\frac{\Delta_r}{\beta}\right)^2 \exp(-1)$$

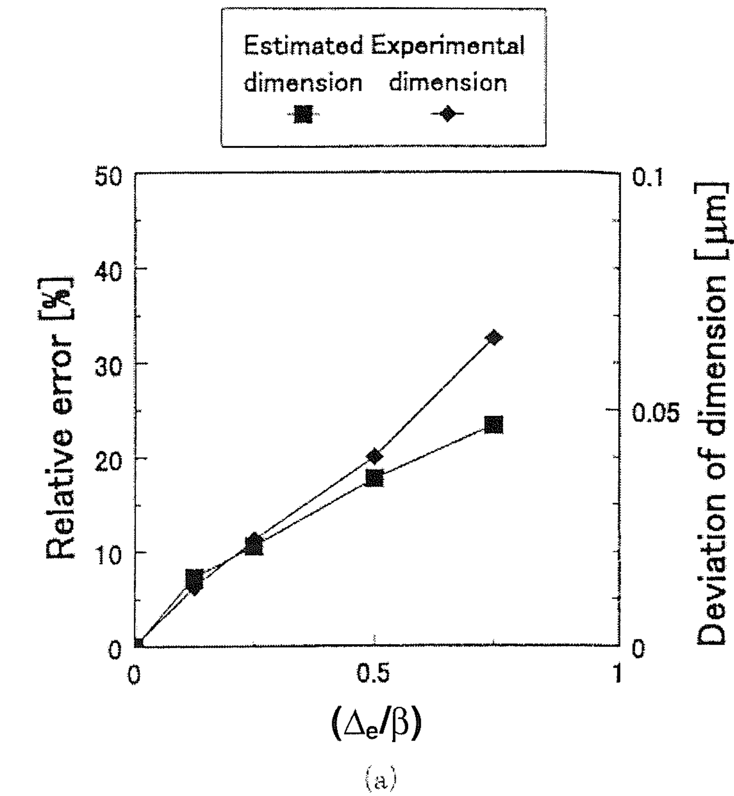
and

$$Error_E = \pm\lambda\left[\frac{\eta\Delta_e}{\beta\sqrt{\pi}(1+0.5\eta)}\right]$$

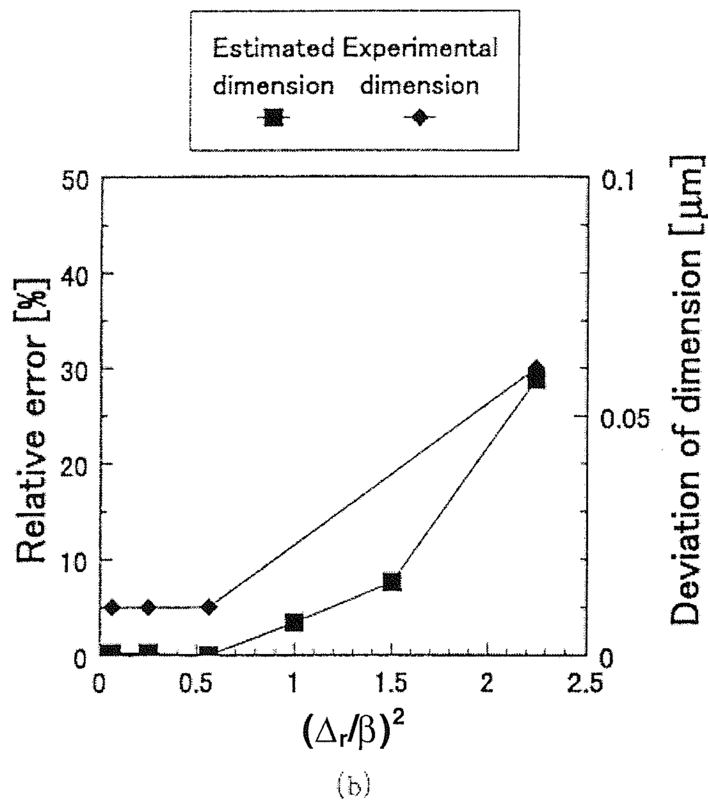
respectively, where λ is the pattern area density[1] in the region. Experimental results (for a line/space pattern with equal width of 0.2μm) [121] shown in Figure 4-9 confirm the dependency of the intrinsic error (Figure 4-9(b)) and the E error (Figure 4-9(a)) with $(\Delta_r/\beta)^2$ and (Δ_e/β) respectively. The results also indicate that the maximum size for $2\Delta_r$ and $2\Delta_e$ are 15μm (1.67β) and 5μm (0.556β) respectively for a line width control of ±10%. When these sizes are used, the correction time is about 15 minutes for a 15 × 15 mm^2 test pattern with CD = 0.2μm, using a parallel computer system with four processors capable of executing 50 MIPS each [121]. [54] contains description of the test pattern.

---

[1] The pattern area density is defined as the ratio of the written area to the total area in a region.



(a)



(b)

Figure 4-9 Comparison between experimental and simulation results.

Oogi *et al.* [122] have developed a dedicated hardware system that computes the value of $U(r)$ at the centre of each element region (equation (4-11)). The system consists of 25 processing units and has a throughput of about 2×10^7 $U(r)$ values per second.

4.4.6.2 Advantages and disadvantages of RFPEC

Because each reference region always has one representative rectangle regardless of the λ value in the region, so the evaluation time of the optimum dose is independent of the λ value. This technique is fast as it computes incident doses on an element region basis, instead of on an exel basis. Since the values of $U(r)$ are always positive, so the values of the optimum doses are also always positive.

Since the calculated incident doses are fairly constant in the centre of large rectangles, so it is unnecessary to partition them along element region boundaries. Thus, the pattern fracturing employed in RFPEC is not optimum. This system does not compensate for the forward scattering effect.

4.4.7 The pattern area density proximity effect correction(PADPEC) method

This method evolved from the observation that the deposited energy density, E_b by back-scattered electrons in small region within the back-scattered characteristic length (β) is almost constant for high voltage beams (≥50kV). This section contains a brief precis of the PADPEC method described in [55].

1. The circuit pattern is partitioned with a fixed sized mesh. The mesh size (M) is chosen such that the variation of E_b within a single mesh site is negligible (M is thus a function of the beam energies).
2. λ, (the pattern area density) is computed in each mesh site, giving a λ map for the circuit pattern.
3. The λ map is convolved with a filter, giving a smoothed λ map, λ^{sm} *map*.
4. Each circuit shape (polygon) is fractured into rectangles; and a λ value (called λ') is assigned to each rectangle. Taking the position of the rectangle to be its centre,

λ' is the linear interpolation of the λ^{sm} values of the four nearest mesh sites to the rectangle (*bilinear interpolation*).

5. Finally, an incident dose

$$D = \omega \frac{2(1+\eta)}{1+2\lambda'\eta} \qquad \textbf{(4-12)}$$

is assigned to each rectangle. ω is a constant and its value depends on the specific resist and the beam voltage used.

Two types of filters have been used to smooth the λ map.

1. Murai *et al.* [55] used a form of template convolution; λ^{sm} is given by:

$$\lambda^{sm}(i,j) = \sum_{l=i-\frac{R_{sm}-1}{2}}^{i+\frac{R_{sm}-1}{2}} \sum_{m=j-\frac{R_{sm}-1}{2}}^{j+\frac{R_{sm}-1}{2}} a(l,m)\lambda(l,m) \qquad \textbf{(4-13)}$$

where

$$\begin{aligned} a(i,j) &= \left[erf\left(\frac{M}{2\beta}\right)\right]^2 \\ a(i\pm 1, j) = a(i, j\pm 1) &= \frac{1}{2}\left[erf\left(\frac{M}{2\beta}\right)\right]\left[erf\left(\frac{3M}{2\beta}\right) - erf\left(\frac{M}{2\beta}\right)\right] \\ a(i\pm 1, j\pm 1) &= \frac{1}{4}\left[erf\left(\frac{3M}{2\beta}\right) - erf\left(\frac{M}{2\beta}\right)\right]^2 \end{aligned} \qquad \textbf{(4-14)}$$

and R_{sm} is the *order* of the convolution matrix (***A***); here $R_{sm} = 3$. $a(l,m)$ is the fraction of the energy of the back-scattered electrons deposited at the *(i,j)* mesh site *centre point* due to a uniform incident dose applied to the *entire (l,m)* mesh site.

2. Kasuga *et al.* [123] used a uniform filter to smooth the map:

$$\lambda^{sm}(i,j) = \sum_{l=i-\frac{R_{sm}-1}{2}}^{i+\frac{R_{sm}-1}{2}} \sum_{m=j-\frac{R_{sm}-1}{2}}^{j+\frac{R_{sm}-1}{2}} a(l,m)\lambda(l,m) \qquad \textbf{(4-15)}$$

where $a(l,m) = \frac{1}{R_{sm}^{2}}$. Note that R_{sm} is constrained to be an odd number, but may not necessarily be equal to 3. R_{sm} is assumed to be 3 for this filter type in the rest of this thesis, unless stated otherwise.

The former and the latter methods are referred as MPADPEC and KPADPEC respectively in the rest of this thesis.

Repeating equation (4-15) iteratively produces a correction result that first improves with increasing iteration number and then degrades. The best iteration count occurs when the weighting distribution of the resultant cascaded uniform filter is approximately equal to the E_b distribution. This is actually the synthesis of a Gaussian filter by cascaded uniform filters [101, 102]. The variance, σ^2 of the synthesized filter is given by:

$$\sigma^2 = n\frac{\left(w^2 - 1\right)}{12}$$

where w and n are the width of the uniform filter and the convolution (filtering) iteration count respectively. Since $\beta^2 = 2(variance)$ and $w = R_{sm}$, so the optimum iteration count, n, occurs when $\sigma\sqrt{2}$ is closest to β:

$$n = f\left(\frac{6\beta^2}{M^2(R_{sm}^{2} - 1)} + 0.5\right) \quad \textbf{(4-16)}$$

where $f(x)$ is a function that returns the nearest integer to x. This technique will fail to synthesize the Gaussian filter that represents the E_b distribution if the minimum achievable $\sigma\sqrt{2}$ value is greater than β. This occurs when:

$$M > \frac{\beta}{\sqrt{\frac{1}{6}\left(R_{sm}^{2} - 1\right)}}$$

Kasuga *et al.* [123] suggested a pattern-fracturing algorithm using gradient vectors derived from the λ map. This method partitions only shapes in critical regions into

smaller shapes with acceptable dimensions. In this way, it minimises the increase in data size and improves correction accuracy.

4.4.7.1 Derivation of the optimum incident dose formula

Let E_u and E_w be the resultant deposited energy density at unwritten and written points respectively in a mesh site. Assuming the values of E_b and λ are uniform and the electron forward scattering range is negligible, then E_u is caused by back-scattered electrons from nearby shapes while E_w is caused by both forward scattered and back-scattered electrons. Specifically:

$$E_u = C\lambda\eta D_I$$

and

$$\begin{aligned} E_w &= CD_I + E_u \\ &= C(1+\lambda\eta)\,\mathrm{D_I} \end{aligned}$$

where $C = k\ /(1+\eta)$. The middle energy density level, E_m is the average of the above equations:

$$\begin{aligned} E_m &= \frac{1}{2}(E_w + E_u) \\ &= \frac{C}{2}(1+2\lambda\eta)D_I \end{aligned}$$

To make E_m uniform for all mesh sites, the incident dose distribution, D_I must satisfy the following relation:

$$D_I = \frac{2E_m}{C(1+2\lambda\eta)}$$

When $E_m = \omega\ /k$, we arrive at equation (4-12).

4.4.7.2 Performance indicators of PADPEC

The preparation time of a λ map is the sum of time taken for computing and smoothing the λ map. The former depends on the total number of shapes in a layout, whereas the latter depends on the number of mesh sites, which in turn depends on the chip size and M. With the aid of special hardware [55], it can take 9 seconds to prepare the λ map for a 400 mm^2 memory large scale integration chip consisting of 10^7 shapes and a M value of 10μm. Using the same M value and the hardware, the estimated λ map preparation time is about 14 seconds for a 100 mm^2 ASIC chip containing 4×10^7 shapes. The hardware computes λ' and the optimum incident dose for each rectangle (the correction procedure steps 4 and 5 are described in section 4.4.7) in real time. On the other hand, PADPEC software running on a minicomputer (capable of executing 177 MIPS) takes about 26 minutes, 20 minutes and 9 minutes for correcting device isolation, transistor gate and capacitor layers respectively of a 64 Mbit DRAM device [124].

Sources of correction error in PADPEC are described and analysed in Chapter 5. Using the MPADPEC method, Murai *et al.* [55] has successfully fabricated a test pattern with a CD value of 0.2 μm and a line width control of less than 0.02 μm. Figure 4-10 shows experimental results of the MPADPEC corrected line/space pattern on a negative resist, AG170-MP [125] for $M = \beta = 9$μm. The incident dose tolerance is defined as the maximum amount of fluctuation from the correct value in an incident dose such that the developed minimum size feature is within ±10% of its design values. For a minimum incident dose tolerance of 10%, the λ value must be less than 55% for the line/space pattern with CD = 0.2μm according to the results.

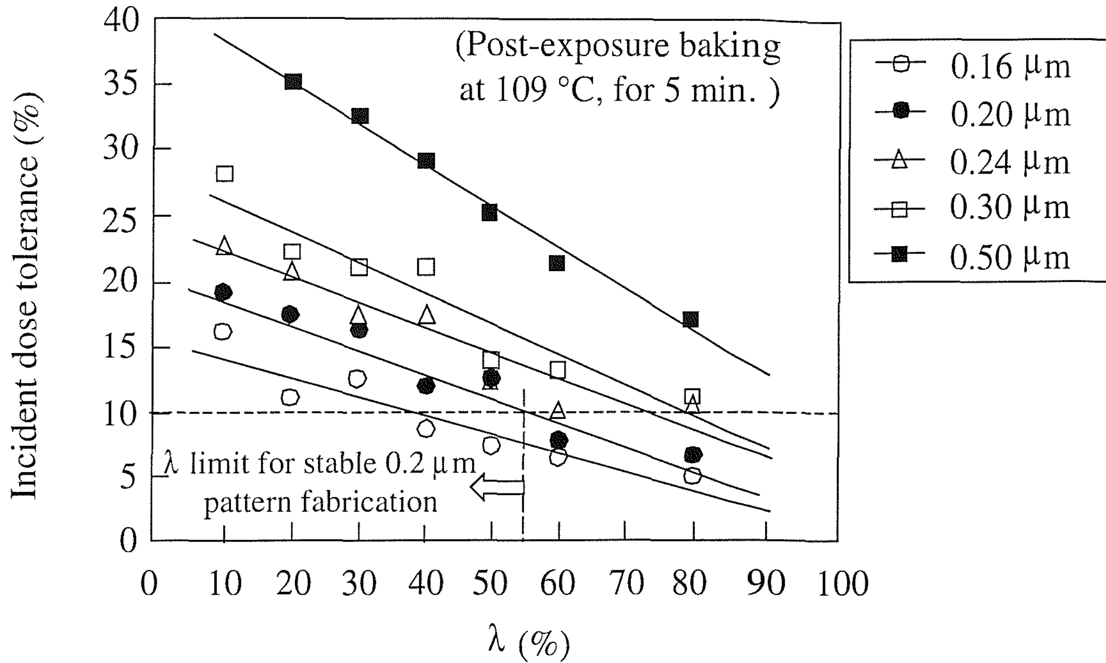


Figure 4-10 Variation of λ with incident dose tolerance for various CD values.

Kise *et al.* [126] investigated the influence of the beam energy and the beam diameter on the maximum allowable mesh size of MPADPEC for achieving a line width control of ±10% and a wall angle greater than 80° on a 0.1 μm wide infinite line. The line is 0.1 μm away from a semi-infinite shape. The pattern is written on a X-ray mask substrate composed of a 0.2 μm thick ZEP520 resist on top of a Tungsten (0.5 μm)/Silicon substrate. Changing the beam acceleration voltage from 50kV to 100kV, the maximum mesh size increases from 0.3 μm to 5 μm. When the beam diameter is reduced from 60nm to 30nm at 100kV, the maximum mesh size increases from 0.4 to 5μm.

4.4.7.3 Advantages and disadvantages of PADPEC

The evaluation time of the optimum dose in PADPEC is independent of the λ value as in the RFPEC system. This approach is computationally simple, leading to fast correction times. The scheme can also be modified easily for heterogeneous substrates. Because λ values and filter weights are always positive, so the values of the optimum doses are also always positive.

As this scheme aims to equalise E_m instead of E_w, large written shapes are over-exposed. This might decreases the throughput of an EBL machine [127]. This system does not compensate for the forward scattering effect.

4.5 Comparison of various PEC schemes

A comparison between the techniques outlined here is given in Table 4-5. The memory requirements of a PEC scheme are estimated as the sum of memory space needed for storing the incident doses and any additional space required by the scheme. In estimating the memory requirements and the total number of floating point operations, Φ needed for computing the incident doses in a sequential processing environment, the following assumptions are made for:

- GHOST
  - Equations (4-1) and (4-2) are used for calculating the GHOST parameters and these evaluations take about 7 floating point operations.
  - N is the total number of shapes.
- PYRAMID
  - The correction time depends linearly on the number of rectangles (the correction times reported in [69] support this assumption). The evaluation time of the incident doses is the dominant component of the correction time.
  - The correction time (203.2s) for 131072 rectangles on a SPARC 2 workstation [69] is used to estimate Φ. The workstation is capable of 4.2 MFLOPS.
  - N is the total number of rectangles.
  - The calculation of the shape adjustment of a rectangle involves only rectangles which are within a distance of $w = 2\beta$ from the centre point of the rectangle. Therefore, the additional space needed for storing the look-up tables is roughly equal to $\left(\frac{4\beta}{\Delta}\right)^2$.
- The self-consistent method
  - The PEC scheme uses the zoning algorithm described in section 4.4.1.1, where $w = 2\beta$ and $s = 2w$. Thus, the additional space needed for storing the adjacency matrix is $\left(\frac{2w+s}{\Delta}\right)^4 = \left(\frac{8\beta}{\Delta}\right)^4$.

- N is the total number of exels.

- CAPROX
  - CAPROX is around 20.5 times faster than the self-consistent method. This figure is obtained from Table 4-2.
  - N is the total number of shapes.
  - The memory requirements are a fraction of the space needed for the self-consistent method, depending on the value of the hierarchy factor.
- ME
  - N is the total number of exels.
  - Each iteration needs 10^4 FLOP per exel and the technique takes 100 iterations for convergence [105].
  - The memory requirements are the same as for the self-consistent method.
- PROXECCO
  - The sample spacing used in FFT is equal to the exel spacing.
  - N is the total number of exels.
  - Additional space needed for storing the FFTs of $e(r)$ and $f(r)$ is equal to $2N$. The former storage space is reused for storing the incident doses.
- The neural network approach
  - N is the total number of exels.
  - The computation cost of training is negligible compared with the convolution cost.
  - The order of the correction kernel is equal to 5β.
- RFPEC
  - Each element region contains one shape.
  - N is the total number of element regions.
  - As the evaluation of $U(r)$ is the most computationally intensive step in evaluating the incident doses, so Φ is taken to be equal to the total number of floating point operations needed for computing $U(r)$.
  - Evaluation of the incident dose of each element region involves $\left(\frac{5\beta}{2\Delta_r}\right)^2$ reference regions.
  - Four evaluations of the error function are needed for calculating the contribution of each reference region to $U(r)$.

- Each evaluation of the error function needs about 34 floating point operations[1].
- Additional space is needed for storing the representative figures. Each representative figure occupies two storage units: one for its centre of gravity and the other for its area. Thus, the total additional storage space is equal to $N\left[2\left(\frac{\Delta_e}{\Delta_r}\right)^2\right]$.

❖ PADPEC

- N is the total number of mesh sites.
- $R_{sm} = (5\beta/M)$.
- Since the evaluation of λ^{sm} is the most time consuming step in evaluating the incident doses, so Φ is taken to be equal to the total number of floating point operations needed for computing λ^{sm}.
- Additional space needed for storing the λ map is equal to N.
- Each mesh site contains one shape.

The relative speed of each PEC scheme is given by the ratio of the Φ value of the scheme to the Φ value of PADPEC.

Table 4-5 gives a rough comparison of the performance of the various PEC schemes. The values of the relative speed (relative to the speed of PADPEC.) in the table are computed for $N = 10^8$, $\beta = 10\mu m$, $\Delta = 100nm$ and $M = 2\Delta_r = \beta$.

---

[1] The average runtimes of repeat executions of the error function and a floating point multiplication, each in a separate tight loop, are used to estimate the number of floating point multiplication per execution of the error function.

PEC schemes	Experimentally achieved CD	Φ	Relative speed	Estimated memory requirements	Advantages	Disadvantages
GHOST	0.15μm ± 5% [59]	7	2.80×10^{-9}	N	• computationally trivial. • correction beam parameters, σ_c and Q_c are independent of written shapes. • applicable to all types of EBL machines.	• an additional correction dose exposure per wafer, reducing significantly the throughput of an EBL machine. • needs high contrast resists. • high correction dose could cause a serious pinhole problem in resists. • causes a scumming problem in low contrast negative resists. • reduces significantly the line width control of isolated lines. • does not compensate for the forward scattering effect.
PYRAMID	0.1μm ± 10% [69]	$6512N$	2.60×10^{2}	$N+\left(\frac{4\beta}{\Delta}\right)^2$	• fast correction due to the use of pre-computed look-up tables for all the correction calculations. • easily extended to correct patterns on heterogeneous substrates. • applicable to all type of EBL machines.	• not general enough for a wide range of circuit patterns. • shape adjustment reduces edge contrast. • produces lower process latitude than dose correction schemes. • needs a fine address spacing.

Table 4-5(a) Summary of the performance of GHOST and PYRAMID.

PEC schemes	Experimentally achieved CD	Φ	Relative speed	Estimated memory requirements	Advantages	Disadvantages
The self-consistent method	0.15 μm ± 5% [89]	$4096\left(\frac{\beta}{\Delta}\right)^4 N$	1.64×10^{10}	$N+\left(\frac{8\beta}{\Delta}\right)^4$	• compensates for the forward scattering effect.	• computationally intensive. • requires huge storage space. • possibility of ill-conditioned matrices. • negative incident doses. • needs an EBL machine capable of dose modulation.
CAPROX	0.35μm ± 1% [94]	$200\left(\frac{\beta}{\Delta}\right)^4 N$	8.00×10^{8}	$\tau\left[N+\left(\frac{8\beta}{\Delta}\right)^4\right]$ where $\tau \leq 1$	• compensates for the forward scattering effect.	• same as the disadvantages of the self-consistent method except it is much faster and needs less memory space. • needs an EBL machine capable of dose modulation.
ME	0.125μm ± 2% [98] (simulation result)	$10^6 N$	4.00×10^{4}	$N+\left(\frac{8\beta}{\Delta}\right)^4$	• the solution is constrained to non-negative values. • deals with adjacency matrix singularity problems.	• computationally intensive. • needs huge storage space. • needs an EBL machine capable of dose modulation.

Table 4-5(b) Summary of the performance of the self-consistent method, CAPROX and ME.

PEC schemes	Experimentally achieved CD	Φ	Relative speed	Estimated memory requirements	Advantages	Disadvantages
PROXECCO	1 ± 0.04 μm [110]	$3N\log_2 N$	3.19×10^0	$2N$	• availability of fast special hardware for FFT.	• negative incident doses. • non-straight-forward practical implementation. • needs an EBL machine capable of dose modulation.
Neural network approach	1 ± 0.05μm [114]	$\left(\frac{5\beta}{\Delta}\right)^2 N$	1.00×10^4	$N+\left(\frac{5\beta}{\Delta}\right)^2$	• much faster and needs less memory space than the conventional matrix inversion approach.	• needs an EBL machine capable of dose modulation. • correction accuracy depends on the training patterns and the PEC scheme used to generate the patterns.
RFPEC	0.2μm ± 10% [121]	$34\left(\frac{5\beta}{2\Delta_r}\right)^2 N$	3.40×10^1	$N\left[1+2\left(\frac{\Delta_e}{\Delta_r}\right)^2\right]$ where $\frac{\Delta_e}{\Delta_r}\leq 1$	• the evaluation time of the optimum dose is independent of the pattern area density. • fast correction as it computes incident doses on an element region basis.	• uses non-optimum pattern fracturing algorithm. • does not compensate for the forward scattering effect. • needs an EBL machine capable of dose modulation.

Table 4-5(c) Summary of the performance of PROXECCO, the neural network approach and RFPEC.

PEC schemes	Experimentally achieved CD	Φ	Relative speed	Estimated memory requirements	Advantages	Disadvantages
PADPEC	0.1 µm ± 10% [126]	$\left(\frac{5\beta}{M}\right)^2 N$	1.00×10^0	$2N$	• the evaluation time of the optimum dose is independent of the pattern area density. • fast correction due to simple computation. • easily extended to correct patterns on heterogeneous substrates.	• large written shapes are over-exposed. • does not compensate for the forward scattering effect. • needs an EBL machine capable of dose modulation.

Table 4-5(d) Summary of the performance of PADPEC.

4.6 Rationale for selecting PADPEC for improvement in this research

Among all the reviewed PEC schemes, the self-consistent method, FPPEC and the hierarchical approach produce D_I with negative values. The optimisation approach is computationally too intensive, whereas GHOST reduces significantly the throughput of an EBL machine. Although PYRAMID, RFPEC, PADPEC and the neural network approach are fast, PYRAMID reduces pattern edge contrast and requires an EBL machine to have a fine address. Furthermore, PYRAMID is theoretically less sound than the other three methods. The neural network approach is slower than RFPEC and PAPPEC. Unlike the other dose modulation PEC schemes, a larger value of β *reduces* computational time and memory requirement for PADPEC and RFPEC methods. RFPEC and PADPEC have comparable correction accuracy, but the latter is faster computationally. Therefore, the PADPEC technique has been chosen for improvement in this research.

Chapter 5

The enhanced pattern area density proximity effect correction algorithm

This chapter first describes and analyses the sources of correction error in the PADPEC scheme. Next, it gives a detailed account of the enhancements made to PADPEC before discussing the implementation of EPADPEC in software.

Unless stated otherwise, the threshold energy density model (section 3.2) with a normalized resist threshold energy density, $E_T = 1$ is used as the developer model to predict the developed resist image and a normalised double Gaussian EDF (equation (3-1)), with $\alpha = 74$ nm, $\beta = 9.6$ μm and $\eta = 0.83$, is assumed to describe exactly the deposited energy density profile at some resist depth due to a point source electron beam for the rest of the thesis. The double Gaussian EDF parameter values [123] are experimentally determined using a 0.5 μm thick chemically amplified negative electron beam resist, SAL601 and the HL800D direct write system [6] with an acceleration voltage of 50kV. The double Gaussian model is used as it has a clear physical meaning (section 3.2) and several desirable mathematical features (section 3.2.1). Note any EDF model is applicable in the EPADPEC scheme.

5.1 Sources of correction error in the PADPEC technique.

There are five sources of error in the PADPEC algorithm described in section 4.4.7:

1. interpolation;
2. assuming a constant incident dose per shape;
3. smoothing;
4. neglect of forward scattering;
5. the fundamental accuracy of the incident dose formula (equation (4-12)) itself.

The following sections contain an analysis of these error components. As all the contributions are pattern dependent, the one-dimensional pattern of Figure 5-2(a) is used as a basis for the analysis. Figure 5-1 defines the figure of merit (edge error) used to compare the contributions from sources 1, 2 and 5 [121]. (It is not applicable to sources 3 and 4, so a different comparison criterion will be introduced in sections 5.1.3 and 5.1.4).

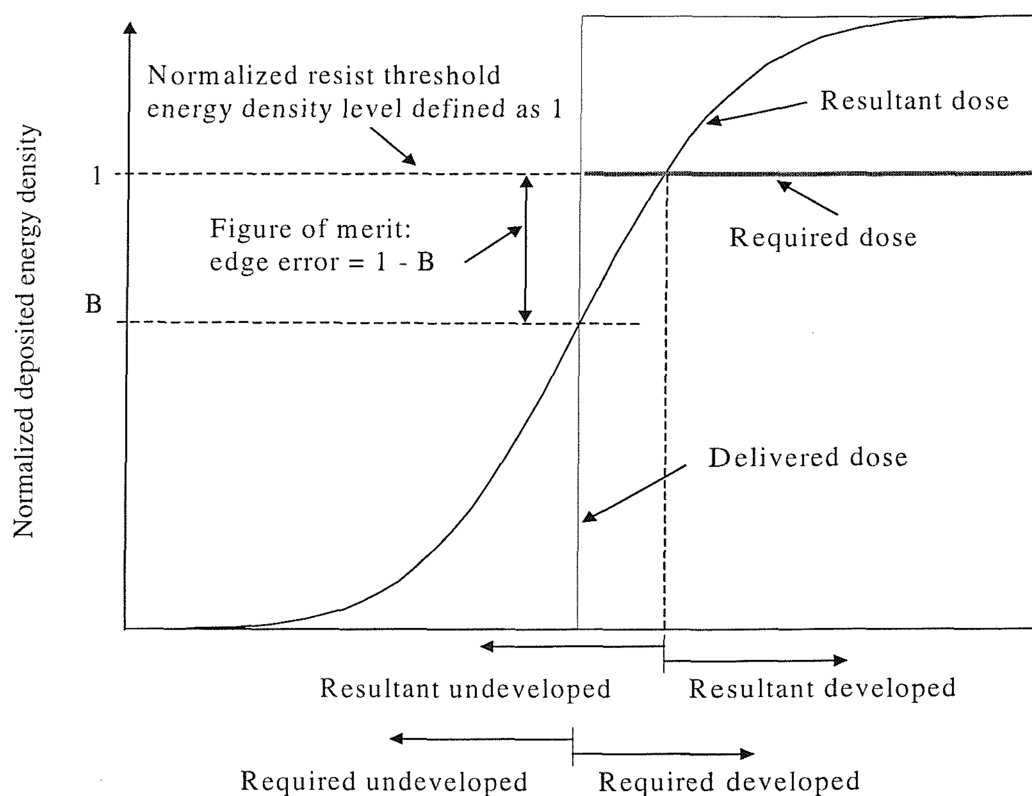
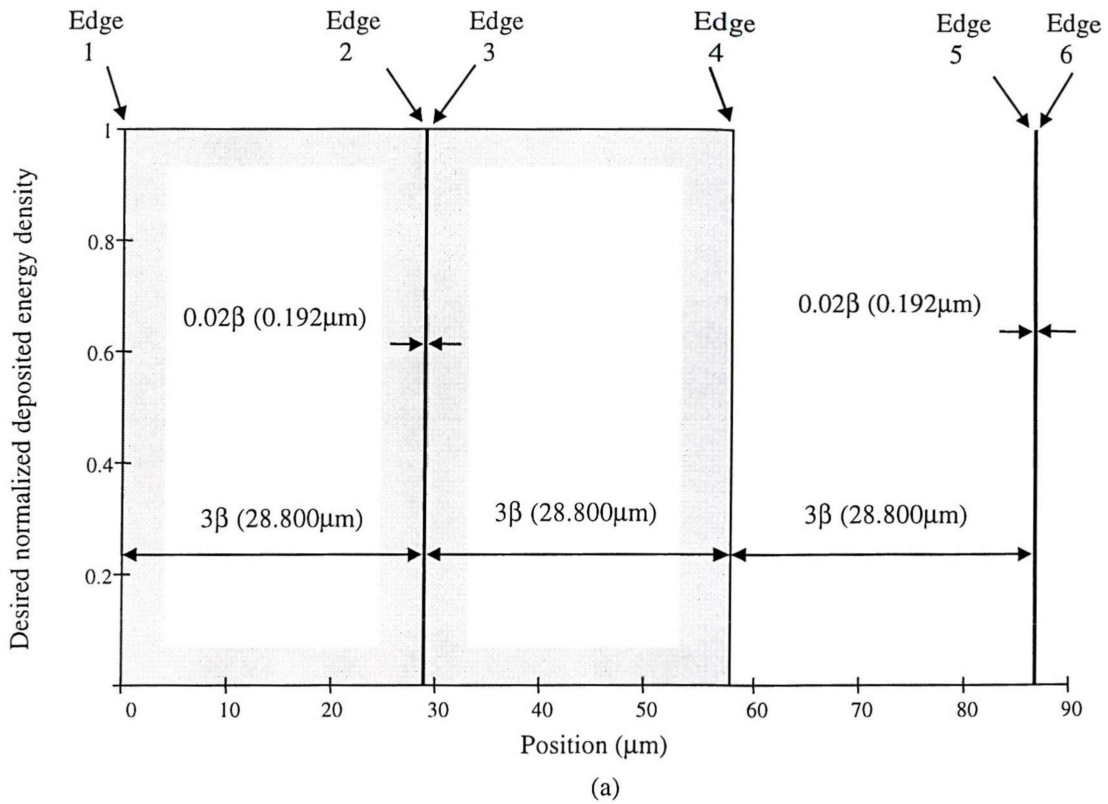


Figure 5-1 Definition of edge error.



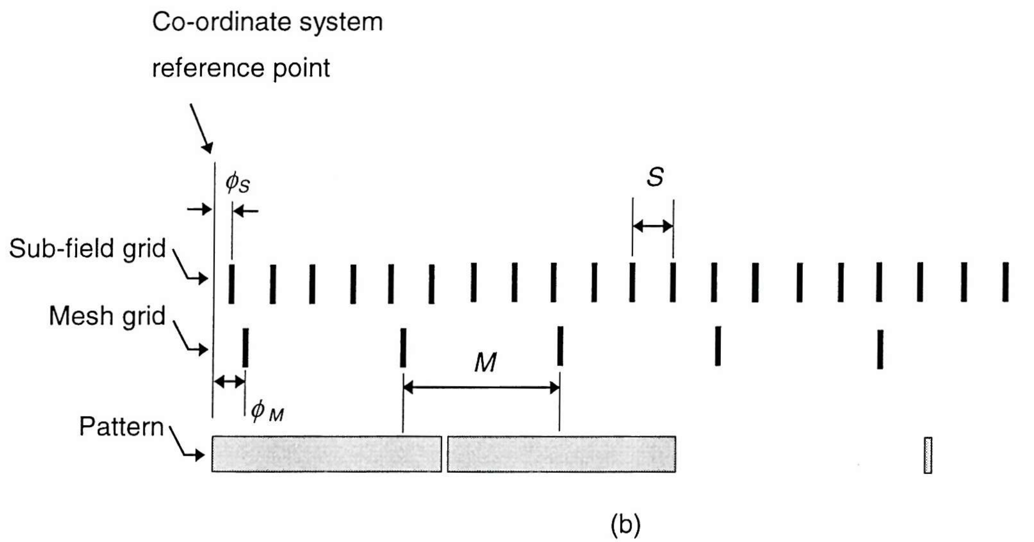


Figure 5-2 (a) Test pattern for the relative error analysis of section 5.1. (b) Geometric frames of reference of the pattern, mesh and sub-field grids.

5.1.1 Interpolation

The first two effects are considered together. In the overall data flow outlined in section 4.4.7, λ' is derived from λ^{sm} using linear interpolation between four nearest

mesh centroids and the rectangle centre. Clearly, as M decreases, the errors introduced by this step will also decrease.

5.1.2 Assuming a constant incident dose per shape

Assuming the incident dose level throughout a shape to be constant can also lead to errors[1], especially if the shape is large. This is handled by fracturing each shape into a number of *sub-shapes* before processing begins. The fracturing is performed by overlapping a grid (the *sub-field* grid) onto the pattern; the sub-field grid spacing (S) is independent of, and usually less than, M. As with the previous effect, allowing S to approach zero will minimise this source of error.

It is worth looking in some detail at the sensitivity of edge error (Figure 5-1) with respect to the geometric base of the calculations. Figure 5-2(b) shows the one-dimensional pattern of Figure 5-2(a), overlaid with the mesh grid and the sub-field grid. Without loss of generality, the origins of both these grid systems are assumed to coincide with the left-hand edge of the pattern. When M is large (comparable to the pattern feature size), the sensitivity of edge error with respect to ϕ_M (the *phase* of the mesh grid) will be high, and this will decrease as the size of M decreases. Figure 5-3 shows the variation in edge error for edge 1, with $M = 0.2\beta$ and $S = 0.02\beta$, as ϕ_M and ϕ_S move from 0 to 2π. The amplitude of this phase plot (here 1.7×10^{-3}) can be used to derive a tolerance on the value of edge error.

Figure 5-4 shows the edge errors appearing on the pattern of Figure 5-2(a) as M and S are varied. The tolerances of edge errors are shown as error bars. Note that, in varying M, the order of A in equation (4-13) was also varied such that the physical area[2] convolved with A remained constant; this allows a more sensible comparison. For all edges and values of S, the edge errors saturate as M decreases. Edges 1, 2 and 5 have the same tolerances of edge errors as edges 4, 3, and 6 respectively. The differences in edge errors of similar edge pairs are due to the geometric base of the calculations as

[1] This is the E error as described in section 4.4.6.1.

[2] The physical area is large enough to include the back-scattered energy from the entire test pattern at any mesh site centre point.

mentioned in the previous paragraph. (Note the M abscissa co-ordinate system in Figure 5-4(e) and Figure 5-4(f) is reversed to make the shape of the surface clearer.)

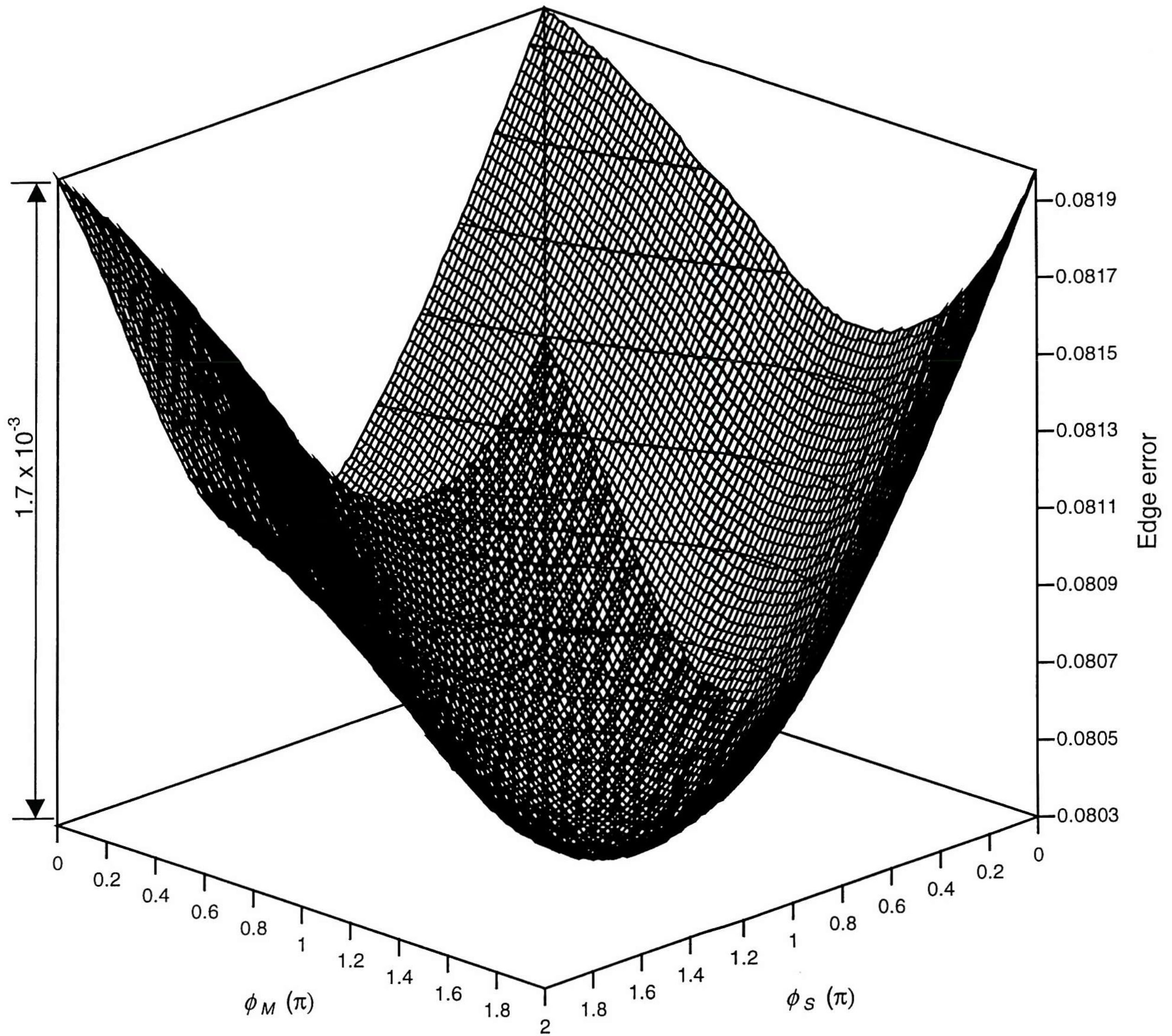


Figure 5-3 Variation in edge error for edge 1, with $M = 0.2\beta$ and $S = 0.02\beta$, as ϕ_M and ϕ_S move from 0 to 2π.

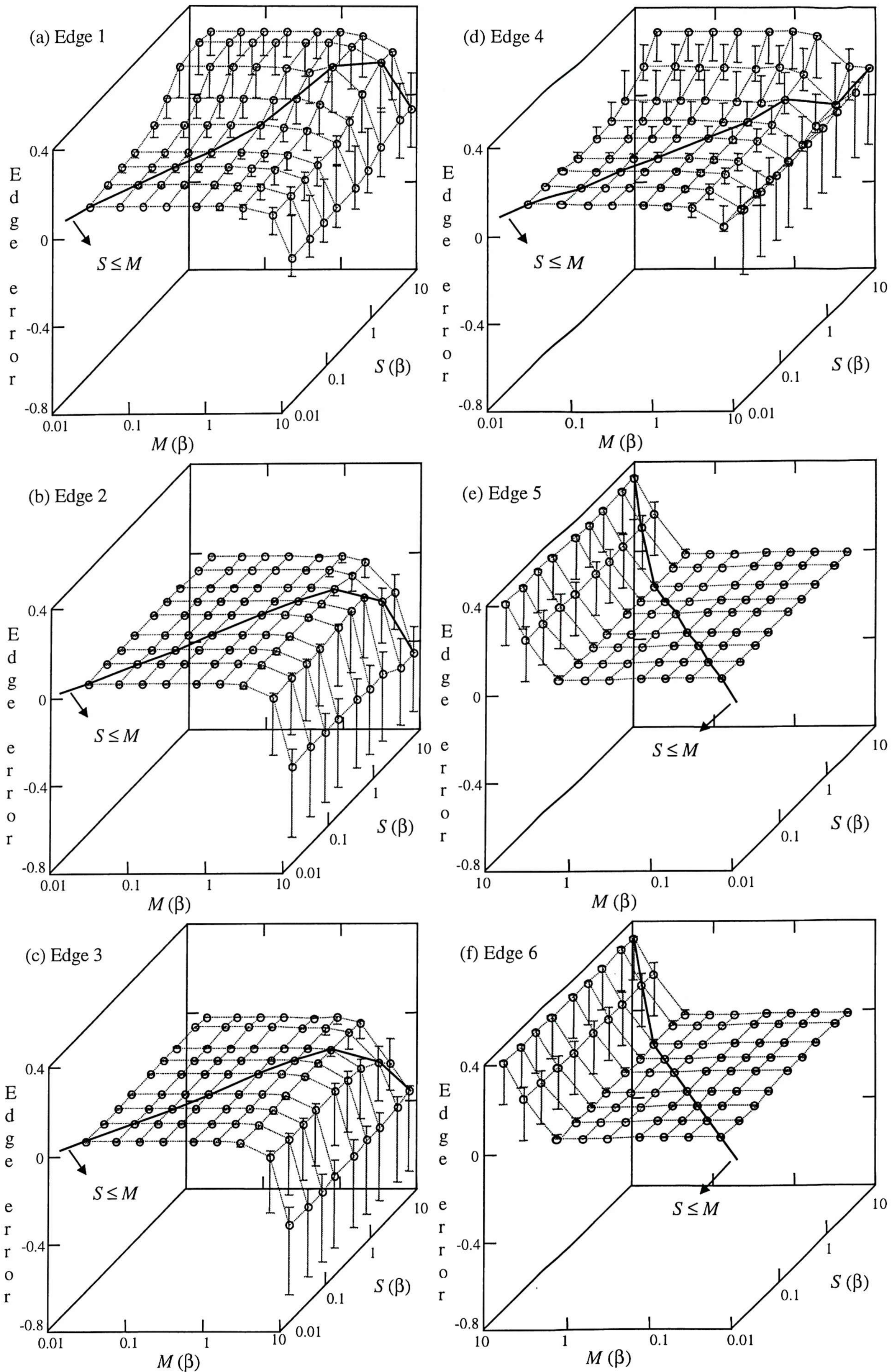


Figure 5-4 Variation of edge errors due to changing the values of *M* and *S*.

5.1.3 Smoothing

E_b at a point (x,y) due to a unit constant incident dose applied to a rectangular shape is given by

$$E_b(x,y) = \frac{k\eta}{4(1+\eta)}\left[erf\left(\frac{R-x}{\beta}\right) - erf\left(\frac{L-x}{\beta}\right)\right]\left[erf\left(\frac{B-y}{\beta}\right) - erf\left(\frac{T-y}{\beta}\right)\right] \quad \textbf{(5-1)}$$

where *R, L, T* and *B* are the right, left, top and bottom co-ordinates of the rectangle. *k* is the charge to energy conversion factor as defined in section 3.2. (See Appendix E.4 for the derivation of equation (5-1)). Smoothing λ effectively approximates *erf(r)* with a piece wise linear function, *P(r)* as illustrated in Figure 5-5. Obviously, the smaller the value of *M*, the more accurate the approximation.

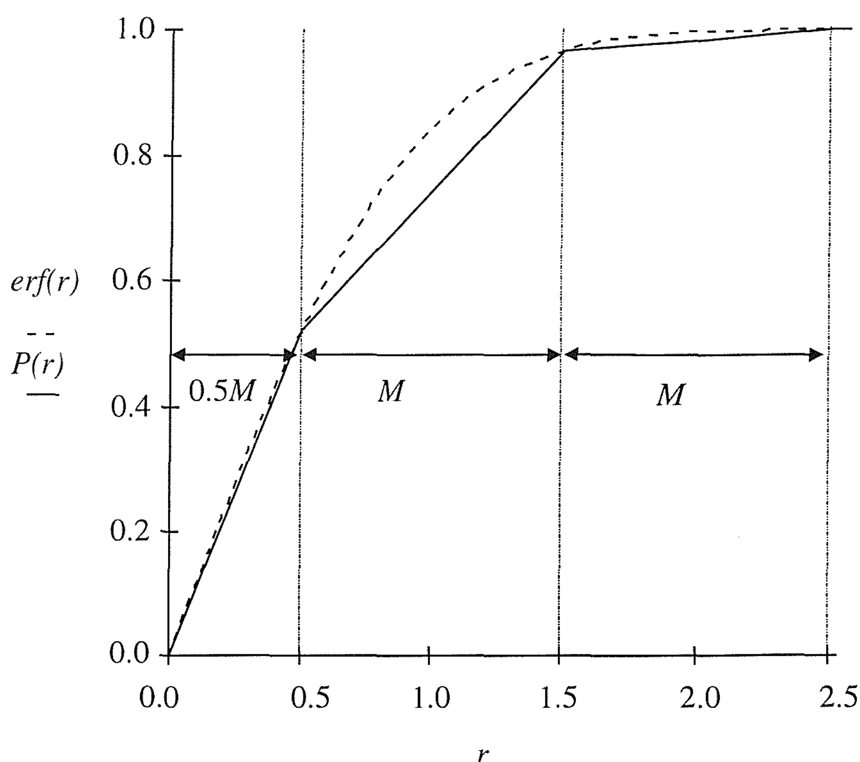


Figure 5-5 Approximation of *erf(r)* by a piece wise linear function, *P(r)*.

The errors introduced by the smoothing process, performed by the template convolution of equation (4-13), can also be affected by the order of A. Although the effects of this error source are pattern dependent, we can utilise, without loss of

generality, a figure of merit ζ, defined as the ratio of the energy absorbed in the area[1] covered by the filter to the total back-scattered beam energy:

$$\zeta = \frac{\int_{-0.5MR_{sm}}^{0.5MR_{sm}} \int_{-0.5MR_{sm}}^{0.5MR_{sm}} \frac{\eta}{\pi(1+\eta)\beta^2} \exp\left(\frac{-x^2-y^2}{\beta^2}\right) dxdy}{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \frac{\eta}{\pi(1+\eta)\beta^2} \exp\left(\frac{-x^2-y^2}{\beta^2}\right) dxdy}$$

$$= \frac{\int_{-0.5MR_{sm}}^{0.5MR_{sm}} \int_{-0.5MR_{sm}}^{0.5MR_{sm}} \frac{1}{\pi\beta^2} \exp\left(\frac{-x^2-y^2}{\beta^2}\right) dxdy}{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \frac{1}{\pi\beta^2} \exp\left(\frac{-x^2-y^2}{\beta^2}\right) dxdy}$$

The denominator is simply equal to unity as the term is normalized. Using the result of 2-dimensional integral of a normalized Gaussian function (equation (E-4) in Appendix E.3), ζ becomes:

$$\zeta = \frac{1}{4}\left[erf\left(\frac{MR_{sm}}{2\beta}\right) - erf\left(\frac{-MR_{sm}}{2\beta}\right)\right]\left[erf\left(\frac{MR_{sm}}{2\beta}\right) - erf\left(\frac{-MR_{sm}}{2\beta}\right)\right]$$

$$= erf\left(\frac{MR_{sm}}{2\beta}\right)^2$$

This function is shown in Figure 5-6. When $MR_{sm} = 4\beta$, ignoring the contribution outside the filter area in estimating E_b introduces a negligible error as ζ evaluates to 0.99067.

[1] The filter area is $(MR_{sm})^2$ - see equations (4-13) and (4-14).

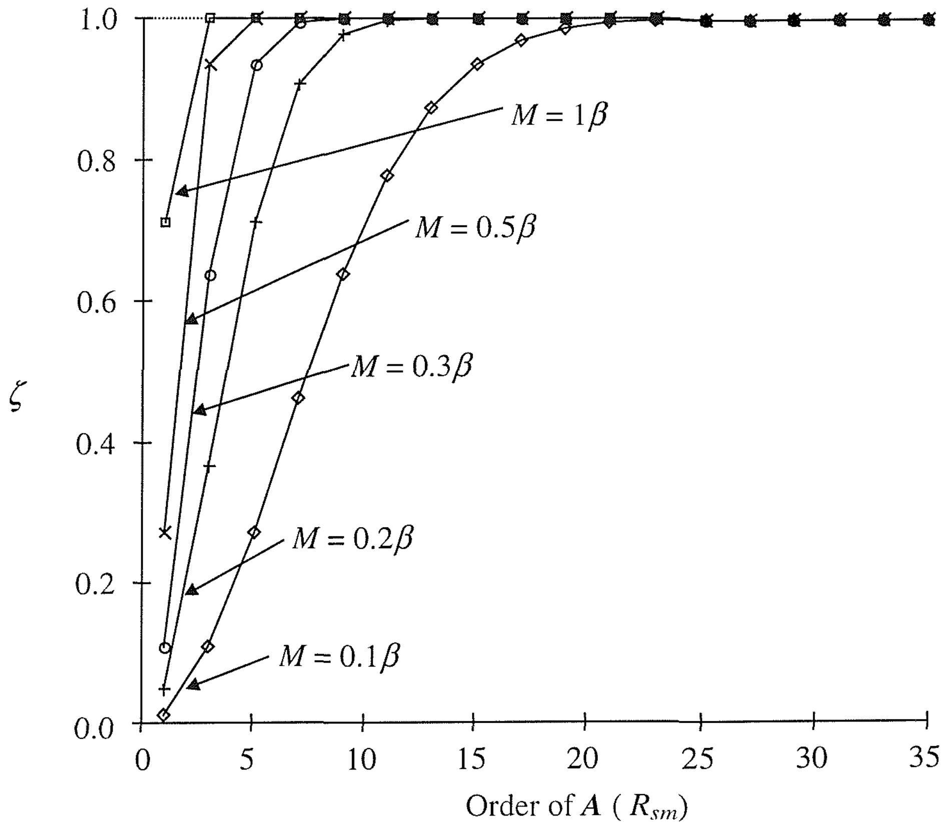
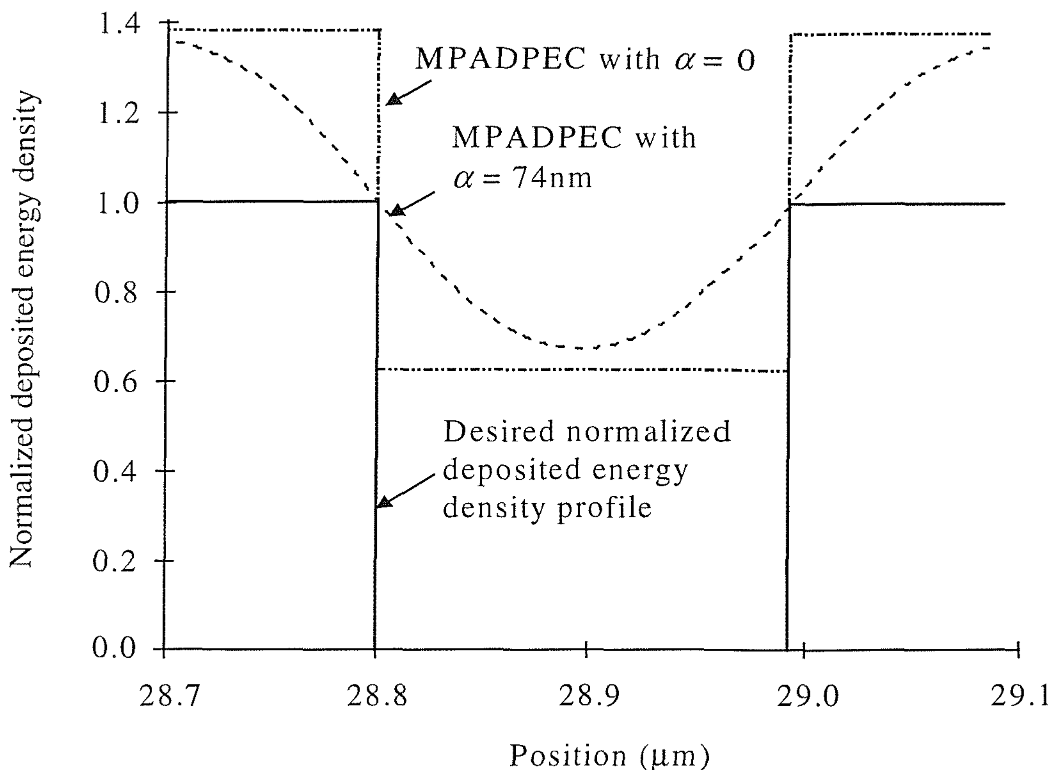


Figure 5-6 Relative errors associated with varying the order of the convolution matrix.

5.1.4 Neglect of forward scattering

This source of error arises from replacing the first term $\left(\frac{1}{\alpha^2}\exp\left(-\left[\frac{r}{\alpha}\right]^2\right)\right)$ in equation (3-1) with a delta term, $\delta(r)$. As the value of α is typically small, proximity effects are only noticeable within around 0.1μm of an edge. The figure of merit introduced earlier is not useful in this case; if we attempt to analyse the errors due to forward-scattered electrons alone, the integral of the first term in equation (3-1) at the edge can be taken as equal to 0.5 as long as the minimum feature size is several times larger than α. The effects of including the α term as a Gaussian on edges 2 and 3 in Figure 5-2(a) are shown in Figure 5-7. Although the inclusion does not affect the edge placement accuracy, it reduces edge contrast that is defined as the slope of deposited energy density profile at shape edges.



Both M and S are set to 0.01β. The order of A is large enough to include the back-scattered energy from the entire test pattern at any mesh site centre point.

Figure 5-7 Simulated resultant normalized deposited energy density distribution - detail.

5.1.5 Incident dose formula

In the derivation of the optimum incident dose formula (equation (4-12)), Murai *et. al.* have assumed a uniform λ value. However, most practical circuit patterns have non-uniform λ values. Using λ^{sm} instead of λ in the formula is an attempt to take into account the non-uniformity in λ values. λ^{sm} includes energy deposited by electrons back scattered from shapes in nearby mesh sites, therefore, they depend also on the incident doses applied to those neighbouring shapes. Murai *et al.* have assumed incident doses are the same for all mesh sites of any λ value in deriving the formula. Clearly, this assumption is no longer true after the correction. This is a form of the recursive effect mentioned in section 4.3, where the calculation result becomes a function of the internal ordering of the calculation.

To construct the *perfect* PADPEC solution, we first form a self-consistent set of linear equations using the deposited energy density, E_B, due to back scattered electrons and the total deposited energy density, E_{total}, at each sub-shape centre point:

$$E_{total}(r) = k\left[\frac{d_I(r) + \eta\sum_{l=1}^{N}\left[P(l,r)d_I(l)\right]}{1+\eta}\right]$$

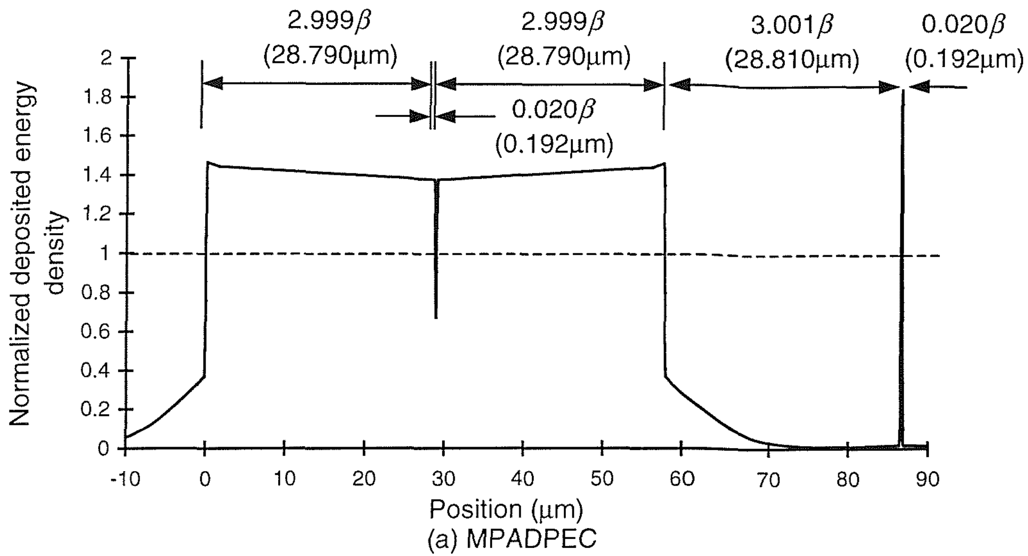
$$E_B(r) = k\left[\frac{\eta\sum_{l=1}^{N}\left[P(l,r)d_I(l)\right]}{1+\eta}\right]$$

where N and $d_I(r)$ are the total number of sub-shapes and the incident dose of rth sub-shape respectively. $P(l,r)$ is the fraction of energy of back-scattered electrons deposited at the lth sub-shape centre point due to a unitary incident dose applied to the rth sub-shape. Taking the average of the above equations forms the linear equation system:

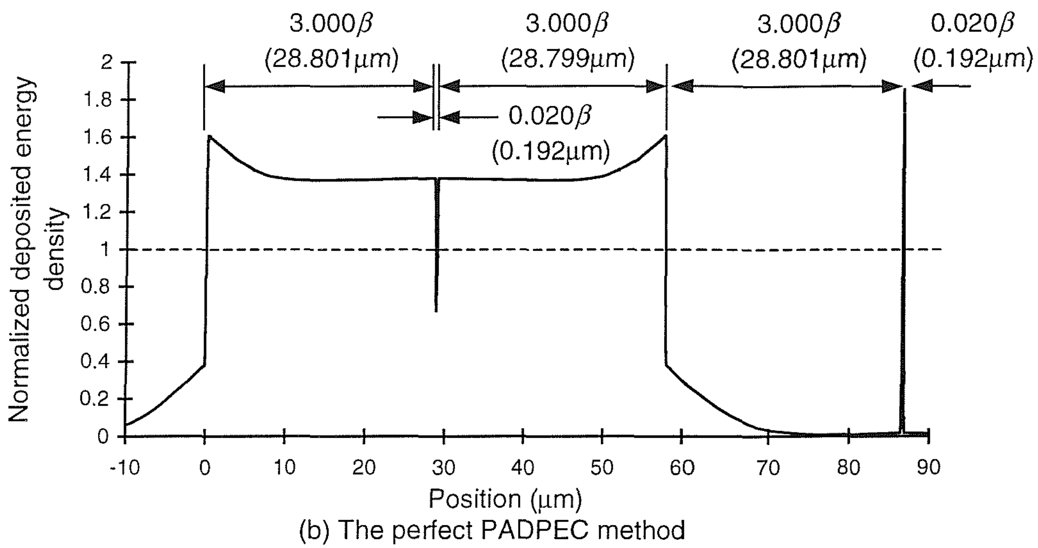
$$E_m(r) = k\left[\frac{d_I(r) + 2\eta\sum_{l=1}^{N}\left[P(l,r)d_I(l)\right]}{2(1+\eta)}\right]$$

E_m is set equal to the resist threshold energy density level. The system of linear equations can then be solved for d_I using any standard linear equation algorithm such as Gauss-Jordan elimination.

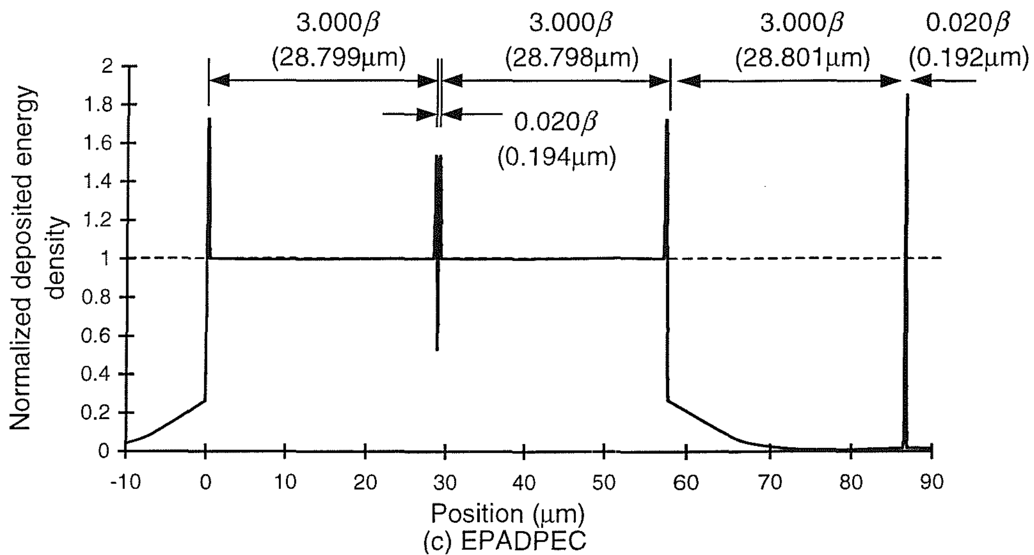
Figure 5-8(a) and (b) show the simulated resultant deposited energy density profiles for the MPADPEC corrected pattern and the perfect PADPEC scheme corrected pattern respectively (Figure 5-8(c) is discussed in section 6.2.2.1). The corresponding edge errors of the six edges in the test pattern are shown in Table 5-1. The negligibly small edge errors of the perfect PADPEC corrected pattern arise from assuming a constant incident dose per sub-shape as discussed in section 5.1.2. The large differences of edge errors at edges 1 and 4 between the two schemes implies that the



(a) MPADPEC



(b) The perfect PADPEC method



(c) EPADPEC

Both M and S are set to 0.01β. The order of A is large enough to include the back-scattered energy from the entire test pattern at any mesh site centre point.

Figure 5-8 Simulated resultant normalized deposited energy density distribution.

saturated edge errors at small values of S in Figure 5-4(a) and (d) are mainly due to the recursive effect.

PEC scheme	Edge 1	Edge 2	Edge 3	Edge 4	Edge 5	Edge 6
MPADPEC	8.10×10^{-2}	1.60×10^{-4}	1.60×10^{-4}	8.10×10^{-2}	2.48×10^{-4}	2.48×10^{-4}
the perfect PADPEC	2.20×10^{-3}	-9.18×10^{-5}	-9.18×10^{-5}	2.20×10^{-3}	2.40×10^{-4}	2.40×10^{-4}
EPADPEC	-1.42×10^{-3}	7.04×10^{-4}	7.30×10^{-4}	1.54×10^{-3}	2.40×10^{-4}	2.40×10^{-4}

Table 5-1 The corresponding edge errors of the simulated deposited energy density profiles in Figure 5-8.

5.1.6 Conclusions

The errors of Figure 5-4 indicate that errors due to interpolation (section 5.1.1) approach a horizontal asymptote when $M \leq 0.3\beta$ for all six edges. Varying values of S (section 5.1.2) also have a weak effect on the magnitude of the error, if $S \leq M$ and $M \leq 0.3\beta$.

Figure 5-6 and section 5.1.3 indicate that for $M = 0.3\beta$, the error contribution caused by varying the order of $\boldsymbol{A}$ is less than 0.1% if the order of $\boldsymbol{A}$ is greater than 13. Neglecting forward scattering introduces no serious error as long as the minimum feature size is at least three times greater than α, ensuring that the forward scattering from a shape does not deposit significant amount of energy on adjacent shapes [128].

Finally, using a "perfect" incident dose formula decreases the largest errors of Figure 5-4 by a further order of magnitude. As the underlying mathematics indicates that the asymptotic error contribution from sections 5.1.1 and 5.1.2 should approach zero as M and S approach zero, we deduce that the dominant source of error in the overall process is the formulation of equation (4-12) itself. If significant gains are to be made, this is the area that should be attacked.

5.2 Overview of differences between the EPADPEC and PADPEC schemes.

The EPADPEC technique provides two significant advantages; it corrects the recursive effect (section 5.1.5) and increases shape edge contrast, giving a more accurate edge placement.

Firstly, it includes a framing procedure that increases shape edge contrast. Each rectangle is partitioned into peripheral parts (called frames) and a central rectangle as shown in Figure 4-6 (section 4.4.4). The details of the framing procedure are discussed in section 5.2.1.2.

Secondly, it minimizes the recursive effect by using an iterative algorithm. After constructing an additional D_I map from the λ^{sm} map, it refines the D_I map using the iterative algorithm. Instead of interpolating the λ^{sm} map for each rectangle as in the conventional PADPEC schemes, EPADPEC interpolates the D_I map to assign D_I to frames and central rectangles. Sections 5.2.1.1 and 5.2.1.3 describe the iterative algorithm and the interpolation strategy in detail. Figure 5-9 highlights the differences in correction procedures between the conventional PADPEC and the EPADPEC system. The input patterns are assumed to be non-overlapped rectangles.

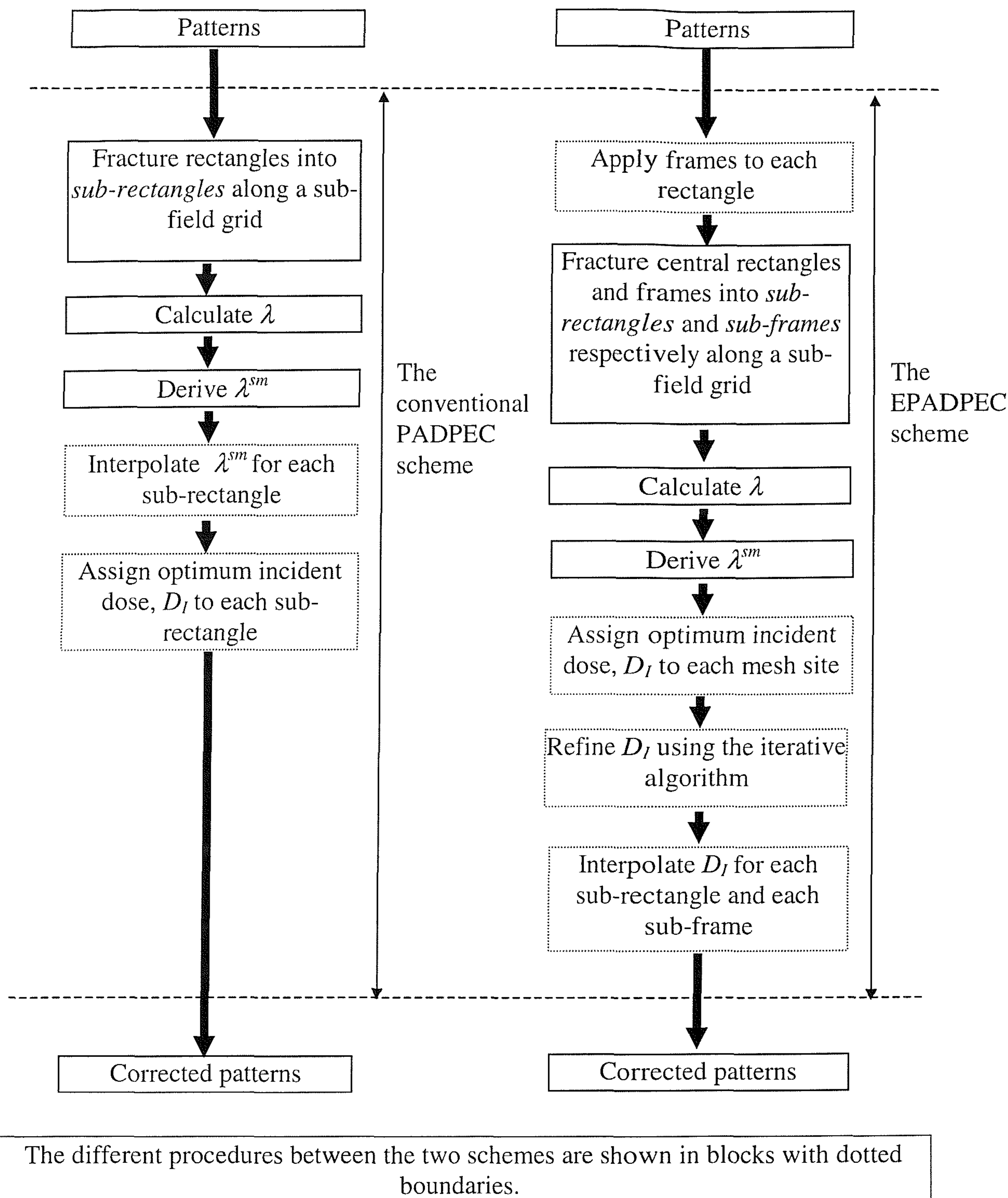


Figure 5-9 Correction procedures in the conventional PADPEC and the EPADPEC schemes.

5.2.1 Theoretical background

5.2.1.1 The iterative algorithm

Assuming the electron forward scattering is negligible and the value of E_b is uniform within a mesh site (i.e. the same assumption used in section 4.4.7.1), then the deposited energy density at nominally unwritten points, $e_u(r)$ and nominally written points, $e_w(r)$ in each mesh site, due to an incident dose distribution, $d_I(r)$ is given by

$$e_u(r) = k\left[\frac{\eta\sum_{l=1}^{N}\left[a(l,r)\lambda(l)d_I(l)\right]}{1+\eta}\right]$$

and

$$\begin{aligned} e_w(r) &= \frac{kd_I(r)}{1+\eta} + e_u(r) \\ &= k\left[\frac{d_I(r)+\eta\sum_{l=1}^{N}\left[a(l,r)\lambda(l)d_I(l)\right]}{1+\eta}\right] \end{aligned} \qquad \textbf{(5-2)}$$

respectively, where N, $d_I(r)$ and $\lambda(r)$ are the total number of mesh sites, the incident dose and the λ value of rth mesh site respectively. $a(l,r)$ is the same matrix as defined in section 4.4.7. The middle energy density level, $e_m(r)$, which is the average of the above equations is therefore:

$$e_m(r) = k\left[\frac{d_I(r)+2\eta\sum_{l=1}^{N}\left[a(l,r)\lambda(l)d_I(l)\right]}{2(1+\eta)}\right]$$

Setting all the $e_w(r)$ to E_T (the normalized resist threshold energy density), we can then apply a classical relaxation iteration method to solve the above equation for $d_I(r)$ as follows:

Jacobi method :

$$d_I^{(n)}(r) = \frac{-k\eta\sum_{\substack{l=1\\ l\neq r}}^{N} a(l,r)\lambda(l)d_I^{(n-1)}(l) + (1+\eta)e_w(\mathrm{r})}{k[1+\eta a(r,r)\lambda(r)]}$$

Gauss - Seidel method :

$$d_I^{(n)}(r) = \frac{-k\eta\left[\sum_{l=1}^{r-1} a(l,r)\lambda(l)d_I^{(n)}(l) + \sum_{l=r+1}^{N} a(l,r)\lambda(l)d_I^{(n-1)}(l)\right] + (1+\eta)e_w(\mathrm{r})}{k[1+\eta a(r,r)\lambda(r)]}$$

where

$$d_I^{(0)}(r) = \frac{(1+\eta)e_w(r)}{k\left[1+\eta\sum_{l=1}^{N} a(l,r)\lambda(l)\right]} \qquad \text{for both methods.}$$

(5-3)

The superscripts in the equations denote the number of iterations. In the expression for the initial $d_I^{(0)}(r)$, the summation term is equivalent to λ^{sm} derived by using the template convolution. Although the expression can assign a maximum incident dose of $(1+\eta)e_w/k$ to mesh sites with a λ value of zero, these doses have no effect in computing the optimum incident doses for other mesh sites. This is because contributions from other mesh sites to $d_I(r)$ of rth mesh site are weighted by their λ values. For $\eta > 1$ and a sufficiently small value of M, the underlying matrix of equation (5-2) can become non-diagonally dominant. Under these conditions, the Jacobi method might fail to converge.

If we aim to equalise $e_m(r)$ as in the conventional PADPEC methods, then the above iterative algorithms become:

Jacobi method :

$$d_I^{(n)}(r) = \frac{-k\left[\eta \sum_{\substack{l=1 \\ l \neq r}}^{N} a(l,r)\lambda(l) d_I^{(n-1)}(l)\right] + (1+\eta)e_w(\mathrm{r})}{k\left[0.5 + \eta a(r,r)\lambda(r)\right]}$$

Gauss - Seidel method :

$$d_I^{(n)}(r) = \frac{-k\eta\left[\sum_{l=1}^{r-1} a(l,r)\lambda(l) d_I^{(n)}(l) + \sum_{l=r+1}^{N} a(l,r)\lambda(l) d_I^{(n-1)}(l)\right] + (1+\eta)e_w(\mathrm{r})}{k\left[0.5 + \eta a(r,r)\lambda(r)\right]}$$

where

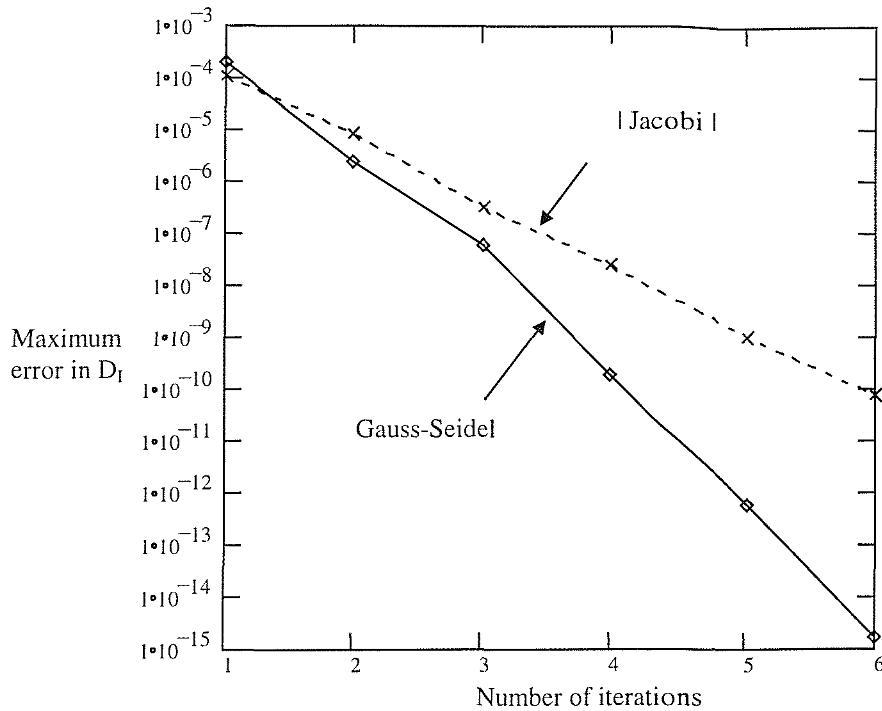
$$d_I^{(0)}(r) = \frac{2(1+\eta)e_w(r)}{k\left[1 + 2\eta \sum_{l=1}^{N} a(l,r)\lambda(l)\right]} \qquad \text{for both methods.}$$

In practice, when computing $d_I(r)$, the iterative algorithms use only mesh sites whose centre points are within a distance of 2β from the rth mesh site centre currently under evaluation.

Figure 5-10 shows variations of the maximum error in D_I from its correct solution with number of iterations for the test pattern shown in Figure 5-2(a). The error is defined as

$$\varepsilon(r) = \frac{d^n{}_I(r) - D(r)}{D(r)}$$

where $D(r)$ is the correct value of rth mesh site. From the graphs, it is obvious that the Gauss-Seidel iterative method has the best convergence rate, as one would expect in a sequential processing environment. Thus, the EPADPEC system uses the Gauss-Seidel iterative method in the rest of this thesis, unless stated otherwise.



M and frame width are set to 2β and 4α respectively. The correct solution is taken as the converged solution obtained using those iterative algorithms.

Figure 5-10 Variations of the maximum error in D_I with number of iterations.

5.2.1.2 Framing

Figure 5-11 shows the normalised convolution, $P(x)$ between a normalised Gaussian distribution with a characteristic length of σ and an isolated semi-infinite shape that occupies the right half plane. The value of $P(x)$ at the shape edge is exactly half its maximum value and the slope at the shape edge is maximally steep. Using equation (E-5) in Appendix E.4, $P(x)$ is given by:

$$
\begin{aligned}
P(x) &= \frac{1}{2}\left[erf\left(\frac{\infty - x}{\sigma}\right) - erf\left(\frac{0 - x}{\sigma}\right)\right] \\
&= \frac{1}{2}\left[1 + erf\left(\frac{x}{\sigma}\right)\right]
\end{aligned}
$$

where x is the distance from the shape edge. The value of $P(x)$ can be taken as 1 when x is greater than 4σ. If D_0 is the nominal incident dose required to produce $P(x \geq 4\sigma) = 1 = E_T$, then an incident dose of $2D_0$ will give a normalized deposited energy density at the shape edge exactly equal to E_T.

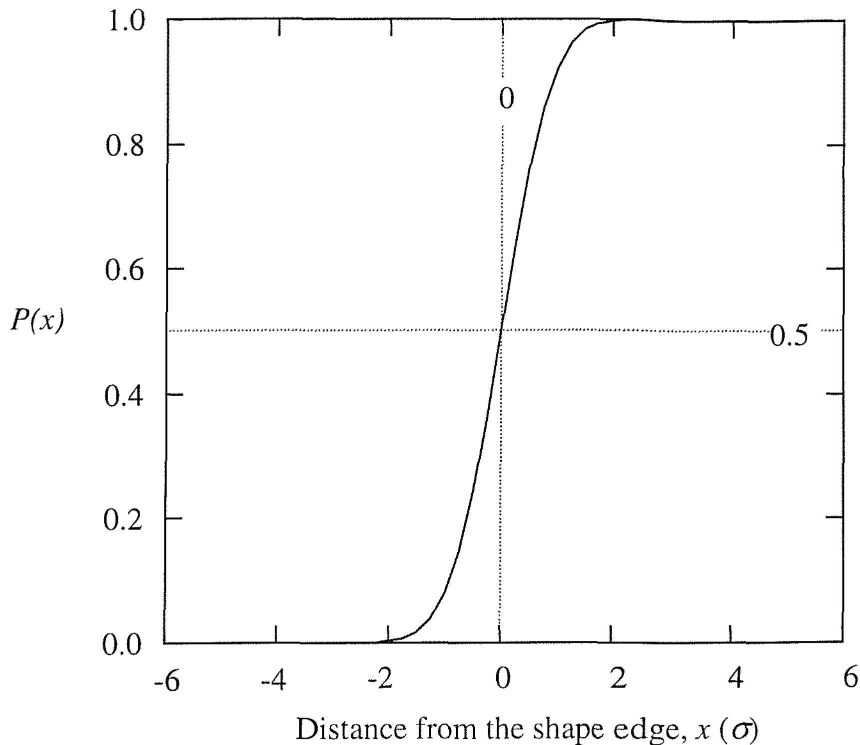


Figure 5-11 Normalised convolution, *P(x)* between a normalised Gaussian distribution and an isolated semi-infinite shape.

EPADPEC uses the above property to simplify the evaluation of optimum incident doses for frames. For a maximally steep deposited energy density profile at pattern edges, we need infinitely broad frames [110] with respect to forward scattering characteristic length, α and the optimum incident dose of these frames is equal to twice the interpolated value. In practice, a minimum frame width of 4α is sufficient to realise an infinitely broad frame. Hence, EPADPEC uses a frame width of 4α as its default value. The higher incident dose of frames gives a higher proximity effect. EPADPEC takes into account the additional proximity effect by increasing the λ value of the frame by a factor of two when computing the λ map.

5.2.1.3 Interpolation and quantization strategies

If we employ the bilinear interpolation of the D_I values of the four nearest mesh sites to the sub-frame centres or the sub-rectangle centres (as in the conventional PADPEC methods) in EPADPEC, the resultant normalized deposited energy density in some regions of a large shape might fall below E_T. This results in the formation of *interior holes* in the developed resist image (the *interior hole* problem). Figure 5-12

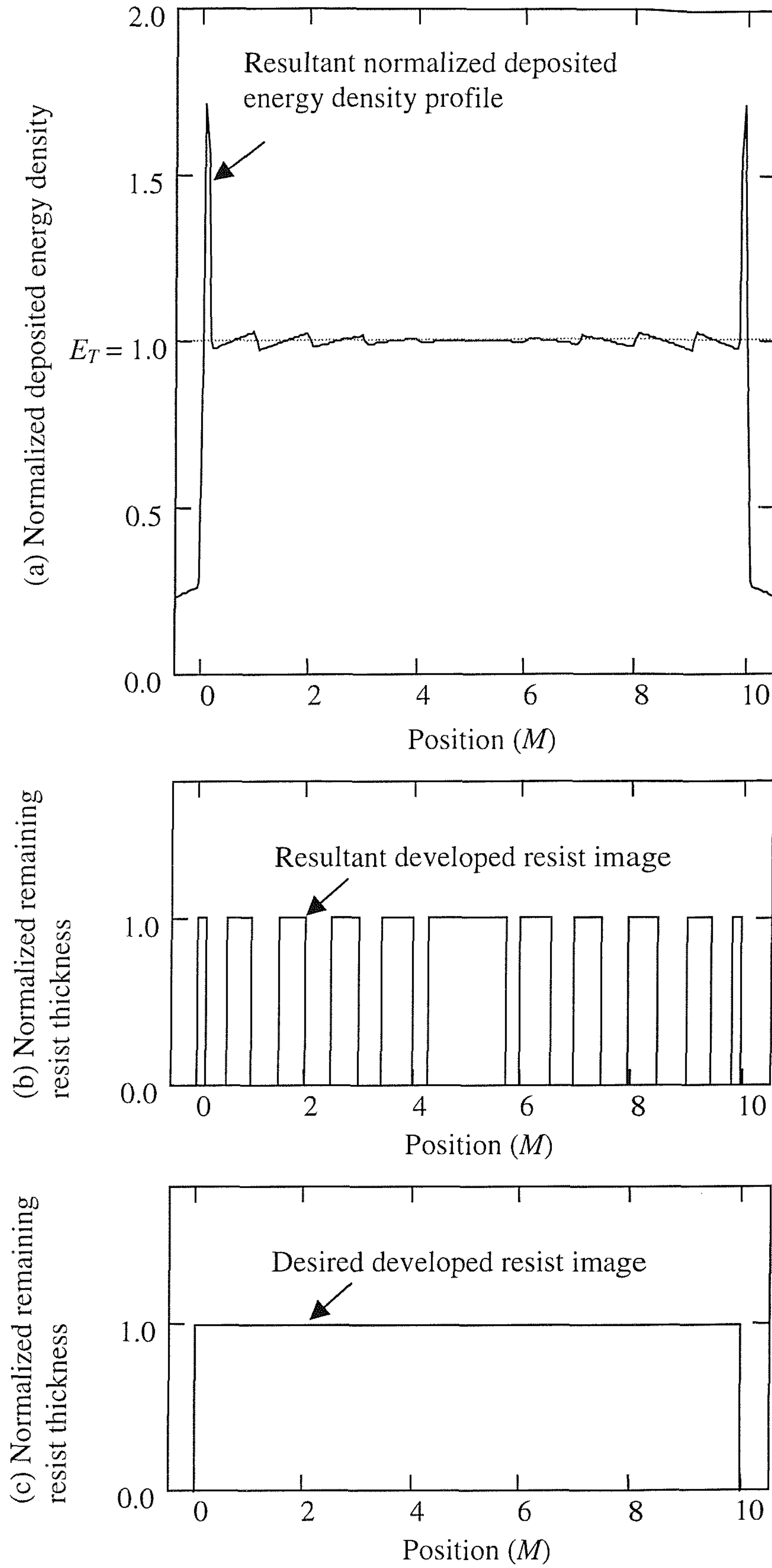
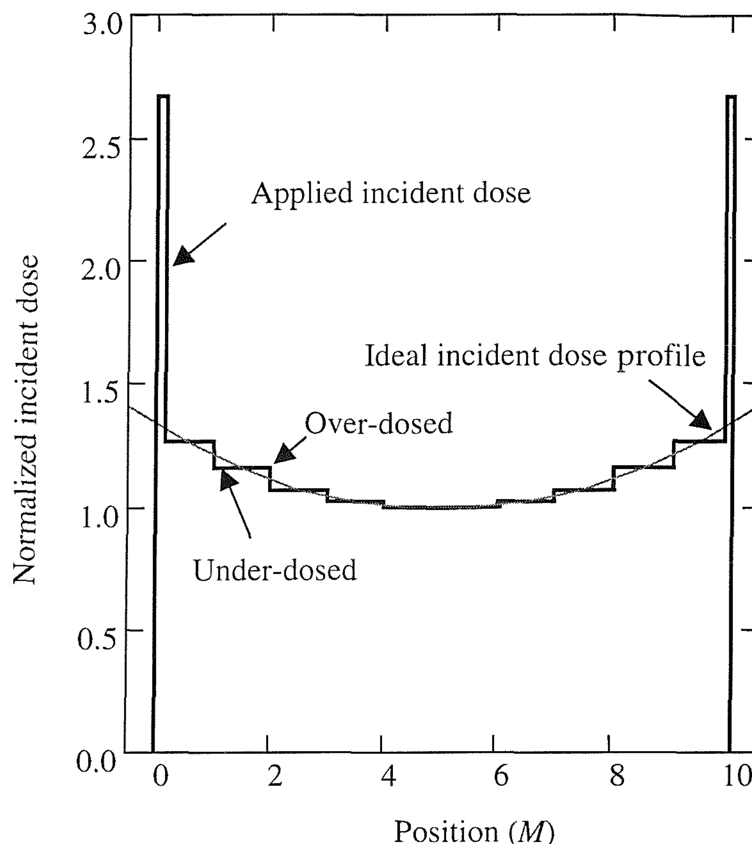


Figure 5-12 The interior hole problem.

demonstrates this problem using an isolated large one dimensional pattern in which S (sub-field size) and M (mesh size) are set to 2.56 μm. The problem arises from approximating the ideal (continuous) incident doses with a constant incident dose per sub-rectangle or sub-frame (Figure 5-13), i.e. the E error as mentioned in sections 4.4.6.1 and 5.1.2.



The doses are generated using the interpolation strategy of the conventional PADPEC methods

Figure 5-13 The applied and ideal incident dose distribution of the isolated large one-dimensional pattern.

A simple strategy to overcome this problem is to compute the D_I values at the four corners and the centre of the sub-rectangle using the bilinear interpolation, and assign the highest value to the sub-rectangle (The *five point* interpolation strategy). Since EPADPEC assigns twice the interpolated D_I values to frames, there is no risk that the normalized deposited energy density within frames will fall below E_T. Thus, it is unnecessary to change the interpolation strategy in sub-frames since the five point interpolation strategy is less accurate than the conventional interpolation strategy. Figure 5-14 shows that this interpolation strategy successfully eliminates the interior hole problem on the isolated pattern.

(a) Normalized deposited energy density

2.0
1.5
$E_T = 1.0$
0.5
0.0

Resultant normalized deposited energy density profile

0 2 4 6 8 10

Position (M)

(b) Normalized remaining resist thickness

1.0
0.0

Resultant developed resist image

0 2 4 6 8 10

Position (M)

(c) Normalized remaining resist thickness

1.0
0.0

Desired developed resist image

0 2 4 6 8 10

Position (M)

Figure 5-14 Elimination of the interior hole problem.

As all EBL machines only support discrete D_I values, there is a need to quantize the continuous (*analogue*) D_I values generated by EPADPEC. As in the interpolation case, some quantization strategies are susceptible to the interior hole problem. To avoid the problem, EPADPEC uses the uniform *upper-level* quantization strategy in which the quantized D_I values, D_q are always greater or equal to their mapped D_I values as shown in Figure 5-15.

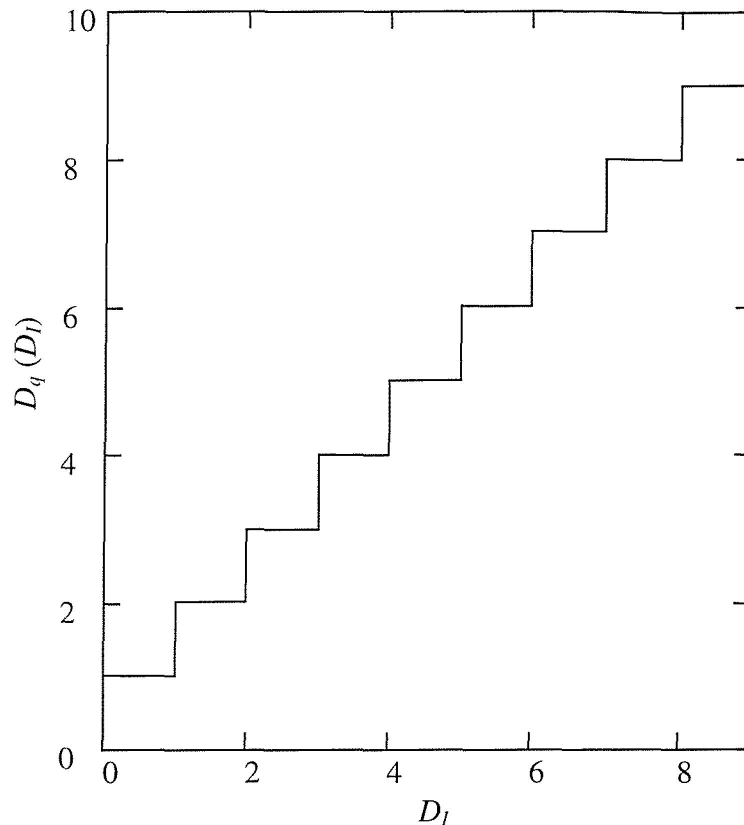


Figure 5-15 The uniform upper-level quantization strategy.

5.2.2 Algorithm implementation

This section addresses issues involved in the software implementation of EPADPEC. Figure 5-16 shows the data flow of the current software version of the EPADPEC system. P21 and P8 are explained in the next section. The processors are the additional processing for a Gaussian spot, step and write EBL machine. The current system deals only with orthogonal shapes specified in the GDSII format. P1 fractures the input shapes into rectangles and removes overlapped parts of the rectangles. Next, P22 builds frames from the rectangles and partitions frames and the rectangles into sub-frames and sub-rectangles respectively along a fixed size sub-field grid.

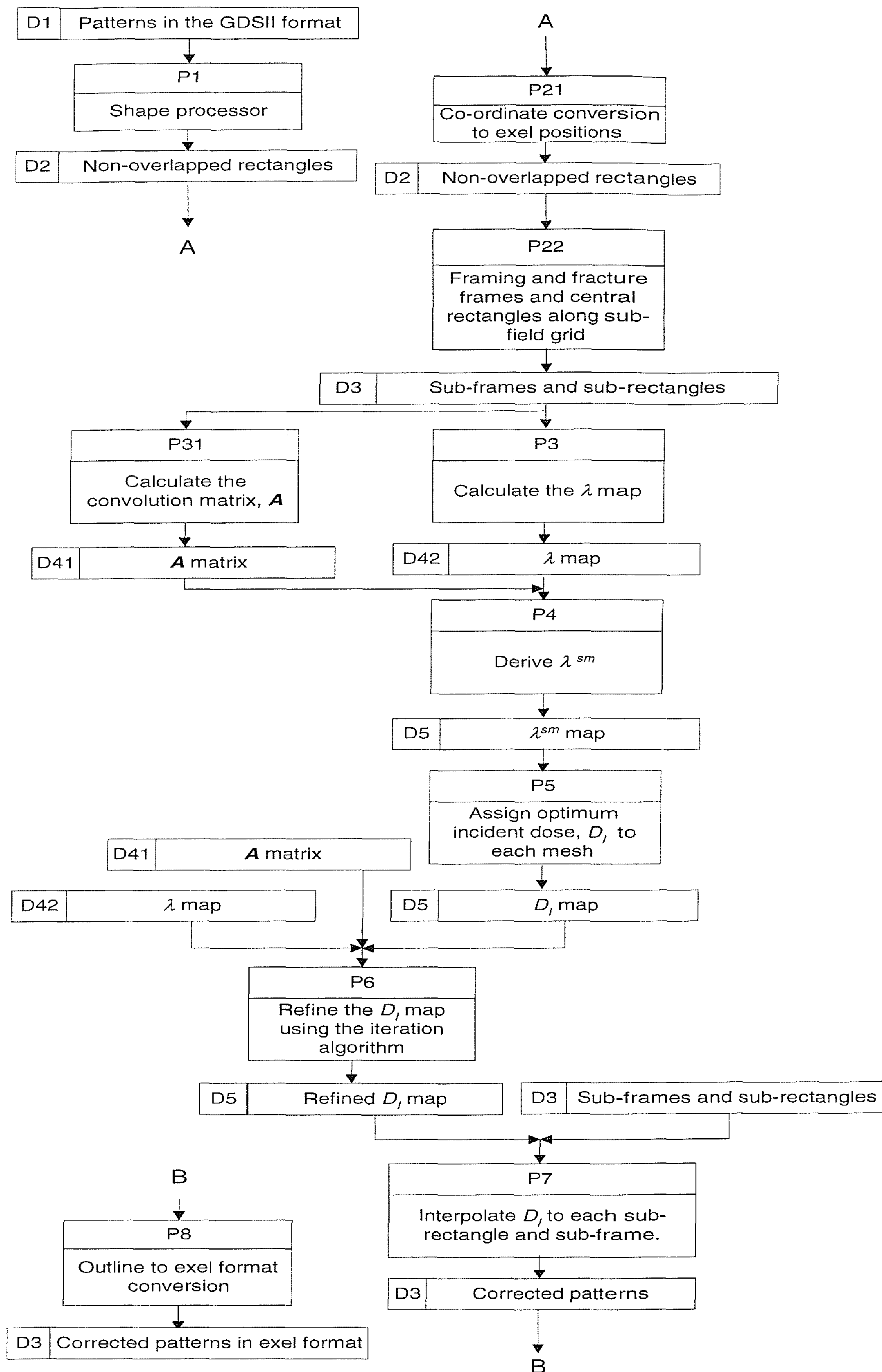


Figure 5-16 EPADPEC data flow.

In practice, this grid size might correspond to the sub-field size, the maximum beam size and the cell size of a Gaussian spot machine with a step-and-write stage movement system, a VSB machine and a cell projection machine respectively. The reference point of the mesh and sub-field grids is set to the top-left (the minimum y and x values) point of the pattern shadow. For rectangles whose dimension is smaller than twice the frame width (4α), the entire rectangles are treated as frames.

D2 is a two-dimensional array. Each element of the array contains a pointer to a data structure (CPattern) list; CPattern stores the co-ordinates of a rectangle and its incident dose. The list holds all the sub-frames and sub-rectangles within a sub-field grid site. P22 sets the D_I values of sub-frames and sub-rectangles to 2 and 1 respectively, so that P7 can use this to differentiate a sub-rectangle from a sub-frame. P7 updates the D_I value of a sub-rectangle or a sub-frame by multiplying it with the interpolated D_I value. P7 also performs the quantization on the incident doses if applicable.

5.2.2.1 Modifications to the software version of EPADPEC for a Gaussian spot, step and write EBL machine

For a Gaussian spot, step and write EBL machine, it is necessary to express the co-ordinates of a shape in terms of exel positions (*exel format*) and the evaluation of the convolution matrix, A in P31 involves discrete convolution instead of continuous convolution as mention in section 3.2. Because shape co-ordinates might not coincide with the exel positions, P21 maps them to the nearest exel positions when converting them to the exel format.

The binary format native to the EBL machine usually defines shapes in the exel format. In the exel format, the co-ordinate plane is considered as a digitised grid and the co-ordinates of the start and the end of a shape are given as the centres of these digitised points that correspond to exel positions. The shape edges are assumed to be located at the middle between adjacent exel positions as shown in Figure 5-17. Shapes defined in a CAD format such as the GDSII format are usually in *outline format* in which the co-ordinates of a shape refer to the shape edges. Thus, in converting a shape from the outline to exel format, we will need to reduce the shape size by one exel in both the x and y dimension. P8 (Figure 5-16) performs the outline to exel format conversion. It reduces the bottom-right (the maximum y and x values) point of

a shape by one exel and maintains the top-left point of the shape. The co-ordinate conversion (P21) and the format conversion (P8) are not done together, so that we do not need to change the implementation of P22, P3, P42, P5 and P7.

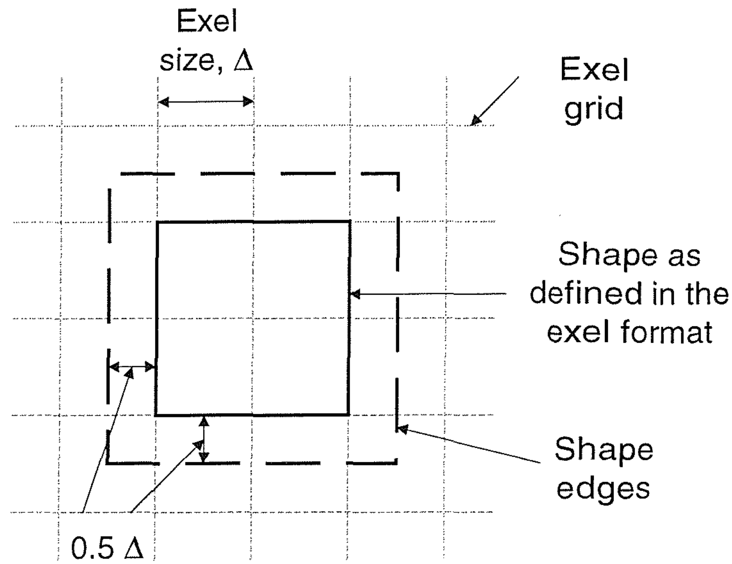


Figure 5-17 Shape definition in the exel format.

Because the discrete convolution is not normalized, we will need to normalize it by replacing the forward scattering contribution of 1 (the first term in the square brackets in the denominator of equation (5-3)) with the sum of the forward scattering discrete convolution over all space at each exel position, α_{sum}:

$$\alpha_{sum} = \sum_{x=-\infty}^{\infty} \sum_{y=-\infty}^{\infty} \frac{1}{\pi\alpha^2} e^{-\left(\frac{x^2+y^2}{\alpha^2}\right)}$$

and replacing $(1+\eta)$ with the sum of the forward and backward scattering discrete convolution over all space at each exel position, $\alpha_{sum}+\eta\beta_{sum}$:

$$\beta_{sum} = \sum_{x=-\infty}^{\infty} \sum_{y=-\infty}^{\infty} \frac{1}{\pi\beta^2} e^{-\left(\frac{x^2+y^2}{\beta^2}\right)}$$

in the formulae of the iterative algorithm (equation (5-3)):

Jacobi method :

$$d_I^{(n)}(r) = \frac{-k\eta \sum_{\substack{l=1 \\ l \neq r}}^{N} a(l,r)\lambda(l) d_I^{(n-1)}(l) + (\alpha_{sum} + \eta\beta_{sum}) e_w(\mathrm{r})}{k\left[\alpha_{sum} + \eta a(r,r)\lambda(r)\right]}$$

Gauss - Seidel method :

$$d_I^{(n)}(r) = \frac{-k\eta\left[\sum_{l=1}^{r-1} a(l,r)\lambda(l) d_I^{(n)}(l) + \sum_{l=r+1}^{N} a(l,r)\lambda(l) d_I^{(n-1)}(l)\right] + (\alpha_{sum} + \eta\beta_{sum}) e_w(\mathrm{r})}{k\left[\alpha_{sum} + \eta a(r,r)\lambda(r)\right]}$$

where

$$d_I^{(0)}(r) = \frac{(\alpha_{sum} + \eta\beta_{sum}) e_w(r)}{k\left[\alpha_{sum} + \eta \sum_{l=1}^{N} a(l,r)\lambda(l)\right]}$$ for both methods.

This version of EPADPEC is referred as the *discrete* version, while the previous version (section 5.2.2), which uses continuous convolutions is referred as the *continuous* version in the rest of this thesis.

Chapter 6

Algorithm evaluation

This chapter compares the EPADPEC algorithm with KPADPEC, MPADPEC, CAPROX and the Leica Cambridge proximity effect correction software (LPEC) in terms of correction accuracy and speed. Hughes *et al.* [129, 130, 131] describes the correction procedures of LPEC. The correction method is based on the self-consistent scheme. CAPROX supports Manhattan and physical shape fracturing algorithms, which will be described in details in section 7.2.1.

Unless stated otherwise, all PEC schemes use the normalised double Gaussian EDF (equation (3-1)), with $\alpha = 74$ nm, $\beta = 9.6$ μm and $\eta = 0.83$, to describe exactly the deposited energy density profile at some resist depth due to a point source electron beam in this chapter.

6.1 Test pattern selection and PEC settings

For the correction accuracy comparison among KPADPEC, MPADPEC and EPADPEC, the same test pattern, KasugaTP (Figure 6-1) as reported in Kasuga *et al.* [123] is used. The dummy shapes can be of various λ values. KasugaTP represents the worst case scenario for PEC as the monitor shape passes through a region (between the dummy shapes) with maximum proximity effect (when λ = *100%* for the dummy shapes) to regions with minimum proximity effect (outside the dummy shapes or when λ = *0%*). The simulation result of KPADPEC corrected KasugaTP will be compared with the result reported in [123] which used no quantization on incident doses, a sub-field grid size of 1 μm and continuous convolution (equation (3-2)) in computing the normalized deposited energy density. Therefore, when correcting KasugaTP, EPADPEC uses the continuous version with no quantization and the same sub-field size of 1 μm.

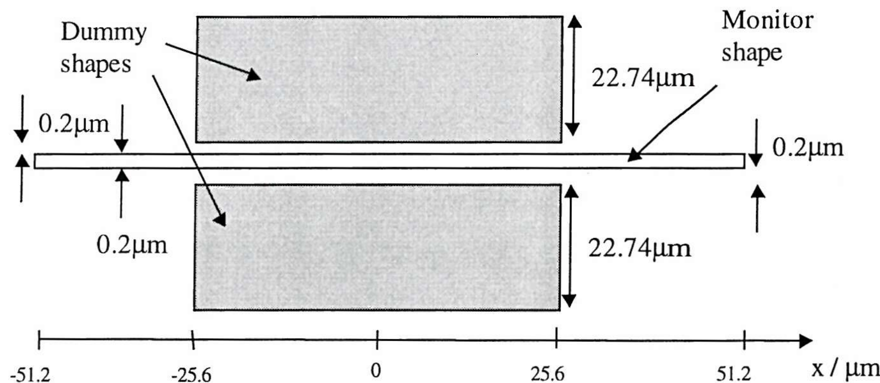


Figure 6-1 KasugaTP for comparing various PADPEC systems.

After partitioning along the sub-field grid, the huge number of fractured dummy shapes that are within the electron scattering range causes LPEC to fail [131]. Thus, instead of KasugaTP, a modified version of the test pattern, RauTP (Figure 6-2) as reported in Rau *et al.* [100] is used to compare the correction accuracy among LPEC, CAPROX and EPADPEC. The dark regions are the written regions. RauTP is good for assessing the effectiveness of a PEC scheme in correcting fine features.

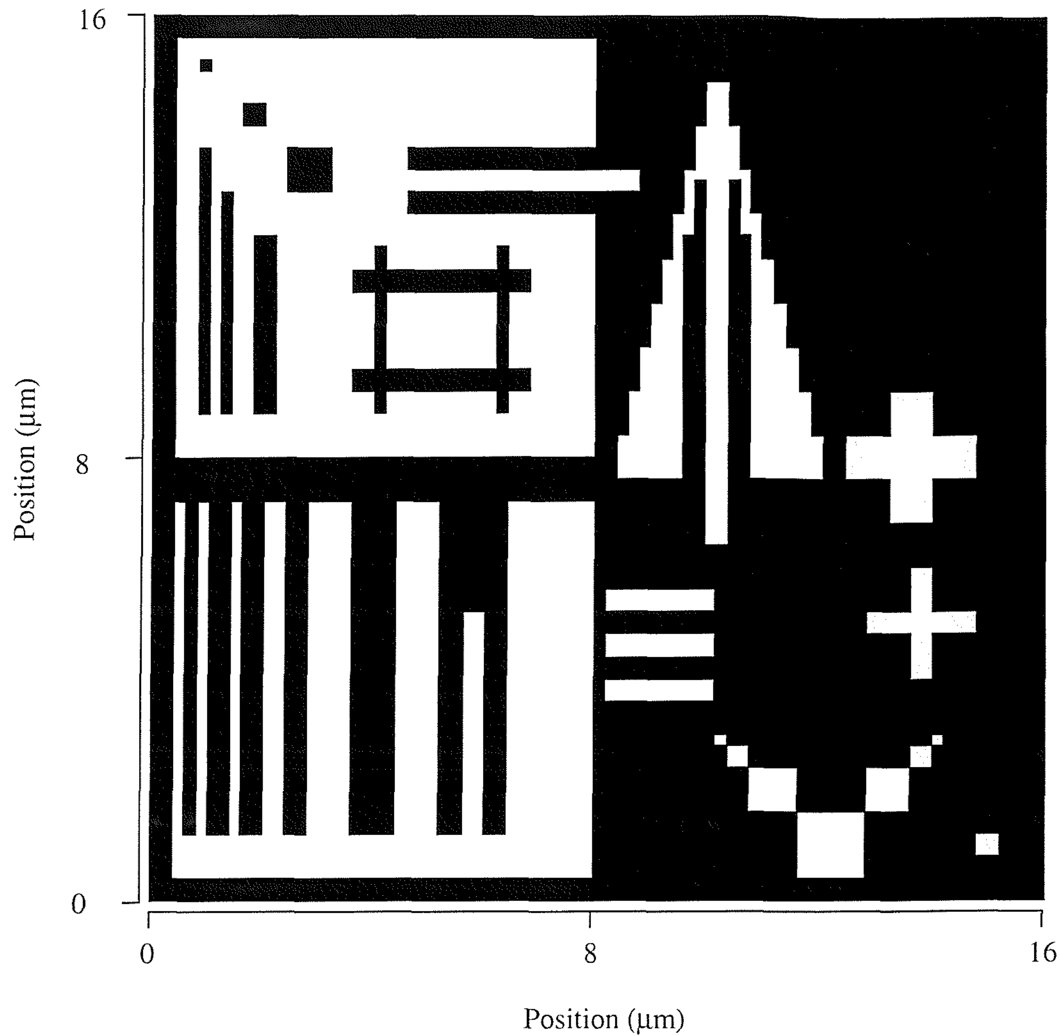


Figure 6-2 RauTP for comparing LPEC, CAPROX and EPADPEC.

As real circuit patterns are several order of magnitudes larger than KasugaTP and RauTP in size, so a correction speed comparison using these test patterns does not truly reflect the relative speed of various PEC schemes on realistic data. Thus, a complementary-metal-oxide-silicon 16-bit microprocessor (MP16, excluding pad drivers) is used for the speed comparison. Table 6-1 lists the layout characteristics of various layers in MP16 and the circuit components formed using the mask of each layer (Figure 6-3).

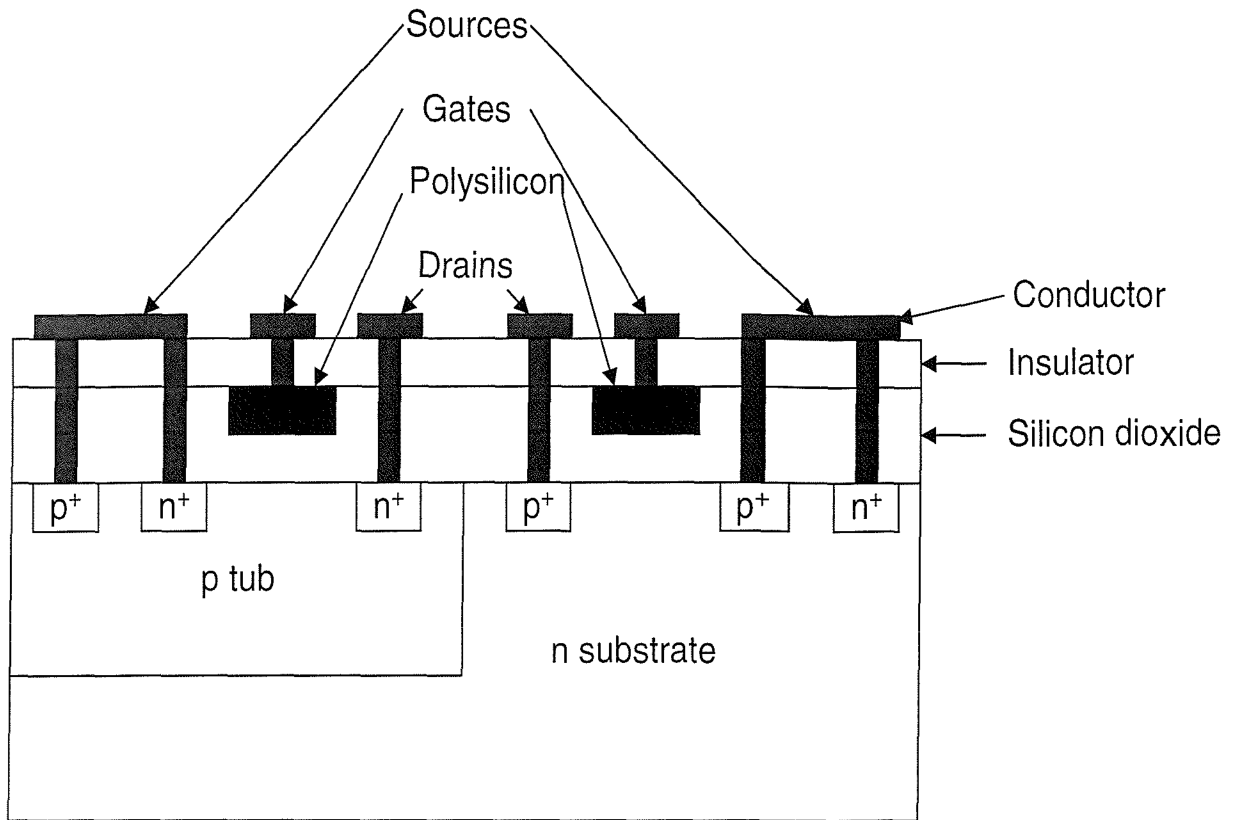


Figure 6-3 Cross section of a basic building block of the complementary-metal-oxide-silicon integrated circuit [162].

Layer number	Circuit component formed using the layer mask	Shape count	Minimum feature size (μm)	Layout shadow width (μm)	Layout shadow Height (μm)
1	p tub	3222	14.0	1396.8	1176.2
2	Silicon dioxide covering the drain and source	6958	2.0	1398.8	1175.8
11	Polysilicon gate	19016	0.8	1405.2	1207.0
12	p^+	3758	2.0	1400.8	1174.0
14	n^+	3728	2.0	1394.0	1174.0
16	Contact window for drain, source and gate	27124	1.0	1403.4	1206.0
17	Conductor	50844	1.2	1494.4	1262.6
18	Interconnection between conductor layers	6655	1.0	1491.8	1261.6
19	Conductor	25541	1.2	1494.0	1264.4

Table 6-1 Characteristics of different layers of MP16.

The department of Electronics and Computer Science, University of Southampton has a Leica EBMF 10.5 EBL machine that supports only 16 distinct values of incident doses in a pattern and has a fixed sub-field size of 1024 exels. Since we are interested in the correction accuracy and speed of the PEC schemes on patterns fabricated using the machine, so all the PEC schemes use a quantization level count of 16 and a default machine sub-field size of 1024 exels when correcting RauTP and MP16. For a fairer comparison, all the PEC systems[1] use the uniform upper-level quantization strategy (section 5.2.1.3). Because LPEC is designed only for Leica's Gaussian beam, step and write EBL machines (such as the EBMF 10.5 EBL machine), so EPADPEC uses the discrete version (section 5.2.2.1) when correcting RauTP and MP16.

LPEC has an input parameter called correction range that specifies the electron scattering range. Shapes outside the range from the evaluation point are ignored. When correcting RauTP and MP16, LPEC uses a correction range value of 20 μm ($\cong 2\beta$) that is sufficient to keep the error under 1% (section 4.4.1.1). For its other specific correction parameters, LPEC uses the default values [131] when correcting RauTP and MP16. Similarly, CAPROX also uses the default values [132] for its specific correction parameters, when correcting RauTP and MP16.

6.2 Correction accuracy

For cases without any PEC, the normalized incident dose is set to $(1+\eta)$.

6.2.1 Figures of merit

For KasugaTP, the feature of interest is the monitor shape. Thus, the line width deviation of the developed monitor shape is a suitable figure of merit for assessing the correction accuracy. The line width deviation is monitored at a regular interval of 1.28 μm along the x-axis, starting at –50.52 μm position.

---

[1] The default quantization strategy in CAPROX equalises the values of the relative deviation (dose range/(quantized dose value, D_q)). On the hand, LPEC uses either the uniform *mid-level* or *lower-level* quantization strategy. The former strategy quantizes the incident doses to the nearest D_q values, while the latter strategy quantizes the incident doses to the nearest lower D_q values. These strategies can be converted easily to the upper-level quantization strategy.

On the other hand, the entire pattern of RauTP is of interest. Here, it is more sensible to use the absolute area difference between the simulated and the desired developed resist images as the figure of merit instead of the line width deviation. As part of the simulated developed resist image might be larger while the other part of the image might be smaller than the desired developed resist image, so the absolute area difference is a better figure of merit than the signed area difference.

6.2.2 Comparison

6.2.2.1 Simulation results

Figure 5-8(c) shows the resultant deposited energy profile of the test pattern in Figure 5-2(a) when it is corrected by EPADPEC. The edge errors are shown in Table 5-1. From the results in the table, EPADPEC has clearly made a significant improvement to the worst edge error of the previous algorithm. As the conventional PADPEC schemes aim to equalise $e_m(r)$ for all mesh sites, so large written shapes are over-exposed as depicted in Figure 5-8(a). This over-dosing is significant, because it may reduce the throughput of an EBL machine. In contrast, EPADPEC aims to unify $e_w(r)$ for all mesh sites. Thus, EPADPEC eliminates the over-dosing problem on large shapes as shown in Figure 5-8(c). Figure 6-4 shows that the EPADPEC corrected pattern has a steeper deposited energy profile at shape edges as compared with MPADPEC corrected pattern. The former also gives a lower deposited energy at the isolated space. Both M and S are set to 0.01β. The order of A is large enough to include the back-scattered energy from the entire test pattern at any mesh site centre point.

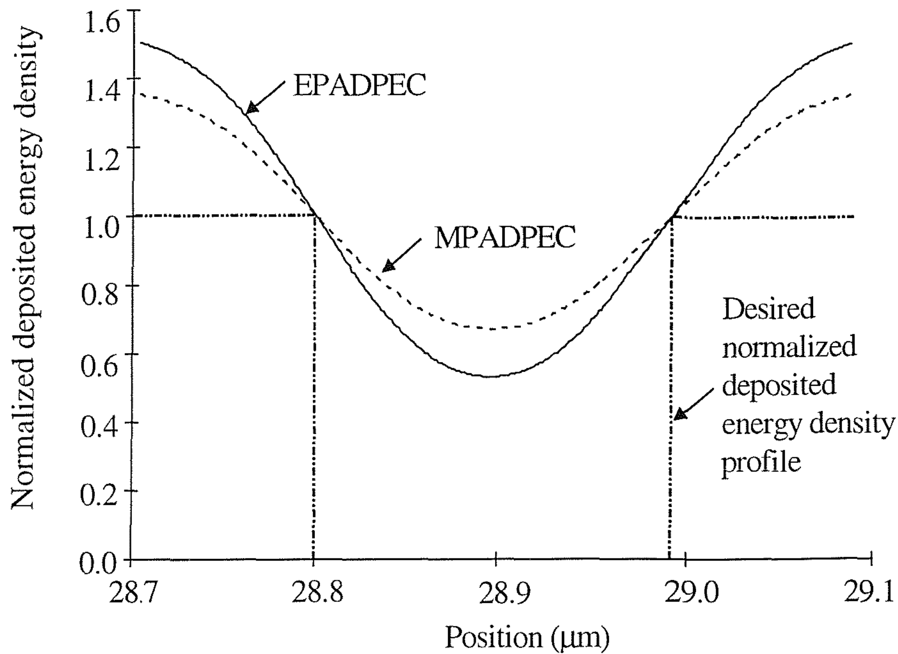


Figure 6-4 Simulated resultant normalized deposited energy distributions at edges 2 and 3 of the test pattern.

Figure 6-5 shows the line width variation of the monitor shape (KasugaTP) without any PEC. The λ value of the dummy shapes is 75%. Figure 6-6 shows the monitor line width variation for KasugaTP corrected by MPADPEC, KPADPEC and EPADPEC with $M = 5.12\mu\text{m}$ and dummy shapes with a λ value of 75%. MPADPEC and EPADPEC use A with an order of 9. KPADPEC uses the optimum iteration count of 2. The EPADPEC curve is produced with a single iteration; increasing the iteration count to 2 produces a slight improvement in the monitor line width inside the dummy shapes (Figure 6-7) but none elsewhere; increasing the iteration count still further produces negligible improvement. In practice, an atomic force microscope will be used to verify these simulation results instead of a scanning electron microscope. This is because the required measurement resolution (<2nm) is beyond the limit of the scanning electron microscope.

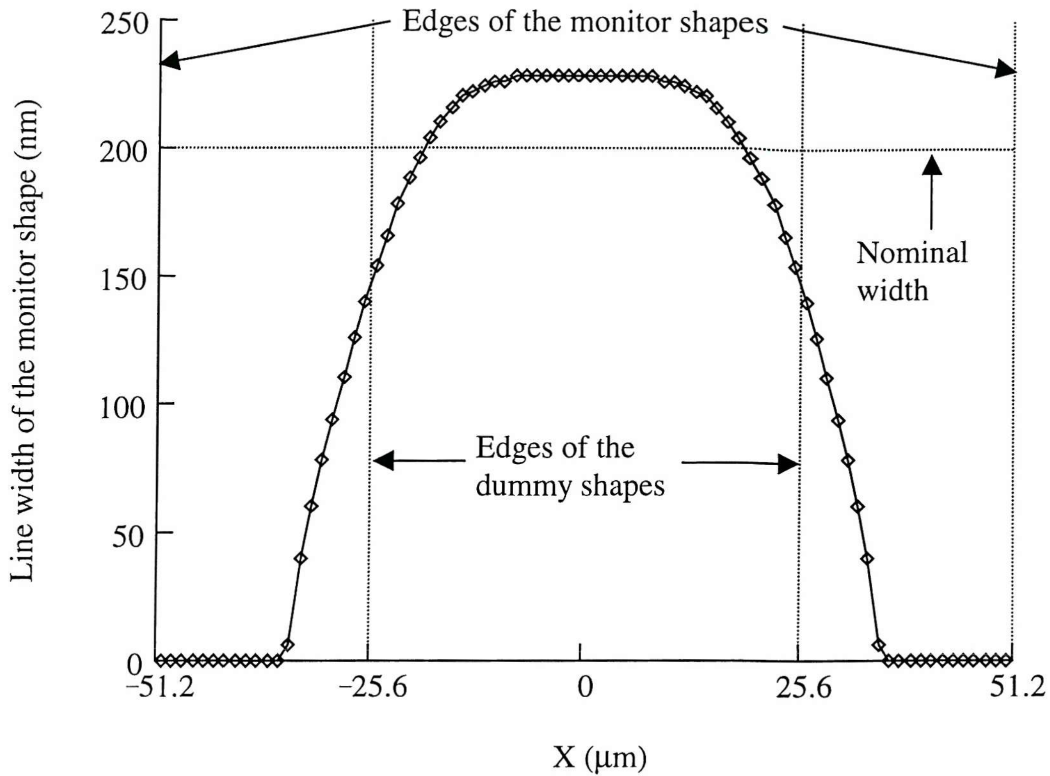


Figure 6-5 Line width profile of the monitor shape without any PEC.

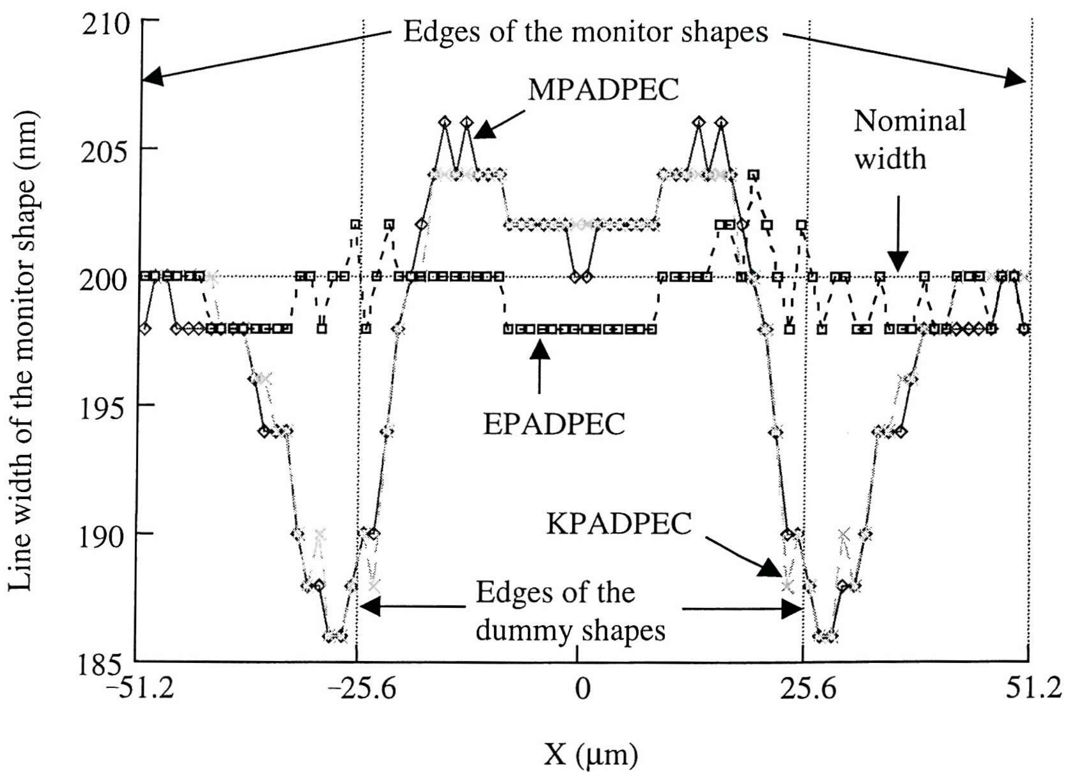


Figure 6-6 Line width variations of the monitor shape for KasugaTP corrected by MPADPEC, KPADPEC and EPADPEC.

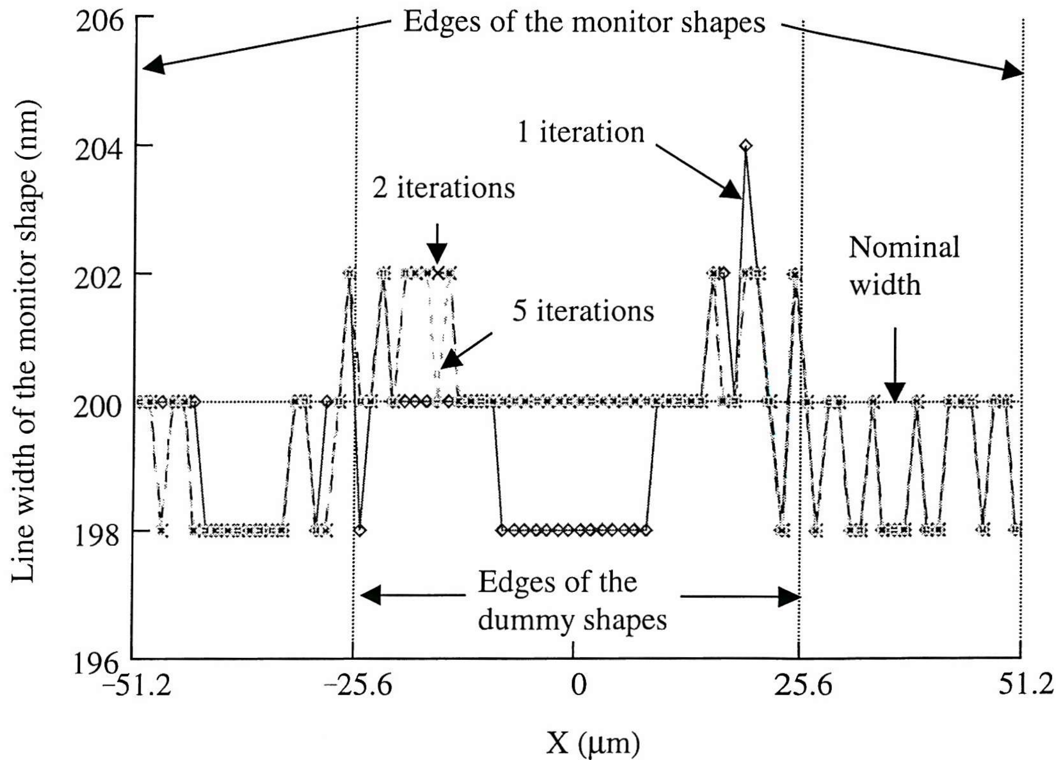


Figure 6-7 Line width variations of the monitor shape for KasugaTP corrected by EPADPEC.

Table 6-2 summarizes the maximum and the average magnitude of the monitor line width errors of the uncorrected KasugaTP and KasugaTP corrected by various PADPEC schemes for various values of M and λ of dummy shapes. In varying M, the order of $\boldsymbol{A}$ in equation (4-13) was also varied such that the physical area convolved with $\boldsymbol{A}$ remained roughly constant. EPADPEC with one iteration improves the maximum line width error by factors of 11(4) and 11(4) over KPADPEC and MPADPEC respectively for M = 2.56 μm and dummy shapes with λ = 100% (50%). For M = 2.56 μm / 5.12 μm / 10.24 μm and dummy shapes with λ = 75%, EPADPEC with one iteration reduces the maximum line width error by factors of 7.00 / 3.50 / 2.17 and 7.00 / 3.50 / 1.83 over KPADPEC and MPADPEC respectively. The significant smaller improvement factor of the M = 10.24 μm case is mainly due to the error in smoothing and interpolation steps as mentioned in section 5.1. Apart from the case of M = 5.12 μm and dummy shapes with λ = 75%, additional iterations in EPADPEC do not improve the maximum line width error. Thus, one iteration is the default setting for EPADPEC in the rest of this thesis, unless stated otherwise.

λ of the dummy shapes (%)	Mesh size (μm)	PEC used in the simulation	Smoothing filter size, $R_{sm} \times R_{sm}$ (meshes)	Maximum line width error (nm)	Average line width error (nm)
50	-	Without PEC	-	200	107.38
	2.56	MPADPEC	17×17	8	2.50
		KPADPEC after 10 iterations	3×3	8 (7)	2.25
		EPADPEC after 1 iteration	17×17	2	0.83
		EPADPEC after 2 iterations	17×17	2	0.38
		EPADPEC after 5 iterations	17×17	2	0.38
75	-	Without PEC	-	200	95.20
	2.56	MPADPEC	17×17	14	4.65
		KPADPEC after 10 iterations	3×3	14 (13)	4.25
		EPADPEC after 1 iteration	17×17	2	0.95
		EPADPEC after 2 iterations	17×17	2	0.45
		EPADPEC after 5 iterations	17×17	2	0.45
100	-	Without PEC	-	200	112.30
	2.56	MPADPEC	17×17	22	6.80
		KPADPEC after 10 iterations	3×3	22 (21)	6.30
		EPADPEC after 1 iteration	17×17	2	0.80
		EPADPEC after 2 iterations	17×17	2	0.40
		EPADPEC after 5 iterations	17×17	2	0.40
75	5.12	MPADPEC	9×9	14	5.05
		KPADPEC after 2 iterations	3×3	14 (13)	3.55
		EPADPEC after 1 iteration	9×9	4	1.08
		EPADPEC after 2 iterations	9×9	2	0.85
		EPADPEC after 5 iterations	9×9	2	0.83
	10.24	MPADPEC	5×5	22	10.40
		KPADPEC after 1 iteration	3×3	26 (27)	13.00
		EPADPEC after 1 iteration	5×5	12	4.53
		EPADPEC after 2 iterations	5×5	12	4.65
		EPADPEC after 5 iterations	5×5	12	4.68

The maximum line width error values in brackets are the results reported in [123].

Table 6-2 The maximum and the average line width errors of the uncorrected KasugaTP and KasugaTP corrected by the various PADPEC schemes.

For the case of the conventional PADPEC systems and $M = 2.56$ μm, the line width errors increase as λ of the dummy shapes increase (Table 6-2). This trend is expected since the proximity effect is more serious for denser dummy shapes. Figure 6-8 shows the variation of the line width errors with M and S for EPADPEC corrected KasugaTP. The dummy shapes have a λ value of 100% and this setting represents the worst case scenario for PEC. As in section 5.1.2, in varying M, the order of $\boldsymbol{A}$ in equation (4-13) was also varied such that the physical area convolved with $\boldsymbol{A}$ remained roughly constant (Table 6-3); this allows a more sensible comparison. For all values of S, the maximum line width errors saturate as M decreases. The minimum saturated value of the maximum line width error is 2 nm. For $S \leq M$ and a line width control of ±10% (±5%), the maximum allowable value of M is about 0.75β (0.4β). In the cases of KPADPEC and MPADPEC, the maximum allowable value of M is about 0.05β for a line width control of ±10% and $S \leq M$ (Figure 6-9). Thus, EPADPEC has increased the maximum allowable value of M by a factor of 15.

M (β)	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**
Physical width of $\boldsymbol{A}$ (β)	10.7	11.0	11.1	10.8	11.5	11.4	11.9	12.0	11.7	11.0

Table 6-3 Physical widths covered by the convolution matrix, *A* for various values of *M*.

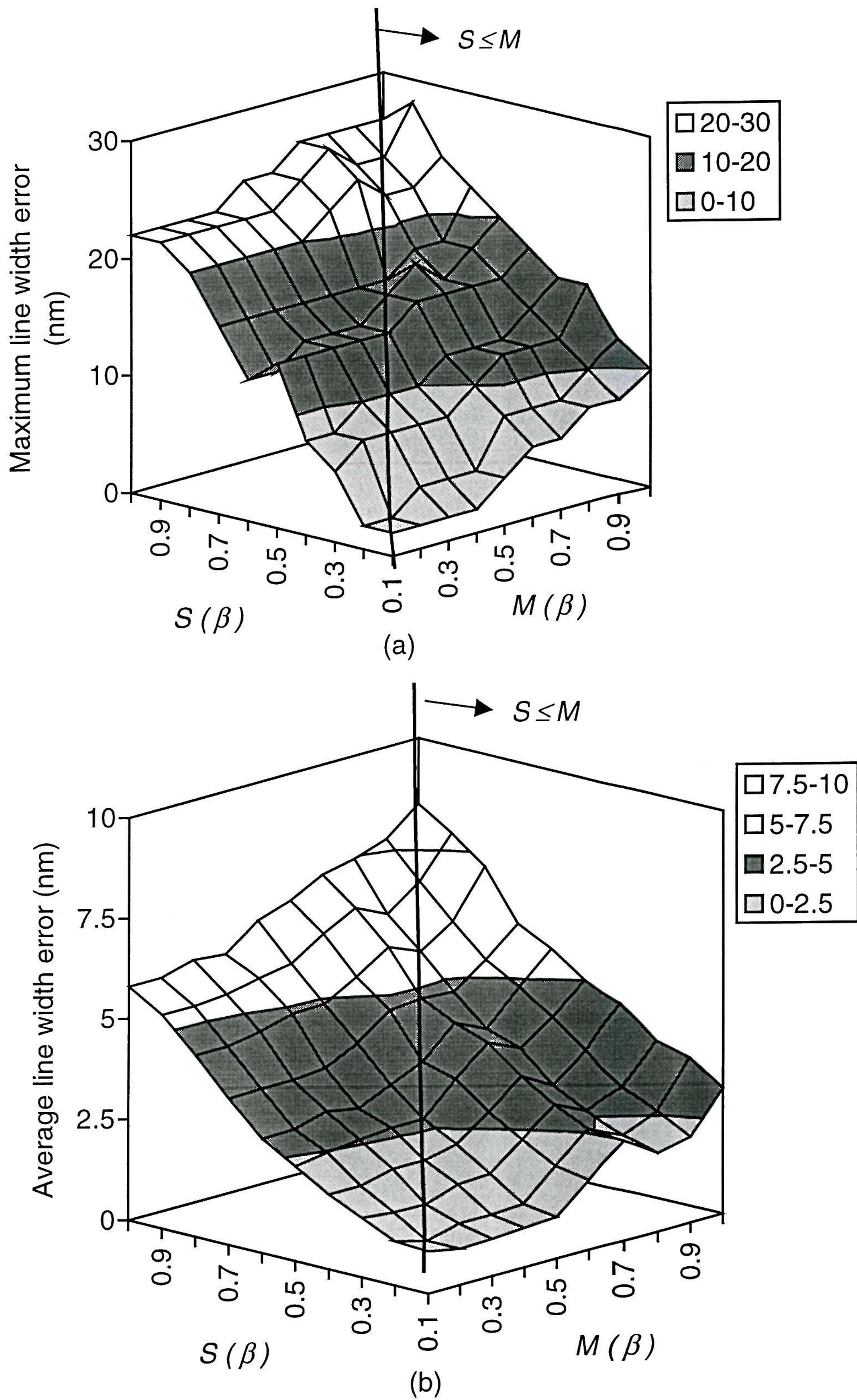
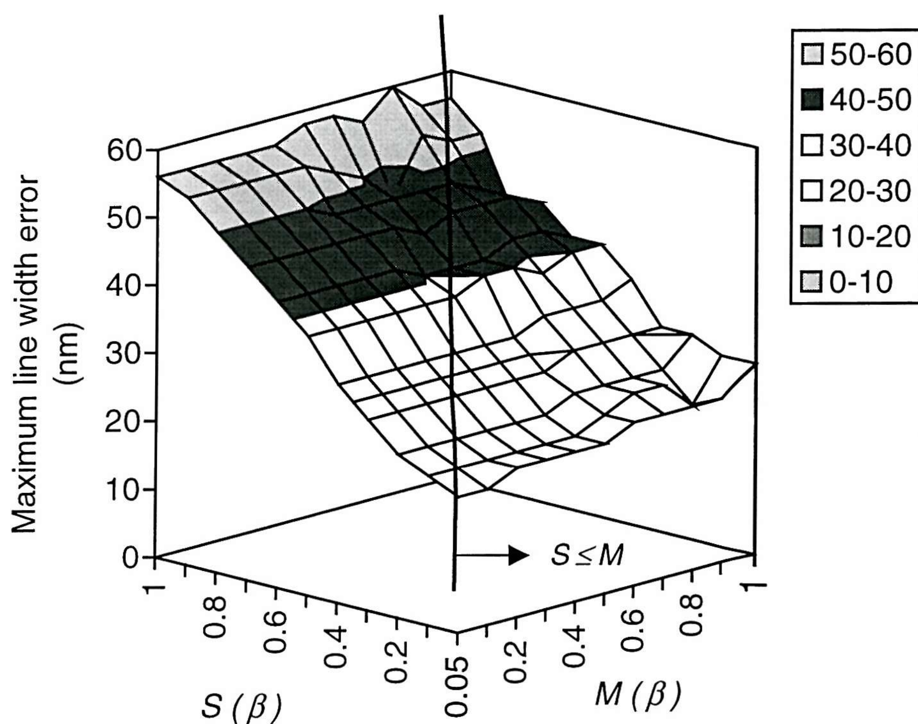
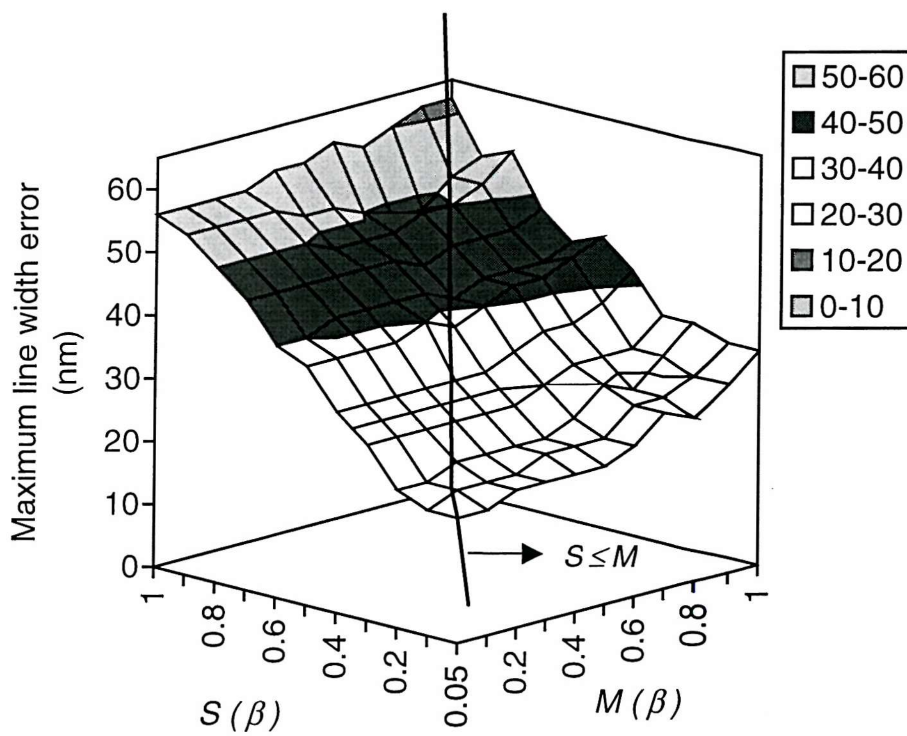


Figure 6-8 Variation of the line width errors with *M* and *S* for EPADPEC corrected KasugaTP.



(a)



(b)

KPADPEC uses the optimum iteration count as calculated according to equation (4-16). The interval between adjacent marks along the S and M axis is 0.1β, except for the first interval, which is 0.05β. In varying M, the same physical widths of A in Table 6-3 are used, and for $M=0.05\beta$, the width is 10.75β.

Figure 6-9 Variation of the maximum line width errors with *M* and *S* for (a) MPADPEC and (b) KPADPEC corrected KasugaTP.

Figure 6-10 shows the variation of the line width errors with the physical widths covered by the convolution matrix, *A*, for EPADPEC corrected KasugaTP. The dummy shapes have a λ value of 100% and $S=M=0.1\beta$. These values of *S* and *M* give the minimum saturated value of the maximum line width error (Figure 6-8), thus, they have negligible effects on the line width errors. The maximum line width error saturates at 2 nm when the physical width of *A* is greater or equal to 3.7β. There is a good correlation between the values of the maximum line width and the figure of merit, ζ (section 5.1.3); the larger the maximum line width, the smaller the value of ζ.

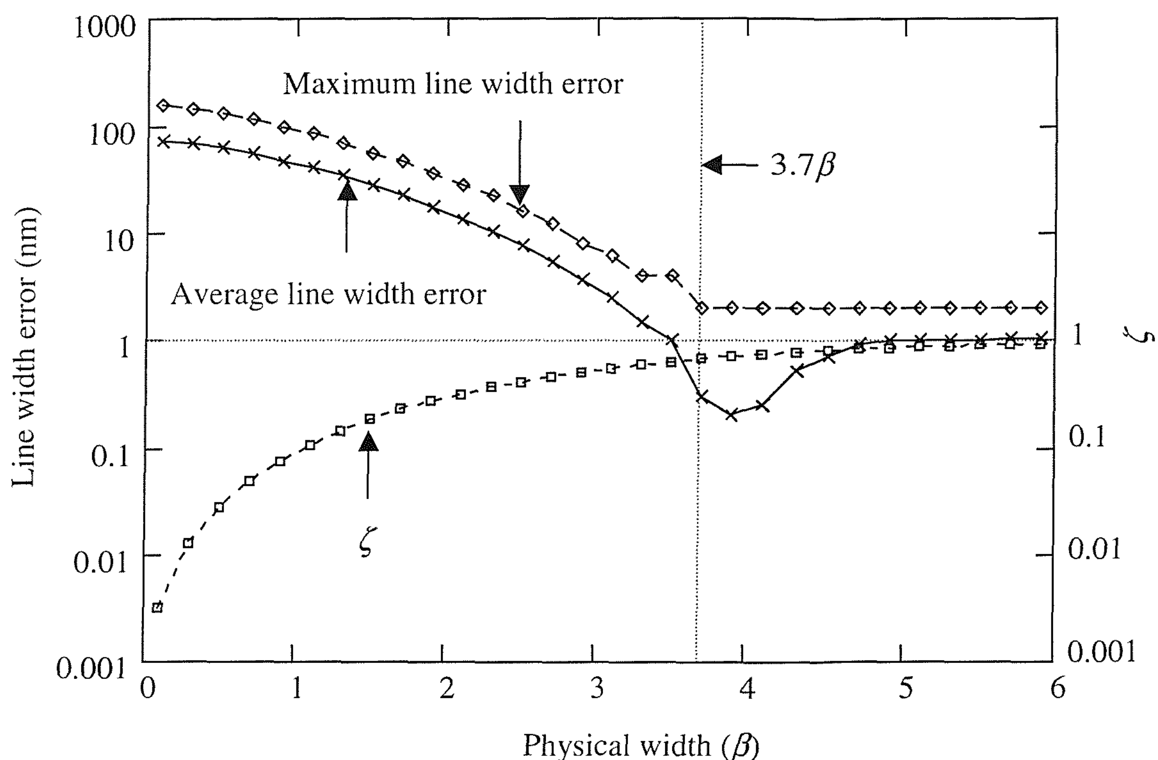


Figure 6-10 Variation of the line width errors and ζ with the physical widths covered by *A* for EPADPEC corrected KasugaTP.

Figure 6-11shows the simulated developed resist image (shaded regions) of EPADPEC corrected RauTP for an exel size of 25 nm, a simulation sample grid size of 25 nm, $M=S=0.75\beta$ and *A* with an order of 5. The values of *M* and *S* are the maximum allowable values for a ±10% line width control and $S\leq M$ (Figure 6-8(a)). The physical width of *A* is 3.75β that is sufficient to cover most of the back scattered electrons (Figure 6-10). The inner regions of the outline rectangles are the desired developed resist image. These rectangles are produced as a result of the framing and

shape fracturing procedures in EPADPEC. The simulated developed resist image matches the desired image except at shape corners.

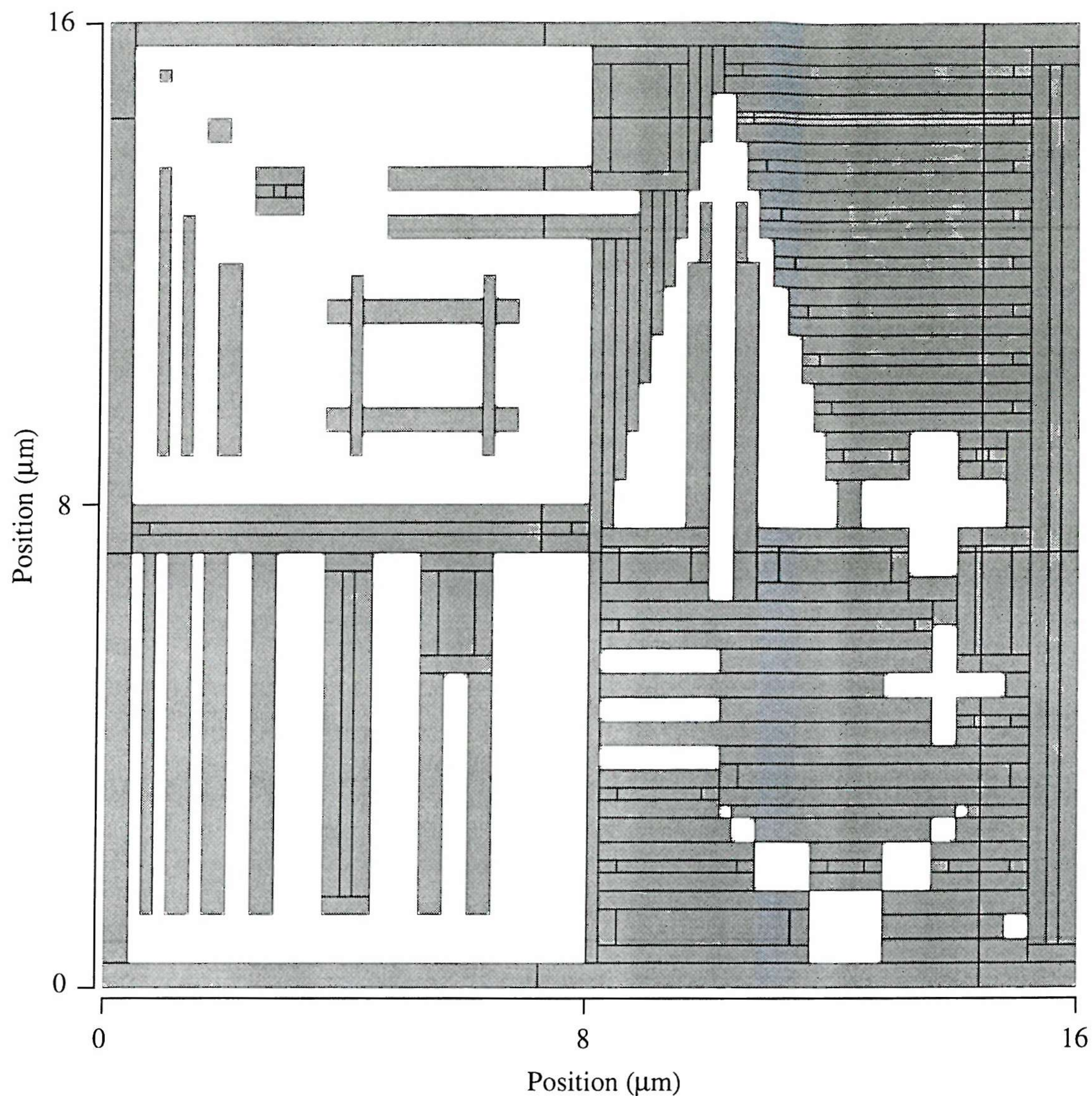


Figure 6-11 The simulated developed resist image (shaded regions) of EPADPEC corrected RauTP.

For LPEC (Figure 6-12), CAPROX with Manhattan fracturing (Figure 6-13), CAPROX with physical fracturing (Figure 6-14) corrected RauTP, the shape edges in the simulated developed resist image fall short of the desired shape edges. The interior holes that appear in the simulated developed resist image of CAPROX with Manhattan fracturing corrected RauTP (Figure 6-13) are the result of assigning the incident dose at the rectangle centre to the entire rectangle as illustrated earlier in section 5.2.1.3. By increasing all the incident doses by a constant amount, the correction accuracy can be improved. However, the amount of bias that produces the

optimum correction accuracy is pattern dependent. Thus, we will need to determine the optimum bias value experimentally or by simulations for each pattern, which is a highly unsatisfactory situation. The results clearly show the superiority of the correction accuracy achieved by EPADPEC over other PEC methods.

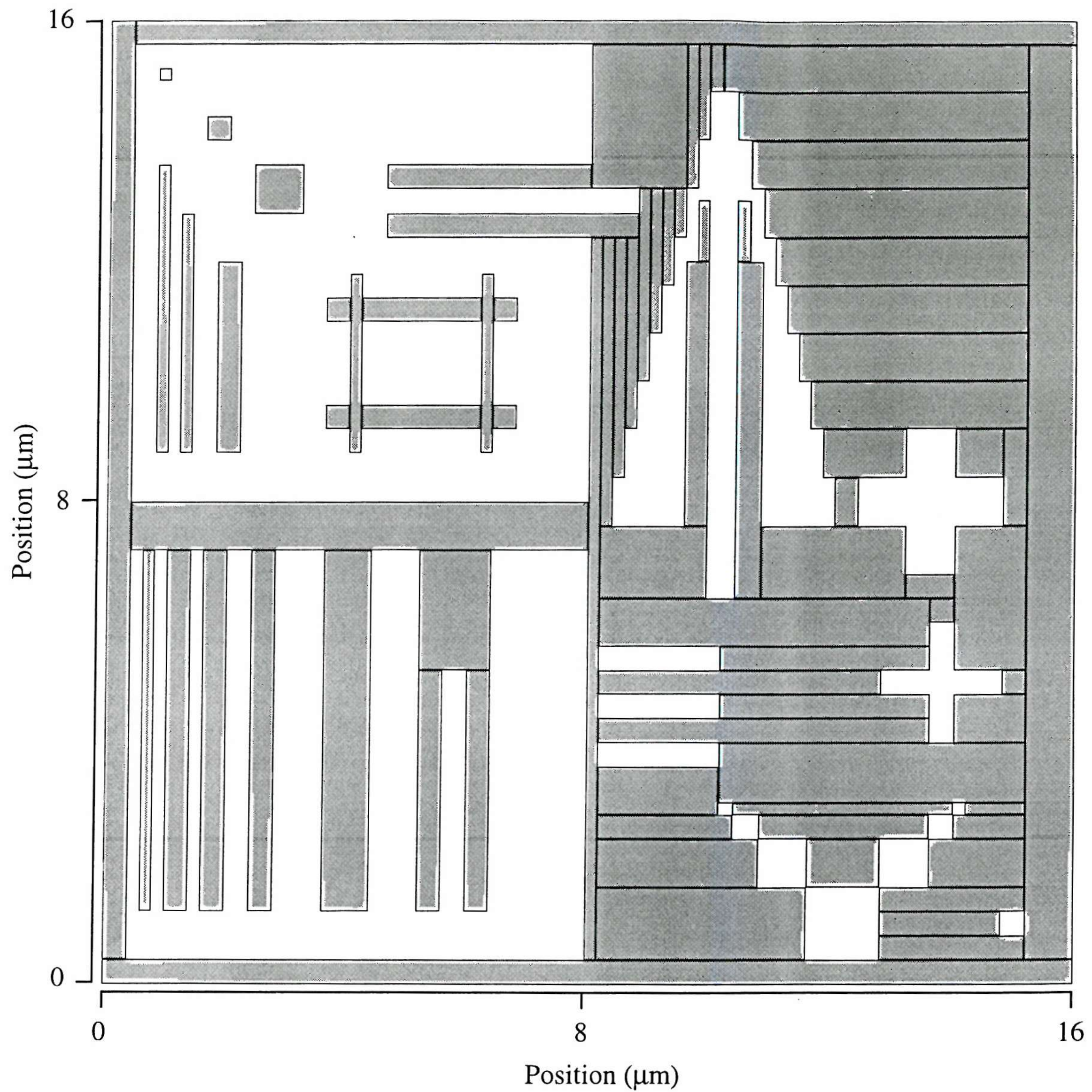


Figure 6-12 The simulated developed resist image (shaded regions) of LPEC corrected RauTP.

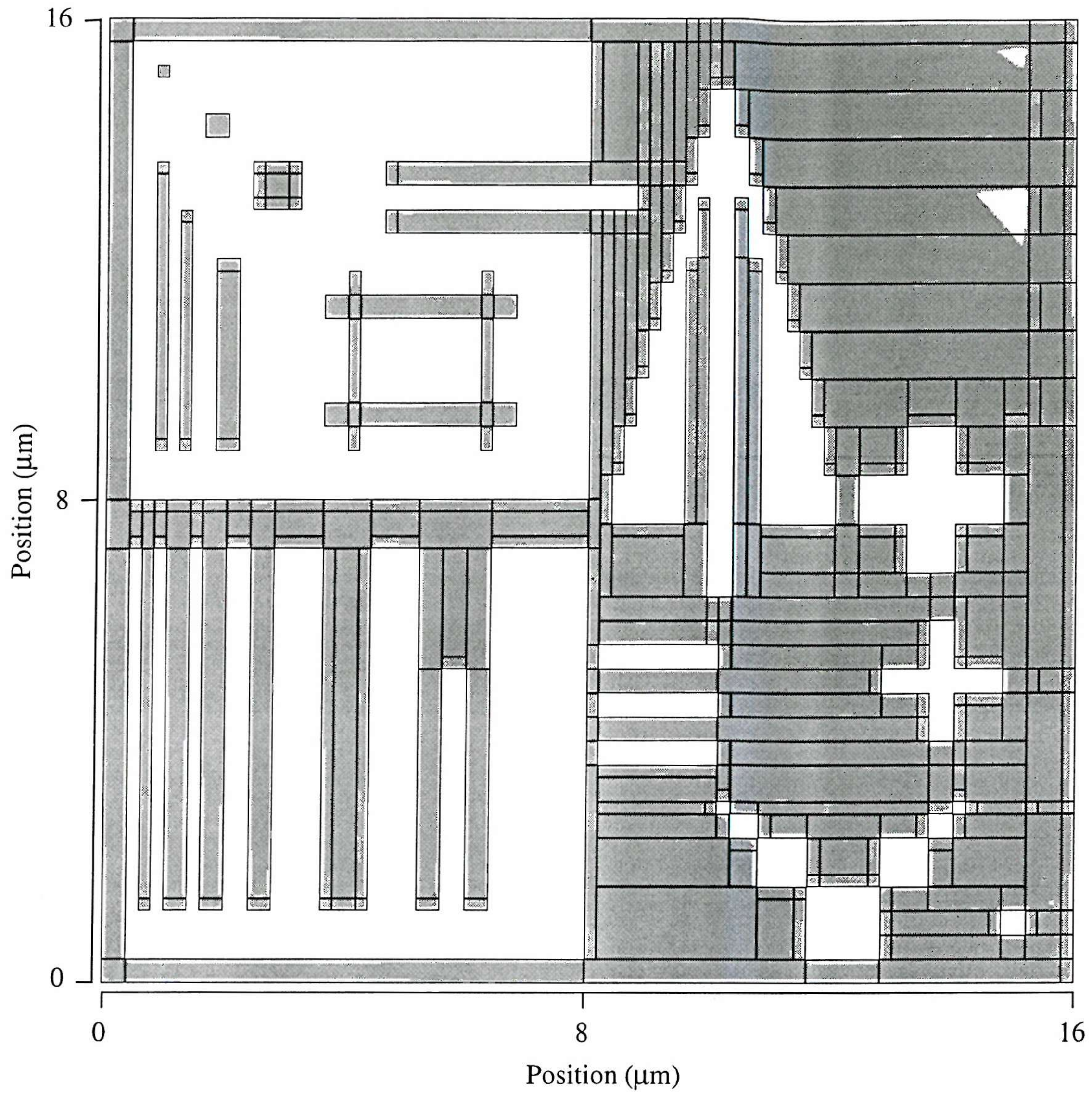


Figure 6-13 The simulated developed resist image (shaded regions) of CAPROX with Manhattan fracturing corrected RauTP.

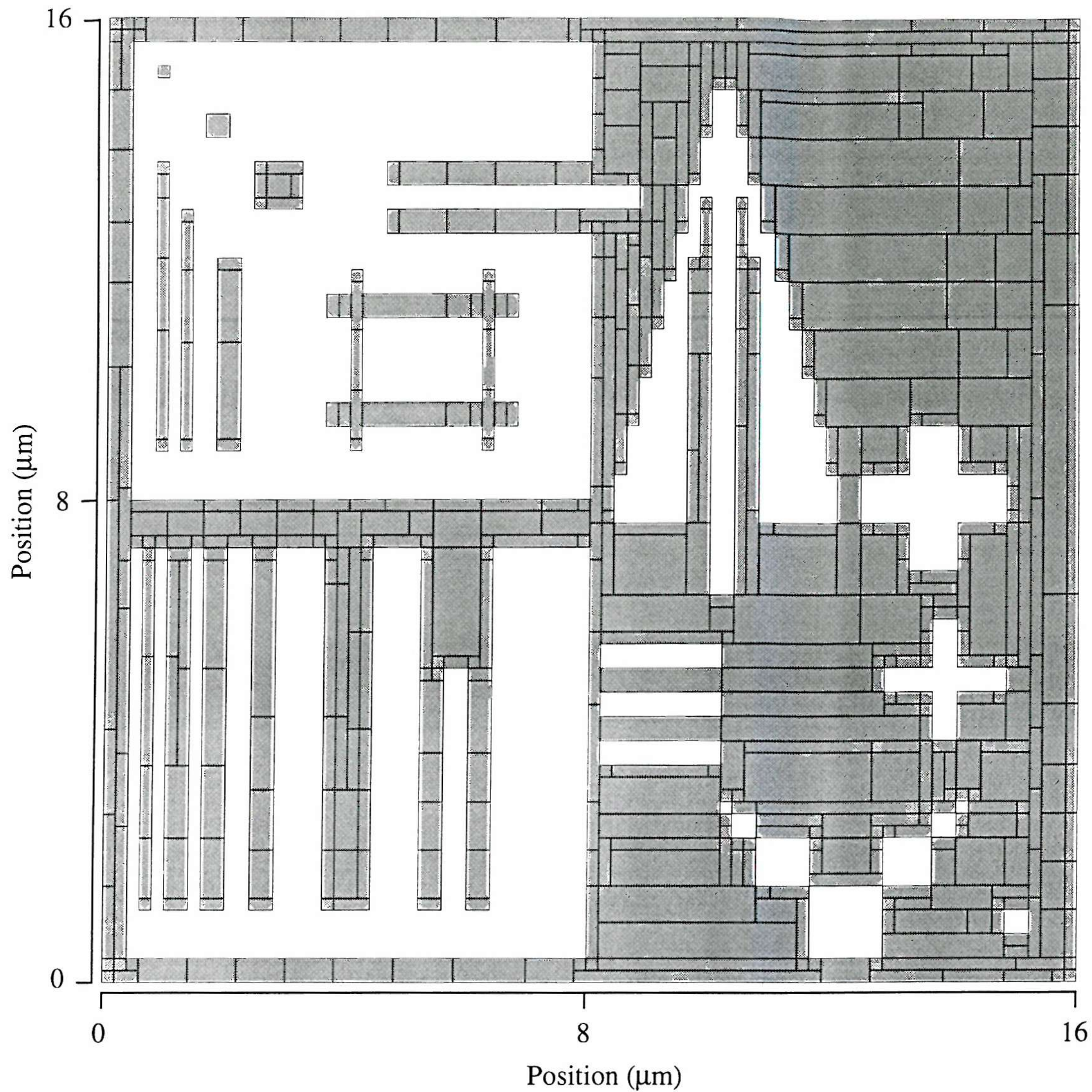


Figure 6-14 The simulated developed resist image (shaded regions) of CAPROX with physical fracturing corrected RauTP.

Table 6-4 shows the absolute area difference between the developed and the desired resist images of RauTP corrected by various PEC schemes. EPADPEC has reduced the absolute area difference by a factor of 40.3 / 42.8 / 29.1 compared with LPEC / CAPROX with Manhattan fracturing / CAPROX with physical fracturing.

PEC scheme	EPADPEC	LPEC	CAPROX with Manhattan fracturing	CAPROX with physical fracturing
Absolute area difference $(25nm)^2$	793	31981	33952	23099
Rectangle count	273	83	373	577

Table 6-4 The absolute area difference between the developed and the desired resist images of RauTP corrected by various PEC schemes.

6.2.2.2 Experimental results

Using the Leica EBMF 10.5 EBL machine in the Department of Electronics and Computer Science, University of Southampton, RauTP (with a minimum feature size of 0.4μm) with and without PEC has been fabricated on a silicon wafer covered with a 0.5μm thick UVIII resist. The EBL machine uses an acceleration voltage of 30kV, a beam current of 0.25nA, an exel size of 50nm and a *VRU*[1] value of zero.

The wafer was dehydrated at 190°C for 30 minutes. Next, it underwent HMDS (prima) for 3 minutes. By spinning the resist at 2500 rpm for 30 seconds, a 0.5μm thick layer of the UVIII resist was deposited on the wafer surface. Before exposure, the wafer is pre-baked for 60 seconds on a hot plate regulated to 140°C. After exposure, the wafer is post-baked for 60 seconds on a hot plate regulated to 140°C before it is developed in a MF322 developer for 90 seconds. Finally, it is rinsed for 60 seconds before it is spun dry at 4000 rpm for 60 seconds. Before examining the developed resist images using a scanning electron microscope, the wafer surface is sputtered with a thin layer of gold.

Figure 6-15 shows good agreement between the experimental line spread EDD and the line spread EDD predicted by a double Gaussian EDF for the lithography conditions stated in the preceding two paragraphs. The extracted parameter values of the double Gaussian EDF are α=0.168μm, β=4.18μm and η=0.25. [131] contains the details of the experiment and the procedure for extracting the parameter values.

---

[1] This parameter allows the use of a large beam diameter to write every 2nd, 4th or 8th exel of the field. Every 2 to the power of VRU exel is exposed, so a VRU value of 0 will expose every exel.

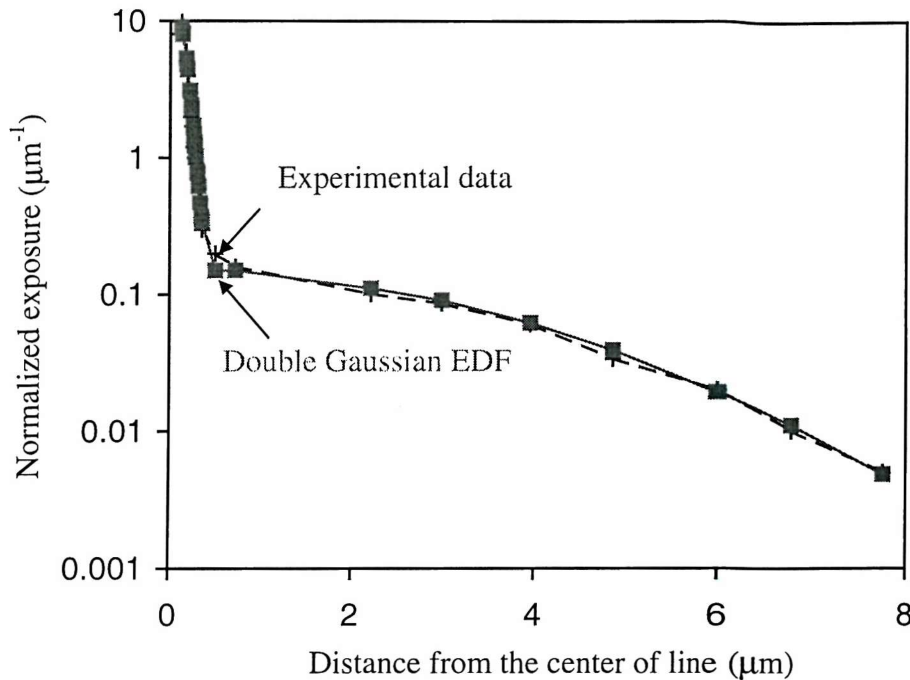
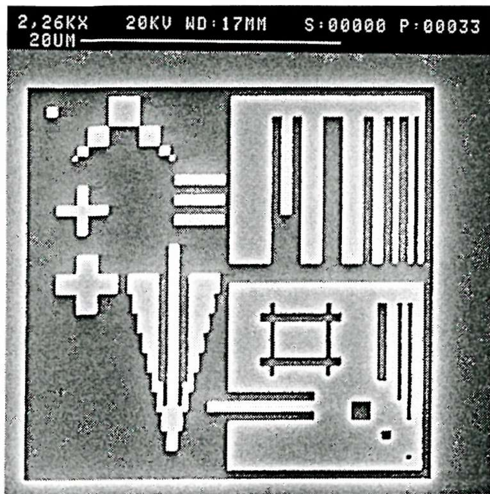
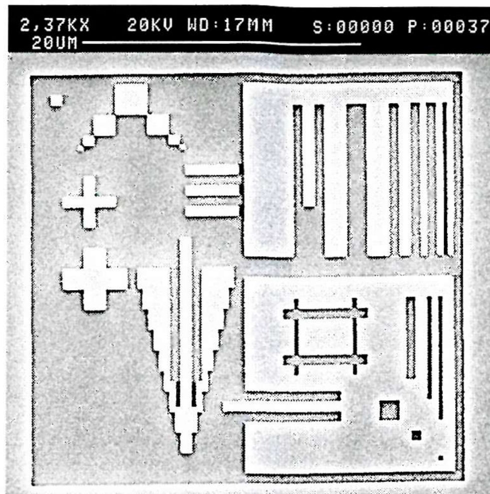


Figure 6-15 Experimental and analytical line spread EDDs.

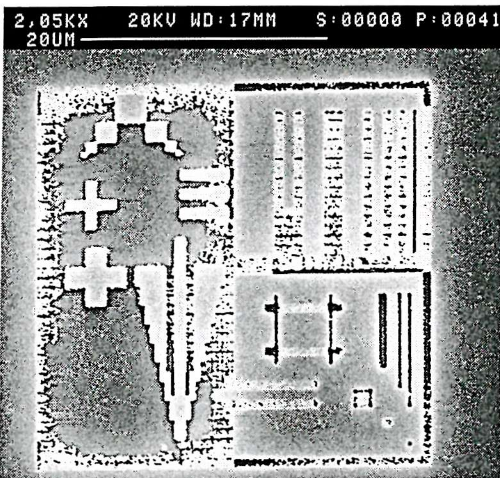
Figure 6-16 shows the scanning electron microscope photographs of RauTP with and without PEC. The standard clock frequency is set to 2.506 MHz. The standard clock frequency is assigned to shapes with a normalized incident dose of 1. Without PEC, the resist fails to develop (disappear) fully at written regions. LPEC and CAPROX with Manhattan fracturing produce some improvement, but most of the written regions are still under-developed. EPADPEC and CAPROX with physical fracturing manage to produce the desired resist image. Although the quality of the results is comparable, EPADPEC is much faster than CAPROX with physical fracturing: this is discussed in section 6.3.2.



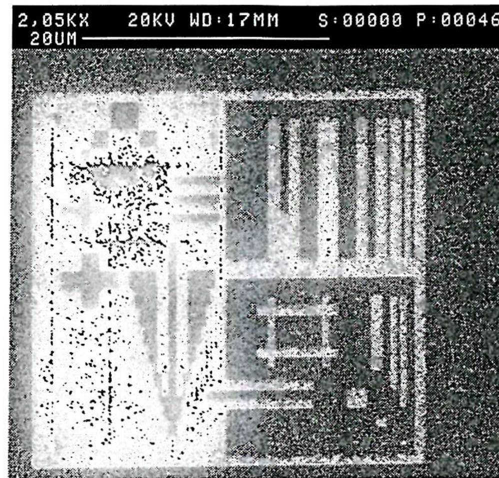
(a) EPADPEC



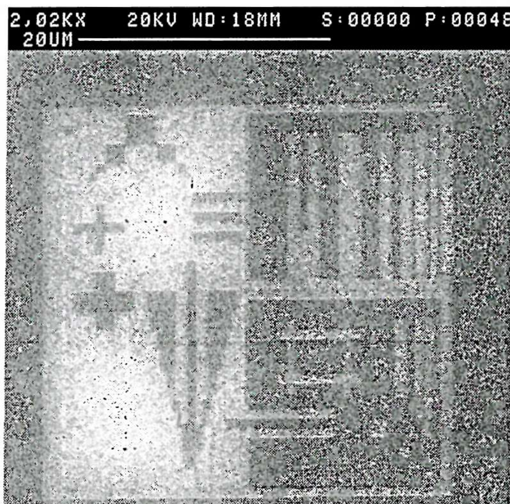
(b) CAPROX with physical fracturing



(c) CAPROX with Manhattan fracturing



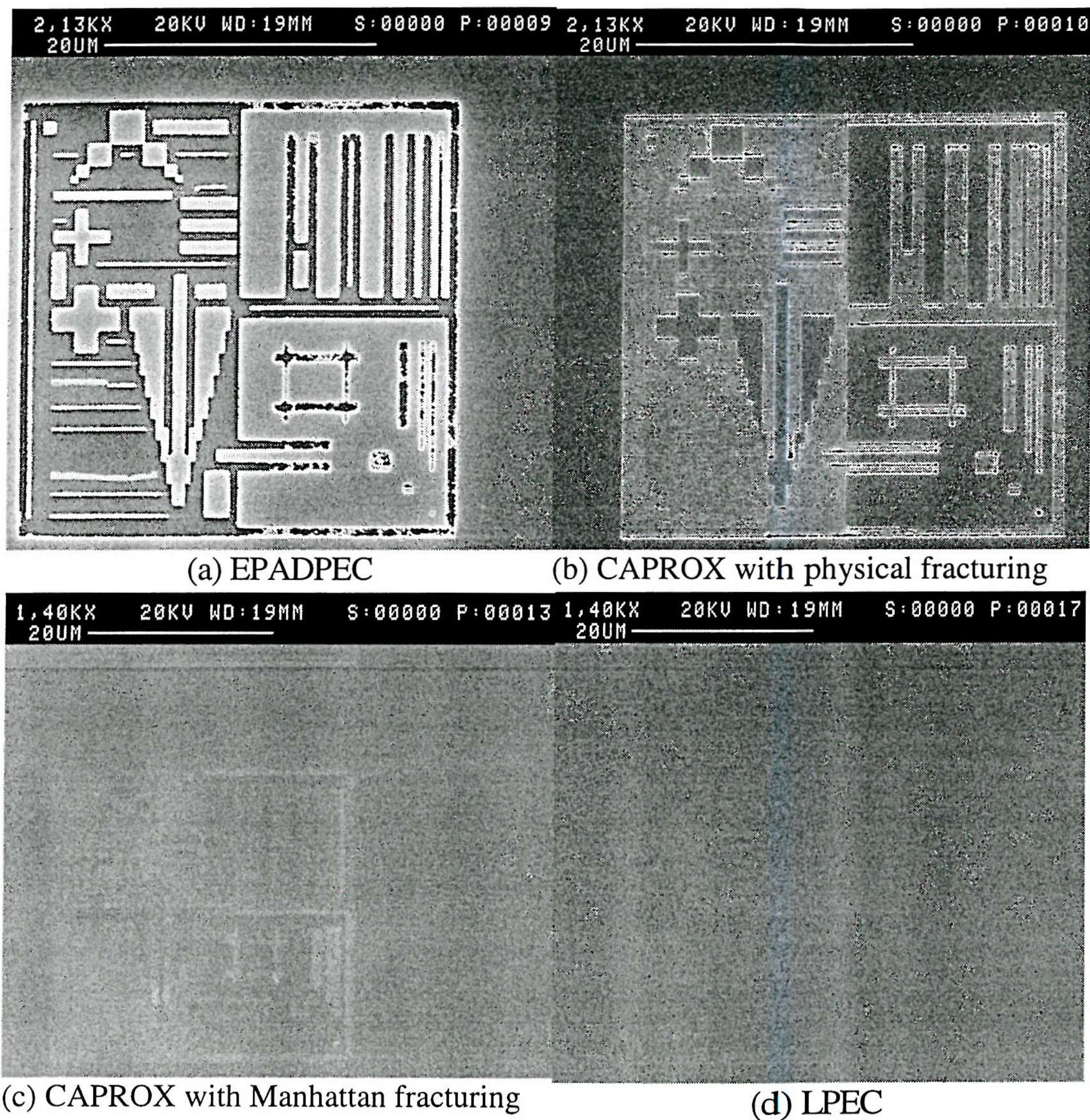
(d) LPEC



(e) No PEC

Figure 6-16 Scanning electron microscope photographs of RauTP with and without PEC.

Being a positive tone *chemically amplified* resist, the UVIII resist has a low process latitude [133, 134], i.e. the resist threshold energy density is sensitive to process variations. Figure 6-17 shows the scanning electron microscope photographs of RauTP with PEC, fabricated under identical lithographic conditions except for an additional delay of two days between the pre-bake and the exposure. Without PEC, the resist remains intact at written regions. The value of the resist threshold energy density has increased by about 50%. Further work is needed to determine the resist threshold energy density value accurately, and to characterize and optimize the process so that identical results are reproducible under identical process procedures. The results (Figure 6-16) demonstrate the clear superiority of EPADPEC over most other PEC schemes in terms of correction accuracy.



(a) EPADPEC (b) CAPROX with physical fracturing

(c) CAPROX with Manhattan fracturing (d) LPEC

Figure 6-17 Scanning electron microscope photographs of RauTP with PEC.

6.3 Timing

Figure 6-18 shows the definition of correction time, t_{PEC} of a PEC method in the data flow of a typical electron beam data preparation system. The relative correction speed of various PEC schemes are compared based on their t_{PEC} values. t_{PEC} of EPADPEC corresponds to the computation time between points A and B in the data flow of EPADPEC (Figure 5-16).

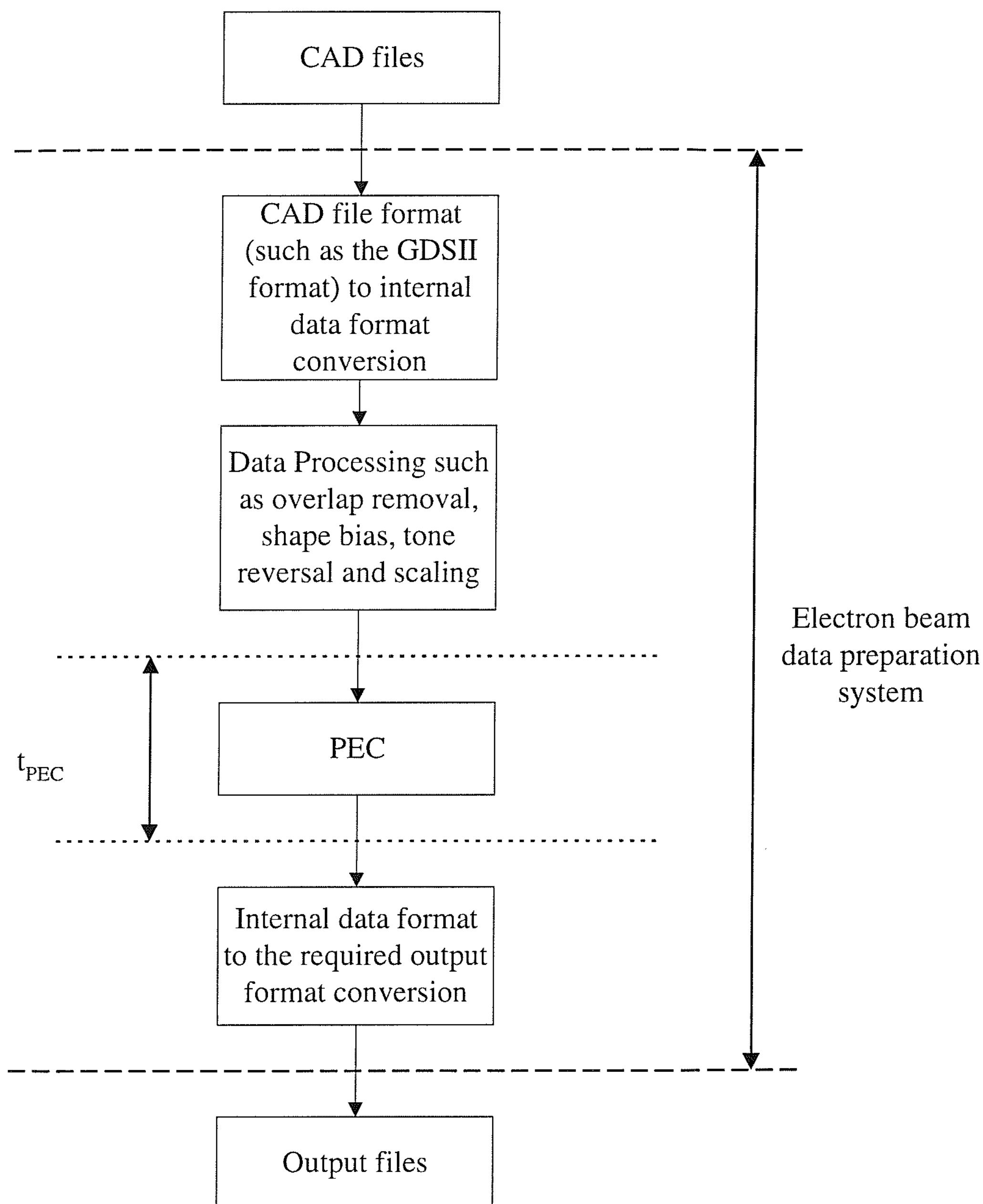


Figure 6-18 The data flow of a typical electron beam data preparation system.

6.3.1 Method of time measurement

As the data preparation system from Leica [135] only writes the computation time for the *entire* conversion in a log file, so t_{PEC} of LPEC is calculated as the time difference between the conversions with and without LPEC. On the other hand, CAPROX does write t_{PEC} in a log file. The library functions, QueryPerformanceFrequency() and QueryPerformanceCounter() of Microsoft Visual C++ 4.0 are used to measure the t_{PEC} values of the PADPEC methods as shown by the following pseudo codes:

```
QueryPerformanceFrequency(Frequency);
QueryPerformanceCounter(Start_time);
PEC codes
QueryPerformanceCounter(End_time);
tPEC = ( End_time - Start_time ) / Frequency;
```

The counter interval is 0.84 μs. For each pattern, the t_{PEC} value of a PEC system is the average value of five time measurements.

The PADPEC methods and CAPROX are run on a Gateway 2000 G6-180 system with a Pentium Pro 180 MHz microprocessor, while LPEC is run on a Digital Equipment Corporation 3000 M600 workstation with an Alpha 21064 175 MHz processor. To estimate the relative speed of the two systems, the computation times of the following codes on both systems are measured:

```
unsigned long i;
double dSum=0.0;
double dFactor = 1.23456789;
for(i=0; i< 0xFFFFFFFF; i++)
{
        dSum *= dFactor;
        dSum += dFactor;
        dSum /= dFactor;
}
```

Taking the average of five measurements, the value on G6-180 is 0.811 ± 0.0001 relative to M600. The floating point performance on the benchmarking program, SPECfp92 of an Intel Alder system with a Pentium Pro 180 MHz microprocessor is 0.649 relative to a Digital Equipment Corporation 3000 M600S server with an Alpha 21064 175 MHz [136]. The t_{PEC} value of LPEC on G6-180 is estimated as (t_{PEC} value on M600) $\times$ 0.811.

6.3.2 Comparison

All the t_{PEC} values reported in this section are measured or estimated times on the G6-180 system. To assess the influence of M, S and the order of A on the t_{PEC} of EPADPEC, layer 11 (polysilicon gate layer) of MP16 is used (see Figure 6-3 and Table 6-1) as the critical layer contains the smallest feature size, even though it does not contain the highest number of shapes. Figure 6-19 shows that t_{PEC} is effectively proportional to the square of the order of A. In varying the order of A, EPADPEC uses M=S=0.75β.

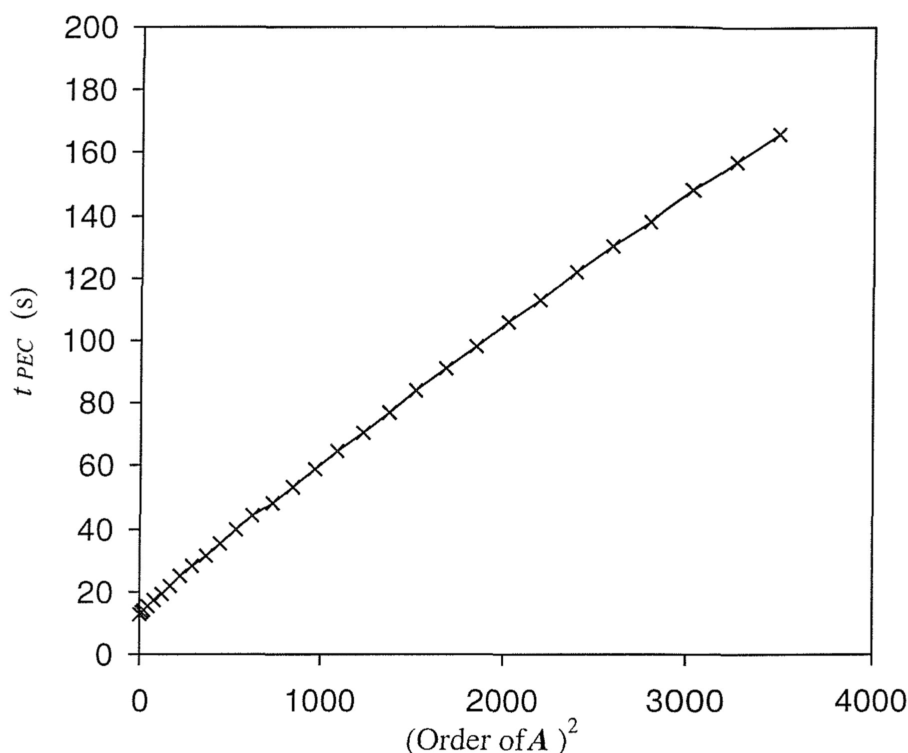
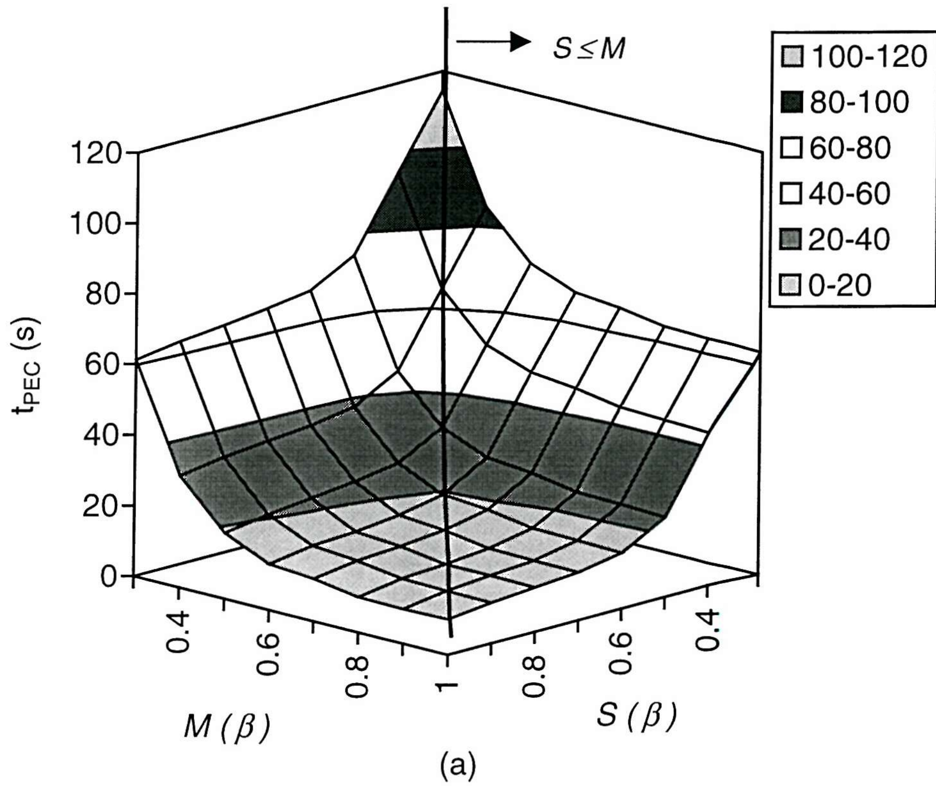
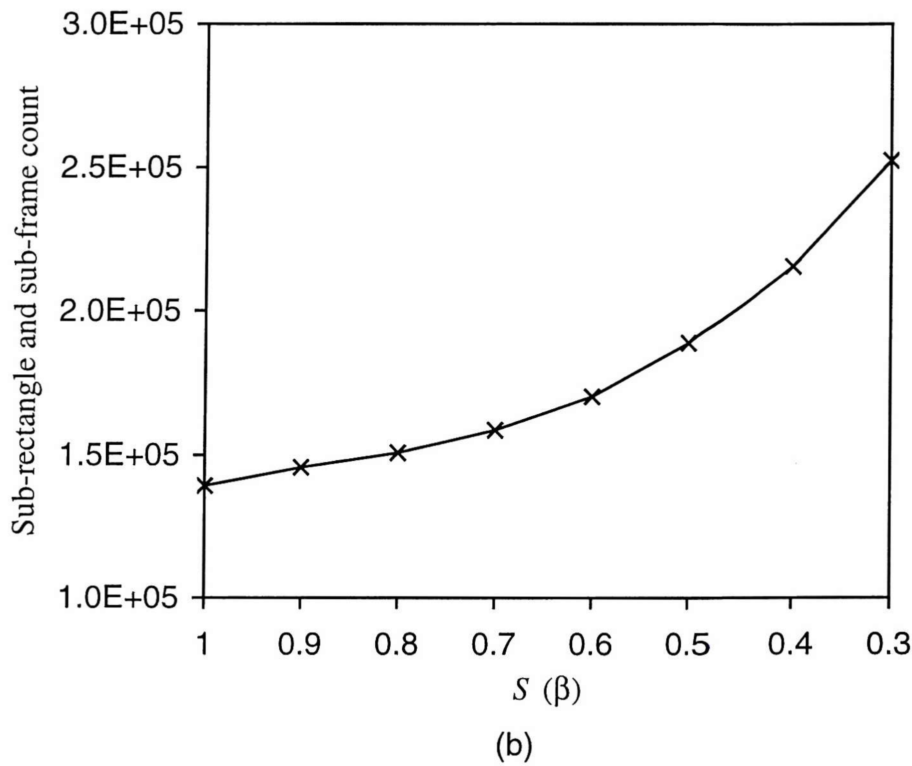


Figure 6-19 Variations of t_{PEC} of EPADPEC with the square of the order of A for layer 11 of MP16.

Figure 6-20(a) shows the variations of t_{PEC} of EPADPEC with M and S for layer 11 of MP16. In varying M and S, EPADPEC uses the smallest order of A that satisfies the minimum physical width of 3.7β. (The minimum width for achieving the saturated maximum monitor line width error of KasugaTP (section 6.2.2)). As the physical width is roughly constant, the order of A is inversely proportional to M. Since *t[ComputeDIMap]* (the computational time for deriving the D_I map from the λ map (P4-P6 in Figure 5-16)) is proportional to the square of the order of A and the total





(a) and (b) use the same S abscissa co-ordinate system for clearer comparison.

Figure 6-20 (a) Variation of t_{PEC} of EPADPEC with M and S for layer 11 of MP16. (b) Variation of the sub-rectangle and sub-frame count after shape fracturing with S for layer 11 of MP16.

number of mesh sites, so t_{PEC} is inversely proportional to M^4 when *t[ComputeDIMap]* is the dominant component. This trend is shown by the variation of t_{PEC} with small values of M for a fixed S value (Figure 6-20(a)). Figure 6-20(b) shows the variation of the sub-rectangle and sub-frames count after shape fracturing with S. Comparing this variation and the variation of t_{PEC} with S for a fixed M value (Figure 6-20(a)) suggests that t_{PEC} is proportional to the sub-rectangle and sub-frames count in these cases. The variations of t_{PEC} of other PADPEC schemes (Figure 6-21 and Figure 6-22) show similar dependency on S as for the EPADPEC case (Figure 6-20(a)). However, t_{PEC} depends less strongly on M compared with the EPADPEC case.

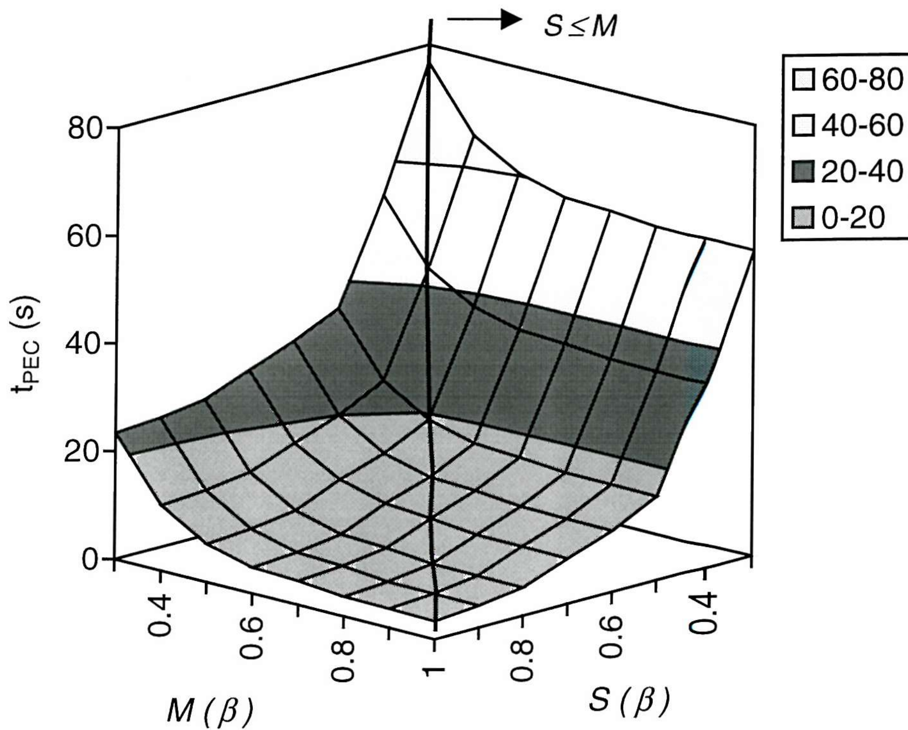


Figure 6-21 Variations of t_{PEC} of MPADPEC with M and S for layer 11 of MP16.

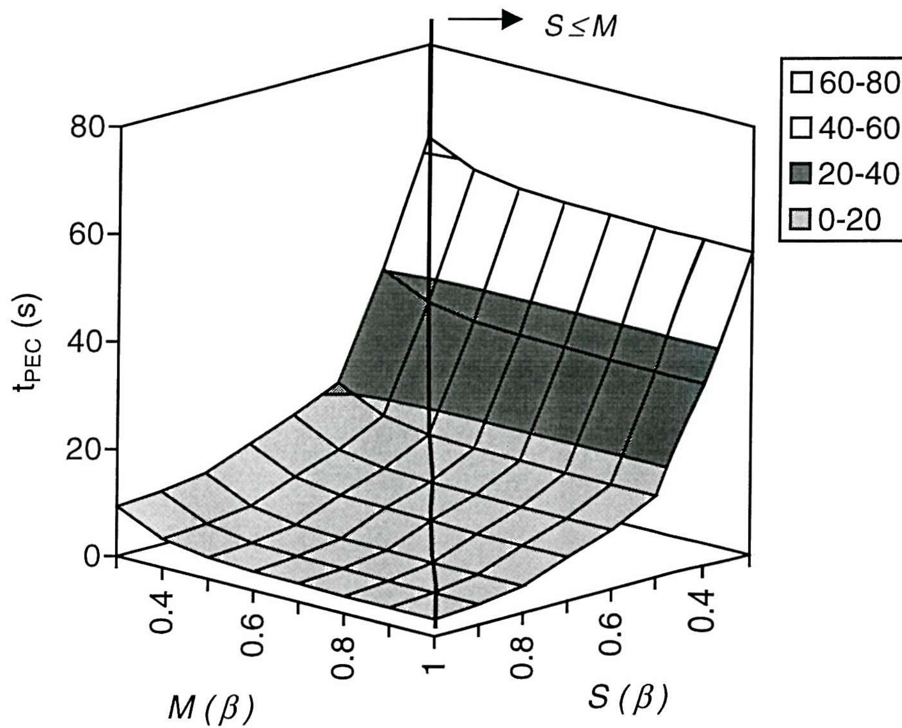
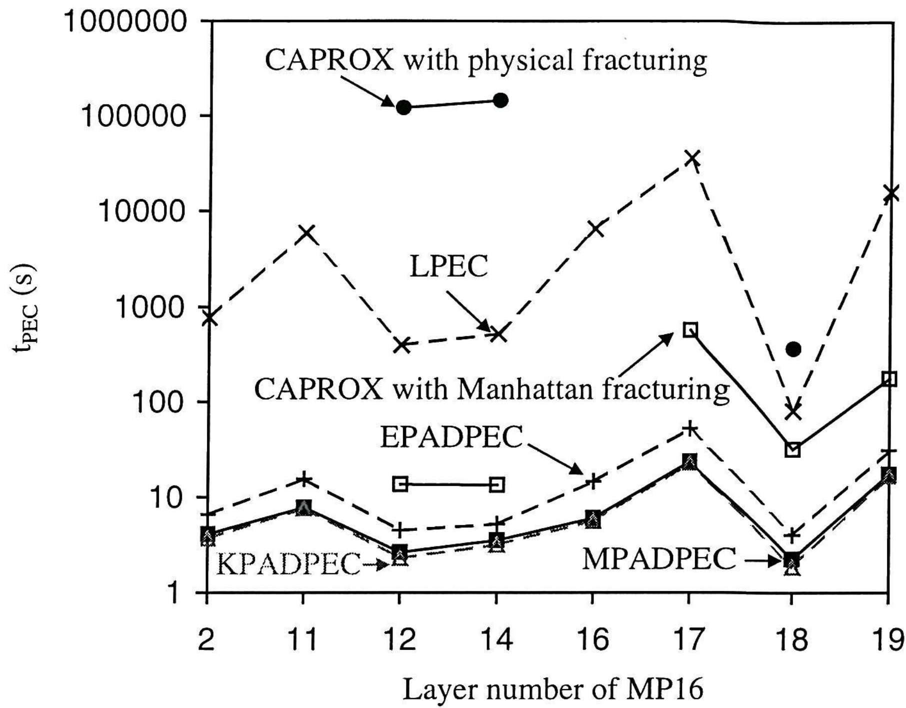
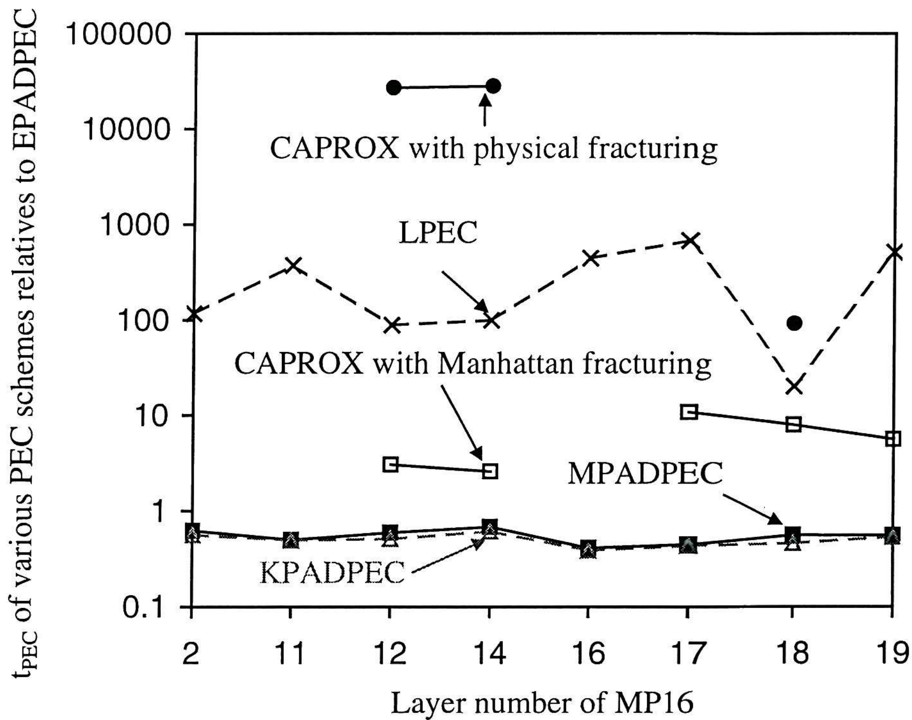


Figure 6-22 Variations of t_{PEC} of KPADPEC with *M* and *S* for layer 11 of MP16.

Figure 6-23(a) shows the variations of t_{PEC} of various PEC schemes with various layers of MP16, whereas Figure 6-23(b) shows the corresponding t_{PEC} of various PEC schemes relative to t_{PEC} of EPADPEC. All the PADPEC schemes use $M=S=7.2\mu m=0.75\beta$ that are the maximum allowable values for a ±10% line width control and $S\leq M$ (Figure 6-8(a)). MPADPEC and EPADPEC set the order of $\boldsymbol{A}$ to 5 with the corresponding physical width of 3.75β that is sufficient to cover most of the back scattered electrons (Figure 6-10). On the other hand, KPADPEC uses the optimum iteration count for an order of $\boldsymbol{A}$ of 3. Layers 2,11 and 16 of MP16 cause internal errors in CAPROX, while layer 17 requires too much memory (>1 GByte) and layer 19 causes internal errors in CAPROX with physical fracturing. Therefore, the corresponding data points are omitted in the graphs (Figure 6-23).



(a)



(b)

Figure 6-23 Speed comparison among various PEC schemes.

From Figure 6-23(b), the t_{PEC} values of CAPROX with physical fracturing (LPEC/CAPROX with Manhattan fracturing/MPADPEC/KPADPEC) relative to the corresponding t_{PEC} values of EPADPEC ranges from 92.3 (20.2/2.59/0.412/0.385) to

28000 (679/10.8/0.681/0.609). Although EPADPEC is computationally slower than other PADPEC schemes in these cases, EPADPEC gives better correction accuracy as shown in section 6.2.2. However, if we require the PADPEC methods to produce about the *same* correction accuracy (estimated using the maximum monitor line width error of KasugaTP), then EPADPEC is computationally faster than other PADPEC schemes. For example, if the required maximum line width error is 28nm, then the t_{PEC} values of MPADPEC and KPADPEC relative to the t_{PEC} value of EPADPEC are 5.23 and 5.08 respectively. These values are the minimum values that satisfy the $S \leq M$ condition (From Figure 6-8(a) and Figure 6-20(a), M=1.0β and S=0.9β for EPADPEC; from Figure 6-9(a) and Figure 6-21, M=0.6β and S=0.3β for MPADPEC; from Figure 6-9(b) and Figure 6-22, M=0.5β and S=0.3β for KPADPEC).

6.4 Effect of quantizing the incident dose

As some EBL machines (for example the Leica EBMF 10.5), only support a finite number of distinct values of the incident dose, D_I in a pattern, so it is necessary to quantize the continuous (*analogue*) D_I values generated by a PEC scheme. Thus, it is worthwhile investigating the effect of this quantization on the correction accuracy of EPADPEC.

Figure 6-24 and Figure 6-25 show the variation of the line width errors with the number of quantization levels for EPADPEC corrected KasugaTP. The dummy shapes have a λ value of 100% and S=M. EPADPEC uses the uniform upper-level quantization strategy (section 5.2.1.3). As in section 6.2.2, the same physical widths covered by A are used for various M values (Table 6-3). For all values of S and M, except 0.2β, the maximum line width errors saturate when the number of quantization levels is greater or equal to 256. For S=M=0.4β (line width control of ±5%, see section 6.2.2) and 0.75β (line width control of ±10%), a quantization level count of at least 128 is necessary to achieve the saturated maximum line width error.

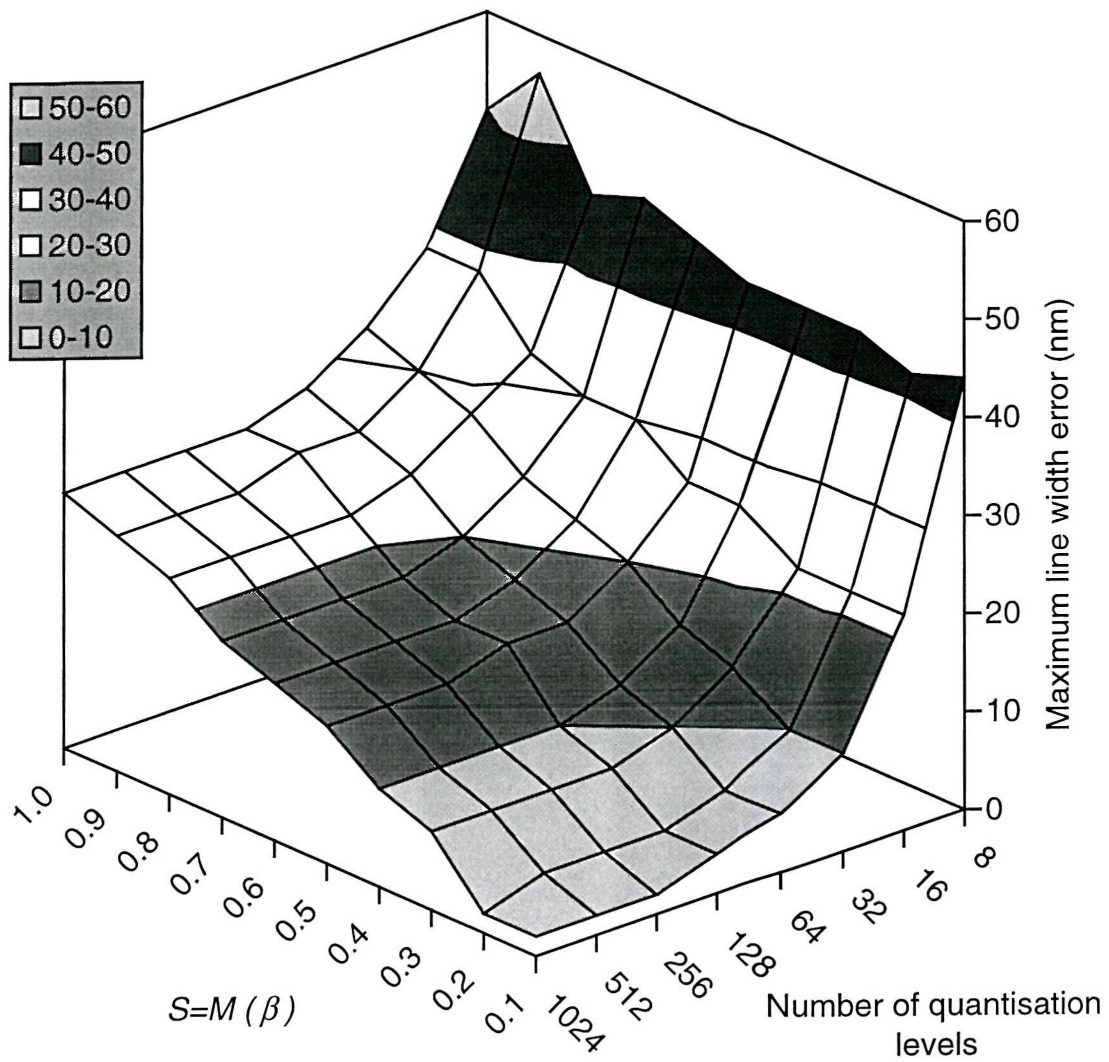


Figure 6-24 Variation of the maximum line width errors with the number of quantization levels for EPADPEC corrected KasugaTP.

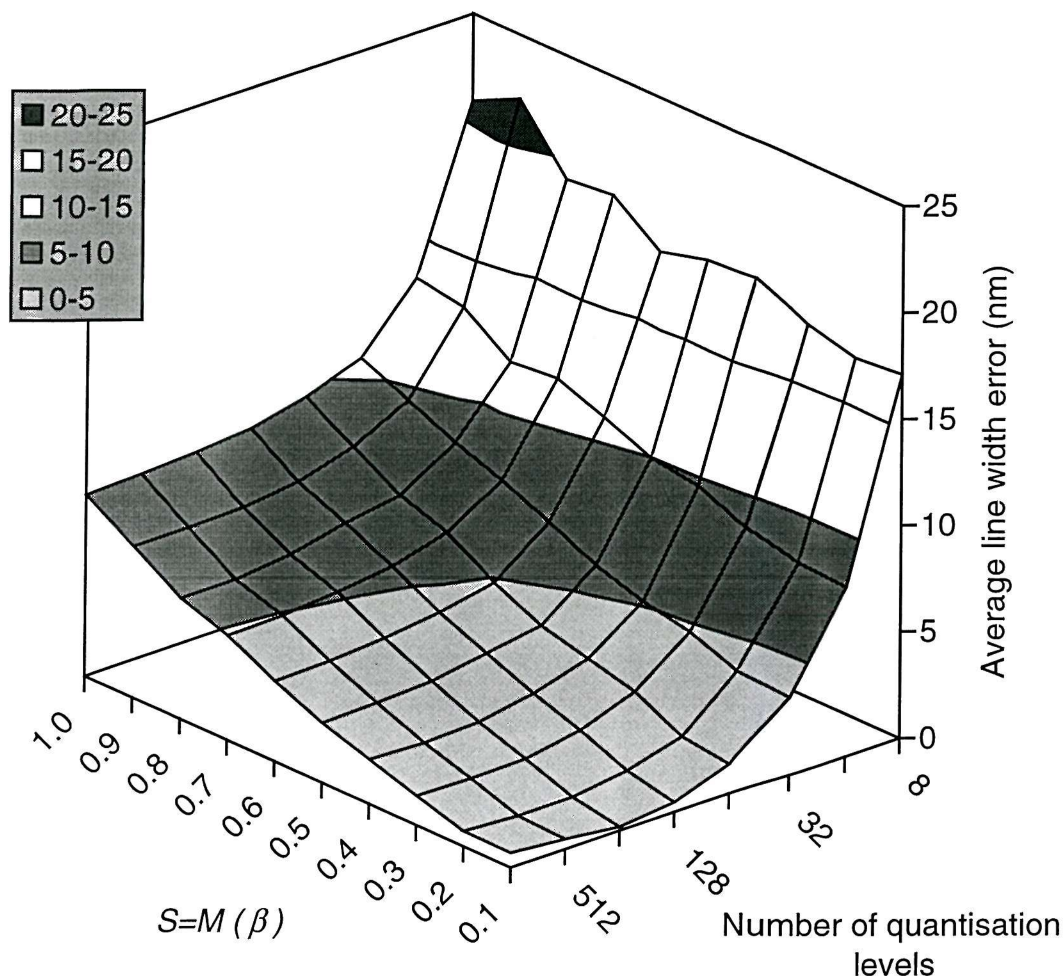


Figure 6-25 Variation of the average line width errors with the number of quantization levels for EPADPEC corrected KasugaTP.

6.5 Conclusions

The results presented in this chapter have demonstrated the performance superiority of EPADPEC over other PEC schemes in terms of correction accuracy and speed. Compared to KPADPEC and MPADPEC, EPADPEC reduces the maximum line width error of KasugaTP by factors ranging from 4 to 11 for $M = 2.56\mu m$. Although the correction speed of EPADPEC is 1.47-2.60 times slower than other PADPEC schemes for the same S and M sizes, EPADPEC is 5.08-5.23 times faster if the same correction accuracy is required from all the PADPEC schemes. On the other hand, EPADPEC cuts down the absolute area difference of RauTP by factors ranging from 29.1 to 40.3 compared with LPEC and CAPROX. EPADPEC is between 2.59 and

28000 times faster than LPEC and CAPROX. Table 6-5 summarises the performance of the PEC schemes relative to EPADPEC.

In general, the maximum line width error of EPADPEC corrected KasugaTP decreases as M and S decreases. For all values of S, the maximum line width errors saturate as M decreases. The minimum saturated value of the maximum line width error is 2 nm. For $S \leq M$ and a line width control of ±10%, EPADPEC has increased the maximum allowable value of M by a factor of 15 compared with MPADPEC and KPADPEC. The correction error associated with using a finite filter size is negligible as long as the filter (A) covers a physical width of at least 3.7β. For line width controls of ±10% and ±5%, a quantization level count of at least 128 is needed to minimize the quantization errors when quantizing the incident dose. When only one parameter is varied at a time and other parameters are kept constant, the correction speed of EPADPEC increases as the order of A and the sub-rectangle and sub-frames count decrease, and as M increases.

PEC scheme	Correction speed relative to EPADPEC		Correction accuracy relative to EPADPEC			The superiority of EPADPEC
	Test pattern	t_{PEC}	Test Pattern	Maximum line width error for M=2.56β and S=1μm (nm)	Absolute area difference (25 nm)^2	
KPADPEC	MP16	0.385-0.609	KasugaTP (Figure 6-1)	4-11	-	• EPADPEC is more accurate. • EPADPEC is faster for the same accuracy requirement. • Large shapes are not over-exposed in EPADPEC. • EPADPEC increases shape edge contrast.
MPADPEC	MP16	0.412-0.681	KasugaTP	4-11	-	• same as the superiority of EPADPEC over KPADPEC.
LPEC	MP16	20.2-679	RauTP (Figure 6-2)	-	40.3	• EPADPEC is faster and more accurate. • EPADPEC does not generate negative incident doses in the solution.
CAPROX with Manhattan fracturing	MP16	2.59-10.8	RauTP	-	42.8	• same as the superiority of EPADPEC over LPEC.
CAPROX with physical fracturing	MP16	92.3-28000	RauTP	-	29.1	• same as the superiority of EPADPEC over LPEC.

Table 6-5 Summary of the performance of various PEC schemes relative to EPADPEC.

Chapter 7

Further improving the EPADPEC algorithm

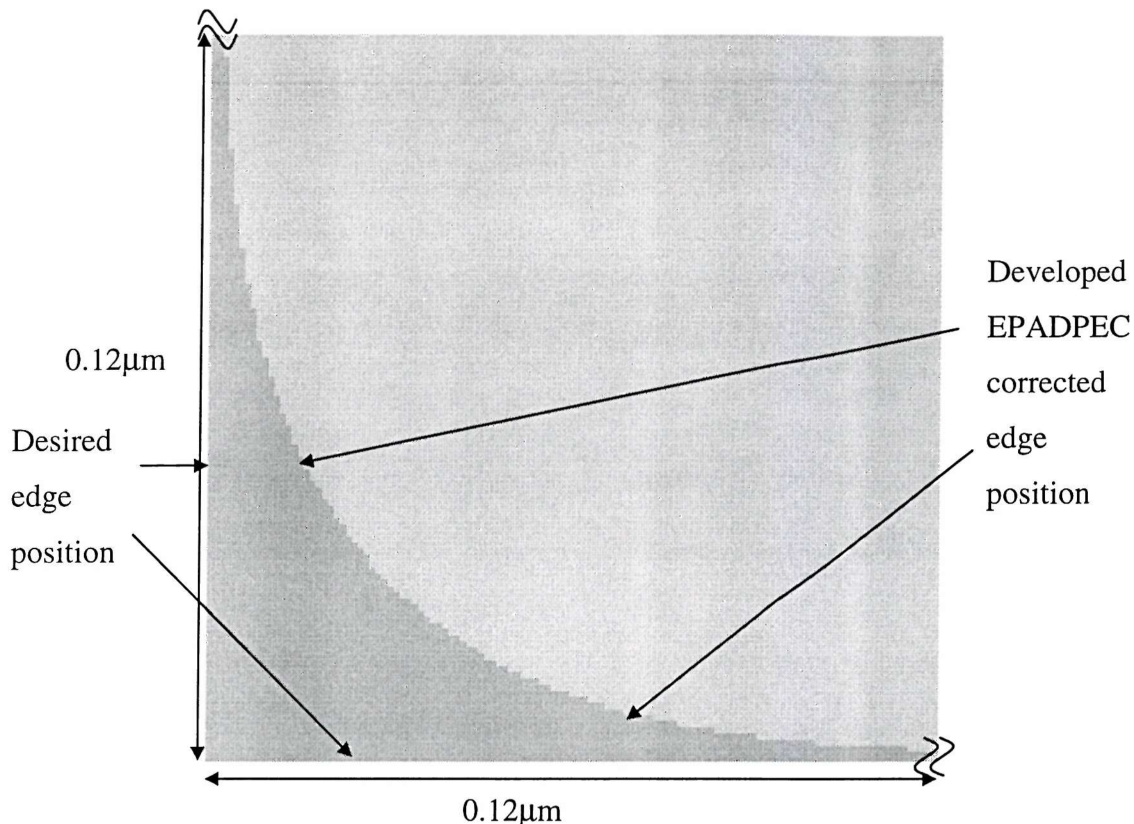
This chapter describes two further improvements to EPADPEC which have been implemented. These enhancements are discussed separately here because their impact is not directly comparable with any of the alternative "conventional" correction schemes described in the preceding chapters. The first enhancement (a corner correction algorithm) improves the shape fidelity at shape corners, at the expense of correction speed. The formulation of the algorithm is discussed, and the impact of the enhancement on the correction accuracy and speed of EPADPEC is assessed.

Section 7.2 details the second improvement that minimizes the shape count after shape partitioning. Again, a detailed description of the algorithm is followed by an assessment of the impact on the correction accuracy and speed of the overall system.

Unless stated otherwise, EPADPEC uses the continuous version and does not quantize the incident dose in the rest of this chapter.

7.1 Corner correction

Even with the EPADPEC correction, shapes still suffer from corner rounding (Figure 7-1) whose size is comparable with the forward characteristic length, α. This is intuitively reasonable: since EPADPEC corrects the electron *back* scattering, corner rounding is due mainly to the electron *forward* scattering.



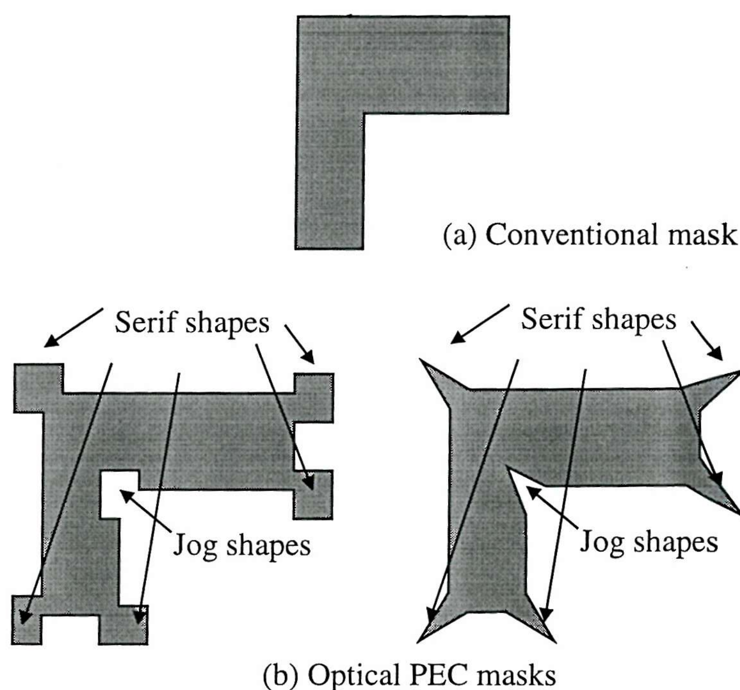
The square is corrected by EPADPEC with S = 0.5 μm and M = 2.56 μm. The simulation uses a sample grid size of 1 nm.

Figure 7-1 Corner rounding of a 12.8×12.8 μm^2 isolated square.

7.1.1 Survey of existing corner correction algorithms

An optical lithography system has a lower resolution performance than an EBL system due to the larger wavelength of light used in the optical lithography system. Thus, shape corner rounding is more prominent in optical lithography. Most of the corner correction schemes reported in the literature are devised for optical lithography and only *Lee et al.* [68] wrote briefly about a corner correction method used in PYRAMID for EBL.

An optical proximity effect mask [137] is used to suppress the corner rounding in optical lithography. In the mask, *serif* shapes are added to *outer* 90° corners and *jog* shapes are removed from *inner* 90° corners [138, 139] as shown in Figure 7-2. Ziegler *et al.* [140] uses square serifs to reduce the corner rounding on a mask in a laser lithography system. They found that there exists an optimum serif size that minimizes the corner rounding. The optimum size is independent of shape size as expected since the radius of the corner rounding in a photo-mask is mainly determined by the beam spot diameter used for printing, and the resist and chrome etching process. Without many modifications, it is possible to apply the ideas for corner rounding reduction to an EBL system if the need for corner correction arises. PYRAMID uses rectangular jogs for inner 90° corners, while serif shapes added to an isolated outer 90° corner is optional.



(a) Conventional mask

(b) Optical PEC masks

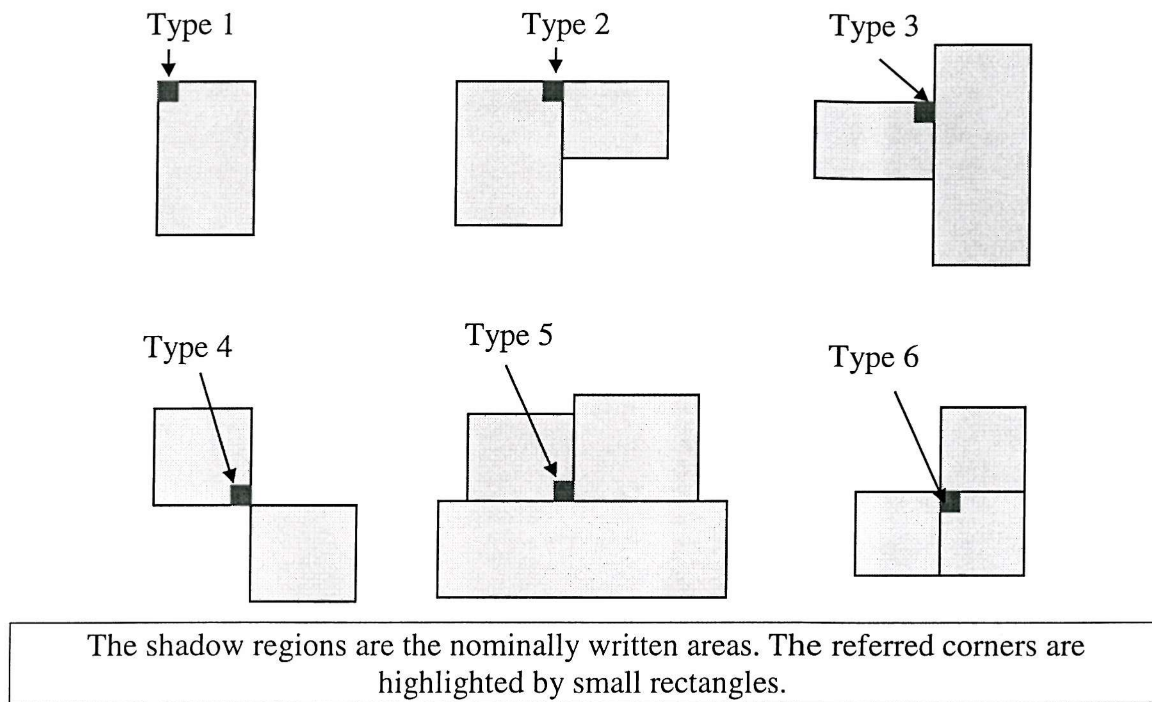
The shaded regions are optically transparent.

Figure 7-2 Mask patterns, showing serif and jog modifications.

7.1.2 Description of the corner correction algorithm

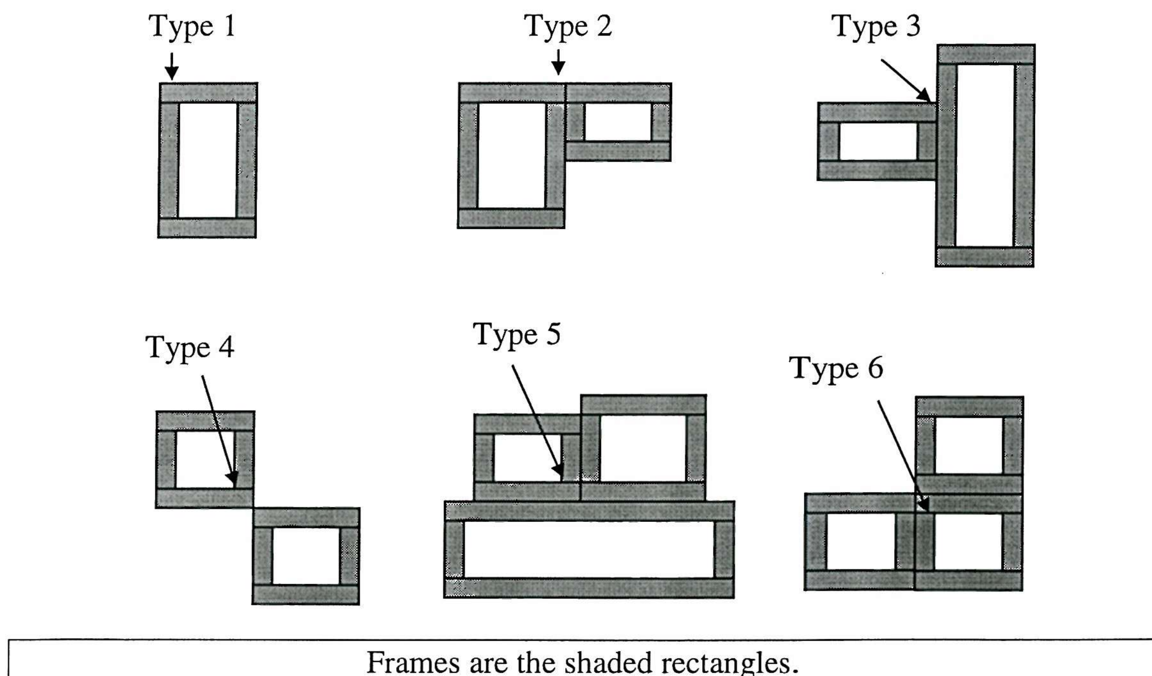
There are six distinct types of corners for Manhattan shapes (rectangles) as shown in Figure 7-3. Types 2 and 5 corners are not the overall shape corners, so they do not need any correction. Without any corner correction, EPADPEC will fracture the

shapes into frames and central rectangles as shown in Figure 7-4. Although frames along abut edges do not have any significant influence on the correction accuracy, it may reduce the throughput of an EBL machine. Thus, the framing step of EPADPEC is modified to exclude the formation of these redundant frames.



The shadow regions are the nominally written areas. The referred corners are highlighted by small rectangles.

Figure 7-3 Six distinct types of corners for rectangles.



Frames are the shaded rectangles.

Figure 7-4 Frames and central rectangles formed by EPADPEC without the corner correction algorithm.

The corner correction scheme assumes that the minimum feature size of a pattern is at least 4α, so that the forward scattering from one shape has negligible contribution to the energy density deposited on other shapes. Thus, the scheme can correct corners of the same kind in the same way.

To correct a type 1 corner, a square (corner square) is added at the corner as shown in Figure 7-5. The size, l_1 and the weight, w_1 of the incident dose and the pattern area density of the square are the correction parameters. The rectangular serif shapes (Figure 7-2(b)) are not used here because the serif shapes decrease the spacing between adjacent shapes and hence increase the influence of forward scattering from a shape to other shapes.

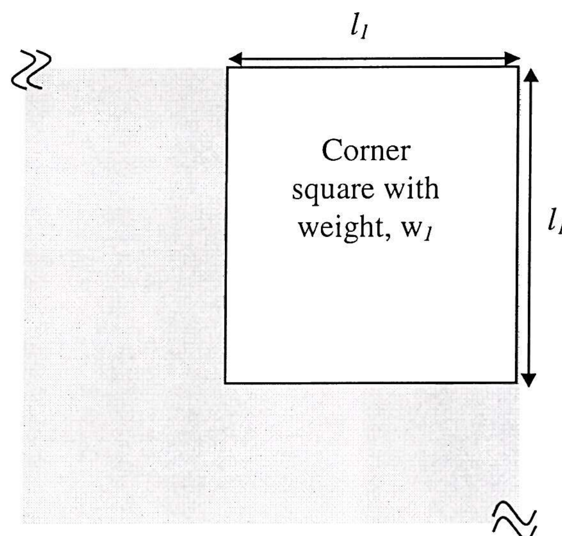


Figure 7-5 Type 1 corner correction.

As type 3 and 6 corners refer to the same kind of corner (the inner 90° corner) in the overall shape (Figure 7-3), they are considered together in the corner correction scheme. Rectangular jogs (Figure 7-2(b)) are used to correct these corners. Reducing the frame length forms the jogs shown in Figure 7-6. The reduction of the frame length, l_2 is the only correction parameter.

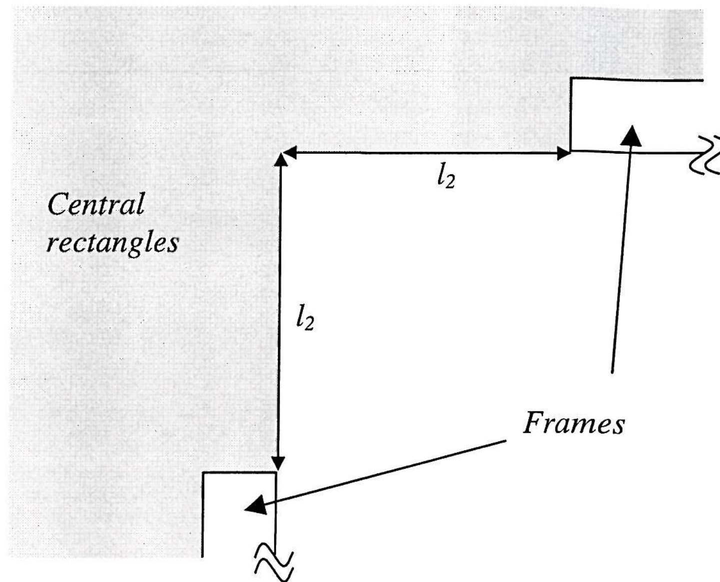


Figure 7-6 Inner 90° corner correction.

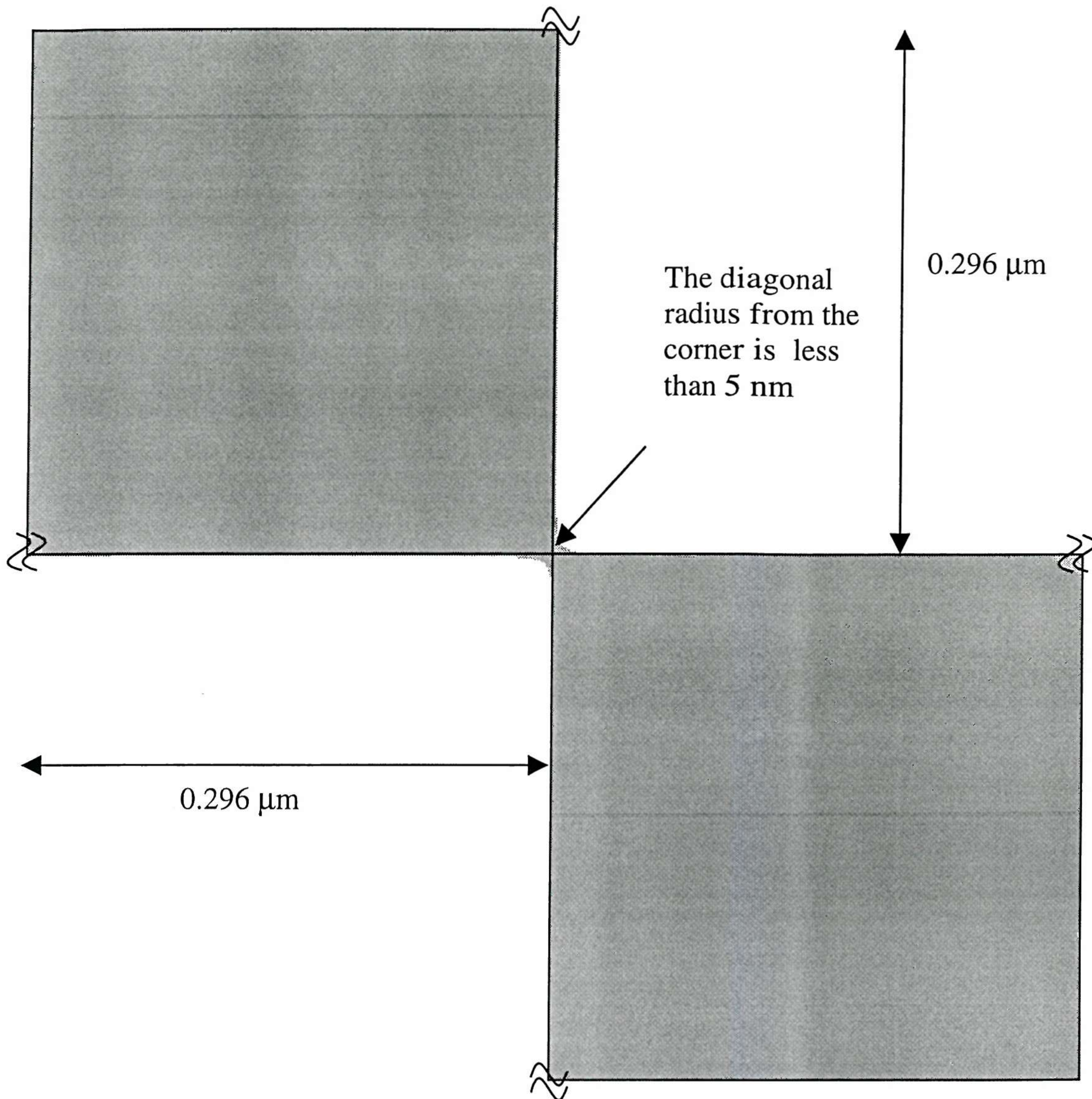
Because the energy density function is symmetrical, the deposited energy density due to forward scattering alone along the edges of type 4 and type 2 corners is the same, assuming all the shape dimensions are at least 4α. Thus, the type 4 corners do not need any correction, as the corner rounding is insignificant as shown in Figure 7-7.

7.1.3 Algorithm implementation

Before the corner correction can be performed, it is necessary to identify the type of shape corners. To incorporate the corner correction scheme into the data flow of EPADPEC (Figure 5-16), the following modifications are necessary:

- P1 carries out the corner type identification step.
- D2 has additional data members for storing the corner types and abut edge segments (needed for identification of redundant frames) of each rectangle.

P22 performs the corner correction. It removes the redundant frames and reduces the frame lengths for type 3 corners. It also inserts corner squares for type 1 corners and sets their D_I values equal to w_I. For rectangles whose dimension is smaller than twice the frame width (4α), half their smallest dimension is used as the frame width instead of treating the entire rectangles as frames.



The pattern is corrected by EPADPEC with S=M=2.56 μm. The simulation uses a sample grid size of 1 nm.

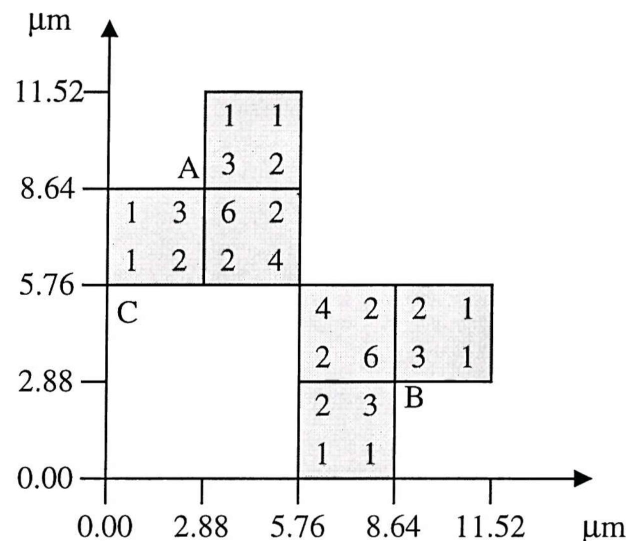
Figure 7-7 The corner rounding at a type 4 corner of two 2.56 × 2.56 μm^2 isolated squares.

7.1.4 Estimation of the optimum correction parameter values

To estimate the optimum values of the correction parameters, l_1, w_1 and l_2, the absolute area difference between the simulated and the desired developed resist images of a test pattern is computed for various values of the parameters. The optimum values are the values that minimize the absolute area difference. To avoid the distortion of the optimum values by the *intrinsic* errors (section 5.1) of the

EPADPEC method, EPADPEC uses $M = S = 0.1\beta$ and physical width of 10.7β covered by $\boldsymbol{A}$, which minimises the errors (Figure 6-8(a)).

The test patterns introduced in Chapter 6 are not suitable for the determination of the optimum parameter values; they either do not contain any type 3 corners or are too large and complex. Thus, a new simple test pattern (Figure 7-8) is used. The pattern contains all the distinct corner types except type 5 corners which is not a problem with EPADPEC. The minimum feature size of 2.88 µm = 0.3β (α=0.074µm) is large enough that the forward scattering from the additional corner square at a shape corner has negligible contribution to the energy density deposited at other shape corners.



The shadow regions are the nominally written areas and the numbers at shape corners indicate the corner types.

Figure 7-8 Test pattern for evaluating the optimum parameter values of the corner correction scheme.

Because EPADPEC compensates for any changes in back scattering due to the corner correction scheme, so any changes to the degree of corner rounding are mainly due to changes of forward scattering. Since the forward-scattered electrons have an effective range of about 4α (section 5.2.1.2) and the maximum value of l_1 and l_2 is limited to α, so the simulation windows shown in Figure 7-9 are sufficient to monitor the changes. When determining the optimum parameter values of the corner correction method, the absolute area differences within the simulation windows are used and a sample grid size of 1nm is used in the simulation.

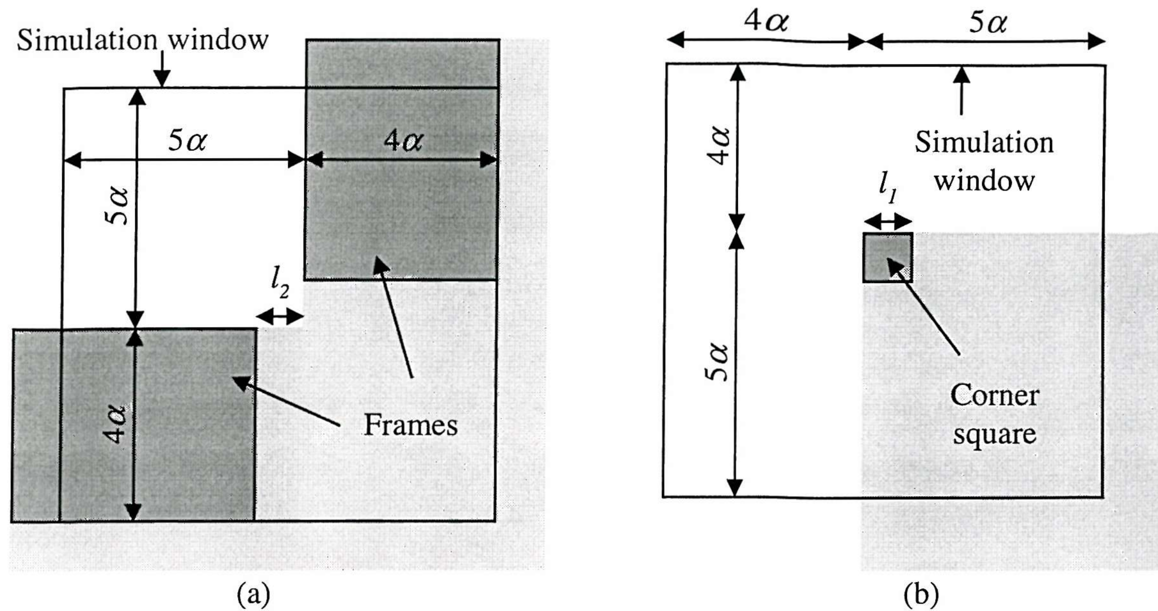


Figure 7-9 Simulation windows at (a) inner 90° and (b) outer 90° corners.

Figure 7-10 shows the variation of the improvement factor on the absolute area difference of the two inner 90° corners (Corners A and B in Figure 7-8) with l_2. The improvement factor is the ratio of the average absolute area difference at the two corners with the corner correction scheme to the corresponding values without the corner correction method. The improvement factor is maximum (4.19) when the value of l_2 is about 0.3α which is taken as the optimum value of l_2. For $l_2 \geq 0.6\alpha$, the corner correction reduces the correction accuracy.

For the eight type 1 corners in the test pattern, Figure 7-11 shows the variation of the improvement factor on the absolute area difference with l_1 and w_1. The optimum values of l_1 and w_1 occur at 0.3α and 10 respectively when the improvement factor is maximum (2.46). When both the parameter values are large, the corner correction reduces the correction accuracy.

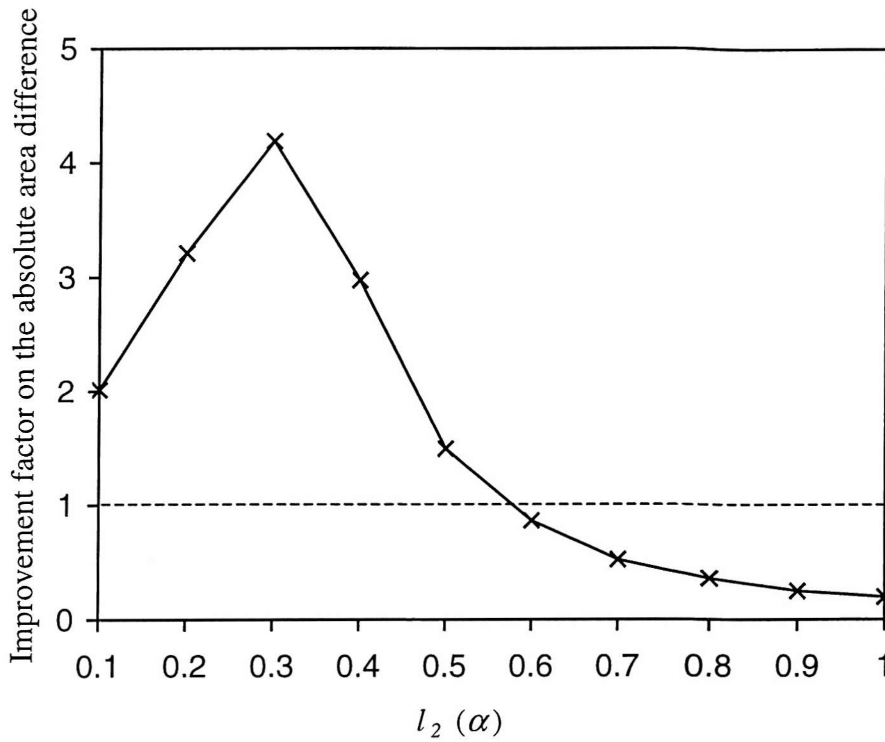


Figure 7-10 Variation of the improvement factor on the absolute area difference at corners A and B (Figure 7-8) with l_2.

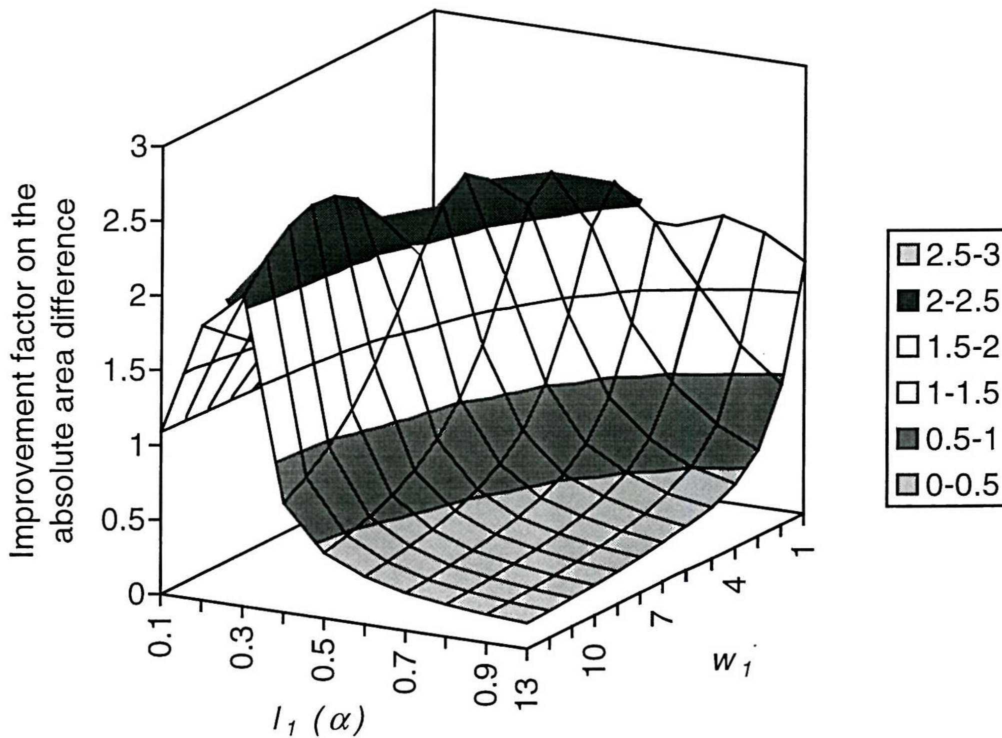
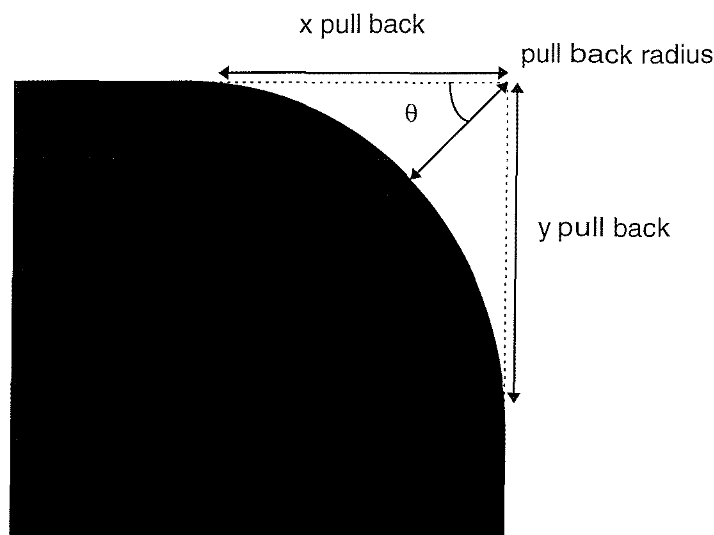


Figure 7-11 Variation of the improvement factor on the absolute area difference at the eight type 1 corners (Figure 7-8) with l_1 and w_1.

7.1.5 Computation overhead and correction accuracy improvement of the corner correction scheme

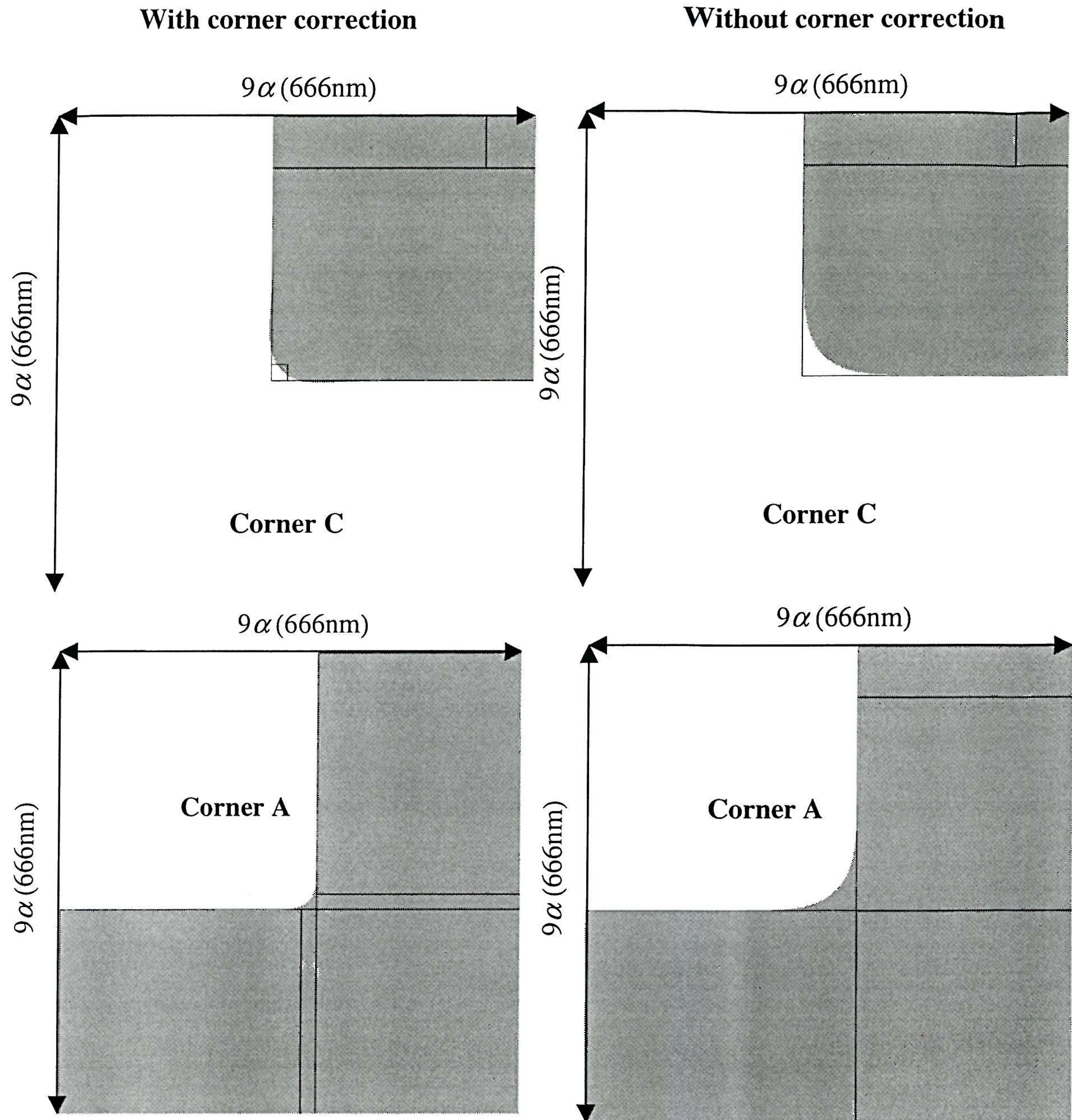
Tran *et al.* [141] introduces five types of figure of merit for corner rounding as shown in Figure 7-12. Using their image processing analysis software, they found that only the pull back radius, θ and the absolute area difference could be measured accurately from scanning electron microscope images of the fabricated mask patterns. Thus, these three figures of merit will be used to access the gain in correction accuracy by the corner correction method.



The pull back radius is the minimum distance between the perfect corner and the rounded corner. The fifth figure of merit is the absolute area difference used in Chapter 6 and section 7.1.4.

Figure 7-12 Shape corner fidelity metrics.

Figure 7-13 shows the simulated developed resist images (shaded regions) at corners A and C of the test pattern (Figure 7-8) corrected by EPADPEC with and without the corner correction method. The corresponding values of the figures of merit are listed in Table 7-1. The figures and the values of the figures of merit clearly demonstrate the reduction of corner rounding by the correction scheme.



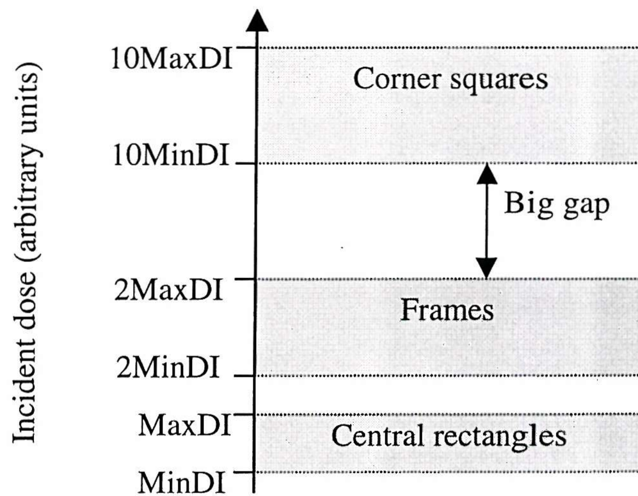
The corner correction scheme uses the optimum parameter values estimated in section 7.1.4. The EPADPEC and the simulation settings are stated in section 7.1.4.

Figure 7-13 Simulated developed resist images (shaded regions) of the test pattern (Figure 7-8) corrected by EPADPEC.

		Pull back radius (nm)	θ (°)	Absolute area difference (nm^2)
Corner A	With corner correction	17.0	45	570
	Without corner correction	38.9	46	2388
Corner C	With corner correction	20.5	47	1244
	Without corner correction	41.0	45	3035

Table 7-1 Values of the figures of merit at corners A and C of the test pattern (Figure 7-8) corrected by EPADPEC with and without the corner correction method.

Because the optimum value of w_1 is 10, so there is a big gap between the incident dose ranges of corner squares and frames as shown in Figure 7-14. Quantizing the incident dose range (*MinDI* to *10×MaxDI*) using the uniform upper-level quantization strategy is inefficient as the quantization step is larger than necessary. This large

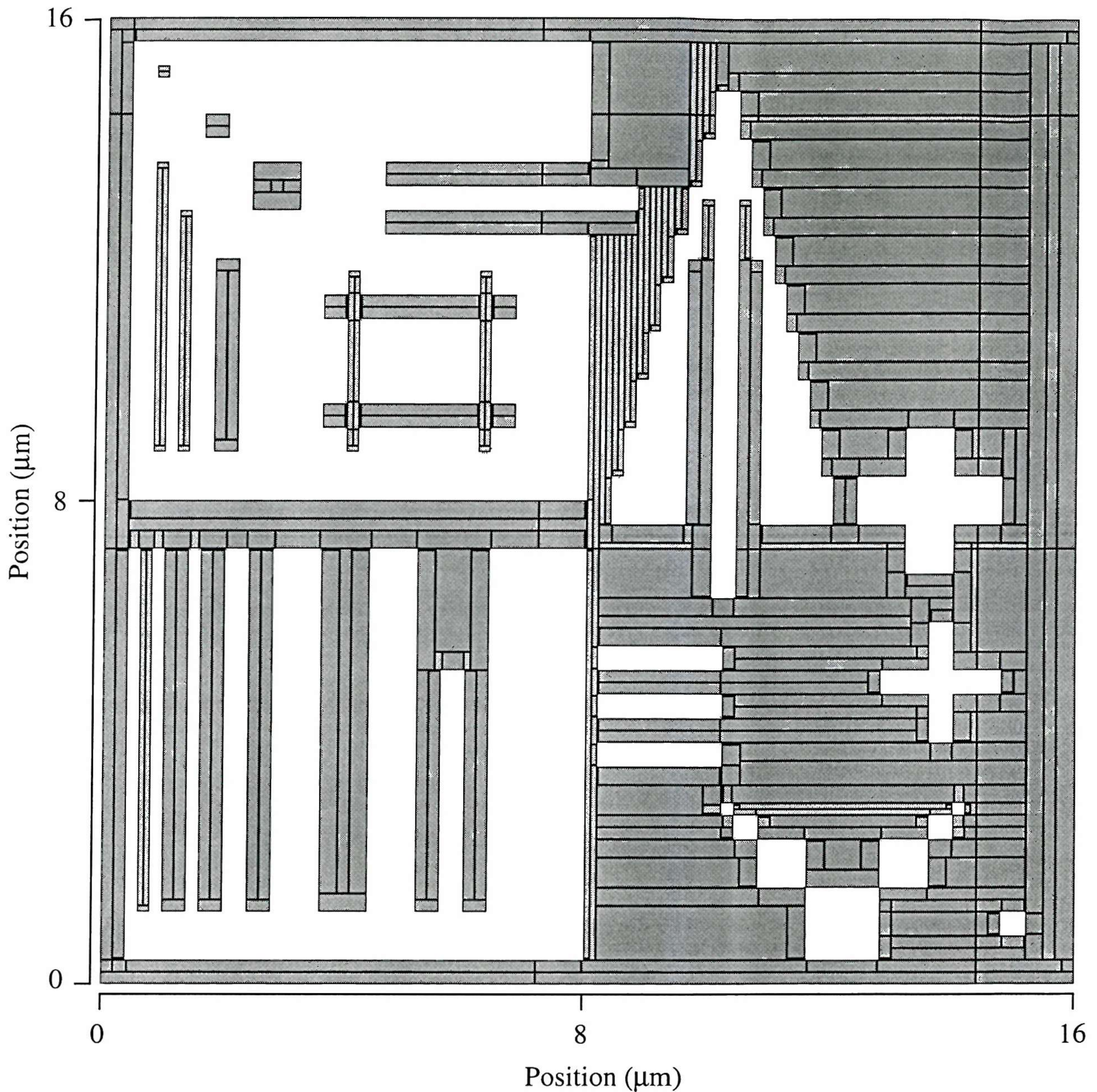


MinDI and *MaxDI* are the minimum and maximum values of the D_I map respectively.

Figure 7-14 The incident dose ranges of central rectangles, frames and corner squares.

quantization step degrades the correction accuracy. To overcome this problem, the number of quantization levels is split evenly among the three incident dose bands. Each band is quantized using the uniform upper-level quantization strategy. If the number of quantization levels is not a multiple of 3, then the extra levels are given to

the lower bands. When the corner correction with the new quantization strategy is applied to correct RauTP, the absolute area difference is reduced from 793 to 363 sample points $(25\text{nm})^2$ (Figure 7-15). Using the old quantization strategy, the absolute area difference is increased from 793 to 8189 sample points $(25\text{nm})^2$.

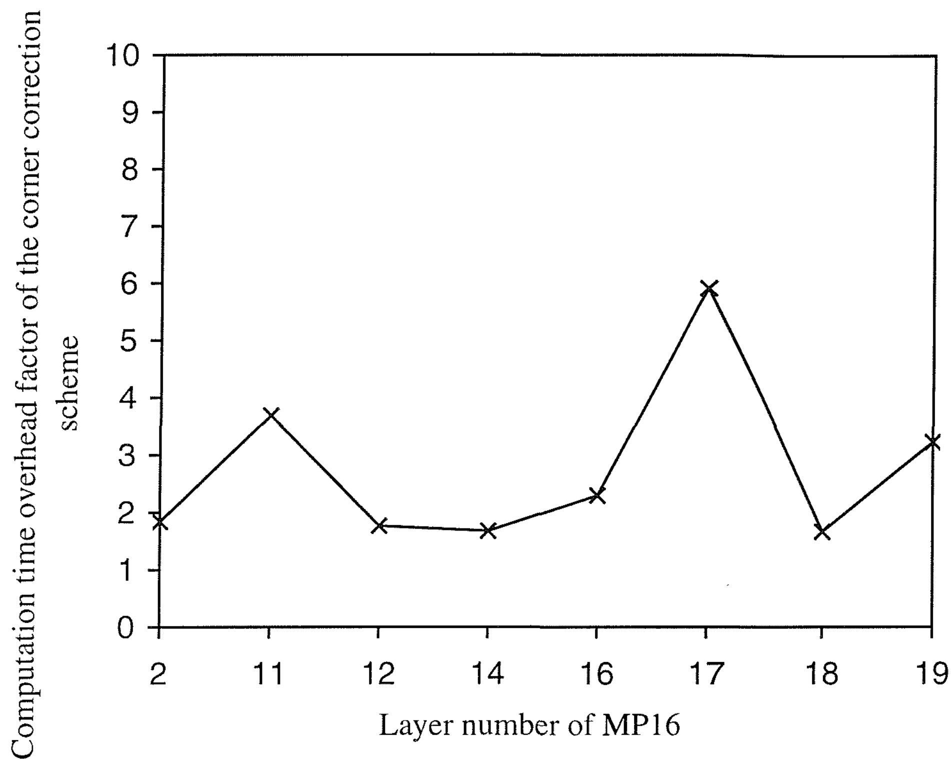


The corner correction scheme uses the optimum parameter values, $l_1=l_2=25\text{nm}\approx0.34\alpha$ and $w_1=10$.

Figure 7-15 The simulated developed resist image (shaded regions) of RauTP corrected by EPADPEC with the corner correction.

As the corner correction algorithm affects the computation time of shape overlap removal (part of the shape processor (P1) in Figure 5-16) and t_{PEC}, so their sum is

used for calculating the computation overhead of the algorithm. The algorithm increases the sum by factors ranging from 1.67 to 5.91 for various layers of MP16 (Figure 7-16).



The corner correction scheme uses the improved quantization strategy, l_1=l_2=100nm≈1.35α and w_1=10.

Figure 7-16 The computation time overhead factor of the corner correction method for various layers of MP16.

7.1.5.1 Conclusions

Although the results in the previous section have clearly demonstrated the reduction of corner rounding by the corner correction scheme, there is an increase in the computation time.

7.2 Shape fracturing

For a large region with a uniform value of λ, the optimum incident dose for shapes around the centre of the region are the same, so it is unnecessary to fracture those shapes into smaller sub-shapes along a sub-field grid. Thus, the shape partitioning algorithm using a fixed size sub-field grid (the sub-field fracturing algorithm (SFA)) is not optimum. As partitioning shapes into sub-shapes will inevitably decrease the throughput of the EBL machine, there is clearly a need for a shape-fracturing algorithm that minimises the number of sub-shapes generated.

7.2.1 Survey of existing shape fracturing algorithms

The following is a summary of the shape partitioning strategy of Parikh et al. [142-144]:

1. A given pattern is corrected by a PEC scheme.
2. Various sample points throughout the pattern are used to assess the quality of the corrected pattern.
3. If the corrected pattern quality falls below a certain *goodness* criteria at certain sample points, the associated region of the point is subdivided.
4. The partitioned pattern is re-corrected by the PEC scheme until the quality criteria has been satisfied or no further shape fracturing is possible based on the physical limitation of the EBL machine, e.g. the exel size.

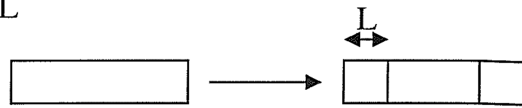
This strategy is potentially very time consuming as it might require several correction iterations before the quality criteria are met.

Figure 7-17 illustrates the shape partitioning scheme of Kratschmer[145]. The inter-shape fracturing is more time consuming than the intra-shape fracturing.

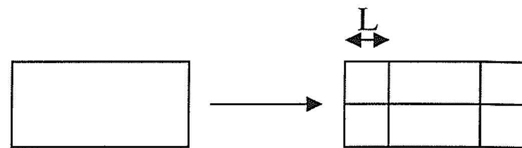
$3\alpha < L < \beta$ and shape dimensions: W by H

Case 1: $W \leq L$ and $H \leq L$ → No fracturing

Case 2: $W \approx L$ or $H \approx L$

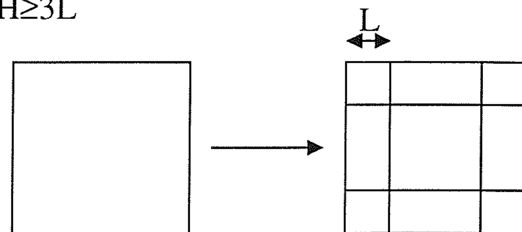


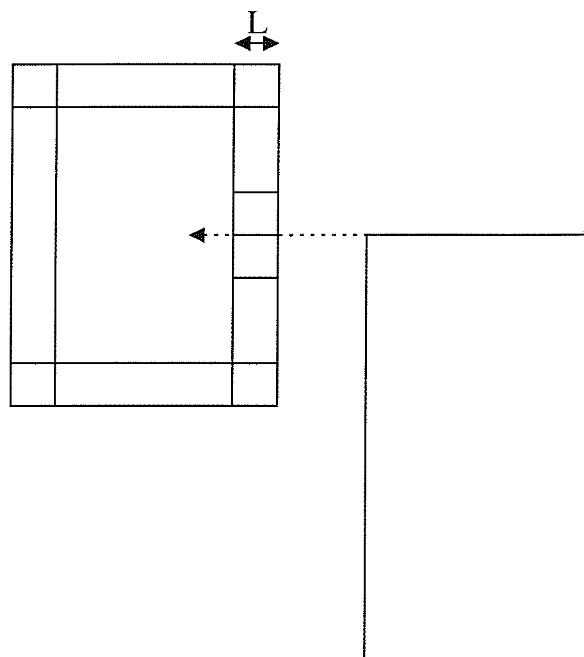
Case 3: $W \approx 2L$ or $H \approx 2L$



Intrashape fracturing

Case 4: $W \geq 3L$ and $H \geq 3L$





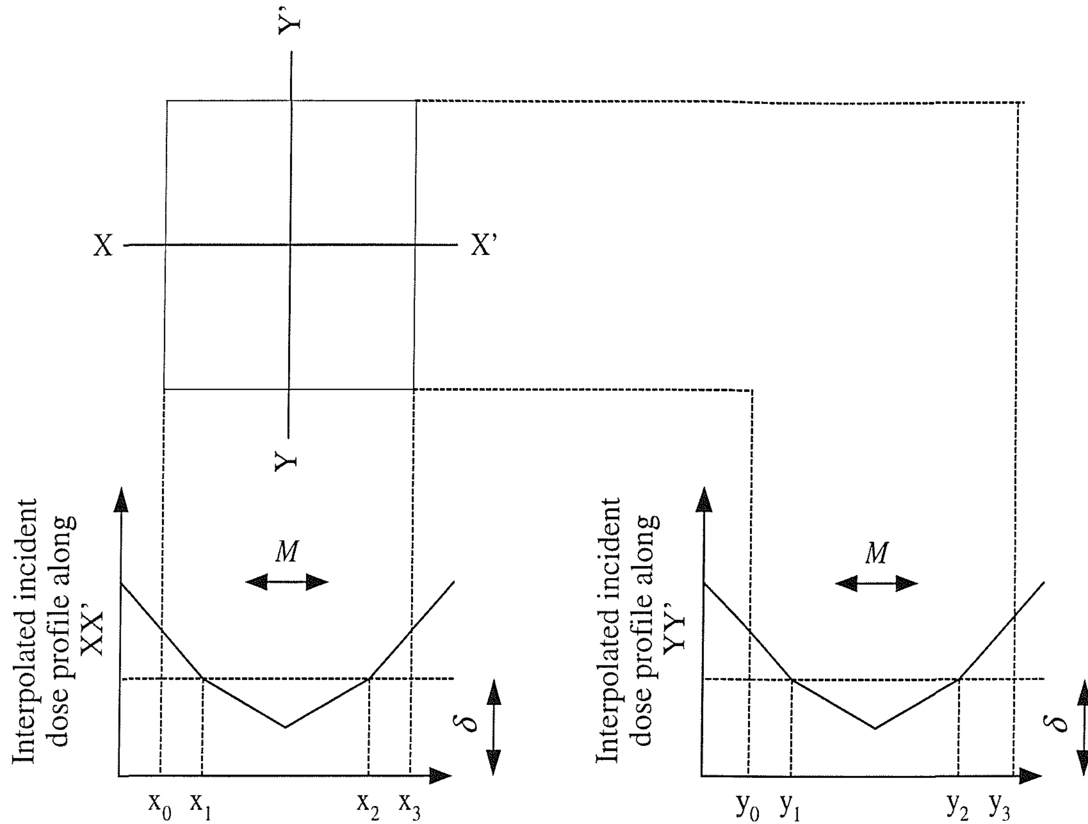
Intershape fracturing

Figure 7-17 Kratschmer's shape partitioning scheme.

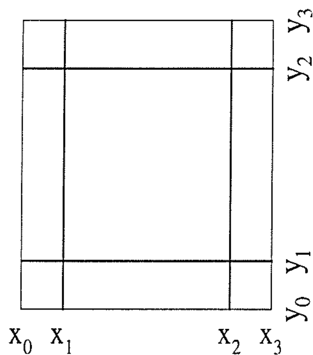
CAPROX supports two shape partitioning algorithms: Manhattan and physical schemes. The former method is the same as Kratschmer's method. On the other hand, the physical fracturing algorithm partitions a shape along the contours (isodose lines) of the incident dose distribution. When the step size between adjacent isodose lines is equal to the step size of a uniform quantization scheme, this algorithm provides the optimum fracturing method in the sense that it minimizes the quantization error with the minimum number of sub-shapes. To maintain the correction accuracy, the process grid used in CAPROX needs to be at least as small as the minimum feature size. This fracturing method can thus be very time consuming. It is possible to implement the physical fracturing scheme in EPADPEC without the above drawback, because the mesh size is much larger than the minimum feature size.

7.2.2 Description and implementation of the physical fracturing algorithm (PFA)

Figure 7-18 illustrates the procedures of PFA. Interpolating the D_I map linearly, the algorithm first computes the one-dimensional incident dose profiles along the two lines that pass the centre of a rectangle along the x and y axis. Next, it determines the intersecting positions between the profiles and the required isodose lines. Finally, it partitions the rectangle along these intersecting positions as shown in Figure 7-18. This is an approximation to the ideal shape partitioning along isodose lines as the algorithm assumes the one-dimensional incident dose profile at any lines along the same axis in a rectangle is the same, which might not be true in practice.



Step 1: Calculate the positions in which the required isodose lines intersect with the interpolated incident dose profile of the two lines, which pass the rectangle centre along the x and y axis. δ is the incident dose step size between two adjacent isodose lines.



Step 2: Fractures the rectangle along the intersecting positions computed in step 1.

Figure 7-18 Procedures of PFA.

PFA is carried out by process P7 in the data flow diagram of EPADPEC (Figure 5-16). P22 (the process that performs the sub-field partitioning scheme) is omitted. The required isodose line count, the minimum incident dose step size between two adjacent isodose lines and the minimum allowable fractured size are specified by the

user. The last two parameters correspond to physical limitations of the EBL machine, such as the incident dose resolution and the exel size. When the calculated dose step size is smaller than the specified size, the given isodose lines count is modified in such a way that the new dose step size is the smallest value that is greater or equal to the minimum step size.

7.2.3 Speed and correction accuracy comparison between PFA and the sub-field fracturing algorithm (SFA)

KasugaTP with fully filled dummy shapes is used for comparing the performance between PFA and SFA. For all the results presented in this section, EPADPEC uses a mesh size of 2.56 μm ($\approx 0.267\beta$) and A with an order of 17, corresponding to a physical coverage width of about 4.53β. These values are sufficient to suppress the maximum magnitude of the monitor line width error to below 10% (section 6.2.2.1). Figure 7-19 shows the variations of the maximum line width error of the monitor shape with the parameters of both algorithms. For a fixed value of the isodose line count, the maximum line width error decreases as the minimum fractured size decreases. Similarly, the maximum line width error decreases and eventually saturates as the isodose line count increases when the minimum fractured size is kept constant. For the minimum fractured size$<S$ (the sub-field size used in SFA) and the isodose line count is at least 32, PFA produces a smaller or equal maximum line width error than SFA. To achieve the saturated minimum maximum line width error (2nm), the isodose line count ≥ 256 and the minimum fractured rectangle size $\leq 0.1\beta$ are needed in PFA, whereas, $S \leq 0.1\beta$ is needed for SFA.

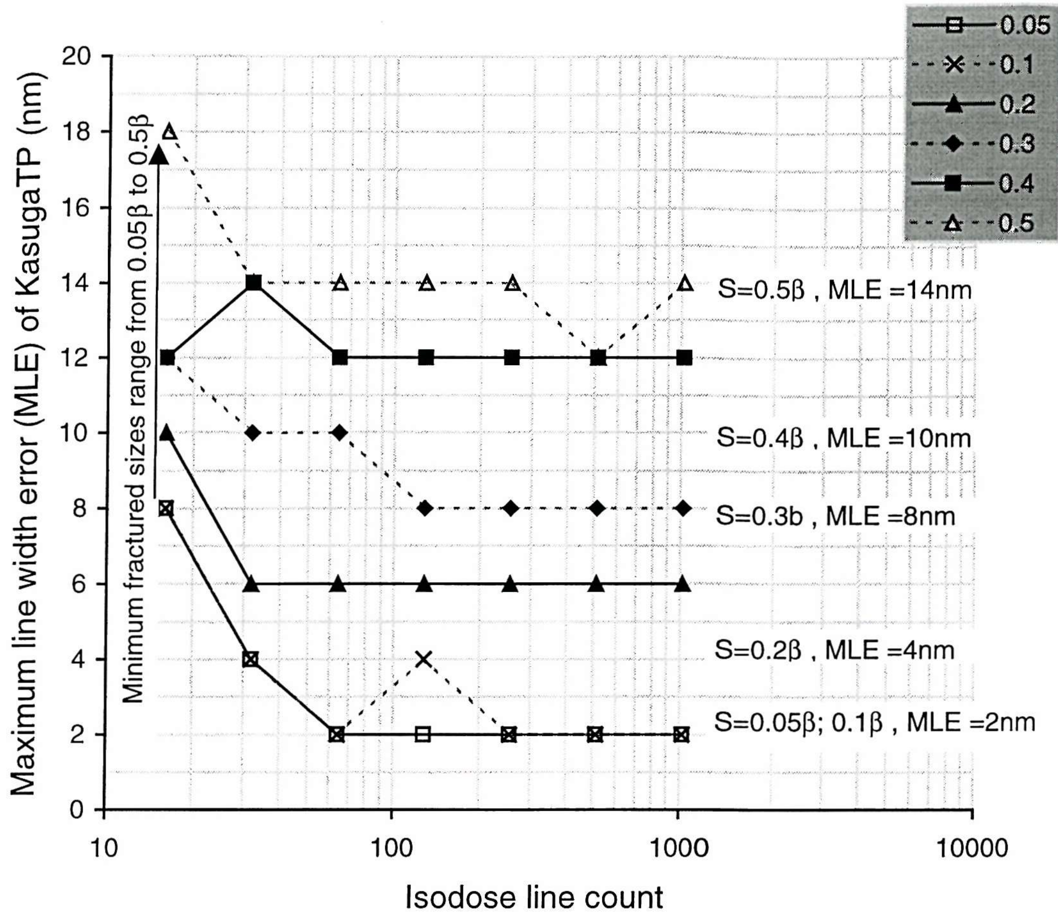
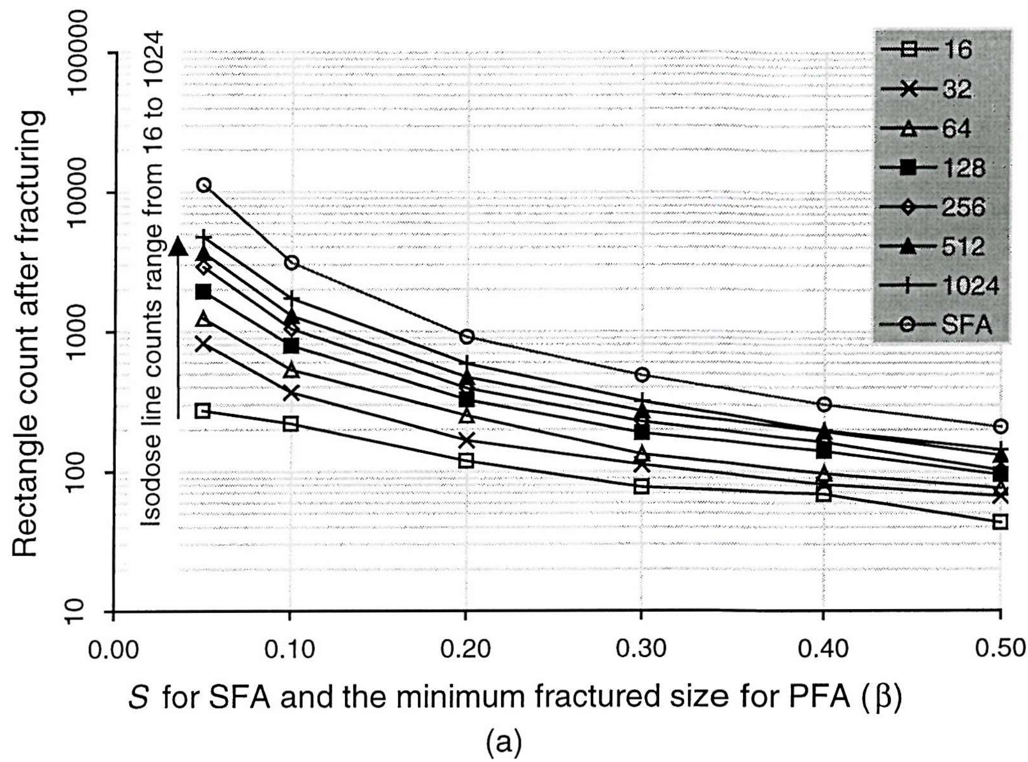
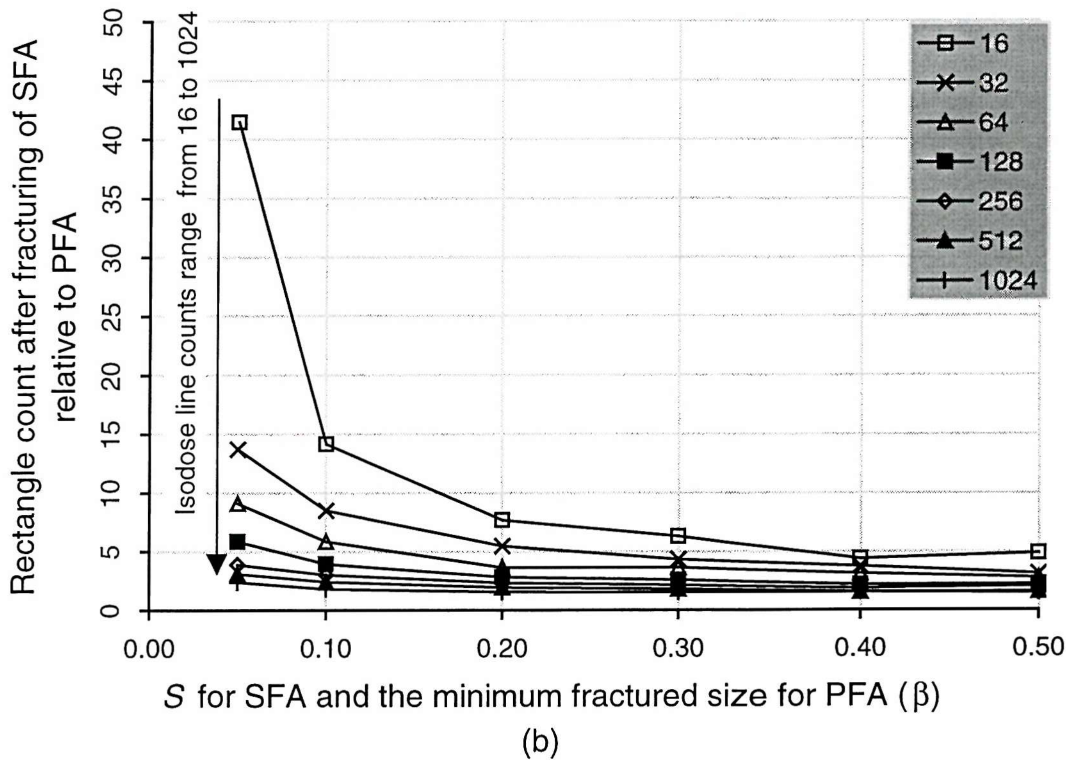


Figure 7-19 Variation of the maximum line width error of KasugaTP with *S* of SFA, and the isodose line count and the minimum fractured size of PFA.

Figure 7-20 shows the variation of the rectangle count after shape fracturing with the parameters of both algorithms. For S equal to the minimum fractured size and all the values of the isodose line count in Figure 7-20(a), PFA produces less rectangle counts than SFA. For *S* equals to the minimum fractured size, the relative rectangle counts of SFA to PFA increase as *S* decreases (Figure 7-20(b)).



(a)



(b)

Figure 7-20 Variation of the rectangle count after shape fracturing with *S* of SFA and the isodose line count and the minimum fractured size of PFA.

For the case where the minimum fractured size equals S, PFA is increasing faster than SFA as S decreases (Figure 7-21). This is due to the higher reduction factor on rectangle count after fracturing by PFA over SFA at smaller S values (Figure 7-20(b)).

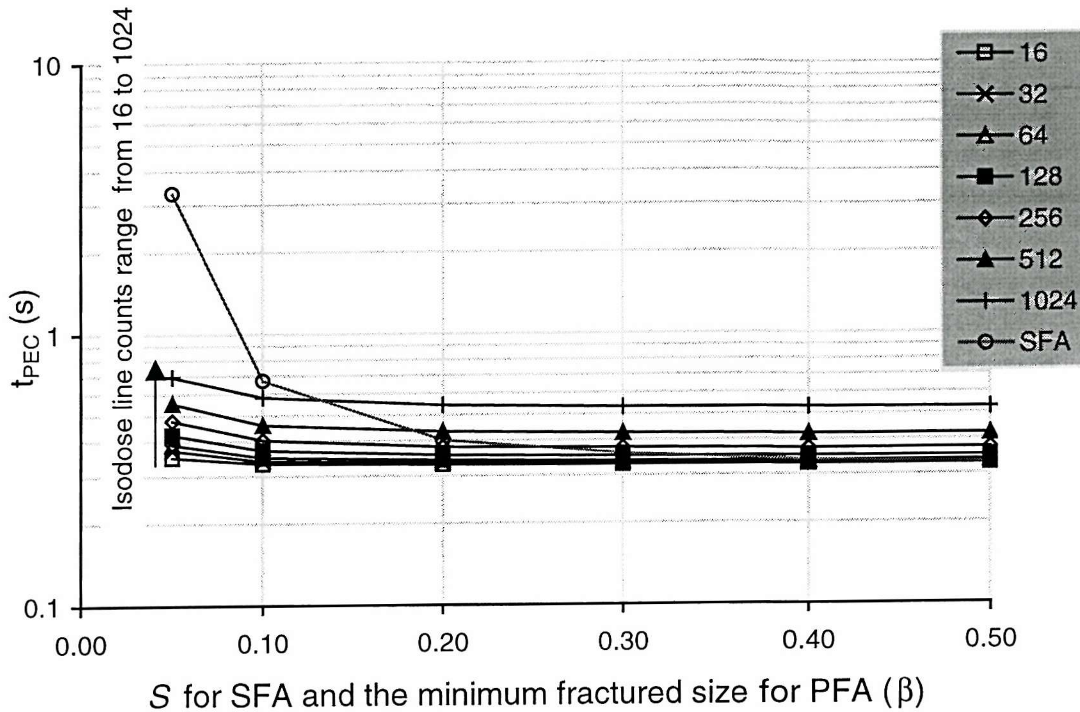


Figure 7-21 Variation of t_{PEC} with the sub-field size of SFA and the isodose line count and the minimum fractured size of PFA.

Figure 7-22 shows the variation of the relative speed and rectangle count between SFA and PFA with the average shape size for the minimum fractured rectangle equal to S = 0.1β and the isodose line count equal to 256. The parameter values are the minimum values needed for achieving the minimum saturated maximum line width as determined from Figure 7-19. Different average shape size values are obtained by changing the total number of shapes making up the dummy shapes while keeping the area occupied by the shapes constant. From the graphs, PFA is faster than SFA if the average shape area is greater than about 100μm^2, which is much bigger for the practical layout of advanced devices. The relative rectangle count does not increase smoothly with the average shape size as its values also depend on the shape layout in the original pattern with respect to the incident dose distribution of the pattern.

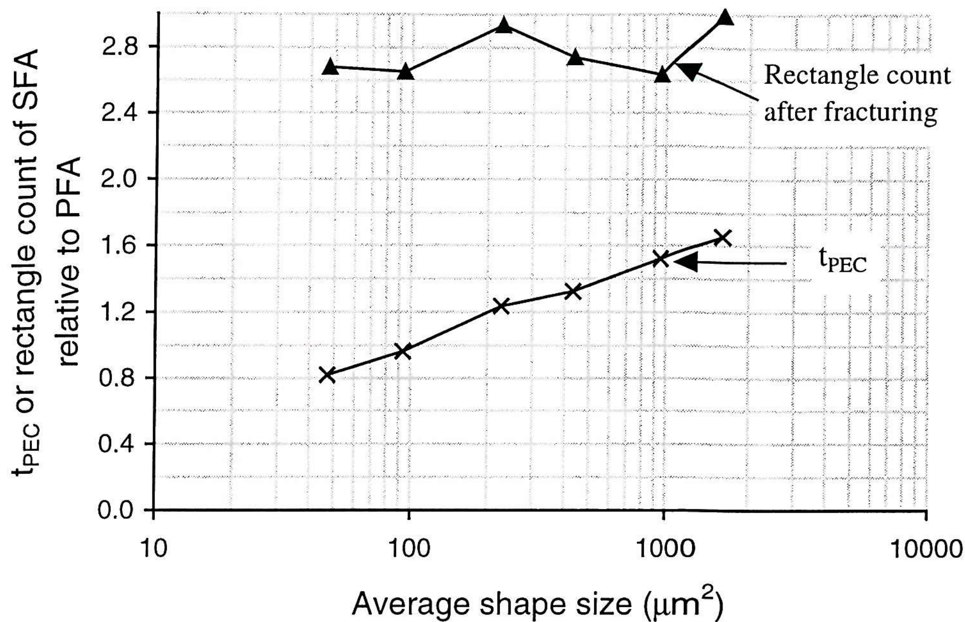


Figure 7-22 The relative speed and rectangle count between SFA and PFA for various average shape sizes.

7.2.4 Conclusions

PFA is an approximation to the ideal shape partitioning along isodose lines as the algorithm assumes the one-dimensional incident dose profile at any lines along the same axis in a rectangle is the same, which might not be true in practice. In theory, PFA should give more accurate correction result than SFA. However, because of the use of the five point interpolation strategy (section 5.2.1.3) in EPADPEC, larger rectangles cause larger error. Thus, the correction accuracy of PFA is degraded as it produces larger rectangles than SFA does (This effect is quantified in Figure 7-19).

PFA is computationally more expensive than SFA. However, PFA produces fewer rectangles (for S=the minimum fractured size), which in turn reduces the amount of computation time needed for interpolating the incident dose from the D_I map. Thus, in absolute terms, PFA can be faster than SFA without sacrificing the correction accuracy if the average shape size is large enough ($\geq 100\mu m^2$). In conclusion, PFA is superior to SFA if the final output file size after PEC is critical to an electron beam data processing system throughput and/or the average shape size is atypically large in a circuit pattern.

Chapter 8

Feasibility study of performing EPADPEC in real time

To maximize the throughput of an EBL system, each component in the entire system obviously has to operate as fast as possible. From the perspective of this research, if the speed of the correction system can be increased to the point that it is not rate-limiting, then (a) the system can be in-lined into the EBL machine itself, and (b) PEC has effectively disappeared from all throughput calculations.

This chapter first identifies and quantifies all the time-consuming processes occurring in the process of exposing a wafer; and establishes the qualitative nature of the problem. Observations indicate that a large, but feasible, increase in processing speed will accelerate the correction process to the desired speed.

Section 8.2 details the timing components in the data flow of EPADPEC. The timing values on MP16 are used to estimate the correction times on future devices. Based on the projected values, it is possible to identify the bottlenecks in the data flow and calculate the required speedup factors on the various components of EPADPEC for real-time correction processing. Having describing ways to overcome the bottlenecks, the chapter concludes by identifying the best way to achieve real time EPADPEC correction.

Unless stated otherwise, it is assumed that a Gaussian spot, step and write EBL machine writes shapes with the minimum normalized incident dose using the maximum clock frequency (10 MHz for Leica EBMF 10.5 EBL machine) in this chapter. All the values of the timing components stated in this chapter are measured or estimated times on a Gateway 2000 G6-180 system with a Pentium Pro 180 MHz microprocessor.

8.1 Bottlenecks in the data flow of the existing EBL system

Figure 8-1 shows the existing data flow diagram of the EBL system in the Department of Electronics and Computer Science, University of Southampton. $t_{MachineSetup}$ is the overhead time for the machine set up, calibration and self-testing, and has a constant value regardless of the size of the exposure pattern. $t_{FileTransfer}$ and $t_{FileLoading}$ are directly proportional to the file size of the exposure pattern in the EBF binary format. t_{PEC} is the computational time of PEC while t_{Other} is the computational time of other data processing on the DEC 3000 M600 workstation. t_{Write} is the time taken by the electron beam hardware to write the exposure pattern.

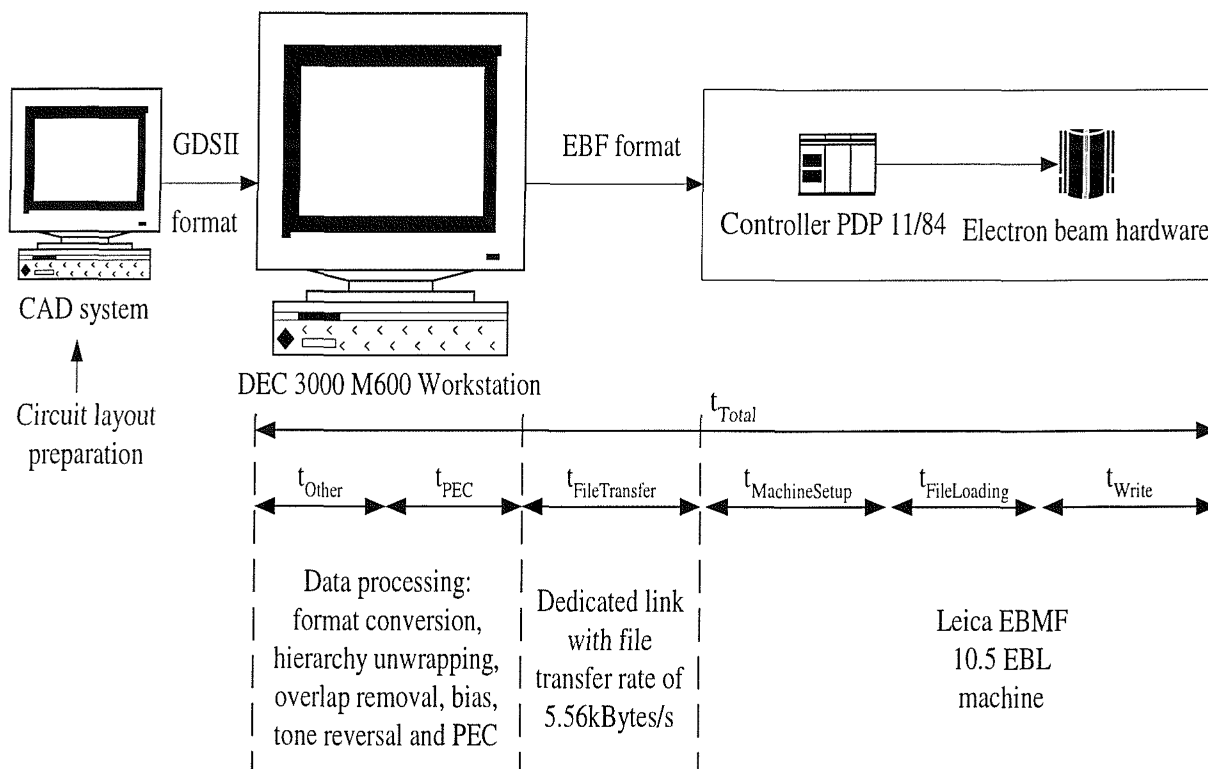


Figure 8-1 Existing data flow diagram of the EBL system.

Figure 8-2 shows the estimated computation times of various components in the EBL system on layer 11 (polysilicon gate layer) of MP16. The same estimation method in section 6.3.1 is used to translate t_{Other} and t_{PEC} of LPEC on M600 to G6-180. The values of t_{PEC} are the same as those values stated in Figure 6-23(a), in other words LPEC and EPADPEC use the same PEC settings and exel size as in Figure 6-23(a). It is assumed that each shape in the EBF binary format needs 8 bytes of storage space and the EBL systems with EPADPEC and LPEC has the same value of t_{Other}.

Appendix B.4 details the calculations of $t_{MachineSetup}$, $t_{FileLoading}$ and t_{Write}. From the chart in Figure 8-2, it is clear that t_{PEC} is the bottleneck in the data flow of the EBL system with LPEC. On the other hand, EPADPEC does not rate limit the throughput of the EBL system. However, its shape fracturing algorithms (the sub-field fracturing and framing algorithms) produce more sub-shapes, resulting in an increase of the EBF file size by a factor of about 8. This increase in file size causes $t_{FileTransfer}$ and $t_{FileLoading}$ to multiply by the same factor, resulting them to become the bottlenecks in the data flow. In-lining the correction process would remove the need for the intermediate (EBF binary) data files of many GByte by allowing the much smaller pattern data file (in GDSII format) to be input directly to the EBL machine (Figure 8-3). In order to perform the data processing, it is clearly necessary to incorporate sufficient computing capability into the machine itself. Since the new hardware and the controller (Figure 8-3) can operate in parallel, so the EBL machine can perform the data processing and the machine set up at the same time, reducing t_{Total} further.

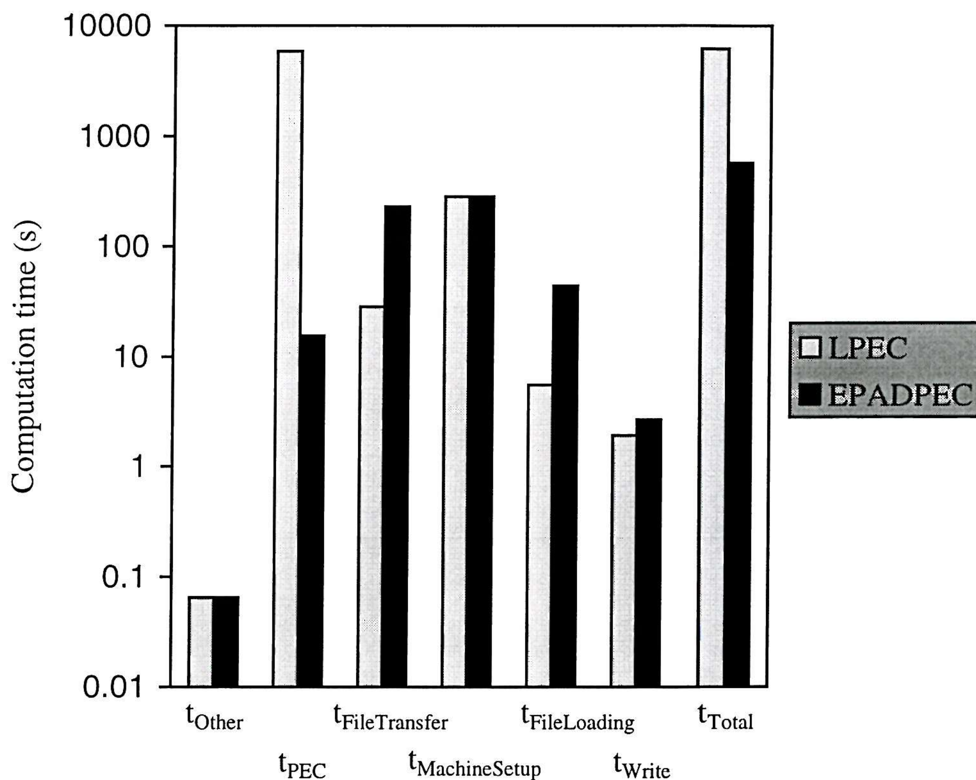


Figure 8-2 Estimated computation times of various components in the EBL system on layer 11 (polysilicon gate layer) of MP16.

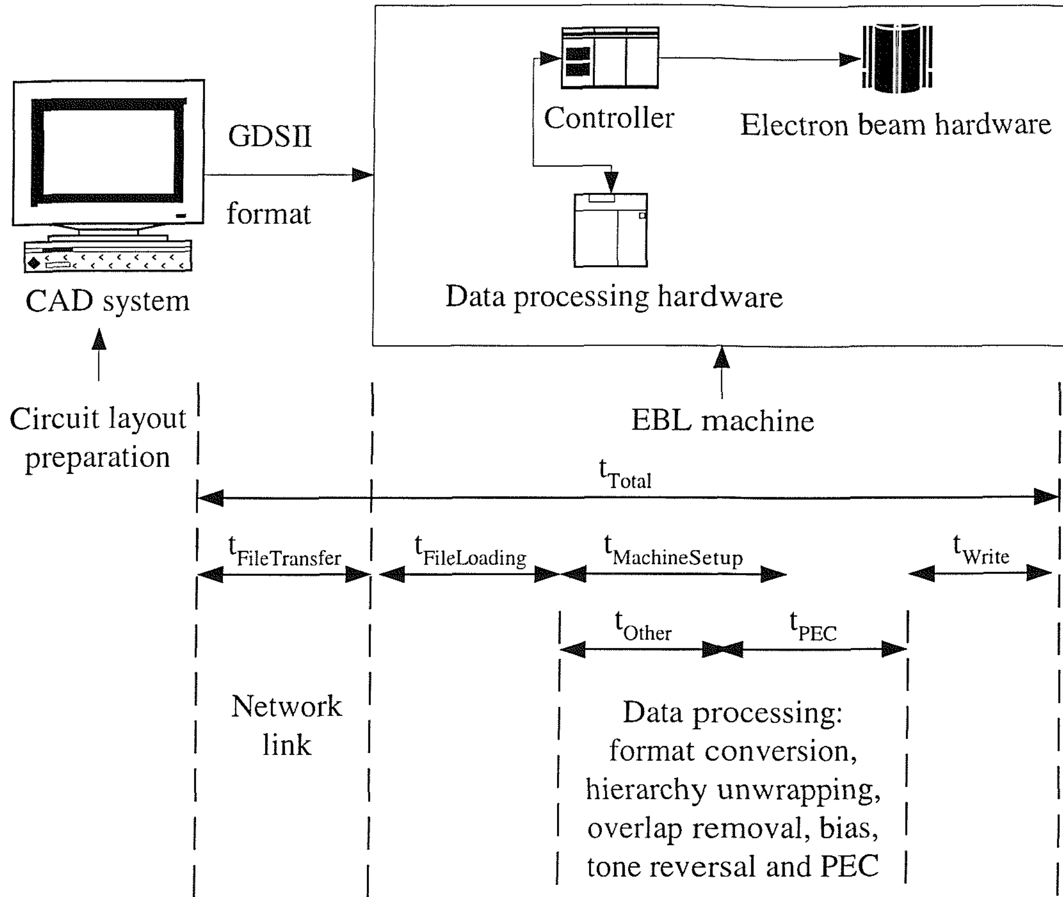


Figure 8-3 Data flow of the proposed EBL system.

8.2 Computational time requirement projection of EPADPEC for real time correction on future semiconductor devices using future EBL machines

Figure 8-4 shows the data flow diagram of the data processing hardware in the proposed EBL system. To realize real time PEC, the computational time of EPADPEC (P2), $t_{EPADPEC}$ must be less than the EBL machine write time, t_{Write}. Figure 8-5 shows the timing components in the data flow of the software version of EPADPEC. The computation times of processors P21 and P22 are not included in $t_{EPADPEC}$ as they are carried in P1 in the data processing hardware (Figure 8-4).

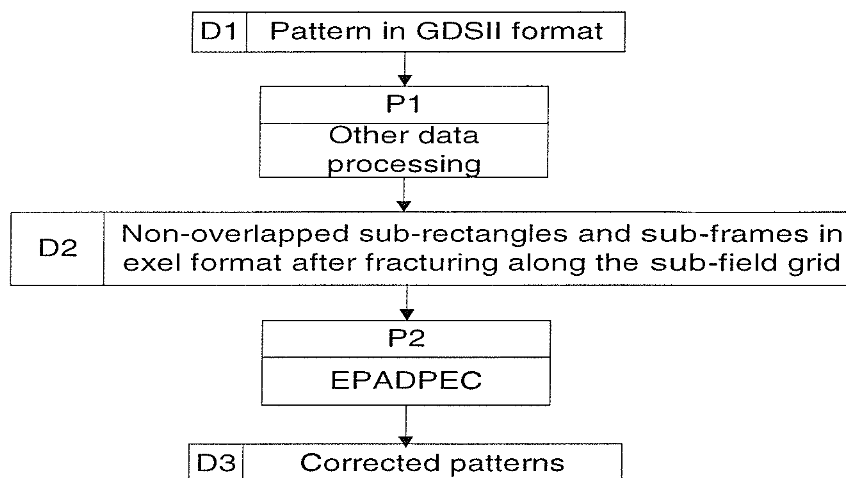


Figure 8-4 Data flow diagram of the data processing hardware in the proposed EBL system.

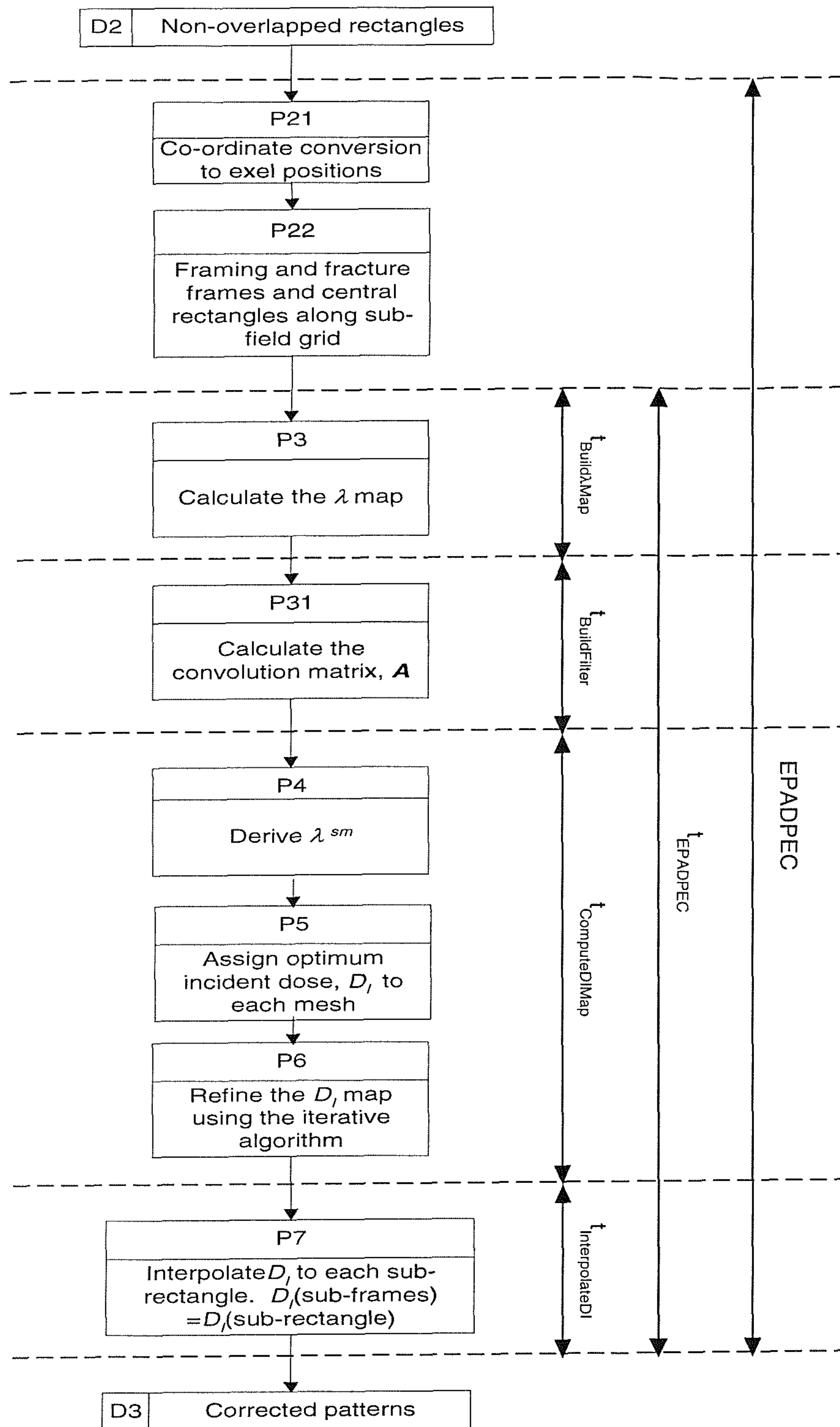


Figure 8-5 Timing components of EPADPEC.

Table 8-1 shows the prediction of the characteristics of future semiconductor devices by the Semiconductor Industry Association. Since EPADPEC needs a minimum frame width of 4α and the CD (critical dimension) values are less than the frame width (4α=296 nm) at 50kV, so a higher acceleration voltage will be needed to meet

the requirement. Current state-of-art EBL machines such as Leica VB6 (maximum clock frequency of 25 MHz) support an acceleration voltage of up to 100kV. At 100kV, the reported parameters values of the double Gaussian EDF are α=7nm, β=31.2µm and η=0.74 [146]. The α value gives a frame width that is smaller than the CD values of future devices. Thus, an EBL machine with 100kV acceleration voltage has sufficient resolution to support the required CD of future devices using EPADPEC.

Year of first product shipment	1999	2002	2005	2008	2011
CD of isolated lines (microprocessor gate length, nm)	140	100	70	50	35
CD control (nm)	14	10	7	5	4
Microprocessor chip area (mm^2)	300	360	430	520	620

Table 8-1 The Semiconductor Industry Association roadmap for future semiconductor devices [147].

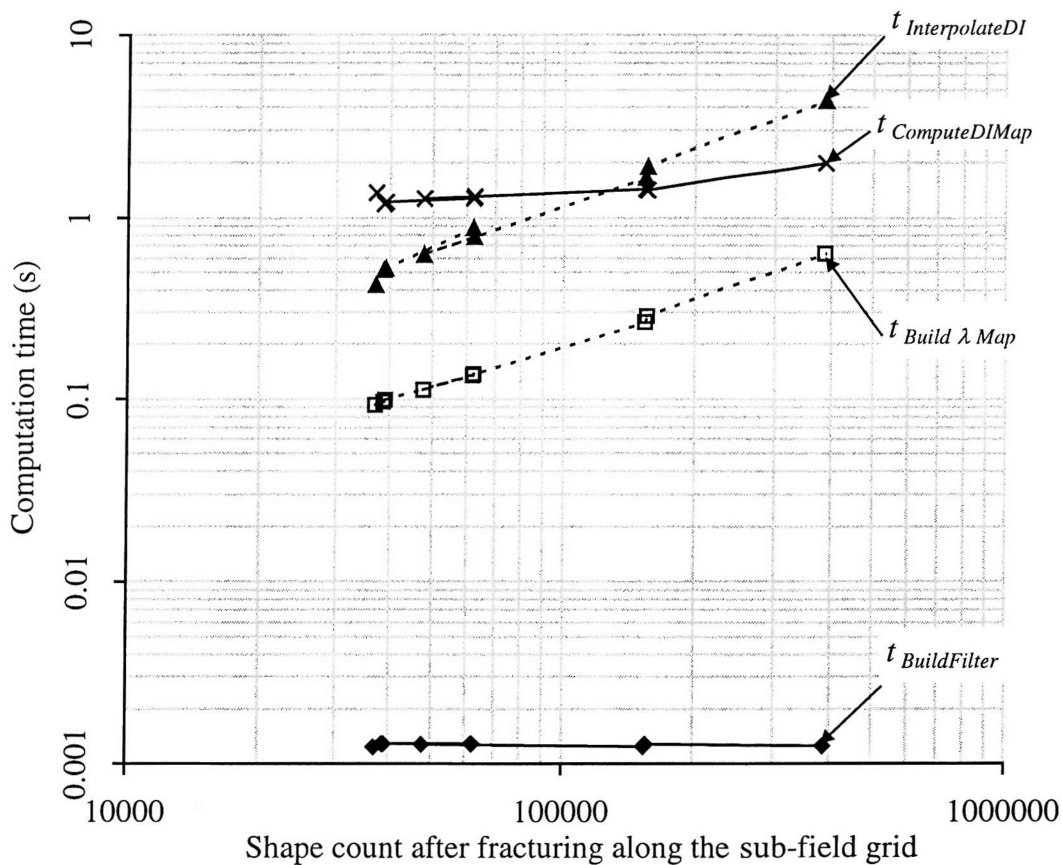
To compare $t_{EPADPEC}$ of the software version of EPADPEC (Figure 8-5) for future devices with t_{Write}, $t_{EPADPEC}$ of layer 11 (polysilicon gate layer) of MP16 is used as a basis for extrapolation to smaller geometry devices. It is assumed that:

- The target EBL machine is a 100kV, Gaussian spot, step and write machine and the parameter values of the double Gaussian EDF are α=7nm, β=31.2µm and η=0.74. Thus, EPADPEC uses these parameter values to obtain $t_{EPADPEC}$ of layer 11 of MP16.
- t_{Write} is inversely proportional to the maximum clock speed of the EBL machine. The target EBL machine is assumed to have a maximum clock speed of 100 MHz.
- Since a CD control of 10% is needed in lithography (see Table 8-1), EPADPEC uses M=S=0.75β=23.4µm, an order of A of 5 and a quantization level count of 128 (results from Chapter 6) to obtain $t_{EPADPEC}$ of layer 11 of MP16.
- When writing a pattern on a Gaussian spot, step and write EBL machine, the exel size is at least 5 times smaller than the CD value in normal practice [128]. Since t_{Write} increases as the exel size decreases for the same pattern, so an exel size of a fifth of the CD value will give the worst case scenario for performing real time EPADPEC correction. The exel size is always a fifth of the CD value.
- The shape and exel count is proportional to the chip area and is inversely proportional to the square of CD [119]:

$$Q(Future) = \left(\frac{ChipSize(Future)}{ChipSize(Present)} \right) \left(\frac{CD(Present)}{CD(Future)} \right)^2 Q(Present)$$

where $Q(r)$ is either the shape or exel count and the input parameters *Future* and *Present* stand for future and present devices respectively. As $t_{InterpolateDI}$ and $t_{Build\lambda Map}$ are proportional to the shape count (Figure 8-6) and t_{Write} is proportional to the exel count, so the above relationship also holds for them.

- The mesh size remains constant. Since $t_{ComputeDIMap}$ is proportional to the mesh site count, so $t_{ComputeDIMap}$ is proportional to the chip area.
- The order of A remains constant. Since $t_{BuildFilter}$ is proportional to the exel count in a mesh site, so $t_{BuildFilter}$ is inversely proportional to the square of the CD.
- The sub-field size is proportional to the minimum feature size;



Breakdown of t_{PEC} of EPADPEC in Figure 6-23(a) (section 6.3.2) into its components. From the graphs, the shape count has no influence on $t_{BuildFilter}$ and $t_{ComputeDIMap}$ while the rest of the timing components are proportional to the shape count. The slight difference in the $t_{ComputeDIMap}$ values is caused by the slight difference in the layout shadow among the various layers of MP16.

Figure 8-6 Breakdown of t_{PEC} of EPADPEC.

Figure 8-7 shows the projected computational times of EPADPEC components relative to t_{Write}. For all the CD values considered, $t_{EPADPEC}$ is about 24.6 times bigger than t_{Write}. Thus, for real-time correction, we will need to reduce $t_{EPADPEC}$ by at least 24.6 times. Since only $t_{InterpolateDI}$ and $t_{Build\lambda Map}$ are larger than t_{Write}, so we would only need to speed up these components.

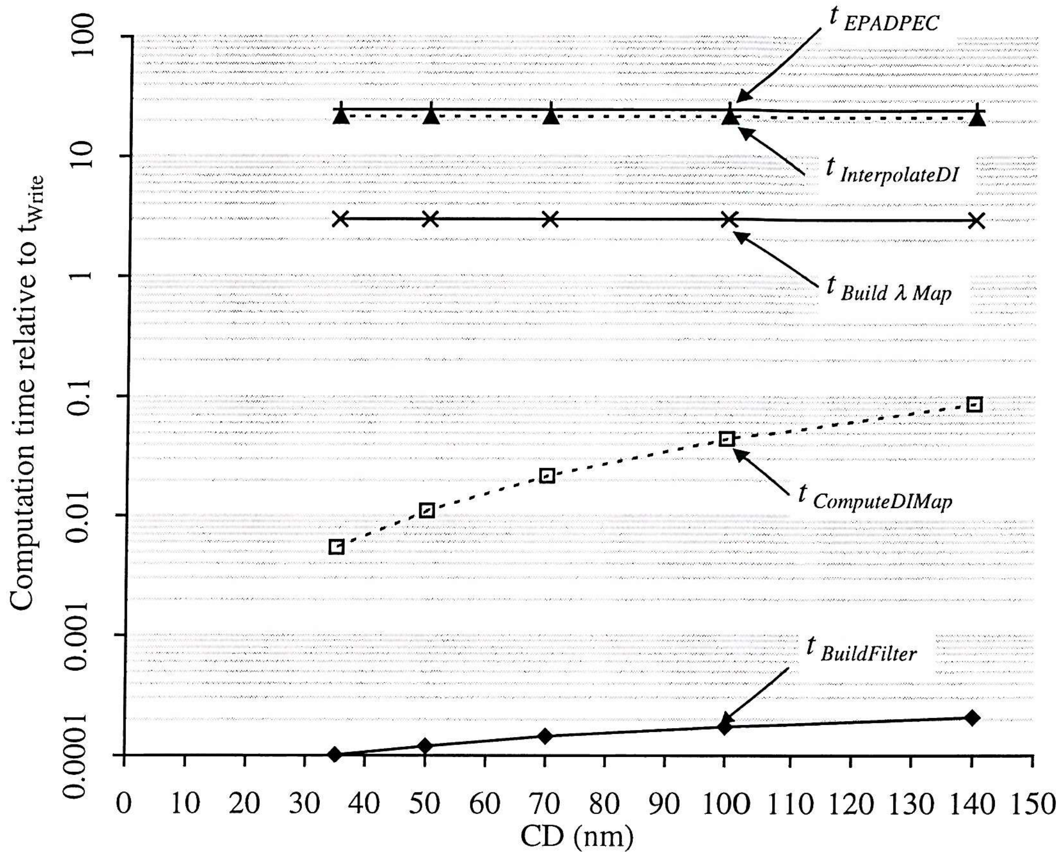


Figure 8-7 Projected computational times of EPADPEC components relative to t_{Write}.

8.3 Ways to achieve the required speed up factor

In general, there are two approaches to reduce $t_{EPADPEC}$:

1. Using faster computers.
2. Using dedicated hardware.

The following two sub-sections will discuss the expected speed up factor, the cost and the ease of implementation of these approaches. In estimating the cost, the following exchange rates of £1 = 2.51 Swiss Francs and £1 = US$1.62 are assumed.

8.3.1 Faster computers

In this section, the floating point benchmarking index, SPECfp95 is used to compare the relative speed between various computers. The higher the SPECfp95 value of a computer, the faster the computer is. The scale of the index is linear, i.e. a computer with a SPECfp95 value of 1 is twice as slow as a computer with a SPECfp95 value of 2. It is assumed that the Intel Alder system with a Pentium Pro 180 MHz microprocessor has the same SPECfp95 value as the Gateway 2000 G6-180 system with a Pentium Pro 180 MHz microprocessor. [148] contains the SPECfp95 values of various computers.

The easiest way to make a speed gain is to use a faster computer. According to the SPECfp95 results, the fastest workstation currently available in the market is the Compaq AlphaStation XP1000 with a single 667 MHz Alpha 21264 processor, costing about £8000 [149]. The workstation (SPECfp95=68.1) has a SPECfp95 value of 10.8 relative to the Intel Alder system (SPECfp95=6.08). The speed gain from running EPADPEC on the workstation is slightly less than half the minimum required speed gain, 24.6 for real time correction. Digital Equipment Corporation predicts the next generation of Alpha processor, 21364 (expected SPECfp95=120 and would be available in year 2000, costing about £8,000.) is about twice as fast as the 667 MHz Alpha 21264 processor [150]. Thus, this approach is only feasible by year 2000 at the earliest.

Another approach is to perform EPADPEC in parallel by partitioning spatially a circuit layout into smaller parts, and distributing the parts evenly either to each computer in a distributed network of computers [75] or to each processor in a multiprocessor system (such as a supercomputer or a multiprocessor server). Although a supercomputer can meet the required speed gain easily, it is the least favourable approach as the price of the supercomputer can exceed the price of an EBL machine. For example, a 256 processor Cray T3E 1200 supercomputer costs about

£12million[1] [151], while a Leica VB-6 EBL machine costs about £2million [152]. Based on the SPECfp95 results, the fastest workstation currently available in the market is the Compaq AlphaServer GS140 Model 6/700 with up to 14 Alpha 21264 processors. Each processor runs on a clock frequency of 700 MHz. A Compaq AlphaServer GS140 Model 6/700 with 8 processors cost about £185000 [153] and just meets the speed gain requirement with a SPECfp95 value of 26.0[2] relative to the Intel Alder system.

8.3.2 Dedicated hardware

Since only $t_{InterpolateDI}$ and $t_{Build\lambda Map}$ are larger than t_{Write} (Figure 8-7), so we would only need to speed up these components using some dedicated hardware while we still could perform the other components in software; making it a hardware/software co-design. Figure 8-8 shows the block diagram of such a hardware/software co-design implementation of EPADPEC. The host computer uses the BuildλMap hardware to compute the λ map before evaluating the convolution matrix, A and the D_I map. The host computer is at least as fast as the G6-180 system (Pentium Pro 180 MHz). The dedicated hardware performs the interpolation and quantization of the incident dose on the fly as the exposure shapes are sent to the electron beam hardware. As shapes are sorted in sub-field sites and M=S, for each new sub-field site, the incident doses of the nine nearest mesh sites are read into the registers to avoid the need of requesting the values repeatedly from the host computer for each shape in the same sub-field site. Since the BuildλMap hardware and the rest of the dedicated hardware are not used simultaneously, so the hardware can be shared between the two modules.

[1] Each processor is capable of 1.2 GFLOPS and each MFLOPS of the supercomputer costs US$64.

[2] The SPECfp95 value of a Compaq AlphaServer GS140 Model 6/700 with 8 processors is not available. The value is estimated as

$$SPECfp95[Model6/700(8CPU)] = \frac{SPECfp95[Model6/525(8CPU)]}{SPECfp95[Model6/525(1CPU)]} SPECfp95[Model6/700(1CPU)]$$

where *SPECfp95[Model6/525(1CPU)]* is the SPECfp95 value of a single processor Compaq AlphaServer GS140 Model 6/525. Model 6/525 uses the same processor as Model 6/700 except the processor operates at a lower clock frequency of 525 MHz.

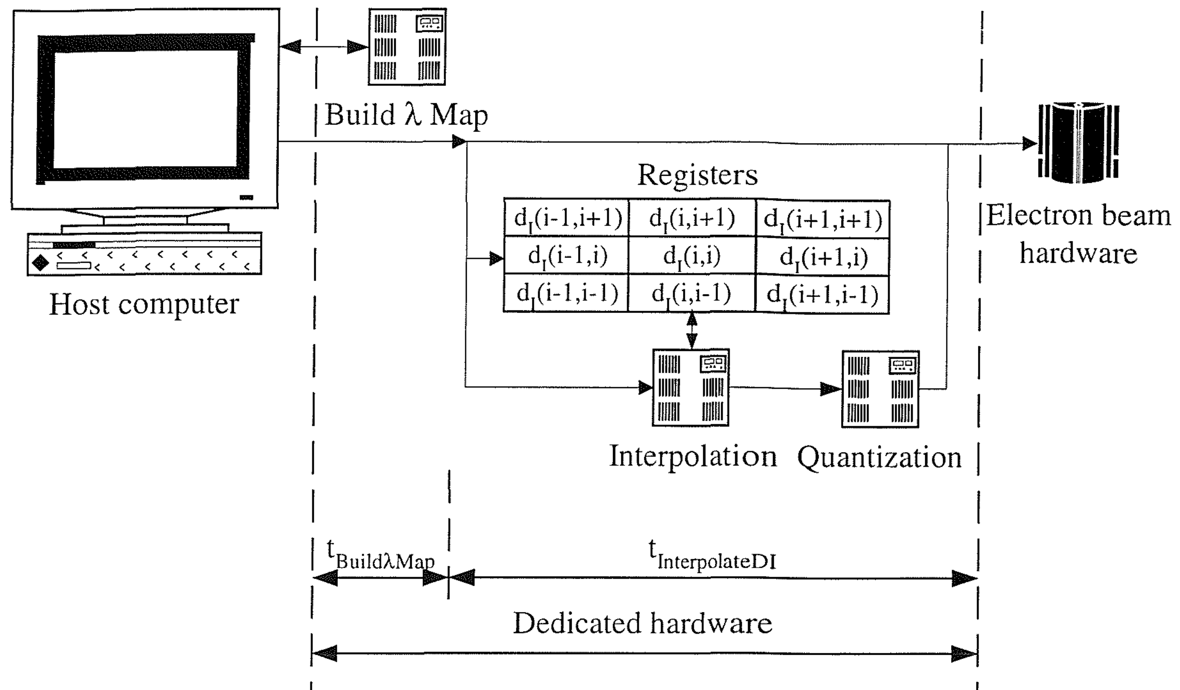
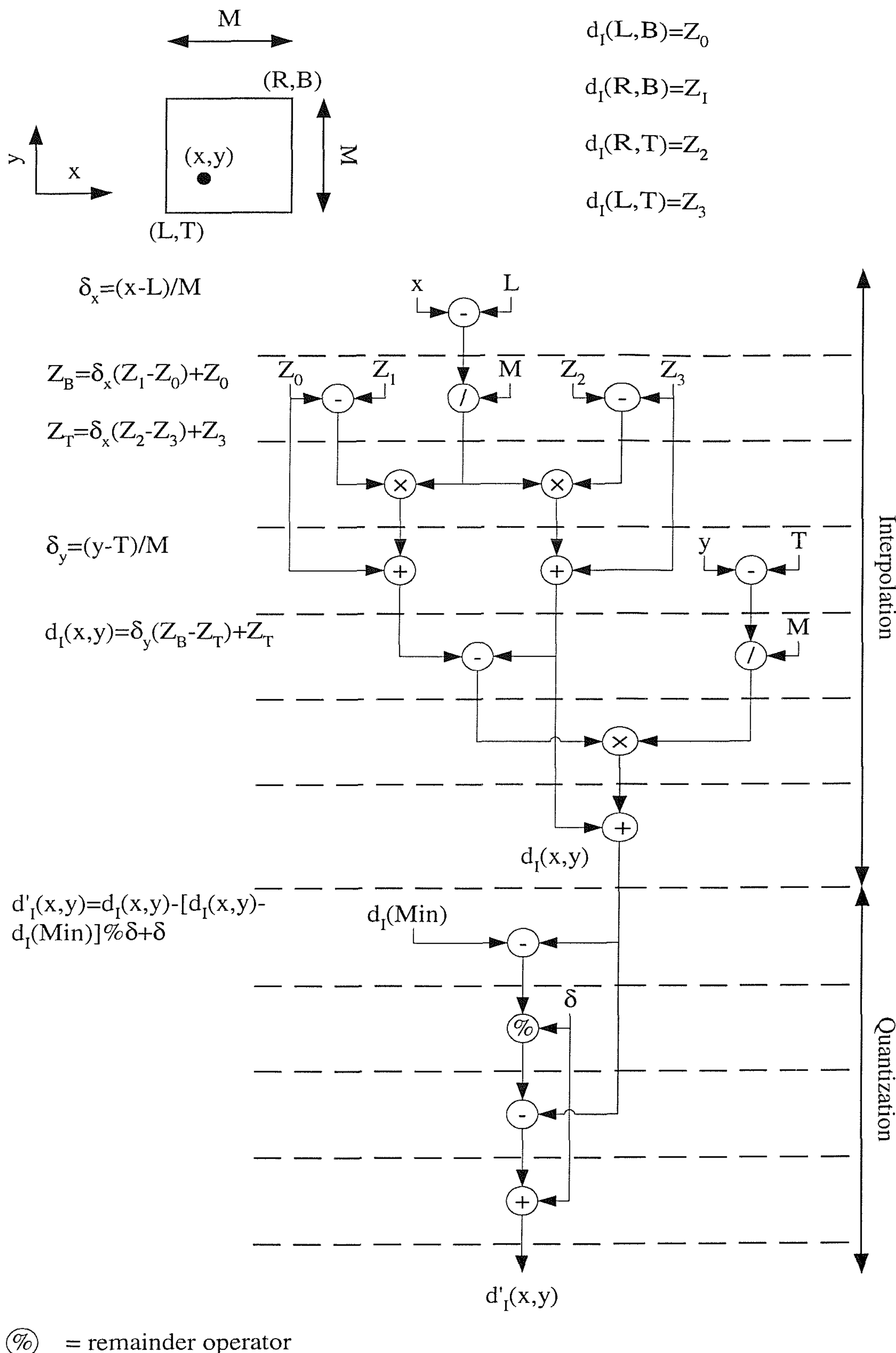


Figure 8-8 Block diagram of a hardware/software co-design implementation of EPADPEC.

The minimum t_{Write} value of a shape is 10ns when the shape consists of only one exel. (The projected layout in Figure 8-7 has an average exel count of 43.6, giving a t_{Write} value of 436ns.) Thus, to perform the interpolation and quantization in real time, the dedicated hardware must take ≤ 10ns per shape. The interpolation and quantization can be fully pipelined and paralleled to give one result per clock cycle as shown in Figure 8-9[1] (In practice, the algorithm would be specified in a *behavioural* language and use a behavioural synthesis tool to generate the circuit). L, R, T and B are the co-ordinates of the centre points of the four nearest mesh sites, while $d_I(min)$ and δ are the minimum incident dose and the step size between adjacent quantized levels respectively.

[1] The floating point number has a 12-bit mantissa and a 6-bit exponent.



Z_0, Z_1, Z_2, Z_3, δ and d_I(min) are floating point numbers, while x, y, L,T, R, B and M are 32-bit numbers. The dash line denotes the boundary of pipeline stages. The registers between the pipeline stages have been omitted in the diagram for clarity.

Figure 8-9 A pipelined and parallel hardware implementation of the interpolation and quantization procedures.

Loading from the nine registers (Figure 8-8) through a multiplexor, the loaded values of Z_0, Z_1, Z_2 and Z_3 depend on the relative position of the interpolation point, (x,y) to the centre point, (C_x, C_y) of the current mesh site[1] (Figure 8-10). Because of the five point interpolation strategy for central rectangles (section 5.2.1.3), another four instances of the interpolation hardware and a comparator[2] are necessary to select the maximum interpolated value before passing it to the quantization hardware.

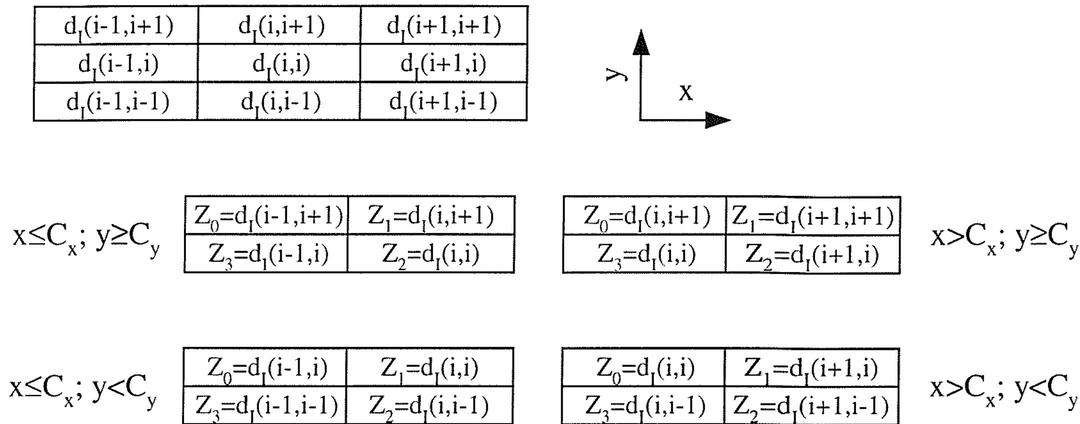


Figure 8-10 Selection of the incident dose values of the four nearest mesh sites from the nine registers.

A maximum clock frequency of 167 MHz is possible for ASIC devices fabricated using the Europractice's Alcatel Microelectronics 0.35 μm C035M-D technology process [154]. Thus, the process can well meet the minimum required clock frequency of 100 MHz for real-time interpolation and quantization. Table 8-2 lists the required arithmetic units for the dedicated hardware, while Table 8-3 shows the estimated chip area of the dedicated hardware fabricated using the technology process. The estimated cost of the synthesized approach is currently about £4000 (Table 8-4).

[1] The calculation of the relative position needs two 32-bit adders.

[2] The comparator is made of 4 floating point adders.

Module	Floating point adder	Floating point multiplier	Floating point divider
Interpolation	6×5+4	3×5	0
Quantization	3	0	1
Module	32-bit adder	32-bit divider	Multiplexor
Interpolation	4×5	2×5	1×5
Quantization	0	0	0

Table 8-2 The number of arithmetic units required by the dedicated hardware.

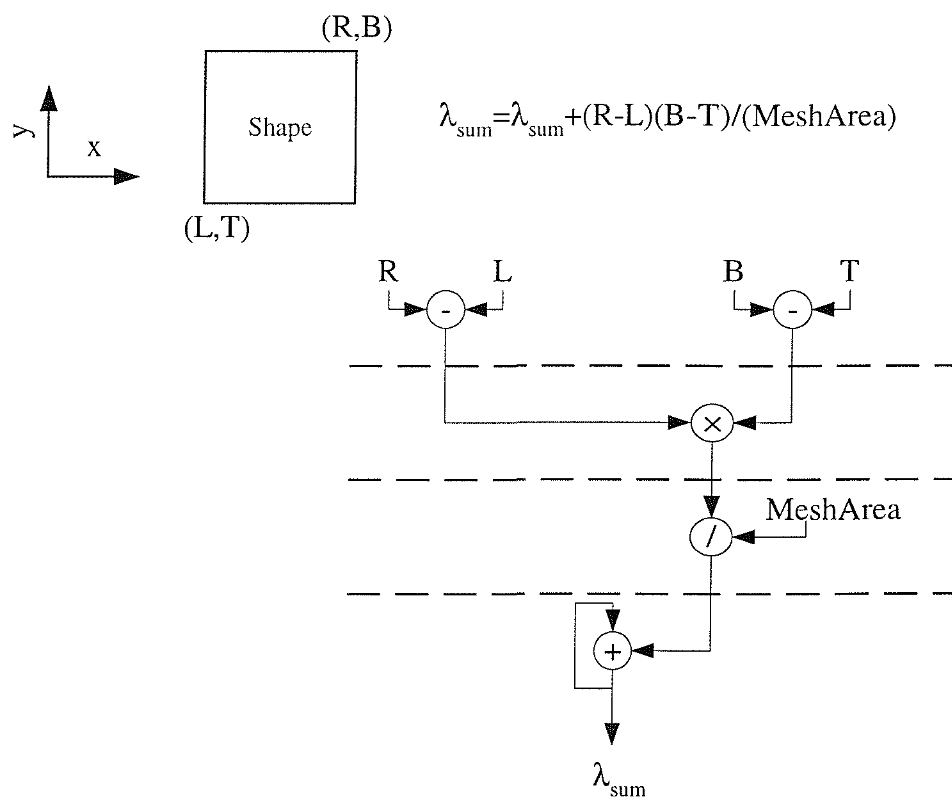
Arithmetic unit	Estimated chip area per unit (mm^2)	Required number of units	Sub-total chip area (mm^2)
Floating point adder	0.4	37	14.80
Floating point multiplier	0.2	15	3.00
Floating point divider	0.4	1	0.40
32-bit adder	0.004	20	0.08
32-bit divider	0.4	10	4.00
Multiplexor	0.002	5	0.01
Total chip area (mm^2)			22.29
Total chip area (mm^2) adjusted for registers and routing spaces in the circuit layout.			30.00

Table 8-3 The estimated chip area of the dedicated hardware fabricated using the 0.35 μm C035M-D technology process.

Item	Cost (£)
Host computer	2300
ASIC fabrication cost at 450 Euro per 10mm^2 chip size [155]. (Educational price rate; £1=1.58Euro)	580
Chip packaging	400
Prototyping circuit board, connectors, cables and others.	720
Total cost	4000

Table 8-4 Estimated cost of the hardware/software co-design approach.

From Figure 8-7, $t_{BuildFilter}$ and $t_{ComputeDIMap}$ are at most 0.002% of $t_{EPADPEC}$. Their values are negligible compared to t_{Write}. $t_{Build\lambda Map}$ is about 3 times larger than t_{Write}. Thus, it is necessary to reduce $t_{Build\lambda Map}$ by at least the factor for real-time EPADPEC correction. 61% of $t_{Build\lambda Map}$ is spent in evaluating the λ values of shapes, while the rest of it (39%) is spent on accessing the shape co-ordinates and the λ map values from the memory. The G6-180 system runs on a bus speed of 60 MHz and produces about one million λ values per second. Using two 32-bit adders, one 32-bit multiplier, one floating point divider and a floating point adder, the computation of the λ value of a shape can be fully pipelined and paralleled easily to give one result per clock cycle as shown in Figure 8-11. No additional hardware is required in this



λ_{sum} is a floating point number, while L,T, R, B and MeshArea are 32-bit numbers. The dash line denotes the boundary of pipeline stages. The registers between the pipeline stages have been omitted in the diagram for clarity.

Figure 8-11 A pipelined and parallel hardware implementation of the evaluation of the λ values of shapes.

module as the required arithmetic units can be shared with the units in the interpolation and quantization module. Assuming the bus speed of the dedicated

hardware and the clock frequency are 100 MHz, then $t_{Build\lambda Map}$ is reduced by a factor of

$$\frac{100\%}{39\%\left(\frac{60MHz}{100MHz}\right)+61\%\left(\frac{1MHz}{100MHz}\right)} \approx 4.16$$

which is sufficient to meet the speed gain requirement. Thus, real-time EPADPEC correction is feasible using the hardware/software co-design approach.

8.4 Conclusions

Table 8-5 summarizes the estimated speed up factor and the cost of various approaches that accelerate the computation of EPADPEC. Although the workstation approach is the easiest to implement, it fails to provide enough speed gain for real-time EPADPEC correction. The supercomputer method is too expensive to be practical. The hardware/software co-design and the server method are the feasible ways to achieve real-time EPADPEC correction. The former approach offers the best value for money, but its implementation requires more effort than the latter scheme.

Approach	**Estimated speed up factor relative to the G6-180 system (Pentium Pro 180 MHz)**	**Estimated cost (£)**
Supercomputer	>100.0	10,000,000
Single processor workstation	10.8	8,000
8 processors server	26.0	185,000
Hardware/software co-design	≥24.6	4,000

Table 8-5 Summary of the estimated speed up factor and cost of various approaches to accelerate the computation of EPADPEC.

Chapter 9

Conclusions

The last chapter summarizes the contributions of this research and briefly outlines further work arising from the outcomes.

9.1 Contributions

The principal contributions of this research are:

- The development of the electron-beam data preparation system.
- The development of auxiliary tools: the proximity effect simulator, the electron beam write time simulator and the GDSII layout browser. (presented in Appendices B and C.)
- The formulation and development of EPADPEC.
- The formulation and development of the corner correction and physical shape fracturing algorithm.
- The feasibility study of performing electron beam proximity effect correction in real time.

9.2 Further work

9.2.1 Hardware/software co-design

From the feasibility study in Chapter 8, the best way to achieve real time EPADPEC correction is the hardware/software co-design approach. The system should be sythesized from a behavioural description of the algorithm using a synthesis tool. Much complementary work in this area is undertaken by other members of the research group.

Pattern pre-processing: overlap removal, scaling, tone reversal and biasing is best done in software on a host computer. The custom hardware will perform EPADPEC. The hardware is interfaced to the host computer at the address/data bus level, making it a true co-design.

9.2.2 Correction of small shapes whose width is smaller than the frame width (4α)

EPADPEC and the corner correction algorithm are designed to work on large shapes, that is, shapes where the minimum dimension is $> 4\alpha$. For shapes whose dimension is smaller than 4α, the optimum parameter values of the corner correction scheme (section 7.1.4) and the weight of 2 for λ and D_I of frames are not applicable. These small shapes need new correction method. As CD approaches α, the proximity problem becomes more ill-posed (section 4.1), making the correction more difficult. The impact of the method on the correction accuracy and speed of EPADPEC will need to be studied.

9.2.3 Extension for non-Manhattan shapes

The electron beam data preparation system supports only Manhattan shapes. To process polygons of arbitrary shapes, the system will need to be able to process trapeziums as well as rectangles. It is necessary to modify the procedures for pattern pre-processing, computing the λ values of shapes and calculating the shape interpolation points.

9.2.4 Adaptation of EPADPEC to projection type EBL machines

As mentioned in section 2.4, the low throughput of the conventional EBL machines can be addressed by changing their serial writing nature (direct write) to a parallel writing nature (projection type). Unfortunately, projection type EBL machines are not capable of dose modulation, which is required by EPADPEC. Although dose modulation can be achieved on the machines by having multiple passes of the same pattern, this will reduce significantly the throughput of the machines, the very thing they are designed to accelerate. Recently, Sigma-C, (the developer of CAPROX) have developed a method for translating dose modulation into the corresponding shape adjustment. The method is not very accurate, but it is the only reported technique for the conversion and does not suffer from the throughput degradation of the former

method. Thus, it is the best way to adopt EPADPEC for projection type EBL machines at the moment.

9.2.5 Formatter for other EBL machine formats

In theory, EPADPEC will work on any EBL machines that are capable of dose modulation. For machines without the dose modulation capability, EPADPEC can be adapted to work on these machines as mentioned in section 9.2.4. Currently, the electron beam preparation system only supports Leica EBMF EBL machines. The output formatter of the system converts exposure patterns from the internal format to Cambridge source pattern data language format [156]. To drive Leica EBMF EBL machines, it is necessary to convert the Cambridge source format into the EBMF binary format using the converter software provided by Leica [157]. To support other EBL machines, the system needs additional formatters that convert the internal format to the specific format used by the machines.

9.2.6 Resist calibration

Quantitative assessment of the efficacy of the correction scheme is difficult without a calibrated resist model.

As mentioned in section 6.2.2.2, further work is needed to determine accurately the resist threshold energy density value of the UVIII resist, and to characterize and optimize the process so that identical results are reproducible under identical process procedures. This is outside the scope of this thesis.

9.2.7 Cost and deliverables

Upon completion of the further work described in sections 9.2.1 to 9.2.6, the following deliverables would sensibly be expected:

I An electron beam data preparation system that could process Manhattan shapes and trapeziums. This system would include the hardware/software co-design of EPADPEC and (optional) formatters for other EBL machines.

II Two papers would be published:

1 This paper would present the results from the work discussed in sections 9.2.2 and 9.2.4. It would present the improvement in the correction accuracy by the new algorithm for small shapes using scanning electron microscope photographs of fabricated patterns that are corrected using EPADPEC with and without the new algorithm.
2 This paper would describe the hardware/software co-design of EPADPEC and report the correction speed. It would demonstrate the enhancement in shape fidelity by the co-design EPADPEC system using scanning electron microscope photographs of fabricated uncorrected and EPADPEC corrected patterns.

A researcher for about three years would be necessary to carry out the further work. Table 9-1 and Table 9-2 list the estimated resources for this. A *likelihood* figure is assigned to each item in Table 9-2. The figure indicates the degree of risk involved in the work: a value of *speculative* means the work has a high chance of failure, while a value of *pure development* means the work has a high chance of success.

Item	**Cost (£)**
1. Salary	85,000
2. A Sun workstation for designing the ASIC	10,000
3. A personal computer plus software and peripherals, for documentation and software development.	4,000
4. Software licenses	10,000
5. Fabrication cost (50 runs at £1000 each)	50,000
6. Consumable items, e.g. papers, toners, stationery etc.	1,000
7. Cost of the dedicated hardware (for two runs of ASIC fabrications and prototyping using field gate programmable array devices, section 8.3.2).	10,000
8. Attendences to conferences and trips to industry collaborators' offices.	10,000
Total cost	180,000

Table 9-1 Estimated cost for achieving the deliverables outlined.

Further work	Required resources (excluding salary)		Likelihood figure
	Money (£)	Man-months	
1. Hardware/software co-design (section 9.2.1).	30,000 (Items 2, 4 and 7 in Table 9-1)	13	Pure development
2. Correction of small shapes (section 9.2.2).	4,000* (Item 3 in Table 9-1)	3	Speculative
3. Extension for non-Manhattan shapes (section 9.2.3).	4,000* (Item 3 in Table 9-1)	3	Pure development
4. Adaptation to projection type EBL machines (section 9.2.4).	4,000* (Item 3 in Table 9-1)	3	Probably possible
5. Additional formatters (section 9.2.5).	4,000* (Item 3 in Table 9-1)	3	Pure development
6. Resist calibration (section 9.2.6).	50,000 (Item 5 in Table 9-1)	6	Probably possible

* The total estimated cost for further work 2-5 is £4,000.

Table 9-2 Estimated resources for achieving the deliverables outlined.

Appendix A

Publication:

Enhanced pattern area density proximity effect correction,

by C. S. Ea and A. D. Brown,

Journal of Vacuum Science and Technology **B**, Vol. 17(2), Mar/Apr 1999, pp. 323-333.

Enhanced pattern area density proximity effect correction

C. S. Ea and A. D. Brown
Department of Electronics and Computer Science, University of Southampton, Highfield, Southampton SO17 1BJ, United Kingdom

(Received 6 July 1998; accepted 8 November 1998)

In this article, we describe and analyze the sources of correction error in the pattern area density proximity effect correction (PADPEC) method of Murai *et al.* [F. Murai, H. Yoda, S. Okazaki, N. Saitou, and Y. Sakitani, J. Vac. Sci. Technol. B **10**, 3072 (1992)]. By focusing on the dominant contribution to the overall error, we are able to enhance the technique further, developing the enhanced PADPEC, or EPADPEC method. EPADPEC further reduces linewidth errors by factors ranging from 4 to 11, while increasing the computation time by a factor of only 1.23. [Migrating EPADPEC to a dedicated (parallel) environment will enable the data throughput rate to be dramatically increased.] © *1999 American Vacuum Society.* [S0734-211X(99)01102-6]

I. INTRODUCTION

Current trends of reducing the physical size of circuit components in integrated circuit technology are rapidly approaching the point at which electron beam lithography (EBL) with its high resolution capability will play a pivotal role in device fabrication. In theory, the resolution of electron optical systems can approach 0.1 nm.[1] However, the ultimate resolution of EBL is set by the electron proximity effect and the resist development mechanisms. For high voltage electron beams ($\geqslant$50 kV), the *variation* of deposited energy, E_b in a resist by backscattered electrons is negligible in a small area within the backscattering radius (the *absolute* value is a strong function of the beam energy). This property simplifies proximity effect correction (PEC) and Murai *et al.*[2] have employed it to develop a fast PEC scheme called pattern area density proximity effect correction (PADPEC). The speed of PADPEC is gained at the expense of correction accuracy.

In this article, we first describe briefly the correction procedures of the PADPEC system before discussing the sources of correction error. Next, we present an enhanced PADPEC (EPADPEC) technique which improves the correction accuracy, followed by simulation results. Throughout the article, we assume an infinite contrast resist and a normalized double Gaussian expression[3] describes exactly the deposited energy density profile at some resist depth due to a point source electron beam. The deposited energy density distribution function contains two normalized Gaussian terms, a narrow one for forward scattered electrons and a wide one for backscattered electrons:

$$f(r)=\frac{1}{\pi(1+\eta)}\left[\frac{1}{\alpha^2}\exp\left(\frac{-r^2}{\alpha^2}\right)+\frac{\eta}{\beta^2}\exp\left(\frac{-r^2}{\beta^2}\right)\right], \quad (1)$$

where r is the distance from the point of incidence, α and β reflect the beam broadening effects of the forward and backward scattering processes, and η is the deposited energy ratio of backscattered electrons to forward scattered electrons. Unless stated otherwise, the resist threshold deposited energy density is assumed to be 1 and the parameter values of the normalized double Gaussian deposited energy density distribution function are assumed to be $\alpha=74$ nm, $\beta=9.6\ \mu$m, and $\eta=0.83$ in this article. These parameter values[4] are experimentally determined using a 0.5-μm-thick chemically amplified negative electron beam resist, SAL601 and the HL800D direct write system[5] with an acceleration voltage of 50 kV.

II. PADPEC METHOD

This section contains a brief precis of the pattern area density proximity effect correction method described by Murai *et al.*[2]

1. The circuit pattern is partitioned with a fixed sized mesh. The mesh size (M) is chosen such that the variation of E_b within a single mesh site is negligible (M is thus a function of the beam energies).

2. The pattern area density (λ) (defined as the ratio of the written area to the total area in a region) is computed in each mesh site, giving a λ map for the circuit pattern.

3. The λ map is convolved with a filter, giving a smoothed λ map, λ^{sm} map.

4. Each circuit shape (polygon) is fractured into rectangles; and a λ value (called λ') is assigned to each rectangle. Taking the position of the rectangle to be its center, λ' is the linear interpolation of the λ^{sm} values of the four nearest mesh sites to the rectangle.

5. Finally, an incident dose (charge per unit area)

$$D=C\frac{2(1+\eta)}{1+2\lambda'\eta} \quad (2)$$

is assigned to each rectangle. C is a constant, depending on the specific resist and the beam voltage used.

Two types of filters have been used to smooth the λ map. 1. Murai *et al.*[2] used a form of template convolution; λ^{sm} is given by

$$\lambda^{sm}(i,j)=\sum_{l=i-(R_{sm}-1)/2}^{i+(R_{sm}-1)/2}\ \sum_{m=j-(R_{sm}-1)/2}^{j+(R_{sm}-1)/2} a(l,m)\lambda(l,m), \quad (3)$$

where

$$a(i,j)=\left[\mathrm{erf}\left(\frac{M}{2\beta}\right)\right]^2,$$

$$a(i\pm 1,j)=a(i,j\pm 1)$$

$$=\frac{1}{2}\left[\mathrm{erf}\left(\frac{M}{2\beta}\right)\right]\left[\mathrm{erf}\left(\frac{3M}{2\beta}\right)-\mathrm{erf}\left(\frac{M}{2\beta}\right)\right], \quad (4)$$

$$a(i\pm 1,j\pm 1)=\frac{1}{4}\left[\mathrm{erf}\left(\frac{3M}{2\beta}\right)-\mathrm{erf}\left(\frac{M}{2\beta}\right)\right]^2$$

and R_{sm} is the *order* of the convolution matrix (A); here $R_{sm}=3$. $a(l,m)$ is the fraction of the energy of the backscattered electrons deposited at (i,j) mesh site center point due to a uniform incident dose applied to the entire (l,m) mesh site.

Kasuga *et al.*[4] used a uniform filter to smooth the map

$$\lambda^{sm}(i,j)=\sum_{l=i-(R_{sm}-1)/2}^{i+(R_{sm}-1)/2}\ \sum_{m=j-(R_{sm}-1)/2}^{j+(R_{sm}-1)/2} a(l,m)\lambda(l,m), \quad (5)$$

where $a(l,m)=1/R_{sm}^2$. Note that R_{sm} is constrained to be an odd number, but may not necessarily be equal to 3.

The former and the latter methods are referred as MPADPEC and KPADPEC, respectively, in the rest of this article.

Repeating Eq. (5) iteratively produces a correction result that first improves with increasing iteration number and then degrades. The best iteration count occurs when the weighting distribution of the resultant cascaded uniform filter is approximately equal to the E_b distribution. This is actually the synthesis of a Gaussian filter by cascaded uniform filters.[6,7] The optimum iteration count, n, is given by the following expression:

$$n=f\left(\frac{6\beta^2}{M^2(R_{sm}^2-1)}\right),$$

where $f(x)$ is a function that returns the nearest integer to x. This technique will fail to synthesize the Gaussian filter that represents the E_b distribution if

$$M>\frac{\beta}{\sqrt{\frac{1}{6}(R_{sm}^2-1)}}.$$

III. SOURCES OF CORRECTION ERROR IN THE PADPEC TECHNIQUE

There are five sources of error in the PADPEC algorithms described in the previous section:

1. interpolation;
2. assuming a constant incident dose per shape;
3. smoothing;
4. neglect of forward scattering;
5. the fundamental accuracy of the incident dose formula [Eq. (2)] itself.

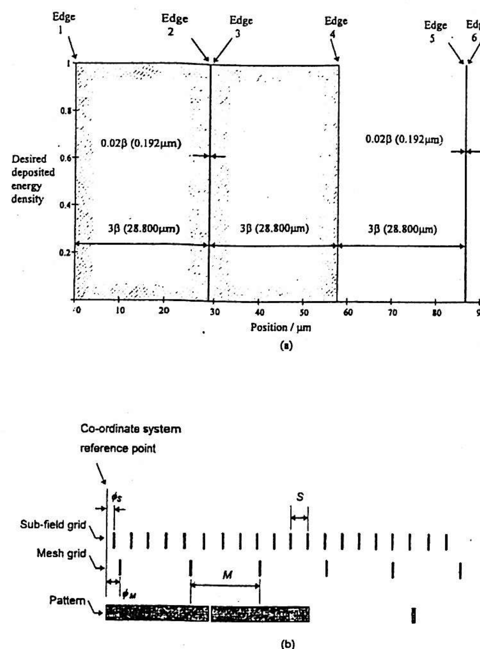


FIG. 1. (a) Test pattern for the relative error analysis of Sec. III. (b) Geometric frames of reference of the pattern, the mesh, and the subfield grid.

The following sections contain an analysis of these error components. As all the contributions are pattern dependent, the one-dimensional pattern of Fig. 1(a) is used as a basis for the analysis. This pattern is commonly used in the field to benchmark analysis of this type. The figure represents an isolated line and an isolated space, and form an ideal basis for our analysis. However, intermediate spatial frequencies are not presented by this pattern. The two-dimensional test pattern of Fig. 13 allows us to do this by varying λ (the average occupancy of the dummy shapes). The effects are tabulated in Table II. Figure 2 defines the figure of merit

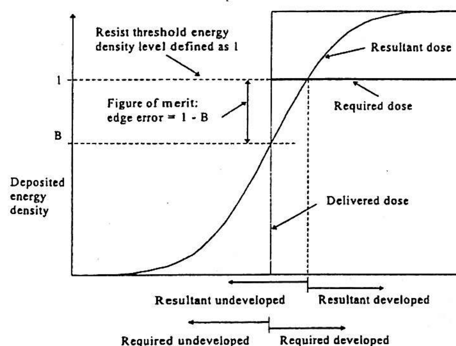


FIG. 2. Definition of edge error.

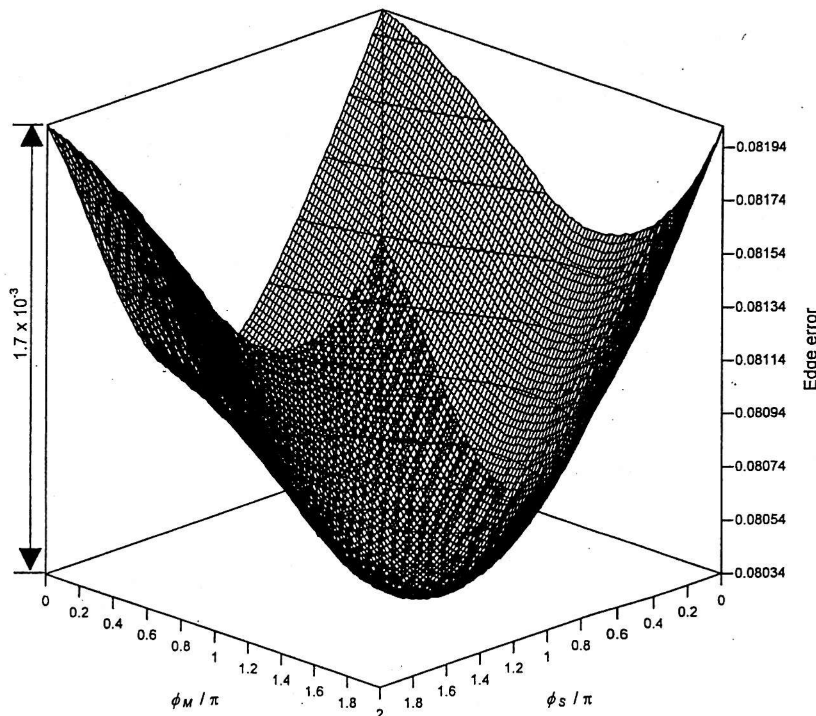


FIG. 3. Variation in edge error for edge 1, with $M=0.2\beta$ and $S=0.02\beta$, as ϕ_M and ϕ_S move from 0 to 2π.

(edge error)[8] used to compare the contributions from sources 1, 2, and 5. (It is not applicable to sources 3 and 4, so a different comparison criteria will be introduced in Secs. III C and III D.) The edge error is related to shape dimension error by the tangent of the resultant deposited energy density profile at the shape edge.[8] As this is a constant over the entire pattern for a given deposited energy density distribution function, we will use the former figure of merit only throughout the article.

A. Interpolation

The first two effects (interpolation and assuming a constant incident dose per shape) are considered together. In the overall data flow outlined in Sec. II, λ' is derived from λ^{sm} using linear interpolation between four nearest mesh centroids and the rectangle center. Clearly, as M decreases, the errors introduced by this step will also decrease. This is discussed in Sec. III B.

B. Assuming a constant incident dose per shape

Assuming the incident dose level throughout a shape to be constant can also lead to errors, especially if the shape is large. This is handled by fracturing each shape into a number of *subshapes* before processing begins. The fracturing is performed by overlapping a grid (the *subfield* grid) onto the pattern; the subfield grid spacing (S) is independent of, and usually less than, M. As with the previous effect, allowing S to approach zero will minimize this source of error.

It is worth looking in some detail at the sensitivity of edge error (Fig. 2) with respect to the geometric baseline (i.e., the origin of the frame of reference) of the calculations. Figure 1(b) shows the one-dimensional pattern of Fig. 1(a), overlaid with the mesh grid and the subfield grid. Without loss of generality, the origins of both these grid systems are assumed to coincide with the left hand edge of the pattern. When M is large (comparable to the pattern feature size), the sensitivity of edge error with respect to ϕ_M (the *phase* of the mesh grid) will be high, and this will decrease as the value of M decreases. Figure 3 shows the variation in edge error for edge 1, with $M=0.2\beta$ and $S=0.02\beta$, as ϕ_M and ϕ_S move from 0 to 2π. The amplitude of this phase plot (here 1.7 $\times 10^{-3}$ see Fig. 3) can be used to derive an upper limit to the tolerance on the value of the edge error.

Figure 4 shows the edge errors appearing on the pattern of Fig. 1(a) as M and S are varied. The tolerances of edge errors are shown as error bars. Note that, in varying M, the order of A in Eq. (3) was also varied such that the physical area convolved with A remained constant; this allows a more sen-

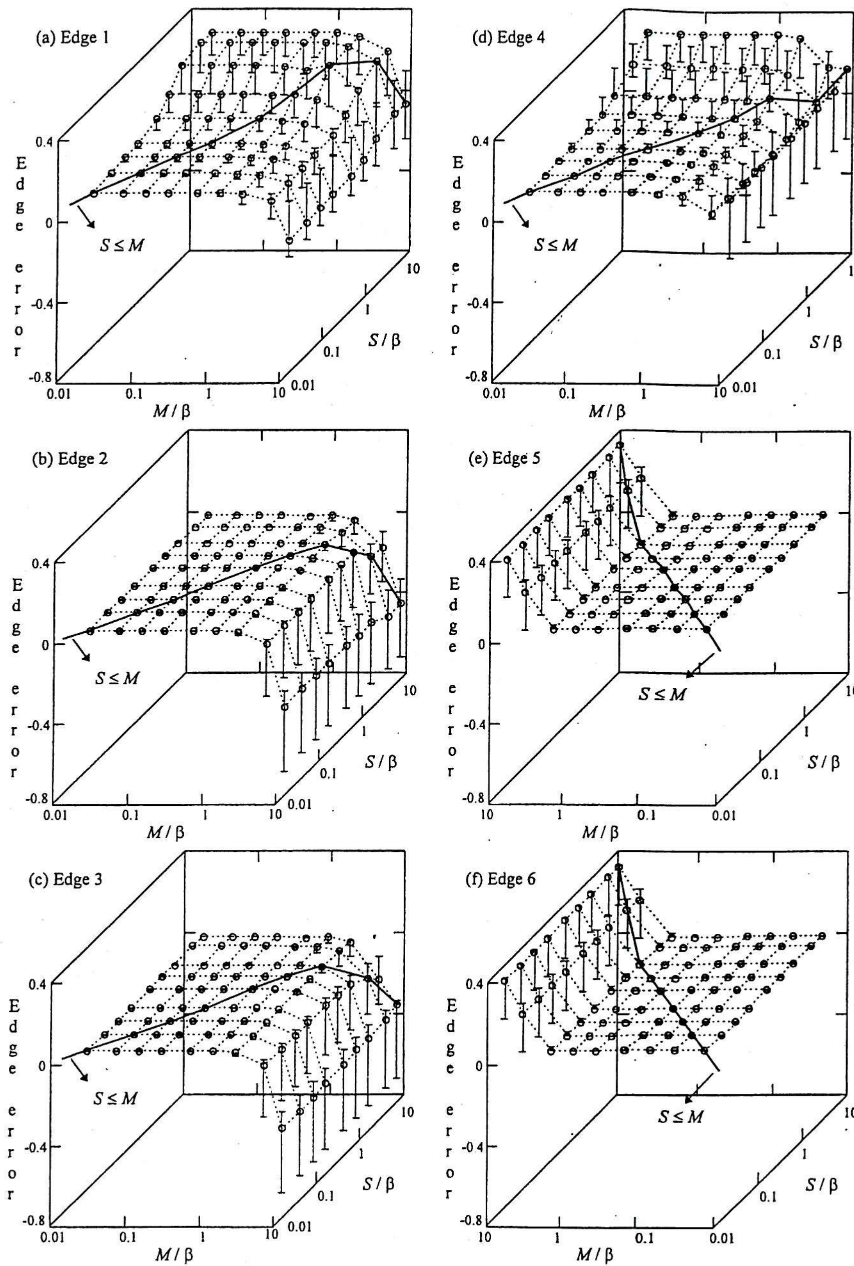


FIG. 4. Variation of edge errors due to changing the values of M and S.

sible comparison. The physical area is large enough to include the backscattered energy from the entire test pattern at any mesh site center point. For all edges and values of S, the edge errors saturate as M decreases. Edges 1, 2, and 5 have the same tolerances of edge errors as edges 4, 3, and 6, respectively. The differences in edge errors of a similar edge pair are due to the geometric base of the calculations mentioned in the previous paragraph. [Note the M abscissa coor

dinate system in Figs. 4(e) and 4(f) is reversed to make the shape of the surface clearer.] (As an aside, we note that decreasing M and S will results in a polynomial increase in computation time.)

C. Smoothing

E_b at a point (x,y) due to a unit constant incident charge applied to a rectangular shape is given by

$$E_b(x,y)=\frac{k\eta}{4(1+\eta)}\left[\operatorname{erf}\left(\frac{R-x}{\beta}\right)-\operatorname{erf}\left(\frac{L-x}{\beta}\right)\right]\times\left[\operatorname{erf}\left(\frac{B-y}{\beta}\right)\operatorname{erf}\left(\frac{T-y}{\beta}\right)\right],$$

where R, L, T, and B are the right, left, top, and bottom coordinates, respectively, of the rectangle. k is the charge to energy conversion factor and is a constant. The value of k depends on the specific resist and the beam voltage used. Smoothing λ effectively approximates erf(r) with a piece wise linear function, $P(r)$ as illustrated in Fig. 5. Obviously, the smaller the value of M, the more accurate the approximation.

The errors introduced by the smoothing process, performed by the template convolution of Eq. (3), can also be affected by the order of **A**. Although the effects of this error source are pattern dependent, we can utilize, without loss of generality, a figure of merit, ζ, defined as the ratio of the backscattered beam energy absorbed in the area [the filter area is $(MR_{\mathrm{sm}})^2$— see Eqs. (3) and (4)] covered by the filter to the total backscattered beam energy

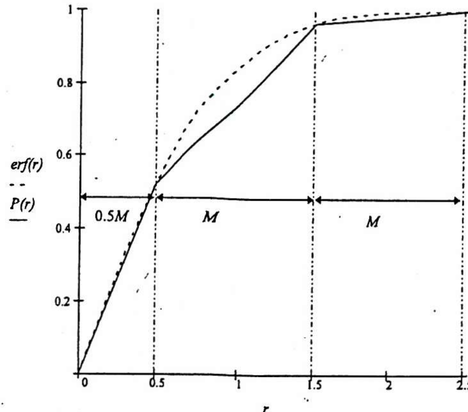


FIG. 5. Approximation of erf(r) by a piece wise linear function, $P(r)$.

$$\zeta=\frac{\int_{-0.5MR_{\mathrm{sm}}}^{0.5MR_{\mathrm{sm}}}\int_{-0.5MR_{\mathrm{sm}}}^{0.5MR_{\mathrm{sm}}}\eta/[\pi(1+\eta)\beta^2]\exp(-x^2-y^2/\beta^2)dx\,dy}{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\eta/[\pi(1+\eta)\beta^2]\exp(-x^2-y^2/\beta^2)dx\,dy}=\operatorname{erf}\left(\frac{MR_{\mathrm{sm}}}{2\beta}\right)^2.$$

This function is shown in Fig. 6. When $MR_{\mathrm{sm}}=4\beta$, ignoring the contribution outside the filter area in estimating E_b introduces a negligible error as $\zeta=0.990\,67$.

D. Neglect of forward scattering

This source of error arises from replacing the first term $[1/(\alpha^2)\exp(-[r/\alpha]^2)]$ in Eq. (1) with a delta term, $\delta(r)$. As the value of α is typically small (around 74 nm — see Sec. I), proximity effects are only noticeable within around 0.1 μm of an edge. The figure of merit introduced earlier is not useful in this case; if we attempt to analyze the errors due to forward scattered electrons alone, the integral of the first term in Eq. (1) at the edge can be taken as equal to 0.5 as long as the minimum feature size is several times larger than α. The effects of including the α term as a Gaussian on edges 2 and 3 in Fig. 1(a) are shown in Fig. 7. Although the inclusion does not affect the edge placement accuracy, it reduces shape edge contrast which is defined as the slope of a deposited energy density profile at shape edges.

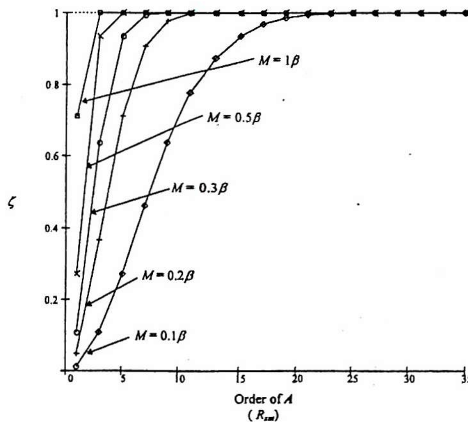


FIG. 6. Relative errors associated with varying the order of the convolution matrix.

E. Incident dose formula

In the derivation of the optimum incident dose formula [Eq. (2)], Murai *et al.* have assumed a uniform λ value. However, most practical circuit patterns have nonuniform λ values. Using λ^{sm} instead of λ in the formula is an attempt to take into account the nonuniformity in λ values. λ^{sm} includes energy deposited by electrons backscattered from shapes in nearby mesh sites, therefore, they depend also on the incident doses applied to those neighboring shapes. Murai *et al.* have assumed incident doses are the same for all mesh sites of any λ value in deriving the formula. Clearly, this assumption is no longer true after the correction. This is a form of recursive effect[9] in which correction of a shape affects how other previously corrected shapes should have been corrected. Thus, the calculation result becomes a function of the internal ordering of the calculation.

To construct the *exact* PADPEC solution, we first form a self-consistent set of linear equations using the deposited energy density, E_B, due to backscattered electrons and the total deposited energy density, E_{total}, at each subshape center point

$$E_{\text{total}}(r)=k\left[\frac{d_l(r)+\eta\Sigma_{l=1}^{N}[P(l,r)d_l(l)]}{1+\eta}\right],$$

$$E_B(r)=k\left[\frac{\eta\Sigma_{l=1}^{N}[P(l,r)d_l(l)]}{1+\eta}\right],$$

where N and $d_l(r)$ are the total number of subshapes and the incident dose of rth subshape, respectively. $P(l,r)$ is the fraction of energy of backscattered electrons deposited at l^{th} subshape center point due to a unitary incident dose applied to the rth subshape. The linear equation system is formed by taking the average of the above equations

$$E_m(r)=k\left[\frac{d_l(r)+2\eta\Sigma_{l=1}^{N}[P(l,r)d_l(l)]}{2(1+\eta)}\right].$$

E_m is set equal to the resist threshold energy density level. We can then solve the linear equations system for d_l using any standard linear equation algorithm such as Gauss–Jordan elimination.

Figures 8(a) and 8(b) show the simulated resultant deposited energy density profiles for the MPADPEC corrected pattern and the exact PADPEC scheme corrected pattern, respectively [Fig. 8(c) is discussed in Sec. V]. The corresponding edge errors of the six edges in the test pattern are shown in Table I. The negligibly small edge errors of the exact PADPEC corrected pattern arise from assuming a constant incident dose per subshape as discussed in Sec. III B. The large differences of edge errors at edges 1 and 4 between the two schemes implies that the saturated edge errors at small values of S in Figs. 4(a) and 4(d) are mainly due to the recursive effect.

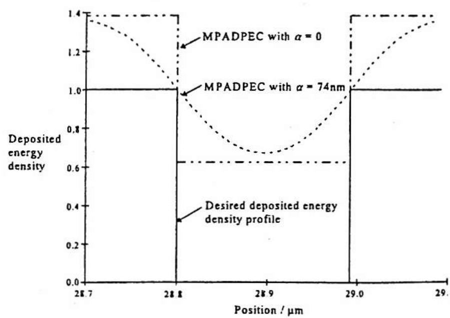


FIG. 7. Simulated resultant deposited energy density distributions around edges 2 and 3 of the test pattern [Fig. 1(a)] corrected using MPADPEC when $\alpha=0$ and $\alpha=74$ nm. Both M and S are set to 0.01β. The order of A is large enough to include the backscattered energy from the entire test pattern at any mesh site center point.

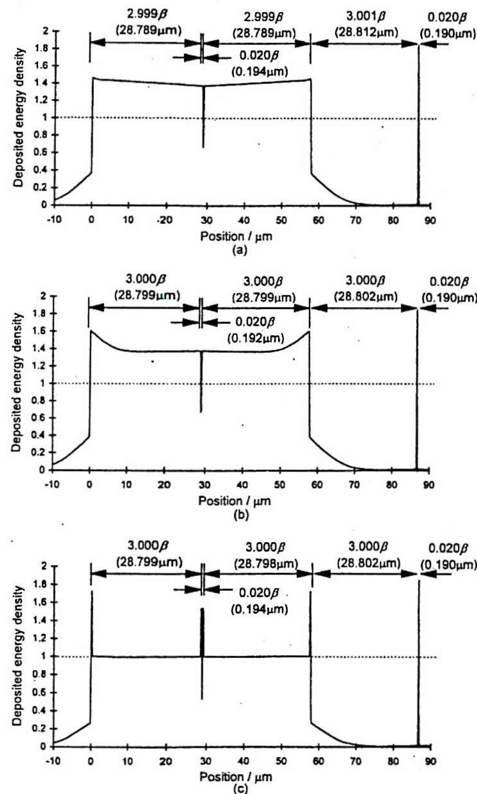


FIG. 8. Simulated resultant deposited energy density distributions for the test pattern [Fig. 1(a)] corrected using (a) MPADPEC, (b) the exact PADPEC method, and (c) EPADPEC. Both M and S are set to 0.01β. The order of A is large enough to include the backscattered energy from the entire test pattern at any mesh site center point.

TABLE I. Corresponding edge errors of the simulated deposited energy density profiles in Fig. 8. The values in brackets are the shrinkage (in nanometers) of the corresponding developed line from its intended position.

PEC scheme	Edge 1	Edge 2	Edge 3	Edge 4	Edge 5	Edge 6
MPADPEC	8.10×10^{-2}	1.60×10^{-4}	1.60×10^{-4}	8.10×10^{-2}	2.48×10^{-4}	2.48×10^{-4}
	(10)	(1)	(1)	(10)	(1)	(1)
the exact	2.20×10^{-3}	-9.18×10^{-5}	-9.18×10^{-5}	2.20×10^{-3}	2.40×10^{-4}	2.40×10^{-4}
PADPEC	(1)	(0)	(0)	(1)	(1)	(1)
EPADPEC	-1.42×10^{-3}	7.04×10^{-4}	7.30×10^{-4}	1.54×10^{-3}	2.40×10^{-4}	2.40×10^{-4}
	(0)	(1)	(1)	(1)	(1)	(1)

F. Conclusions

The errors of Fig. 4 indicate that errors due to interpolation (Sec. III A) approach a horizontal asymptote when $M \leq 0.3\beta$ for all six edges. Varying values of S (Sec. III B) also has a weak effect on the magnitude of the error, if $S \leq M$ and $M \leq 0.3\beta$, both are reasonable practical assumptions.

Figure 6 and Sec. III C indicate that for $M = 0.3\beta$, the error contribution caused by varying the order of A is less than 0.1% if the order of A is greater than 13. Neglecting forward scattering introduces no serious error as long as the minimum feature size is much greater than α.[10]

Finally, using an "exact" incident dose formula decreases the largest errors of Fig. 4 by a further order of magnitude. As the underlying mathematics indicates that the asymptotic error contribution from Secs. III A and III B should approach zero as M and S approach zero, we deduce that the dominant source of error in the overall process is the formulation of Eq. (2) itself. If significant gains are to be made, this is the area that should be attacked.

IV. EPADPEC METHOD

A. Overview of differences between the EPADPEC and the PADPEC schemes

The EPADPEC technique has two advantages; it corrects the recursive effect (Sec. III E) and increases shape edge contrast, giving a more accurate edge placement.

First, it eliminates the recursive effect by using an iterative algorithm to refine D_I. Instead of interpolating the λ^{sm} map for each rectangle as in the conventional PADPEC schemes, EPADPEC assigns D_I to each rectangle by interpolating the D_I values of the four nearest mesh sites to the rectangle center.

Second, EPADPEC includes a framing procedure that increases shape edge contrast. Each rectangle is partitioned into a peripheral part (called a frame) and a central rectangle as shown in Fig. 9. The details of the framing procedure are discussed in Sec. IV B 1. Figure 10 highlights the differences in correction procedures between the conventional PADPEC and the EPADPEC system.

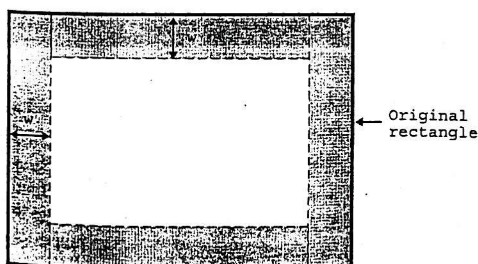


FIG. 9. Framing procedure for each rectangle. The shaded regions in the diagram are the frames.

B. Theoretical background

1. Framing

Figure 11 shows the normalized convolution, $G(x)$ between a normalized Gaussian distribution with a characteristic length of σ and an isolated semi-infinite shape that occupies the right half plane. The value of $G(x)$ at the shape edge is exactly half its maximum value and the slope at the shape edge is maximally steep. $G(x)$ is given by

$$G(x) = \tfrac{1}{2}(1 + \mathrm{erf}(x)),$$

where x is the distance from the shape edge. The value of $G(x)$ can be taken as 1 when x is greater than 4σ.

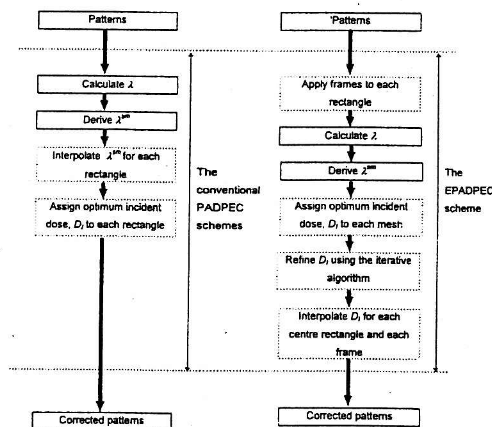


FIG. 10. Correction procedures in the conventional PADPEC and the EPADPEC scheme. The different procedures between the two schemes are shown in blocks with dotted boundaries.

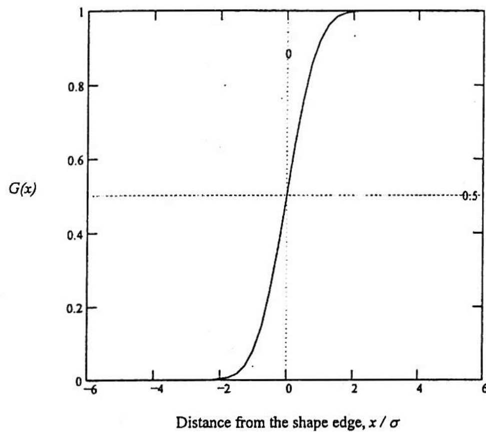


FIG. 11. Normalized convolution $G(x)$ between a normalized Gaussian distribution with a characteristic length of σ and an isolated semi-infinite shape that occupies the right half plane.

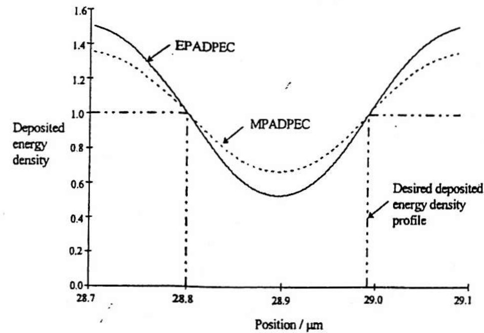


FIG. 12. Simulated resultant deposited energy density distributions at edges 2 and 3 of the test pattern [Fig. 1(a)] corrected using MPADPEC and EPADPEC. Both M and S are set to 0.01β. The order of $\mathbf{A}$ is large enough to include the backscattered energy from the entire test pattern at any mesh site center point.

EPADPEC uses the above property to simplify the evaluation of optimum incident doses for frames. For a maximally steep deposited energy density profile at shape edges, we need "infinitely" broad frames[11,12] with respect to a forward scattering characteristic length (α), and the optimum incident dose of these frames is equal to twice the incident dose of the center shape. In practice, a minimum frame width of 4α is sufficient to realize an infinitely broad frame. The higher incident dose of frames gives a higher proximity effect. EPADPEC takes into account the additional proximity effect by increasing the λ value of the frame by a factor of two when computing the λ map.

2. Iterative algorithm

Assuming the electron forward scattering is negligible and the value of E_b is uniform within a mesh site (i.e., the same assumption used in Sec. III), then the deposited energy density at nominally written points, $e_w(r)$ and nominally unwritten points, $e_u(r)$ in each mesh site, due to an incident dose distribution, $d_I(r)$ is given by

$$e_w(r)=k\left[\frac{d_I(r)+\eta\Sigma_{l=1}^{N}[a(l,r)\lambda(l)d_I(l)]}{1+\eta}\right] \tag{6}$$

and

$$e_u(r)=k\left[\frac{\eta\Sigma_{l=1}^{N}[a(l,r)\lambda(l)d_I(l)]}{1+\eta}\right],$$

respectively, where N, $d_I(r)$, and $\lambda(r)$ are the total number of mesh sites, the incident dose, and the λ value of rth mesh site, respectively. $a(l,r)$ is the same matrix as defined in Sec. II. The middle energy density level, $e_m(r)$, which is the average of the above equations is therefore

$$e_m(r)=k\left[\frac{d_I(r)+2\eta\Sigma_{l=1}^{N}[a(l,r)\lambda(l)d_I(l)]}{2(1+\eta)}\right].$$

Setting all the $e_w(r)$ to a desired constant value (similar to the self-consistency method of Parikh),[13] we can then apply a classical relaxation iteration method to solve the above equation for $d_I(r)$ as follows:

Jacobi method:

$$d_I^{(n)}(r)=\frac{-k\eta\Sigma_{l=1|l\neq r}^{N}a(l,r)\lambda(l)d_I^{(n-1)}(l)+(1+\eta)e_w(r)}{k[1+\eta a(r,r)\lambda(r)]};$$

Gauss-Seidel method:

$$d_I^{(n)}(r)=\frac{-k\eta\left[\Sigma_{l=1}^{r-1}a(l,r)\lambda(l)d_I^{(n)}(l)+\Sigma_{l=r+1}^{N}a(l,r)\lambda(l)d_I^{(n-1)}(l)\right]+(1+\eta)e_w(r)}{k[1+\eta a(r,r)\lambda(r)]},$$

where

$$d_I^{(0)}(r)=\frac{(1+\eta)e_w(r)}{k[1+\eta\Sigma_{l=1}^{N}a(l,r)\lambda(l)]}.$$

The superscripts in the equations denote the number of iterations. In the expression for the initial $d_I^{(0)}(r)$, the summation term is equivalent to λ^{sm} derived by using the template convolution. Although the expression assigns an incident dose of $(1+\eta)e_w$ to mesh sites with a λ value of zero, these doses have no effect in computing the optimum incident doses for other mesh sites. This is because contributions from other

mesh sites to $d_I(r)$ of rth mesh site are weighted by their λ values. For $\eta > 1$ and a sufficiently small value of M, the underlying matrix of Eq. (6) can become nondiagonally dominant. Under these conditions, the Jacobi method might fail to converge.

If we aim to equalize $e_m(r)$ as in the conventional PADPEC methods, then the above iterative algorithms become

Jacobi method:

$$d_I^{(n)}(r) = \frac{-k\left[\eta \sum_{l=1, l\neq r}^{N} a(l,r)\lambda(l) d_I^{(n-1)}(l)\right] + (1+\eta)e_w(r)}{k[0.5+\eta a(r,r)\lambda(r)]};$$

Gauss–Seidel method:

$$d_I^{(n)}(r) = \frac{-k\eta\left[\sum_{l=1}^{r-1} a(l,r)\lambda(l) d_I^{(n)}(l) + \sum_{l=r+1}^{N} a(l,r)\lambda(l) d_I^{(n-1)}(l)\right] + (1+\eta)e_w(r)}{k[0.5+\eta a(r,r)\lambda(r)]},$$

where

$$d_I^{(0)}(r) = \frac{2(1+\eta)e_w(r)}{k[1+2\eta\sum_{l=1}^{N} a(l,r)\lambda(l)]}.$$

In practice, when computing $d_I(r)$, the iterative algorithms use only mesh sites whose center points are within a distance of 2β from the rth mesh site center currently under evaluation. Experiments show that the Gauss–Seidel algorithm gives the best convergence rate, as would be expected.

V. SIMULATION RESULTS

Unless stated otherwise, EPADPEC uses a frame width of 4α and the Gauss–Seidel iterative method with one iteration.

Figure 8(c) shows the resultant deposited energy density profile of the test pattern in Fig. 1(a) when it is corrected by EPADPEC. The edge errors are shown in Table I. From the results in the table, EPADPEC has clearly made a significant improvement to the worst edge error of the previous algorithm. As the conventional PADPEC schemes aim to equalize $e_m(r)$ for all mesh sites, so large written shapes are overexposed as shown in Fig. 8(a). This over-dosing is significant, because it may reduce the throughput of an EBL machine. In contrast, EPADPEC aims to unify $e_w(r)$ for all mesh sites. Thus, EPADPEC eliminates the over-dosing problem on large shapes as in Fig. 8(c). Figure 12 shows that the EPADPEC corrected pattern has a steeper deposited energy density profile at shape edges as compared with the MPADPEC corrected pattern. Also the former gives a lower deposited energy density at the isolated space.

The same test pattern as reported by Kasuga *et al.*[4] (Fig. 13) is used for comparing the correction accuracy between the conventional PADPEC and the EPADPEC system. For cases without any PEC, the incident dose is set to $(1+\eta)$. The test pattern is fractured along a 1×1 μm^2 subfield grid. Because the test pattern is symmetric about the y axis, the deposited energy density profile on the right half plane will be the mirror image for the left half plane. Thus, it is sufficient to evaluate the linewidth deviation of the right half part of the monitor shape. The developed linewidth of the monitor shape is computed at a regular interval of 1.28 μm along the x axis, starting at 0.68 μm position.

Figure 14 shows the linewidth variation of the monitor shape without any PEC. The λ value of the dummy shapes is 75%. Figure 15 shows the monitor linewidth variation for the test pattern corrected by MPADPEC, KPADPEC, and EPADPEC with $M=5.12$ μm and dummy shapes with a λ value of 75%. The EPADPEC curve is produced with a single iteration; increasing the iteration count to two produces a slight improvement in the monitor linewidth inside the dummy shapes (Fig. 16) but none elsewhere; increasing

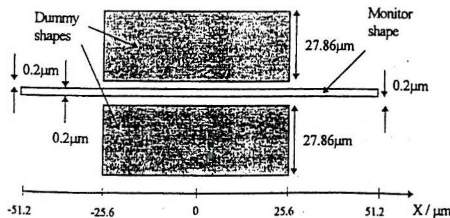


FIG. 13. Test pattern for comparing the correction accuracy of various PADPEC systems. The dummy shapes can be of various λ values.

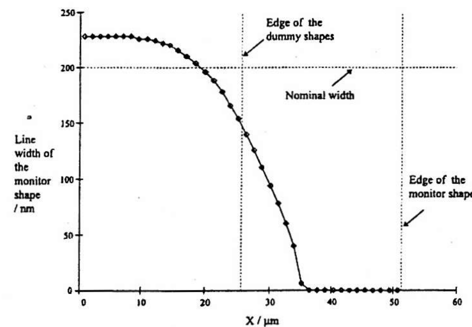


FIG. 14. Linewidth profile of the monitor shape without any PEC. The λ value of the dummy shapes is 75%.

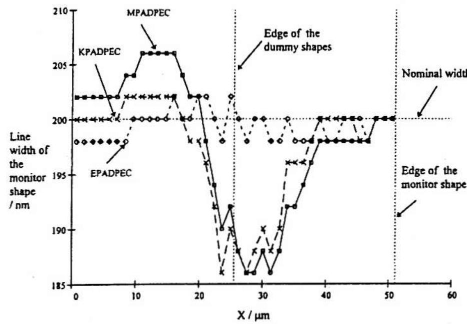


FIG. 15. Linewidth variations of the monitor shape for the test pattern corrected by MPADPEC, KPADPEC, and EPADPEC with $M=5.12$ μm and dummy shapes with a λ value of 75%. KPADPEC uses the optimum iteration count of 2.

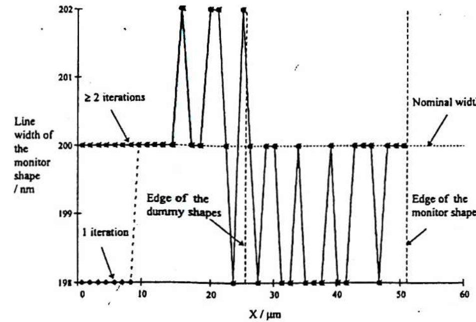


FIG. 16. Linewidth variations of the monitor shape for the test pattern corrected by EPADPEC with number of iterations for $M=5.12$ μm and dummy shapes with a λ value of 75%.

the iteration count still further produces no further improvement.

Table II summarizes the maximum magnitude of the monitor linewidth error (MLE), and magnitude sum of the monitor linewidth error (SLE), of the uncorrected test pattern and the test pattern corrected by various PADPEC schemes for various values of M and λ of dummy shapes. EPADPEC with one iteration improves MLE by factors of 11 (4) and 11 (4) over KPADPEC and MPADPEC, respectively, for $M=2.56$ μm and dummy shapes with $\lambda=100\%$ (50%). For $M=2.56$ μm/5.12 μm/10.24 μm and dummy shapes with $\lambda=75\%$, EPADPEC with one iteration reduces MLE by factors of 7/7/2.17 and 7/7/1.83 over KPADPEC and MPADPEC, respectively. The significant smaller improvement fac-

TABLE II. MLE and SLE of the uncorrected test pattern and the test pattern corrected by the various PADPEC schemes. The MLE values in brackets are the results reported by Kasuga *et al.* (Ref. 4).

λ of the dummy shapes (%)	Mesh size (μm)	PEC used in the simulation	Smoothing filter size ($R_{sm}\times R_{sm}$) (meshes)	MLE (nm)	SLE (nm)
50	···	Without PEC	···	200	4295
	2.56	MPADPEC	17×17	8	100
		KPADPEC after 10 iterations	3×3	8 (7)	90
		EPADPEC after 1 iteration	17×17	2	32
		EPADPEC after 2 iterations	17×17	2	20
		EPADPEC after 5 iterations	17×17	2	16
75	···	Without PEC	···	200	3808
	2.56	MPADPEC	17×17	14	186
		KPADPEC after 10 iterations	3×3	14 (13)	170
		EPADPEC after 1 iteration	17×17	2	38
		EPADPEC after 2 iterations	17×17	2	20
		EPADPEC after 5 iterations	17×17	2	16
100	···	Without PEC	···	200	4492
	2.56	MPADPEC	17×17	22	276
		KPADPEC after 10 iterations	3×3	22 (21)	250
		EPADPEC after 1 iteration	17×17	2	46
		EPADPEC after 2 iterations	17×17	2	22
		EPADPEC after 5 iterations	17×17	2	20
75	5.12	MPADPEC	9×9	14	202
		KPADPEC after 2 iterations	3×3	14 (13)	142
		EPADPEC after 1 iteration	9×9	2	42
		EPADPEC after 2 iterations	9×9	2	28
		EPADPEC after 5 iterations	9×9	2	28
	10.24	MPADPEC	5×5	22	416
		KPADPEC after 1 iteration	3×3	26 (27)	520
		EPADPEC after 1 iteration	5×5	12	182
		EPADPEC after 2 iterations	5×5	10	178
		EPADPEC after 5 iterations	5×5	10	178

tor of the $M = 10.24$ μm case is mainly due to the error in smoothing and interpolation steps as mentioned in Sec. III. Apart from the case of $M = 10.24$ μm and dummy shapes with $\lambda = 75\%$, additional iterations in EPADPEC do not improve MLE.

The processing time requirements for the KPADPEC and the EPADPEC curve of Fig. 15 (relative to the MPADPEC processing time) are 0.865 and 1.23, respectively. These figures are obtained from a conventional sequential computing environment; if we move to a parallel environment and replace the Gauss–Seidel algorithm with the Jacobi algorithm, we would expect to see comparable execution times for the MPADPEC and the EPADPEC profile (although there may be a problem of numerical instability).

VI. CONCLUSION

We have discussed and analyzed the sources of error in the conventional PADPEC systems. By isolating and attacking the dominant source of error in the PADPEC technique, the EPADPEC system produces more accurate correction results than the conventional systems. Simulation results show that EPADPEC reduces the maximum linewidth error by factors ranging from 4 to 11 for $M = 2.56$ μm. We have also demonstrated that EPADPEC (employing the Gauss–Seidel iterative solution technique) produces a satisfactory solution in just one iteration; further iterations produce no practical enhancement. The processing time overhead is a factor of 1.23; however, it is our intention to migrate the application to a special hardware platform to reduce this factor.

ACKNOWLEDGMENT

The authors would like to thank Dr. Takashi Kasuga for his help on the simulation details.

[1] A. N. Broers, A. C. F. Hoole, and J. M. Ryan, Microelectron. Eng. **32**, 131 (1996).
[2] F. Murai, H. Yoda, S. Okazaki, N. Saitou, and Y. Sakitani, J. Vac. Sci. Technol. B **10**, 3072 (1992).
[3] T. H. P. Chang, J. Vac. Sci. Technol. **12**, 1271 (1975).
[4] T. Kasuga, M. Konishi, T. Oda, and S. Moriya, J. Vac. Sci. Technol. B **14**, 3870 (1996).
[5] Y. Sakitani, H. Yoda, H. Todokoro, Y. Shibata, T. Yamazaki, K. Ohbitu, N. Saitou, S. Moriyama, S. Okazaki, G. Matuoka, F. Murai, and M. Okumura, J. Vac. Sci. Technol. B **10**, 2759 (1992).
[6] W. M. Wells, III, IEEE Trans. Pattern. Anal. Mach. Intell. **8**, 234 (1986).
[7] R. Andonie and E. Carai, Comput. Artificial Intell. **11**, 363 (1992).
[8] T. Nakasugi, T. Abe, and S. Yamasaki, Jpn. J. Appl. Phys., Part 1 **34**, 6644 (1995).
[9] S. Y. Lee, B. Liu, and B. D. Cook, Microelectron. Eng. **35**, 491 (1997).
[10] G. P. Watson, L. A. Fetter, and J. A. Liddle, J. Vac. Sci. Technol. B **15**, 2309 (1997).
[11] H. Eisenmann, T. Waas, and H. Hartmann, Proc. SPIE **2194**, 310 (1994).
[12] E. Kratschmer, J. Vac. Sci. Technol. **19**, 1264 (1981).
[13] M. Parikh, J. Appl. Phys. **50**, 4371 (1979).

Appendix B

The electron beam data preparation system (EDAPS)

Figure B-1 shows the overall architecture of EDAPS. The system is written in C++ language, using Microsoft Visual C++ Developer Studio version 4.0. Currently, the PEC module contains all the variants of the PADPEC method (EPADPEC, KPADPEC and MPADPEC) and the target EBL machine of the system is the Leica EBMF series EBL machine which uses the Gaussian spot beam and step-and-write stage movement technology. Besides the PEC module, the system includes several auxiliary tools to facilitate this research. The tools are:

- A GDSII layout browser for viewing circuit layout in the GDSII format.
- A shape processor that converts shapes into rectangles and removes overlapped parts among shapes. The processor uses the scan line algorithm to perform the above operations [158].
- A proximity effect simulator for visualising the proximity effect for a given incident dose distribution.
- An EBL machine write time simulator that estimates the write time of a Leica EBMF 10.5 EBL machine for a given pattern.
- Viewers for browsing exposure patterns and result files of the shape processor and the proximity effect simulators.

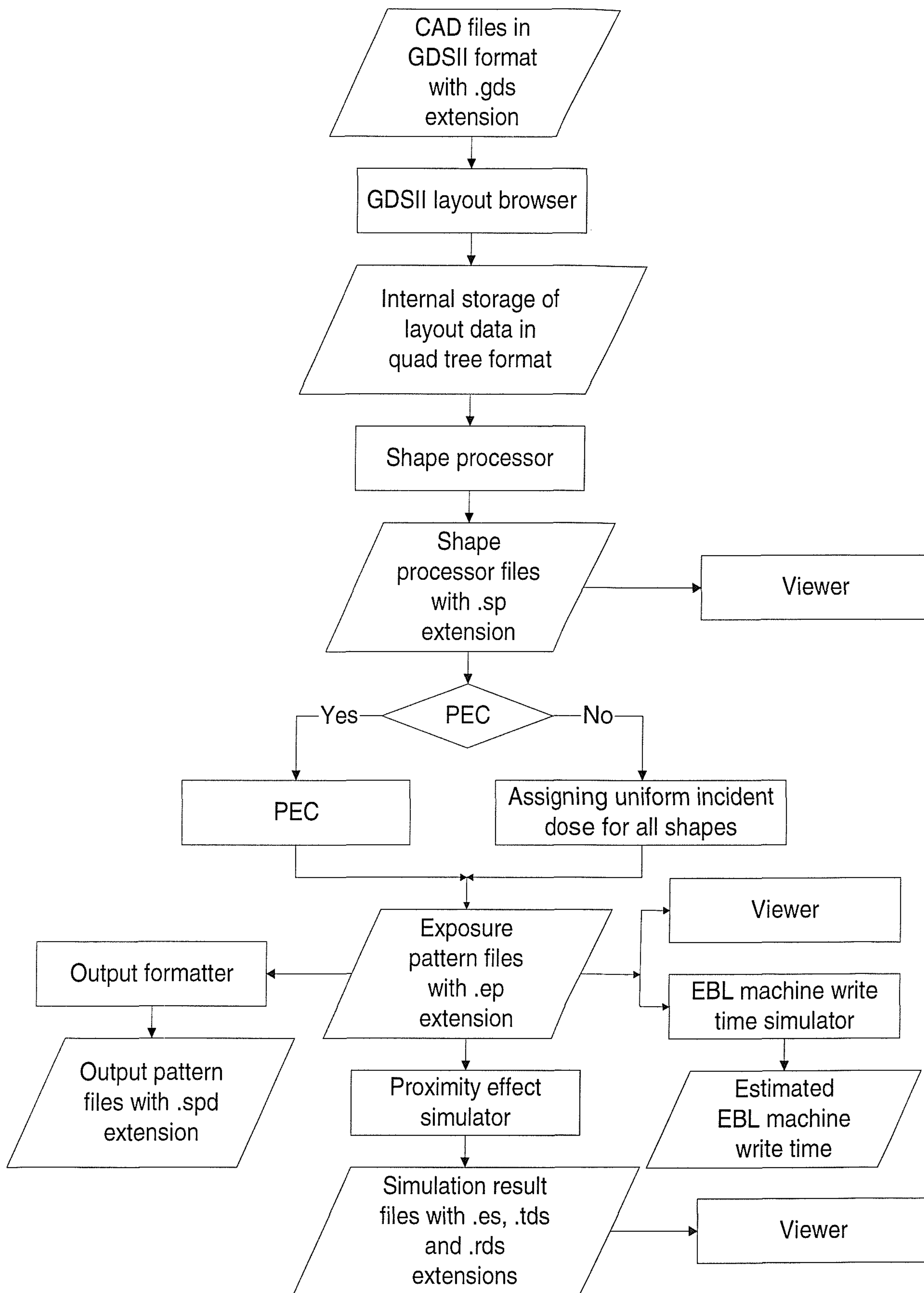


Figure B-1 The overall architecture of EDAPS.

- An output formatter that converts exposure patterns from the internal format to Cambridge source pattern data language format [156]. To drive Leica EBMF EBL machines, it is necessary to convert the Cambridge source format into the EBMF binary format using the converter software provided by Leica [157].

B.1 The PEC module

Figure B-2 shows the architecture of the PEC module. The fracturing algorithms and the corner correction scheme are detailed in Chapter 7. For KPADPEC and MPADPEC with the physical fracturing algorithm, the λ^{sm} map is converted to the D_I (incident dose) map using equation (4-12). Thus, the incident dose of each fractured rectangle is interpolated from the D_I map instead of the λ^{sm} map.

From the results in sections 6.2.2.1 and 6.3.2, the physical width of the convolution matrix, $\boldsymbol{A}$ should be at least 3.7β and the correction time increases with the order of $\boldsymbol{A}$. Thus, to minimize the correction time, EPADPEC and MPADPEC use the minimum order of $\boldsymbol{A}$ that meets the minimum physical width requirement. On the other hand, KPADPEC uses an order of $\boldsymbol{A}$ of 3 and the optimum iteration count as calculated from equation (4-16).

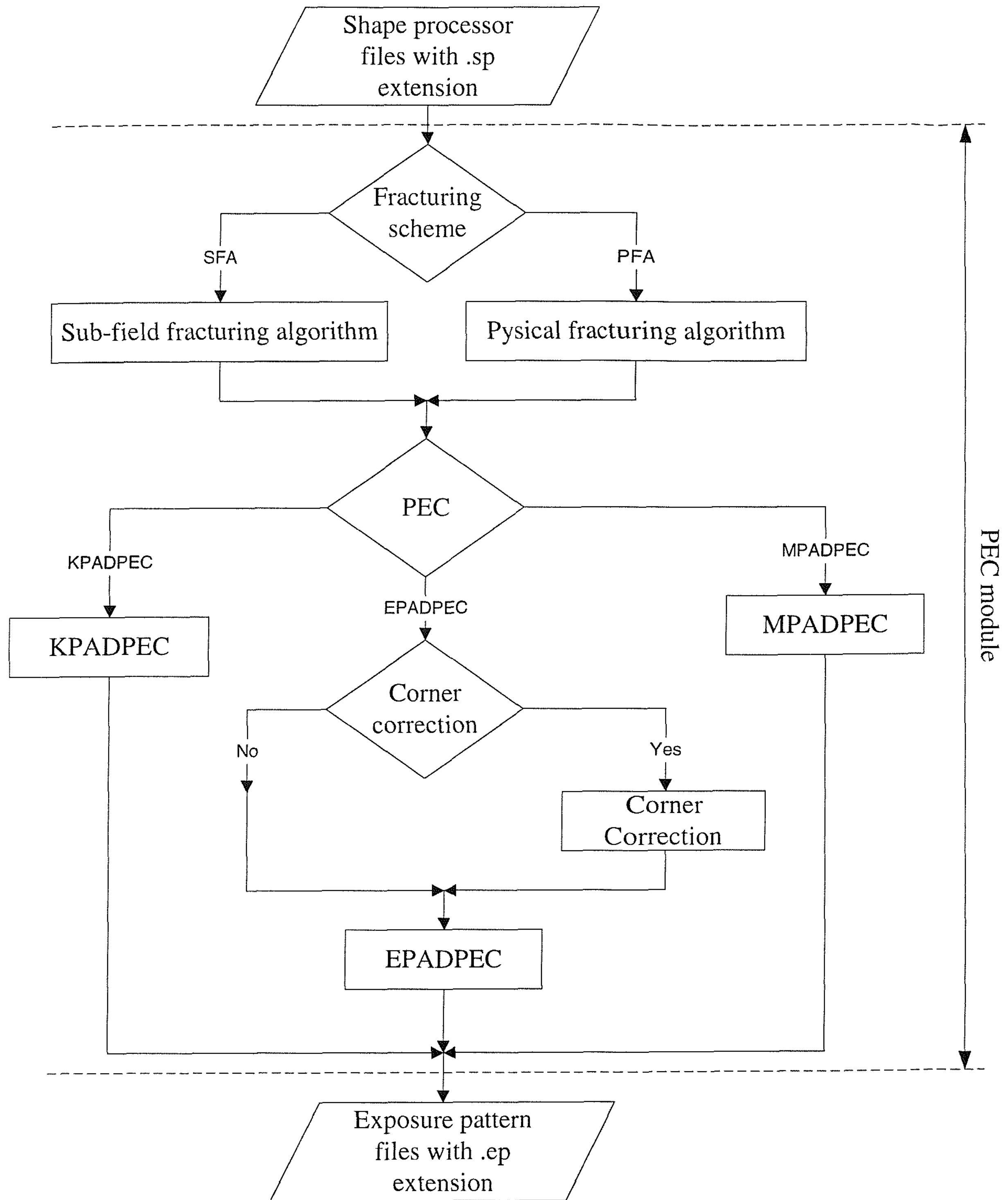


Figure B-2 The architecture of the PEC module.

B.2 GDSII layout browser

The GDSII layout browser enables the user to view a circuit layout stored in the GDSII format. Besides the standard Windows scroll bars facility, the browser provides the following features to assist the user in browsing the circuit layout:

- Zooming.
- Layer visibility selection. A circuit layout usually consists of several layers. This facility allows the user to view the layout of particular layers. The browser assigns a different colour for each layer.
- Display grid. The user can alter the display grid sizes and toggle it on or off.
- Layout display mode. The browser provides two layout display modes, which are filled and outline mode. In the former mode, the browser displays shapes as filled polygons, while it outlines the shapes in the latter mode.

B.2.1 Algorithm implementation

The GDSII format has two basic components: *structures* and *elements*. A structure is a collection of elements and that may also contain instances of other structures, recursion is not allowed. There are seven kinds of elements:

3. A BOX that defines a rectangular geometry;
4. A BOUNDARY that defines a filled polygon;
5. A PATH that defines a wire;
6. A TEXT that defines a text string;
7. A NODE that defines an electrical path;
8. A STRUCTURE REFERENCE (SREF) that refers to a structure;
9. An ARRAY REFERENCE (AREF) that defines an array of structures formed from a referred structure.

The browser represents these elements as C++ classes with a hierarchy structure as shown in Figure B-3.

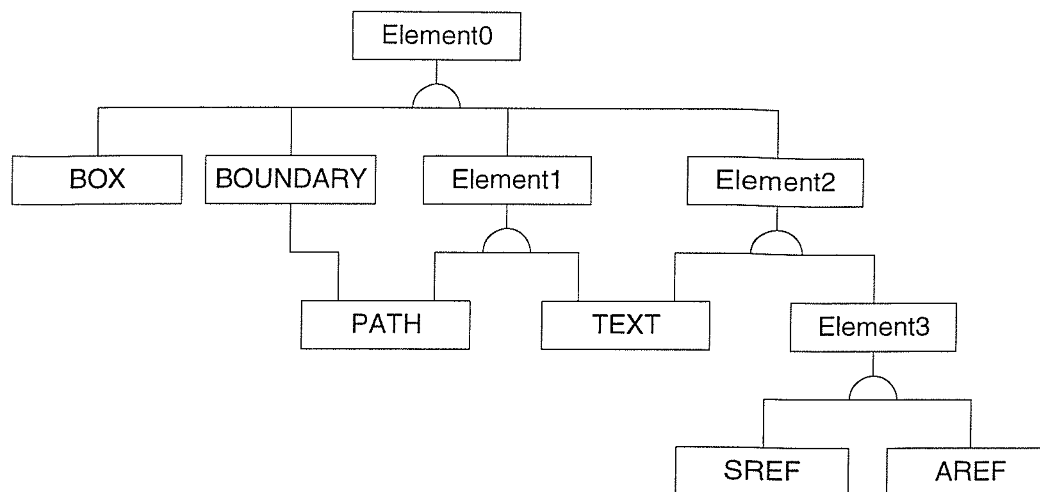
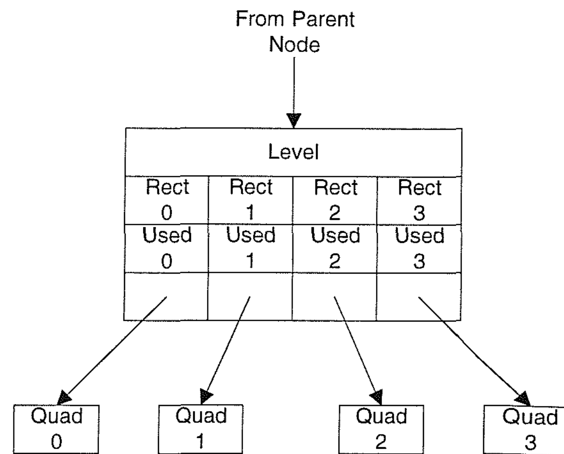
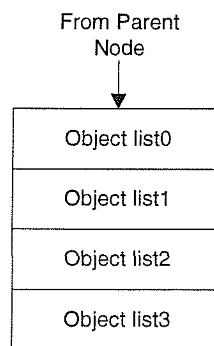


Figure B-3 Hierarchy chart of C++ classes used to model the GDSII elements.

To improve the browsing performance, the browser organises elements in each structure of a circuit layout in an adaptive quad list quad tree [159, 160]. Figure B-4 shows a tree node and a tree leaf of the quad tree. A *quad* is either a tree node or a leaf node. When a *quad* is a leaf node, the corresponding *used* field value denotes the number of objects contained in it; otherwise the *used* field value is negative, indicating the quad is further divided. The *rect* field stores the shadow co-ordinates of its associate *quad.* Each leaf node contains four lists of pointers to element objects. To simplify the construction of the quad tree, element shadows are used instead of the actual space occupied by the elements. The algorithms for retrieving data from the quad tree are discussed in detail in [159, 160].



(a) A tree node



(b) A leaf node

Figure B-4 Data elements.

B.3 Proximity effect simulator

Figure B-5 shows the architecture of the proximity effect simulator. The exposure simulator employs the double Gaussian EDF as the exposure model for estimating the deposited energy density distribution in a pattern for a given incident dose distribution. The threshold developer simulator uses the threshold energy density model as the developer model to predict the developed resist image, while the resist characteristic curve developer simulator employs a resist characteristic curve to estimate the remaining resist thickness based on the deposited energy density. In all the simulators, the user specifies the co-ordinates of the simulation window and the sample grid sizes used in the simulation.

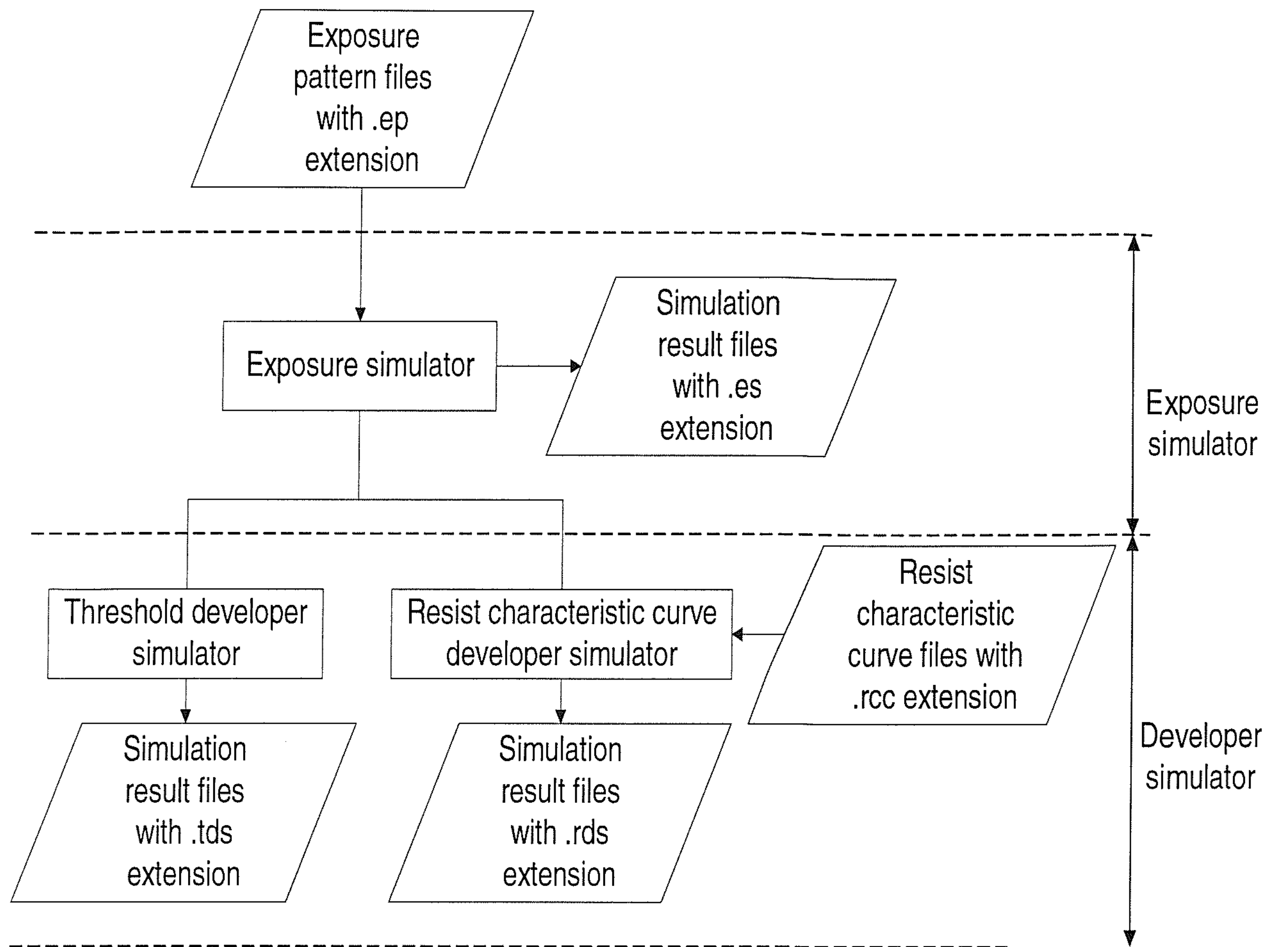


Figure B-5 The architecture of the proximity effect simulator.

B.3.1 Theoretical background

From section 3.2, the deposited energy density distribution, $e_R(r)$ is given by equation (3-2):

$$e_R(r) = k\int_{-\infty}^{\infty} f(r-\lambda)d_I(\lambda)d\lambda$$

Since the input shapes are restricted to rectangles only and each rectangle is assigned a constant incident dose, so $e_R(r)$ becomes the sum of the convolutions between the double Gaussian EDF and the rectangles. Using the convolution property of a Gaussian term with a rectangle (Appendix E.4), $e_R(r)$ becomes:

$$e_R(x,y)=\frac{k}{4(1+\eta)}\sum_{\text{All rectangles, i}} d_I(i)\left[err(x,y,L_i,T_i,R_i,B_i,\alpha)+\eta err(x,y,L_i,T_i,R_i,B_i,\beta)\right]$$

in the Cartesian co-ordinate system, where L_i, T_i, R_i and B_i are the left, top, right and bottom co-ordinates of the ith rectangle respectively, $L_i< R_i$, $T_i< B_i$ and

$$err(x,y,L,T,R,B,b)=\left\{erf\left(\frac{R-x}{b}\right)-erf\left(\frac{L-x}{b}\right)\right\}\left\{erf\left(\frac{B-y}{b}\right)-erf\left(\frac{T-y}{b}\right)\right\}$$

For a Gaussian spot, step and write EBL machine, $e_R(x,y)$ is given by:

$$e_R(x,y)=k\sum_{\text{All rectangles, i}} d_I(i)\left[\left\{\sum_{j=L_i-x}^{R_i-x} f(j)\right\}\left\{\sum_{k=T_i-y}^{B_i-y} f(k)\right\}\right] \qquad \textbf{(B-1)}$$

where $f(j)$ is a line spread double Gaussian EDF (i.e. an one-dimensional EDF) and the separable property of a Gaussian term (Appendix E.1) has been used to simplify the calculation.

B.3.2 Algorithm implementation

As the evaluation of the error function and the inner summations in equation (B-1) are time consuming, they are implemented as memory access to lookup tables, which hold pre-computed values of the function or the summation. For the summation, the lookup table is accessed by the upper limit of the summation and each element value is the summation from zero to the upper limit:

$$Lookup(u)=\sum_{i=0}^{u} f(i)$$

Thus, the evaluation of each inner summation in equation (B-1) needs to access the lookup table twice:

$$\sum_{i=l}^{u} f(i)=Lookup(u)-Lookup(l-1)$$

The tables cover the function/summation values for a distance from zero to the proximity range, R (This is referred as *Half width of the coverage square* in Appendix C.3.2.2). For distances greater than R from the evaluation point, the

function/summation values are taken to be equal to the value at R. The error of this table truncation is at worst equal to:

$$1-\frac{1}{1+\eta}\left[erf\left(\frac{R}{\alpha}\right)^2+\eta erf\left(\frac{R}{\beta}\right)^2\right]$$

For β=9.6μm, α=74nm, η=0.83 and R=2.5β, the error is about 0.04% which is negligible. Thus, 2.5β is the default value for R.

In the exposure simulator, the deposited energy density profile of a pattern is stored in a two dimensional array of *double* (64-bit data elements). For the threshold developer simulator and a positive tone resist, the resist at a sample point is either intact (undeveloped) or none left (developed). Thus, we can represent the developed resist image as a two dimensional binary bitmap. However, as a circuit layout can be very big, so it is necessary to store the image in a compact form. Instead of storing the individual value at each sample point, only the beginning and end co-ordinates of a continuous block of developed sample points in a row are stored. (Run Length Limited – RLL – compression) The data structure is shown schematically in Figure B-6.

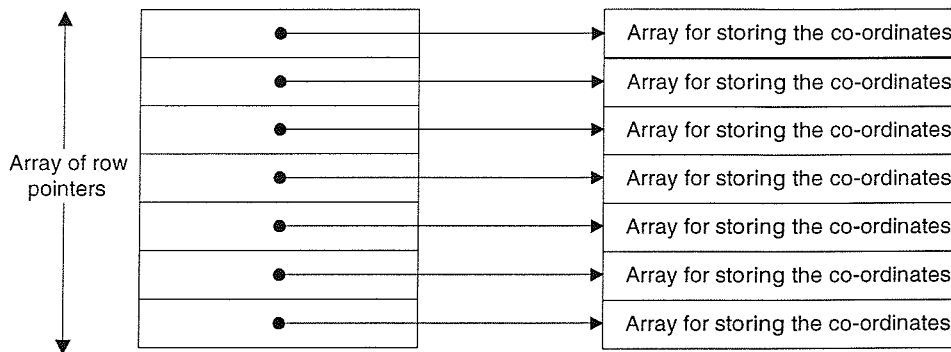


Figure B-6 Data structure for storing the developed resist image in the threshold developer simulator.

Representative points along a resist characteristic curve are stored in files with a *.rcc* extension. A maximum of 256 points is allowed per curve. Thus, the remaining resist thickness image is stored in a two dimensional array of byte (8-bit data elements). The

remaining resist thickness is quantized to the nearest lower value as shown in Figure B-7.

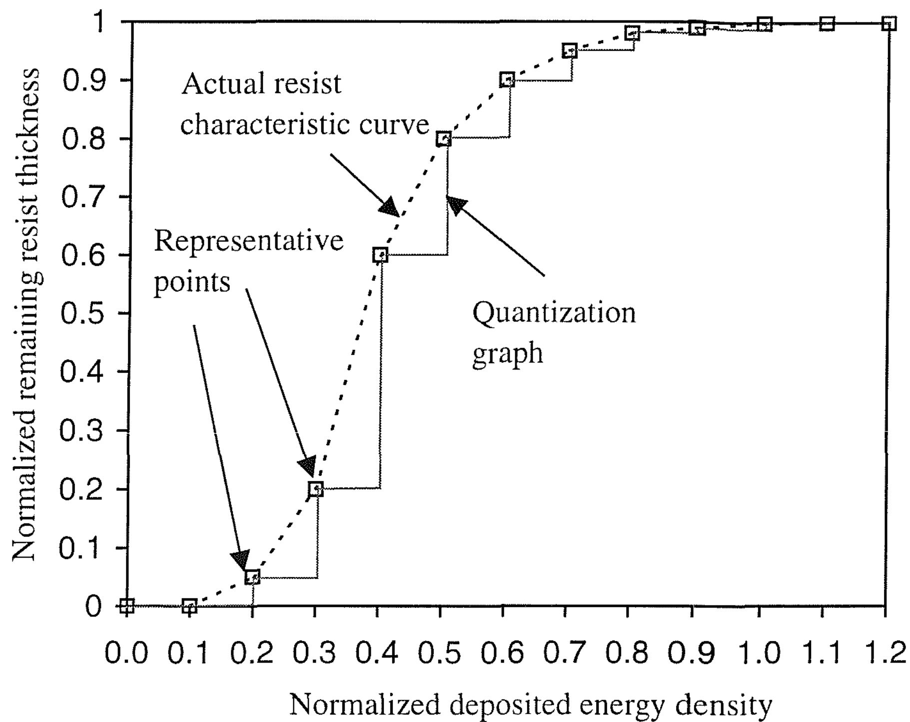


Figure B-7 Quantization of a resist characteristic curve.

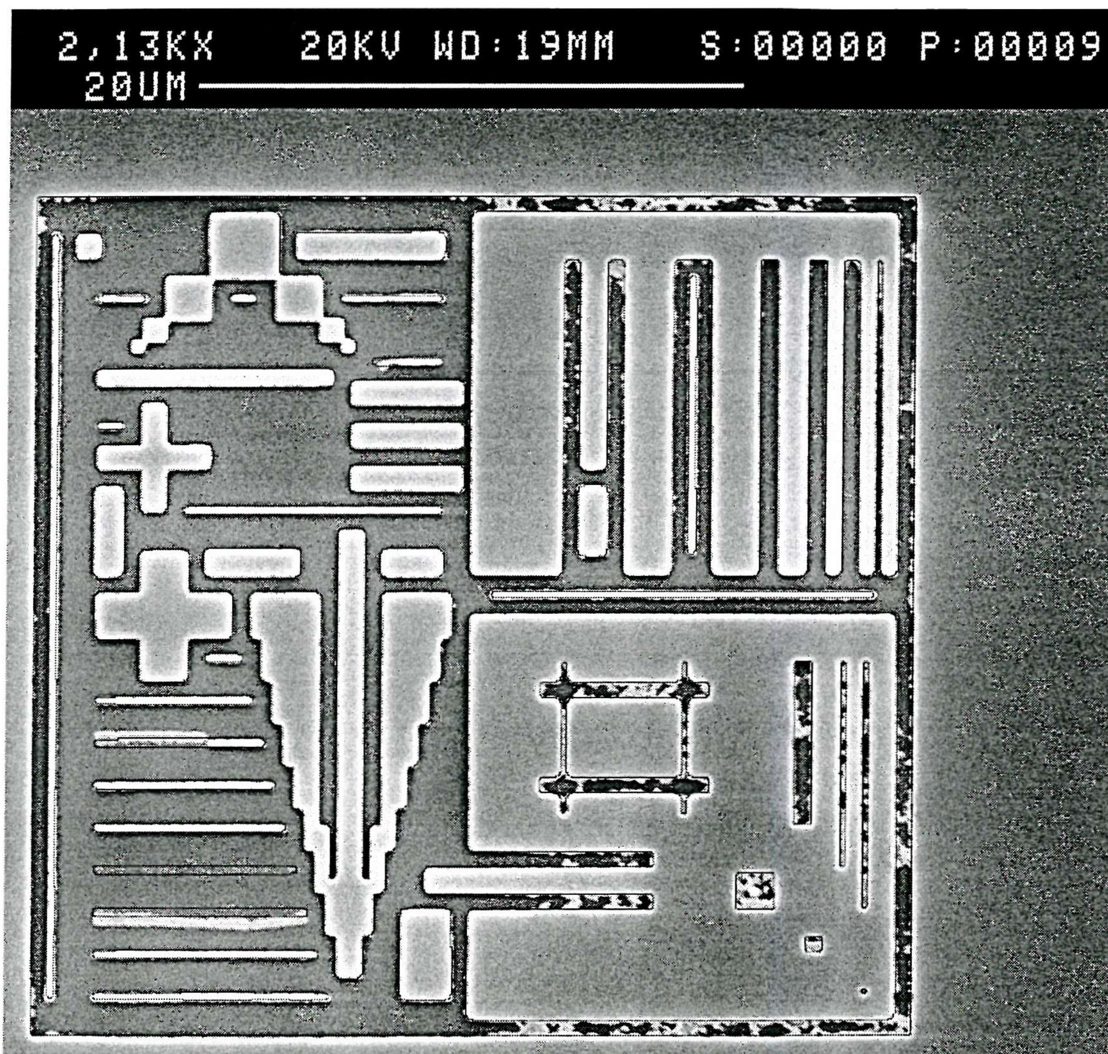
B.3.3 Calibration using scanning electron microscope photographs of developed resist image

Figure B-8 shows good agreement between the simulation and the experimental results. Some of the fine lines at the lower left corner of the photograph have collapsed due to insufficient mechanical strength of the fine resist lines. The test pattern, RauTP is fabricated under the same lithographic conditions as in section 6.2.2.2 except for an additional delay of two days between the pre-bake and the exposure.

Using the method described in [131], the standard clock frequency is measured to be 1.635 MHz on the same wafer in which RauTP (in Figure B-8) was fabricated. The standard clock frequency is mapped to the constant incident dose, D_0 required to achieve the resist threshold energy density value, E_T for an infinite written region, i.e.

$$kD_0 \int_{-\infty}^{\infty} f(r-\lambda)d\lambda = E_T$$

The measured standard clock frequency value is about 50% smaller than the applied value (2.506 MHz) used in the fabrication, causing the developed pattern to be under-developed (as clock frequency is inversely proportional to dose). The normalized resist threshold energy density value used in the simulation is given by the applied standard clock frequency divided by the measured standard clock frequency. Comparing the photographs in Figure B-8 and Figure 6-16(a) suggests that the increase in the resist threshold energy density value between the runs might be due to the additional delay between the pre-bake and the exposure.



The simulated edges (black lines) are superimposed on top of the photograph.

Figure B-8 Scanning electron microscope photograph of RauTP corrected with EPADPEC.

In an ideal case, the simulation will provide information similar to that derived from a scanning electron microscope photograph. The simulation relies on a user to provide an accurate value of the resist threshold energy density (i.e. the standard clock frequency) that takes into account such process influences as delay times, pre-bake, post-bake, resist development and thermal effects. The simulation does not, however take into account the resist mechanical strength (or lack thereof) that might result in collapse of fine features in the developed resist image as shown in Figure B-8.

B.4 EBL machine write time simulator

The EBL machine write time simulator estimates the write time of an exposure pattern on a Leica EBMF 10.5 EBL machine. The simulator computes the write time, t_{Write} in seconds according to the following analytic formula:

$$t_{Write} = \text{ChipCount}\left[\text{TheoreticalWriteTime} + a.\text{FieldCount} + b.\text{ShapeCount} + c\right] + d \tag{B-2}$$

Here, a, b, c and d are constants while the rest of the terms are variables. *ChipCount* is the number of repetition of an exposure pattern (chip) written by the machine. The terms in the bracket are the write time for a single pattern (chip) and d represents the overhead time for the machine set up, calibration and self-testing. *ShapeCount* is the total number of sub-shapes after shape fracturing along the sub-field grid of the EBL machine. The machine has a fixed grid size of 1024 exels. As the file size of an exposure pattern in the EBMF binary format is proportional to the shape counts in the pattern, so b represents the overhead time per shape for loading the file (shape data) onto the machine. *FieldCount* is the number of field sites occupied by an exposure pattern and the machine has a fixed field size of 2^{15} exels. Thus, a represents the average time for the machine to move from one field site to another site. c represents the machine set up time for writing a chip (pattern).

TheoreticalWriteTime is the minimum achievable write time and is equal to the write time of an ideal EBL machine, which is the sum of the write time of each shape exposed at its designated clock frequency. It is assumed that the shape assigned with the minimum normalized incident dose, $D_I(min)$ uses the standard clock frequency in Hz, *StandardClockFrequency*. Thus, *TheoreticalWriteTime* is computed as follows:

$$\text{TheoreticalWriteTime} = \sum_{\text{For each shape in a pattern, } i} \frac{\text{ExelCount(i)}}{D_I(\text{min})} \frac{D_I(i)}{\text{StandardClockFrequency}} \frac{1}{4^{VRU}}$$

where *ExelCount(i)* is the total number of exels enclosed by i^{th} shape and $D_I(i)$ is the normalized incident dose of ith shape. *VRU* determines the resolution of the EBL machine. Every 2 to the power of *VRU* exel is exposed. Thus, a VRU value of 0 will expose every exel, a *VRU* value of 1 will expose every 2nd exel, a *VRU* value of 2 will expose every 4th exel and a *VRU* value of 3 will expose every 8th exel.

Fitting the model (equation (B-2)) to experimental machine write times on various patterns using linear regression [161], values of a=0.297s, b=2.79×10^{-4}s, c=25.5s and d=256s are obtained. Figure B-9 shows close agreement between the simulated and the experimental machine write times on various patterns. The simulator uses a=0.297s, b=2.79×10^{-4}s, c=25.5s and d=256s.

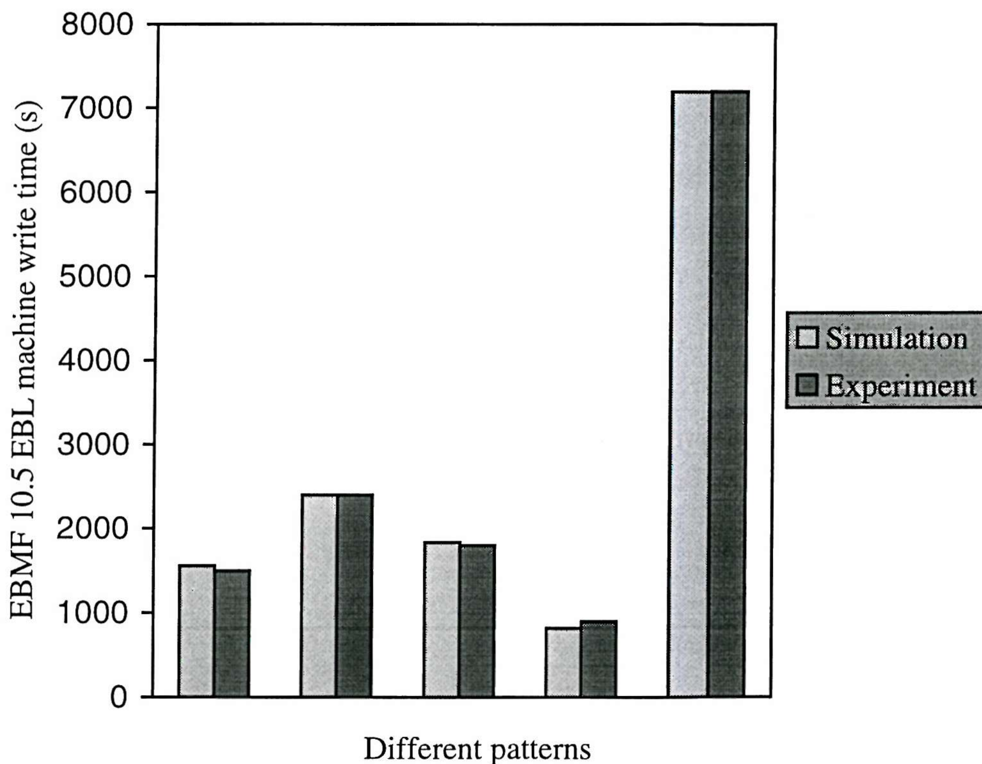


Figure B-9 Experimental and simulated EBMF 10.5 EBL machine write times.

Appendix C

The user manual of the electron beam data preparation system (EDAPS)

This is a stand-alone document, separate from the rest of the thesis.

C.1 Introduction

A computer aided design (CAD) system usually codes pattern data in a relatively high level format such as the GDSII stream format which is the industry standard format for interchanging integrated circuit design between CAD systems. The format supports multi-vertex polygons, repetitive structures and hierarchy. To drive an electron beam lithography (EBL) machine directly, the data needs to be converted into a low-level format, which is native to the machine. Besides hierarchy unwrapping in the format conversion, the data might need significant extra processing such as overlap removal and proximity effect correction (PEC) that compensates for the proximity effect caused by electron scattering. PEC is the most numerically intensive step and is a well-established step in the data processing chain from a CAD workstation to an EBL machine.

EDAPS is a software package for performing the above format conversion. It is designed to run on Windows NT platforms. At the moment, it supports only Manhattan shapes (polygons) in the GDSII source and its target EBL machine is the Leica EBMF series EBL machine that uses the Gaussian spot beam and the step-and-write stage movement technologies. It produces output patterns in the Cambridge source pattern data language format [156]. To drive Leica EBMF EBL machines, it is necessary to convert the Cambridge source format into the EBMF binary format using the converter software provided by Leica [157].

The PEC module of EDAPS contains all the variants of the pattern area density proximity effect correction (PADPEC) methods (enhanced PADPEC (EPADPEC), Kasuga's PADPEC (KPADPEC) and Murai's PADPEC (MPADPEC)). Besides the PEC module, the system also includes the following auxiliary tools:

- A GDSII layout browser for viewing circuit layout in the GDSII format.
- A shape processor that converts shapes into rectangles and removes overlapped parts among shapes.
- A proximity effect simulator for visualising the proximity effect for a given incident dose distribution.
- An EBL machine write time simulator that estimates the write time of a Leica EBMF 10.5 EBL machine for a given pattern.
- Viewers for browsing exposure patterns and result files of the shape processor and the proximity effect simulators.

Figure C-1 shows the overall architecture of EDAPS.

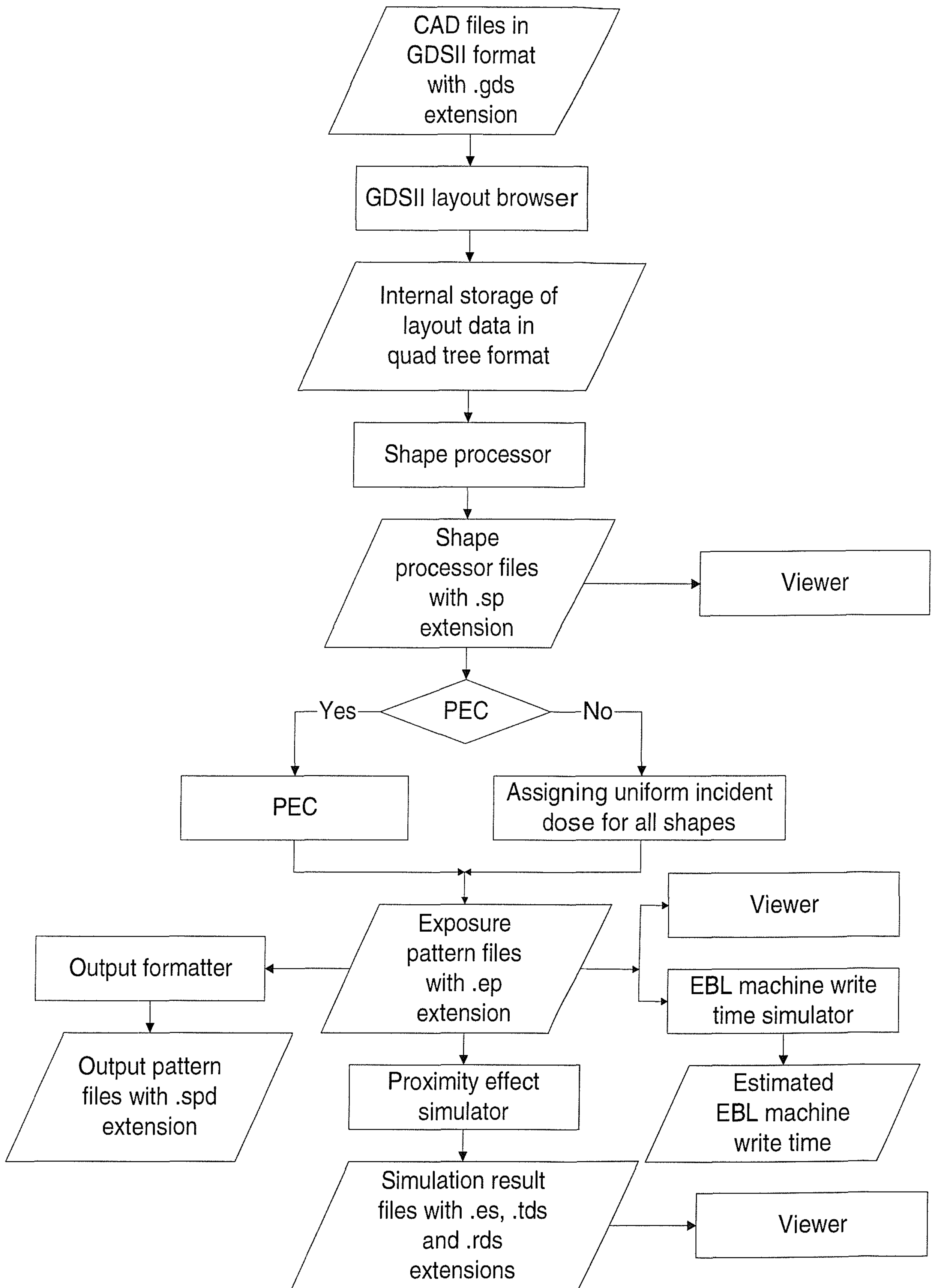


Figure C-1 The overall architecture of EDAPS.

C.1.1 Conventions used in this manual

Names of menus, commands and pushbuttons on toolbars are in bold. To distinguish commands with the same name in different menus, the menu name is prefixed to the command name and the names are joined by a '|' character, e.g. **File|New** refers to the new command in the file menu.

C.1.2 Getting started

Just like any other Windows NT applications, EDAPS can be started in the normal way. Once started, the main user interface (Figure C-2) appears on the screen. Until a file has been loaded, the client area in the main window remains empty.

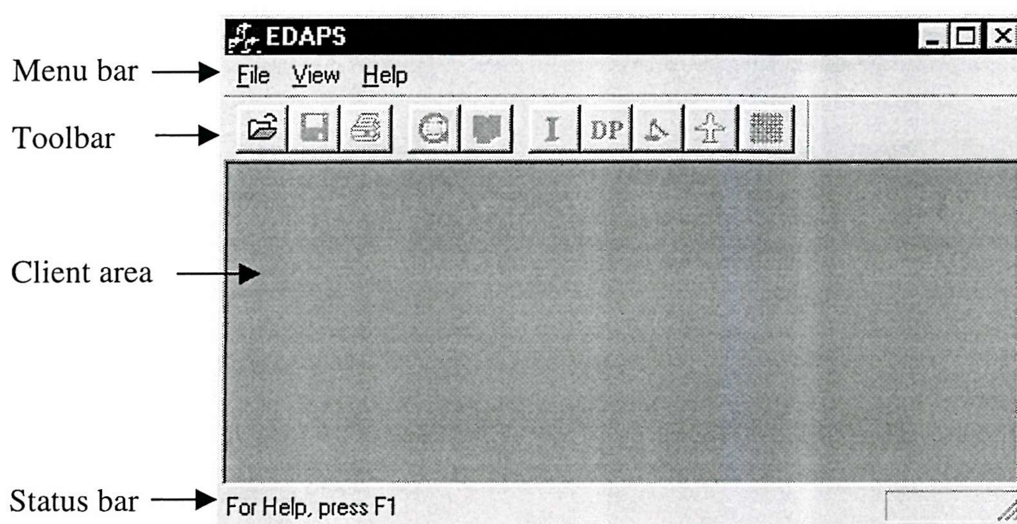


Figure C-2 Main user interface of EDAPS.

There are ten pushbuttons in the toolbar (Figure C-3):

1. **Open** – this corresponds to the **File|Open** command. (See section C.2.1.1 for further details).
2. **Save** – this corresponds to the **File|Save As** command and is disabled for the GDSII layout browser. (See section C.2.1.3 for further details).
3. **Print** – this corresponds to the **File|Print** command. (See section C.2.1.4 for further details).
4. **ZoomByArea** - this corresponds to the **View|Zoom By Area** command. (See section C.2.2.5 for further details).

5. **LayerSelection** – this is applicable only to GDSII layout browser windows and corresponds to the **View|Select Layer Visibility** command (See section C.3.1.2 for further details).
6. **Isotropic** – this acts as a toggle switch for the layout-mapping mode. When the pushbutton appears to be *pushed*, the mapping mode is *MM_ISOTROPIC*. Otherwise, the mapping mode is *MM_ANISOTROPIC* (See section C.2.3.2 for further details).
7. **DesiredPattern** – this is not available in GDSII layout browser and shape processor windows and corresponds to the **View|Desired Pattern** command (See section C.6.1 for further details).
8. **Fill** – this is applicable only to GDSII layout browser and shape processor windows and corresponds to the **View|Fill Drawing Objects** command (See section C.3.1.1 for further details).
9. **AerialViewer** – this corresponds to the **View|Aerial Viewer** command (See section C.2.2.3 for further details).
10. **Grid** – this corresponds to the **View|Grid** command (See section C.2.2.4 for further details).

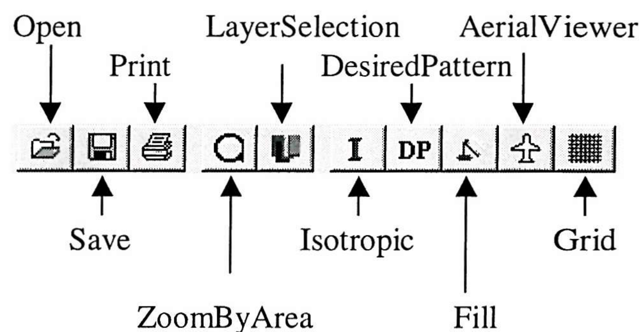


Figure C-3 Toolbar of EDAPS.

C.2 Standard menu for various viewers

This section describes the common menu commands in all the viewers.

C.2.1 The File menu

This menu provides the standard functions for opening, closing, saving, printing and terminating the viewer.

C.2.1.1 The Open command

This command issues a standard *Windows 95 Open File* dialog box. EDAPS supports input file formats with the following file extensions:

- gds (The GDSII format);
- tds (The threshold developer simulator data files);
- rds (The resist characteristic curve developer simulator data files);
- es (The exposure simulator data files);
- ep (The exposure pattern data files);
- sp (The shape processor data files);

C.2.1.2 The Close command

This command closes the currently selected viewer window.

C.2.1.3 The Save As command

This command issues a standard *Windows 95 Save As* dialog box to allow the user to save the pattern in the currently selected viewer window. If the specified file name is already exist, then a *confirmation* dialog box will prompt the user to confirm overwriting the existing file. This command is not available to the GDSII layout browser.

C.2.1.4 The Print command

This command brings out a standard Windows 95 *Print* dialog box (Figure C-4). The dialog box allows the user to:

- change the default printer by selecting a different printer from the *dropdown combo*.
- change the selected printer properties by clicking on the **Properties** pushbutton.
- set the number of layout copies to be printed.

Upon left clicking the **OK** pushbutton, EDAPS prints the layout in the client area of the currently selected viewer window to the selected printer.

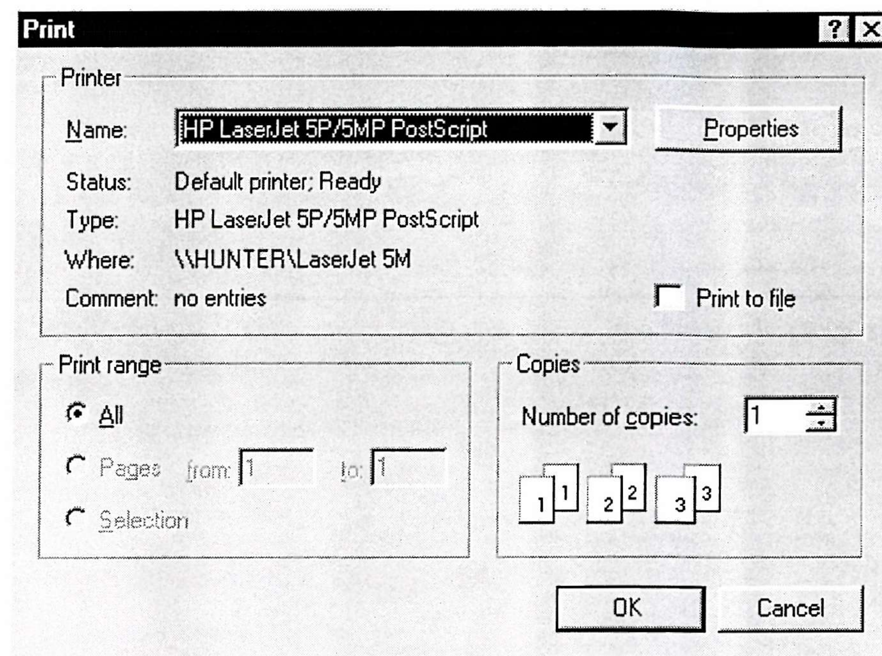


Figure C-4 Standard Windows 95 Print dialog box.

C.2.1.5 The Print Preview command

This command changes the viewer window into a *Print Preview* window as shown in Figure C-5. The file name of the displayed layout and the printing date are displayed on the upper left and right corner respectively of the page. There are three different levels of zooming settings and the user can change between the settings by pressing the **Zoom In** and **Zoom Out** pushbuttons. The user can switch back to the normal viewer window by pressing the **Close** pushbutton.

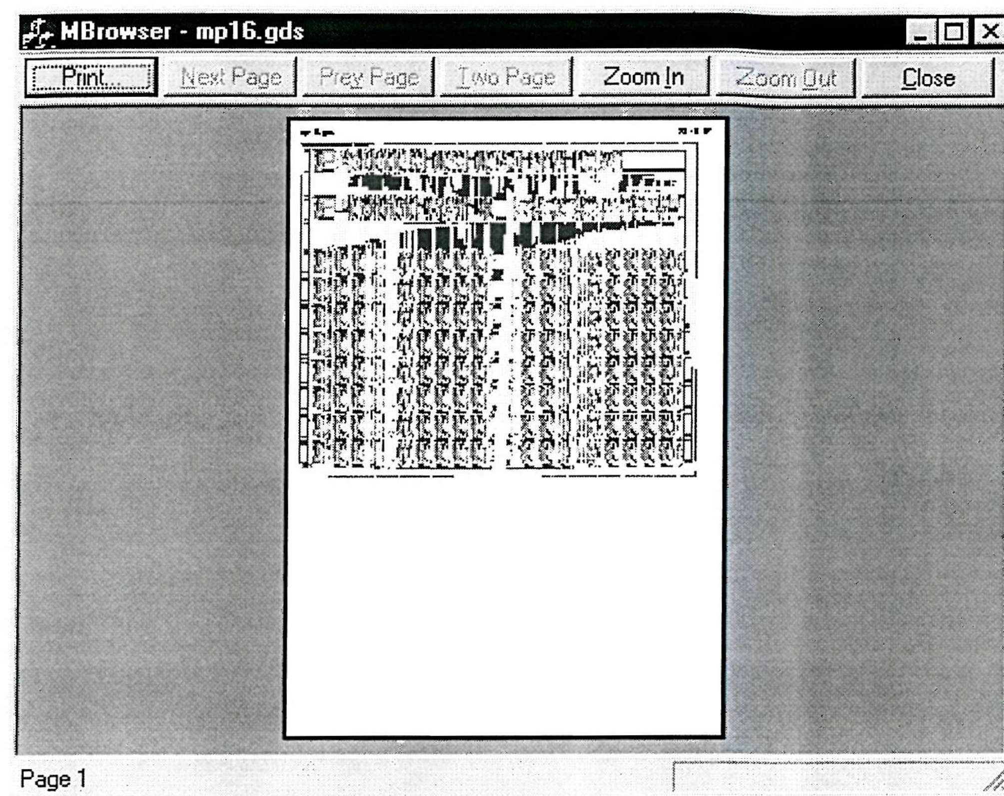


Figure C-5 Print Preview window.

C.2.1.6 The Print Setup command

This command issues a standard Windows 95 *Print Setup* dialog box (Figure C-6). The dialog box allows the user to:

- change the default printer by selecting a different printer from the *dropdown combo*.
- alter the selected printer properties by clicking on the **Properties** pushbutton.
- modify the paper size and source used by the selected printer.
- change the printing layout orientation.

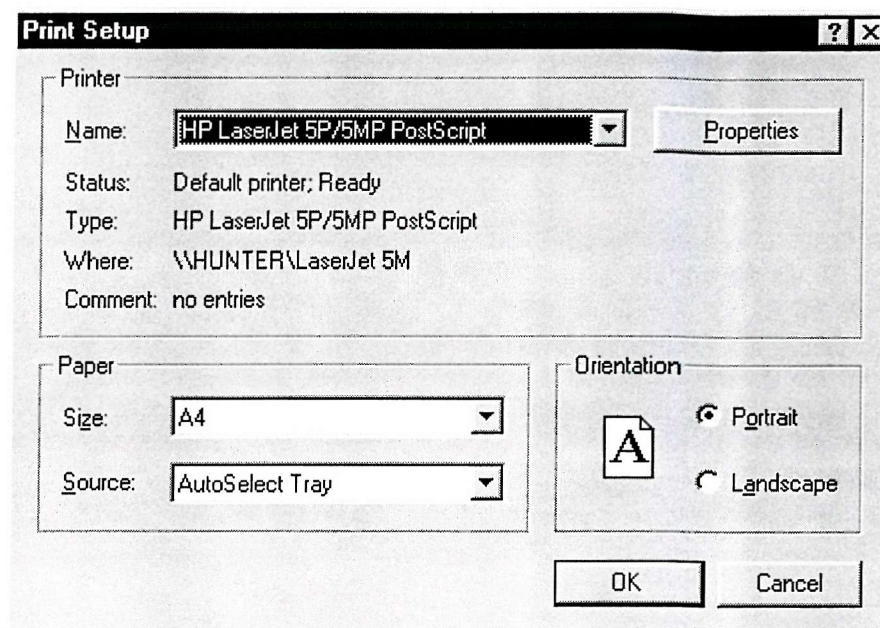


Figure C-6 Standard Windows 95 Print Setup dialog box.

C.2.1.7 The Exit command

This command terminates EDAPS.

C.2.2 The View menu

All viewer windows have zooming facilities. The user can zoom in a particular area of a layout using the **View|Zoom By Area** command (section C.2.2.5). A tool window called Aerial Viewer (Figure C-9 on page 238) keeps track on all the user's zoomed viewing areas of a layout. These areas are displayed as rectangles in Aerial Viewer. The current viewing area is shown as a yellow rectangle, while the rest of the rectangles are shown in white colour. The user can go back to previous viewing area by clicking inside its corresponding rectangle in Aerial Viewer. The user can also view other area of the layout with the same zoom setting by dragging the rectangle to the area of interest.

C.2.2.1 The Toolbar command

This command toggles the display of the toolbar (Figure C-3). When the toolbar is displayed, a tick mark is shown next to the command in the menu.

C.2.2.2 The Status Bar command

This command toggles the display of the status bar (Figure C-2). When the status bar is displayed, a tick mark is shown next to the command in the menu.

C.2.2.3 The Aerial Viewer command

This command toggles the display of Aerial Viewer. When Aerial Viewer is displayed, a tick mark is shown next to the command in the menu and the **AerialViewer** pushbutton in the toolbar appears as a pushed icon.

C.2.2.4 The Grid command

This command toggles the display of the grid lines in the *client* window. When the grid is on, a tick mark is shown next to the command in the menu and the **Grid** pushbutton in the toolbar appears as a pushed icon.

C.2.2.5 The Zoom By Area command

This command enables the user to zoom in the area of interest in the client window. The user select an area by holding the left mouse button down while dragging out the area. A warning message is issued if the selected area is less than 1 database square. The user can select an area for zooming in the client window of either Aerial Viewer or the viewer window.

C.2.3 The Settings menu

This menu lets the user change the settings of the displayed grid size and the displayed mapping mode. Upon left clicking the menu, a Settings property sheet (Figure C-7) is displayed.

C.2.3.1 The Grid Size property page

Figure C-7 shows the Grid Size property page. The user can change the display grid size by modifying the values in the *edit* boxes. The grid sizes must be integers and the default grid size is 100 database units in both dimensions. The page also displays the layout dimensions in the currently selected viewer window.

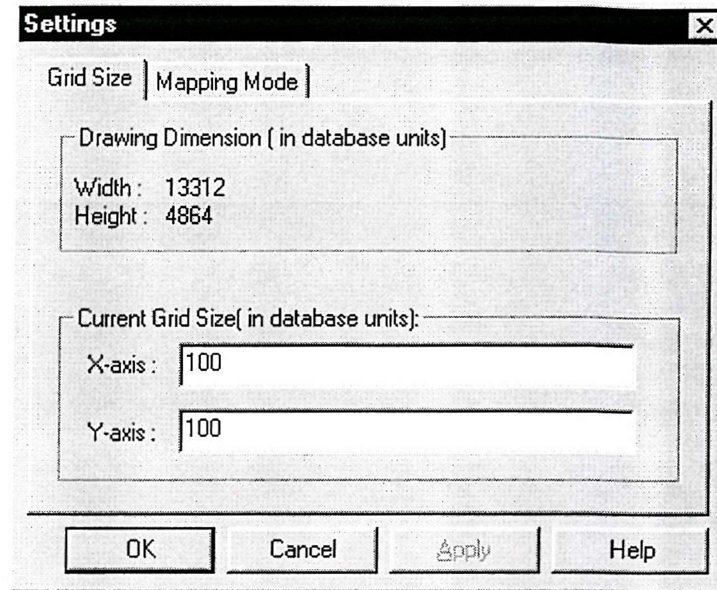


Figure C-7 Grid Size property page.

C.2.3.2 The Mapping Mode property page

There are two modes for mapping the database unit of a layout to the *device* unit of a display or printer. The *MM_ISOTROPIC* mode ensures a 1:1 aspect ratio, which is useful when it is important to preserve the exact shape of an image. The *MM_ANISOTROPIC* mode allows the x- and y-coordinates to be adjusted independently. The default mapping mode is *MM_ISOTROPIC.* The desired mapping mode can be selected by left clicking on the appropriate radio button in the Mapping Mode property page (Figure C-8).

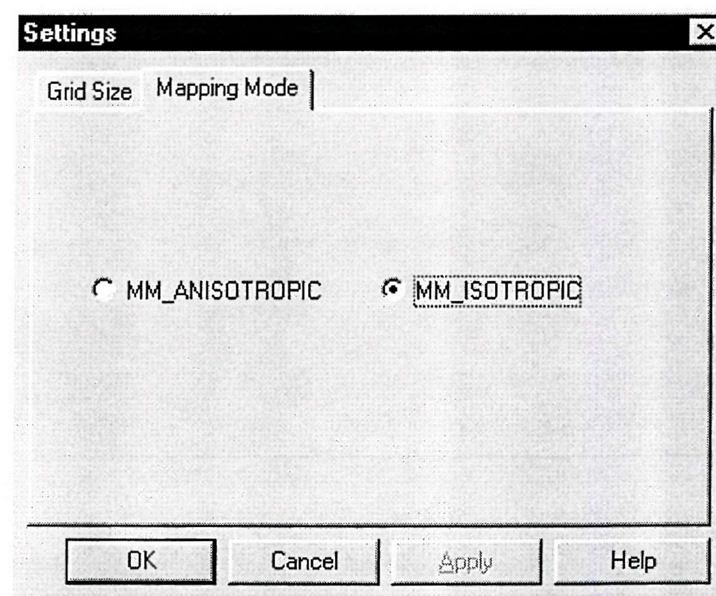


Figure C-8 Mapping Mode property page.

C.2.4 The Window menu

This menu provides commands for creating new viewer windows and functions for arranged all the displayed viewer windows in the client area of the main window.

C.2.4.1 The New Window command

This command creates an additional window for the currently selected viewer window.

C.2.4.2 The Cascade command

This command arranges all the non-*minimized* viewer windows in a *cascade* fashion.

C.2.4.3 The Tile command

This command arranges all the non-minimized viewer windows in a *tile* fashion.

C.2.4.4 The Arrange Icons command

This command arranges all the minimized viewer windows abut to one another at the lower part of the client area of the main window.

C.3 GDSII layout browser

Figure C-9 shows the user interface of the GDSII layout browser. The browser displays the co-ordinates (in database units) of the mouse position and the size of a database unit (in nanometres) in its status bar. The browser tracks the co-ordinates of the mouse position if the position is within its client area.

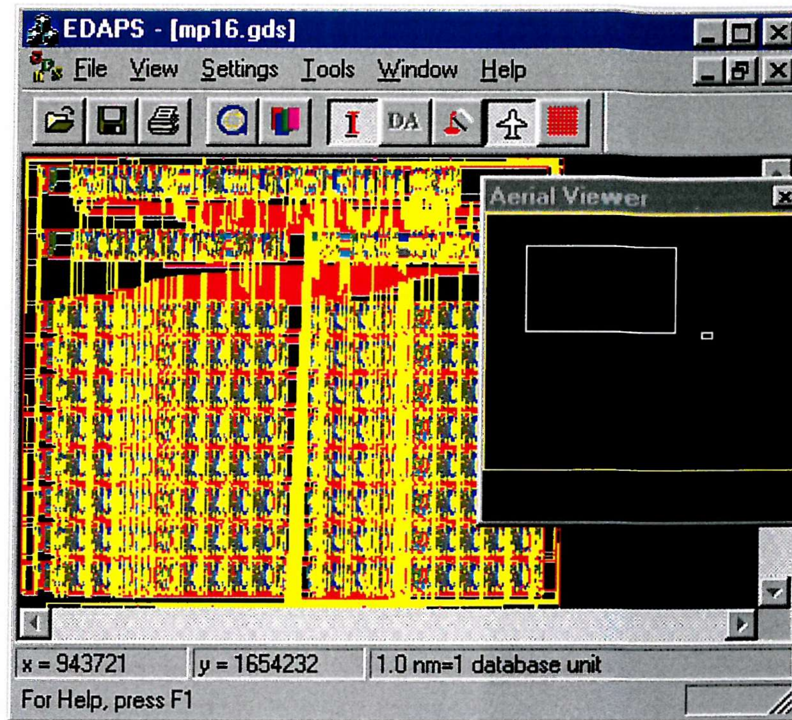


Figure C-9 User interface of the GDSII layout browser.

Whenever a new GDSII layout browser window is created either through the **File|Open** command or the **Window|New** command, it is possible to select any *GDSII structures* in the selected GDSII file for display in the new window through a Cell Selection dialog box (Figure C-10). The tree list in the dialog box represents the hierarchical structure of the structures. The name at each tree node corresponds to the name of the structure. A structure can be selected or de-selected by left clicking on the button next to the tree node. When a structure is selected, all the referred structures at the lower hierarchy of the selected (*parent*) structure are selected as well. It is not possible to de-select any referred structure of a selected parent structure. A *tick* button in a tree node indicates that the corresponding structure is selected. By left clicking on the **Select All** and the **Deselect All** pushbutton, it is possible to select and deselect all the structures respectively.

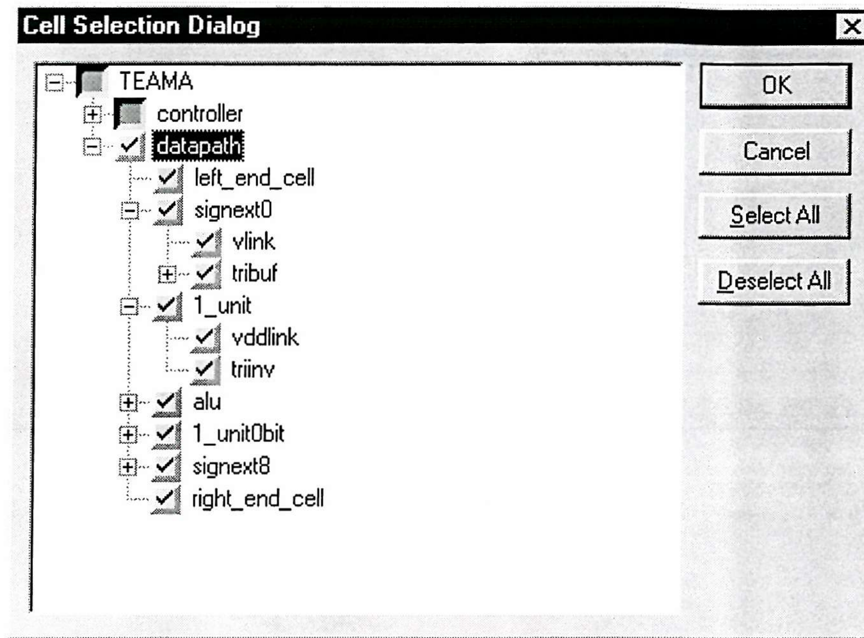


Figure C-10 Cell Selection dialog box.

C.3.1 The View menu

Besides the standard **View** commands, the GDSII layout browser provides additional commands to improve its browsing capabilities.

C.3.1.1 The Fill Drawing Objects command

The GDSII layout browser draws shapes in a layout either as outline or filled polygons. The Fill Drawing Objects command toggles between these drawing modes. When the current drawing mode is the filled polygon mode, a tick mark is shown next to the command in the menu and the **Fill** pushbutton in the toolbar appears as a pushed icon.

C.3.1.2 The Select Layer Visibility command

This command issues a Layer Visibility Selection dialog box (Figure C-11). The dialog box lists all the layers in the layout of the currently selected GDSII layout browser window. The colour of the rectangle besides the layer number is the colour used for drawing shapes on the layer in the GDSII layout browser window. It is possible to select and de-select any layers by left clicking on the button next to the colour rectangle of the layer number. A tick button besides a layer number indicates

that the layer is selected. By left clicking on the **Display All Layers** pushbutton and the **Hide All Layers** pushbutton, it is possible to select and de-select all the layers respectively.

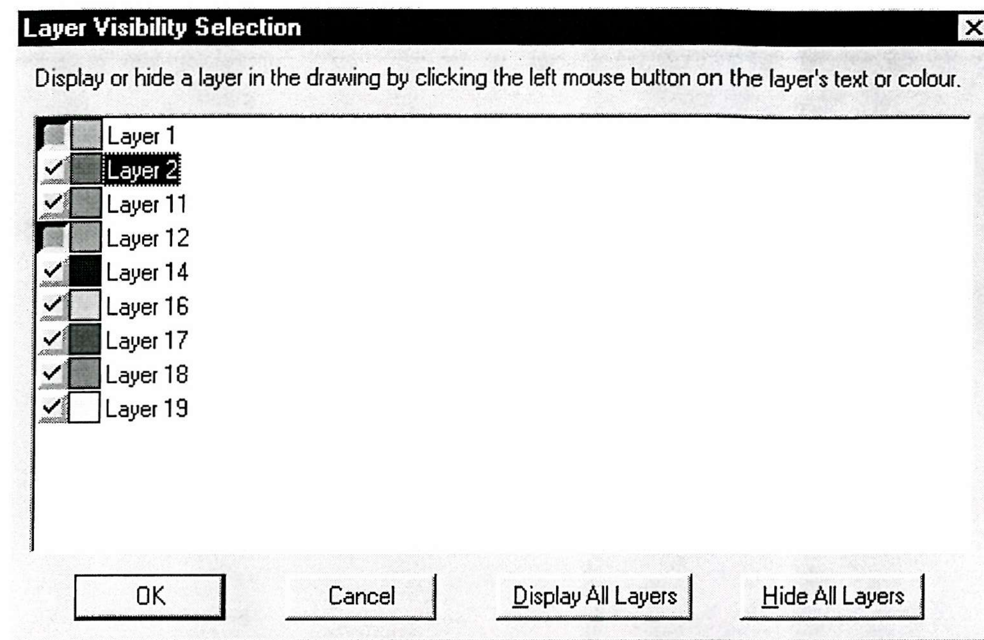


Figure C-11 Layer Visibility Selection dialog box.

C.3.2 The Settings menu

Besides the standard **Settings** commands, the GDSII layout browser provides additional property pages for setting the parameters of the deposited energy density function (EDF) and the EBL machine.

C.3.2.1 The EBL property page

The target EBL machine can be selected from the dropdown combo in the property page (Figure C-12). So far, Leica EBMF 10.5 and non-Gaussian spot, step and write machines are supported. The settings of the selected EBL machine can be changed by left clicking on the **Option** pushbutton.

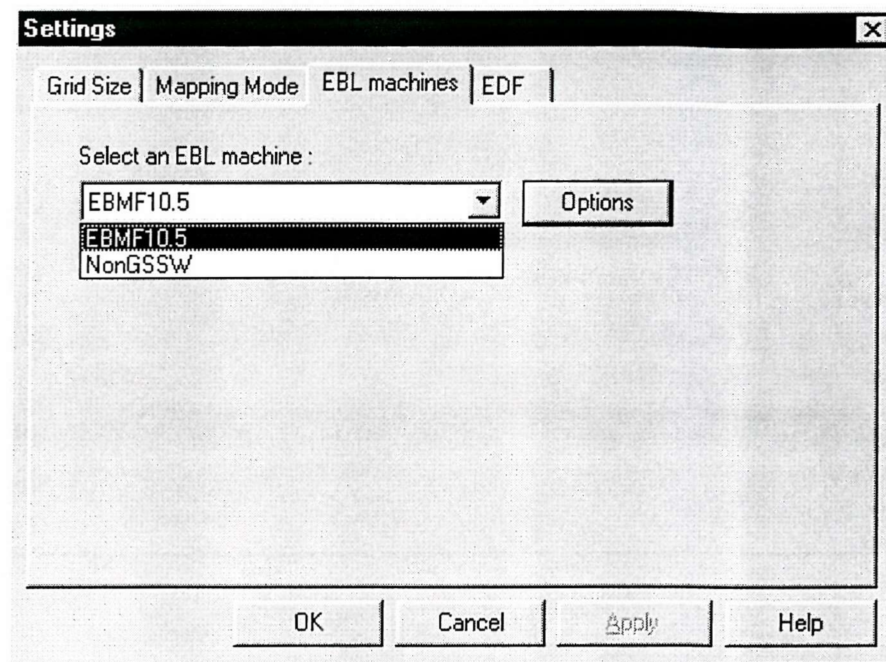


Figure C-12 EBL Machine property page.

Upon pushing the **Option** pushbutton, the appropriate option dialog box of the selected EBL machine will be displayed. The dialog box of the Leica EBMF 10.5 EBL machine contains the following options (Figure C-13):

- **Beam current (nA)** – The current of the exposure electron beam in nano-Ampere. The permitted values are from 0.125nA to 500nA.
- **Resist sensitivity (μC/cm^2)** – The threshold energy density of the resist used in the lithography. The permitted values range from 0.001 μC/cm^2 to 1000 μC/cm^2.
- **Exel size (nm)** - The distance in nanometres between adjacent exels. The permitted values for the exel size range from 1 nm to 300 nm. Since each field is 2^{15} exels by 2^{15} exels, so the allowable field sizes range from 32.768 μm to 9.8304 mm.
- **VRU** - This parameter allows the use of a large beam diameter to write every 2nd, 4th or 8th exel of the field. Every 2 to the power of VRU exel is exposed, therefore a VRU value of 0 will expose every exel, a VRU value of 1 will expose every 2nd exel, a VRU value of 2 will expose every 4th exel and a VRU value of 3 will expose every 8th exel.
- **Chip count** – The number of chips (pattern) to be repeatedly exposed by the EBL machine. The default value is 1 and the valid values are non-zero positive integer

numbers. This edit box is only enabled for the **Tools|EBL Machine Write Time Simulator** command in the exposure pattern viewer (section C.5.2.4).

- **Sub-field Size (exel)** – Shapes are fractured along a sub-field grid in the PEC module. This parameter specifies the size of the sub-field grid in exels and the corresponding size in nanometres will be updated whenever the parameter value is changed. The given parameter values must be non-zero positive integer.
- **Shift the grid origin at the top left point of the layout shadow by (exel)** – This parameter allows the user to specify the relative position of the sub-field grid from the top left point of the layout shadow. The valid values are integers range from 0 up to the corresponding **Sub-field Size (exel)** values.

EBMF10.5 options
Beam current (nA): 1
Exel size (nm) : 100
Resist sensitivity in microC/(cm*cm): 6.65
VRU : 0
Stardard clock frequencies (Hz) : 1503759.39849624
Total number of clock classes: 16
Chip count: 1
Sub-field settings:
X-axis :
Y-axis :
Size (exel) : 1024 1024
Size (nm) : 102400 102400
Shift the grid origin at the top left point of the layout shadow by (exel) : 0 0
OK
Cancel

Figure C-13 Leica EBMF 10.5 option dialog box showing the default values.

The dialog box also states the standard clock frequency calculated from the given parameter values and the maximum number of distinct clock frequencies (16) in a single pattern file, supported by the EBL machine. The standard clock frequency is computed as follows:

$$\text{Standard clock frequency (Hz)} = \frac{\text{Beam current (nA)}}{\text{Resist sensitivity }(\mu\text{C/cm}^2)}\left(\frac{3.2768}{\text{Field size (mm)}}\right)^2 \frac{10^7}{4^{\text{VRU}}}$$

The clock frequency of a normalized incident dose is calculated as the product of the standard clock frequency divided by the normalized incident dose.

Figure C-14 shows the options dialog box for non-Gaussian spot, step and write EBL machines. All the parameters have the same meaning and valid value ranges as for the corresponding parameters in the options dialog box for Leica EBMF10.5 EBL machines. When the total number of clock classes is zero, this implies that the EBL machine supports an infinite number of clock classes in a single pattern file, i.e. no quantization on the incident dose is required.

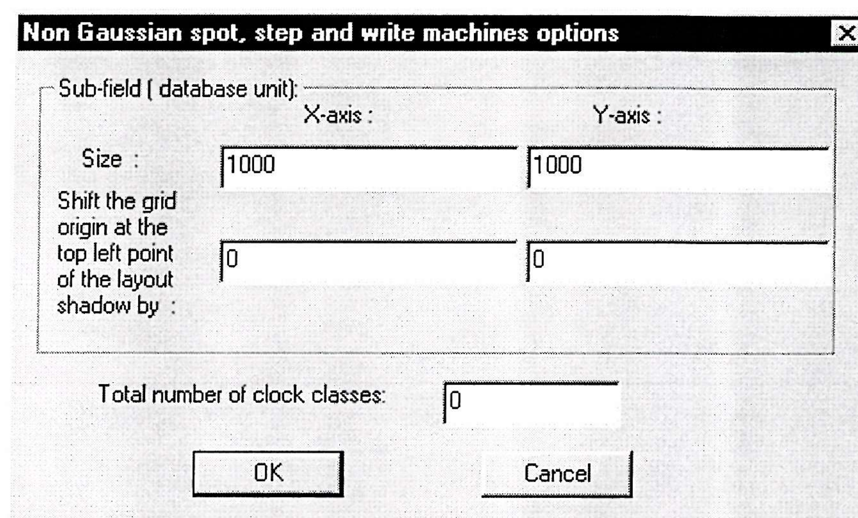


Figure C-14 Non-Gaussian spot, step and write EBL machines options dialog box with the default parameter values.

C.3.2.2 The EDF property page

This property page (Figure C-15) lets the user specify the parameter values of the double Gaussian EDF used in the PEC module and the proximity effect simulators. To simplify the evaluation of the EDF values, the EDF values of points outside *a coverage square* are taken as zero. The square centre coincides with the evaluation point. The page contains the following options:

- **Alpha** – The characteristic length, α in micrometers of the electron forward scattering.
- **Beta** – The characteristic length, β in micrometers of the electron back scattering.
- **Eta** - The deposited energy ratio, η of back scattered electrons to forward scattered electrons.

- **Half width of the coverage square (database)** – The required half width of the coverage square. Whenever this value is changed, the corresponding values in the **Tolerance** edit box will be updated.
- **Tolerance** – The error of ignoring the EDF values of points outside the coverage square. Whenever this value is changed, the corresponding values in the **Half width of the coverage square (database)** edit box will be updated.

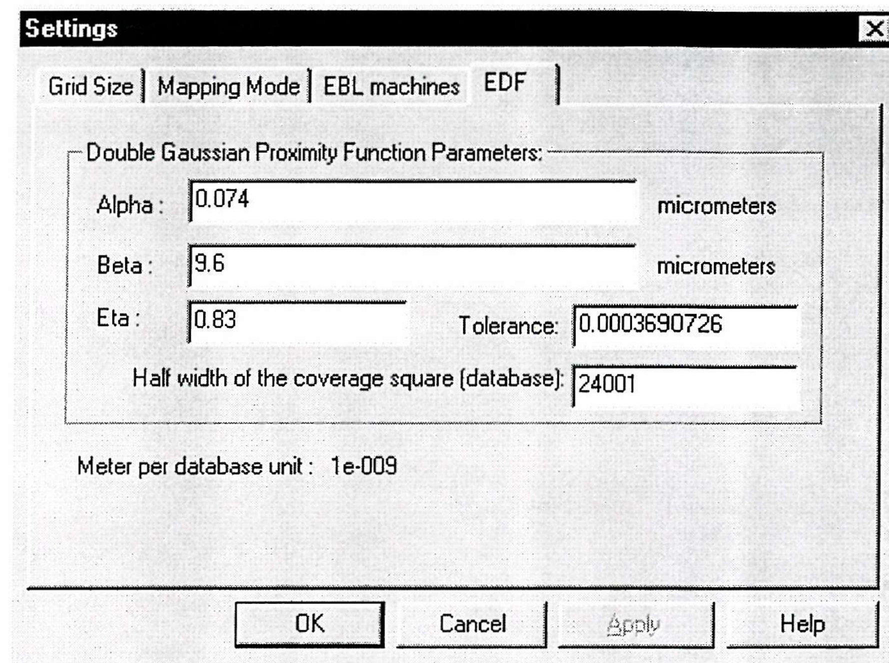


Figure C-15 EDF property page with the default parameter values.

C.3.3 The Tools menu

This menu contains two commands:

1. **Shape Processor**.
2. **Shape Processor With Corner Type Identification.**

Both commands convert the shapes in the current layout into rectangles and remove any overlapped parts among the shapes. To use the corner correction (section C.4.1.1) in the PEC module, it is necessary to use the second command that identifies the corner type of every rectangle corners.

If there is more than one layer displayed in the GDSII layout browser, an error message dialog box is issued and the commands will not be executed. Upon successful execution of the command, the GDSII layout browser prompts the user to

save the processed layout using a standard Windows 95 Save As dialog box. The GDSII layout browser sets the default file name in the dialog box as follows:

- The file name is the concatenation of the GDSII file name of the layout, the layer number of the displayed layer, the number of GDSII structures in the displayed layer layout, and either 'wocc' for the **Shape Processor** command or 'wcc' for the other command, separated by an underscore. E.g. for a GDSII file name of test.gds, a displayed layer number of 2 and a number of GDSII structures in the displayed layer layout of 3, the default file name is:
  - test_2_3_wocc.sp for the **Shape Processor** command.
  - test_2_3_wcc.sp for the other command.

After the file is saved, the processed layout is displayed in a newly created shape processor viewer window (Figure C-16).

C.4 The shape processor viewer

This viewer allows the user to browse the output pattern of the shape processor after overlap removal and conversion of polygons to rectangles. The viewer has a similar user interface (Figure C-16) as the GDSII layout browser.

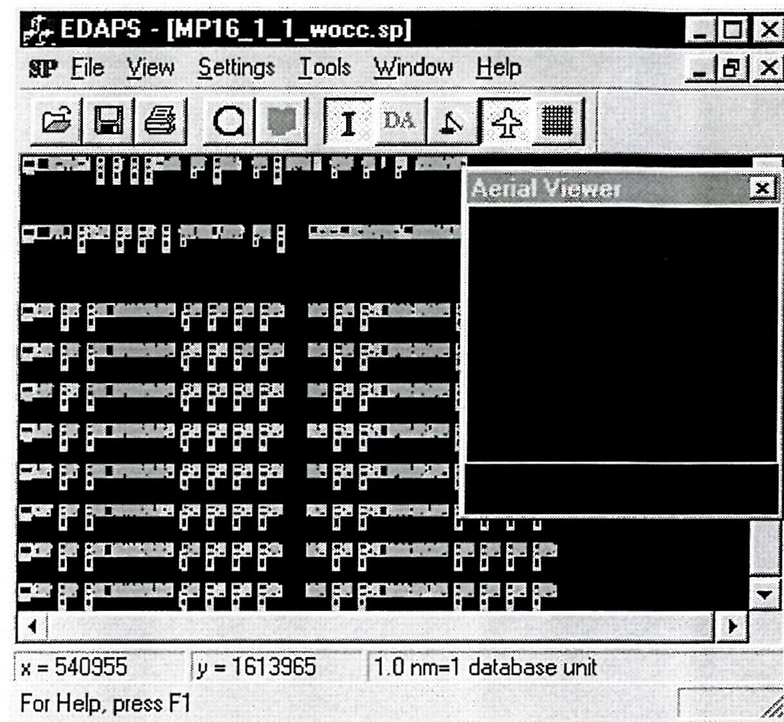


Figure C-16 User interface of the shape processor viewer.

C.4.1 The Tools menu

This menu let the user produce an exposure pattern of the entire layout pattern in the currently selected viewer window, with (through the **PADPEC** command) or without PEC (through the **NoPEC** command). Upon successful execution of the commands, the viewer issues a prompt to save the exposure pattern using a standard Windows 95 Save As dialog box. The viewer sets the default file name in the dialog box as follows:

- "_PADPEC" (for the **PADPEC** command) or "_NoPEC" (for the **NoPEC** command) is added to the end of the shape processor file name of the layout. E.g. for a shape processor file name of test.sp, the default file names are:

- test_PADPEC.ep for the **PADPEC** command.
- test_NoPEC.ep for the **NoPEC** command.

After the file is saved, the exposure pattern is displayed in a newly created exposure pattern viewer window (Figure C-20 on page 252).

For the **PADPEC** command, the viewer issues an Exposure Pattern Viewer Display Settings dialog box (Figure C-17) prior to the Windows 95 Save As dialog box. The exposure pattern viewer categories the normalized incident doses in the exposure pattern into equally spaced dose classes. Each dose class is represented by a unique colour in the display of the exposure pattern. Besides stating the range of the normalized incident dose in the exposure pattern, the dialog box (Figure C-17) also allows the user to specify:

- **Contour counts** – The dose class count. The valid values are integer numbers from 1 to 255. The default value is 16.
- **Contour step size** – The difference in the normalized incident dose values between two adjacent dose classes. The permitted values are non-zero positive real number. The default value is the range of the normalized incident dose of the exposure pattern divided by 16.
- **Minimum contour value** – The minimum normalized incident dose of the dose classes. The allowable values are positive real numbers. The default value is the minimum value of the normalized incident dose of the exposure pattern.
- **Shading** – There are two shading modes:
  - **Colour** – Each dose class is represented by a unique colour. This is the default setting.
  - **Greyscale** – Each dose class is represented by a unique grey colour. The lower the value of the normalized incident dose of a dose class, the darker its associated grey colour.

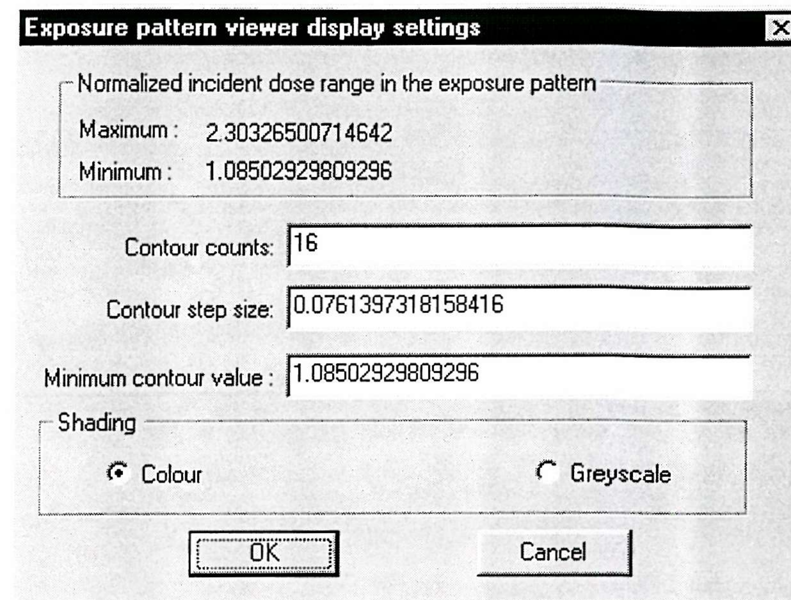


Figure C-17 Exposure Pattern Viewer Display Settings dialog box

C.4.1.1 The PADPEC command

This command issues the dialog box for setting the options of the PADPEC schemes (Figure C-18). The dialog box states the size of a database unit of the layout and the electron back scattered characteristic length, β in meters. The dialog box contains the following options:

- **Number of quantization levels for the incident dose (0 for no quantization)** – The permitted values are integers ranging from 0 to 256. When applicable, EDAPS quantizes the incident dose of the pattern using *uniform upper-level quantization strategy.*
- **Mesh Grid (database)** – These contain the settings for the mesh grid used in the PADPEC schemes.
  - **Size** – These parameters allow the user to specify the mesh sizes in database units of the layout. The valid values are non-zero positive integer numbers.
  - **Shift the grid origin at the top left point of the layout shadow by** – This parameter allows the user to specify the relative position (in database units) of the mesh grid from the top left point of the layout shadow. The valid values are integer numbers range from 0 up to the corresponding mesh grid sizes.

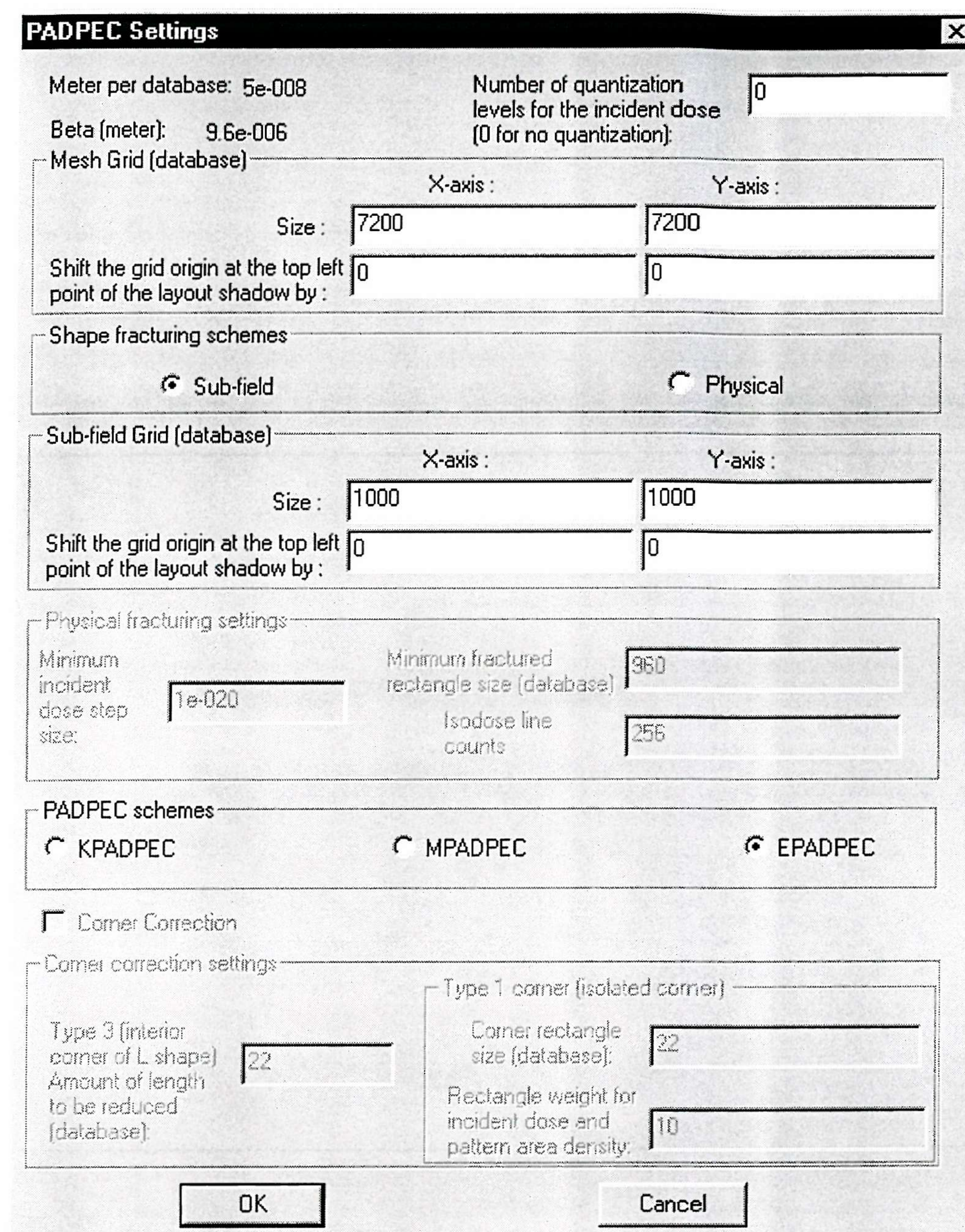


Figure C-18 PADPEC Settings dialog box with the default parameter values.

- **Shape fracturing schemes** – The user can choose between the sub-field and the physical algorithm for shape fracturing by left clicking on the appropriate radio button. Selecting a fracturing algorithm activates its associated options settings and deactivates the settings of the other scheme. The sub-field fracturing scheme partitions shapes along a fixed size sub-field grid, while the other scheme fractures shapes along the contours (isodose lines) of the incident dose distribution.

- **Sub-field Grid (database)** – These contain the options settings of the sub-field fracturing scheme.
  - **Size** – These parameters let the user specify the sub-field grid sizes in database units of the layout. The valid values are non-zero positive integer numbers.
  - **Shift the grid origin at the top left point of the layout shadow by** – This parameter allows the user to specify the relative position (in database units) of the sub-field grid from the top left point of the layout shadow. The valid values are integer numbers range from 0 up to the corresponding sub-field grid sizes.
- **Physical fracturing settings** – These contain the options settings of the physical fracturing scheme.
  - **Minimum incident dose step size** – This parameter specifies the minimum allowable difference of the incident dose between two adjacent isodose lines. The valid values are non-zero positive real number.
  - **Minimum fractured rectangle size (database)** – This parameter specifies the minimum allowable rectangle size in database units of the layout after shape fracturing. The valid values are non-zero positive integer numbers.
  - **Isodose line counts** - This parameter specifies the required number of isodose lines along which shapes are partitioned. The valid values are non-zero positive integer numbers.

  When the calculated incident dose step size is smaller than the specified minimum size, the given isodose lines count is modified in such a way that the new dose step size is the smallest value that is greater or equal to the minimum step size.
- **PADPEC schemes** – The user can choose one of the three PADPEC methods by left clicking on the appropriate radio button.
- **Corner Correction** – This check box is enabled only if EPADPEC is selected and the shape processor file has been processed using the **Tools|Shape Processor With Corner Type Identification** command in the GDSII layout browser. Checking the check box activates the options settings for the corner correction.
- **Corner correction settings** – These contain the option settings of the corner correction scheme.
  - **Type 3 (interior corner of L shape) Amount of length to be reduced (database)** – The valid values are positive integer numbers.

➢ **Type 1 corner (isolated corner)** – These contain the settings for type 1 corners.

- **Corner rectangle size (database)** – The valid values are positive integer numbers.
- **Rectangle weight for incident dose and pattern area density** - The valid values are positive real numbers.

C.4.1.2 The No PEC command

This command prompts the user for the uniform normalized incident dose value using a No PEC dialog box (Figure C-19). The default dose value is 1.0 and the given value is assigned to all the shapes in the layout. The valid values are non-zero positive real numbers

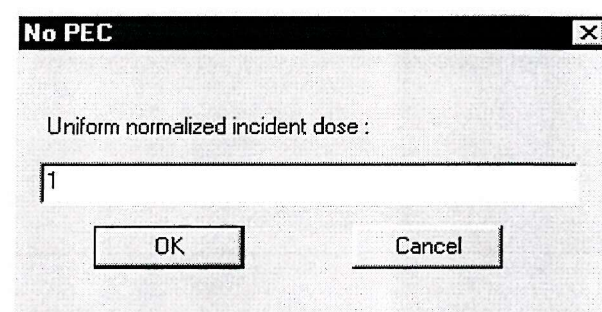


Figure C-19 No PEC dialog box.

C.5 The exposure pattern viewer

Figure C-20 shows the user interface of the exposure pattern viewer. The window is divided into two panes; the right pane displays the exposure pattern while the left pane shows the range of each *dose class*. Each dose class is represented by a unique colour. When displaying the exposure pattern, the viewer draws each shape as a filled polygon with the colour of the dose class, in which the normalized incident dose of the shape is within the range of the dose class. If the exposure pattern is produced using the **Tools|NoPEC** command in the shape processor viewer, the first pane will not be displayed, as there is only one dose class. The viewer highlights the rectangle in the pattern, in which the current mouse position is. It displays the co-ordinates (in database units) of the mouse position, the normalized incident dose assigned to the highlighted rectangle, and the size of a database unit (in nanometres) in its status bar.

It tracks the co-ordinates of the mouse position if the position is within the client area of the right pane.

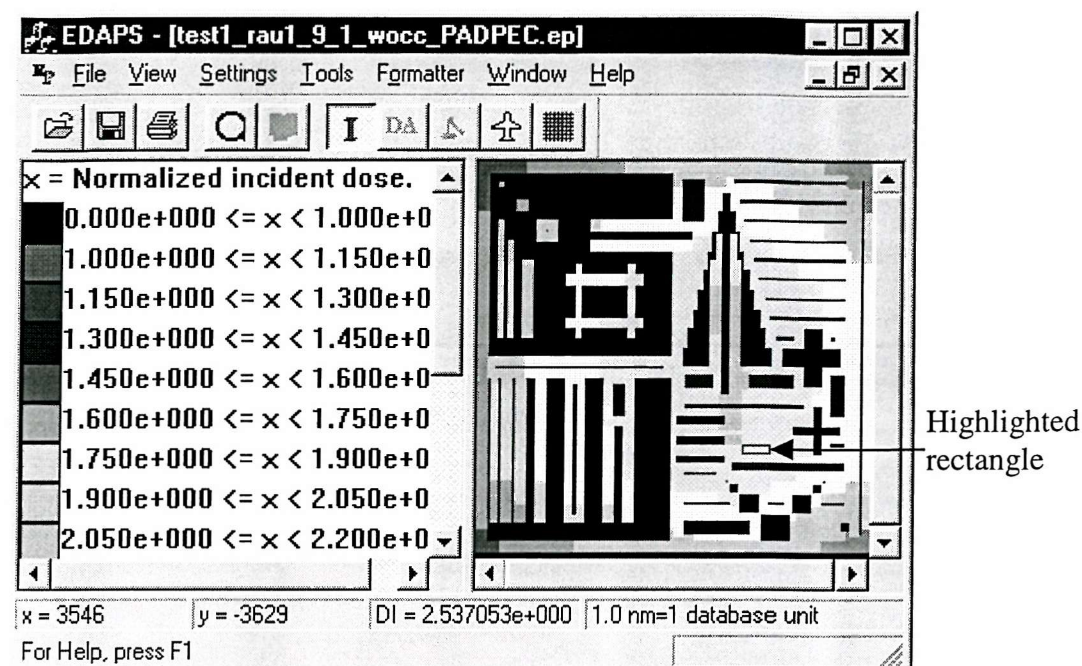


Figure C-20 User interface of the exposure pattern viewer.

C.5.1 The View menu

Besides the standard **View** commands, the viewer has two additional commands:

- **Reverse Y Axis Orientation**;
- **PEC Settings**.

Selecting the former command causes the viewer to re-draw the exposure pattern in the reverse orientation on the y-axis, while the latter command displays the PEC settings used to produce the exposure pattern in a dialog box.

C.5.2 The Tools menu

The user can inspect the influence of the proximity effect on the exposure pattern using the proximity effect simulator available via the **Tools** menu. In all the simulators, the user specifies the co-ordinates of the simulation window and the sample grid sizes used in the simulation.

C.5.2.1 The Exposure Simulator command

This command issues an Exposure Simulator Settings dialog box (Figure C-21). Besides stating the co-ordinates of the exposure pattern displayed in the viewer and the database unit size, the dialog box let the user specify the co-ordinates of the required simulation window and the required sample grid sizes in the simulation. The default simulation window is the same as the displayed size of the exposure pattern in the viewer window. The valid values of the co-ordinates are integer numbers and the right and bottom co-ordinates must be equal to or greater than the left and top co-ordinates respectively of the simulation window. On the other hand, the default values for the sample grid sizes are 1 database unit and the permitted values are non-zero positive integer numbers. Upon left clicking on the **OK** pushbutton, the exposure simulator computes the normalized deposited energy density at each sample point on the sample grid in the simulation window. An Exposure Simulator Display Settings dialog box (see section C.6.2.1 for further details.) is issued before the simulation result is displayed in a newly created exposure simulator viewer window (Figure C-24 on page 258).

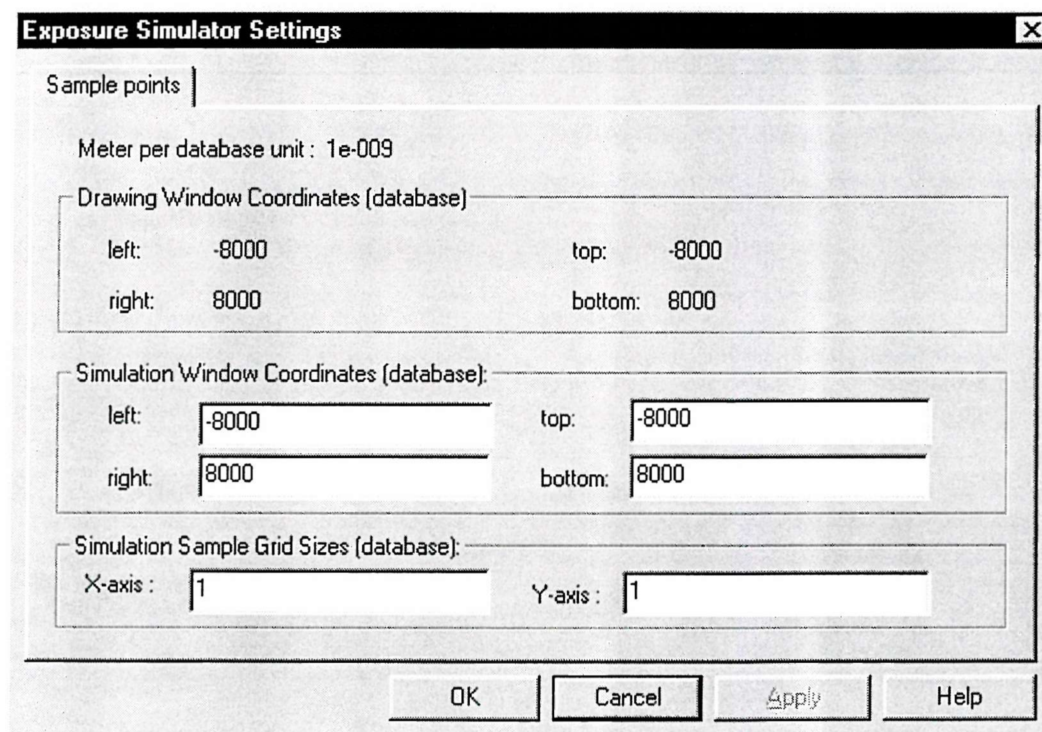


Figure C-21 Exposure Simulator Settings dialog box.

C.5.2.2 The Threshold Developer Simulator command

This command issues a Threshold Developer Simulator Settings *wizard*-type dialog box (Figure C-22). The user specifies the desired normalized resist threshold energy density. A point in a resist is *developed* if the normalized deposited energy density at that point is equal to or greater than the resist threshold value; otherwise the point is *undeveloped*. Upon left clicking on the **Next** pushbutton, a new dialog box appears on the screen. The dialog box is similar to the one shown in Figure C-21 except the **OK**, **Cancel** and **Apply** pushbuttons are replaced with **Back**, **Finish** and **Cancel** pushbuttons respectively.

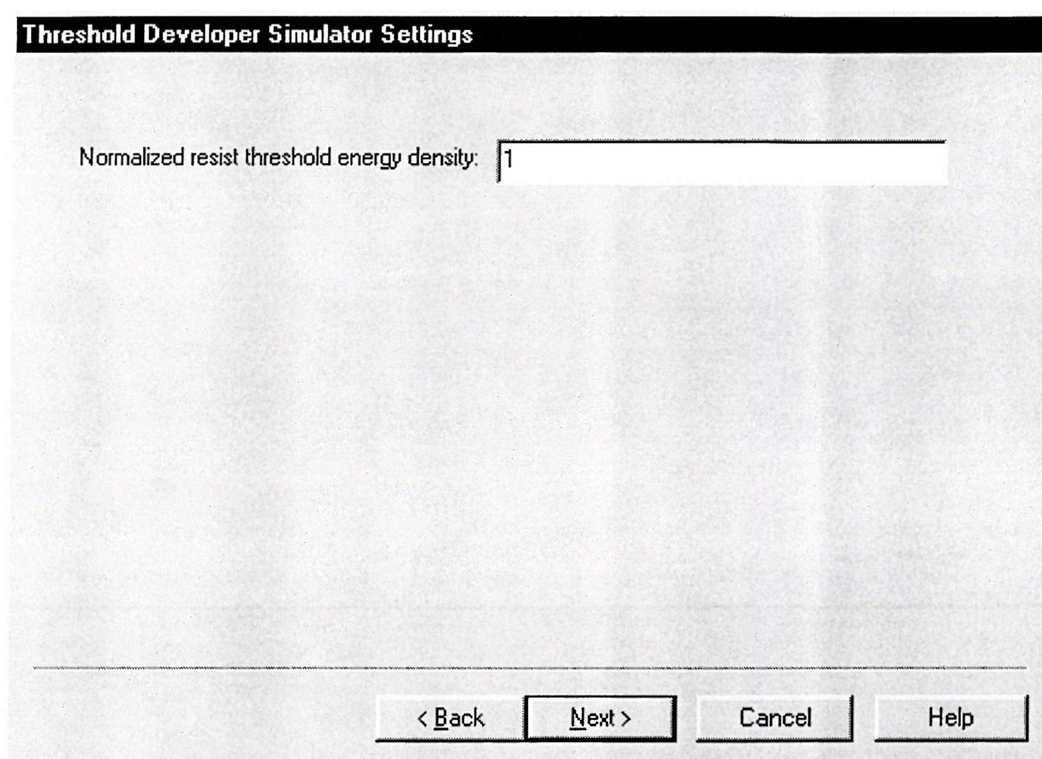


Figure C-22 Threshold Developer Simulator Settings *wizard*-type dialog box.

Upon left clicking on the **Finish** pushbutton, the threshold developer simulator checks whether each sample point on the grid in the simulation window is developed or not by first evaluating the normalized deposited energy density at each sample point and comparing the computed value with the resist threshold value. The simulation result is displayed in a newly created threshold developer simulator viewer window (Figure C-26 on page 259).

C.5.2.3 The Resist Characteristic Curve Developer Simulator command

This command issues a Resist Characteristic Curve Developer Simulator Settings *wizard*-type dialog box (Figure C-23). The dialog box lists all the available resist characteristic curve files in the displayed directory. A resist characteristic curve file contains the mapping of a specific resist between the remaining resist thickness at a point after resist development and the normalized deposited energy density at the point. The default setting is to select the first file in the list. Upon left clicking on the **Next** pushbutton, a new dialog box appears on the screen. The dialog box is similar to the one shown in Figure C-21 except the **OK**, **Cancel** and **Apply** pushbuttons are replaced with **Back**, **Finish** and **Cancel** pushbuttons respectively.

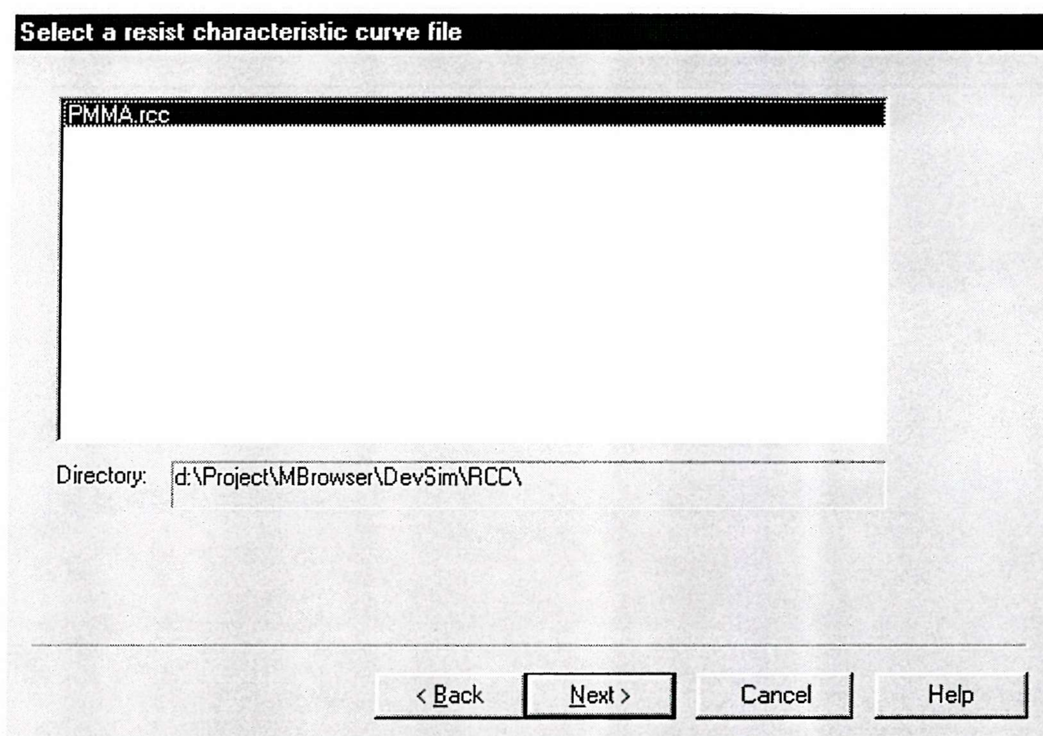


Figure C-23 Resist Characteristic Curve Developer Simulator Settings dialog box.

Upon left clicking on the **Finish** pushbutton, the resist characteristic curve developer simulator computes the remaining resist thickness at each sample point on the grid in the simulation window. The simulation result is displayed in a newly created resist characteristic curve developer simulator viewer window (Figure C-27 on page 260).

C.5.2.4 The EBL Machine Write Time Simulator command

This command estimates the write time of the exposure pattern on a Leica EBMF 10.5 EBL machine. The command is not available if the selected EBL machine is not EBMF 10.5 when creating the exposure pattern. The command firstly issues the Leica EBMF 10.5 option dialog box (Figure C-13) to get the standard clock frequency and the number of chips to be exposed by the EBL machine. Next, it displays the estimated write time in a dialog box.

C.5.3 The Formatter menu

This menu contains commands for converting the exposure pattern into different file formats. Currently, EDAPS can only convert the exposure pattern into the Cambridge Source Pattern Data Language format.

Besides the format conversion, the **Cambridge Source Pattern Data Language (SPD)** command also translates the normalized incident dose into clock frequencies. The converted exposure pattern files have a *spd* extension while the clock frequencies are stored in files with a *job* extension. The command firstly prompts the user for the standard clock frequency using the Leica EBMF 10.5 option dialog box (Figure C-13). The sub-field settings in the dialog box are disabled. Next, it issues a standard Windows 95 Save As dialog box to allow the user to enter the name for the output files. The default file name is the name of the exposure pattern file. If the specified file name is already exist, then a confirmation dialog box prompts the user to confirm overwriting the existing file.

C.6 Viewers of the proximity effect simulator

The viewers of the exposure, the threshold developer and the resist characteristic curve developer simulators have the same **View** menu.

C.6.1 The View menu

Besides the standard **View** commands, the viewer has three additional commands:

- **Reverse Y Axis Orientation**;
- **Desired Pattern**;
- **Simulation Settings**.

The first command performs the same task as the **View| Reverse Y Axis Orientation** command in section C.5.1, while the last command displays the simulation settings in a dialog box.

The second command acts as a toggle switch for the display of the desired pattern. The viewer draws the desired pattern on top of the simulation results as white outline rectangles (The threshold developer simulator viewer displays the desired pattern as black outline rectangles.) When the desired pattern is displayed, a tick mark is shown next to the command in the menu and the **DesiredPattern** pushbutton is displayed as a pushed icon. The default setting does not display the desired pattern.

C.6.2 The exposure sim ulator viewer

Figure C-24 shows the user interface of the exposure simulator viewer. The window is divided into two panes; the right pane displays the simulation result while the left pane shows the range of each *energy class*. Each energy class is represented by a unique colour. When displaying the result, the viewer draws each sample point as a filled rectangle with the colour of the energy class, in which the computed normalized deposited energy density of the point is within the range of the energy class. The viewer displays the co-ordinates (in database units) of the mouse position and the sample point on the simulation grid, the normalized deposited energy density of the sample point and the size of a database unit (in nanometres) in its status bar. It tracks the co-ordinates of the mouse position and the sample point if the position is within the client area of the right pane.

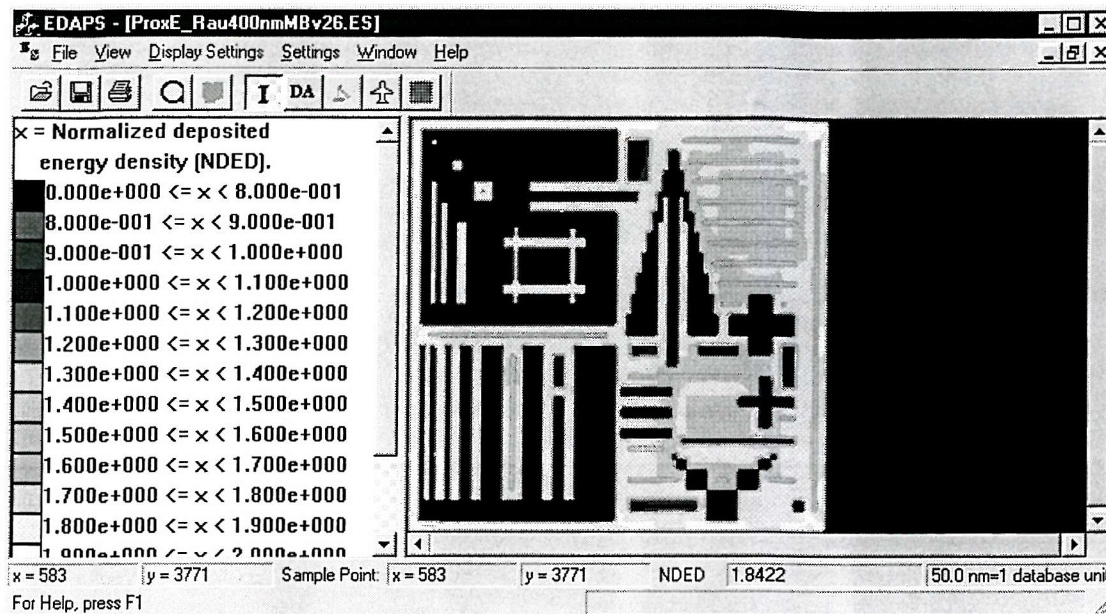


Figure C-24 User interface of the exposure simulator viewer.

C.6.2.1 The Display Settings menu

This menu issues an Exposure Simulator Display Settings dialog box (Figure C-25). The dialog box performs the same task, as the Exposure Pattern Viewer Display Settings dialog box in section C.4.1 except the quantity is the normalized deposited energy density instead of the normalized incident dose.

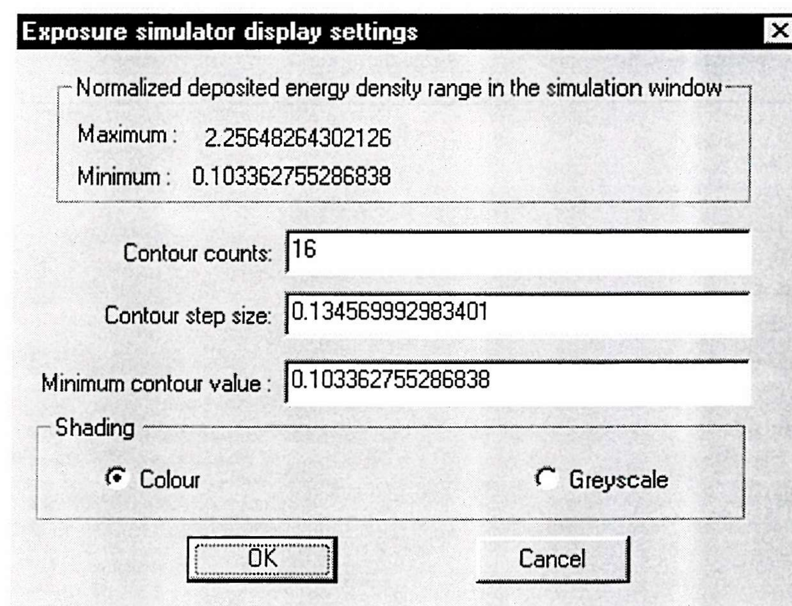


Figure C-25 Exposure Simulator Display Settings dialog box.

C.6.3 The threshold developer simulator viewer

Figure C-26 shows the user interface of the threshold developer simulator viewer. The viewer displays the co-ordinates (in database units) of the mouse position and the sample point on the simulation grid and the size of a database unit (in nanometres) in its status bar. It tracks the co-ordinates of the mouse position and the sample point if the position is within the client area.

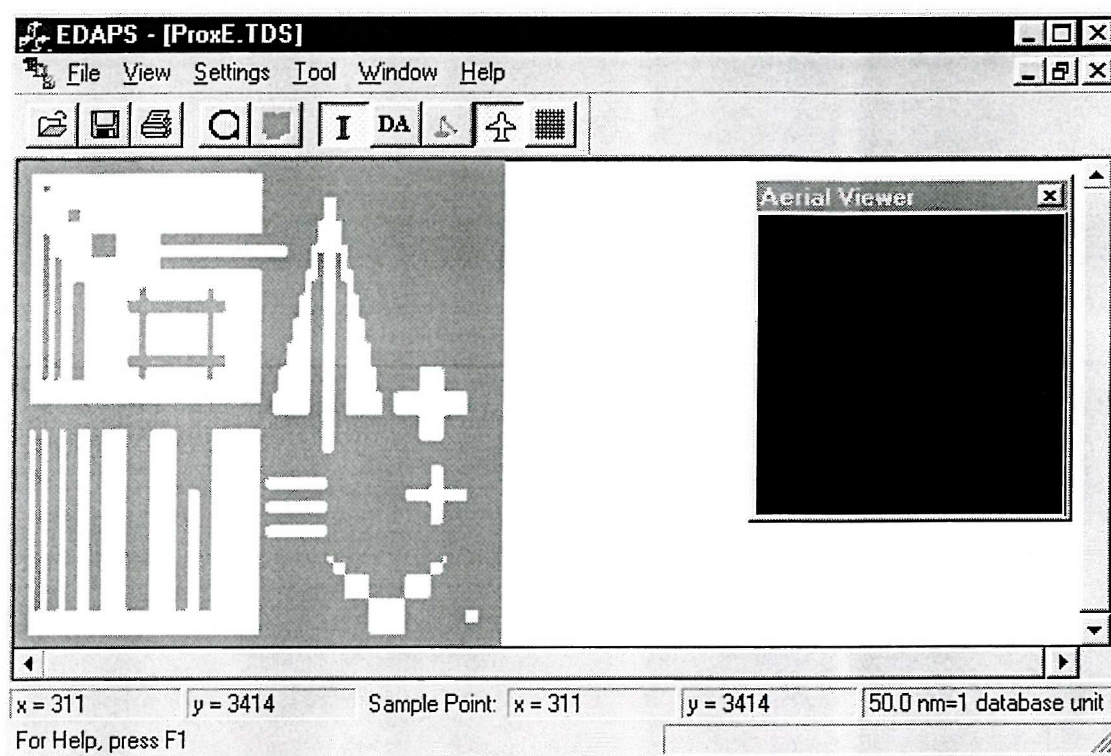


Figure C-26 User interface of the threshold developer simulator viewer.

C.6.3.1 The Tool menu

The **Compute Absolute Area Difference Sum Error** command in this menu calculates the absolute area difference between the simulated and the desired developed resist images in units of sample points. The viewer displays the result in a dialog box.

C.6.4 The resist characteristic curve developer simulator viewer

Figure C-27 shows the user interface of the resist characteristic curve developer simulator viewer. The user interface is similar to the user interface of the exposure simulator viewer except the quantity displayed in the left pane and the status bar is the normalized remaining resist thickness instead of the normalized deposited energy density.

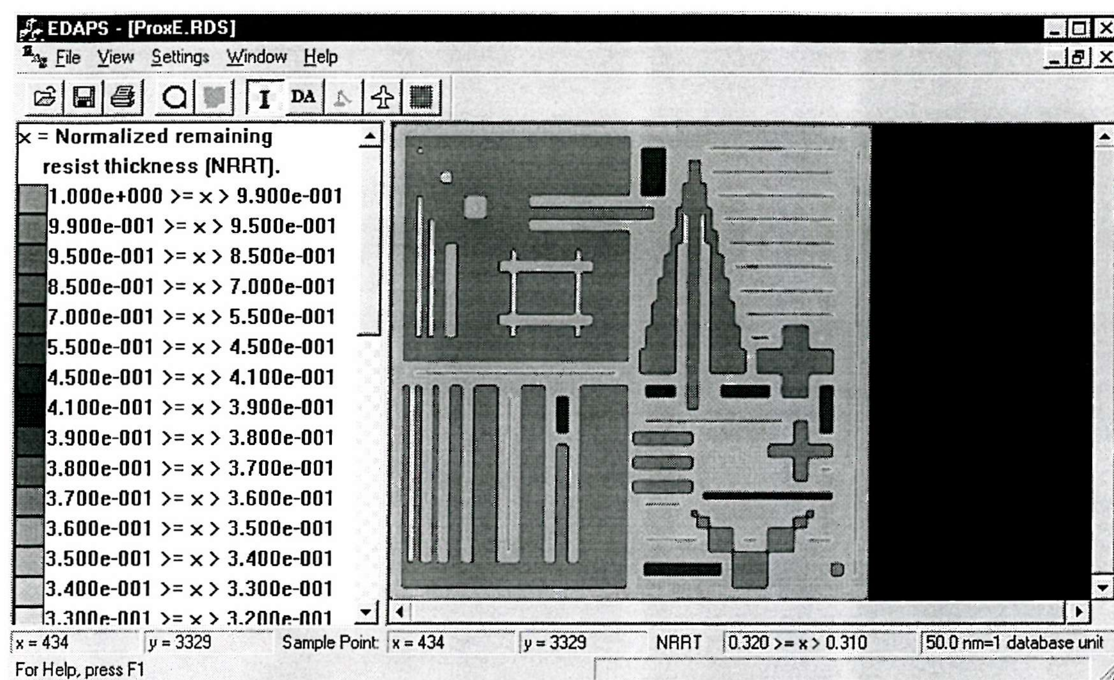


Figure C-27 User interface of the resist characteristic curve developer simulator viewer.

Appendix D

Calculation of the spot radius and the peak charge density of the correction beam used in GHOST, for an energy density function (EDF) with n Gaussian terms

For a normalized EDF with n Gaussian terms, $f(r)$:

$$f(r) = \frac{1}{\pi \sum_{i=1}^{n} \eta_i} \sum_{i=1}^{n} \frac{\eta_i}{\sigma_i^2} e^{-\left(\frac{r}{\sigma_i}\right)^2}$$

and an electron beam with a Gaussian charge density distribution, $q(r)$:

$$q(r) = Qe^{-\left(\frac{r}{\sigma}\right)^2}$$

the resultant EDF, $f'(r)$, is the convolution between $q(r)$ and $f(r)$. Using the result of convolution between two Gaussian terms (Appendix E.5), $f'(r)$ becomes:

$$f'(r) = \frac{Q}{\pi \sum_{i=1}^{n} \eta_i} \sum_{i=1}^{n} \frac{\eta_i}{\sigma_i^2 + \sigma^2} e^{-\frac{r^2}{\sigma_i^2 + \sigma^2}}$$

Thus, for a primary exposure beam with a spot radius of σ_e and a peak charge density of Q_e, the back scattering terms of the resultant EDF are:

$$\frac{Q_e}{\pi \sum_{i=1}^{n} \eta_i} \sum_{i=2}^{n} \frac{\eta_i}{\sigma_i^2 + \sigma_e^2} e^{-\frac{r^2}{\sigma_i^2 + \sigma_e^2}} \qquad \textbf{(D-1)}$$

and, for a correction beam with a spot radius of σ_c and a peak charge density of Q_c, the resultant EDF is:

$$\frac{Q_c}{\pi\sum_{i=1}^{n}\eta_i}\sum_{i=1}^{n}\frac{\eta_i}{\sigma_i^2+\sigma_c^2}e^{-\frac{r^2}{\sigma_i^2+\sigma_c^2}} \qquad \textbf{(D-2)}$$

Using equations (D-1) and (D-2), the two conditions imposed on the correction beam (see section 4.2) can then be expressed as

$$Q_c\sum_{i=1}^{n}\frac{\eta_i}{\sigma_i^2+\sigma_c^2}=Q_e\sum_{i=2}^{n}\frac{\eta_i}{\sigma_i^2+\sigma_e^2} \qquad \textbf{(D-3)}$$

and

$$Q_c=Q_e\frac{\sum_{i=2}^{n}\eta_i}{\sum_{i=1}^{n}\eta_i} \qquad \textbf{(D-4)}$$

respectively. Substituting equation (D-4) into equation (D-3) gives:

$$\sum_{i=1}^{n}\frac{\eta_i}{\sigma_i^2+\sigma_c^2}=\frac{\sum_{i=1}^{n}\eta_i}{\sum_{i=2}^{n}\eta_i}\sum_{i=2}^{n}\frac{\eta_i}{\sigma_i^2+\sigma_e^2} \qquad \textbf{(D-5)}$$

The right hand side of equation (D-5) is a constant, while the left-hand side decreases monotonically with σ_c^2. Thus, we can use the bisection method or other root finding methods [161] to solve equation (D-5) for σ_c.

Assuming that $\sigma_e \ll \sigma_2$ and $\sigma_1 \ll \sigma_c$, and replacing $n = 2$, $\sigma_2 = \beta$, $\eta_1 = 1$ and $\eta_2 = \eta$ for the double Gaussian EDF, equations (D-5) and (D-4) become

$$\sigma_c=\frac{\beta}{(1+\eta)^{\frac{1}{4}}} \qquad \textbf{(D-6)}$$

$$Q_c=Q_e\frac{\eta}{(1+\eta)} \qquad \textbf{(D-7)}$$

These are the same equations as derived by Owen *et al.* [52].

Appendix E

Proof of desirable mathematical features of a Gaussian function for PEC

A Gaussian function, $g(x)$ has the following general form:

$$g(x) = Ce^{-\left(\frac{x}{\sigma}\right)^2}$$

where C is a constant. σ is the characteristic length of the function and is equal to $\sqrt{2}$ times the standard deviation of the Gaussian distribution.

E.1 Separable property

A Gaussian function can be factored into its components/variables, i.e. the variables are separable:

$$\begin{aligned} g(x, y) &= Ce^{-\left(\frac{x^2+y^2}{\sigma^2}\right)} \\ &= Ce^{-\left(\frac{x^2}{\sigma^2}\right)}e^{-\left(\frac{y^2}{\sigma^2}\right)} \end{aligned}$$

This property reduces a multi-dimensional integral of a Gaussian function into a product of one-dimensional integrals of each of its components, simplifying the calculation considerably.

E.2 Normalization of a Gaussian function

A Gaussian function, $g(x)$ is normalized when its integral over all space is equal to unity:

$$\int_{-\infty}^{\infty} g(x)dx \equiv 1$$

Solving the above equation for C gives the coefficient for a normalized Gaussian function:

$$\int_{-\infty}^{\infty} g(x)dx = 1$$

$$C\int_{-\infty}^{\infty} e^{-\left(\frac{x}{\sigma}\right)^2} dx = 1$$

Substituting $\lambda = x/\sigma$ and $dx = \sigma d\lambda$ into the above equation gives

$$C\int_{-\infty}^{\infty} \sigma e^{-\lambda^2} d\lambda = 1 \qquad \textbf{(E-1)}$$

As

$$\int_{-\infty}^{\infty} e^{-\lambda^2} d\lambda = \sqrt{\pi}$$

so equation (E-1) becomes:

$$C\sigma\sqrt{\pi} = 1$$

$$C = \frac{1}{\sigma\sqrt{\pi}}$$

Thus, a normalized Gaussian function, $G(x)$ is as follows:

$$G(x) = \frac{1}{\sigma\sqrt{\pi}} e^{-\left(\frac{x}{\sigma}\right)^2}$$

Using the separable property (section E.1) of the Gaussian function, a normalized two-dimensional Gaussian function, *G(x,y)* is the product of *G(x)* and *G(y)*:

$$\begin{aligned} G(x,y) &= \frac{1}{\sigma\sqrt{\pi}} e^{-\left(\frac{x}{\sigma}\right)^2} \frac{1}{\sigma\sqrt{\pi}} e^{-\left(\frac{y}{\sigma}\right)^2} \\ &= \frac{1}{\sigma^2\pi} e^{-\left(\frac{x^2+y^2}{\sigma^2}\right)} \end{aligned}$$

E.3 Integration

The integral of a normalized Gaussian function, *G(x)* from *L* to *U* is:

$$\int_L^U G(x)dx = \int_L^U \frac{1}{\sqrt{\pi}\sigma} e^{-\left(\frac{x}{\sigma}\right)^2} dx$$

Substituting $\tau = x/\sigma$ and $dx = \sigma d\tau$ into the above equation gives:

$$\begin{aligned} \int_L^U \frac{1}{\sqrt{\pi}\sigma} e^{-\left(\frac{x}{\sigma}\right)^2} dx &= \int_{\frac{L}{\sigma}}^{\frac{U}{\sigma}} \frac{1}{\sqrt{\pi}\sigma} e^{-\tau^2} (\sigma d\tau) \\ &= \frac{1}{2} \int_{\frac{L}{\sigma}}^{\frac{U}{\sigma}} \frac{2}{\sqrt{\pi}} e^{-\tau^2} d\tau \\ &= \frac{1}{2}\left[\int_0^{\frac{U}{\sigma}} \frac{2}{\sqrt{\pi}} e^{-\tau^2} d\tau + \int_{\frac{L}{\sigma}}^{0} \frac{2}{\sqrt{\pi}} e^{-\tau^2} d\tau \right] \\ &= \frac{1}{2}\left[\int_0^{\frac{U}{\sigma}} \frac{2}{\sqrt{\pi}} e^{-\tau^2} d\tau - \int_0^{\frac{L}{\sigma}} \frac{2}{\sqrt{\pi}} e^{-\tau^2} d\tau \right] \end{aligned} \qquad \textbf{(E-2)}$$

Since the error function, *erf(x)* is

$$erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-\lambda^2} d\lambda$$

so equation (E-2) becomes:

$$\frac{1}{2}\left[\int_0^{\frac{U}{\sigma}} \frac{2}{\sqrt{\pi}} e^{-\tau^2} d\tau - \int_0^{\frac{L}{\sigma}} \frac{2}{\sqrt{\pi}} e^{-\tau^2} d\tau\right] = \frac{1}{2}\left\{erf\left(\frac{U}{\sigma}\right) - erf\left(\frac{L}{\sigma}\right)\right\}$$

Thus, the *proper* integral of a normalized Gaussian function is a linear combination of error functions:

$$\int_L^U \frac{1}{\sigma\sqrt{\pi}} e^{-\left(\frac{x}{\sigma}\right)^2} dx = \frac{1}{2}\left\{erf\left(\frac{U}{\sigma}\right) - erf\left(\frac{L}{\sigma}\right)\right\} \quad \textbf{(E-3)}$$

Using the separable property (section E.1) of the Gaussian function, the proper two-dimensional integral of a normalized Gaussian function, *G(x,y)* is:

$$\int_{L_y}^{U_y}\int_{L_x}^{U_x} \frac{1}{\sigma^2\pi} e^{-\left(\frac{x^2+y^2}{\sigma^2}\right)} dxdy = \left[\int_{L_x}^{U_x} \frac{1}{\sigma\sqrt{\pi}} e^{-\left(\frac{x}{\sigma}\right)^2} dx\right]\left[\int_{L_y}^{U_y} \frac{1}{\sigma\sqrt{\pi}} e^{-\left(\frac{y}{\sigma}\right)^2} dy\right]$$

$$= \frac{1}{4}\left\{erf\left(\frac{U_x}{\sigma}\right) - erf\left(\frac{L_x}{\sigma}\right)\right\}\left\{erf\left(\frac{U_y}{\sigma}\right) - erf\left(\frac{L_y}{\sigma}\right)\right\} \quad \textbf{(E-4)}$$

E.4 Convolution with an infinite line or rectangle

An infinite line can be expressed mathematically as follows:

$$l(x) = u(x - L) - u(x - R)$$

where *u(x)* is a step function while *R* and *L* are the co-ordinates of the right and the left edge of the line respectively. When the line is convolved with a normalized Gaussian function, *G(x)* the resultant function, *P(x)* is:

$$P(x) = \int_{-\infty}^{\infty} G(x-\lambda) l(\lambda) d\lambda$$

$$= \int_{-\infty}^{\infty} \frac{1}{\sqrt{\pi}\sigma} e^{-\left(\frac{x-\lambda}{\sigma}\right)^2} [u(\lambda - L) - u(\lambda - R)] d\lambda$$

$$= \int_{L}^{R} \frac{1}{\sqrt{\pi}\sigma} e^{-\left(\frac{x-\lambda}{\sigma}\right)^2} d\lambda$$

Substituting $\tau = (x\text{-}\lambda)$ and $d\lambda = -d\tau$ into the above equation gives:

$$\int_{L}^{R} \frac{1}{\sqrt{\pi}\sigma} e^{-\left(\frac{x-\lambda}{\sigma}\right)^2} d\lambda = \int_{x-L}^{x-R} \frac{1}{\sqrt{\pi}\sigma} e^{-\left(\frac{\tau}{\sigma}\right)^2} (-d\tau)$$

$$= \int_{L-x}^{R-x} \frac{1}{\sqrt{\pi}\sigma} e^{-\left(\frac{\tau}{\sigma}\right)^2} d\tau$$

Using the result in equation (E-3), the above expression becomes:

$$\int_{L-x}^{R-x} \frac{1}{\sqrt{\pi}\sigma} e^{-\left(\frac{\tau}{\sigma}\right)^2} d\tau = \frac{1}{2}\left\{ erf\left(\frac{R-x}{\sigma}\right) - erf\left(\frac{L-x}{\sigma}\right)\right\}$$

Thus

$$P(x) = \frac{1}{2}\left\{ erf\left(\frac{R-x}{\sigma}\right) - erf\left(\frac{L-x}{\sigma}\right)\right\} \quad \textbf{(E-5)}$$

Using the separable property (section E.1) of the Gaussian function, the convolution, *P(x,y)* between a normalized two-dimensional Gaussian function, *G(x,y)* and a rectangle, r(x,y) is the product of *P(x)* and *P(y)*:

$$
\begin{aligned}
P(x,y) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} G(x-\lambda, y-\tau) r(\lambda,\tau) d\lambda d\tau \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \frac{1}{\sigma^2\pi} e^{-\left(\frac{(x-\lambda)^2+(y-\tau)^2}{\sigma^2}\right)} [u(\lambda-L)-u(\lambda-R)][u(\tau-T)-u(\tau-B)] d\lambda d\tau \\
&= \left[\int_{-\infty}^{\infty} \frac{1}{\sigma\sqrt{\pi}} e^{-\left(\frac{(x-\lambda)^2}{\sigma^2}\right)} [u(\lambda-L)-u(\lambda-R)] d\lambda\right]\left[\int_{-\infty}^{\infty} \frac{1}{\sigma\sqrt{\pi}} e^{-\left(\frac{(y-\tau)^2}{\sigma^2}\right)} [u(\tau-T)-u(\tau-B)] d\tau\right] \\
&= P(x)P(y) \\
&= \frac{1}{4}\left\{erf\left(\frac{R-x}{\sigma}\right) - erf\left(\frac{L-x}{\sigma}\right)\right\}\left\{erf\left(\frac{B-y}{\sigma}\right) - erf\left(\frac{T-y}{\sigma}\right)\right\}
\end{aligned}
$$

(E-6)

where L, R, T and B are the co-ordinates of the left, right, top and bottom edges of the rectangle respectively, $B>T$ and $R>L$.

E.5 Convolution with another Gaussian function

When a normalized Gaussian, $G_1(x)$ with σ_1 is convolved with another normalized Gaussian, $G_2(x)$ with σ_2, the resultant function, $H(x)$ is:

$$
\begin{aligned}
H(x) &= \int_{-\infty}^{\infty} G_1(x-\lambda) G_2(\lambda) d\lambda \\
&= \int_{-\infty}^{\infty} \frac{1}{\sigma_1\sqrt{\pi}} e^{-\left(\frac{x-\lambda}{\sigma_1}\right)^2} \frac{1}{\sigma_2\sqrt{\pi}} e^{-\left(\frac{\lambda}{\sigma_2}\right)^2} d\lambda \qquad \textbf{(E-7)} \\
&= \frac{1}{\sigma_1\sigma_2\pi} \int_{-\infty}^{\infty} e^{-\left(\frac{x-\lambda}{\sigma_1}\right)^2 - \left(\frac{\lambda}{\sigma_2}\right)^2} d\lambda
\end{aligned}
$$

Rearranging the exponents:

$$-\left(\frac{x-\lambda}{\sigma_1}\right)^2-\left(\frac{\lambda}{\sigma_2}\right)^2=-\left[\frac{\sigma_2^2\left(x^2-2x\lambda+\lambda^2\right)+\sigma_1^2\lambda^2}{(\sigma_1\sigma_2)^2}\right]$$

$$=-\left[\frac{\sigma_2^2x^2-2x\lambda\sigma_2^2+\left(\sigma_1^2+\sigma_2^2\right)\lambda^2}{(\sigma_1\sigma_2)^2}\right]$$

$$=-\left[\frac{\sigma_2^2x^2+\left(\sqrt{\sigma_1^2+\sigma_2^2}\lambda-\frac{\sigma_2^2x}{\sqrt{\sigma_1^2+\sigma_2^2}}\right)^2-\frac{\sigma_2^4x^2}{\sigma_1^2+\sigma_2^2}}{(\sigma_1\sigma_2)^2}\right]$$

$$=-\left[\frac{\sigma_1^2\sigma_2^2x^2+\sigma_2^4x^2-\sigma_2^4x^2}{(\sigma_1\sigma_2)^2\left(\sigma_1^2+\sigma_2^2\right)}+\left(\frac{\sqrt{\sigma_1^2+\sigma_2^2}\lambda-\frac{\sigma_2^2x}{\sqrt{\sigma_1^2+\sigma_2^2}}}{\sigma_1\sigma_2}\right)^2\right]$$

$$=-\left[\frac{x^2}{\left(\sigma_1^2+\sigma_2^2\right)}+\left(\frac{\sqrt{\sigma_1^2+\sigma_2^2}\lambda}{\sigma_1\sigma_2}-\frac{\sigma_2^2x}{\sigma_1\sigma_2\sqrt{\sigma_1^2+\sigma_2^2}}\right)^2\right]$$

Substituting the above expression for the exponents and

$$\tau=\frac{\sqrt{\sigma_1^2+\sigma_2^2}\lambda}{\sigma_1\sigma_2}\text{ and }d\lambda=\frac{\sigma_1\sigma_2}{\sqrt{\sigma_1^2+\sigma_2^2}}d\tau$$

into equation (E-7) gives:

$$\frac{1}{\sigma_1\sigma_2\pi}\int_{-\infty}^{\infty}e^{-\left(\frac{x-\lambda}{\sigma_1}\right)^2-\left(\frac{\lambda}{\sigma_2}\right)^2}d\lambda=\frac{e^{-\left(\frac{x}{\left(\sigma_1^2+\sigma_2^2\right)}\right)^2}}{\sigma_1\sigma_2\pi}\int_{-\infty}^{\infty}e^{-\left(\tau-\frac{\sigma_2^2x}{\sigma_1\sigma_2\sqrt{\sigma_1^2+\sigma_2^2}}\right)^2}\frac{\sigma_1\sigma_2}{\sqrt{\sigma_1^2+\sigma_2^2}}d\tau$$

$$=\frac{e^{-\left(\frac{x}{\left(\sigma_1^2+\sigma_2^2\right)}\right)^2}}{\pi\sqrt{\sigma_1^2+\sigma_2^2}}\int_{-\infty}^{\infty}e^{-\left(\tau-\frac{\sigma_2^2x}{\sigma_1\sigma_2\sqrt{\sigma_1^2+\sigma_2^2}}\right)^2}d\tau \tag{E-8}$$

As

$$\int_{-\infty}^{\infty}e^{-\lambda^2}d\lambda=\sqrt{\pi}$$

so equation (E-8) becomes:

$$\frac{e^{-\left(\frac{x}{\left(\sigma_1^2+\sigma_2^2\right)}\right)^2}}{\pi\sqrt{\sigma_1^2+\sigma_2^2}}\int_{-\infty}^{\infty} e^{-\left(\tau-\frac{\sigma_2^2 x}{\sigma_1\sigma_2\sqrt{\sigma_1^2+\sigma_2^2}}\right)^2} d\tau = \frac{e^{-\left(\frac{x}{\left(\sigma_1^2+\sigma_2^2\right)}\right)^2}}{\pi\sqrt{\sigma_1^2+\sigma_2^2}}\sqrt{\pi}$$

$$= \frac{1}{\sqrt{\pi}\sqrt{\sigma_1^2+\sigma_2^2}} e^{-\left(\frac{x}{\left(\sigma_1^2+\sigma_2^2\right)}\right)^2}$$

which is another normalized Gaussian function with a new characteristic length of $\sqrt{\sigma_1^2+\sigma_2^2}$.

Glossary and acronyms

ASIC – Application Specific Integration Circuit – fully custom designed integrated circuit.

CAD - Computer Aided Design.

CAPROX - Computer Aided PROXimity correction – a commercial proximity effect correction software package based on the hierarchy approach.

CASINO – Cambridge Algorithms for Scattering Interactions and Numerical Optimisation – A Monte Carlo simulation program that simulates electron scattering in solids. The program has been extensively researched and developed by the team at the Microelectronics Research Centre, of the Cavendish Laboratory, University of Cambridge under Professor Haroon Ahmed.

CD – Critical Dimension – the minimum feature size of a circuit pattern.

DRAM – Dynamic Random Access Memory – a type of volatile memory device.

EBL – Electron Beam Lithography – the process of printing patterns on a wafer using electron beams.

EDAPS – Electron beam DAta Preparation System – the software suite developed in this research program to facilitate the research.

EDD – Energy Density Distribution.

EDF – Energy Density Function – an analytic function that describes a deposited energy density distribution on a resist/substrate structure.

Edge contrast - the slope of a developed resist profile at shape edges.

EPADPEC – Enhanced Pattern Area Density Proximity Effect Correction – the proximity effect correction scheme detailed in this thesis.

FFT – Fast Fourier Transform – an algorithm that computes discrete Fourier transform. The algorithm needs a total number of arithmetic operations in the order of $N\log_2 N$ operations, where N is the total number of sample points.

FLOP – FLoating point OPerations.

FLOPS – FLoating point OPerations per Second – a figure of merit used for assessing the floating point performance of a computer.

FPPEC – Fourier Pre-compensation Proximity Effect Correction – a proximity effect correction scheme based on de-convolution.

GHOST - a proximity effect correction technique based on dose equalisation.

KPADPEC - Kasuga's Pattern Area Density Proximity Effect Correction - a type of dose correction scheme for proximity effect correction.

LADC – Local Area Dose Correction - a proximity effect correction scheme based on the optimisation approach.

LPEC – Leica Cambridge Proximity Effect Correction - a type of dose correction scheme for proximity effect correction.

ME – Maximum Entropy.

MIPS – Mega-Instructions Per Second - a figure of merit used for accessing performance of a computer.

MPADPEC - Murai's Pattern Area Density Proximity Effect Correction - a type of dose correction scheme for proximity effect correction.

PADPEC - Pattern Area Density Proximity Effect Correction - a type of dose correction scheme for proximity effect correction.

PEC - Proximity Effect Correction.

PFA – Physical Fracturing Algorithm - a shape partitioning scheme that fractures shapes along the contours (isodose lines) of the incident dose distribution.

PMMA - PolyMethyl MethAcrylate – a standard positive tone electron beam resist.

PROXECCO - PROXimity Effect Correction by de-COnvolution - a commercial proximity effect correction software package based on de-convolution.

PYRAMID - a hierarchical, rule-based scheme for proximity effect correction.

RFPEC – Representative Figure Proximity Effect Correction - a type of dose correction scheme for proximity effect correction.

rpm – Rotations Per Minute.

SFA – Sub-field Fracturing Algorithm – a shape partitioning scheme that fractures shapes along a sub-field grid.

VSB – Variable Shape Beam.

References

1. H. C. Pfeiffer and T. R. Groves, "Progress in E-beam Mask Making for Optical and X-ray Lithography", Microelectronic Engineering, Vol. 13, 1991, pp.141-149.

2. J. Walko, "TSMC Starts Shipping First Commercial 0.18 μm Parts", Electronics Times, No. 949, 17 May 1999, pp. 6.

3. A. N. Broers, A. C. F. Hoole and J. M. Ryan, "Electron Beam Lithography - Resolution Limits", Microelectronic Engineering, Vol. 32, 1996, pp. 131-142.

4. S. M. Rubin, "Computer Aids for VLSI Design", Addison-Wesley Publishing Company 1987, ISBN 0-201-05824-3, Appendix C.

5. R. DeJule, "E-beam Lithography: The Debate Continues", Semiconductor International, Vol. 19, September 1996, pp. 85-92.

6. Y.Sakitani, H. Yoda, H. Todokoro, Y. Shibata, T. Yamazaki, K. Ohbitu, N. Saitou, S. Moriyama, S. Okazaki, G. Matuoka, F. Murai and M. Okumura, "Electron-beam Cell-projection Lithography System", Journal Of Vacuum Science And Technology B, Vol. 10, 1992, pp. 2759-2763.

7. N. Saitou and Y. Sakitani, "Cell Projection Electron Beam Lithography", The Proceedings Of The SPIE - The International Society For Optical Engineering, Vol. 2194, 1994, pp. 11-21.

8. H. Satoh, Y. Someda, N. Saitou, K. Kawasaki, K. Mizuno, Y. Kadowaki, M. Hoga and T. Soga, "Performance Improvement in E-beam Reticle Writer HL-800M", The Proceedings Of The SPIE - The International Society For Optical Engineering, Vol. 3096, 1997, pp. 72-83.

9. C. P. Umbach, S. Washburn, R. A. Webb, R. Koch, M. Bucci, A. N. Broers and R. B. Laibowitz, "Observation of H/E Aharonov-Bohm Interference Effect in Submicron Diameter, Normal Metal Rings", Journal Of Vacuum Science And Technology B, Vol. 4, 1986, pp. 383-385.

10. V. Chandrasekhar, M. J. Rooks, S. Wind and D. E. Prober, "Observation of Aharonov-Bohm Electron Interference Effects with Periods h/e and h/2e in Individual Micron-Size, Normal-Metal Rings", Physics Review Letters, Vol. 55, 1985, pp. 1610-1613.

11. S. Washburn and R. A. Webb, "Aharonov-Bohm Effect in Normal Metal Quantum Coherence and Transport", Advances in Physics, Vol. 35, 1986, pp. 375-422.

12. M.D. Levenson, N. S. Viswanathan and R. A. Simpson, "Improving Resolution in Photolithography with a Phase-shifting Mask", IEEE Transactions on Electron Devices, Vol. ED-29, 1982, pp. 1828-1836.

13. H. I. Smith and H. G. Craighead, "Nanofabrication", Physics Today, Feb. 1990, pp. 24-30.

14. H. Ryssel, K. Haberger and H. Kranz, "Ion-beam Sensitivity of Polymer Resists", Journal Of Vacuum Science And Technology, Vol. 19, 1981, pp. 1358-1362.

15. S. C. Minne, S. R. Manalis, A. Atalar and C. F. Quate, "Independent Parallel Lithography Using the Atomic Force Microscope", Journal Of Vacuum Science And Technology B, Vol. 14, 1996, pp. 2456-2461.

16. T. H. P. Chang, D. P. Kern and L. P. Muray, "Microminiaturization of Electron Optical System", Journal Of Vacuum Science And Technology B, Vol. 8, 1990, pp. 1698-1705.

17. S. D. Berger, J. M. Gibson, R. M. Camarda, R. C. Farrow, H. A. Huggins, J. S. Kraus and J. A. Liddle, "Projection Electron-beam Lithography: a New Approach", Journal Of Vacuum Science And Technology B, Vol. 9, 1991, pp. 2996-2999.

18. L. R. Harriott, S. D. Berger, C. Biddick, M. I. Blakey, S. W. Bowler, K. Brady, R. M. Camarda, W. F. Connelly, A. Crorken, J. Custy, R. Dimarco, R. C. Farrow, J. A. Fetter, R. Freeman, L. Hopkins, H. A. Huggins, C. S. Knurek, J. S. Kraus, J. A. Liddle, M. M. Krtychan, A. E. Novembre, M. L. Peabody, R. G. Tarascon, H. H. Wade, W. K. Waskiewicz, G. P. Watson, K. S. Werder and D. Windt, "Preliminary Results From a Prototype Projection E-beam Stepper-scattering With Angular Limitation Projection Electron Beam Lithography Proof-of-concept System", Journal Of Vacuum Science And Technology B, Vol. 14, 1996, pp. 3825-3828.

19. D. J. D. Carter, H. I. Smith, K. W. Rhee and C. Marrian, " Sub-40 nm Pattern Replication With +/- 20% Process Lattitude by Soft-contact X-ray Lithography", The Proceedings Of The SPIE - The International Society For Optical Engineering, Vol. 3676, 1999, pp. 70-78.

20. T. J. Stark, K. M. Edenfeld, D. P. Griffis, Z. J. Radzimski and P. E. Russell, "Proximity Effects In Low-energy Electron-beam Lithography", Journal Of Vacuum Science & Technology B, Vol. 11, 1993, pp. 2367-2372.

21. K. Murata, T. Matsukawa and R. Shimizu, "Monte Carlo Calculations on Electron Scattering in a Solid Target", Japanese Journal Of Applied Physics, Vol. 10, No. 6, 1971, pp. 678-686.

22. M. Parikh and D. F. Kyser, "Energy Deposition Functions in Electron Resist Films on Substrates", Journal Of Applied Physics, Vol. 50, February 1979, pp. 1104-1111.

23. W. W. Molzen, A. N. Broers, J. J. Cuomo, J. M. E. Harper and R. B. Laibowitz, "Materials and Techniques Used in Nanostructure Fabrication", Journal Of Vacuum Science And Technology, Vol. 16, 1979, pp. 269-272.

24. I. Adesida, T. E. Everhart and R. Shimizu, "High Resolution Electron-beam Lithography on Thin Films", Journal Of Vacuum Science And Technology, Vol. 16, 1979, pp. 1743-1748.

25. R. J. Hawryluk, "Exposure and Development Models Used in Electron Beam Lithography", Journal Of Vacuum Science And Technology, Vol. 19, 1981, pp. 1-17.

26. S.A. Rishton and D.P. Kern, "Point Exposure Distribution Measurements for Proximity Correction in Electron Beam Lithography on a Sub-100nm Scale", Journal Of Vacuum Science And Technology B, Vol. 5, 1987, pp. 135-141.

27. C.R.K. Marrian, F.K. Perkins, D. Park, E.A. Dobisz and M.C. Peckerar, "Modelling of Electron Elastic and Inelastic Scattering", Journal Of Vacuum Science And Technology B, Vol. 14, 1996, pp. 3864-3869.

28. T.A. Fretwell and P.L. Jones, "Energy Intensity Distributions of 30keV Electrons on Indium Phosphide: Experiment and Simulation", Microelectronic Engineering, Vol. 23, 1994, pp. 97-99.

29. T.H.P. Chang, "Proximity Effect in Electron-beam Lithography", Journal Of Vacuum Science And Technology, Vol. 12, 1975, pp. 1271-1275.

30. T.H.P. Chang, D.P. Kern, E. Kratschmer, K.Y. Lee, H. E. Luhn, M.A. McCord, S.A. Rishton and Y. Vladimirdky, "Nanostructure Technology", IBM Journal Of Research and Development, Vol. 32, July 1988, pp. 462-493.

31. S.J. Wind, M.G. Rosenfield, G. Pepper, W.W. Molzen and P.D. Gerber, "Proximity Correction for Electron Beam Lithography Using a Three-Gaussian Model of the Electron Energy Distribution", Journal Of Vacuum Science And Technology B, Vol. 7, 1989, pp.1507-1512.

32. M. Gentili, L. Grella, A. Lucchesini, L. Luciani, L. Mastrogiacomo and P. Musumeci, "Energy Density Function Determination in Very-high-resolution Electron-beam Lithography", Journal Of Vacuum Science And Technology B, Vol. 8, 1990, pp. 1867-1871.

33. K. Yu and J. Lee, "Curve Fitting a Quadruple Gaussian Function to the Monte Carlo Simulation Results for Proximity Effect Correction Parameters", Journal Of The Korean Physical Society, Vol. 28, No. 4, August 1995, pp.524-528.

34. S. Aya, K. Kise, H. Yabe and K. Marumoto, "Validity of Double and Triple Gaussian Functions for Proximity Effect Correction in X-ray Mask Writing", Japanese Journal Of Applied Physics, Vol. 35, Part 1, No. 3, March 1996, pp. 1929-1936.

35. O. W. Otto and A. K. Griffith, "Proximity Correction on the AEBLE-150", Journal Of Vacuum Science & Technology B, Vol. 6, 1988, pp. 443-447.

36. J.J. More, "The Levenberg-Marquardt Algorithm: Implementation and Theory", Proceedings of the Biennial Conference on Numerical Analysis, *Springer-Verlag, Berlin, West Germany,* 1978, pp.105-116.

37. M.G. Rosenfield, S.J. Wind, W.W. Molzen and P.D. Gerber, "Determination of Proximity Effect Correction Parameters for 0.1 μm Electron-beam Lithography", Microelectronic Engineering, Vol. 11, 1990, pp. 617-623.

38. T.A. Fretwell, R. Gurung and P.L. Jones, "Curve Fitting to Monte Carlo Data for the Determination of Proximity Effect Correction Parameters", Microelectronic Engineering, Vol. 17, 1992, pp. 389-394.

39. L. Stevens, R. Jonckheere, E. Froyen, S. Decoutere and D. Laneer, "Determination of the Proximity Parameters in Electron Beam Lithography Using Doughnut-structures", Microelectronic Engineering, Vol. 5, 1986, pp. 141-150.

40. J. Zhu, Z. Cui and P.D. Prewett, "Experimental Study of Proximity Effect Corrections in Electron Beam Lithography", The Proceedings Of The SPIE - The International Society For Optical Engineering, Vol. 2437, 1995, pp. 375-382.

41. S.V Dubonos, B.N. Gaifullin, H.F. Raith, A. A. Svintsov and S.I. Zaitsev, "Evaluation, Verification and Error Determination of Proximity Parameters α, β and η in Electron Beam Lithography", Microelectronic Engineering, Vol. 21, 1993, pp. 293-296.

42. R.M. Weiss and R.M. Sills, "Application of an Electron Beam Scattering Parameter Extraction Method for Proximity Correction in Direct-write Electron Beam Lithography", The Proceedings Of The SPIE - The International Society For Optical Engineering, Vol. 1465, 1991, pp. 192-200.

43. S. Uchiyama, S. Ohki and T. Matsuda, "A New Proximity Parameter Evaluation Method Utilizing Auxiliary Patterns for Dose Compensation", Japanese Journal Of Applied Physics, Vol. 32, Part 1, No. 12B, December 1993, pp. 6028-6034.

44. K. K. Christenson, R. G. Viswanathan, and F. J. Hohn, "X-ray Mask Fogging by Electrons Backscattered Beneath the Membrane", Journal Of Vacuum Science And Technology B, Vol. 8, 1990, pp. 1618-1623.

45. C. P. Umbach and A. N. Broers, "Experimental Determination of the Proximity Effect from 25 to 100keV in Electron Beam Patterned X-ray Masks", Journal Of Vacuum Science And Technology B, Vol. 8, 1990, pp. 1614-1617.

46. Y. Yau, R. F. W. Pease, A. Iranmanesh, and K. Polasko, "Generation and Applications of Finely Focused Beams of Low-energy Electrons", Journal Of Vacuum Science And Technology, Vol. 19, 1981, pp. 1048-1052.

47. F. Cerrina and C. Marrian, "A Path to Nanolithography", MRS Bulletin, Vol. 21, 1996, pp. 56-62.

48. C.R.K. Marrian and E.S. Snow, "Proximal Probe Lithography and Surface Modification", Microelectronic Engineering, Vol. 32, 1996, pp. 173-189.

49. M. A. McCord and T. H. Newman, "Low Voltage, High Resolution Studies of Electron Beam Resist Exposure and Proximity Effect", Journal Of Vacuum Science And Technology B, Vol. 10, 1992, pp. 3083-3087.

50. B. Houli, V. Umansky and M. Heiblum, "Low-energy Electron Beam Lithography with 30 nm Resolution", Semiconductor Science and Technology, Vol. 8, 1993, pp. 1490-1492.

51. P.D. Gerber, "Exact Solution of the Proximity Effect Equation by a Splitting Method", Journal Of Vacuum Science And Technology B, Vol. 6, 1988, pp. 432-435.

52. G. Owen and P.Rissman, "Proximity Effect Correction for Electron Beam Lithography by Equalization of Background Dose", Journal Of Applied Physics, Vol. 54, June 1983, pp. 3573-3581.

53. H. Eisenmann, T. Waas and H. Hartmann, "PROXECCO - Proximity Effect Correction By Convolution", Journal Of Vacuum Science And Technology B, Vol. 11, 1993, pp. 2741-2745.

54. T. Abe, S. Yamasaki, R. Yoshikawa and T. Takigawa, "Representative Figure Method for Proximity Effect Correction", Japanese Journal Of Applied Physics, Vol. 30, Part 2, No. 3B, March 1991, pp. 528-531.

55. F. Murai, H.Yoda, S. Okazaki, N. Saitou and Y. Sakitani, "Fast Proximity Effect Correction Method Using a Pattern Area Density Map", Journal Of Vacuum Science And Technology B, Vol. 10, 1992, pp. 3072-3076.

56. T. K. Leen, "Theory and Practice of Proximity Correction by secondary Exposure", Journal Of Applied Physics, Vol. 65, February 1989, pp. 1776-1781.

57. J.A. Liddle, G. P. Watson, S.D. Berger and P.D. Miller, "Proximity Effect Correction in Projection Electron Beam Lithography (Scattering with Angular Limitation Projection Electron-beam Lithography)", Japanese Journal Of Applied Physics, Vol. 34, Part 1, No. 12B, December 1995, pp. 6672-6678.

58. G. P. Watson, S. D. Berger, J. A. Liddle and W. K. Waskiewicz, "A Background Dose Proximity Effect Correction Technique for Scattering with Angular Limitation Projection Electron Lithography Implemented in Hardware", Journal Of Vacuum Science And Technology B, Vol. 13, 1995, pp. 2504-2507.

59. M.A. Gesley and M.A. McCord, "100kV GHOST Electron Beam Proximity Correction on Tungsten X-ray Masks", Journal Of Vacuum Science And Technology B, Vol. 12, 1994, pp. 3478-3482.

60. G. Owen, P. Rissman and M.F. Long, "Application of the GHOST Proximity Effect Correction Scheme to Round Beam and Shape Beam Electron Lithography System", Journal Of Vacuum Science And Technology B, Vol. 3, 1985, pp. 153-158.

61. R.L. Kostelak, E.H. Kung, M.G.R. Thomson and S. Vaidya, "GHOST Proximity Correction Technique: Its Parameters, Limitations, and Process Latitude", Journal Of Vacuum Science And Technology B, Vol. 6, 1988, pp. 448-455.

62. K. Moriizumi and A. N. Broers, "Tolerance On Alignment Error in GHOST Proximity Effect Correction", Journal Of Vacuum Science & Technology B, Vol. 11, 1993, pp. 2114-2120.

63. M. Peckerar, C. Marrian and F.K. Perkins, "Feature Contrast in Dose-equalisation Schemes Used for Electron-beam Proximity Control", Journal Of Vacuum Science And Technology B, Vol. 14, 1996, pp. 3880-3886.

64. A. Muray, R.L. Lozes, K. Milner and G. Hughes, "Proximity Effect Correction at 10 keV Using GHOST and Sizing for 0.4 μm Mask Lithography", Journal Of Vacuum Science And Technology B, Vol. 8, 1990, pp. 1775-1779.

65. A. Muray and R. L. Dean, "Proximity Effect Correction On MEBES For 1X Mask Fabrication: Lithography Issues and Tradeoffs At 0.25 Micron", The Proceedings Of The SPIE - The International Society For Optical Engineering, Vol. 1496, 1990, pp. 171-179.

66. B. C. Cha, J. M. Kim, B. G. Kim, S. W. Choi, H. S. Yoon and J. M. Sohn, "Evaluation of the Practical Use of GHOST Technique for Various E-beam Resists", The Proceedings Of The SPIE - The International Society For Optical Engineering, Vol. 3546, 1998, pp. 55-62.

67. M. Parikh, "Corrections to Proximity Effects in Electron Beam Lithography. I. Theory", Journal Of Applied Physics, Vol. 50, No. 6, June 1979, pp. 4371-4377.

68. S.-Y. Lee, J.S. Jacob, C.-M. Chen, J.A. McMillan and N.C. Mac-Donald, "Proximity Effect Correction In Electron-beam Lithography: A Hierarchical Rule-based Scheme - PYRAMID", Journal Of Vacuum Science And Technology B, Vol. 9, 1991, pp. 3048-3053.

69. J. S. Jacob, S-Y Lee, J.A. McMillan and N.C. MacDonald, "Fast Proximity Effect Correction: An Extension of PYRAMID for Circuit Patterns of Arbitrary Size", Journal Of Vacuum Science And Technology B, Vol. 10, 1992, pp. 3077-3082.

70. B.D. Cook and S.-Y. Lee, "Fast Proximity Effect Correction: An Extension of PYRAMID for Thicker Resists", Journal Of Vacuum Science And Technology B, Vol. 11, 1993, pp. 2762-2767.

71. S.-Y. Lee and B.D. Cook, "Interior Area Removal Method for PYRAMID", Journal Of Vacuum Science And Technology B, Vol. 12, 1994, pp. 3449-3454.

72. S. Y. Lee and B. Liu, "Region-wise Proximity Effect Correction for Heterogeneous Substrates in Electron-beam Lithography: Shape Modification", Journal Of Vacuum Science & Technology B, Vol. 14, 1996, pp. 3874-3879.

73. B.D. Cook and S.-Y. Lee, "PYRAMID - A Hierarchical, Rule-based Approach Toward Proximity Effect Correction - Part II: Correction", IEEE Transactions On Semiconductor Manufacturing, Vol. 11, No. 1, February 1998, pp. 117-128.

74. S. Y. Lee, B. Liu and B. D. Cook, "Reducing Recursive Effect for Fast Proximity Correction", Microelectronic Engineering, Vol. 35, 1997, pp. 491-494.

75. S. Y. Lee and G. D. Ghare, "Distributed Proximity Effect Correction on a Network of Workstation", Microelectronic Engineering, Vol. 46, 1999, pp. 291-294.

76. B.D. Cook and S.-Y. Lee, "Performance Analysis and Optimal Parameter Selection for PYRAMID", Microelectronic Engineering, Vol. 30, 1996, pp. 61-64.

77. B. D. Cook and S. Y. Lee, "Dose, Shape, and Hybrid Modifications for PYRAMID in Electron Beam Proximity Effect Correction", Journal Of Vacuum Science And Technology B, Vol. 15, 1997, pp. 2303-2308.

78. S. K. S. Ma and M. Parikh, "Proximity Corrections in a Raster Scan Electron Lithography Machine", Journal Of Vacuum Science & Technology, Vol. 19, 1981, pp. 1275-1278.

79. P. Nehmiz, H. Bohlen, J. Greschner, E. Bretschev and P. Vettiger, "Approach to Correct the Proximity Effect in Electron Beam Proximity Printing", Journal Of Vacuum Science & Technology, Vol. 19, 1981, pp. 1291-1295.

80. M. Parikh, "Proximity Effects in Electron Lithography: Magnitude and Correction Techniques", IBM Journal Of Research and Development, Vol. 24, July 1980, pp. 438-451.

81. C. J. Song and P. Y. Qiu, "A New Approach to Proximity Effect Correction", Microelectronic Engineering, Vol. 3, 1985, pp. 93-98.

82. K. Harafuji, A. Misaka, K. Kawakita and N. Nomura, "Proximity Effect Correction Data Processing System for Electron Beam Lithography", Journal Of Vacuum Science And Technology B, Vol. 10, 1992, pp. 133-142.

83. W. Lu, H. Shen, J. Tao, N. Gu and Y. Wei, "Improved Proximity Correction Algorithm for Electron-beam Lithography", The Proceedings Of The SPIE - The International Society For Optical Engineering, Vol. 2194, 1994, pp. 323-330.

84. K. Nakajima, Y. Kojima, S. Hirasawa, H. Mukai, S. Ishida, T. Hirota, K. Kondoh and N. Aizaki, "0.25 μm Electron Beam Direct Write Techniques for 256Mbit Dynamic Random Access Memory Fabrication", Japanese Journal Of Applied Physics, Vol. 32, Part 1, No. 12B, December 1993, pp. 6023-6027.

85. E. D. Fabrizio, L. Grella, M. Baciocchi, M. Gentili, D. Peschiaroli, L. Mastrogiacomo and R. Maggiora, "Fast Proximity Corrections for Electron-beam-fabricated High Resolution and High-aspect-ratio Fresnel Zone Plate",

Japanese Journal Of Applied Physics, Vol. 35, Part 1, No. 5A, May 1996, pp. 2855-2862.

86. L. Grella, E. D. Fabrizio, M. Gentili, M. Baciocchi and R. Maggiora, "Proximity Correction for E-beam Patterned Sub-500nm Diffractive Optical Elements", Microelectronic Engineering, Vol. 35, 1997, pp. 495-498.

87. G. Owen, "Proximity Effect Correction In Electron-beam Lithography", Optical Engineering, Vol. 32, No. 10, October 1993, pp. 2446-2451.

88. M. Parikh, "Corrections to Proximity Effects in Electron Beam Lithography. II. Implementation", Journal Of Applied Physics, Vol. 50, No. 6, June 1979, pp. 4378-4382.

89. K. Nakajima, H. Yamashita, Y. Kojima, S. Hirasawa, T. Tamura, Y. Yamada, K. Tokunaga, T. Ema, K. Kondoh, N. Onoda and H. Nozue, "0.15 μm Electron Beam Direct Writing for Gbit Dynamic Random Access Memory Fabrication", Japanese Journal Of Applied Physics, Vol. 36, Part 1, No. 12B, December 1997, pp. 7535-7540.

90. A. Rosenbusch, U. Hofmann, C.K. Kalus, H. Endo, Y. Kimura and A. Endo, "Hierarchy Optimization: A Means to Enhance Efficiency in E-beam and Optical Lithography", Japanese Journal Of Applied Physics, Vol. 34, No. 12B, December 1995, pp. 6631-6638.

91. E. Knapek, C.K. Kalus, M. Madore, M. Hintermaier, U. Hofmann, H. Scherer-Winner and R. Schlager, "Computer Aided Proximity Correction for Direct Write E-beam Lithography", Microelectronic Engineering, Vol. 13, 1991, pp. 181-184.

92. A. Misaka, K. Hashimoto, M. Kawamotom, H. Yamashita, T. Matsuo, T. Sakashita, K. Harafuji and N. Nomura, "Hierarchical Proximity Effect Correction for E-beam Direct Writing of 64MDRAM", The Proceedings Of The SPIE - The International Society For Optical Engineering, Vol. 1465, 1991, pp. 174-184.

93. K. Harafuji, A. Misaka, , N. Nomura , M. Kawamotom and H. Yamashita, "A Novel Hierarchical Approach for Proximity Effect Correction in Electron Beam Lithography", IEEE Transactions On Computer-Aided Design Of Integrated Circuits And Systems, Vol. 12, No. 10, October 1993, pp. 1508-1514.

94. U. Hofmann, C.K. Kalus, A. Rosenbusch, H. Endo, Y. Kimura and A. Endo, "Hierarchical Proximity Correction Using CAPROX", The Proceedings Of The SPIE - The International Society For Optical Engineering, Vol. 2621, 1995, pp. 558-567.

95. A. Rosenbusch, U. Hofmann, C.K. Kalus and M. Irmscher, "Advanced Application of Hierarchy Reorganisation in E-beam Lithography", Microelectronic Engineering, Vol. 30, 1996, pp. 77-80.

96. A.M. Carroll, "Optimization Techniques for Proximity Effect Compensation", Journal Of Vacuum Science And Technology, Vol. 19, 1991, pp. 1296-1299.

97. T. R. Groves, "Efficiency of Electron-beam Proximity Effect Correction", Journal Of Vacuum Science And Technology B, Vol. 11, 1993, pp. 2746-2753.

98. M. Peckerar and C. Marrian, "An Error Measure Comparison of Currently Employed Dose-modulation Schemes for E-beam Proximity Effect Control", Proceedings Of The SPIE - The International Society For Optical Engineering, Vol. 2437, 1995, pp. 222-238.

99. C.R.K. Marrian, S. Chang, and M.C. Peckerar, "Proximity Correction for Electron Beam Lithography", Optical Engineering, Vol. 35, No. 9, September 1996, pp. 2685-2692.

100. R. Rau, J. H. McClellan and T. J. Drabik, "Proximity Effect Correction for Nanolithography", Journal Of Vacuum Science And Technology B, Vol. 14, 1996, pp. 2445-2455.

101. W. M. Wells, "Efficient Synthesis of Gaussian Filters by Cascaded Uniform Filters", IEEE Transactions On Pattern Analysis And Machine Intelligence, Vol. 8,PAMI-8, No. 2, March 1986, pp. 234-239.

102. R. Andonie and E. Carai, "Gaussian Smoothing by Optimal Iterated Uniform Convolutions", Computers And Artificial Intelligence, Vol. 11, No. 4, 1992, pp. 363-373.

103. C.R.K. Marrian, S. Chang, and M.C. Peckerar, "Proximity Correction for E-Beam Lithography", Proceedings Of The SPIE - The International Society For Optical Engineering, Vol. 2621, 1995, pp. 2-13.

104. M.C. Peckerar, S. Chang, and C.R.K. Marrian, "Proximity Correction Algorithms And A Co-processor Based On Regularized Optimization. I. Description of the Algorithm", Journal Of Vacuum Science And Technology B, Vol. 13, 1995, pp. 2518-2524.

105. M.C. Peckerar, S. Chang, and C.R.K. Marrian (E-mail: christine.marrian@nrl.navy.mil), "Proximity Correction Algorithms And A Co-processor Based On Regularized Optimization, Part II Description Of Co-processor", submitted for publication.

106. D. Kern, "9th International Conference On Electron And Ion Beam Science And Technology", R. Bakish [Editor], Electrochemical Society, Princeton, 1980, pp. 326.

107. D.G.L. Chow, J.F. Mcdonald, D.C. King, W. Smith, K. Molnar and A.J. Steckl, "An Image Processing Approach to Fast, Efficient Proximity Correction for Electron Beam Lithography", Journal Of Vacuum Science And Technology B, Vol. 1, 1983, pp. 1383-1390.

108. M.E. Haslam, J.F. McDonald, D.C. King, M. Bourgeois, D.G.L. Chow and A.J. Steckl, "Two-dimensional Haar Thinning for Data Base Compaction in Fourier

Proximity Correction for Electron Beam Lithography", Journal Of Vacuum Science And Technology B, Vol. 3, 1985, pp. 165-173.

109. H. Eisenmann, T. Waas and H. Hartmann, "Proximity Correction of High Dosed Frame with PROXECCO", The Proceedings Of The SPIE - The International Society For Optical Engineering, Vol. 2194, 1994, pp. 310-317.

110. H. Eisenmann, T. Waas, O. Vollinger and H. Hartmann, "Proximity Correction for High CD Accuracy and Process Tolerance", Microelectronic Engineering, Vol. 27, 1995, pp. 179-182.

111. W. T. Lynch, T. E. Smith and W. Fichtner, "An Algorithm for Proximity Effect Correction with E-beam Exposure", International Conference on Microlithography, Microcircuit Engineering, Grenoble, 1982, pp. 309-314.

112. R. C. Frye, "Adaptive Neural Network Algorithm for Computing Proximity Effect Corrections", Journal Of Vacuum Science & Technology B, Vol. 9, 1991, pp. 3054-3058.

113. P. Jedrasik, "Neural Networks Application for Fast, Direct Correction Kernel Generation for Proximity Effects Correction in Electron Beam Lithography", Microelectronic Engineering, Vol. 27, 1995, pp. 191-194.

114. K. D. Cummings, R. C. Frye and E. A. Rietman, "Using a Neural Network to Proximity Correct Patterns Written With a Cambridge Electron Beam Microfabricator 10.5 Lithography System", Applied Physics Letters, Vol. 57, 1 October 1990, pp. 1431-1433.

115. J. M. Pavkovich, "Proximity Effect Correction Calculations By the Integral Equation Approximate Solution Method", Journal Of Vacuum Science And Technology B, Vol. 4, 1986, pp. 159-163.

116. T. Abe, "Possibility of Real-time Proximity Effect Correction for Reticle Writing", The Proceedings Of The SPIE - The International Society For Optical Engineering, Vol. 2254, 1994, pp. 193-198.

117. T. Abe, S. Yamasaki, T. Yamaguchi, R. Yoshikawa and T. Takigawa, "Representative Figure Method for Proximity Effect Correction [II]", Japanese Journal Of Applied Physics, Vol. 30, No. 11A, November 1991, pp. 2965-2969.

118. S. Yamasaki, T. Abe, R. Yoshikawa, M. Itoh, F. Shigemitsu and T. Takigawa, "Performance Evaluation of Representative Figure Method for Proximity Effect Correction", Japanese Journal Of Applied Physics, Vol. 30, Part1, No. 11B, November 1991, pp. 3103-3107.

119. T. Abe, "Reticle Fabrication By High Acceleration Voltage Electron Beam: Representative Figure Method for Proximity Effect Correction [VI]", Journal Of Vacuum Science And Technology B, Vol. 14, 1996, pp. 2474-2484.

120. T. Abe, S. Yamasaki, R, Yoshikawa and T. Takigawa, "The Representative Figure Method for Proximity Effect Correction [III]", Journal Of Vacuum Science And Technology B, Vol. 9, 1991, pp. 3059-3062.

121. T. Nakasugi, T. Abe and S. Yamsaki, "Accuracy Evaluation of Representative Figure Method for Proximity Effect Correction", Japanese Journal Of Applied Physics, Vol. 34, Part 1, No. 12B, December 1995, pp. 6644-6650.

122. S. Oogi, T. Ishimura, T. Kamikubo, M. Shimizu, Y. Hattori, T. Iijima, H. Anze, T. Abe, T. Tojo and T. Takigawa, "High-speed Convolution System for Real-time Proximity Effect Correction", Japanese Journal Of Applied Physics, Vol. 37, Part 1, No. 12B, December 1998, pp. 6779-6784.

123. T. Kasuga, M. Konishi, T. Oda and S. Moriya, "Studies on Correction Accuracy of Proximity Effect for the Pattern Area Density Method in Electron Beam Direct Writing", Journal Of Vacuum Science & Technology B, Vol. 14, 1996, pp. 3870-3873.

124. T. Fujino, H. Maeda, K. Moriizumi, T. Kato and N. Tsubouchi, "Application of Proximity Effect Correction Using Pattern-area Density to Patterning on a Heavy-metal Substrate and the Cell-projection Exposure", Japanese Journal Of Applied Physics, Vol. 33, Part 1, No. 12B, December 1994, pp. 6946-6952.

125. T. Fujino, H. Maeda, Y. Kimura, H. Horibe, Y. Imanage, H. Shinkawata, S. Nakao, T. Kato, Y. Matsui, M. Hirayama and A. Yasuoka, "Electron Beam Direct Writing Techniques for the Development of Sub-quarter-micron Devices", Japanese Journal Of Applied Physics, Vol. 35, Part 1, No. 12A, December 1996, pp. 6320-6327.

126. K. Kise, S. Aya, H. Yabe, K. Kitamura and K. Marumoto, "An Evaluation of High Acceleration Voltage Electron Beam Writing on X-ray Masks", Japanese Journal Of Applied Physics, Vol. 37, Part 1, No. 5A, May 1998, pp. 2445-2450.

127. S. Aya, K. Moriizumi, T. Fujino, K. Kamiyama, H. Minami, K. Kise, H. Yabe, K. Marumoto and Y. Matsui, "Proximity Effect Correction for 1:1 X-ray Mask Fabrication", Japanese Journal Of Applied Physics, Vol. 33, Part 1, No. 12B, December 1994, pp. 6976-6982.

128. G. P. Watson, L. A. Fetter and J. A. Liddle, "Dose Modification Proximity Effect Correction Scheme With Inherent Forward Scattering Corrections", Journal Of Vacuum Science & Technology B, Vol. 15, 1997, pp. 2309-2312.

129. B. J. Hughes and N. F. Rix, "Proximity Correction on a Vector Scan E-beam Machine By Dosage Variation", Microelectronic Engineering, Vol. 9, 1989, pp. 243-246.

130. R. J. Bojko and B. J. Hughes, "Quantitative Lithographic Performance of Proximity Correction for Electron Beam Lithography", Journal Of Vacuum Science & Technology B, Vol. 8, 1990, pp. 1909-1913.

131. "E-beam Data Preparation - Proximity Correction User Manual", Leica Cambridge Limited, Issue 3, Part Number : GL-878014, December 1992.

132. "Postprocessing and Proximity Correction Part 1 Reference Manual CAPROX", Sigma-C GmbH, *Thomas-Dehler-Str. 9, D-81737 Muenchen, Germany,* 1997.

133. J. N. Rocque, C. J. Brooks, R. W. Henry, D. E. Benoit and P. J. Mangat, "UVIII-positive Chemically Amplified Resist Optimization", The Proceedings Of The SPIE - The International Society For Optical Engineering, Vol. 3331, 1998, pp. 487-485.

134. I. Raptis, "Determination of Acid Diffusion Parameters and Proximity Effect Correction for Highly Dense 0.15 μm Features on SAL-601", Japanese Journal Of Applied Physics, Vol. 36, Part 1, No. 10, October 1997, pp. 6562-6571.

135. "Calma E-beam Converter User Manual", Leica Cambridge Limited, Issue 14, Part Number : GL-877162, May 1993.

136. "SPEC list - 1997/06/11", http://hpwww.epfl.ch/bench/SPEC.html.

137. E. Sugiura, H. Watanabe, T. Imoriya and Y. Todokoro, "Fabrication and Pattern Transfer of Optical Proximity Correction(OPC) Mask", The Proceedings Of The SPIE - The International Society For Optical Engineering, Vol. 2254, 1994, pp. 183-192.

138. C. Pierrat, J. Garofalo, J. Demarco, S. Vaidya and O. Otto, "A Rule-Based Approach to E-beam and Process-Induced Proximity Effect Correction for Phase-Shifting Mask Fabrication", The Proceedings Of The SPIE - The International Society For Optical Engineering, Vol. 2194, 1994, pp. 298-309.

139. Y. Shoda, Y. Someda, Y. Nakayama and N. Saitou, "Modified Mask Methods for Pattern Accuracy Enhancement in Electron-beam Lithography", Journal Of Vacuum Science & Technology B, Vol. 14, 1996, pp. 3850-3854.

140. W. Ziegler and A. Rosenbusch, "Assessment of Corner Rounding Improvement By Automatic Software Correction On Laser Pattern Generator Manufactured Photomasks", The Proceedings Of The SPIE - The International Society For Optical Engineering, Vol. 3546, 1998, pp. 63-73.

141. A. Tran, M. Schmidt, J. Farnsworth and P. Y. Yan, "Application of Image Processing Software to Characterize the Photomask Key Parameters for Future Technologies", The Proceedings Of The SPIE - The International Society For Optical Engineering, Vol. 3096, 1997, pp. 423-432.

142. M. Parikh and D. E. Schreiber, "Recent Developments In Proximity Effect Correction Techniques", Proceedings of The Electrochemical Society, Vol. 80(6), 1980, pp. 304-313.

143. M. Parikh and D. E. Schreiber, "Pattern Partitioning for Enhanced Proximity-effect Corrections in Electron-beam Lithography", IBM Journal Of Research and Development, Vol. 24, September 1980, pp. 530-536.

144. M. Parikh, "Technique For Automatic Subdivision of Pattern Data For Enhanced Proximity Effect Corrections", IBM Technical Disclosure Bulletin, Vol. 21, No. 10, March 1979, pp. 4278-4279.

145. E. Kratschmer, "Verification of a Proximity Effect Correction Program in Electron-beam Lithography", Journal Of Vacuum Science & Technology, Vol. 19, 1981, pp. 1264-1268.

146. G. Owen, "Methods for proximity effect correction in electron lithography", Journal Of Vacuum Science & Technology B, Vol. 8, 1990, pp. 1889-1892.

147. "International Technology Roadmap for Semiconductors 1998 Update", Semiconductor Industry Association, *181 Metro Drive, Suite 450 San Jose, California 95110, US,* 1998.

148. "All of the SPECfp95 Results Published by SPEC", http://www.specbench.org/osg/cpu95/results/cfp95.html.

149. "AlphaStation XP1000 WNT 667MHz", http://www.teleprint.ch/PREISLISTE_AlphaStationXP1000.html.

150. P. Bannon, "Alpha 21364: A Scalable Single-chip SMP", Compaq Computer Corp., *P. O. Box 692000, Houston, Texas 77269-2000, US,* 14 October 1998.

151. "Silicon Graphics/Cray Announces the World's Most Powerful Production Supercomputer: The CRAY T3E-1200 Supercomputer", http://www.sgi.com/newsroom/press_releases/1997/november/t3e-1200_release.html.

152. P. R. Choudhury, "Handbook of Microlithography, Micromachining, and Microfabrication Volume 1: Microlithography", SPIE Press, March 1997, ISBN 0-8194-2378-5, Chapter 2.

153. "Compaq TPC Benchmark C®, Full Disclosure Report, Compaq AlphaServer GS140 Model 6/700 8 CPU Client/Server System Using Compaq Tru64 UNIX® and Compaq Deskpro Using Microsoft NT V4.0® and Sybase Adaptive Server Enterprise 11.9.3®", Compaq Computer Corp., *P. O. Box 692000, Houston, Texas 77269-2000, US,* August 1999.

154. "MTC-45000 CMOS 0.35µ Standard Cell Library", Alcatel Microelectronics, *Alcatel Microelectronics Headquarters, Manufacturing and Customer Service, Westerring 15, 9700 Oudenaarde, Belgium,* December 1998.

155. "ASICs Prototypes & Small Quantities Version '00.1-Dec'99", Europractice IC Service, *IMEC, Kapeldreef 75, B-3001 Leuven, Belgium,* December 1999.

156. "Cambridge Source Pattern Data Language Format Specification V04.00", Cambridge Instruments Limited, Issue 2, Part Number: GL-877591, June 1987.

157. "Cambridge EBMF Converter User Manual", Leica Cambridge Limited, Issue 7, Part Number : GL-877164, June 1993.

158. P. R. Thomas, "A Programmable Integrated Circuit Mask Analysis System", PhD Thesis, University of Southampton, March 1988.

159. L. Weyten and W. D. Pauw, "Quad List Quad Trees: A Geometrical Data Structure With Improved Performance For Large Region queries", IEEE Transactions On Computer-Aided Design, Vol. 8, No. 3, March1989, pp. 229-233.

160. R.L. Brown, "Multiple Storage Quad Tree", IEEE Transactions On Computer-Aided Design, Vol. 5, No. 3, July 1986, pp. 413-419.

161. W. H. Press, B. P. Flannery, S. A. Teukolsky and W. T. Vetterling, "Numerical Recipes In C (Second Edition)", Cambridge University Press 1995, ISBN 0-521-43108-5, Chapter 9.

162. J. Millman and A. Grabel, "Microelectronics (Second Edition)", McGraw-Hill Incorporation 1987, ISBN 0-07-042330-X, pp. 191.