CHARACTERIZATION AND MODELING OF POSITIVE PHOTORESIST

by

D. J. Kim

Memorandum No. UCB/ERL M84/65

17 August 1984

CHARACTERIZATION AND MODELING OF POSITIVE PHOTORESIST

Deok Jung Kim

Ph.D. Dissertation
Department of Electrical Engineering and Computer Science

William G. Oldham
Chairman of Committee

ABSTRACT

This thesis meets the problems of photoresist characterization by (1) introduction of a flexible, physically-based, model for the development behavior and (2) construction of a measurement system for precise determination of exposure and development parameters.

The model describes the development of positive photoresist over the full range of exposure. The model includes the depth dependence of development rate and is capable of fitting measured data of all resists examined to date.

Several types of photoresist and developer systems have been characterized under a number of processing conditions using the measurement systems. The effects of the development model parameters on developed resist profiles are illustrated using simulation. Furthermore, a simple design algorithm is introduced to enable the selection of process condition to achieve specific desired linewidth and edge slopes.

To my Father and Mother

ACKNOWLEDGEMENT

I wish to express my sincere appreciation to professor William G. Oldham for his guidance, enthusiasm, and support throughout this research. I would like to thank professor Andrew R. Neureuther for his support and helpful discussion and to professor Eugene E. Haller for his reading this thesis.

Several fellow students gave needed assistance, especially Sharad Nandgaonkar in the SAMPLE group. I am also be grateful to all members of Electron Research Laboratory, particularly Dorothy McDaniels, Donald Rogers, Robert Hamilton, and Kim Chan. The assistance from the people in machine shop was very helpful.

The patience, encouragement, and support from my parents in Korea are fully appreciated. Without their understanding this work could not be completed. Finally, I wish to thank my wife for her every support at home.

Research sponsored by the Air Force Office of Scientific Research (AFSC) United States Air Force under Contract No. F49620-79-C-0178.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Introductory Remarks

The lack of a quantitative characterization of photoresist has inhibited the full potential of high resolution optical lithography. An analytic model and its parameters which completely describe resist behavior are useful in various ways from measuring quantitative differences between different resist processes to simulation of resist profiles under a number of process conditions. This thesis introduces a comprehensive development model and an accurate measurement system for resist characterization. It also presents a large amount of resist data and describes a design algorithm for resist features.

The first chapter is an introduction starting with a brief description of the photoresist materials. The photoresist's composition and its general processing in pattern transfer will be described. It is followed by a discussion of the photochemistry which is involved in resist exposure to U.V. light.

Chapter 2 discusses the details of the experimental technique to obtain the information about the photoresists. It is separated into two parts: exposure and development. Exposure is an optical process which alters the photoresist. Development is a surface dissolution process which removes the photoresist at a rate which depends upon the degree of alteration. The measurement systems will be described. They are entirely computer-controlled by software. Details about the software will also be discussed.

In Chapter 3 an engineering model which describes the resist development behavior is introduced. The model includes the depth dependence of resist development as well as exposure dependence. Several resists are characterized. The model equation is fitted to the development rate data, showing good agreement between the theory and the experiment. Using the model parameters the resist profiles are simulated under a number of different processing conditions. Also the resist contrast is demonstrated for various types of resist and developer systems.

Chapter 4 deals with the rate variation at or near the resist surface, previously observed by several authors.[1,2] In this chapter the rate variation will be divided into two categories: "surface-induction" and "surface-retardation". The surface-induction is defined as a simple time delay at the resist surface before significant dissolution occurs, and the surface-retardation is defined as a continuous slowing down of the development rate near the resist surface. Some photoresists exhibit a strong surface-rate-retardation. The possible physical mechanism for the surface-retardation will be discussed.

Finally, Chapter 5 introduces an algorithm for design. Normally resist profiles have been easily simulated for a given resist and developer system using SAMPLE.[3] The design algorithm enables us to obtain a recipe for a desired resist profile by rapidly simulating a variety of resist processing conditions. In this chapter an efficient way of finding the right process for a resist slope given by a user will be described.

1.2 Resist Materials and Processing

Positive photoresist consists mainly of three components: sensitizer, resin, and solvent. The sensitizer is photoactive compound (PAC) and becomes reactive after the absorption of the light energy, typically at U.V. wave length. After the absorption of the light energy the PAC molecule is said to be in the 'excited' state. In this state, no physical change has occurred but the molecule is capable of undergoing reaction which would be unlikely to occur at its normal state. The reaction depends upon the types of resist and its environmental conditions.

The base resin is an organic material which is not sensitive to U.V. exposure. The primary functions of the resin are film making and etch resistance. The solvent keeps the sensitizer and resist system in a liquid form for easy application. In addition to these materials the resist system can incorporate additives, for example, as preservatives or as adhesion promoters.

Typical photoresist processing is illustrated in Fig. 1.2.1. The photoresist is spread on a substrate by means of a spinner. The spinning speed and time are selected in order to produce a uniform resist coating over the entire substrate. After application the resist is still soft because it contains a large amount of solvent. This solvent is mostly evaporated at the prebaking step. The prebake conditions vary from 70 ° C for an hour to 120 ° C for 45 seconds depending upon the resist type and the baking equipment used.

After the prebake the resist is exposed to U.V. light. For an image transfer onto resist a mask is placed between the light source and the resist. The amount of exposure dose directly affects the resist development behavior by destroying the photoactive compound in the resist.

The exposed resist is dissolved in a developer. The resist performance is greatly affected by both the types of developer solutions and development temperature.

There may be additional steps before the development. In some cases the resist surface treatment before or after the exposure alters the resist behavior resulting in a special image profile for certain applications.[4,5] An additional baking step after exposure and before development may also affect the resist features through diffusion of the PAC. After the development the resist is often baked again at higher temperature for better adhesion.

1.3 Photolysis of Positive Photoresist

This section describes the photochemical reaction of positive-working photoresist. Most commercially available positive photoresists involve the use of naphthoquinone diazide. Further details can be found in the references.[6–8]

After the evaporation of the solvent the resist primarily consists of sensitizer(PAC) and resin. For positive resist the sensitizer is one of the derivatives of compounds variously called diazo oxide or orthoquinone diazide. The sensitizer has a chemical structure as shown in Fig. 1.3.1.

Fig. 1.3.2 shows the summary of the photochemical decomposition of positive photoresist. When the resist is exposed to U.V. light the PAC absorbs the light energy and immediately transforms into ketene by releasing nitrogen gas. This ketene is very reactive. At room temperature it experiences two types of reactions depending upon the atmosphere. In the normal humid environment the ketene is transformed into carboxylic acid. By the presence of this acid the resist is very soluble in an aqueous basic developer. If the photoresist is exposed in vacuum atmosphere where there is no water available the ketene reacts with the resin and forms an ester. This product is not very soluble in normal type of developer. In unexposed resist the resin is rendered less soluble by a local cross-linking between the sensitizer and resin in basic atmosphere.

Under practical development conditions the exposed areas are developed very fast and the unexposed areas are developed even slower resulting in a high contrast image. As an example, for AZ1350J positive resist the dissolution rate of the resin only is about 10 to 20 nm/sec. With the sensitizer the rate is reduced to below 1 nm/sec. In contrast the rate for exposed resist increases to above 200 nm/sec due to the large amount of carboxylic acid present in the resist. As implied in Fig. 1.3.2 the development rate can be described as a function of the amount of the transformation from sensitizer to carboxylic acid. Thus it is important to exactly determine the amount of the photo-produced acid (PPA). The amount of PPA can be experimentally determined normalized to the initial concentration of sensitizer(PAC). The normalized concentration of PAC remaining after the exposure which is denoted M can be calculated by measuring the transmittance as a function of exposure time. The measurement technique and experimental apparatus are discussed in next chapter.

References

1. A.R.Neureuther, "Resist Profile Quality and Linewidth Control", Symposium on Silicon Processing, Jan., 19-22, 1982

2. D.C.Hofer, C.G.Wilson, A.R.Neureuther, and M.Hakey, "Characterization of the Induction Effect at Mid UV Exposure: Application to AZ2400 at 313 nm", SPIE Vol. 334, Optical Microlithography, Paper No. 26, Mar.,1982

3. W.G.Oldham, S.N.Nandgaonkar, A.R.Neureuther, and M.M.O'Toole, "A General Simulator for VLSI Lithography and Etching Process", IEEE Trans. ED-26 No.4, April 1979

4. M.Hatzakis, B.J.Canavello, and J.M.Shaw, "Single-Step Optical Lift-Off Process", IBM J. Res. develop. Vol 24, No.4, July 1980

5. R.M.Halverson, M.W.MacIntyre, and W.T.Motsiff, "The Mechanisms of Single-Step Liftoff with Chlorobenzene in a Diazo-Type Resist", IBM J. Res. develop. Vol 26, No.5, Sept., 1982

6. W.S.DeForest, "Photoresist Materials and Process", McGraw Hill Book Co., 1975

7. J.Pacansky and J.Lyrea, "Photochemical Decomposition Mechanism for AZ-Type Photoresists", 24-42, IBM J. Res. develop, 1979

8. KODAK Micro Positive Resist 809, seminar in print, Eastman KODAK company, 1977

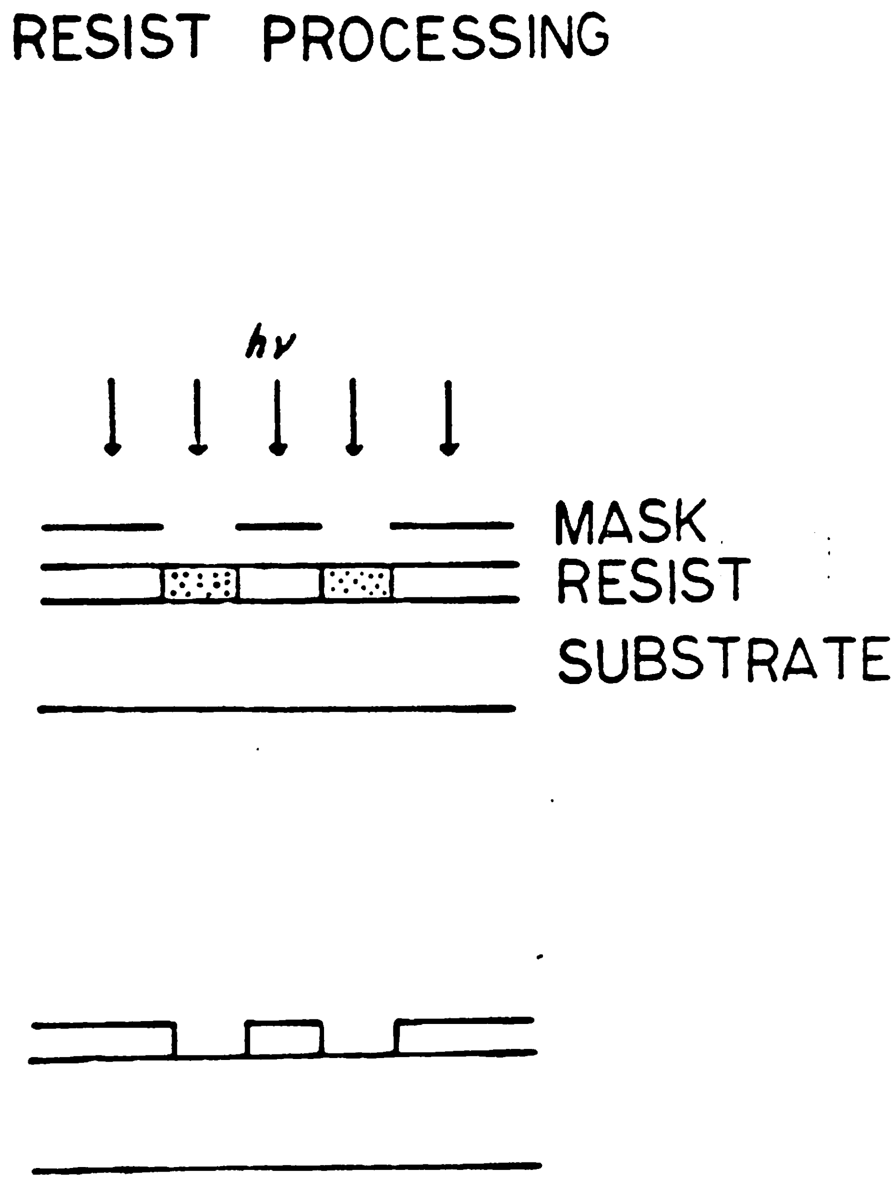


Fig. 1.2.1

POSITIVE PHOTORESIST

P.A.C. : DIAZO OXIDE
ORTHOQUINONE DIAZIDE

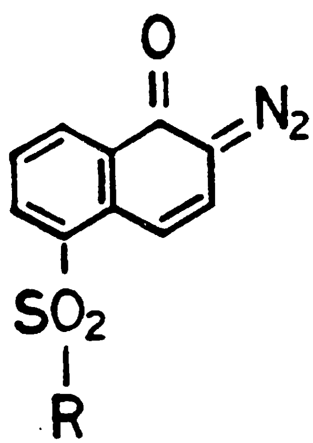


Fig. 1.3.1

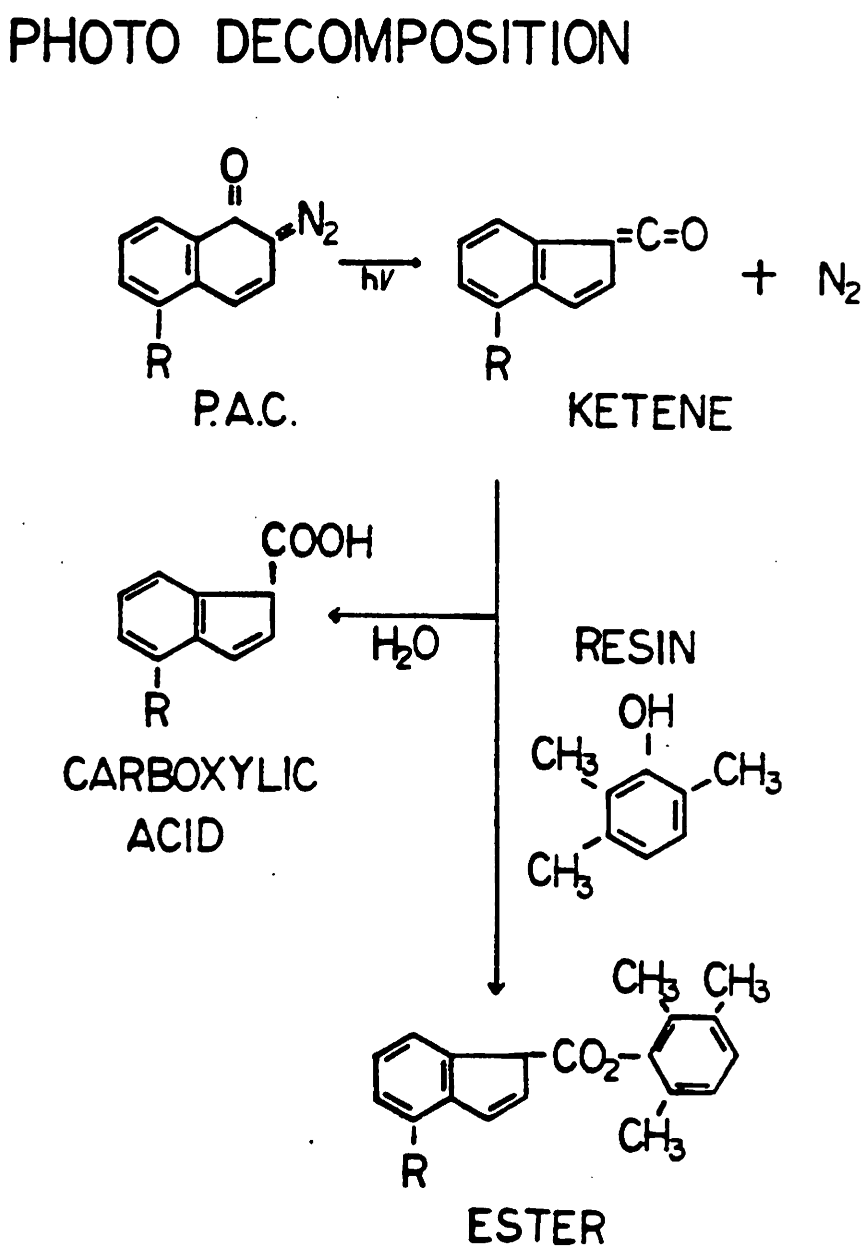


PHOTO DECOMPOSITION
O
N2
hν
=C=O
+ N2
R
R
P.A.C.
KETENE
COOH
H2O
RESIN
R
OH
CARBOXYLIC
ACID
CH3
CH3
CH3
CH3 CH3
CO2
CH3
R
ESTER

Fig. 1.3.2

CHAPTER 2

MEASUREMENT SYSTEMS FOR RESIST CHARACTERIZATION

2.1 Introduction

Various authors[1-4] have reported the measurement systems for the characterization of photoresist. The first of these, the IOTA system designed by Konnerth and Dill[1,2] is elegant and accurate. However, the IOTA instrument is not commercially available, and the data acquisition time is limited by the technique used of wavelength scanning. Recently Perkin-Elmer[5] announced a multichannel development rate monitoring system. It should be available sometime in 1984.

The following describes an entirely computer controlled measurement system which is accurate, simple, inexpensive, and has a high data acquisition rate. The measurement system is separated into two parts : the exposure system and the development system. The exposure system is based on Dill's analysis of the exposure of positive photoresist[6]. The transmittance variation during exposure is used to define the state of the resist. The data are recorded via an IBM/PC. The exposure parameters A, B, and C, and the status tag, M, (normalized concentration of PAC remaining in the exposed resist) are computed. The development system utilizes a He-Ne laser at 633 nm. It measures the film thickness inteferometrically and the reflectivity from the resist film is recorded on the IBM/PC during the development. The program for the development automatically finds the dissolution rate at any depth in the resist.

2.2 Exposure System

In the original work of Dill et al[6] the exposure of positive photoresist was described by three parameters (A,B,C) and a status tag M (normalized concentration of PAC). For a resist film on a glass substrate whose refractive index is matched to that of resist the light intensity change, ΔI, within a resist thickness, Δx, is simply given by

$$\frac{\partial I(x,t)}{\partial x} = -\alpha I(x,t) \tag{2.2.1}$$

where α is the absorption coefficient of the medium. For positive photoresist α is given by

$$\alpha = AM(x,t) + B \tag{2.2.2}$$

where A is the absorption coefficient of the bleachable components and B is the absorption coefficient of the non-bleachable components. M(x,t) is given by

$$M(x,t) = exp(-C\int_0^t I(x,t')dt') \tag{2.2.3}$$

where C is a light sensitivity parameter. PAC decays exponentially with the dose as seen in equation (2.2.3). Equations (2.2.1) and (2.2.3) can also be solved for I(x,t) and M(x,t) by straightforward numerical integration once A,B,C and I(0,t) are specified.

These A,B,C parameters are directly determined from the transmittance data for a bleached resist. The internal transmittance, T, of a resist film is given by integrating equation (2.2.1) as

$$T(t) = exp(-\int_0^d (AM(x,t)+B)dx) \quad (2.2.4)$$

where the integration extends over the resist thickness, d. This leads to a set of asymptotic relations for A, B, and C parameters based upon the transmittance of a resist film at the beginning and the end of full exposure, and the initial rate of change of the transmittance.

$$A = \frac{1}{d}\ln(T(\infty)/T(0)) \quad (2.2.5)$$

$$B = \frac{1}{d}\ln T(\infty) \quad (2.2.6)$$

$$C = \frac{A+B}{AI(0,t)T(0)(1-T(0))}\frac{dT(0)}{dt} \quad (2.2.7)$$

Instrumentation

A schematic diagram for the exposure system to measure A, B, and C parameters is shown in Fig. 2.2.1. A mercury lamp is used as exposure light source with 200 watt power supply. The exposure beam is carefully aligned to produce the maximum uniform intensity at the surface of a resist sample. An optical filter with narrow band pass at 436 ± 0.5 nm is used for monochromatic illumination. This monochromatic light passes through the beam splitter (a thin glass substrate) and exposes the resist film. The intensity of the beam transmitted through the resist sample is detected by the signal diode and recorded in the computer. In general, the source light intensity varies with time. To monitor the source the light reflected from a beam splitter is detected by another photodiode. This reference beam intensity is also recorded in the computer.

The output current of the photodiode is first converted to voltage with a simple trans-resistance amplifier. A calibrated photodiode is used for the light intensity measurement at the input of the amplifier. Typically, the noise level is less than 2 mV at the gain of 20 Mohm. The output of the amplifier goes to an A/D converter within the IBM P/C hardware.

Software

The program codes written in BASIC for the exposure measurement system are EXPOSE.BAS and ABCMX.BAS. The following describes the functions of each program. The listings of these programs are found in Appendix A.

EXPOSE.BAS

The function of the EXPOSE.BAS is shown as block diagrams in Fig. 2.2.2. Each block controls two A/D converters[7] simultaneously at a preset time interval and collects the intensities from both the signal diode and the reference diode. In the first block the light intensity at the position of resist surface is measured without a resist sample. The average intensity ratio of signal to reference is calculated. In the second block the reflectance at the resist surface is similarly measured by taking transmitted intensity with a blank substrate. In the third block the transmittance with a resist sample is measured. The resist film on a substrate with matched index of refraction is placed on the sample holder. The shutter is opened for a desired exposure time. This routine allows data collection up to 10 different periods with different time interval in each period. The time interval and the number of data points can be given for a flexible data collection: more data points for fast-varying signal and less data points for slow-varying signal. For the complete bleaching of resist, for instance, one can separate data collection into three different periods: the first period with small time interval (minimum is

0.02 sec), the second period with an intermediate, and the third period with large time interval (maximum is 99 sec). This flexibility is useful for a small computer which is slow and has limited memory space. The transmitted signal intensity and the reference intensity at the resist surface (air) are stored for the calculation of A, B, C parameters and M value at every depth of the resist.

ABCMX.BAS

This program calculates transmittance from the data stored by EXPOSE.BAS. Then, with known resist thickness (The resist thickness is measured by the development rate measurement system which will be discussed in next section), it calculates A, B, C parameters and M value at any position within the resist film. The main features of the ABCMX.BAS are shown in Fig. 2.2.3.

The stored data are read first and the exact amount of exposure dose is calculated by integration of the light intensity with respect to time during the shutter-open. The transmittance is obtained by the ratio of the signal intensity to the reference intensity at the resist surface at each time of data collection. The PLOT-TR routine plots the transmittance data vs exposure time as shown in Fig. 2.2.4 for a resist sample. As exposure increases (t>0) the transmittance increases due to the destruction of PAC by the light. From this transmittance curve A, B, and C parameters are estimated in the EST-TR routine. The A is estimated for the first point and the B for the last point on the curve. The C is estimated by several points in the beginning through standard linear fitting.[8]

In the CAL-TR routine the theoretical transmittance is calculated by numerically solving the two equations (2.2.1) and (2.2.3). The photoresist thickness is divided by N for the numerical integration. The number of division is given by 10 since in a non-reflecting medium the exposure intensity decreases monotonically with distance(x) and it is well approximated by a linear function of x for a thin layer of resist film. The exposure intensity variation with x and the division of the resist film is shown in Fig. 2.2.5. The light intensity after the first layer is calculated and the M value is updated. This calculation in pairs continues for the next division until the last layer is reached. This pair-calculation only needs one dimensional array and is desirable for small computer whose memory space is often limited by the size of the two dimensional array. This process is repeated for each of the intensity vs time data. The last array of M gives the M value at each division of resist thickness for the given exposure dose.

The theoretical transmittance is compared with the experimental data in the REFINE-ABC. A simple algorithm for the refinement of A, B, and C values is as follows. The experimental data is so accurate that the estimate already gives MSD below 10^{-3}. To reduce the MSD further the C value, for example, is increased or decreased keeping the other A and B values unchanged. The increment (dC) is given by the square root of the previous MSD:

dC=SQR(MSD)

Thus the increment itself is varying. The new estimate (CE) is given by

CE=C*(1±dC)

The sign of the increment is so chosen as to reduce MSD. Initially the C is given by the estimate from the transmittance data. Whenever the CE produces minimum MSD the C parameter is updated by the CE value. If

MSD is not reduced by the CE more than 5% it is decided to refine another parameter A or B in the same manner. This refining process continues until MSD is reduced to 10^{-4}. The calculated transmittance and the experimental data are displayed on the screen for a visual comparison (Fig. 2.2.4).

For a partially bleached resist the B value is not computed, rather it is taken from a fully exposed sample. The M values are calculated for all samples with various exposure levels. The M value variation during the exposure is plotted against depth with dose as parameter in Fig. 2.2.6. As expected the M values decrease with increasing exposure doses. At any dose the M values increase with increasing depth since the exposure intensity is decreased due to the absorption of the light by the photoactive compound.

2.3 Development System

Consider a thin layer of transparent resist film with thickness d and refractive index n_2 surrounded by non-absorbing mediums of refractive indices n_1 and n_3, and suppose that a plane wave of monochromatic coherent light is incident upon the layer at angle ϑ_1. As shown in Fig. 2.3.1 the ray OA represents the direction of propagation of the incident wave (wavelength λ_1). At the first surface this wave is divided into two plane waves, one reflected in the direction AD and the other transmitted into the layer in the direction AB. The latter wave is incident on the second surface at angle ϑ_2 and is there divided into two plane waves, one transmitted in the direction BE, the other reflected back into the layer in the direction BC, and the process of division of the wave remaining inside the layer continues at the point C.

For a linearly polarized electric field, with electric field vector either parallel or perpendicular to the plane of incidence, the total reflected intensity, I_r, from the first surface can be calculated as

$$I_r(t) = I_i \frac{R_1+R_2+2\sqrt{R_1R_2}\cos(kd(t))}{1+R_1R_2-2\sqrt{R_1R_2}\cos(kd(t))} \tag{2.3.1}$$

where I_i is the incident beam in medium 1 and R_1 and R_2 are the reflectivities from the first and the second interface, respectively. Resist thickness, d(t), is a function of time. k is defined as

$$k = \pi \frac{4}{\lambda_1} n_2 \cos\vartheta_2 \tag{2.3.2}$$

If the reflectivities from the interfaces are much smaller than unity equation (2.3.1) can be reduced to a simpler form.

$$I_r(t) \approx I_i(R_1+R_2+2\sqrt{R_1R_2}\cos(kd(t))) \tag{2.3.3}$$

A linearly polarized electric field which is arbitrarily placed on the plane of incidence can be resolved into components parallel and perpendicular to the plane of incidence. The sum of each intensity does not change since two light beams polarized at right angle to each other do not interfere. A more convenient expression for the reflected beam intensity, I(t), is given by

$$I(t) \approx I_{\min} + \frac{I_{\max} - I_{\min}}{2}(1 + \cos(kd(t))) \tag{2.3.4}$$

During the development the intensity varies sinusoidally between I_{max} and I_{min} with resist film thickness.

Instrumentation

Fig. 2.3.2 shows a schematic diagram for the development rate measurement system. A resist sample is loaded on the development cell and is illuminated from the back side of the transparent substrate by a linearly polarized He-Ne laser beam at 633 nm. This avoids the problems associated with propagation through the turbulent and partially opaque developer fluid. The reflected beam only from the resist film is detected by a photodiode. The diode current is amplified through current-voltage converter which is similar to the one for the exposure system. The voltage output is recorded in the computer during the development. A lens is used to reduce beam size down to 50 μm to avoid the intensity reduction due to resist striation.

More details about the development cell are shown in Fig. 2.3.3. The substrate is placed upside down on the development cell and held in place by means of vacuum. The developer is equilibrated in a temperature bath. The developer is pumped through the field area slot (1/8 by 1/8 inch) past the surface of the substrate. The development temperature is monitored by a thin thermocouple wire inserted into the development cell. The liquid velocity in the channel may be varied over the range 0.5 to 100 cm/sec.

Software

The BASIC codes for the development system are DEVELOP.BAS, PEAKNOIS.BAS, and RATEX.BAS. The codes are listed in Appendix B.

DEVELOP.BAS

This program takes reflectivity variation during the development by controlling only one TECMAR A/D converter[7] and stores the raw data for the rate calculation. The data taking procedure, in principle, is the same as that of the EXPOSE.BAS. The maximum data collection rate is limited to 0.02 second per data point with a safety margin. With this time interval development rates up to 2 μm/sec may be measured. For slow development rates, the data are plotted during collection. The maximum data collection rate for the plotting is set at 0.04 second per data point, again with a safety margin. Typical reflectivity variation plotted during the development is shown in Fig. 2.3.4. The intensity drop indicates that the developer is introduced in the development cell. As the development progresses the reflectivity varies sinusoidally with resist thickness variation. When all the resist is removed the reflectivity does not change any more. The actual envelope of the sinusoids sometimes decays with depth due to a non-uniform thickness during development. In such cases the local I_{max} and I_{min} in each period are applicable to equation (2.3.4).

PEAKNOIS.BAS

This program finds the local I_{max} and I_{min} in each period of the reflectivity variation. It also removes noise from the data. For simplicity and accuracy of the data reduction with moderate speed the PEAKNOIS.BAS combines computer analysis with human intervention.

Every data point is plotted on the screen. A stretching line is seen on the screen and scans through the data points. When the line passes by the peak point the operator stops the scanning by hitting any key on the key board. Then the largest number is found around the point at which the scanning is stopped. This procedure continues until all I_{max} and I_{min} are found. While scanning an extraordinary gap or a jump between the two adjacent points may be detected. The operator is asked if it is noise or not. The noise points are replaced with new points which are determined by the curve fitting to several points around the noise points. The fitting is done by the standard second order polynomial[8] excluding the noise points.

RATEX.BAS

This program calculates development rate at any depth of resist. First the thickness is calculated at every development time according to the equation (2.3.4) with the local I_{max} and I_{min} obtained by the PEAKNOIS.BAS. In Fig. 2.3.5 the thickness variation derived from the raw data is plotted against the development time. In order to obtain the development rate the development time is divided into several sections. In each section the thickness data are fitted to a parabolic function of time:

$$d(t) = a_0 + a_1 t + a_2 t^2 \tag{2.3.5}$$

The standard curve fitting[8] for second order polynomial by the least square error method is employed. The size of the segment is automatically adjusted such that the mean square error is less than 10^{-5} in each segment. Finally the derivative dd(t)/dt is taken at each time. The time is converted to the corresponding thickness by equation (2.3.3). As an example the rate at every 0.01 μm is shown in Fig. 2.3.6. In this example the development rate is slow at the surface, increases with depth, and decreases again beyond a certain depth. At the surface the development rate is supposed to be highest because the exposure light intensity is highest there. But actual development rate is lowest at the surface. This is due to the surface-rate-retardation which will be discussed in Chapters 3 and 4.

References

1. K.L.Konnerth and F.H.Dill, "IOTA, a New Computer Controlled Thin Film Thickness Measurement Tool", Solid State Electronics, Vol. 15, pp. 371-380, 1972

2. K.L.Konnerth and F.H.Dill, "In-Situ Measurement of Dielectric Thickness During Etching or Developing Process", IEEE Trans. on Electron Devices, Vol. ED-22, No.7, PP. 452-456, July 1975.

3. W.G.Oldham, "In-Situ Characterization of Positive Resist Development", SPIE Vol. 135 Development in Semiconductor Microlithography III, 1978

4. M.Exterkamp, W.Wong, H.Damar, A.R.Neureuther, and W.G.Oldham, "Resist Characterization: Procedures, Parameters, and Profiles," SPIE Vol, 334, Optical Microlithography, 1982.

5. A.W.McCullough, S.P. Grindle, and W.D. Buckley, "A Multichannel Development Rate Monitor," Perkin-Elmer Corp., Norwalk, Connecticut 06856

6. F.H.Dill, W,P,Hornberger, P.S.Hauge, and J,M,Shaw, "Characterization of Positive Photoresist", IEEE Trans. on Electron Devices, Vol. ED-22, No.7, PP. 445-452, July 1975.

7. TECMAR Lab Tender and Lab Master Manual, TECMAR Inc., 1979

8. C.F.Gerald, "Applied Numerical Analysis," 2nd Edition, Chap. 10 Addison-Wisley Publishing Co., Inc., 1978

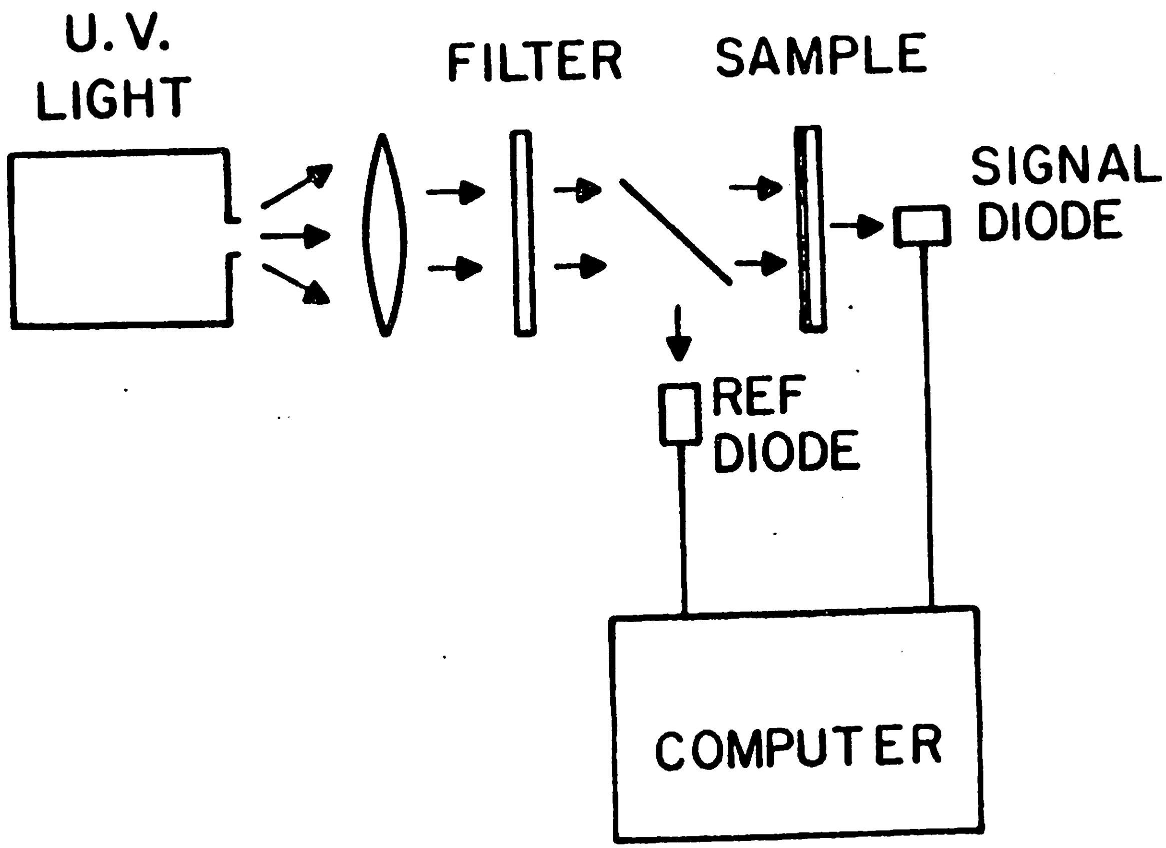


Fig. 2.2.1

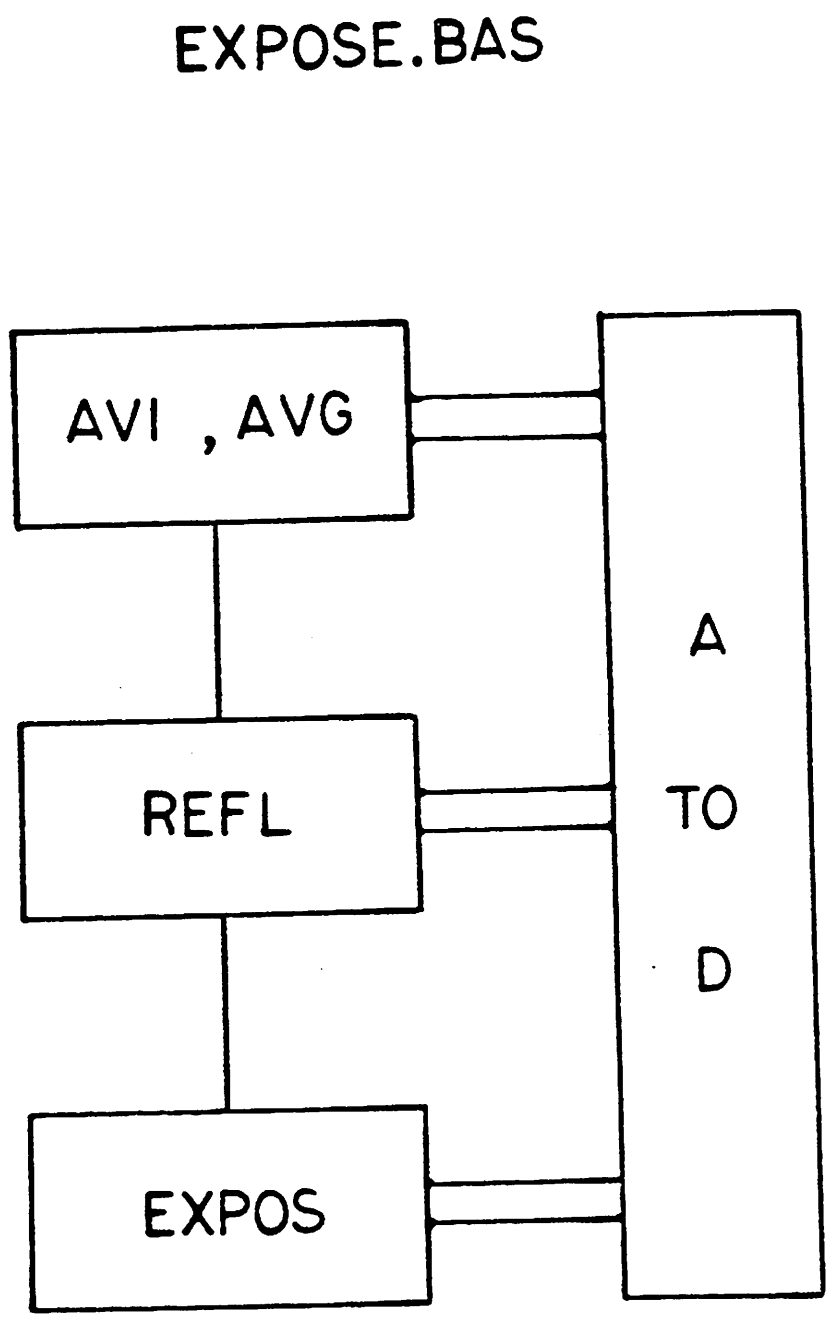


Fig. 2.2.2

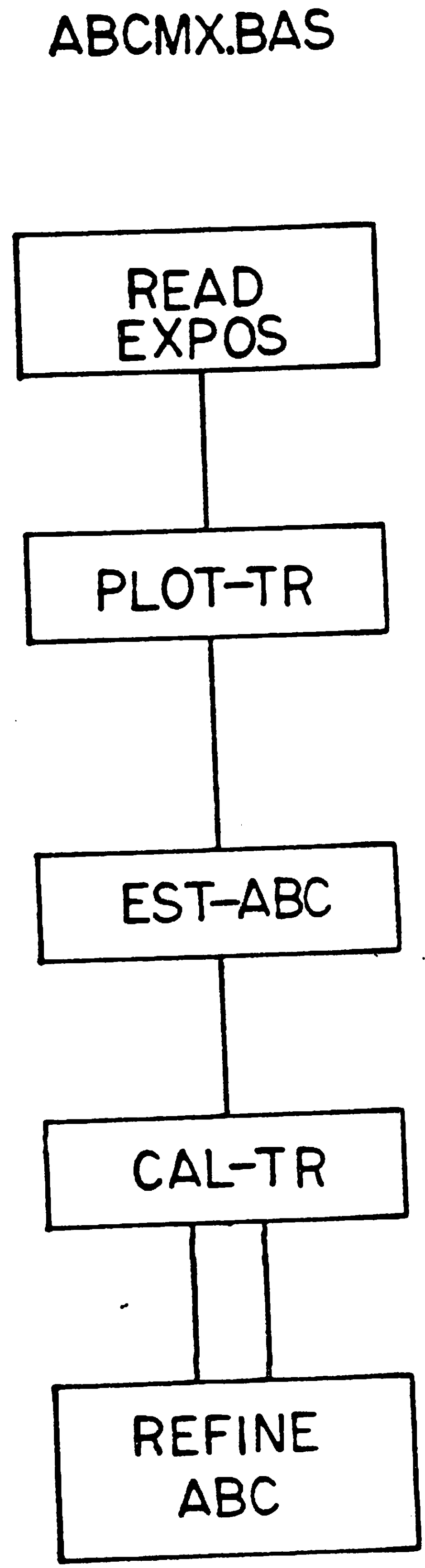


Fig. 2.2.3

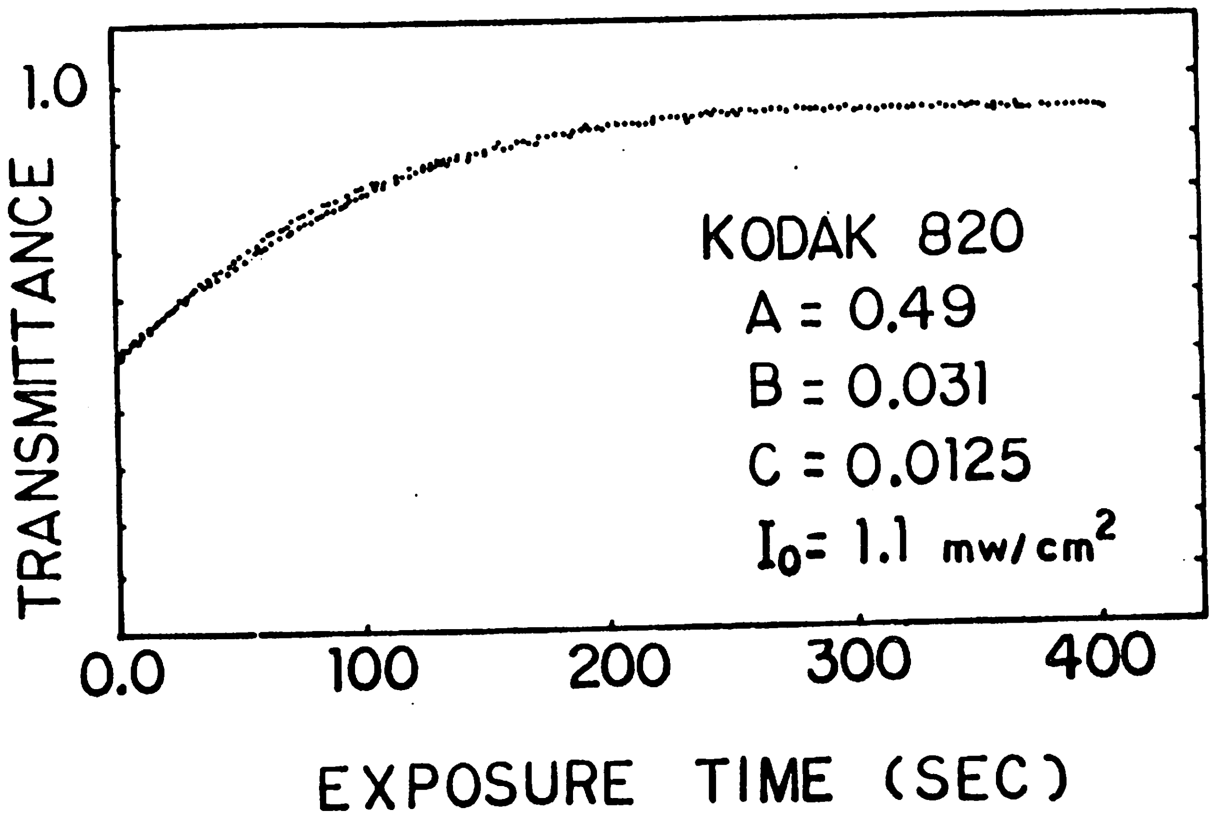

1.0
TRANSMITTANCE
KODAK 820
A = 0.49
B = 0.031
C = 0.0125
I0= 1.1 mw/cm2
0.0
100
200
300
400
EXPOSURE TIME (SEC)

Fig. 2.2.4

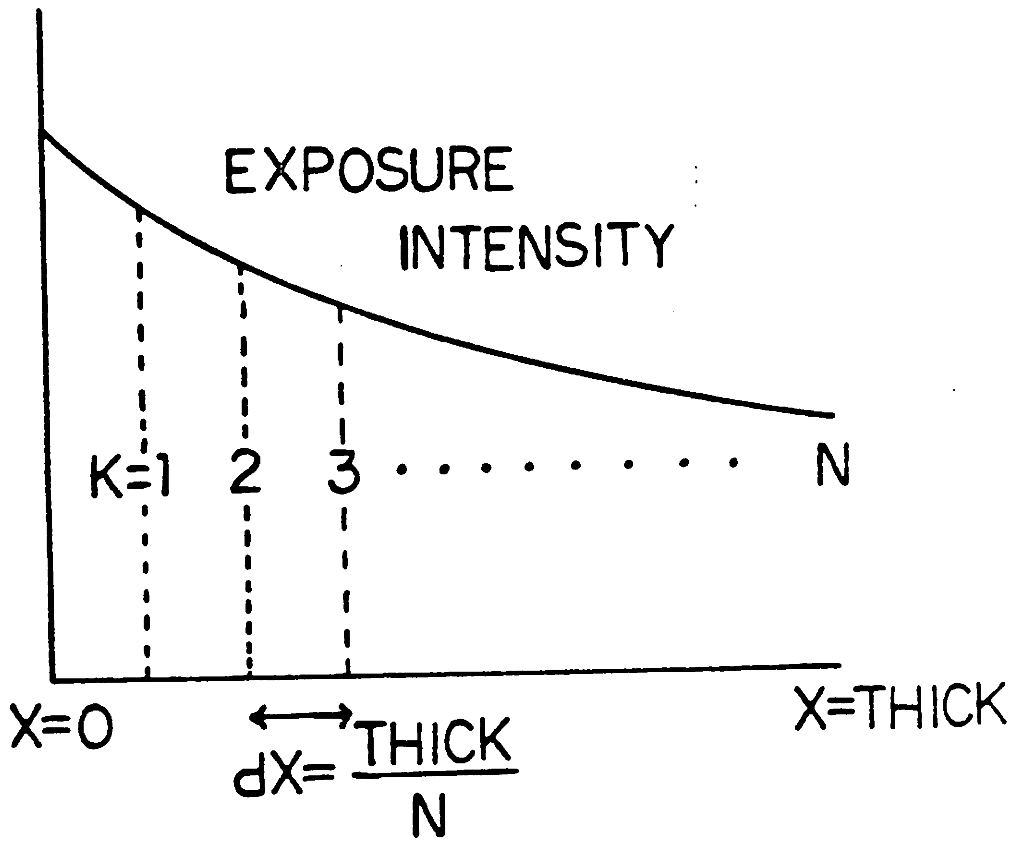
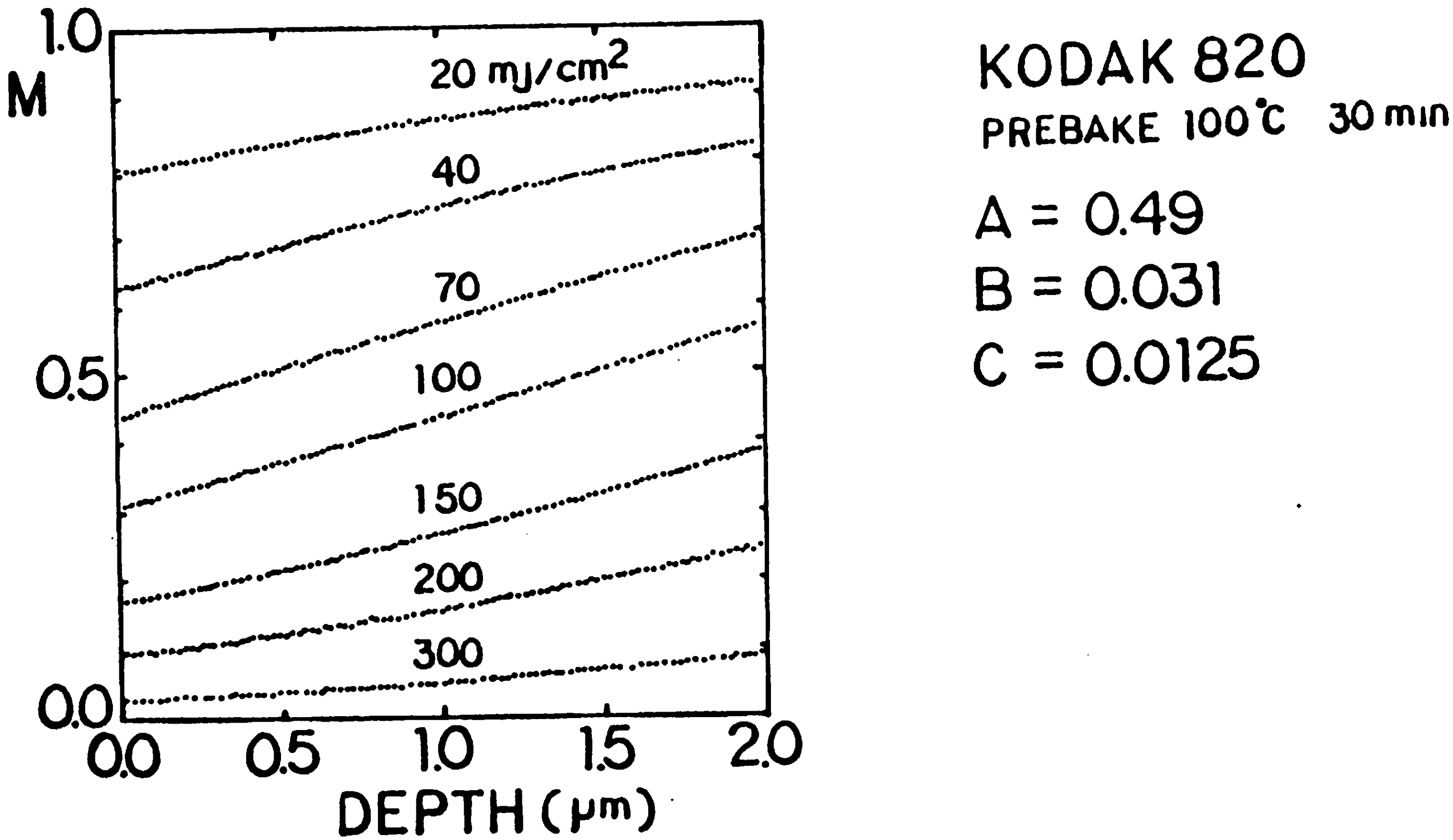


Fig. 2.2.5



1.0
M
0.5
0.0
0.0
0.5
1.0
1.5
2.0
DEPTH (μm)
20 mJ/cm2
40
70
100
150
200
300
KODAK 820
PREBAKE 100°C 30 min
A = 0.49
B = 0.031
C = 0.0125

Fig. 2.2.6

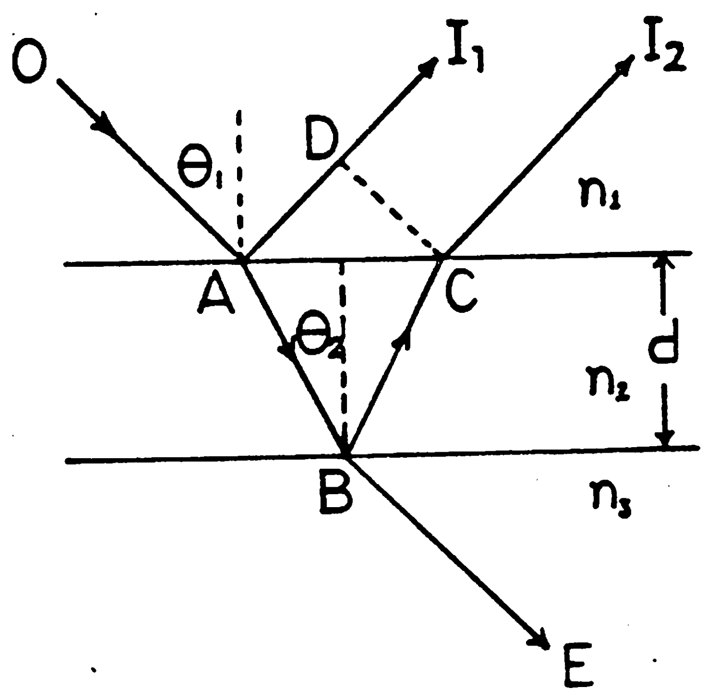


Fig. 2.3.1

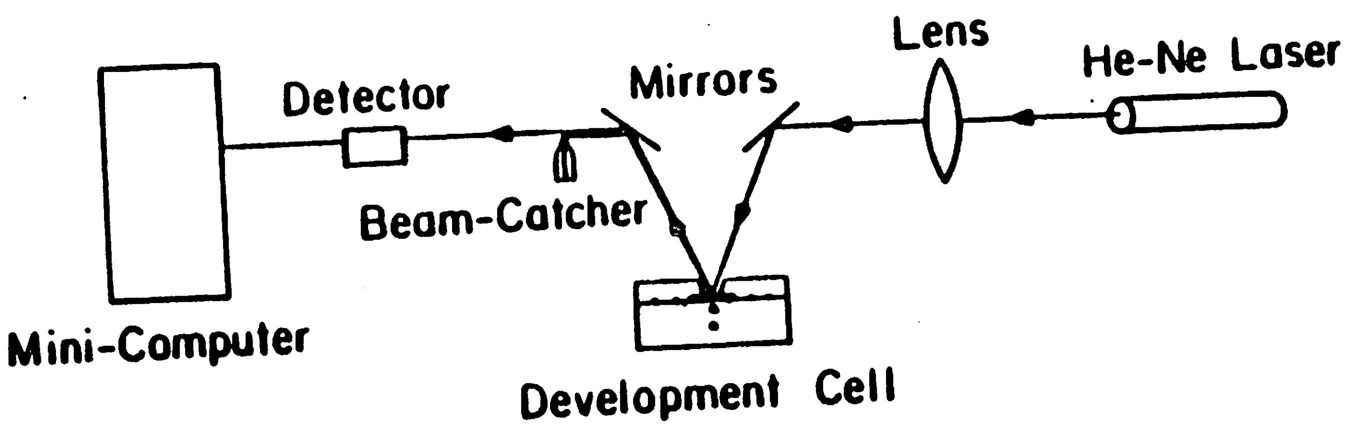

Lens
He-Ne Laser
Mirrors
Detector
Beam-Catcher
Mini-Computer
Development Cell

Fig. 2.3.2

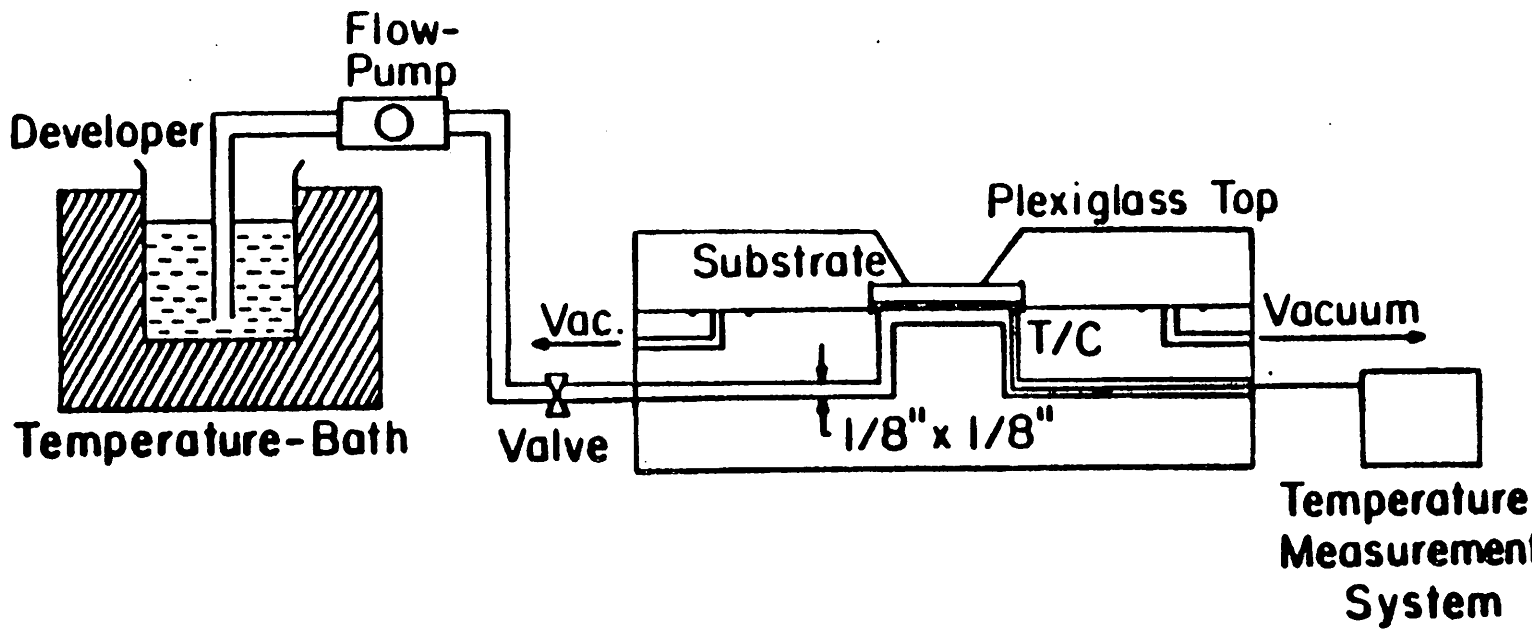

Flow-
Pump
Developer
Temperature-Bath
Valve
Vac.
Substrate
Plexiglass Top
Vacuum
T/C
1/8" x 1/8"
Temperature
Measurement
System

Fig. 2.3.3

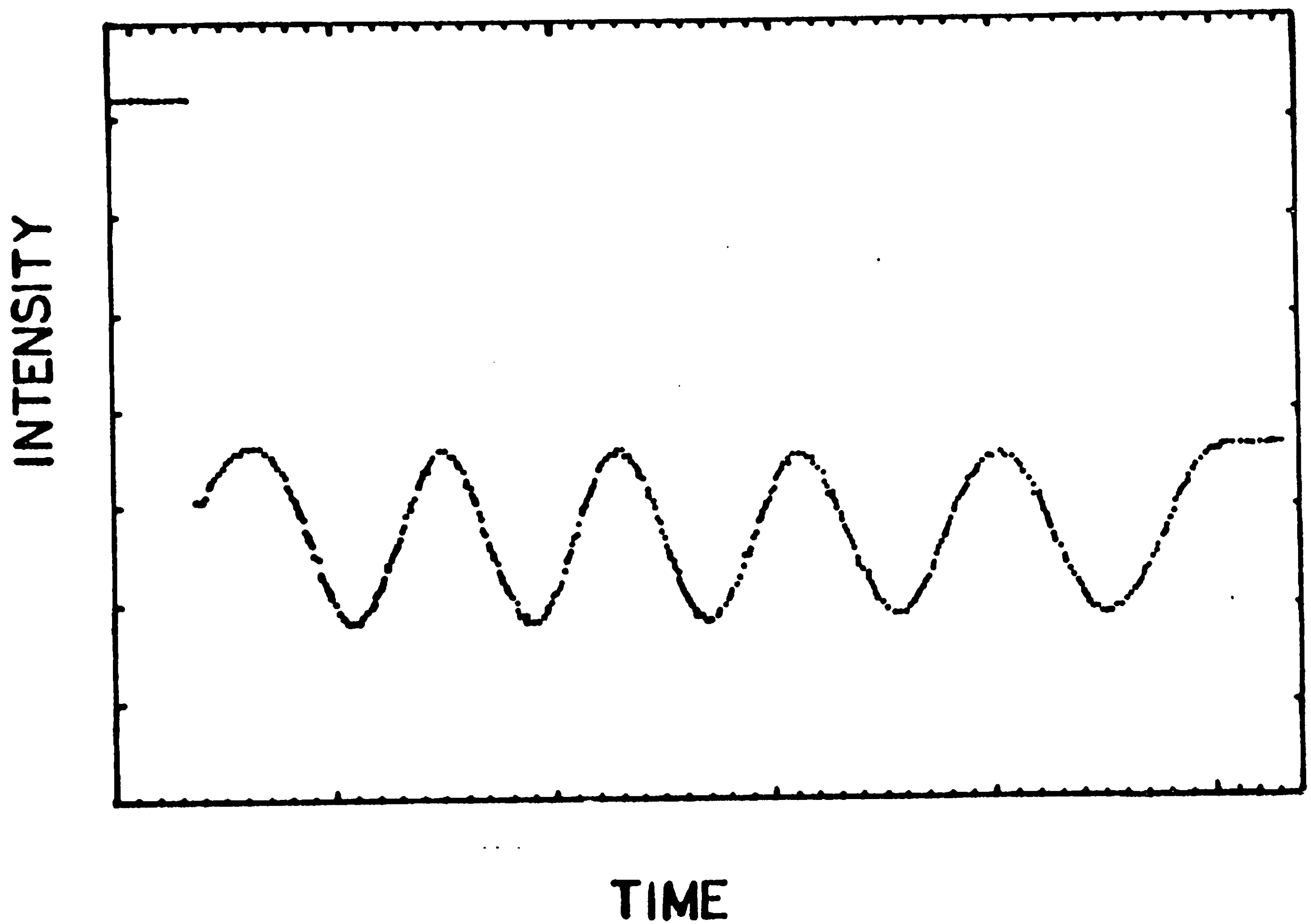


Fig. 2.3.4

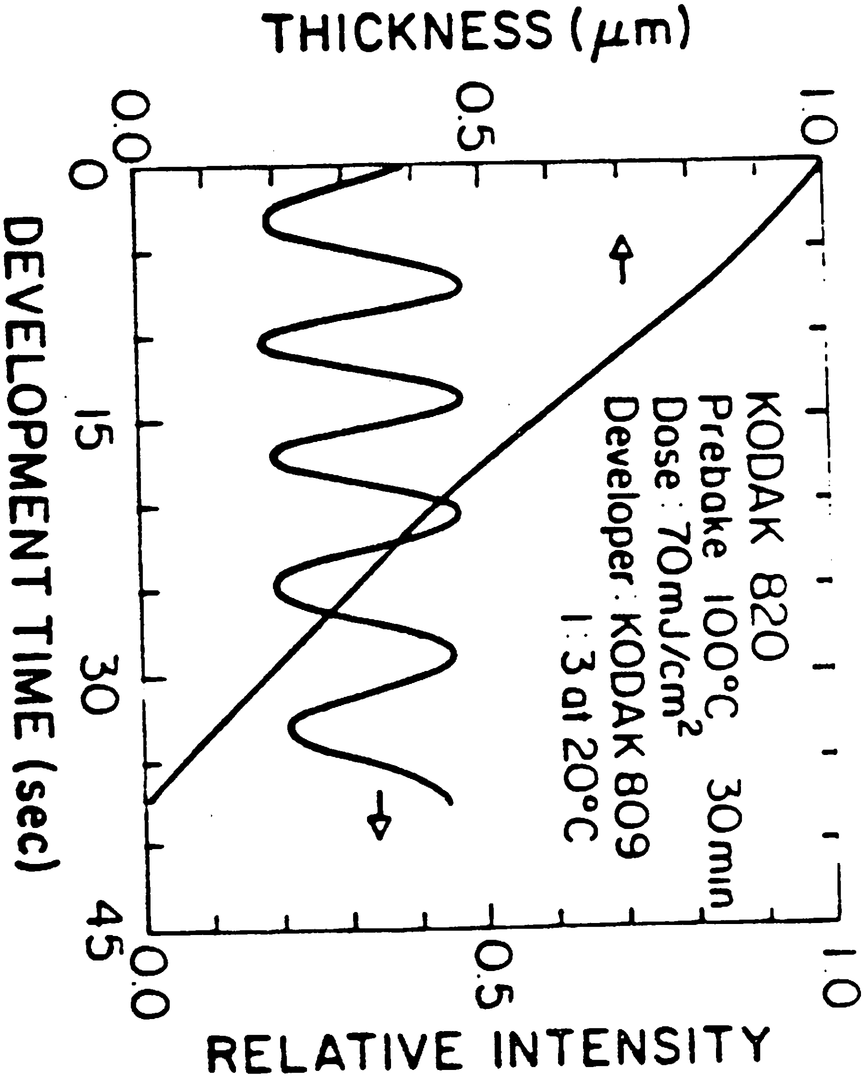

THICKNESS (μm)
0.0
0.5
1.0
KODAK 820
Prebake 100°C 30min
Dose : 70mJ/cm²
Developer : KODAK 809
1 : 3 at 20°C
0
15
30
45
DEVELOPMENT TIME (sec)
0.0
0.5
1.0
RELATIVE INTENSITY

Fig. 2.3.5

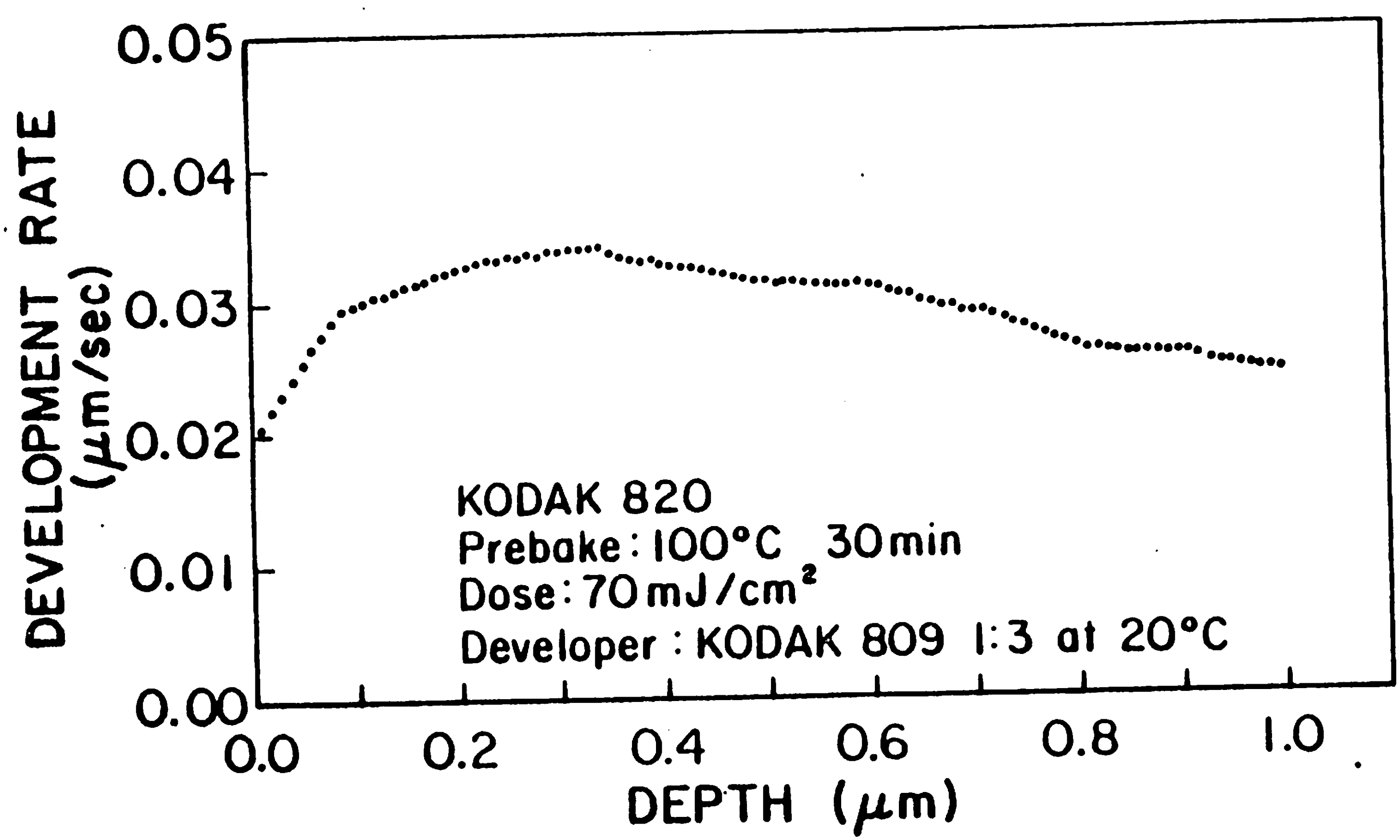

0.05
0.04
0.03
0.02
0.01
0.00
DEVELOPMENT RATE (μm/sec)
KODAK 820
Prebake: 100°C 30min
Dose: 70mJ/cm²
Developer: KODAK 809 1:3 at 20°C
0.0
0.2
0.4
0.6
0.8
1.0
DEPTH (μm)

Fig. 2.3.6

CHAPTER 3

DEVELOPMENT OF POSITIVE PHOTORESIST

3.1 Introduction

In order to quantitatively simulate photolithography, accurate models and sets of model parameters are required for various resists, developers, and processing conditions. In their pioneering studies of positive photoresist Dill, et al,[1,2] proposed a simple model for exposure and development. They proposed the parameters A, B, and C to define the status of exposed resist in terms of a normalized concentration (M) of photo-active compound (PAC). They also described the resist development behavior using a polynomial fit to the logarithm of rate versus M with the parameters E_1, E_2, and E_3 defining the polynomial. Other polynomials[3] and Fermi-Dirac distribution functions[4] have been proposed. To the best of our knowledge all of these functions are somewhat inadequate because they do not describe the rate over the full range of M or the rate variation near the surface. Several other papers have reported the importance of the retardation of the development rate near the surface and proposed various models to include the depth dependence[5,7].

This chapter introduces a new comprehensive model which describes the development behavior for positive photoresists as a function of depth and M over the full range of exposure. Model parameter values derived from development rate data are given for several resist and developer systems under various processing conditions. In Appendix C lists all resist parameters up to date.

3.2 Dissolution Rate Equation

With sufficient agitation (which depends on exposure) the development of positive photoresist can be surface reaction-rate limited, and therefore describable by a kinetic function which depends on the state of the resist, the development chemistry, and the temperature. Dill et al defined the state of the resist by the parameter M, a normalized concentration of PAC remaining after exposure, and described the development behavior of AZ1350J resist only as a function of M, holding the development chemistry and temperature constant. In later studies they showed the presence of a surface effect[8] in which either there was a delay before the onset of dissolution or the region near the surface has a reduced dissolution rate. We will call the former phenomenon 'surface-induction' (the time delay before development starts) and the latter 'surface-rate-retardation' (the retardation of dissolution rate near the surface compared to the dissolution rate in the bulk).

With a retardation effect it is necessary to account for the depth dependence of development rate in a more fundamental way. A new development rate model is needed satisfying several conditions:

(1). The function should be analytic and fit the data over the full range of M from zero to one at any position in the resist.

(2). The number of parameters should be minimized.

(3). The parameters should have physical significance if possible.

(4). The parameters should be easy to extract given a set of rate data.

We have chosen to retain the product formulation in which the retardation of the development rate near the surface is used as a multiplier f(z,M) to the bulk development rate R_{bulk}:

$$Rate(z,M) = f(z,M)R_{bulk}(M) \tag{3.2.1}$$

This product formulation is simple in both conception and implementation for modeling of surface-retardation. It is further attractive physically since we have found that the depth dependence is independent of M, that is f(z,M) can be further separated into individual functions of z and M.[9]

Bulk Development Rate :

The simplest model for bulk development must account, at least, for two processes: (1) the dissolution of base resin modified by the presence of photo-active compound (PAC) and (2) the dissolution of base resin modified by the presence of reacted PAC (largely carboxylic acid, here denoted PPA for photo-produced acid). The state parameter M is essentially the relative concentration of PAC remaining; similarly 1-M is the relative concentration of PPA. In unexposed photoresist, the presence of PAC slows the development in comparison with a pure base resin. On the other hand in a moderately exposed state, the PPA greatly enhances the development rate, again in comparison with a pure base resin. Conceptually the development may be viewed as the dissolution of the various fraction of the resin-PAC and resin-PPA system. The simplest case of combining the processes is to consider an incremental thickness of resist dz and add the times to dissolve the resist fractions. This approach has led us to the following form for the bulk development rate function R_{bulk} :

$$\frac{1}{R_{bulk}} = \frac{1-MP}{R_1} + \frac{MP}{R_2} \tag{3.2.2}$$

The parameters R_1 and R_2 are the simple limiting rates for fully exposed resist and unexposed resist, respectively. The function P describes the enhancement of development by the presence of PPA. We have chosen a function given by

$$P = exp(-R_3(1-M)) \tag{3.2.3}$$

where R_3 is a sensitivity parameter, which is a measure of how fast the development rate increases as the exposure increases. The combination of equations (3.2.2) and (3.3.3) gives

$$R_{bulk} = \frac{1}{R_1^{-1}(1-Mexp(-R_3(1-M))) + R_2^{-1}Mexp(-R_3(1-M))} \tag{3.2.4}$$

Equation (3.2.4) describes the bulk development rate over the full range of exposure from M=0 to M=1. At M=0 and M=1 R_{bulk} reduces to its two limiting rates R_1 and R_2. This satisfies the requirements stated above. Without M in front of the exponential term Equation (3.2.4) can be reduced to the similar form of the Fermi-Dirac distribution function which was proposed earlier by Brochet et al.[4] (but which is unable to fit the data over the wide range of conditions described below). Equation (3.2.4) has the minimum set of parameters which can describe the dissolution of a two-component mixture. In the event that the effective number of components is increased, more parameters are required. For example, if additives or processing is used to direct the exposure-product pathway to a mixture of organic acid and ketene,[10] at least one more parameter would be required physically.

Surface Rate Retardation :

As indicated above, the development rate near the surface is described by the product of the bulk rate and a multiplier f(z,M). The multiplier may depend both on depth and exposure. However, physically we do not expect the form of the depth function to depend on the exposure or development details. We thus model the multiplier as itself the product of two terms. We further choose the simplest physically appealing form for the z dependence, a decaying exponential function. The multiplier may be written as

$$f(z,M) = 1-(1-f(0,M))exp(-z/R_4) \tag{3.2.5}$$

where z is the depth into the resist, R_4 is the characteristic retardation depth, and f(0,M) is the ratio of surface development rate to bulk development rate at any M value. The depth parameter R_4 should in principle depend only on the resist and how it is processed. It measures, in effect, the thickness of the "tough skin". The dependence of f on exposure, reflected in the parameter f(0,M), may also depend strongly on development details (developer, dilution, temperature, etc.). Under the simplest conditions it is found that f(0,M) can be modeled by a simple linear function of M:

$$f(0,M) = R_5 - (R_5 - R_6)\,M \tag{3.2.6}$$

where R_5 is the ratio of surface rate to bulk rate at M=0 and R_6 the ratio at M=1. For other systems, eg., the development of KODAK 820 resist in KODAK 932, a more complicated behavior is seen. In such cases we have chosen to use a piecewise linear function of M, with the minimum number of parameters needed to accurately describe f(0,M).

3.3 Derivation of Model Parameters from Experiments

Details about the measurement technique are described in Chapter 2. A brief summary is presented here. The photoresist is spun on a glass substrate with matched index of refraction and prebaked. The substrates are exposed at 436 nm. For each sample the transmittance variation during the exposure is recorded in a mini-computer and the program ABCMX.BAS calculates the M value at any position within the resist. The exposed substrate is loaded on the development cell for the dissolution rate measurement. The resist is illuminated by He-Ne laser at 633 nm from the back side of the transparent substrate. The desired reflected beam is directed to the detector and the variation of the reflectivity is recorded in the mini-computer. The program RATEX.BAS calculates the development rate at every depth within the resist film.

The bulk development parameters, R_1, R_2, and R_3 are derived by fitting equation (3.2.4) to the bulk rate vs M data. Points deeper than 0.6 μm are considered to be in the bulk. A fitting program is used to minimize the sum of the square error with the initial estimates of R_1, R_2, and R_3. The program is stable and minimizes the square errors from the estimates based on R_1 as the rate at $M=0$, R_2 as the rate at $M=1$, and R_3 as the slope near $M=1$ in the plot of log(rate) vs M.

Once the parameters R_1, R_2, and R_3 are determined, the program finds the other parameters R_4, R_5, and R_6 by fitting equation (3.2.1) to all data, the rate vs M and depth. R_5 and R_6 are easily estimated by the ratios of the surface rate to the bulk rate at $M=0$ and $M=1$, respectively. The estimate for R_4 is not explicit on the rate vs M plot but can be found by a local fitting of equation (3.2.5) to the rate vs depth for the unexposed resist.

3.4 Resist Development

In this section the development behavior is demonstrated for KODAK 820 and Shipley MICROPOSIT 1470 positive photoresists under various processing conditions with several developers. It was first necessary to establish a suitable flow rate for development rate characterization. In this study it was desirable to remain in the surface reaction-rate control regime, consistent with the model equations used. The bulk development rates of normally processed, heavily exposed resist (M=0.12) were measured for samples of MICROPOSIT 1470 and KODAK 820 resists as a function of flow rate across the resist surface. The results, shown in Fig. 3.4.1, indicate that the desired regime is obtained for flow rates exceeding 10 cm/sec. All data were taken at the flow rate of 15 cm/sec.

Fig. 3.4.2 illustrates the effect of prebake temperature on MICROPOSIT 1470. Equation (3.2.4) is fitted to the experimental data, plotted as rate versus M. The standard condition is 20 ° C development in MICROPOSIT 351 developer in 1:5 dilution following 20 minute prebake. Data points near the surface are excluded because of a small surface-retardation effect. However, the complete parameter set, including the retardation model parameters, is given in Table I of Appendix C. For moderately and heavily exposed resists, the development rate decreases with prebake temperature. Presumably the rate reduction is primarily accounted for by the reduced fraction of PPA photo products (carboxylic acid) after U.V.exposure owing to PAC destruction by high temperature bake before exposure.[8]

The impact of prebake temperature on the resist profile is shown in Fig. 3.4.3. (rate data from Fig. 3.4.2). The resist line-edge-profile is shown for a 1.0 μm line in a pattern of equal lines and spaces. The simulated exposure condition uses a lens with NA=0.28, σ=0.7, and δ(defocus)=1.5 μm at the wavelength of 436 nm. The dose shown as a parameter is adjusted to yield a nominal final linewidth of 1.0 μm. No significant effect of prebake is observed below about 100 ° C, but rapid degradation in edge slope, sensitivity, and resist thickness is observed at 120 ° C.

The dissolution rates for several developer solutions are compared in Fig. 3.4.4. It is interesting that different developers but with proper dilutions show similar development behavior. MICROPOSIT developer diluted 1:1, MICROPOSIT 351 developer diluted 1:5, and MICROPOSIT MF312 developer diluted 1:1.5 demonstrate essentially the same development behavior. Increasing the concentration in 351 type developer from 1:5 to 1:3 and in MF312 metal ion free developer from 1:1.5 to 1:1 the overall development rate increases. MF312 type developer diluted 1:1 and 351 type developer diluted 1:3 show an effective higher sensitivity owing to faster development rates over the entire exposure range. But the line edge profile shown in Fig. 3.4.5 is worse than that in the standard developer solution because the unexposed development rate is high (~0.0025 μm/sec) resulting in a large amount of thickness loss at the top.

Whereas the MICROPOSIT resist studied here has a modest surface effect, the KODAK 820 resist shows a large development rate retardation near the surface. In order to illustrate this depth dependence, the development rates are plotted in Fig. 3.4.6 (a) and (b). The measured development rate normalized to the bulk rate is shown for two exposure extremes, 0 and 300 mJ/cm^2. The dashed lines are the exponential fit to the data (small dots) in each figure. The ratio of the surface rate to the bulk rate varies from 0.08 to 0.6 in KODAK 809 type developer and from 0.06 to 0.5 in MICROPOSIT 351 type developer over the exposure range of 0 to 300 mJ/cm^2. In both cases the exponential decay distance is 0.25

μm.

The complete development behavior for KODAK 820 resist in KODAK 809 developer is shown in Fig. 3.4.7. The development rates are plotted vs M for a number of samples covering a full range of exposures from 0 mJ/cm^2(M=1) to 300 mJ/cm^2(M=0.02). The rate for each resist sample is lowest at the surface due to the surface-retardation effect, increases rapidly with depth, and saturates to the bulk rate (which has a small decrease with depth because of absorption). In all cases an exponential decay depth of 0.25μm fits the data well for the surface-retardation effect. The M value can be calculated from the exposure dose at any depth of each resist sample with known A, B, and C parameters. Thus the left-most point on each sample gives the surface rate, and the right-most point gives the bulk rate. The upper dotted curve is the bulk rate from equation (3.2.4). The lower dotted curve is the surface rate from equation (3.2.1), with the forms of equations (3.2.5) and (3.2.6) used for f(z). Note that in the insert both the measured data and the results of the model with the R parameters are shown and indicate good agreement with the model. As mentioned earlier the bulk rate is described by R_1, R_2, and R_3 in which R_1(0.23 μm/sec) is the rate at M=0, R_2(0.0016 μm/sec) the rate at M=1, and R_3(5.6) a sensitivity parameter. The surface rate is described by the retardation factor f(0) to the bulk rate at a fixed M value. The value of f(0) at M=0 is 0.62 and denoted by R_5. The value of f(0) at M=1 is 0.08 and denoted by R_6. The value of f(0) between M=0 and M=1 is described by a linear interpolation between R_5 and R_6 with respect to M. The rate at any depth with a constant M can be obtained by the exponential function with the characteristic retardation depth of R_4(0.25 μm).

An even more pronounced retardation effect was observed with this resist using a different developer, particularly at low exposure. The time to develop the top 0.1 μm in KODAK 932(metal ion free) developer is plotted against exposure in Fig. 3.4.8. For the standard development with KODAK 809 the etch time is continuous over all exposures but with 932 developer a discontinuity is seen at about 40 mJ/cm^2. In 932 type developer for exposures less than 40 mJ/cm^2, the retardation factor f(0) is about 0.02, and for exposures greater than 40 mJ/cm^2 f(0) varies from 0.13 to 0.45.

The complete development behavior in 932 type developer is shown in Fig. 3.4.9. The bulk rate is normal and described by R_1(0.33 μm/sec), R_2(0.0015 μm/sec), and R_3(10) but the surface rate is discontinuous at about M=0.6(dose=40mJ/cm^2). To model this surface behavior the surface ratio is separated into three regions and is modeled by a simple piecewise linear equation $f(0,M)=R_5+M(R_7-R_5)/R_8$ for $0<M<R_8$, $f(0,M)=R_7+(M-R_8)(R_9-R_7)/(R_{10}-R_8)$ for $R_8<M<R_{10}$, $f(0,M)=R_9+(M-R_{10})(R_9-R_6)/(R_{10}-1)$ for $R_{10}<M<1$. This piecewise linear fit, while inelegant, is simple to use and quite flexible. If even more exotic behavior is discovered the function can be extended to more segments in an obvious way. In all regions the best fit for R_4 is 0.25 μm.

The role of surface retardation in achieving vertical resist profiles is examined in Fig. 3.4.10. In this simulation the resist is exposed at 90 mJ/cm^2 and developed for 10 seconds longer than break-through. The image of a 1 μm line was produced with an optical system of NA=0.28, σ=0.7, and δ=1.5 μm at 436 nm. The resist profile is shown for three cases with (I) bulk rate only (in KODAK 809), (II) retardation (in KODAK 809), and (III) retardation (in KODAK 932). In the absence of an retardation effect a large amount of resist loss at the top occurs and the wall angle is about 76 °. Including the retardation the top loss is small and more vertical or even under-cut profiles are obtained. An example of resist profile simulation (SAMPLE) is given in Appendix D.

3.5 Resist Contrast

Resist contrast is often used as a single parameter to compare the performance. Although the definition of the resist contrast for optical resists has not been established we may adapt the definition for electron-beam resists. The resist contrast, γ, is defined by the slope on the plot of resist thickness remaining vs logarithm of dose(D) at fixed resist thickness(T) and development time(t) as shown in Fig. 3.5.1. Analytically γ may be given by

$$\gamma = \frac{\partial T}{\partial log(D)}$$
$$= 2.3\, D \,\frac{\partial M}{\partial D}\frac{\partial R}{\partial M}\frac{\partial T}{\partial R} \qquad (3.5.1)$$

For bulk development rate with the resist filmon a matched substrate

$$\frac{\partial R}{\partial M} = \frac{-R}{M}\frac{R}{R_1}\left(\frac{R_1}{R} - 1\right)(1 + R_3 M) \qquad (3.5.2)$$

$$\frac{\partial M}{\partial D} = -CM \qquad (3.5.3)$$

$$D = \frac{-ln(M)}{C} \qquad (3.5.4)$$

If the film is thin enough for the rate dependence on thickness to be ignored

$$\frac{\partial T}{\partial R} \approx \frac{T}{R} \tag{3.5.5}$$

Inserting equations (3.5.2) to (3.5.5) into equation (3.5.1) yields

$$\gamma \approx 2.3T\frac{R}{R_1}(\frac{R_1}{R}-1)ln(M)(1+R_3M) \tag{3.5.6}$$

If R_1 is much greater than R then γ is further simplified as

$$\gamma \approx 2.3Tln(M)(1+R_3M) \tag{3.5.7}$$

M can be evaluated from equation (3.2.4) through equation (3.5.5) with fixed thickness(T) and development time(t). Including the depth dependence of development rate it is hard to obtain a simple expression for γ. In such case γ can be calculated numerically.

The resist contrast is compared for several types of resist and developer systems with the thickness of 1 μm and a development time of 60 seconds. The thickness remaining is plotted against log(D) in Fig. 3.5.2. For MICROPOSIT 1470 γ is 3.6 from the numerical calculation. γ is highest (5.1) for the strong rate-retardation case with KODAK 820 in 932 type developer. The effect of rate-retardation on γ is illustrated with KODAK 820 in 809 developer. Including the retardation γ increases from 3.2 to 3.9. The dependence of γ on the type of substrate is also shown in this figure for MICROPOSIT 1470 resist on a silicon wafer. The rate oscillates owing to the standing wave effect. The average slope near the bottom of the resist is given for γ (3.3). The effective resist contrast is lower on a reflecting substrate.

References

1. F.H.Dill, W,P,Hornberger, P.S.Hauge, and J,M,Shaw, "Characterization of Positive Photoresist", IEEE Trans. on Electron Devices, Vol. ED-22, No.7, PP. 445-452, July 1975.

2. K.L.Konnerth and F.H.Dill, "In-Situ Measurement of Dielectric Thickness During Etching or Developing Process", IEEE Trans. on Electron Devices, Vol. ED-22, No.7, PP. 452-456, July 1975.

3. M.M. O'Toole and W.J. Grande, "Characterization of Positive Resist Development," IEEE EDL-2, Dec., 1981.

4. Par A. Brochet, G.M.Dubroeucq, et M.Lacombat, "Modelisation des processus dexposition et de developpement dune ressine photosensible positive: Application au masquage par projection", Revue Technique Thomson-CSF, Vol. 9, No 2, Jun., 1977

5. A.R. Neureuther, D.F. Kyser, and C.H. Ting, "Electron-Beam Resist Edge Profile Simulation," IEEE Trans. ED-26, No.4, April, 1979.

6. A.R.Neureuther, "Resist Profile Quality and Linewidth Control", Symposium on Silicon Processing, Jan., 19-22, 1982

7. D.C.Hofer, C.G.Wilson, A.R.Neureuther, and M.Hakey, "Characterization of the Induction Effect at Mid UV Exposure: Application to AZ2400 at 313 nm", SPIE Vol. 334, Optical Microlithography, Paper No. 26, Mar.,1982

8. F.H.Dill and J.M.Shaw, "Thermal Effects on the Photoresist AZ1350J", IBM J. Res.develop. May 1977.

9. Deok J. Kim, W.G.Oldham, and A.R.Neureuther, "Characterization and

Modeling of a Resist with Built-In Induction Effect", KODAK Microelectronics Seminar, Interface 83, San Diego, Nov.,1983

10. J.Pacansky and J.Lyrea, "Photochemical Decomposition Mechanisms for AZ-Type Photoresists", 24-42, IBM J. Res. develop, 1979

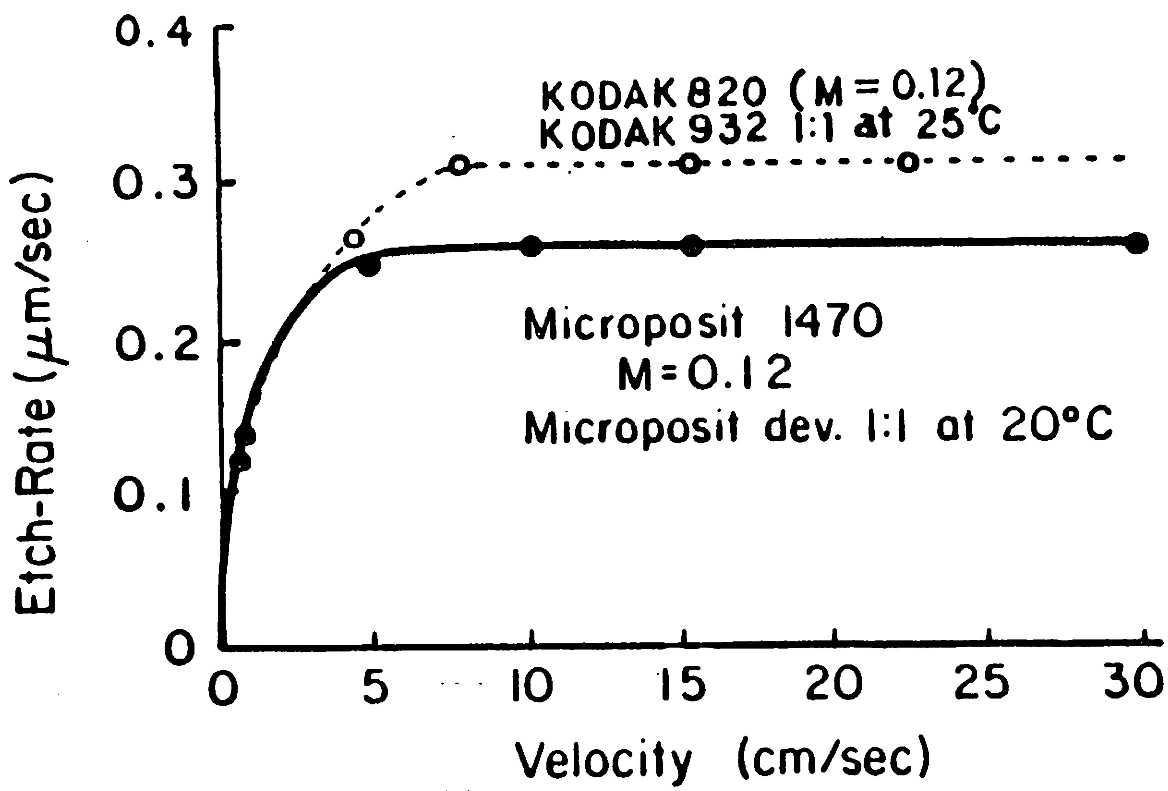

KODAK 820 (M = 0.12)
KODAK 932 1:1 at 25°C
Microposit 1470
M = 0.12
Microposit dev. 1:1 at 20°C
Etch-Rate (μm/sec)
0
0.1
0.2
0.3
0.4
0
5
10
15
20
25
30
Velocity (cm/sec)

Fig. 3.4.1

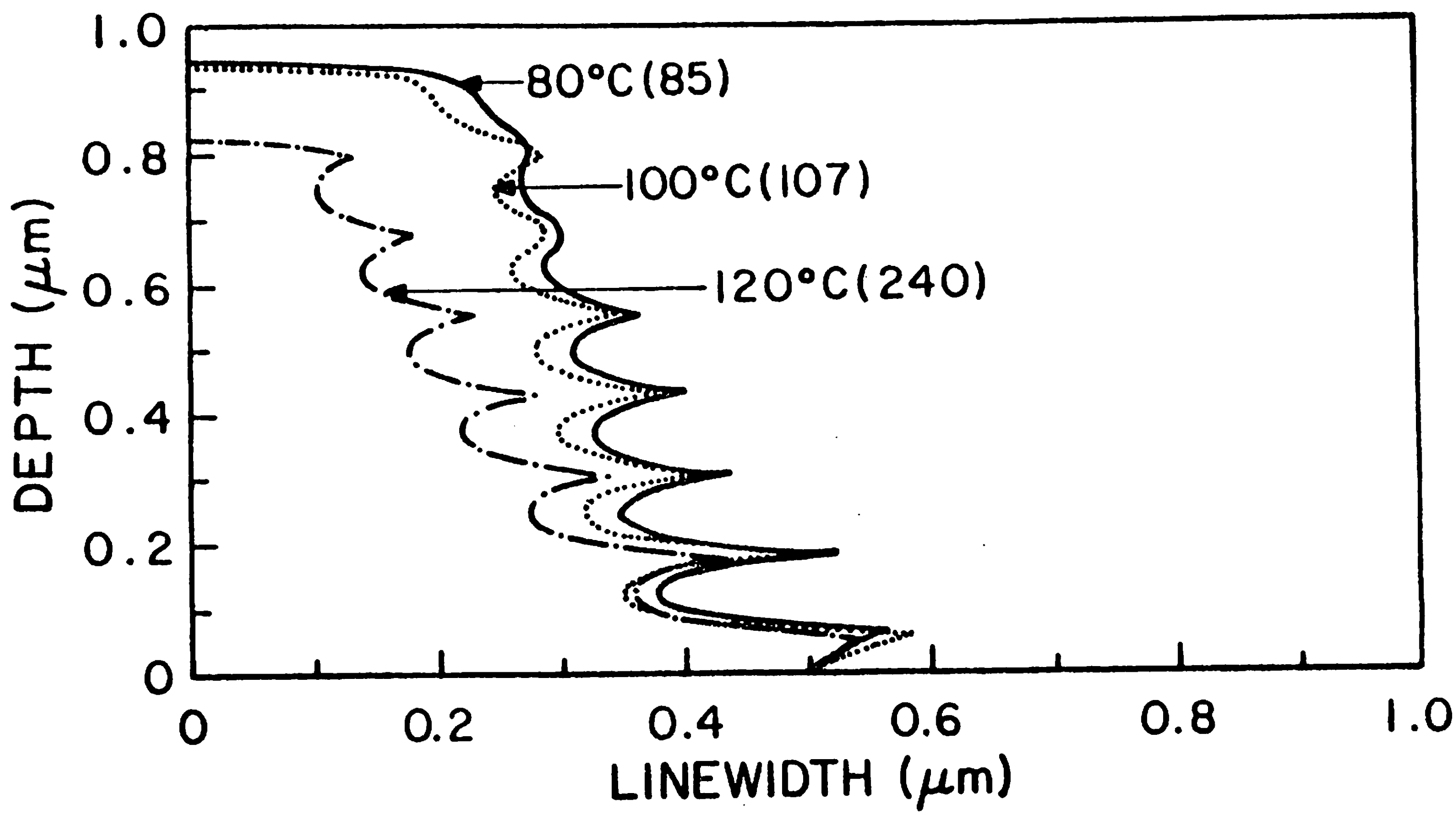

1.0
0.8
0.6
0.4
0.2
0
DEPTH (μm)
0
0.2
0.4
0.6
0.8
1.0
LINEWIDTH (μm)
80°C(85)
100°C(107)
120°C(240)

Fig. 3.4.3

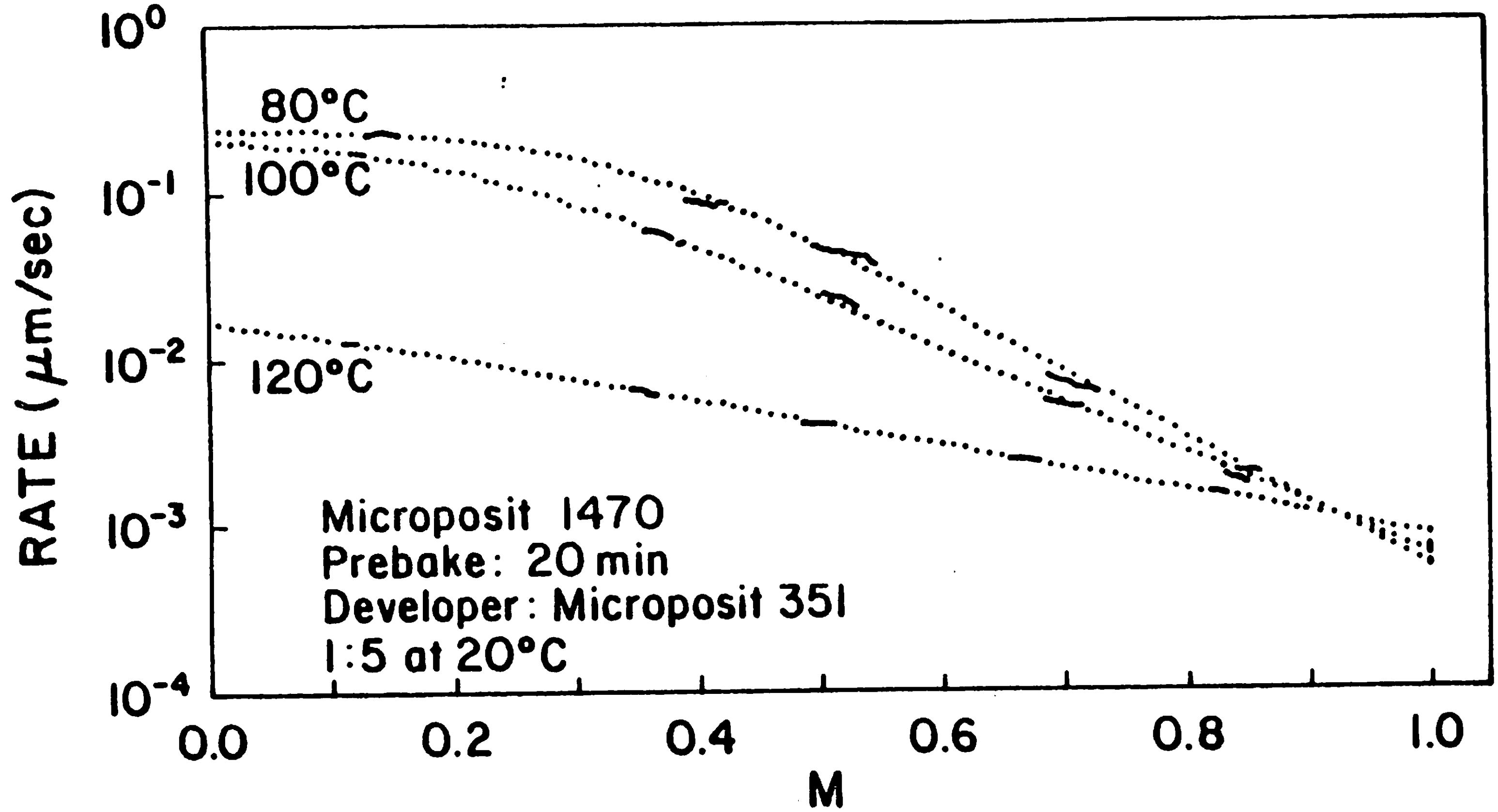
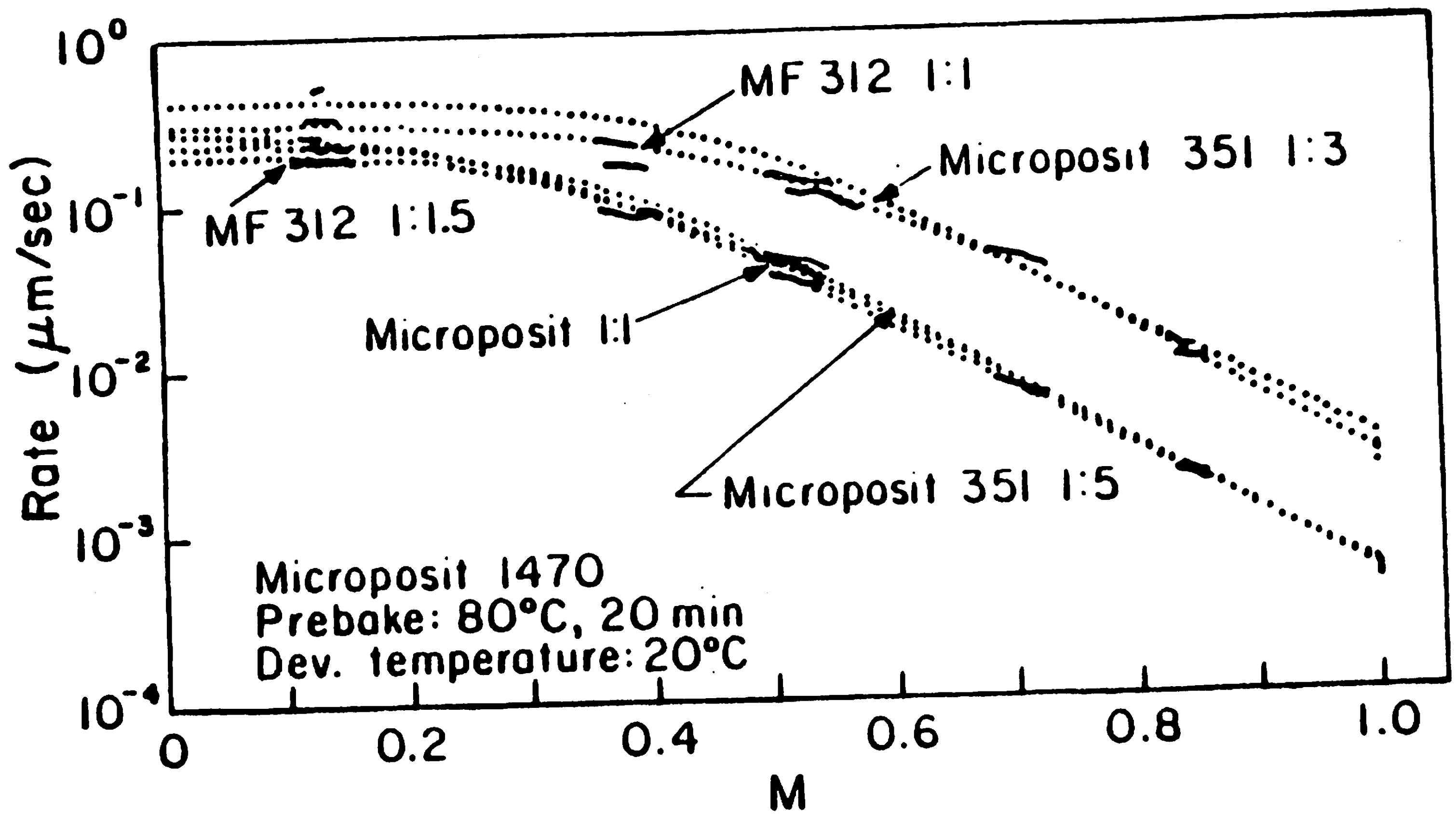

RATE (μm/sec)
10^0
10^-1
10^-2
10^-3
10^-4
80°C
100°C
120°C
Microposit 1470
Prebake: 20 min
Developer: Microposit 351
1:5 at 20°C
0.0
0.2
0.4
0.6
0.8
1.0
M

Fig. 3.4.2



10⁰
10⁻¹
10⁻²
10⁻³
10⁻⁴
Rate (μm/sec)
MF 312 1:1
Microposit 351 1:3
MF 312 1:1.5
Microposit 1:1
Microposit 351 1:5
Microposit 1470
Prebake: 80°C, 20 min
Dev. temperature: 20°C
0
0.2
0.4
0.6
0.8
1.0
M

Fig. 3.4.4

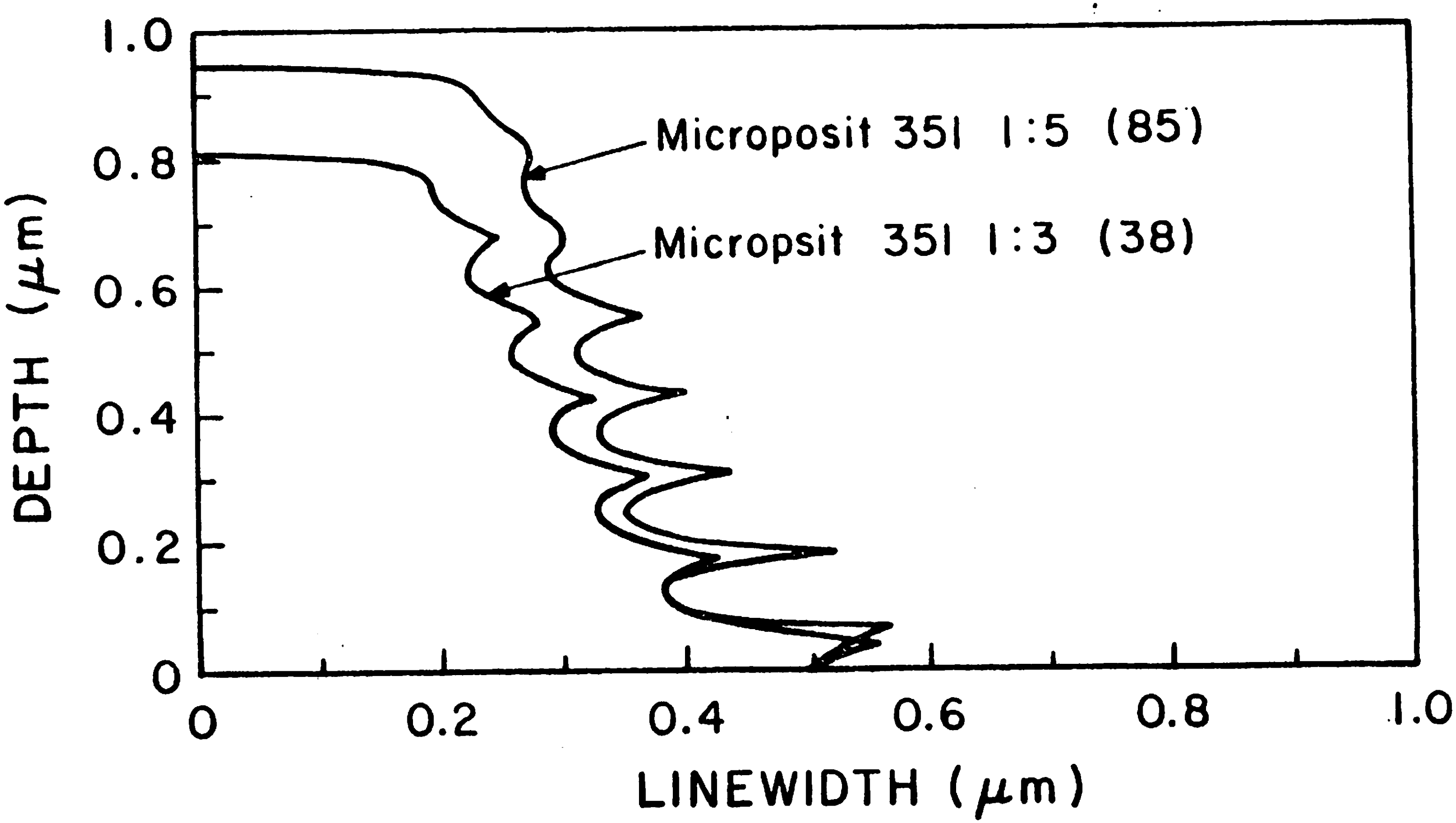
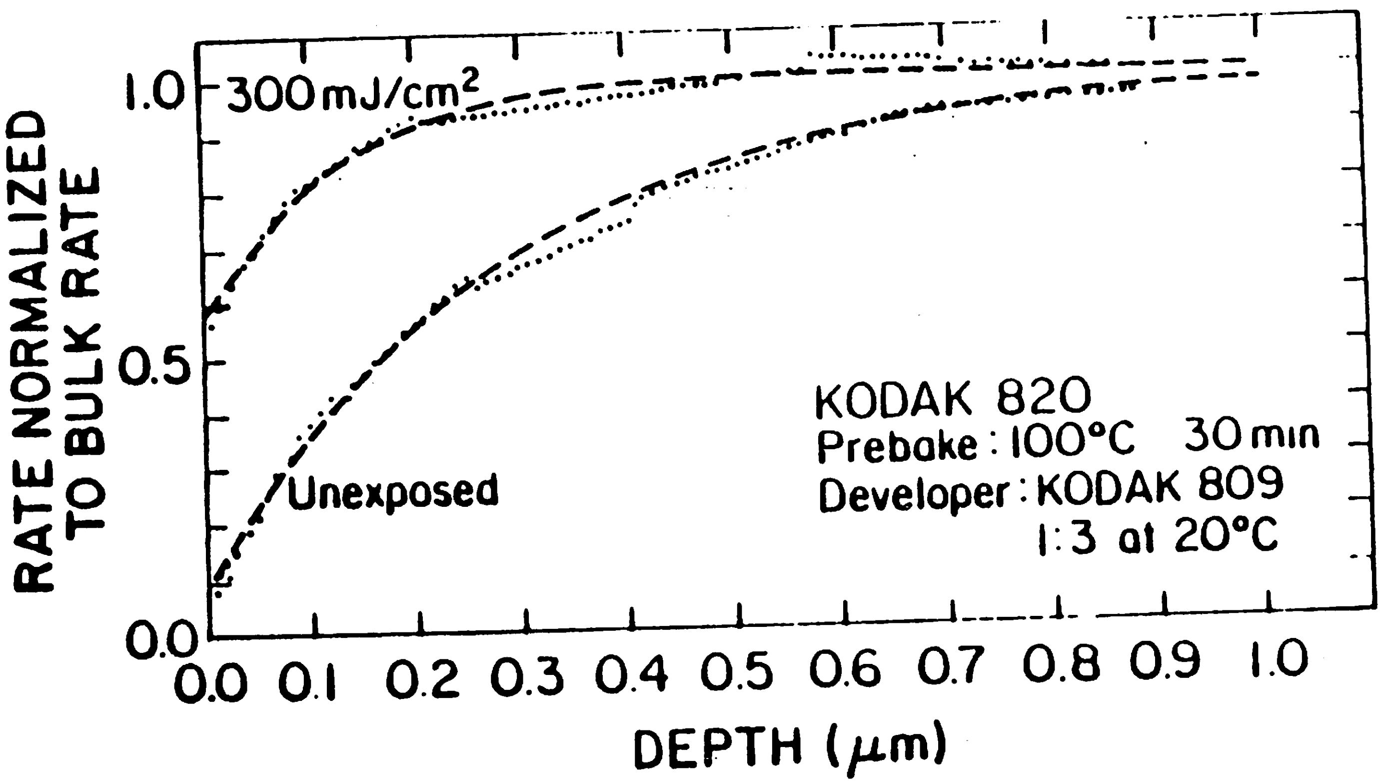

DEPTH (μm)
1.0
0.8
0.6
0.4
0.2
0
Microposit 351 1:5 (85)
Micropsit 351 1:3 (38)
0
0.2
0.4
0.6
0.8
1.0
LINEWIDTH (μm)

Fig. 3.4.5



RATE NORMALIZED TO BULK RATE
1.0
0.5
0.0
300 mJ/cm²
Unexposed
KODAK 820
Prebake : 100°C 30 min
Developer : KODAK 809
1:3 at 20°C
0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
DEPTH (μm)

Fig. 3.4.6(a)

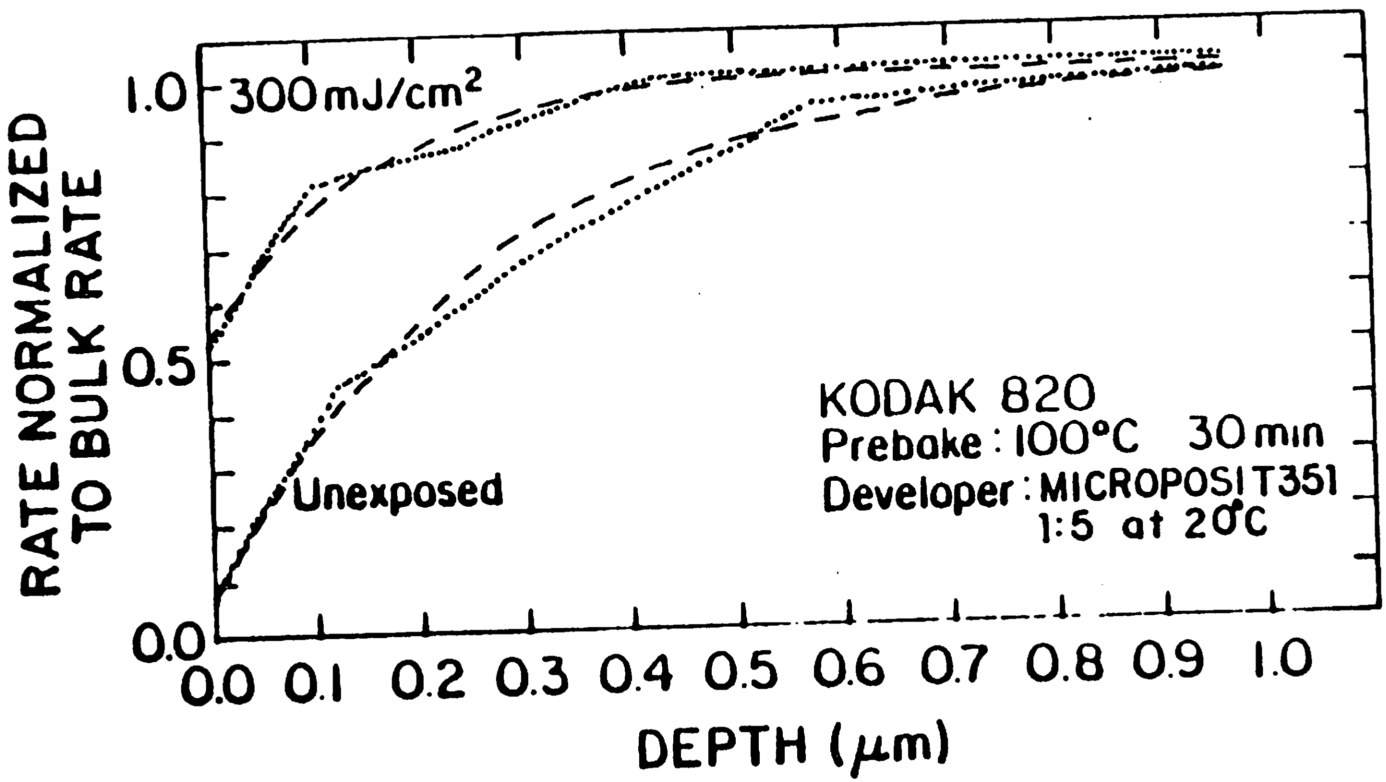

300mJ/cm²
Unexposed
KODAK 820
Prebake: 100°C 30 min
Developer: MICROPOSIT T351
1:5 at 20°C
RATE NORMALIZED
TO BULK RATE
1.0
0.5
0.0
0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
DEPTH (μm)

Fig. 3.4.6(b)

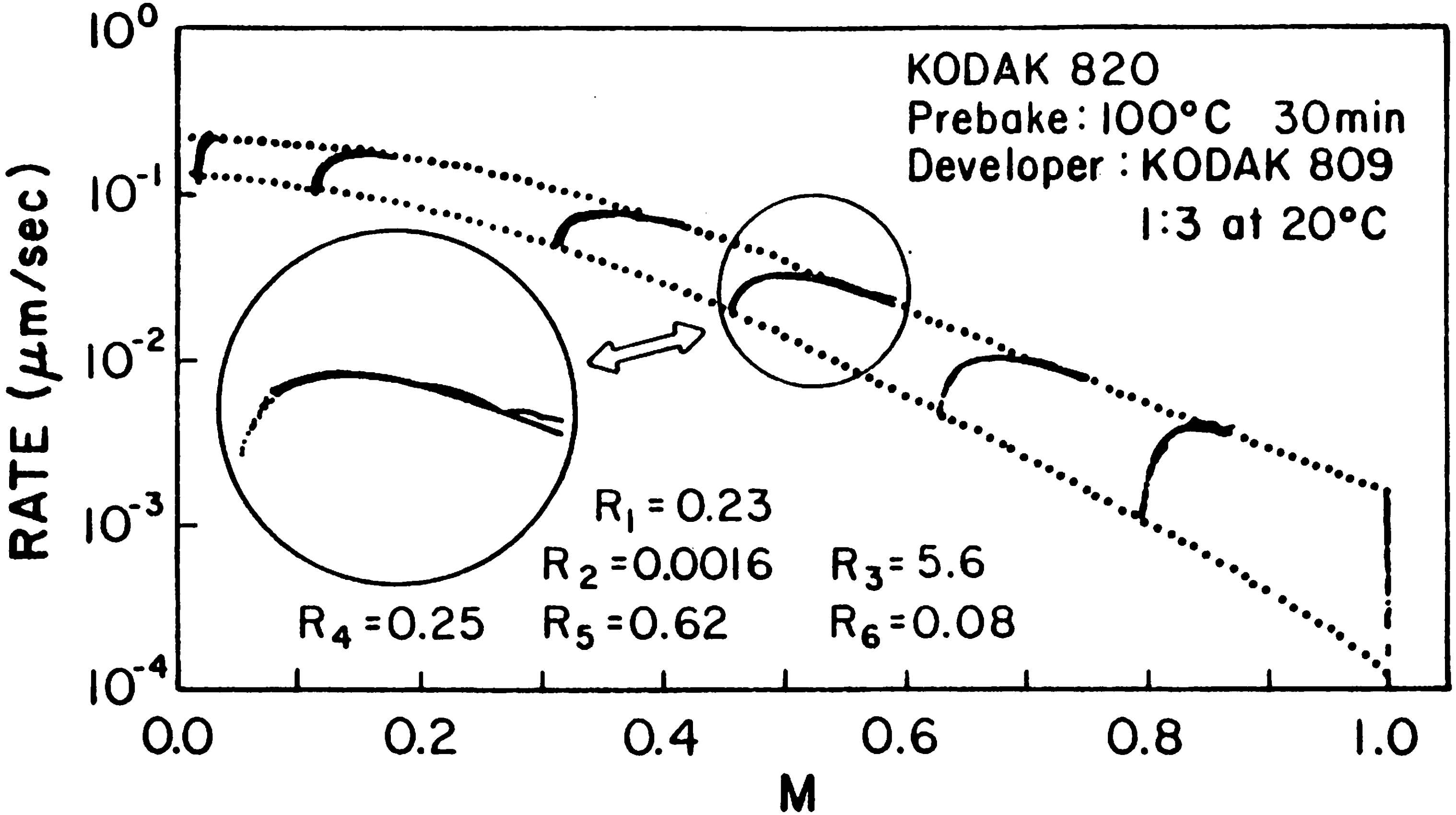

KODAK 820
Prebake: 100°C 30min
Developer: KODAK 809
1:3 at 20°C
R1 = 0.23
R2 = 0.0016
R3 = 5.6
R4 = 0.25
R5 = 0.62
R6 = 0.08
RATE (µm/sec)
10^0
10^-1
10^-2
10^-3
10^-4
0.0
0.2
0.4
0.6
0.8
1.0
M

Fig. 3.4.7

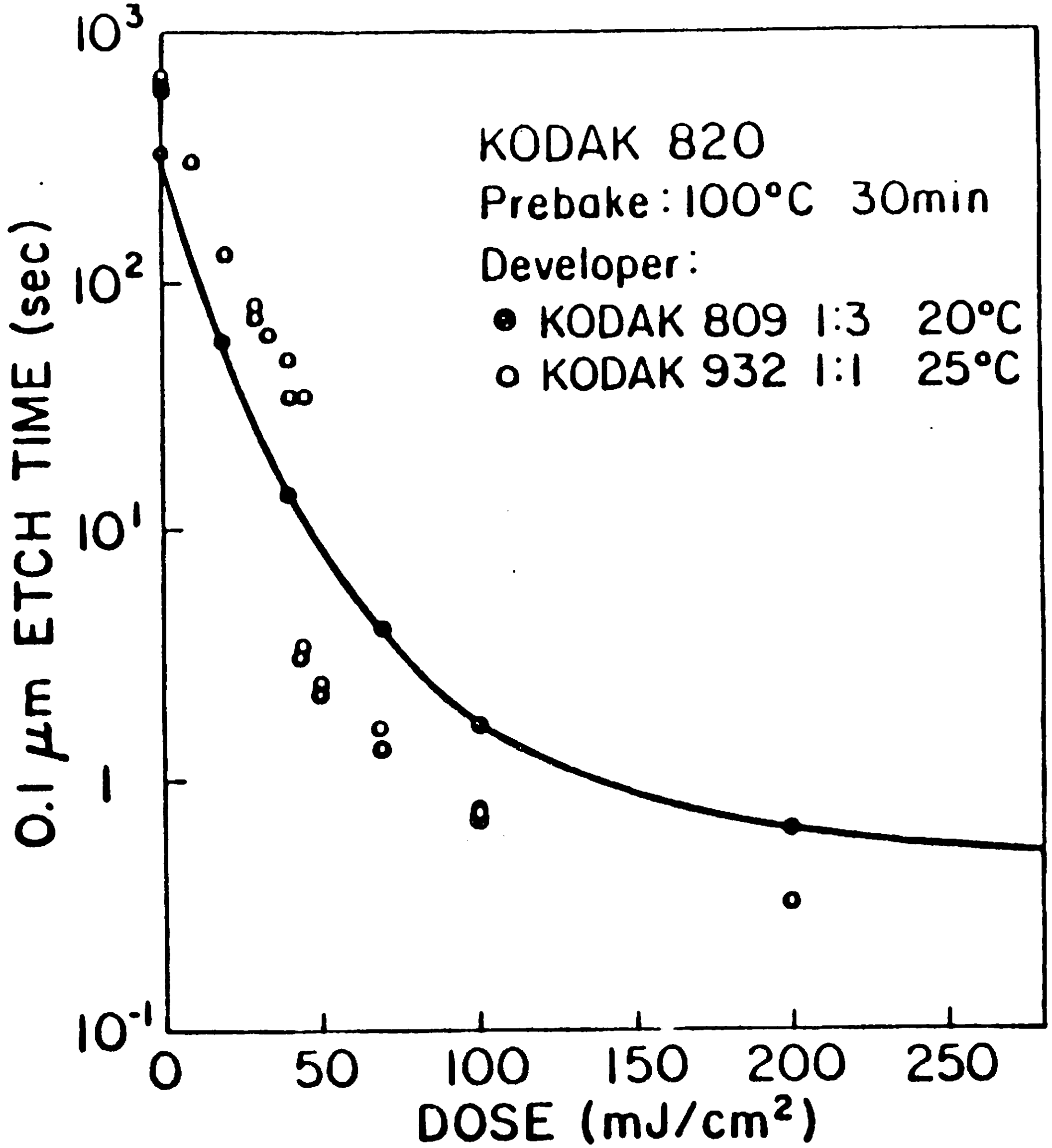

KODAK 820
Prebake: 100°C 30min
Developer:
● KODAK 809 1:3 20°C
○ KODAK 932 1:1 25°C
0.1 μm ETCH TIME (sec)
DOSE (mJ/cm²)
10³
10²
10¹
1
10⁻¹
0
50
100
150
200
250

Fig. 3.4.8

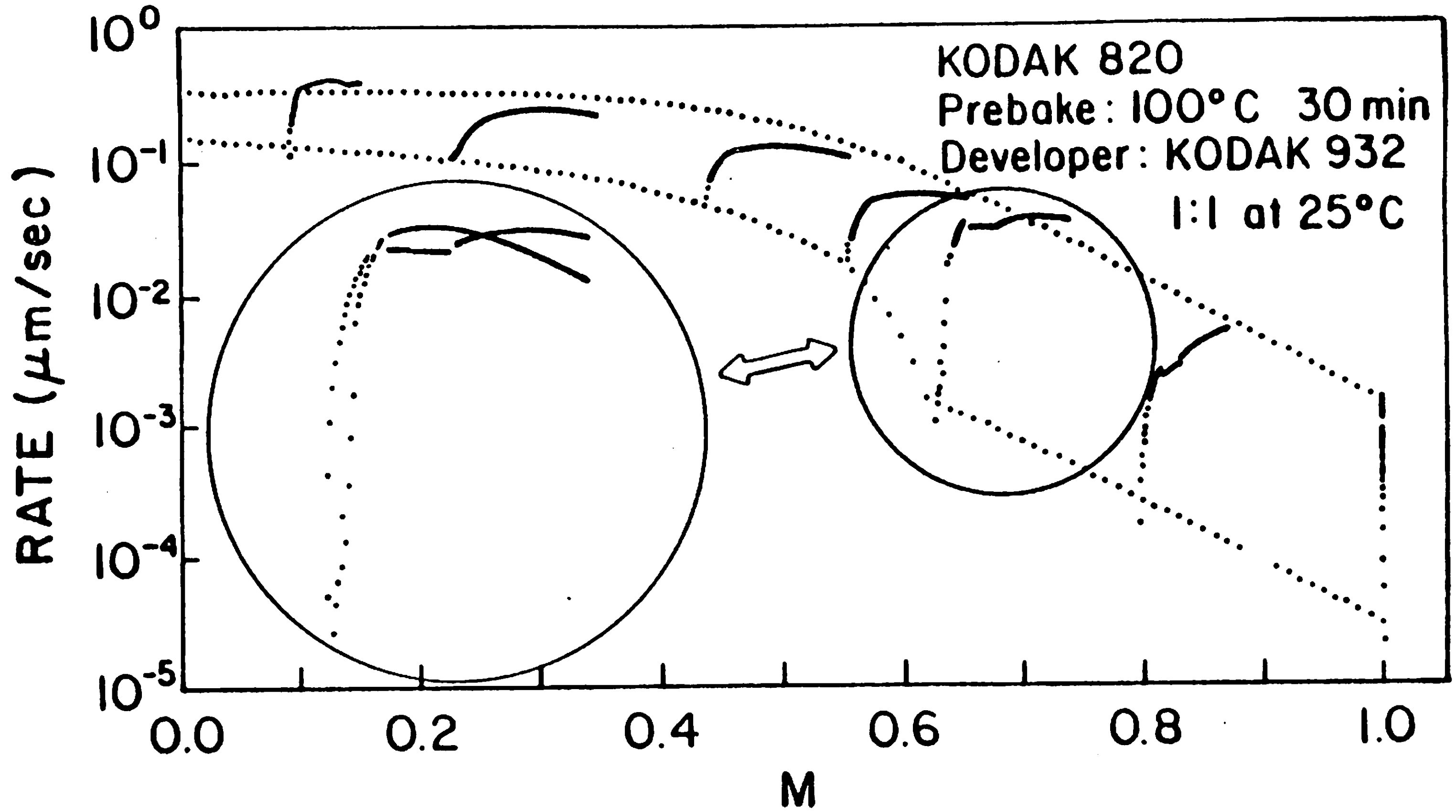
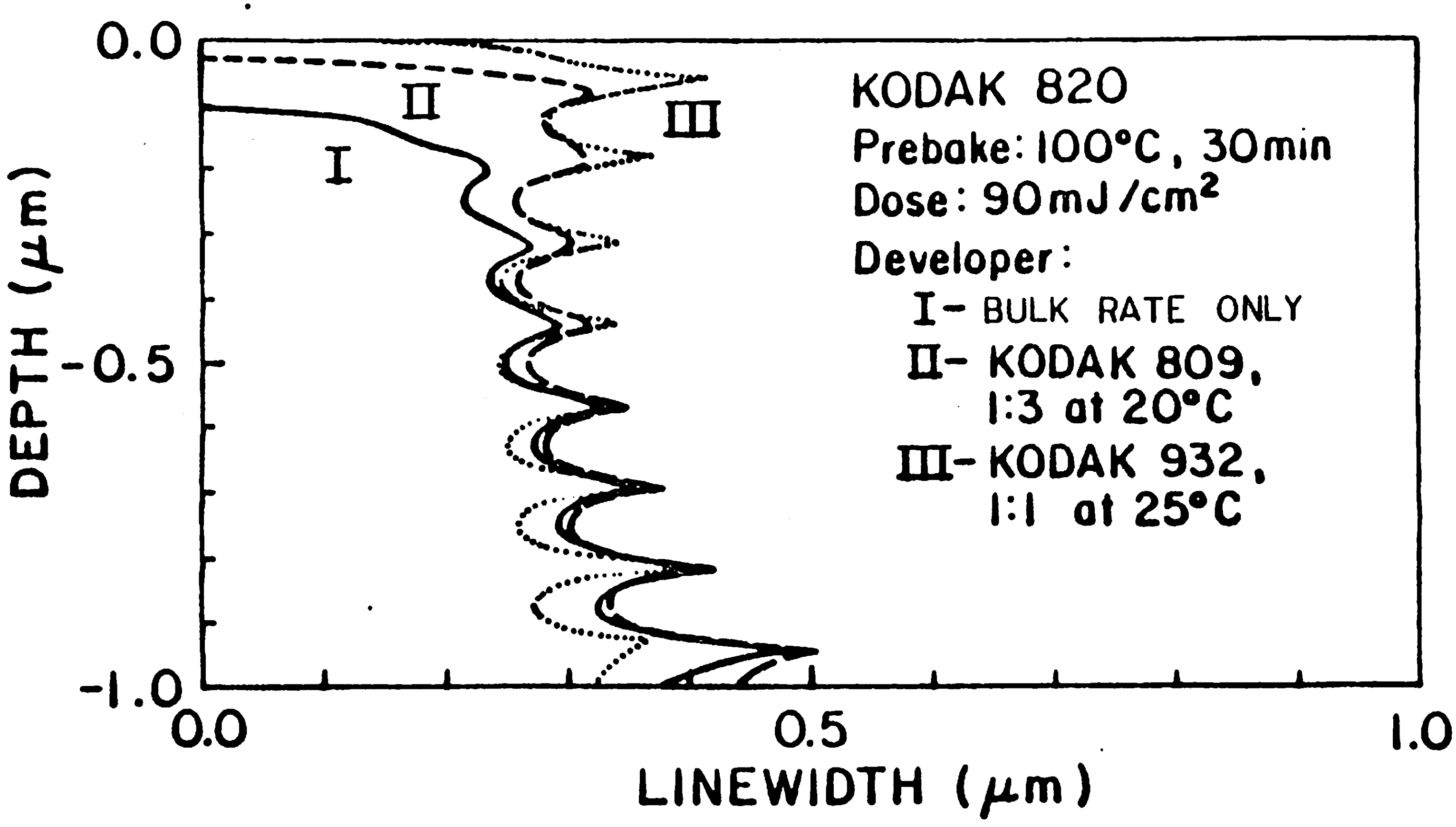

KODAK 820
Prebake : 100°C 30 min
Developer : KODAK 932
1:1 at 25°C
RATE (μm/sec)
10^0
10^-1
10^-2
10^-3
10^-4
10^-5
0.0
0.2
0.4
0.6
0.8
1.0
M

Fig. 3.4.9



0.0
-0.5
-1.0
DEPTH (μm)
0.0
0.5
1.0
LINEWIDTH (μm)
I
II
III
KODAK 820
Prebake: 100°C, 30min
Dose: 90mJ/cm²
Developer:
I - BULK RATE ONLY
II - KODAK 809, 1:3 at 20°C
III - KODAK 932, 1:1 at 25°C

Fig. 3.4.10

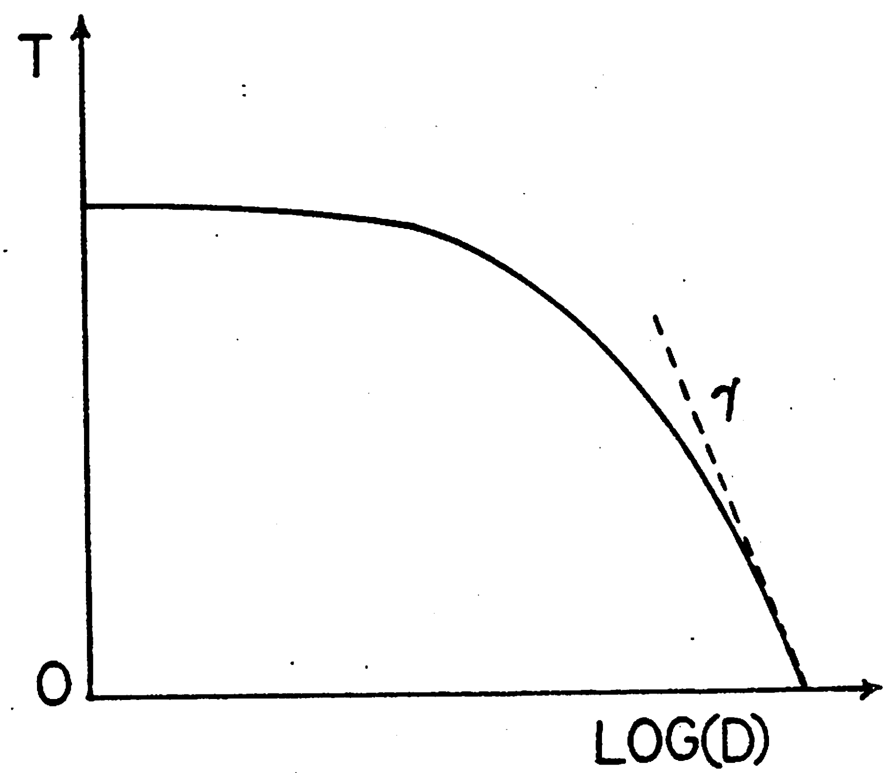


Fig. 3.5.1

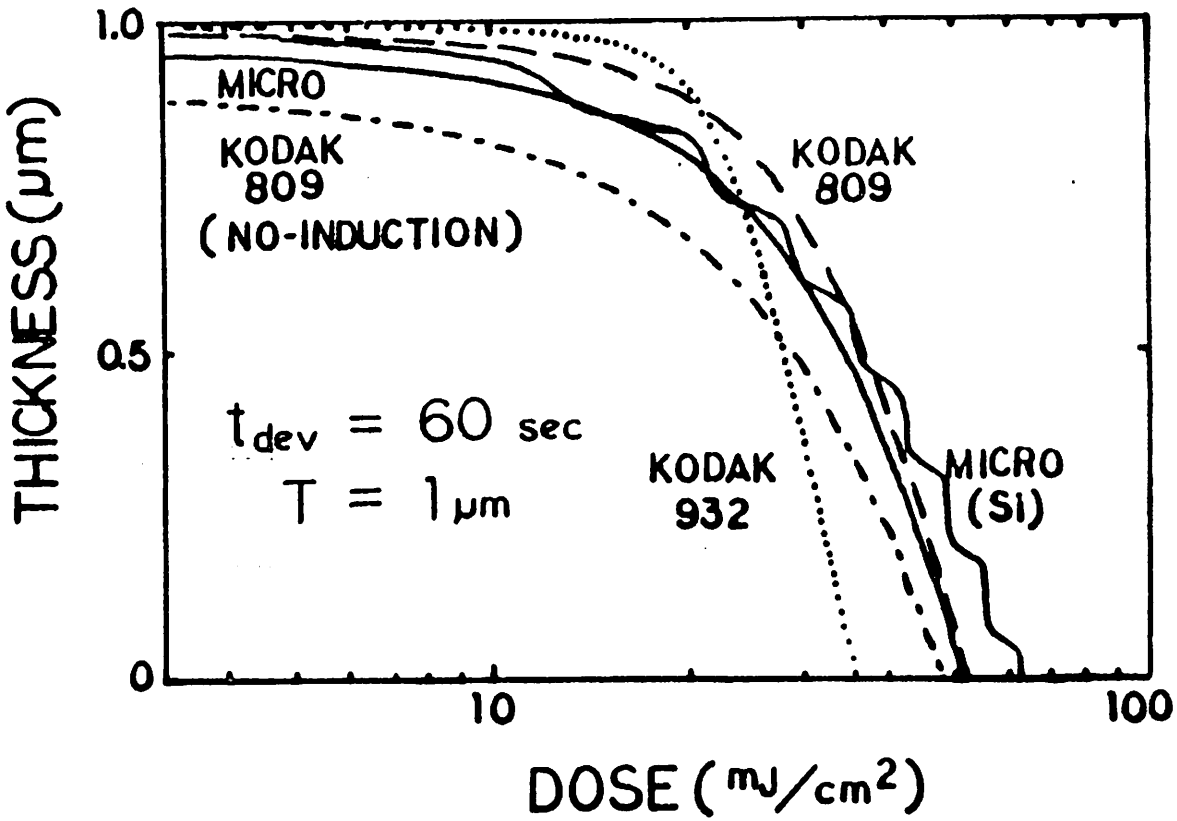

1.0
0.5
0
MICRO
KODAK
809
(NO-INDUCTION)
KODAK
809
t_{dev} = 60 sec
T = 1 µm
KODAK
932
MICRO
(Si)
THICKNESS (µm)
10
100
DOSE (mJ/cm²)

Fig. 3.5.2

CHAPTER 4

SURFACE RATE RETARDATION

4.1 Introduction

For conventional positive photoresists under normal processing conditions the development rate of the resist depends primarily on the exposure dose and increases with increasing dose. The energy absorption in U.V. exposure is highest at the top of the resist film because of the attenuation in the resist, leading to slanted resist profiles.

Various attempts have been made to extend the limit of optical lithography for vertical or even under-cut features (under-cut profiles are desirable for lift-off processes).[1–3] AZ-type photoresists prebaked at high temperature display close to vertical resist profiles.[1] Chlorobenzene treatment on the resist surface result in under-cut profiles.[2,3]

A development delay which is often seen with AZ-type photoresists prebaked at high temperature is termed "surface-induction" because the delay is confined within a very thin layer (~0.005 μm) at the top of the resist. For KODAK 820 resist under normal processing conditions the rate reduction is extended deep into the resist bulk (~0.25 μm). We will call this phenomenon "surface-rate-retardation". As discussed in Chapter 3 simulated resist profiles for KODAK 820 indicate that vertical or under-cut profiles can be obtained with surface-rate-retardation.

In this chapter the physical origin of the strong rate retardation for KODAK 820 resist is explored. Surface treatments to control the surface rate are also discussed.

4.2 Functional Characterization of Surface-Rate-Retardation

The surface-rate-retardation for KODAK 820 is not only observed with one particular type of developer. As shown in Fig. 4.2.1 the rate is retarded with typical types of developers which are commercially available. It might be expected that the prebake procedures affect the surface-rate-retardation behavior. However, in an experiment for KODAK 820 resist as illustrated in Fig. 4.2.2, no big difference was seen between a 30 minute oven bake at 100 ° C (normal baking) and a 1minute hot-plate bake at 120 ° C. A bake in a vacuum (25" Hg) also shows little difference.

The fundamental question about the strong rate-retardation is whether it is a resist property or it comes from the dynamics of chemical reaction between the resist and developer. A number of experiments were carried out to investigate the mechanism of retardation for unexposed KODAK 820 resist. In one simple experiment the development was terminated in the bulk of resist by water and reinitiated by developer. Fig. 4.2.3(a) shows the reflectivity variation during the development obtained from the dissolution rate measurement system (the details about the measurement system are given in Chapter 2). The flat zone in the middle of the reflectivity signal indicates that water is passing through the development cell. The oscillation period of the signal at the start of development is the longest (the slowest development rate at the surface). Such a long period never returns at the second start of development in the middle of the resist. This experiment suggests that the retardation effect is built-in to the resist.

The rate retardation with resist depth is more clear in Fig. 4.2.3(b). The development rates are plotted vs resist depth for two different types of developers. The dotted line is for a developer solution of one part of MICROPOSIT 351 and three parts of water. The dashed line is for undiluted KODAK 932 developer. For both cases the dissolution rates are minimum at the surface, increase with depth, and saturate. The dissolution rates are still continuous except for minor glitches which were introduced when the development was interrupted by water. The dissolution rates for several samples in which the top 0.5 μm of resist was removed by dipping in developer, rinsed with water, and dried with nitrogen-blow were also measured. Although a small development delay was introduced at the start of development the dissolution rate was not so retarded as it had been at the first surface.

Normal resist processing involves spin-coating and baking before development. In order to clarify which step introduces the rate-retardation a sample was spun, dried at room temperature for a month, and developed in undiluted KODAK 932 developer. The dissolution rate is plotted against resist depth in Fig. 4.2.4. No significant rate variation is seen from the surface to bulk. This suggests that the distribution of each component of resist material is uniform within the resist film and that the spin-coating is not responsible for the strong rate-retardation.

A sample was baked at 100 ° C for 15 minutes. The top 0.4 μm of the resist was etched away. The sample was rinsed with water, dried with N_2-blow, and baked again at 100 ° C for another 15 minutes. The dissolution rate for this sample is plotted as a function of resist depth in Fig. 4.2.5 (dotted line). The normal dissolution rate variation is also shown as a comparison (dashed line). The development behavior is not exactly the same as normal one but it suggests that the rate-retardation may be bake-related.

4.3 Surface Rate Control

To modify the surface rate chlorobenzene-dip prior to development is commonly used for AZ-type photoresists.[2,3] MICROPOSIT 1470 resist dipped in chlorobenzene for 10 minutes after exposure(200 mJ/cm^2) was developed in a solution of one part of MICROPOSIT 351 and five parts of water. The reflectivity variations during the development are shown in Fig. 4.3.1. The reflectivity variation for samples with chlorobenzene-dip (dotted line) is quite abnormal as compared with the reflectivity variation for normal samples (solid line). The distortion of the reflectivity signal is owing to non-uniformity as discussed in Chapter 2. The development time for the first oscillation period is much longer than the development time for the last three, indicating slow dissolution rate near the surface. According to the suggested model[3] chlorobenzene diffuses into the resist film, swelling it and forming a gel to the depth of the diffusion. The solvent and low-molecular-weight resin species diffuse out through the chlorobenzene resist gel, leaving a solvent-deficient layer near the surface.

KODAK 820 resist samples prebaked at 100 ° C for 30 minutes were developed in a solution of either one part of KODAK 809 and three parts of water or one part of MICROPOSIT 351 and five parts of water. A 10 minute dip in chlorobenzene after various exposure doses did not affect the surface rate. However, a very long time dip (~48 hour dip) in chlorobenzene after exposure (200 mJ/cm^2) resulted in a very slow surface rate as shown in Fig. 4.3.2(a). Samples prebaked at 80 ° C for 20 minutes exhibited a slow surface rate with a 10 minute dip in chlorobenzene after exposure (200 mJ/cm^2) as shown in Fig. 4.3.2(b). In both figures normal rate behavior (without chlorobenzene-dip) is included for comparison. These experiments imply that chlorobenzene still modifies the surface of KODAK 820 resist according to the model.[3]

Samples of KODAK 820 resist were exposed to the vapor of the resist solvent. For 30 minute vapor-soak the development rate is very high everywhere in the resist. For 3 minute vapor-soak, as shown in Fig. 4.3.3, the development rate is very high near the surface where the solvent vapor penetrated (solid line). But beyond the penetration depth the development rate drops to the normal value. When the vapor-soaked sample was baked again (at 100 ° C for 15 minutes) the rate retardation returned (dotted line). Clearly the solvent vapor plays a role in surface rate control.

References

1. F.H.Dill and J.M.Shaw, "Thermal Effects on the Photoresist AZ1350J," IBM J. Res. Develop, May, 1977.

2. M.Hatzakis, B.J.Canavello, and J.M.Shaw, "Single-Step Optical Lift-Off Process," IBM J. Res. Develop. Vol 24, No.4, July 1980.

3. R.M.Halverson, M.W.McIntyre, and W.T.Motsiff, "The mechanism of Single-Step Lift-Off with Chlorobenzene in a Diazo-Type Resist," IBM J, Res. Develop. Vol 26, No.5, Sept., 1982.

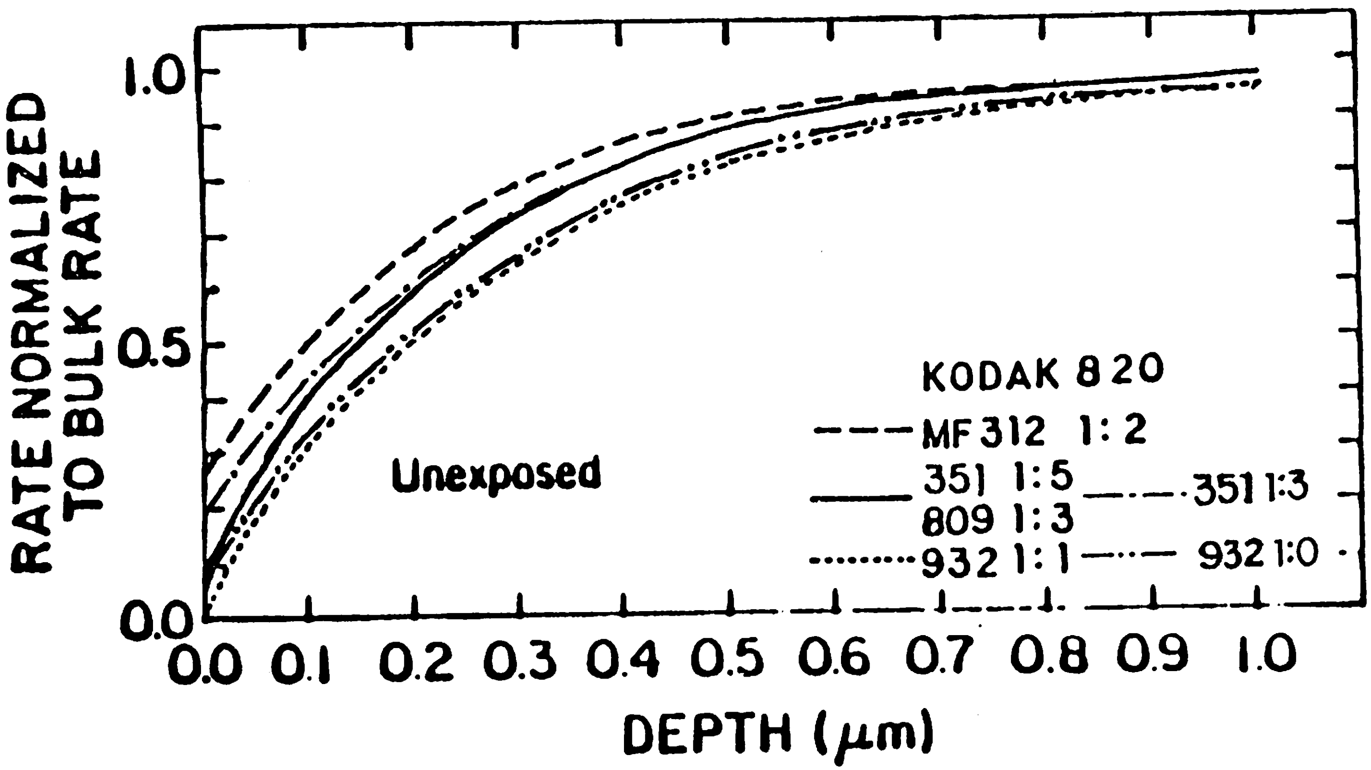
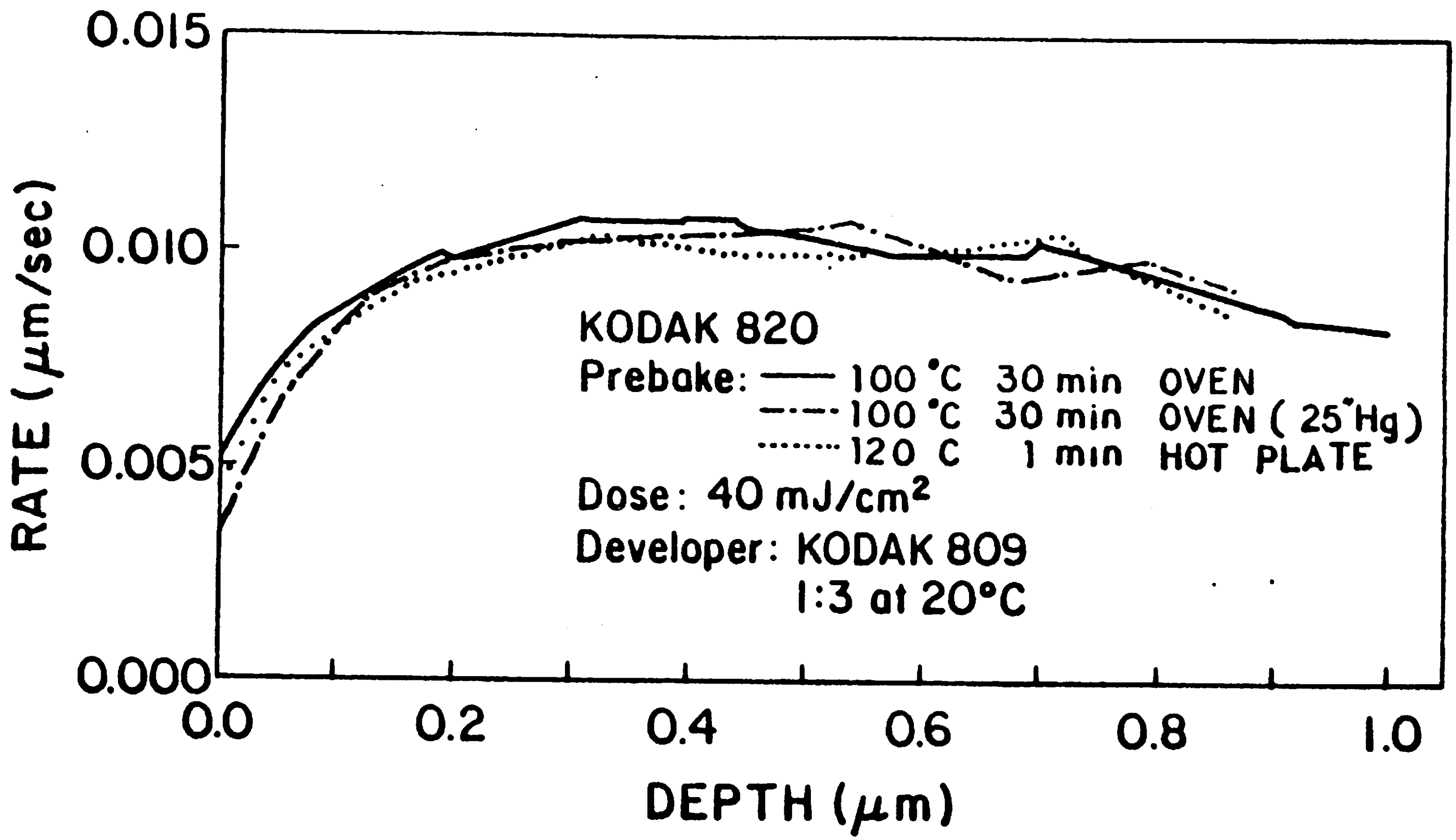

RATE NORMALIZED TO BULK RATE
1.0
0.5
0.0
0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
DEPTH (μm)
Unexposed
KODAK 820
MF312 1:2
351 1:5
809 1:3
932 1:1
351 1:3
932 1:0

Fig. 4.2.1



0.015
0.010
0.005
0.000
RATE (μm/sec)
0.0
0.2
0.4
0.6
0.8
1.0
DEPTH (μm)
KODAK 820
Prebake: 100 °C 30 min OVEN
100 °C 30 min OVEN (25"Hg)
120 C 1 min HOT PLATE
Dose: 40 mJ/cm²
Developer: KODAK 809
1:3 at 20°C

Fig. 4.2.2

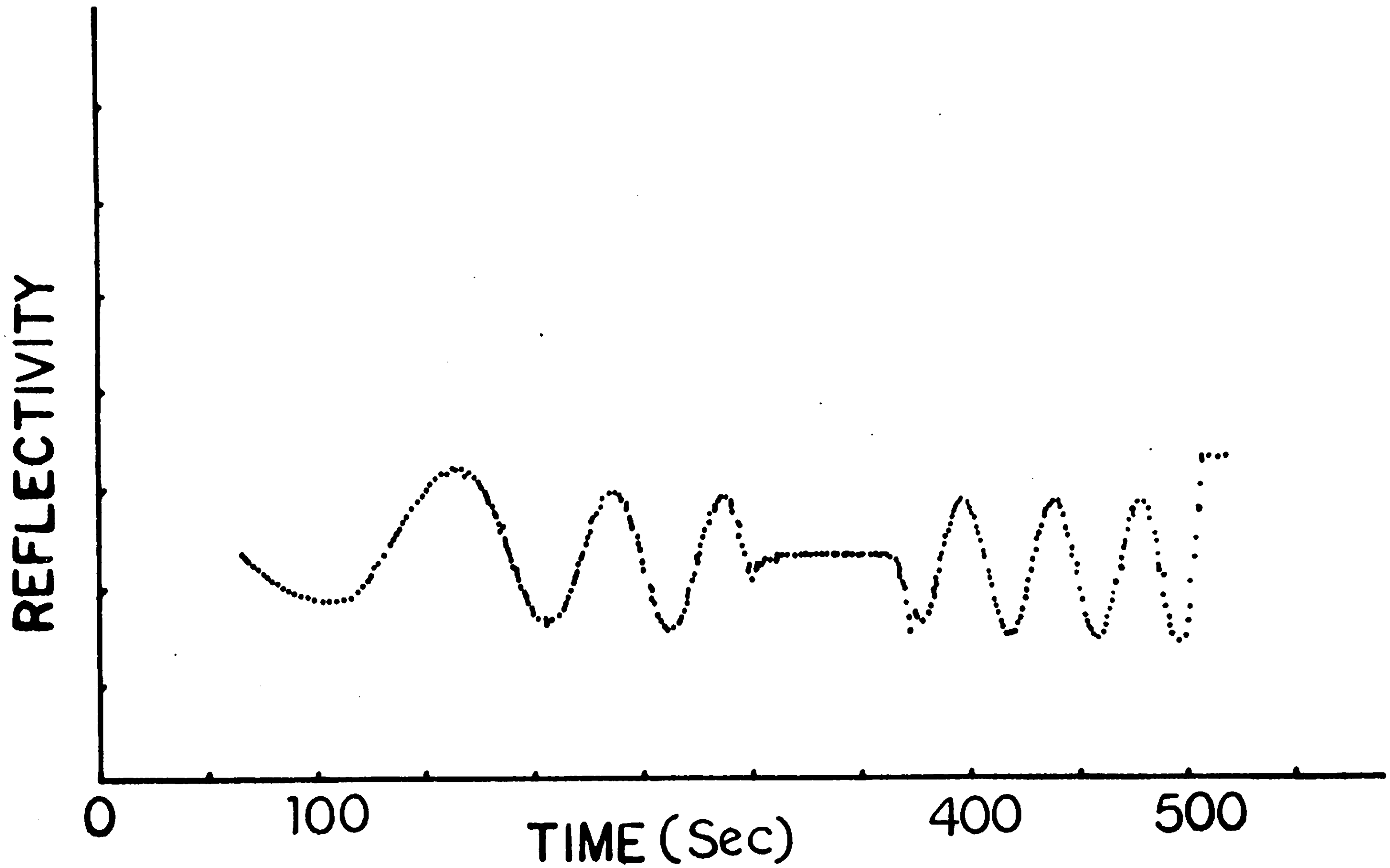


Fig. 4.2.3(a)

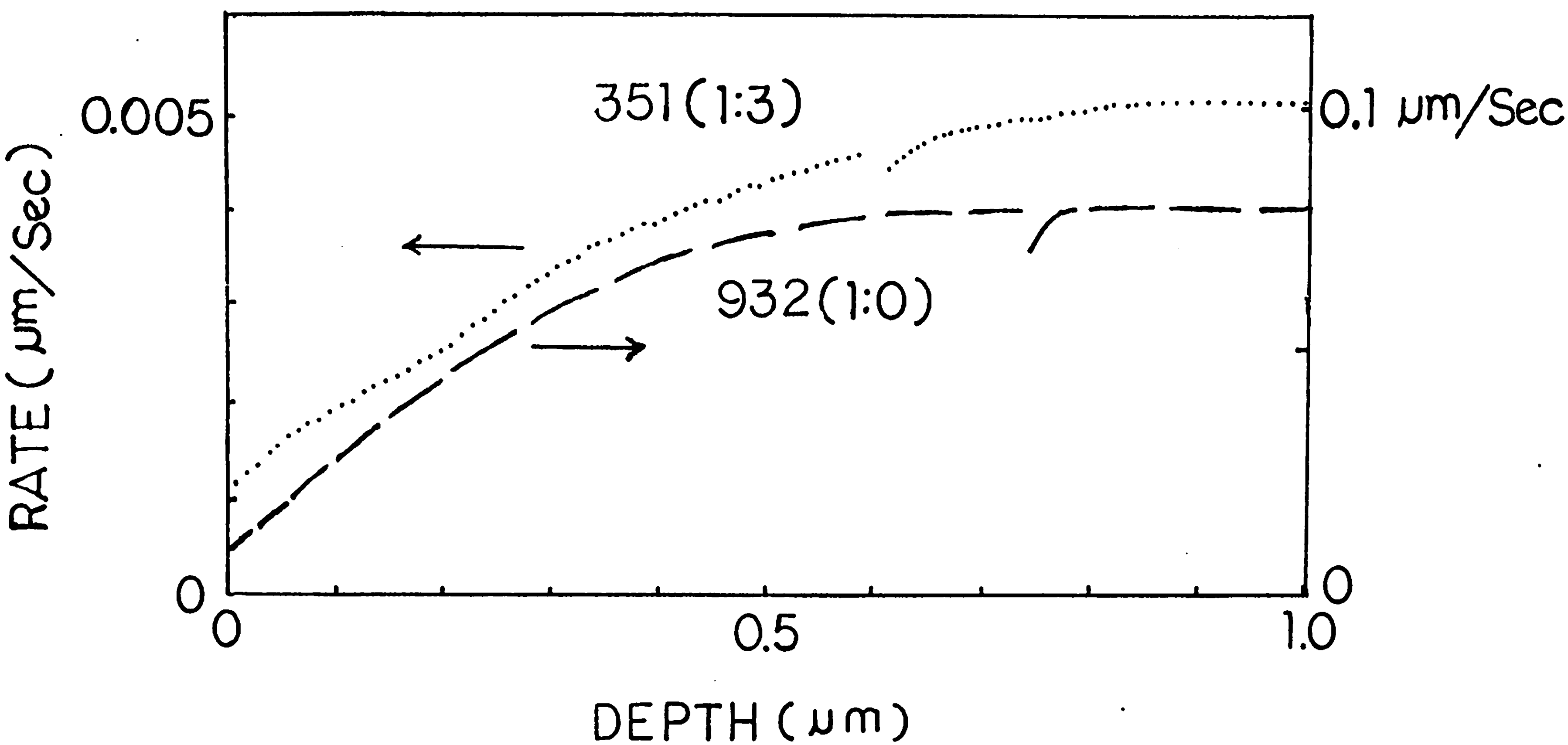


Fig. 4.2.3(b)

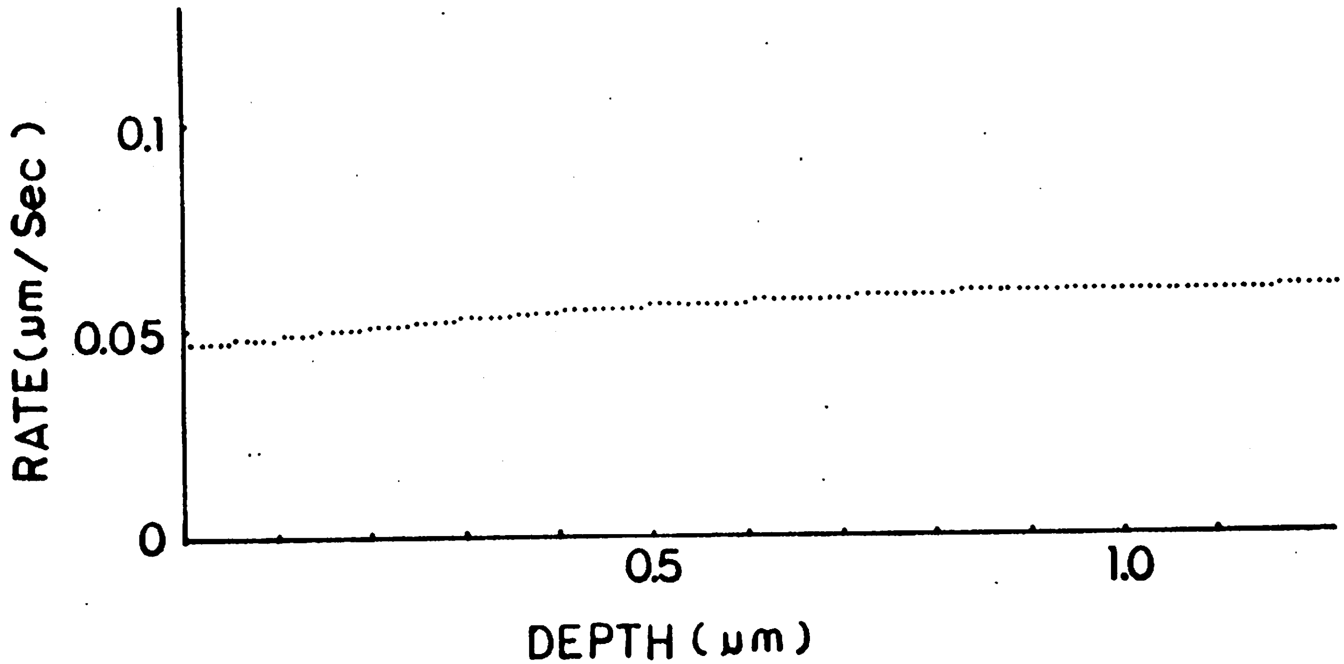


Fig. 4.2.4

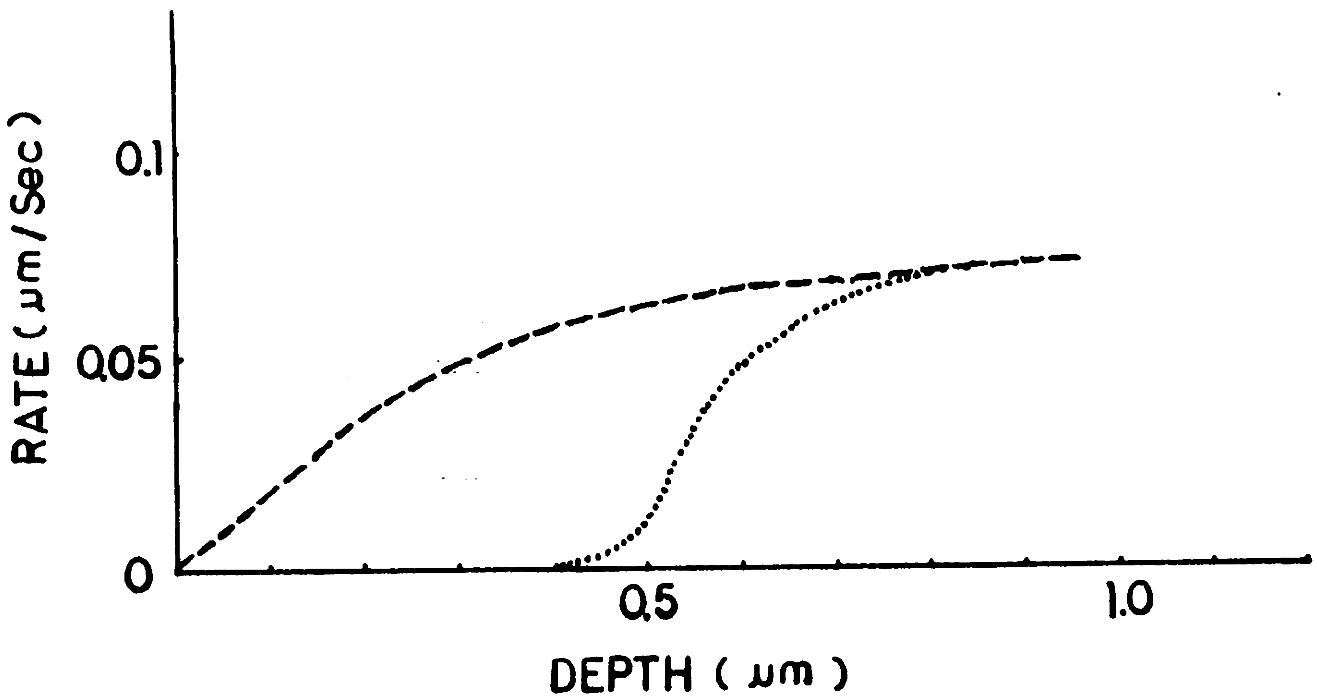


Fig. 4.2.5

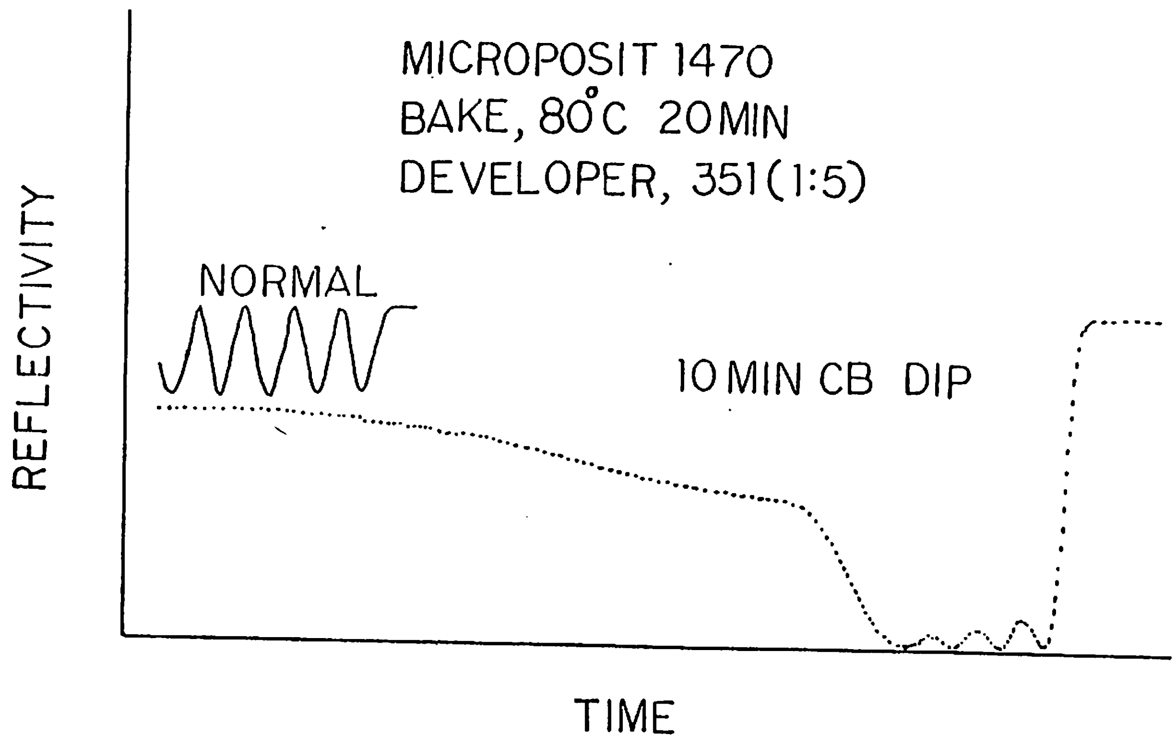

MICROPOSIT 1470
BAKE, 80°C 20MIN
DEVELOPER, 351(1:5)
NORMAL
10MIN CB DIP
REFLECTIVITY
TIME

Fig. 4.3.1

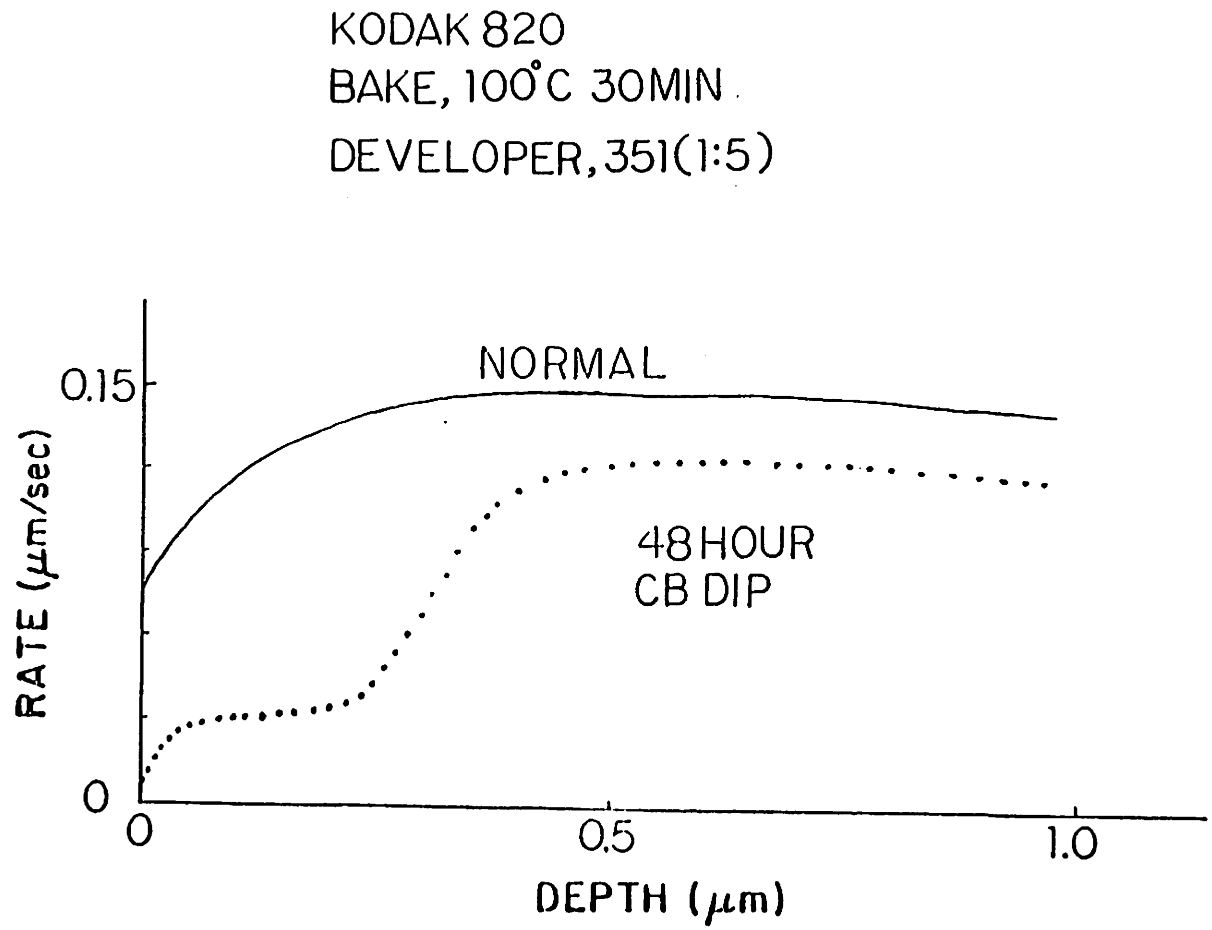

KODAK 820
BAKE, 100°C 30MIN
DEVELOPER, 351(1:5)
NORMAL
48 HOUR
CB DIP
0.15
0
RATE (μm/sec)
0
0.5
1.0
DEPTH (μm)

Fig. 4.3.2(a)

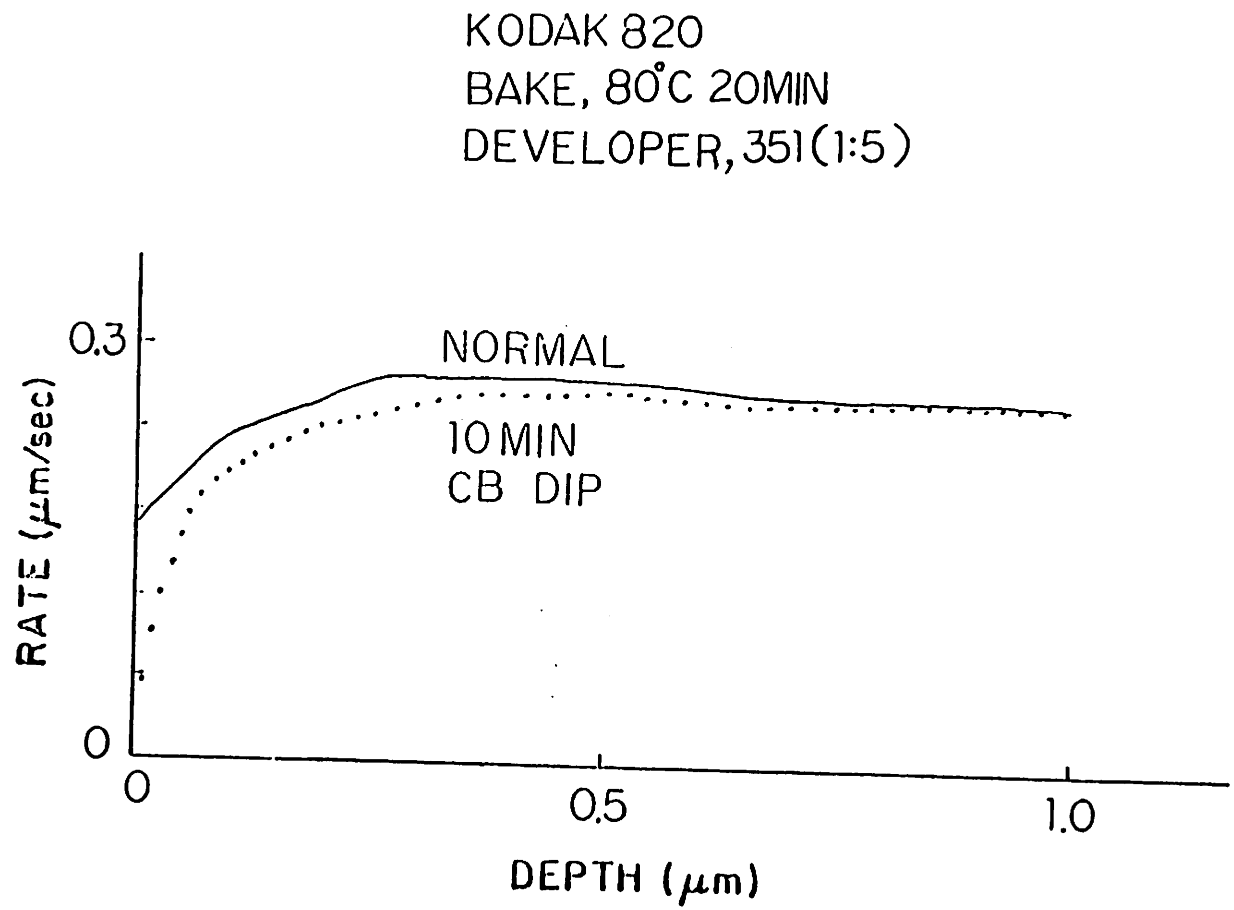

KODAK 820
BAKE, 80°C 20MIN
DEVELOPER, 351(1:5)
0.3
0
NORMAL
10 MIN
CB DIP
RATE (μm/sec)
0
0.5
1.0
DEPTH (μm)

Fig. 4.3.2(b)

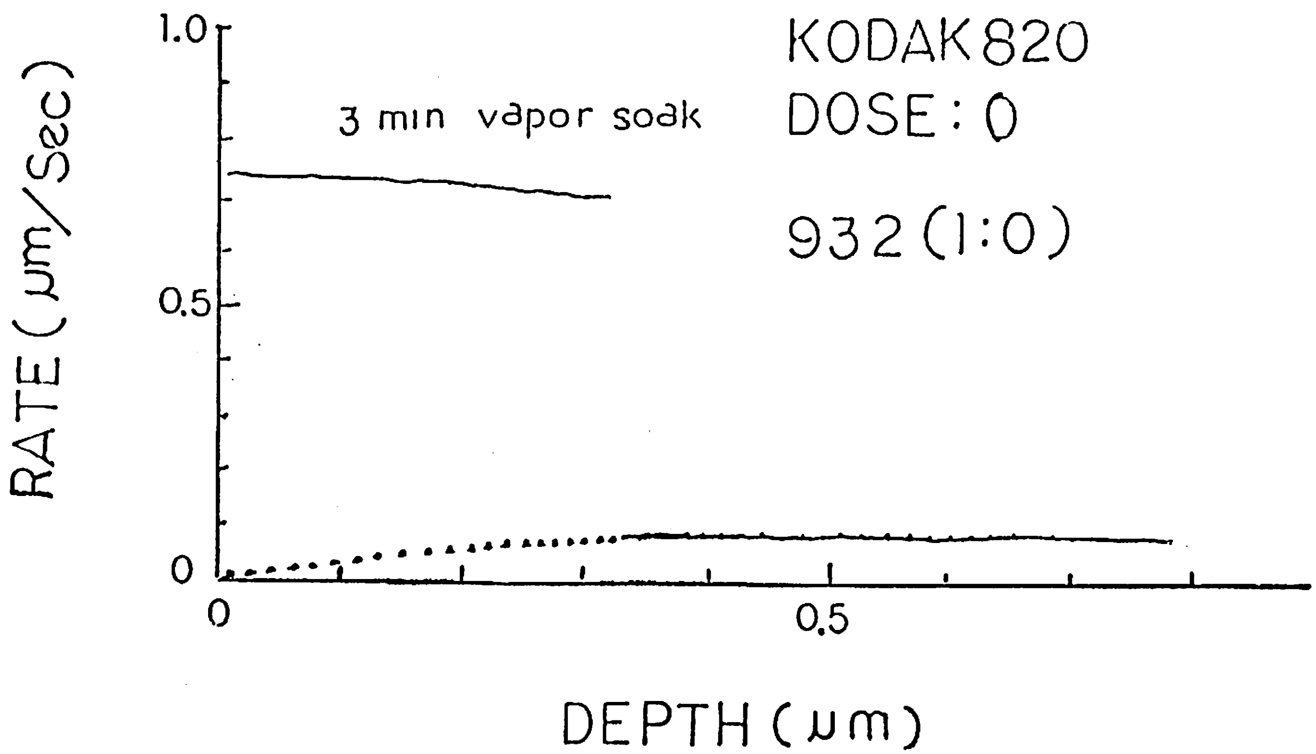

KODAK 820
DOSE: 0
3 min vapor soak
932 (1:0)
1.0
0.5
0
RATE (μm/Sec)
0
0.5
DEPTH (μm)

Fig. 4.3.3

CHAPTER 5

DESIGN ALGORITHM FOR RESIST FEATURES

5.1 Introduction

SAMPLE[1] has been used to simulate resist profiles for a given resist process. In order to design resist features, however, it is not practical to find an optimum resist process by simulating all the profiles under a number of processing conditions. The problem is one of computing time as well as a convergent algorithm. Most of the CPU time is consumed in the development simulation based upon the string development model.[2] Each development simulation starts from the top surface and calculates the etch front over all points in the resist.

In this chapter the feasibility of using a simple model to estimate certain features of the resist profile. The technique used is especially appropriate for resist on a reflecting substrate. An algorithm to obtain the slope and linewidth is described and the results are discussed.

5.2 Simplified Development Algorithm

The development of resist in the present SAMPLE is based on the string development algorithm originally devised by R.Jewett.[2] It follows a development contour by keeping track of the positions of the points on the developing string. This process requires the large amount of computing time. Typically the resist on silicon wafer creates standing wave pattern and the etch front is extremely non-linear. For a good approximation to the exact boundary, the string points around the sharp points should be divided into many segments so that any curve would be well defined. Obviously the computing time increases with every segment. There are several other problems associated with the string point process around the sharp points and the resist boundary.[2,3] The computing time is further increased by dealing with these problems.

For the purpose of finding only the slope and the position of the resist profile it may be sufficient to follow only fastest-moving points. In a typical projection system the maximum intensity occurs at the middle of the clear area. As sketched in Fig. 5.2.1(a) the horizontal intensity varies on the resist surface. For resists on a reflecting substrate the standing wave effect creates a periodic pattern of maximum and minimum energy coupling in the z direction. The resist develops fastest at the points of maximum intensity. Lateral development proceeds from these high-intensity positions, leaving the periodic sharp points on the boundary as shown in Fig. 5.2.1(b). Choosing two points (A and B in Fig. 5.2.1(b)) one can estimate the resist feature as follows.

$$slope(t) = \tan^{-1} \frac{z_A - z_B}{x_B(t) - x_A(t)} \tag{5.2.1}$$

As indicated the slope is a function of the development time. x is given by

$$x = \int_{t(0,z)}^{t(x,z)} R(x,z)\,dt \tag{5.2.2}$$

where R(x,z) is the rate function depending upon the position as well as M value at the position and t(x,z) is the development time to reach the point (x,z) given by

$$t(x,z) = \int_0^z \frac{dz'}{R(0,z')} + \int_0^x \frac{dx'}{R(x',z)} \tag{5.2.3}$$

5.3 Design Algorithm

Using the standard SAMPLE optical algorithm and the development algorithm described in the previous section line-edge slopes are rapidly calculated for each resist process until a desired slope is matched to a calculated slope. A user specifies the optical constants to run the IMAGE and EXPOSE subroutines of SAMPLE. He also specifies the line-edge slope and resist linewidth of a desired resist feature as defined in Fig. 5.3.1.

For a selected set of resist parameters (A,B,C, and R parameters) the rate function, R(M,z), and the M distribution within the resist, M(x,z), are combined to produce the rate as a function of position, R(x,z). The M distribution is provided by the SAMPLE.[1] The slope is calculated with varying dose. At each dose the development time is calculated according to equation (5.2.3) with x=given linewidth and z=given thickness. The slope at this development time is calculated according to equation (5.2.1) through (5.2.2). If a calculated slope is not matched to the given slope within the dose range of the EXPOSE another set of resist parameters is selected and this whole procedure continues. When the given slope is matched, the process recipe is given as the dose, the development time, and the resist process corresponding to the selected resist parameters.

The details for numerical integration of equations (5.2.2) and (5.2.3) are given as follows. For a given rate array R vs x (or R vs z) the integration is performed within each division of x (or z) as shown in Fig. 5.3.2. The increment is chosen depending upon the rate as

```
dt=(x(i+1)-x(i))/maxrate(r(i),r(i+1))/ndev
```

The highest rate in each division is used to determine the maximum time interval. This time interval is divided by an arbitrary number, ndev, to give a small enough time increment. Thus the time increment is varying in each division. The development length, x, progresses by x=x+rate*dt and the development time by t=t+dt. The rate is updated whenever x is updated:

rate=r(i)+(r(i+1)-r(i))*(x-x(i))/(x(i+1)-x(i))

When the process moves from one division, x(i) and x(i+1), to next division, x(i+1) and x(i+2), the development time is trimmed to eliminate the accumulative error as

t=t-(x-x(i+1))/rate

and also x as

x=x(i+1)

The FORTRAN codes to implement this algorithm, input, and output format are found in Appendix E.

5.4 Results

The resist profile based on the simplified development algorithm is shown in Fig. 5.4.1 (The positions of maximum intensity are connected by dotted line). This is obtained for the given slope of 82 degree and the linewidth of 0.8 μm for an image of 1 μm line and spaces with NA=0.28, σ=0.7, and δ(defocus)=1.5 by searching for 3 different resists and 10 different developers in 32 seconds. Each development is done for 20 different doses. An optimum process is found to be KODAK 820 resist prebaked at 100 ° C for 30 minutes. The dose is 60 mJ/cm^2 and the development time is 100 seconds. The developer solution is one part of KODAK 809 developer and three parts of water. The development temperature is 20 ° C. To compare with the detailed resist profiles this process information is entered in the input of SAMPLE. The result of SAMPLE is also shown in Fig. 5.4.1. (solid line) There is no significant difference between the two profiles in terms of the line-edge slope and resist linewidth. Another example of comparison is given in Fig. 5.4.2 for MICROPOSIT 1470. This is searched for the slope of 75 degree at 0.8 μm resist line. The dose is found 41.6 mJ/cm^2 and the development time 52.6 seconds. MICROPOSIT 351 developer diluted 1:3 is selected and development temperature is 20 ° C.

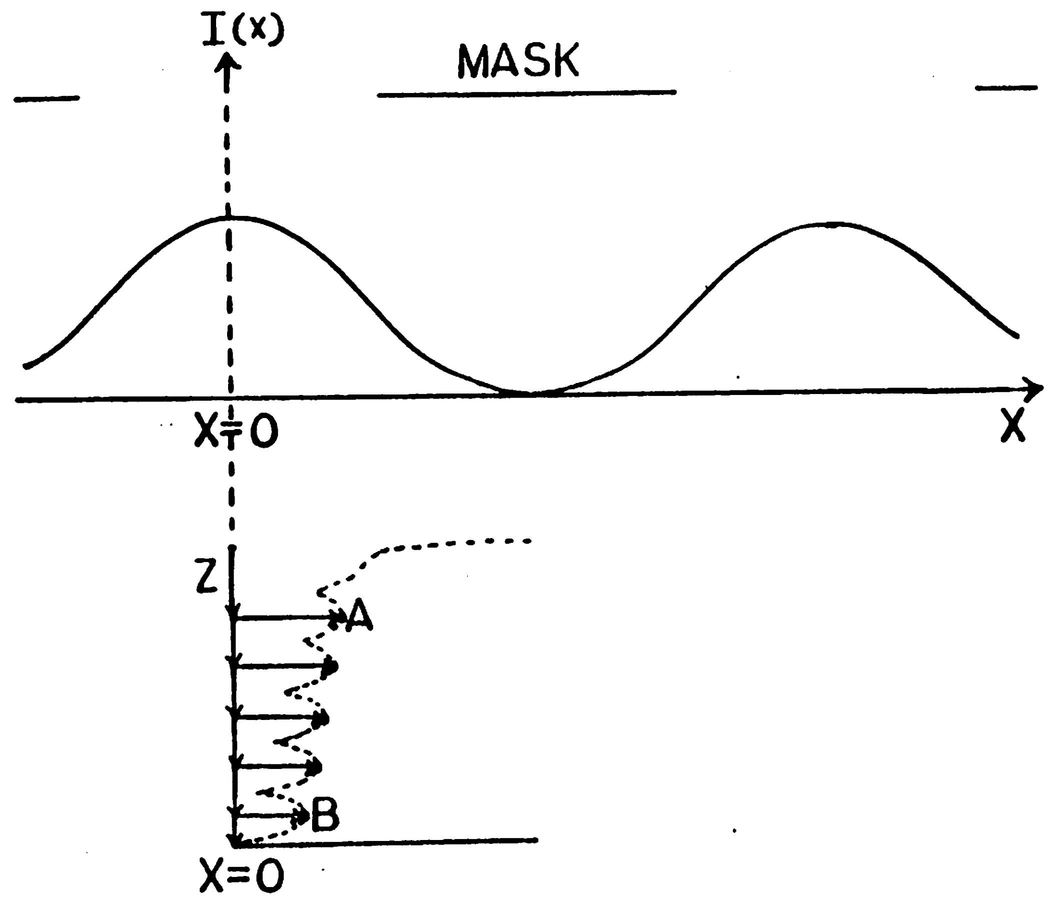
One application of the simplified development algorithm is a rapid evaluation of process sensitivity to resist features. In Fig. 5.4.3(a) the slope is plotted against dose with ΔCD as parameter. (ΔCD is the difference between the resist linewidth and the linewidth on the mask) The slope increases with large ΔCD over all exposure. With the bulk rate only the slope increases rapidly in the low exposure range. At doses higher than 100 mJ/cm^2 the slope is not very sensitive to dose variation. Including the depth dependence with ΔCD=0.2 μm the resist slope is not very sensitive to doses ranging 50 to 200 mJ/cm^2 (Fig. 5.4.3(b)). Clearly the rate-retardation near the surface makes the resist feature less sensitive to dose variation.

References

1. W.G.Oldham, et al, "General Simulator for VLSI Lithography and Etching", IEEE ED-26, No. 4, 1979

2. R.Jewett, "A String Model Etching Algorithm", Memo No. UCB/ERL M79/68, Oct., 1979

3. M.M.O'Toole, "Simulation of Optically Formed Image Profiles in Positive Photoresist", Ph.D. dissertation, EECS, U.C.Berkeley, 1979



(a)

(b)

Fig. 5.2.1(a) & (b)

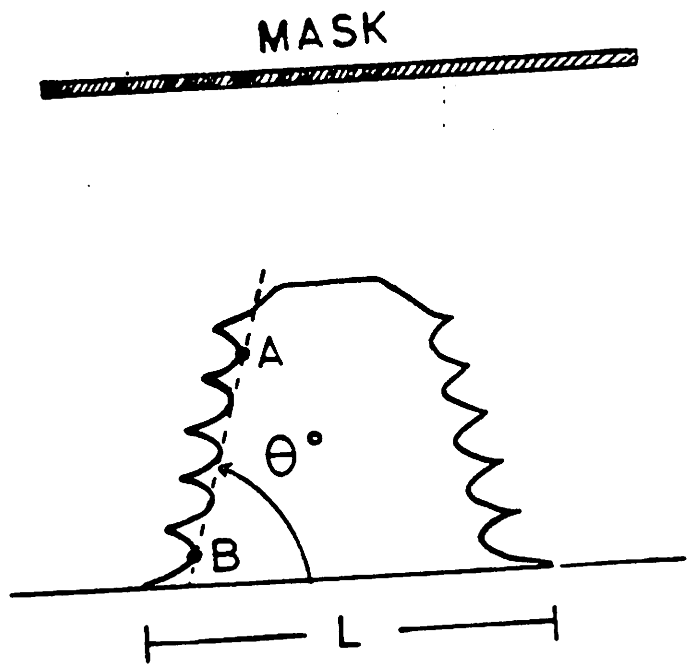


Fig. 5.3.1

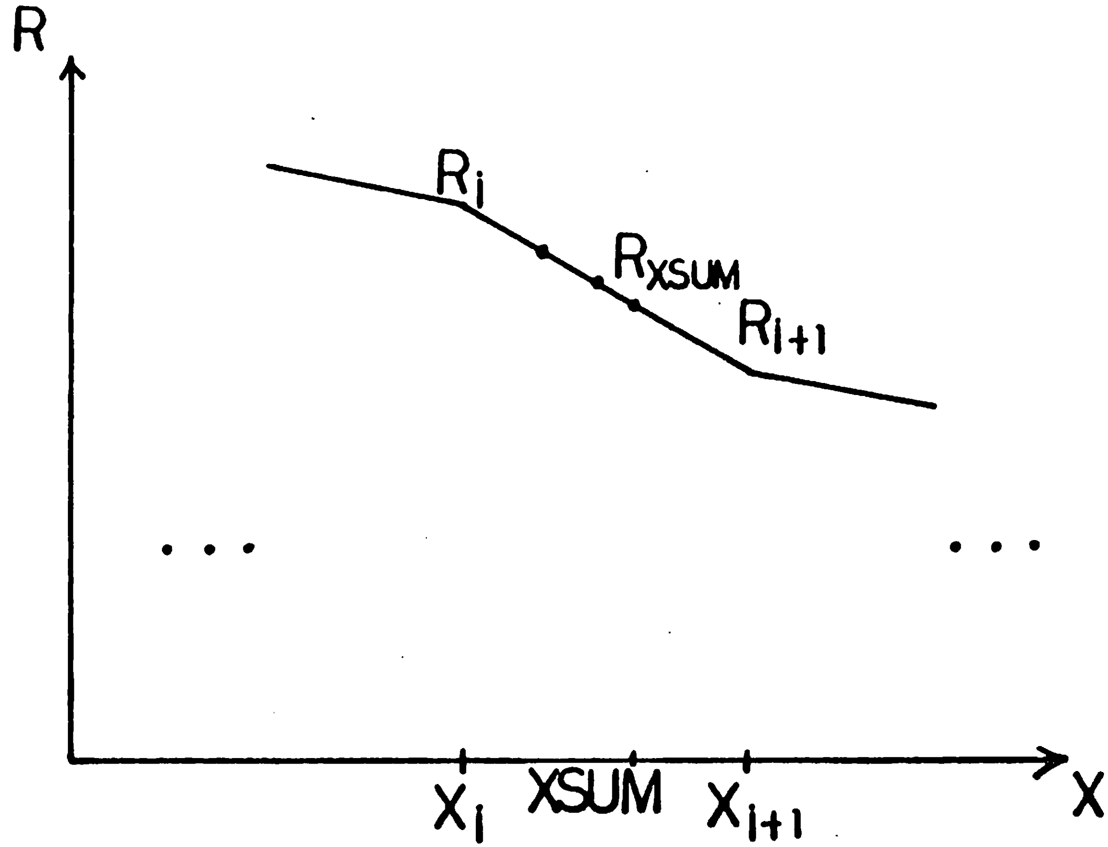


Fig. 5.3.2

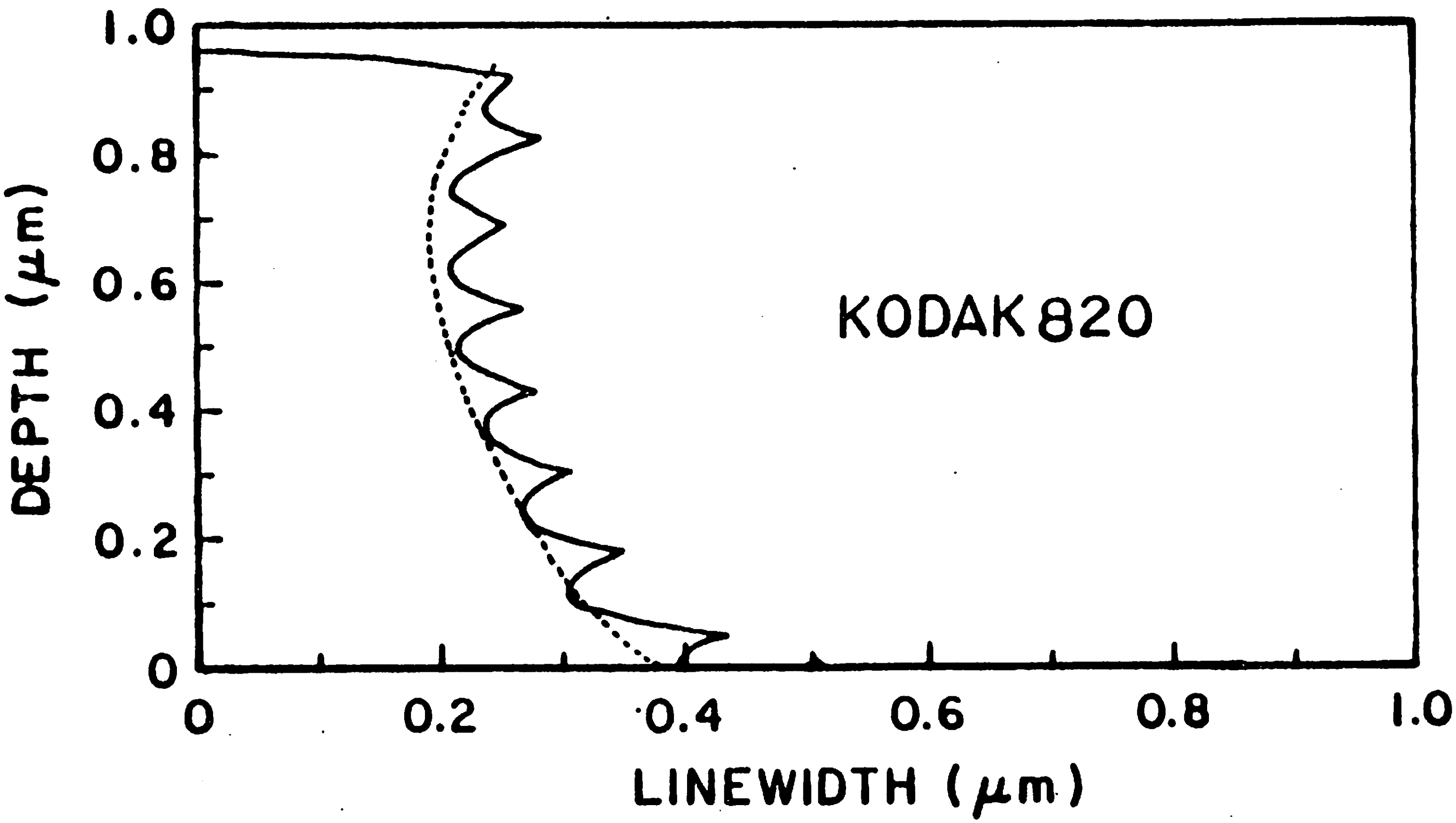


Fig. 5.4.1

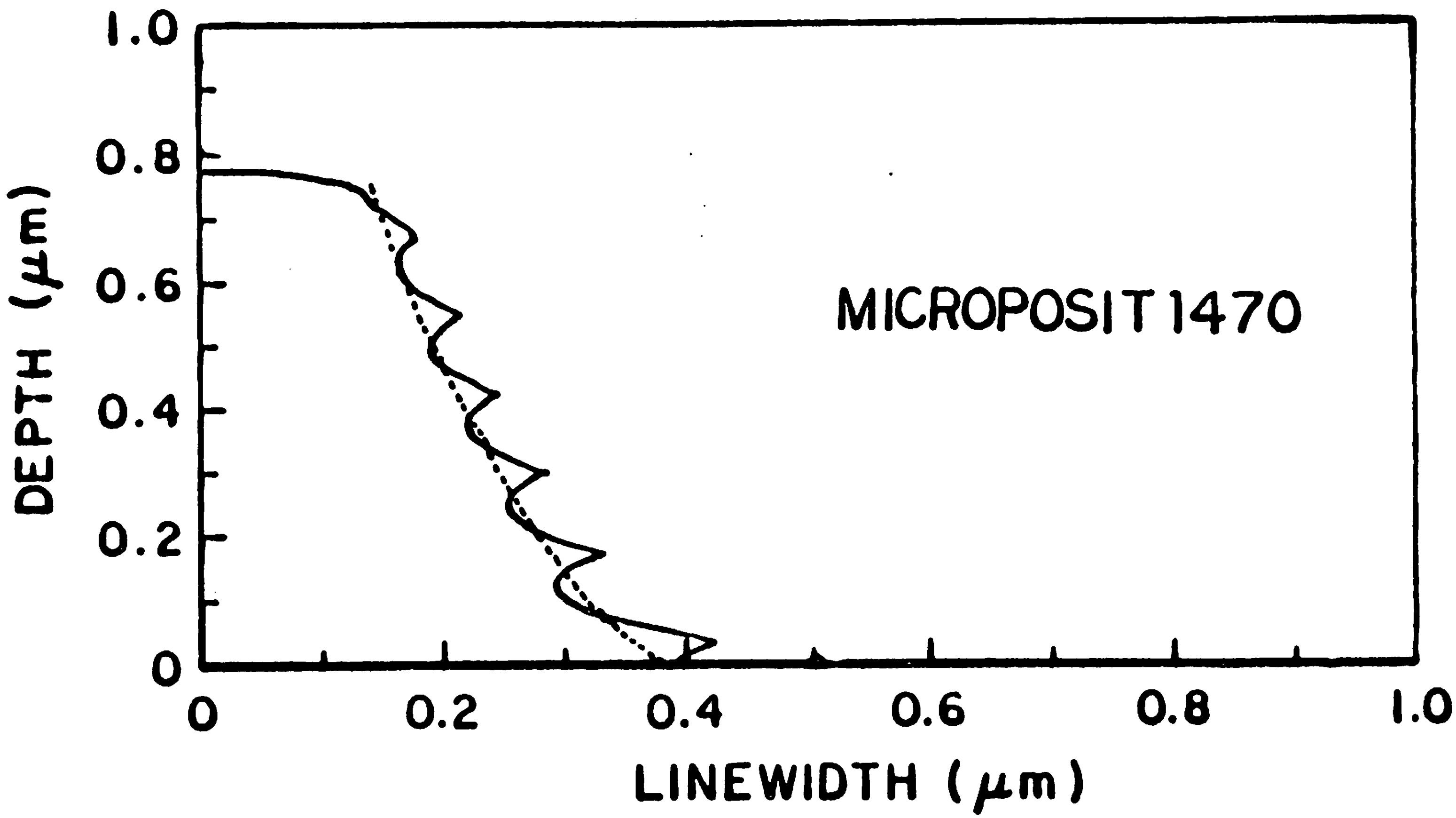


Fig. 5.4.2

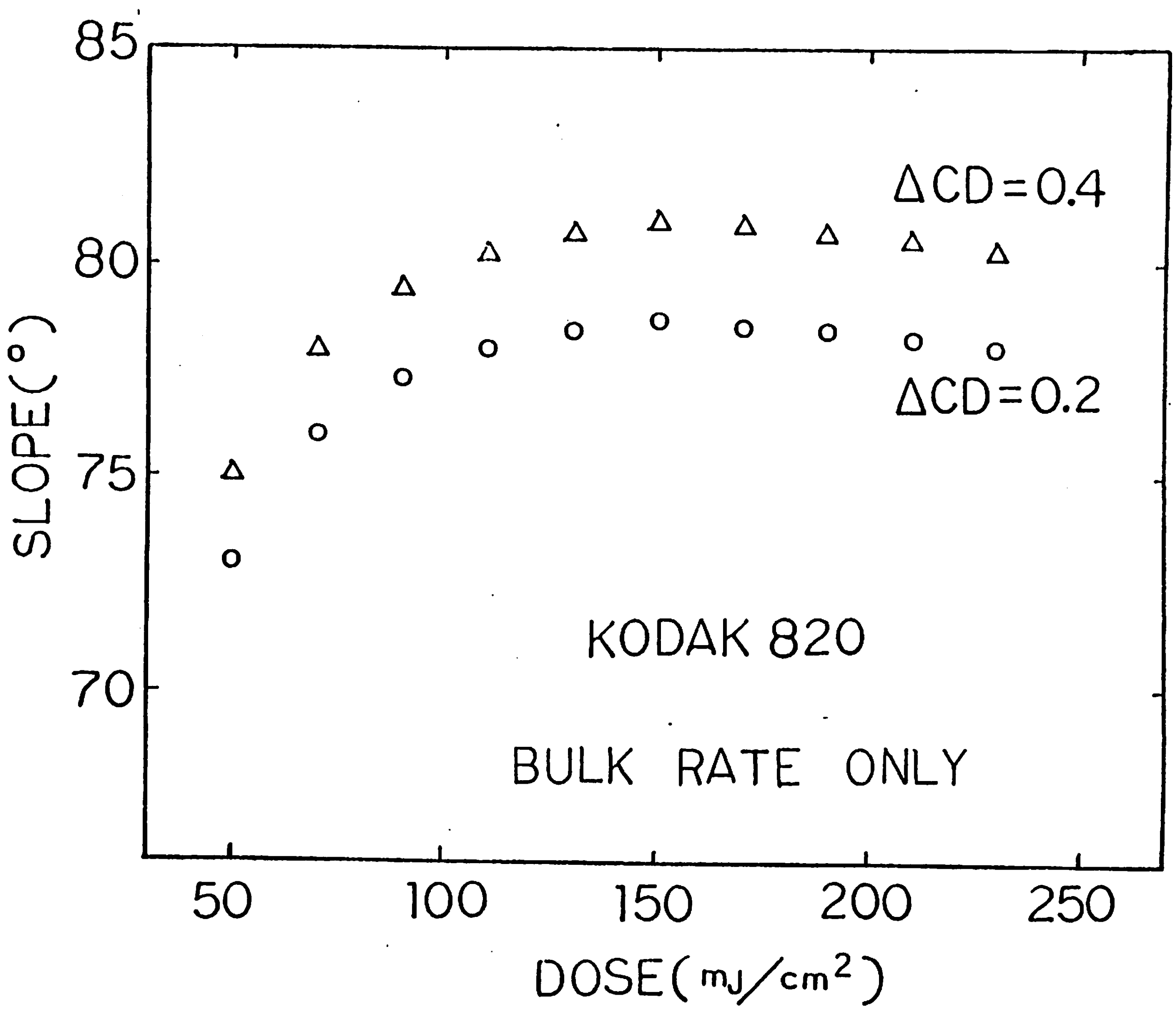

85
80
75
70
SLOPE(°)
ΔCD=0.4
ΔCD=0.2
KODAK 820
BULK RATE ONLY
50
100
150
200
250
DOSE(mJ/cm²)

Fig. 5.4.3(a)

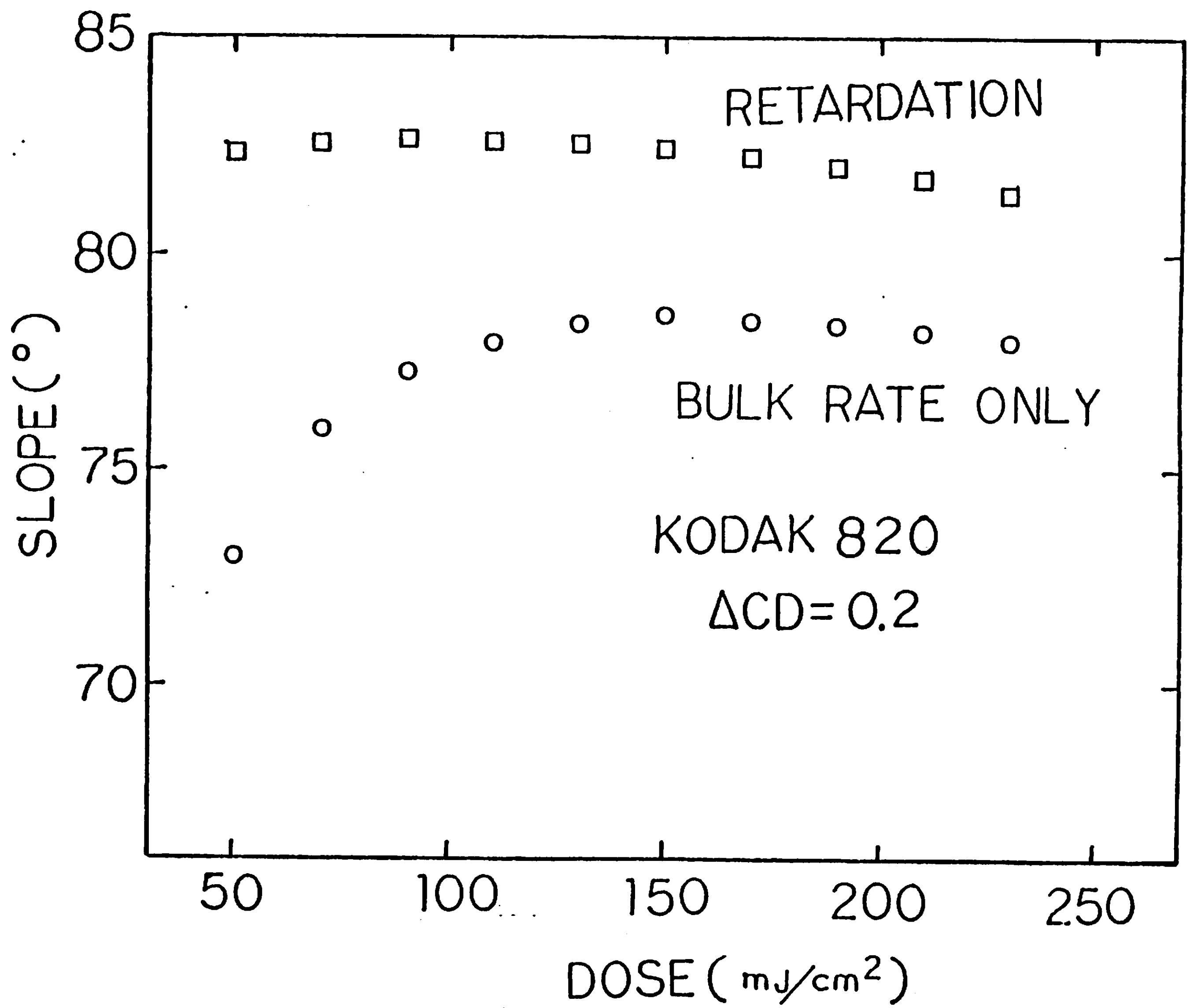

SLOPE (°)
85
80
75
70
50
100
150
200
250
DOSE (mJ/cm2)
RETARDATION
BULK RATE ONLY
KODAK 820
ΔCD= 0.2

Fig. 5.4.3(b)

Appendix A

The BASIC Program for Exposure

```
10 REM Program Name: EXPOSE.BAS  by Deok J. Kim   version: 2.6      Jan 22 '83
20 REM Purpose: to collect data by controlling two Tecmar A/D converters
30 REM          LABMASTER for data and LABTENDER for reference
40 REM maximum data rate for the safety = 50 only for first period
50 REM maximum data rate = 25          with graph in real time
60 REM The first period can be divided by two automatically.
70 DIM H(3000),L(3000),P(3000)    'have room to spare, check with "print fre(x)"
80 DEF FNBCD(X) = X+6*(X\10)    'convert an integer from binary to BCD form, thi
s function is only good for 8 bits data thus 99 is the maximum value
90 REM
100 PRINT " "
110 PRINT "=== TURN OFF ALL LIGHTS EXCEPT EXPOSURE ==="
120 PRINT " "
130 PRINT "******************************************"
140 PRINT "* Check the reference at 4.0 - 4.9 volts *"
150 PRINT "* and   the signal    at 4.5 - 6.5 volts *"
160 PRINT "******************************************"
170 PRINT " "
180 HL=4! :HH=7! :PL=3.6 :PH=4.95  'power limit for safety
190 DP=.05 :P1=1-DP :P2=1+DP       '5 % variation in power allowed
200 PWCUT=.1                       'exposure power limit (mW/cm2)
210 PRINT "Adjust the gain for the right voltages. Hit any key "
220 IF INKEY$ = "" THEN 220
230 PRINT "Hit any key when done "
240 GFLAG$="n"
250 GOSUB 4380
260 PRINT " "
270 PRINT " "
280 IF WRN=1 THEN PRINT "Check the power limit for safety.":GOTO 100
290 PRINT "Enter the signal gain in Kohm. ";:INPUT GAIN
300 IF GAIN < 1 THEN PRINT "Remember the gain is ";GAIN
310 PRINT " "
320 PRINT "Now measure the light intensity and ratio. Hit any key "
330 IF INKEY$ = "" THEN 330
340 PRINT "Choose Exposure Wave length ..."
350 PRINT "type   1  for 4358 "
360 PRINT "       2      4047 "
370 PRINT "       3      3650 "
380 PRINT "... select one ";:INPUT LENG
390 IF LENG < 1 OR LENG > 3 THEN 340
400 IF LENG=1 THEN SE=.2       'UV100B photodiode
410 IF LENG=2 THEN SE=.15
420 IF LENG=3 THEN SE=.14
430 DF=1/1.525              'distant factor between diode and resist surface(air)
440 AREA=.051                 'window area for UV100B
450 VTOMW=DF*SE*AREA*GAIN  'conversion factor : VOLT/(mW/cm2)
460 GFLAG$="y" :KP=150
470 GOSUB 4380
480 IF WRN=1 THEN PRINT "Check the power limit for safety."
490 AVI=HSUM/KP :AVG=RSUM/KP
500 PRINT " "
510 PRINT "Average Intensity of Signal (Air)      = ";AVI;:PRINT " mW/cm2"
520 IF AVI < PWCUT THEN PRINT "Average power is lower than the minimum ";PWCUT
530 PRINT "Average Ratio of Signal to Reference   = ";AVG
540 DEV=0  :NG=0
550 FOR J=1 TO KP
560 DEV=DEV+(H(J)-AVI)^2/AVI
570 IF H(J) >= P2*AVI OR H(J) <= P1*AVI THEN NG=NG+1
```

```
580 NEXT J
590 SDEV=SQR(DEV)/KP*100    'standard deviation in %
600 PRINT "Standard Deviation per Power   (%)       = ";SDEV
610 PRINT "# of out-of-average power ( +- ";DP*100;:PRINT " % ) = ";NG
620 PRINT " "
630 IF SDEV > .1 THEN PRINT "REPLACE BULB FOR BETTER UNIFORMITY !!"
640 PRINT " "
650 PRINT "Measure the reflectance ... "
660 PRINT "Do you know the refractive index of the matched substrate [y/n] ";
670 INPUT RIN$
680 IF RIN$ <> "y" AND RIN$ <> "n" THEN 660
690 IF RIN$="y" THEN 770
700 PRINT "Place the matched substrate on the holder. Then hit any key."
710 IF INKEY$ = "" THEN 710
720 GFLAG$="y"
730 GOSUB 4380
740 AVGR=RSUM/KP :REFL=1-AVGR/AVG
750 IF REFL <= 0 THEN 700
760 GOTO 800
770 PRINT "Give the refractive index. ";:INPUT RINX
780 IF RINX < 1! THEN 770
790 REF=(1-RINX)/(1+RINX) :REFL=REF*REF
800 PRINT "Reflectivity = ";REFL
810 PRINT " "
820 PRINT "Now measure the transmittance during the exposure ... "
830 PRINT "******* ANSWER THE FOLLOWING *******"
840 PRINT "resist name for the measurement ";:INPUT RESIST$
850 PRINT "prebake temp. ['C] ";:INPUT BKTEMP
860 PRINT "         time [min] ";:INPUT BKTIME
870 PRINT " "
880 PRINT " "
890 PRINT "The following measurement is for ..."
900 PRINT "resist name: ";RESIST$
910 PRINT "prebake temp.";BKTEMP
920 PRINT "         time ";BKTIME
930 PRINT "Are these all correct [y/n] ";:INPUT PRC$
940 IF PRC$ <> "y" AND PRC$ <> "n" THEN 930
950 IF PRC$ = "n" THEN 830
960 PRINT " "
970 PRINT "******* Read this instruction *******"
980 PRINT "Each wafer is given a file number by the user: the"
990 PRINT "file number should be of the form mmddrr, where mm is"
1000 PRINT "month, dd is date, and rr is run number."
1010 PRINT "Example: on May 13 we do run 95 "
1020 PRINT "the file name would be given 051395."
1030 PRINT " "
1040 ERASE H,L,P          'erase the previous result
1050 DIM H(3000),L(3000),P(3000)   'have room to spare, check with "print fre(x)"

1060 PRINT "specify file number"; :INPUT A$
1070 IF A$="000000" THEN END
1080 B$="B:"
1090 D$="T.DAT"
1100 I$=B$+A$+D$
1110 PRINT "Enter the dose level [mJ/cm2] ";:INPUT DOSE
1120 PERIOD=1
1130 IF DOSE >= 900 THEN PERIOD=3 :PRINT "Three periods are given."
1140 PRINT " "
1150 DTMIN=.02# :DTMAX=60! :DTPLOT=.04# :T2MIN=1.8
1160 PRINT "minimum data collection rate : t1(sec) =";DTMAX
1170 PRINT "maximum data collection rate:"
1180 REM PRINT "without displaying data in real time"
1190 PRINT "First period          (without plot ) ... t1 >= ";DTMIN
1200 PRINT "                      (with    plot ) ... t1 >= ";DTPLOT
1210 PRINT "Second period and on (with/without) ... t1 >= ";T2MIN
1220 PRINT " "
```

```
1230 NTOT=0
1240 PRINT " "
1250 FOR I=1 TO PERIOD
1260 PRINT "In period ";I;:PRINT " how many sampling points."; : INPUT N(I)
1270 NTOT=NTOT+N(I)
1280 PRINT "time between sampling points (sec)"; : INPUT T1(I)
1290 IF T1(I)<DTMIN AND GFLAG$="n" THEN PRINT "miminum is ";DTMIN:GOTO 1280
1300 IF T1(I)<DTPLOT AND GFLAG$="y" THEN PRINT "miminum is ";DTPLOT:GOTO 1280
1310 IF T1(I) > DTMAX THEN PRINT "maximum is ";DTMAX : GOTO 1280
1320 IF I>=2 AND T1(I)<T2MIN THEN PRINT "T1>=1.8 for 2nd-period on":GOTO 1280
1330 REM fast clk range = .001 to 9.999 sec; slow clk range = .01 to 99.99 sec
1340 IF T1(I) <= 2 THEN FAST(I) = 1 ELSE FAST(I) = 0 'DECIDE TO USE FAST OR SLOW
 CLOCK
1350 IF FAST(I) = 1 THEN P = .001  ELSE P = .01 'P = PERIOD OF TIMER CLK PULSES
1360 TI = T1(I)/P       'ti=timer load value in integer form
1370 REM convert the timer load value from 16-bit binary to 16-bit BCD
1380 REM the timer load values are in BCD format, 0-9999 for 16-bit data
1390 TEMP = TI\100
1400 TLL%(I) = FNBCD(TI-TEMP*100)
1410 TLH%(I) = FNBCD(TEMP)
1420 NEXT I
1430 PWLIM=0 :D=AVI
1440 FOR I=1 TO PERIOD
1450 PWLIM=PWLIM+T1(I)*N(I)*D
1460 NEXT I
1470 IF DOSE >= PWLIM THEN PRINT "Give more points ":GOTO 1240
1480 T2=DOSE/D
1490 PRINT " "
1500 PRINT "Exposure Time is ";T2;:PRINT " sec."
1510 PRINT " "
1520 PRINT "Use stop watch and try to meet the exposure time  "
1530 PRINT " "
1540 PRINT " "
1550 PRINT " "
1560 PRINT "********************************"
1570 PRINT "This is the input you entered. "
1580 PRINT "********************************"
1590 PRINT "The data file name is ";A$
1600 PRINT "Dose Level =";DOSE
1610 FOR I=1 TO PERIOD
1620 PRINT "During period ";I
1630 PRINT "      # of sampling points  ";N(I)
1640 PRINT "      Time interval         ";T1(I)
1650 NEXT I
1660 PRINT "Exposure power        ";D
1670 PRINT "Exposure time [sec]   ";T2
1680 PRINT " "
1690 PRINT "Do you want to change any one above [y/n] ";:INPUT ABO$
1700 IF ABO$ <> "y" AND ABO$ <> "n" THEN 1690
1710 IF ABO$ = "y" THEN 830
1720 REM
1730 PRINT " "
1740 REM the following program segment has been optimized for Lab Master
1750 REM  1817 = adrs+9       adrs=1808
1760 REM  1812 = adrs+4
1770 REM  1813 = adrs+5
1780 REM  1814 = adrs+6
1790 IF GFLAG$="y" THEN GOTO 3270
1800 REM *** between the two marks the program has to run quick as a rabbit
1810 PRINT"hit any key during data collection to stop"
1820 PRINT "*** REMEBER : Exposure Time =";T2;:PRINT " seconds ***"
1830 PRINT "                : Substrate      =";A$
1840 ADRS=1808              'I/O starting address of Tecmar Lab Master Board
1850 ADLT=816               'I/O starting address of Tecmar Lab Tender Board
1860 OUT ADRS+9,255         'master reset the timer
1870 OUT ADRS+9,95          'load all counters
```

```
1880 SM= INP(ADRS+6)        'reset the status bit in Lab Master
1890 ST= INP(ADLT+1)        '       the status bit in Lab Tender
1900 OUT ADRS+4,128         'control byte of A/D (10000000) :Lab Master
1910 OUT ADRS+5,0           'A/D input channel number 0
1920 OUT ADLT,32            'A/D input channel LT
1930 OUT ADRS+9,23         'data pointer to master mode register
1940 OUT ADRS+8,0          'master mode reg l (0000 0 0 00)
1950 OUT ADRS+8,128        'master mode reg h (1 0 0 0 0000)
1960 OUT ADRS+9,5          'data pointer to counter mode reg of counter 5
1970 OUT ADRS+8,49         'counter mode register l (00110 001)
1980 IF FAST(1)=1 THEN CMRH%=14 ELSE CMRH%=15 '14=use f4(.001s),15=use f5(.01)
1990 OUT ADRS+8,CMRH%      'counter mode reg h (000 01110)
2000 OUT ADRS+8,TLL%(1)     'counter load value l      clv=x+6*int(x/10)
2010 OUT ADRS+8,TLH%(1)     'counter load value h
2020 PRINT "When ready type go  ";:INPUT G$
2030 IF G$ <> "go" AND G$ <> "GO" THEN 2020
2040 PRINT "Period ...  1"
2050 OUT 1812,132:OUT 1817,112:OUT 816,96
2060 FOR I=1 TO N(1)
2070  IF INP(1812)<128 OR INP(816)<128 THEN 2070
2080 P(I)=INP(817):L(I)=INP(1813):H(I)=INP(1814):IF INKEY$<>"" THEN 2110
2090 NEXT I
2100 GOTO 2120
2110 N(1)=I
2120 IF PERIOD = 1 THEN 2220          'only one period was specified
2130 NI=N(1)+1
2140 FOR IP=2 TO PERIOD
2150 PRINT "Period ... ";IP
2160     NF=NI+N(IP)-1
2170    GOSUB 3840
2180     NI=NF+1
2190 NEXT IP
2200 GOTO 2220
2210 REM *** the program can now leisurely walk as a turttle
2220 REM convert data from two's complement to floating point format
2230 PRINT "processing data..."
2240 NI=1
2250 FOR J=1 TO PERIOD
2260 NF=NI+N(J)-1
2270 FOR I=NI TO NF
2280  H(I)=256*H(I)+L(I)
2290  IF H(I)>32767 THEN H(I)=H(I)-65536!
2300  H(I)=H(I)/204.8
2310  P(I)=(P(I)-128)/128*5
2320 NEXT I
2330 NI=NF+1
2340 NEXT J
2350 PRINT "Hit any key to continue."
2360 IF INKEY$ = "" THEN 2360
2370 REM Plot the prosessed data
2380 CLS: KEY OFF : SCREEN 2
2390 GOSUB 4210
2400 GRID=0
2410 NI=1
2420 FOR J=1 TO PERIOD
2430 PRINT "Period ... ";J
2440 NF=NI+N(J)-1
2450 FOR I=NI TO NF
2460  PSET(I-(I\639)*639,198-H(I)*20)
2470  PSET(I-(I\639)*639,198-P(I)*20)
2480  IF I/639 = I\639 THEN GRID=1 :CLS
2490  IF GRID = 0 THEN 2520
2500  GOSUB 4210
2510  GRID = 0
2520 NEXT I
2530 NI=NF+1
```

```
2540 NEXT J
2550 REM
2560 PRINT "calculating  intensities ... "
2570 N(0)=0 :NI=1:PSUM=0 :KS=0 :TS=0
2580 FOR IP=1 TO PERIOD
2590 NF=NI+N(IP)-1
2600      FOR I=NI TO NF
2610          H(I)=H(I)/VTOMW :P(I)=P(I)*AVG
2620          IF P(I) >= PWCUT THEN TS=TS+T1(IP)
2630          IF P(I) >= P2*AVI OR P(I) =< P1*AVI THEN 2650
2640          KS=KS+1 :PSUM=PSUM+P(I)
2650      NEXT I
2660 NI=NF+1
2670 NEXT IP
2680 NDS=NF-KS
2690 PRINT "# of cut data = ";NDS
2700 PRINT "Do you want to check the processed data ";:INPUT CD$
2710 IF CD$ <> "y" AND CD$ <> "n" THEN 2700
2720 IF CD$ = "n" THEN 2870
2730 N(0)=0
2740 NI=1
2750 FOR IP=1 TO PERIOD
2760 PRINT "period ";IP
2770 NF=NI+N(IP)-1
2780 PRINT N(IP),T1(IP),D,T2,REFL
2790   FOR I=NI TO NF STEP 7
2800   PRINT USING "####.#####";P(I),P(I+1),P(I+2),P(I+3),P(I+4),P(I+5),P(I+6
2810   PRINT USING "####.#####";H(I),H(I+1),H(I+2),H(I+3),H(I+4),H(I+5),H(I+6
2820   PRINT " "
2830   NEXT I
2840 NI=NF+1
2850 NEXT IP
2860 REM
2870 REM calculate average intensity again during the exposure.
2880 AVI=PSUM/KS           'The average intensity at the resist surface (air)
2890 DEV=0 :JS=0
2900 FOR I=1 TO NF
2910      IF P(I) >= P2*AVI OR P(I) =< P1*AVI THEN 2930
2920      DEV=DEV+(P(I)-AVI)^2/AVI :JS=JS+1  'Standard deviation source power
2930 NEXT I
2940 SDEV=SQR(DEV)/JS*100       'for %
2950 D=AVI :T2=TS :DOS=D*T2
2960 PRINT " "
2970 PRINT "Average.Intensity at the Surface (Air) = ";D
2980 PRINT "Exposure Time (sec)                    = ";T2
2990 PRINT "Exposed Dose (air)   (mJ/cm2)          = ";DOS
3000 PRINT "Standard Deviation per Power  (%)      = ";SDEV
3010 PRINT " "
3020 PRINT "*** want to save it [y/n] ";:INPUT WT$
3030 IF WT$ <> "y" AND WT$ <> "n" THEN 3020
3040 IF WT$ = "n" THEN 880
3050 PRINT "saving data and reference at the resist surface(air) ... "
3060 OPEN I$ FOR OUTPUT AS #1
3070 PRINT #1,RESIST$
3080 PRINT #1,BKTEMP
3090 PRINT #1,BKTIME
3100 PRINT #1,USING "##";PERIOD
3110 NI=1
3120 FOR IP=1 TO PERIOD
3130 NF=NI+N(IP)-1
3140 PRINT #1,N(IP),T1(IP),D,T2,REFL
3150 PRINT #1, "source-channel"
3160 FOR I=NI TO NF    STEP 7
3170 PRINT #1,USING "####.#####";P(I),P(I+1),P(I+2),P(I+3),P(I+4),P(I+5),P(I+6
3180 NEXT I
3190 PRINT #1, "data-channel"
```

```
3200 FOR I=NI TO NF    STEP 7
3210 PRINT #1,USING "####.#####";H(I),H(I+1),H(I+2),H(I+3),H(I+4),H(I+5),H(I+6)
3220 NEXT I
3230 NI=NF+1
3240 NEXT IP
3250 CLOSE #1
3260 GOTO 880
3270 REM This routine is the data collection routine during which the collected
3280 REM data are displayed in real time: consequently, the rate of data
3290 REM collection is slow.  data rate = 1/.032 = 31
3300 REM Note: In order to save exeuction time, this segment of the code works
3310 REM     for positive voltage only ('If h>32767 then h-h-65536!' is skipped
3320 REM graphics information
3330 REM   (   0,  0)  (639,  0)   each coordinate pair = (x,y) or (col,row)
3340 REM   (   0,199)  (639,199)   the whole screen = 640x200
3350 CLS          'clear screen
3360 KEY OFF      'turn off the function-key display
3370 SCREEN 2     'high-resolution graphic screen (640x200)
3380 REM *** between the two marks the program has to run quick as a rabbit
3390 PRINT "    hit any key during data collection to stop    "
3400 PRINT "*** REMEBER : Exposure Time =";T2;:PRINT " seconds ***"
3410 PRINT "                : Substrate     =";A$
3420 GOSUB 4210
3430 REM Setting the boards
3440 ADRS=1808               'I/O starting address of Tecmar Lab Master Board
3450 ADLT=816                'I/O starting address of Tecmar Lab Tender Board
3460 OUT ADRS+9,255          'master reset the timer
3470 OUT ADRS+9,95           'load all counters
3480 SM= INP(ADRS+6)         'reset the status bit in Lab Master
3490 ST= INP(ADLT+1)         '      the status bit in Lab Tender
3500 OUT ADRS+4,128          'control byte of A/D (10000000) :Lab Master
3510 OUT ADRS+5,0            'A/D input channel number 0
3520 OUT ADLT,32             'A/D input channel LT
3530 OUT ADRS+9,23           'data pointer to master mode register
3540 OUT ADRS+8,0            'master mode reg l (0000 0 0 00)
3550 OUT ADRS+8,128          'master mode reg h (1 0 0 0 0000)
3560 OUT ADRS+9,5            'data pointer to counter mode reg of counter 5
3570 OUT ADRS+8,49           'counter mode register l (00110 001)
3580 IF FAST(1)=1 THEN CMRH%=14 ELSE CMRH%=15 '14=use f4(.001s),15=use f5(.01)
3590 OUT ADRS+8,CMRH%        'counter mode reg h (000 01110)
3600 OUT ADRS+8,TLL%(1)      'counter load value l     clv=x+6*int(x/10)
3610 OUT ADRS+8,TLH%(1)      'counter load value h
3620 PRINT "    When ready type go  ";:INPUT G$
3630 IF G$ <> "go" AND G$ <> "GO" THEN 3620
3640 PRINT "Period ...  1"
3650 OUT 1812,132:OUT 1817,112:OUT 816,96
3660 FOR I=1 TO N(1)
3670  IF INP(1812)<128 OR INP(816)<128 THEN 3670
3680 P(I)=(INP(817)-128)/128*5:H(I)=(256*INP(1814)+INP(1813))/204.8:PSET(I-(I\63
9)*639,198-H(I)*20):PSET(I-(I\639)*639,198-P(I)*20):IF INKEY$<>"" THEN 3720
3690 IF I/639 = I\639 THEN CLS
3700 NEXT I
3710 GOTO 3730
3720 N(1)=I
3730 IF PERIOD = 1 THEN 2560          'only one period was specified
3740 NI=N(1)+1
3750 FOR IP=2 TO PERIOD
3760 PRINT "Period ... ";IP
3770 NF=NI+N(IP)-1
3780    GOSUB 3840
3790 NI=NF+1
3800 NEXT IP
3810 GOSUB 4210
3820 GOTO 2560
3830 REM Subroutine Set-Up for data collection
3840 ADRS=1808               'I/O starting address of Tecmar Lab Master Board
```

```
3850 ADLT=816                  'I/O starting address of Tecmar Lab Tender Board
3860 OUT ADRS+9,255            'master reset the timer
3870 OUT ADRS+9,95             'load all counters
3880 SM= INP(ADRS+6)           'reset the status bit in Lab Master
3890 ST= INP(ADLT+1)           '       the status bit in Lab Tender
3900 OUT ADRS+4,128            'control byte of A/D (10000000) :Lab Master
3910 OUT ADRS+5,0              'A/D input channel number 0
3920 OUT ADLT,32               'A/D input channel LT
3930 OUT ADRS+9,23            'data pointer to master mode register
3940 OUT ADRS+8,0             'master mode reg l (0000 0 0 00)
3950 OUT ADRS+8,128           'master mode reg h (1 0 0 0 0000)
3960 OUT ADRS+9,5             'data pointer to counter mode reg of counter 5
3970 OUT ADRS+8,49            'counter mode register l (00110 001)
3980 IF FAST(IP)=1 THEN CMRH%=14 ELSE CMRH%=15 '14=use f4(.001s),15=use f5(.01)
3990 OUT ADRS+8,CMRH%         'counter mode reg h (000 01110)
4000 OUT ADRS+8,TLL%(IP)        'counter load value l      clv=x+6*int(x/10)
4010 OUT ADRS+8,TLH%(IP)      'counter load value h
4020 IF GFLAG$ = "n" THEN 4120
4030 OUT 1812,132:OUT 1817,112:OUT 816,96
4040 FOR I=NI TO NF
4050  IF INP(1812)<128 OR INP(816)<128 THEN 4050
4060 P(I)=(INP(817)-128)/128*5:H(I)=(256*INP(1814)+INP(1813))/204.8:PSET(I-(I\63
9)*639,198-H(I)*20):PSET(I-(I\639)*639,198-P(I)*20):IF INKEY$<>"" THEN 4100
4070 IF I/639 = I\639 THEN CLS
4080 NEXT I
4090 RETURN
4100 N(IP)=I-NI+1
4110 RETURN
4120 OUT 1812,132:OUT 1817,112:OUT 816,96
4130 FOR I=NI TO NF
4140 IF INP(1812)<128 OR INP(816)<128 THEN 4140
4150 PRINT INP(816);INP(1812)
4160 P(I)=INP(817):L(I)=INP(1813):H(I)=INP(1814):IF INKEY$<>"" THEN 4190
4170 NEXT I
4180 RETURN
4190 N(IP)=I-NI+1
4200 RETURN
4210 REM Subroutine GRID
4220 MG=1 :MGY=1
4230 XTIC=10 :YTIC=.5
4240 NGRID=XTIC*MG :LRESET=10*NGRID
4250 LTEN=-1 :LXTEN=1
4260 FOR L=1 TO 631 STEP NGRID
4270 PSET(L,198)
4280 LTEN=LTEN+1
4290 IF LTEN=LXTEN*10 THEN LXTEN=LXTEN+1 :PSET(L,197)
4300 NEXT L
4310 NYSTP=YTIC*20*MGY :NYRESET=2*NYSTP
4320 FOR LY=0 TO 198 STEP NYSTP
4330 PSET(1,198-LY)
4340 IF LY/NYRESET = LY\NYRESET THEN PSET(2,198-LY)
4350 IF LY/5/NYRESET = LY\(5*NYRESET) THEN PSET(3,198-LY) :PSET(4,198-LY)
4360 NEXT LY
4370 RETURN
4380 REM subroutine voltage output
4390 ERASE H, P
4400 DIM H(3000), P(3000)
4410 ADRS=1808                 'I/O starting address of Tecmar Lab Master Board
4420 ADLT=816                  'I/O starting address of Tecmar Lab Tender Board
4430 OUT ADRS+9,255            'master reset the timer
4440 OUT ADRS+9,95             'load all counters
4450 SM= INP(ADRS+6)           'reset the status bit in Lab Master
4460 ST= INP(ADLT+1)           '       the status bit in Lab Tender
4470 OUT ADRS+4,128            'control byte of A/D (10000000) :Lab Master
4480 OUT ADRS+5,0              'A/D input channel number 0
4490 OUT ADLT,32               'A/D input channel LT
```

```
4500 OUT ADRS+9,23        'data pointer to master mode register
4510 OUT ADRS+8,0         'master mode reg l (0000 0 0 00)
4520 OUT ADRS+8,128       'master mode reg h (1 0 0 0 0000)
4530 OUT ADRS+9,5         'data pointer to counter mode reg of counter 5
4540 OUT ADRS+8,49        'counter mode register l (00110 001)
4550 TA=.75 :PA=.001 :CMRH%=14
4560 TTA=TA/PA :TEMPA=TTA\100
4570 TLL%=FNBCD(TTA-TEMPA*100) :TLH%=FNBCD(TEMPA)
4580 OUT ADRS+8, CMRH%
4590 OUT ADRS+8, TLL%
4600 OUT ADRS+8, TLH%
4610 WRN=0    ' chech power limits for safety
4620 IF GFLAG$="y" THEN 4740
4630 PRINT "      V1         VO         V1/VO "
4640 OUT 1817,112 :OUT 1812, 132 :OUT 816,96
4650 IF INP(1812)<128 OR INP(816)<128 THEN 4650
4660 H=256*INP(1814)+INP(1813) :P=(INP(817)-128)/128*5
4670 IF H>32767 THEN H=H-65536!
4680 H=H/204.8
4690 PRINT USING "####.#####";H,P,H/(P+.000001)
4700 IF H < HL OR H > HH THEN WRN=1
4710 IF P < PL OR P > PH THEN WRN=1
4720 IF INKEY$ <> "" THEN 4930
4730 GOTO 4650
4740 REM plot routine
4750 CLS :KEY OFF :SCREEN 2
4760 PRINT "DATA IS TAKEN AT EVERY ";TA ;:PRINT " SEC. "
4770 GOSUB 4210
4780 I=0 :K=0 :HSUM=0 :RSUM=0
4790 OUT 1817,112 :OUT 1812, 132 :OUT 816,96
4800 IF INP(1812)<128 OR INP(816)<128 THEN 4800
4810 I=I+1 :K=K+1
4820 H=256*INP(1814)+INP(1813) :P=(INP(817)-128)/128*5
4830 IF H>32767 THEN H=H-65536!
4840 H=H/204.8
4850 PSET(I,198-H*20):PSET(I,198-P*20)
4860 IF H < HL OR H > HH THEN WRN=1
4870 IF P < PL OR P > PH THEN WRN=1
4880 H(K)=H/VTOMW
4890 HSUM=HSUM+H(K)   :RSUM=RSUM+H(K)/P
4900 IF I/639=I\639 THEN I=0 :CLS
4910 IF K => KP THEN 4930
4920 GOTO 4800
4930 RETURN
```

```
10 REM Program Name: ABCMX.BAS by Deok J. Kim   version: 2.5    Feb. 20  '83
20 REM Function: Calculate BEST A,B,C and M with X from EXPOSE data
30 NDX=10            'The resist thickness is divided by NDX
40 JSTEP=1 :NEACH=100  'skip the data points by JSTEP for data more than Neach
50 BEEP
60 DIM CHO(3000),CH1(3000),TRV(3000)
70 PRINT "Which diskette is your data on [a/b] "::INPUT DISK$
80 IF DISK$ <> "a" AND DISK$ <> "b"  THEN 70
90 FILEEXTENTION$ = "dat"
100 PRINT "do you want to see the data file on disk  [y/n]"; : INPUT ANSWER$
110 IF ANSWER$ <> "y" AND ANSWER$ <> "n" THEN PRINT "answer y for yes and n for
no" : GOTO 100
120 IF ANSWER$ = "y" AND DISK$ = "a" THEN FILES "a:*.dat"
130 IF ANSWER$ = "y" AND DISK$ = "b" THEN FILES "b:*.dat"
140 PRINT "input the name(number only) of your data file"; : INPUT FILENAME$
150 FILET$ = DISK$+":"+FILENAME$+"T" + "." + FILEEXTENTION$
160 OPEN FILET$ FOR INPUT AS #1
170 INPUT #1,RESIST$
180 INPUT #1,BKTEMP
190 INPUT #1,BKTIME
200 INPUT #1, IPERIOD
210 IB=0
220 IP=0
230 IP=IP+1
240 IF IP > IPERIOD THEN GOTO 370
250 INPUT #1, N(IP),T1(IP),P(IP),T2(IP),REFL
260 IA=1+IB
270 IB=IB+N(IP)
280 INPUT #1, SO$(IP)
290 FOR I=IA TO IB STEP 7
300       INPUT #1,CHO(I),CHO(I+1),CHO(I+2),CHO(I+3),CHO(I+4),CHO(I+5),CHO(I+6)
310 NEXT I
320 INPUT #1, S1$(IP)
330 FOR I=IA TO IB STEP 7
340       INPUT #1,CH1(I),CH1(I+1),CH1(I+2),CH1(I+3),CH1(I+4),CH1(I+5),CH1(I+6)
350 NEXT I
360 GOTO 230
370 CLOSE #1
380 N=IB
390 PRINT " "
400 REM
410 RG=1-REFL :PWCUT=.1 '(mW/cm2)
420 REM Check if the source was taken from CHANNEL LT
430 NCH=0 :ICUT=0
440 FOR I=1 TO N
450 IF CHO(I) < PWCUT THEN ICUT=ICUT+1 :GOTO 470
460 TRV(I)=CH1(I)/CHO(I)/RG
470 NEXT I
480 PRINT " "
490 PRINT " "
500 PRINT "Data file   ";:PRINT FILET$ ;:PRINT "  is opened."
510 PRINT " "
520 PRINT "Exposure power was cut below ";PWCUT ;:PRINT " [mw/cm2]"
530 PRINT "           # of cut data is    ";ICUT
540 PRINT " "
550 REM PRINT "Do you want to see the data [y/n]"::INPUT SEE$
560 SEE$="n"
570 IF SEE$ = "n" THEN GOTO 810
```

```
580 REM BEGIN of ECHOING INPUT
590 PRINT " "
600 PRINT "Current Input File ";:PRINT FILET$
610 PRINT " "
620 PRINT RESIST$
630 PRINT "prebake ";BKTEMP,BKTIME
640 PRINT USING "##";IPERIOD
650 IB=0
660 IP=0
670 IP=IP+1
680 IF IP > IPERIOD THEN GOTO 810
690 PRINT N(IP),T1(IP),P(IP),T2(IP),REFL
700 IA=1+IB
710 IB=IB+N(IP)
720 PRINT S0$(IP)
730 FOR I=IA TO IB STEP 7
740       PRINT USING "####.#####";CH0(I),CH0(I+1),CH0(I+2),CH0(I+3),CH0(I+4),CH0
(I+5),CH0(I+6)
750 NEXT I
760 PRINT S1$(IP)
770 FOR I=IA TO IB STEP 7
780       PRINT USING "####.#####";CH1(I),CH1(I+1),CH1(I+2),CH1(I+3),CH1(I+4),CH1
(I+5),CH1(I+6)
790 NEXT I
800 GOTO 670
810 REM END of ECHOING INPUT
820 REM THIS IS THE PROGRAM TO CALCULATE A,B,C PARAMETERS
830 PRINT "Exposure DOSE  calculation is next ... "
840 NI = 1
850 IF CH0(NI) < PWCUT THEN NI=NI+1 :BEEP : GOTO 850
860 NF = N
870 IF CH0(NF) < PWCUT THEN NF=NF-1 :BEEP : GOTO 870
880 TI=CH1(NI)/(CH0(NI)*RG)
890 IF TI > 1 THEN PRINT "TI > 1 at ";NI :NI=NI+1 :GOTO 880
900 IF TI <= 0 THEN PRINT "TI <= 0 at ";NI :NI=NI+1 :GOTO 880
910 TIMEI=T1(1)*NI
920 TIMEF=0
930 DOS=0 :SUMI0=0 :INSUM=0
940 IB=0
950 IP=0
960 IP=IP+1
970 IF IP > IPERIOD GOTO 1080
980 IA=1+IB
990 IB=IB+N(IP)
1000 FOR I=IA TO IB
1010 IF CH0(I) < PWCUT THEN BEEP :GOTO 1060
1020 TIMEF=TIMEF+T1(IP)
1030 PW0=CH0(I)
1040 DOS=DOS+PW0*T1(IP) :SUMI0=SUMI0+PW0
1050 INSUM=INSUM+1
1060 NEXT I
1070 GOTO 960
1080 NF1=NF
1090 AVGI0=SUMI0/INSUM
1100 PRINT "Average Intensity (air) = ";AVGI0
1110 IF CH0(NF1) < PWCUT THEN NF1=NF1-1 :BEEP :GOTO 1110
1120 TF =CH1(NF1)/(CH0(NF1)*RG)
1130 IF TF > 1 THEN PRINT "TF > 1 at ";NF1 :NF1=NF1-1 :GOTO 1110
1140 IF TF <= 0 THEN PRINT "TF <= 0 at ";NF1 :NF1=NF1-1 :GOTO 1110
1150 TI=CH1(NI)/(CH0(NI)*RG)
1160 NF=NF1 :NFINAL=NF1
1170 PRINT " "
1180 PRINT "++++++++++++++++++++++++++++++++++++++++++++++++++"
1190 PRINT " "
1200 PRINT "Exposure starts at ";:PRINT TIMEI;:PRINT "sec"
1210 PRINT "Transmittance is  ";:PRINT TI
```

```
1220 PRINT ""
1230 PRINT "Exposure ends   at ";:PRINT TIMEF;:PRINT "sec"
1240 PRINT "Transmittance is  ";:PRINT TF
1250 PRINT ""
1260 PRINT "Exposed Doses in Air    ";:PRINT DOS ;:PRINT " [mJ/cm2]"
1270 PRINT "Exposed Doses from Data ";:PRINT P(1)*T2(1);:PRINT " [mJ/cm2]"
1280 PRINT ""
1290 PRINT "+++++++++++++++++++++++++++++++++++++++++++++"
1300 PRINT " "
1310 REM get thickness of resist from RX file
1320 DEPTH$=DISK$ + ":" + FILENAME$ + "RX" + ".DAT"
1330 PRINT "Finding thickness from ";DEPTH$
1340 OPEN DEPTH$ FOR INPUT AS #1
1350 INPUT #1, RESIST$
1360 INPUT #1, DEVELOPER$
1370 INPUT #1, DEVTEMP
1380 INPUT #1, THICK
1390 CLOSE #1
1400 PRINT " "
1410 PRINT RESIST$
1420 PRINT DEVELOPER$
1430 PRINT DEVTEMP
1440 PRINT USING "####.#####";THICK
1450 PRINT "----------------------------------------"
1460 PRINT "The thickness of the resist is ";THICK
1470 PRINT "----------------------------------------"
1480 PRINT " "
1490 REM ESTIMATE A,B,C
1500 REM
1510 REM
1520 REM CALCULATE THE SLOPE TO ESTIMATE C PARAMETER
1530 REM BASED ON STRAIGHT LINE FITTING y=T0+S*t
1540 PRINT "Plot of transmittance will be next. Hit any key. "
1550 IF INKEY$ = "" THEN 1550
1560 GOSUB 4080
1570 PRINT "How many data from the beginning to estimate the slope ";:INPUT NS
1580 MVAL=1
1590 FULDOS=1000
1600 IF DOS >=FULDOS THEN MVAL=0 : GOTO 1660
1610 PRINT "Dose < " ;FULDOS ;:PRINT "Give B value ";:INPUT BE
1620 PRINT "How many points at the end would give the last transmittance"
1630 PRINT "If you don't remember the last picture then type back ";:INPUT BAK$
1640 IF BAK$ <> "back" THEN 1660
1650 GOSUB 4080
1660 PRINT "How many data from the end for the last transmittance ";:INPUT NAV
1670 BEEP
1680 NF1=NF :NF5=NF-NAV+1
1690 NCUT=0 :TFSUM=0
1700 IF CH0(NF1) < PWCUT THEN NCUT=NCUT+1 :BEEP :GOTO 1730
1710 TF1=CH1(NF1)/(CH0(NF1)*RG)
1720 TFSUM=TFSUM+TF1
1730 NF1=NF1-1
1740 IF NF1 < NF5 THEN GOTO 1760
1750 GOTO 1700
1760 NAVE=NF-NF1-NCUT
1770 TF=TFSUM/NAVE
1780 TFM=TF
1790 IF MVAL=1 THEN 1810
1800 BE=-(1/THICK)*LOG(TF)
1810 REM Biginning of estimatiom
1820 BEEP
1830 X=0
1840 Y=0
1850 XY=0
1860 XX=0
1870 YY=0
1880 V0=0
```

```
1890 NIS=NI+NS-1
1900 NDEAD=0
1910 FOR I=NI TO NIS
1920 IF CHO(I) < PWCUT  THEN NDEAD=NDEAD+1 :BEEP : GOTO 2010
1930 YI=CH1(I)/(CHO(I)*RG)
1940 XI=T1(1)*(I-NI+1-NDEAD)
1950 Y =Y+YI
1960 X=X+XI
1970 XY=XY+XI*YI
1980 XX=XX+XI*XI
1990 YY=YY+YI*YI
2000 VO=VO+CHO(I)
2010 NEXT I
2020 BEEP
2030 NS=NS-NDEAD
2040 TO=(Y*XX-X*XY)/(NS*XX-X*X)
2050 S =(NS*XY-X*Y)/(NS*XX-X*X)
2060 IO=RG*VO/NS        'intensity on the resist surface
2070 PRINT " "
2080 PRINT "Estimated Slope =";:PRINT S
2090 PRINT "Extraplated  TO =";:PRINT TO
2100 PRINT " "
2110 TF=EXP(-BE*THICK)
2120 AE=(LOG(TF)-LOG(TO))/THICK
2130 CE =(AE+BE)*S/(IO*AE*TO*(1-TO))
2140 IF MVAL=1 THEN ME=-(LOG(TFM)/THICK+BE)/AE
2150 PRINT "Estimated A Parameter ="; :PRINT AE :AEO=AE
2160 PRINT "Estimated B Parameter ="; :PRINT BE :BEO=BE
2170 PRINT "Estimated C Parameter ="; :PRINT CE :CEO=CE
2180 PRINT "Estimated M Parameter ="; :PRINT ME :MEO=ME
2190 PRINT " "
2200 IF MVAL=1 THEN PRINT "B was given from dose > ";FULDOS :GOTO 2230
2210 PRINT "B is based on the last  "; :PRINT NAVE;:PRINT " points."
2220 PRINT "# of dead points are    "; :PRINT NCUT
2230 PRINT "C is based on the first "; :PRINT NS;:PRINT " points."
2240 PRINT "# of dead points are    "; :PRINT NDEAD
2250 REM END OF A,B,C. ESTIMATION
2260 TI=CH1(NI)/(CHO(NI)*RG)
2270 RT=TO/TI
2280 PRINT " "
2290 PRINT "TO/TI =";:PRINT RT
2300 PRINT " "
2310 SLOP=1
2320 PRINT "Do you want to plot the slope in the first section [y/n] ";:INPUT SL
$
2330 IF SL$ <> "y" AND SL$ <> "n" THEN 2320
2340 IF SL$ = "y" THEN GOSUB 4080
2350 SLOP=0
2360 LAST=1
2370 PRINT "Do you want to plot the average TR in the last section [y/n] ";:INPU
T LAST$
2380 IF LAST$ <> "y" AND LAST$ <> "n" THEN 2370
2390 IF LAST$ = "y" THEN GOSUB 4080
2400 LAST=0
2410 PRINT "Do you want to re-estimate A,B,C [y/n] ";:INPUT RE$
2420 IF RE$ <> "y" AND RE$ <> "n" THEN 2410
2430 IF RE$ ="y" THEN 1540
2440 REM
2450 PRINT "=================================="
2460 PRINT "Precision of A,B,C starts now ... "
2470 ERASE CH1
2480 DIM TRO(3000),IR(100),M(100),DEPTH(100)
2490 PRINT "Do you already know closer A,B,C [y/n] ";:INPUT ABCH$
2500 IF ABCH$ <> "y" AND ABCH$ <> "n" THEN 2490
2510 IF ABCH$ = "n" THEN 2550
```

```
2520 PRINT "Give A,B,C ":INPUT AE,BE,CE
2530 IF BE < BEO THEN MVAL=1
2540 IF MVAL=1 THEN MPAR=-(LOG(TFM)/THICK+BE)/AE :PRINT "M = ";MPAR
2550 PRINT " "
2560 A=AE :B=BE :C=CE
2570 PRINT A,B,C
2580 PRINT "The thickness division is given ";NDX
2590 PRINT " "
2600 FOR IP=1 TO IPERIOD
2610 PRINT "PERIOD  ";IP
2620 JSTEP(IP)=JSTEP
2630 PRINT " # of data = ";N(IP);:PRINT "  Current JSTEP = ";JSTEP
2640 IF (NF-NI) < 100 THEN 2670
2650 IF N(IP) > NEACH THEN PRINT "Change  JSTEP ";:INPUT JSTEP(IP)
2660 IF JSTEP(IP) < 1 THEN PRINT "... JSTEP < 1 ... WRONG !!! " :GOTO 2610
2670 NEXT IP
2680 GOSUB 3500
2690 IF MSD < .0001 THEN MSDP=MSD :GOTO 3440
2700 IF MSD > .002 THEN PRINT "ESTIMATE AGAIN ! ":GOTO 2520
2710 PC=SQR(MSD)
2720 PA=PC :PB=-PC
2730 REM Refinement of C,A,and B parameters
2740 MSDP=MSD :MSDRDS=MSD
2750 FOR IROUND=1 TO 3
2760 PRINT "===== ROUND ";IROUND ;:PRINT " ====="
2770 REDU=0
2780 SET=0
2790 C=CE*(1+PC)
2800 PRINT "Previous MSD  = ";MSDP ;:PRINT "  ... process C now "
2810 GOSUB 3500
2820 IF MSD > MSDP THEN 2900
2830 SET=0 :REDU=0
2840 CE=C :MSDP=MSD :PC=SQR(MSD)*PC/ABS(PC)
2850 IF MSD=< .0001 THEN 3440
2860 IF MSD > MSDP*.997 THEN OUTLOOP=1 ELSE OUTLOOP=0
2870 IF OUTLOOP=1 THEN 2940
2880 IF MSD < .0005/IROUND THEN 2940
2890 GOTO 2790
2900 IF SET=0 THEN PC=-PC :SET=1 :GOTO 2790
2910 IF REDU=1 THEN C=CE :GOTO 2940
2920 PC=PC/2 :REDU=REDU+1
2930 GOTO 2790
2940 REDUA=0
2950 SET=0
2960 A=AE*(1+PA)
2970 PRINT "Previous MSD  = ";MSDP ;:PRINT "  ... process A now "
2980 GOSUB 3500
2990 IF MSD > MSDP THEN 3070
3000 SET=0 :REDUA=0
3010 AE=A :MSDP=MSD :PA=SQR(MSD)*PA/ABS(PA)
3020 IF MSD <= .0001 THEN 3440
3030 IF MSD > MSDP*.997 THEN OUTLOOP=1 ELSE OUTLOOP=0
3040 IF OUTLOOP=1 THEN 3110
3050 IF MSD < .0005/IROUND THEN 3110
3060 GOTO 2960
3070 IF SET=0 THEN PA=-PA :SET=1 :GOTO 2960
3080 IF REDUA=1 THEN A=AE :GOTO 3110
3090 PA=PA/2 :REDUA=REDUA+1
3100 GOTO 2960
3110 REDUB=0
3120 SET=0
3130 B=BE*(1+PB)
3140 PRINT "Previous MSD  = ";MSDP ;:PRINT "  ... process B now "
3150 GOSUB 3500
3160 IF MSD > MSDP THEN 3240
3170 SET=0 :REDUB=0
```

```
3180 BE=B :MSDP=MSD :PB=SQR(MSD)*PB/ABS(PB)
3190 IF MSD <= .0001 THEN 3440
3200 IF MSD > MSDP*.997 THEN OUTLOOP=1 ELSE OUTLOOP=0
3210 IF OUTLOOP=1 THEN 3280
3220 IF MSD < .0005/IROUND THEN 3280
3230 GOTO 3130
3240 IF SET=0 THEN PB=-PB :SET=1 :GOTO 3130
3250 IF REDUB=1 THEN B=BE :GOTO 3280
3260 PB=PB/2 :REDUB=REDUB+1
3270 GOTO 3130
3280 NEXT IROUND
3290 PRINT "A = ";A ;:PRINT "B = ";B ;:PRINT "C = ";C ;:PRINT "MSD = ";MSDP
3300 PRINT "PC =";PC ;:PRINT "PA =";PA ;:PRINT "PB = ";PB
3310 IF MSDP > MSDRDS*.997 THEN PRINT "NO CONVERGE !!!":GOTO 3440
3320 PRINT " "
3330 PRINT "Will you continue [y/n]"
3340 GOSUB 4830
3350 IF V$ <> "n" AND V$ <> "y" THEN 2730
3360 IF V$ = "n" THEN 3440
3370 PRINT "GIVE A NEW PC [smaller than old PC] ";:INPUT PC
3380 PRINT "          NEW PA [smaller than old PA] ";:INPUT PA
3390 PRINT "          NEW PB [smaller than old PB] ";:INPUT PB
3400 IF PA=0 AND PC=0 THEN 3440
3410 IF PA=0 AND PB=0 THEN 3440
3420 IF PC=0 AND PB=0 THEN 3440
3430 GOTO 2730
3440 PRINT "FINAL  A = ";A ;:PRINT " B = ";B ;:PRINT " C = ";C
3450 IF BE < BEO THEN MVAL=1
3460 IF MVAL=1 THEN MPAR=-(LOG(TFM)/THICK+B)/A :PRINT "M = ";MPAR
3470 PRINT "FINAL MSD = ";MSDP
3480 GOTO 3850
3490 REM Calculation of MSD
3500 NED=0:SDEV=0:XDEPTH=0
3510 IK=0:XDEPTH=0:DEPTH(0)=0 :M(0)=1
3520 DIX=THICK/NDX
3530       FOR JX=1 TO NDX
3540       XDEPTH=XDEPTH+DIX
3550       DEPTH(JX)=XDEPTH
3560       M(JX)=1
3570       NEXT JX
3580 JSUM=0
3590 FOR IP=1 TO IPERIOD
3600 JSTEP=JSTEP(IP)
3610 PRINT " "
3620 PRINT "Process is under way in period ";IP;:PRINT " out of total ";IPERIOD
3630   NED=NED+N(IP) :NST=NED-N(IP)+1
3640   IF NED > NFINAL THEN NED=NFINAL
3650   DTIME=T1(IP)*JSTEP
3660   FOR J=NST TO NED STEP JSTEP
3670       IF CHO(J) < PWCUT THEN 3800
3680       JSUM=JSUM+1
3690       IR(0)=RG*CHO(J)
3700       IF (J+JSTEP) >= NED THEN DTIME=T1(IP)*(NED-J+1)
3710       FOR L=1 TO NDX
3720         IR(L)=IR(L-1)-IR(L-1)*(A*M(L-1)+B)*DIX
3730         M(L-1)=M(L-1)-M(L-1)*C*IR(L-1)*DTIME
3740       NEXT L
3750     M(NDX)=M(NDX)-M(NDX)*C*IR(NDX)*DTIME
3760     TRO(J)=IR(NDX)/IR(0)
3770     DELTA=ABS(TRO(J)-TRV(J))
3780     SDEV=SDEV+DELTA*DELTA
3790     BEEP
3800   NEXT J
3810 NEXT IP
3820 MSD = SDEV/JSUM
3830 PRINT "*** Mean Square Deviation  = ";MSD
3840 RETURN
```

```
3850 PRINT "Plot for calculated and experimental TR is next. Hit any key."
3860 IF INKEY$ = "" THEN 3860
3870 PREC=1
3880 GOSUB 4080
3890 PREC=0
3900 PRINT " "
3910 PRINT "Do you want to save it [y/n] ";:INPUT MX$
3920 IF MX$ <> "y" AND MX$ <> "n" THEN 3910
3930 IF MX$ = "n" THEN ERASE TRO,IR,M,DEPTH :GOTO 2480
3940 PRINT "Which disk do you want it to be saved  [a/b] ";:INPUT DISK$
3950 FILEMX$ = DISK$ + ":" + FILENAME$ +"mx" +".dat"
3960 OPEN FILEMX$ FOR OUTPUT AS #1
3970 PRINT #1,RESISTS$
3980 PRINT #1,USING "####.#####";BKTEMP
3990 PRINT #1,USING "####.#####";BKTIME
4000 PRINT #1,USING "####.#####";A,B,C
4010 PRINT #1,USING "####.#####";(NDX+1),THICK,DOS,REFL
4020 FOR I=0 TO NDX
4030   PRINT #1,USING "####.#####";M(I),DEPTH(I)
4040 NEXT I
4050 CLOSE #1
4060 END
4070 PRINT "GO GET THE REQUIRED DATA !!!";:END
4080 REM Plot Subroutine
4090 CLS :KEY OFF :SCREEN 2
4100 BEEP
4110 MG=1
4120 MGY=1
4130 PRINT "FILE NAME =";FILENAME$ ;:PRINT "    Io =";AVGIO;:PRINT "[mW/cm2]";
4140 PRINT "  DOSE =";DOS ;:PRINT "[mJ/cm2]"
4150 NSRT=NI:NEND=NF
4160 NSUM=0
4170 FOR IP=1 TO IPERIOD
4180 NSUM =NSUM +N(IP)
4190 PRINT "The end of period ";IP ;:PRINT "  "; NSUM;:PRINT "  dt = ";T1(IP)
4200 NEXT IP
4210 PRINT "Plot Range X1,X2 ";:INPUT NSRT,NEND
4220 PRINT "Magnification X,Y ";:INPUT MG,MGY
4230 NMGSET=0
4240 IF MG < 0 THEN NMGSET=1 :MGA=ABS(MG)
4250 MK=1
4260 IF MG <= 0 THEN MG=1: MK=MGA :NN=INT((NEND-NSRT)/MGA+1) :GOTO 4280
4270 NN=MG*(NEND-NSRT)+1
4280 K=1
4290 GOSUB 4940
4300 IF PREC=1 THEN PRINT "A,B,C,MSD ";:PRINT USING "  #.####^^^^  ";A,B,C,MSD
4310 FOR I=1 TO NN STEP MG
4320     Y=MGY*TRV(K+NSRT-1)
4330     IF PREC = 1 THEN Y1=MGY*TRO(K+NSRT-1)
4340     PSET(I-(I\639)*639,199-Y*20)
4350     IF PREC = 1 THEN PSET(I-(I\639)*639,199-Y1*20)
4360     IF I/639 = I\639 THEN 4770
4370 K=K+MK
4380 NEXT I
4390 IF SLOP <> 1 THEN 4470
4400 K=1
4410 FOR I=1 TO NN STEP MG
4420   IF I < NI THEN 4460
4430   Y=(T0+S*T1(1)*(K+NSRT-1-NI))*MGY
4440   PSET(I-(I\639)*639,199-Y*20)
4450   K=K+MK
4460 NEXT I
4470 IF LAST <> 1 THEN 4540
4480 K=1
4490 FOR I= 1 TO NN STEP MG
```

```
4500   Y=MGY*EXP(-BE*THICK)
4510   PSET(I-(I\639)*639,199-Y*20)
4520   K=K+MK
4530 NEXT I
4540 IF NEND > N GOTO 4820
4550 PRINT "Do you want to check the data in the plot [y/n] ";:INPUT PR$
4560 IF PR$ <> "y" AND PR$ <> "n" THEN 4550
4570 IF PR$ = "n" THEN 4730
4580 PRINT "Beginning number ";NSRT
4590 NQ=INT((NEND-NSRT+1)/7)
4600 NEE=7*NQ+NSRT-1
4610 NR=NEND-NEE
4620 FOR I=NSRT TO NEE STEP 7
4630 PRINT USING "####.#####";TRV(I),TRV(I+1),TRV(I+2),TRV(I+3),TRV(I+4),TRV(I+5
),TRV(I+6)
4640 NEXT I
4650 IF NR=0 THEN 4720
4660 I=0
4670 I=I+1
4680 PRINT USING "####.#####";TRV(NEE+I);
4690 IF I >= NR THEN 4710
4700 GOTO 4670
4710 PRINT "&&&&&&&&&&"
4720 PRINT "Ending    number ";NEND
4730 PRINT "Do you want to see another section [y/n] ";:INPUT SEE$
4740 IF SEE$ <> "y" AND SEE$ <> "n" THEN 4730
4750 IF SEE$="y" THEN CLS : GOTO 4130
4760 IF SEE$ = "n" THEN 4820
4770 PRINT "Do you want to plot further [y/n] ";:INPUT YE$
4780 IF YE$ <> "y" AND YE$ <> "n" THEN 4770
4790 IF YE$ = "y" THEN CLS
4800 GOSUB 4940
4810 GOTO 4380
4820 RETURN
4830 REM AUTO CONTINUE
4840 V$="0":NTON=440/7 :RO=0
4850 M=440
4860 IF RO>10 THEN 4930
4870 SOUND M,.7
4880 V$=INKEY$
4890 IF V$="y" THEN GOTO 4930
4900 IF V$="n" THEN GOTO 4930
4910 IF M>1320 THEN RO=RO+1:M=M-RO*88 :GOTO 4860
4920 M=M+NTON :GOTO 4870
4930 RETURN
4940 REM subroutine grid
4950 SCALE=MG :XTIC=10 :YTIC=.1
4960 IF NMSET=1 THEN SCALE=1/MGA
4970 NGRID=XTIC*SCALE :LRESET=10*NGRID
4980 LTEN=-1 :LXTEN=1
4990 FOR L=1 TO 631 STEP NGRID
5000 PSET(L,199)
5010 LTEN=LTEN+1
5020 IF LTEN=LXTEN*10 THEN LXTEN=LXTEN+1 :PSET(L,198)
5030 NEXT L
5040 NYSTP=YTIC*20*MGY :NYRESET=5*NYSTP
5050 FOR LY=0 TO 199 STEP NYSTP
5060 PSET(1,199-LY)
5070 IF LY/NYRESET=LY\NYRESET THEN PSET(2,199-LY)
5080 IF LY/2/NYRESET=LY\(2*NYRESET) THEN PSET(3,199-LY):PSET(4,199-LY)
5090 NEXT LY
5100 RETURN
```

Appendix B

The BASIC Program for Development

```
10 REM Program Name: DEVELOP.BAS by Deok J. Kim  version: 2.1   Dec 20 '82
20 REM Purpose: to collect data by controlling the Tecmar A/D converter
30 REM maximum data rate = 50  for the  safety
40 REM maximum data rate = 25  with graph in real time
50 REM
60 DIM H(6000),L(6000)    'still have room to spare, check with "print fre(x)"
70 DEF FNBCD(X) = X+6*(X\10)     'convert an integer from binary to BCD form, thi
s function is only good for 8 bits data thus 99 is the maximum value
80 REM
90 PRINT " "
100 PRINT "******* ANSWER THE FOLLOWING *******"
110 PRINT "Resist name    ";:INPUT RESIST$
120 PRINT "Developer [say, MiF 1:3, 3 is the water]";:INPUT DEVELOPER$
130 PRINT "Develop Temperature [°C]";:INPUT DEVTEMP
140 PRINT " "
150 PRINT "each wafer is given a file number by the user; the"
160 PRINT "file number should be of the form mmddrr, where mm is"
170 PRINT "month, dd is date, and rr is run number, example: on"
180 PRINT "May 13 we do run 95 the file number would be given 051395 "
190 PRINT " "
200 ERASE H,L              'erase the previous result
210 DIM H(6000),L(6000)     'still have room to spare, check with "print fre(x)
220 PRINT " "
230 PRINT "The following measurement is for ..."
240 PRINT RESIST$
250 PRINT DEVELOPER$
260 PRINT "Dev Temp ";DEVTEMP
270 PRINT " "
280 PRINT "All correct [y/n] ";:INPUT PRC$
290 IF PRC$ <> "y" AND PRC$ <> "n" THEN 220
300 IF PRC$ = "n" THEN 90
310 PRINT "Specify file number ( 000000 = end ) ";:INPUT A$
320 IF A$="000000" GOTO 2500
330 B$="B:"
340 D$=".DAT"
350 I$ = B$ + A$ + D$
360 IT$ = B$ + A$ + "T" + D$
370 REM
380 NMAX=6000 :DTMIN=.02# :DTMAX=60! :DTPLOT=.04#
390 PRINT "Did you expose the resist [y/n] ";:INPUT EXPO$
400 IF EXPO$ <> "y" AND EXPO$ <> "n" THEN 390
410 IF EXPO$="n" THEN D=0 :T2=0 :GOTO 510
420 PRINT "Reading Exposure File ... "
430 OPEN IT$ FOR INPUT AS #1
440 INPUT #1, REST$
450 INPUT #1, BKTEMP
460 INPUT #1, BKTIME
470 INPUT #1, PERIOD
480 INPUT #1, NT,ET1,D,T2,REFL
490 CLOSE #1
500 IF RESIST$ <> REST$ THEN PRINT "Check the Resist Name. "
510 PRINT "Exposed Dose (mJ/cm2)   ";D*T2
520 PRINT " "
530 PRINT "Give the number of sampling points."; : INPUT N
540 IF N > NMAX THEN PRINT "MAX = 6000 " :GOTO 530
550 PRINT " "
560 PRINT "minimum data collection rate for safety : t1(sec) =";DTMAX
570 PRINT "maximum data collection rate:"
```

```
580 PRINT "without displaying data in real time    : t1(sec) ="；DTMIN
590 PRINT "with    displaying data in real time    : t1(sec) ="；DTPLOT
600 PRINT " "
610 PRINT "time between sampling points (sec)"; : INPUT T1
620 IF T1 < DTMIN  THEN PRINT "miminum is ";DTMIN :GOTO 580
630 IF T1 < DTPLOT THEN PRINT "... plot miminum is ";DTPLOT
640 IF T1 > DTMAX  THEN PRINT "maximum is ";DTMAX : GOTO 610
650 CLS
660 ABO=0
670 PRINT "This is the input."
680 PRINT " "
690 PRINT "The data file name is ";A$ :IF ABO=1 THEN INPUT A$
700 PRINT "Exposed Dose (mJ/cm2)  ";D*T2
710 PRINT "# of sampling points  ";N :IF ABO=1 THEN INPUT N
720 IF N > NMAX THEN PRINT "MAX = ";NMAX :GOTO 710
730 PRINT "Time interval          ";T1 :IF ABO=1 THEN INPUT T1
740 IF T1 < DTMIN  THEN PRINT "miminum is ";DTMIN : GOTO 730
750 IF T1 < DTPLOT THEN PRINT "... plot miminum is ";DTPLOT
760 IF T1 > DTMAX  THEN PRINT "maximum is ";DTMAX : GOTO 730
770 PRINT " "
780 IF ABO=1 THEN ABO=0 :GOTO 670
790 PRINT "Do you want to change any one above [y/n] ";:INPUT ABO$
800 IF ABO$ <> "y" AND ABO$ <> "n" THEN 790
810 IF ABO$ = "y" THEN ABO=1 :GOTO 680
820 REM
830 PRINT " "
840 ADRS=1808               'I/O starting address of Tecmar Lab Master Board
850 OUT ADRS+9,255          'master reset the timer
860 OUT ADRS+9,95           'load all counters
870 X = INP(ADRS+6)         'reset the status bit
880 OUT ADRS+4,128          'control byte of A/D (10000000)
890 OUT ADRS+5,CHAN         'A/D input channel number
900 OUT ADRS+9,23          'data pointer to master mode register
910 OUT ADRS+8,0           'master mode reg l (0000 0 0 00)
920 OUT ADRS+8,128         'master mode reg h (1 0 0 0 0000)
930 OUT ADRS+9,5           'data pointer to counter mode reg of counter 5
940 OUT ADRS+8,49          'counter mode register l (00110 001)
950 TA=.75 :PA=.001 :CMRH%=14
960 TTA=TA/PA :TEMPA=TTA\100
970 TLL%=FNBCD(TTA-TEMPA*100) :TLH%=FNBCD(TEMPA)
980 OUT ADRS+8, CMRH%
990 OUT ADRS+8, TLL%
1000 OUT ADRS+8, TLH%
1005 PRINT " "
1010 PRINT "Load a substrate(RESIST SURFACE at BOTTOM) and align the detector."
1020 PRINT "If you hit any key the voltage will be printed out on the screen."
1030 PRINT "Least background light is desirable. So, turn the room light off."
1040 PRINT "Adjust the detector positioner such that the main beam must be "
1050 PRINT "focussed within the diode window."
1060 PRINT "Also adjust gain such that voltage is around 5 volts at least."
1070 PRINT "When you get the right beam watch at least 10 samples printed out."
1080 PRINT "Then hit any key one more time."
1090 PRINT " "
1100 PRINT "Now hit any key for the voltage measurement."
1110 J=0
1120 IF INKEY$ = "" THEN 1120
1130 OUT 1817,112 :OUT 1812,132
1140 IF INP(1812) < 128 THEN 1140
1150 H=256*INP(1814) + INP(1813)
1160 IF H > 32767 THEN H=H-65536!
1170 H=H/204.8
1180 K=J-(J\10)*10
1190 G(K)=H :J=J+1
1200 PRINT USING "###";K;:PRINT USING "###.####" ;H
1210 IF INKEY$ <> "" THEN 1230
1220 GOTO 1140
```

```
1230 PRINT " "
1240 HMIN=G(0) :HMAX=G(9) :GSUM=0
1250 FOR I=0 TO 9
1260 IF G(I) < HMIN THEN HMIN=G(I)
1270 IF G(I) > HMAX THEN HMAX=G(I)
1280 GSUM=GSUM+G(I)
1290 NEXT I
1300 GAV=GSUM/10 :DHX=(HMAX-GAV)/GAV :DHN=(GAV-HMIN)/GAV
1310 DMAX=DHX :IF DHN > DHX THEN DMAX=DHN
1315 PRINT "The power is averaged for the last 10 samples."
1320 PRINT "Power =";GAV;:PRINT " ( +-";DMAX*100;:PRINT "% )"
1322 IF DMAX <= .01 THEN 1360
1325 PRINT "Deviation is large."
1326 PRINT "Will you align the beam again ? [y/n] " ;:INPUT REALN$
1327 IF REALN$ <> "n" THEN 1005
1360 REM now ready to collect data
1370 ADRS=1808              'I/O starting address of Tecmar Lab Master Board
1380 OUT ADRS+9,255         'master reset the timer
1390 OUT ADRS+9,95          'load all counters
1400 X = INP(ADRS+6)        'reset the status bit
1410 OUT ADRS+4,128         'control byte of A/D (10000000)
1420 OUT ADRS+5,CHAN        'A/D input channel number
1430 OUT ADRS+9,23         'data pointer to master mode register
1440 OUT ADRS+8,0          'master mode reg l (0000 0 0 00)
1450 OUT ADRS+8,128        'master mode reg h (1 0 0 0 0000)
1460 OUT ADRS+9,5          'data pointer to counter mode reg of counter 5
1470 OUT ADRS+8,49         'counter mode register l (00110 001)
1480 REM fast clk range = .001 to 9.999 sec; slow clk range = .01 to 99.99 sec
1490 IF T1 <= 2 THEN FAST = 1 ELSE FAST = 0 'DECIDE TO USE FAST OR SLOW CLOCK
1500 IF FAST = 1 THEN P = .001  ELSE P = .01 'P = PERIOD OF TIMER CLK PULSES
1510 TI = T1/P       'ti=timer load value in integer form
1520 REM convert the timer load value from 16-bit binary to 16-bit BCD
1530 REM the timer load values are in BCD format, 0-9999 for 16-bit data
1540 TEMP = TI\100
1550 TLL% = FNBCD(TI-TEMP*100)
1560 TLH% = FNBCD(TEMP)
1570 IF FAST=1 THEN CMRH%=14 ELSE CMRH%=15  '14=use f4(.001s)  15=use f5(.01s)
1580 OUT ADRS+8,CMRH%     'counter mode reg h (000 01110)
1590 OUT ADRS+8,TLL%       'counter load value l     clv=x+6*int(x/10)
1600 OUT ADRS+8,TLH%      'counter load value h
1610 REM the following program segment has been optimized
1620 REM  1817 = adrs+9      adrs=1808
1630 REM  1812 = adrs+4
1640 REM  1813 = adrs+5
1650 REM  1814 = adrs+6
1660 IF T1 >= DTPLOT THEN GOTO 2050
1670 PRINT "***** NO-PLOT during the data collection *****"
1680 REM *** between the two marks the program has to run quick as a rabbit
1690 PRINT "Hit any key to stop during data collection "
1700 PRINT "when ready type go"
1710 INPUT G$
1720 IF G$<>"go" THEN 1700
1730 PRINT "collecting data..."
1740 OUT 1817,112:OUT 1812,132
1750 FOR I=1 TO N
1760  IF INP(1812)<128 THEN 1760
1770  L(I)=INP(1813):H(I)=INP(1814):IF INKEY$<>"" THEN 1810
1780 NEXT I
1790 REM *** the program can now leisurely walk as a turttle
1800 GOTO 1830
1810 N = I                 'because the data collection was cut short, so is n
1820 PRINT "# of data collected = ";N
1830 REM convert data from two's complement to floating point format
1840 PRINT "processing data..."
1850 FOR I=1 TO N
1860  H(I)=256*H(I)+L(I)
1870  IF H(I)>32767 THEN H(I)=H(I)-65536!
```

```
1880  H(I)=H(I)/204.8
1890 NEXT I
1900 REM
1910 CLS: KEY OFF : SCREEN 2
1920 GOSUB 2520
1930 GRID = 0
1940 FOR I=1 TO N
1950  PSET(I-(I\639)*639,198-H(I)*20)
1960  IF I/639 = I\639 THEN GRID=1 :CLS
1970  IF GRID = 0 THEN 2000
1980  GOSUB 2520
1990  GRID = 0
2000 NEXT I
2010 BEEP :BEEP :BEEP
2020 PRINT "Read the development temperature ";:INPUT DEVTEMP
2030 IF DEVTEMP <=0 OR DEVTEMP > 100 THEN  2010
2040 GOTO 2360
2050 REM This routine is the data collection routine during which the collected
2060 REM data are displayed in real time: consequently, the rate of data
2070 REM collection is slow.  data rate = 1/.032 = 31
2080 REM Note: In order to save exeuction time, this segment of the code works
2090 REM       for positive voltage only.
2100 REM graphics information
2110 REM   (    0,   0)   (639,   0)    each coordinate pair = (x,y) or (col,row)
2120 REM   (    0,199)    (639,199)     the whole screen = 640x200
2130 CLS          'clear screen
2140 KEY OFF      'turn off the function-key display
2150 SCREEN 2     'high-resolution graphic screen (640x200)
2160 REM *** between the two marks the program has to run quick as a rabbit
2170 PRINT "Hit any key to stop during data collection "
2180 GOSUB 2520
2190 PRINT "when ready type go ";:INPUT G$
2200 IF G$<>"go" THEN 2190
2210 OUT 1817,112:OUT 1812,132
2220 FOR I=1 TO N
2230  IF INP(1812)<128 THEN 2230
2240 H(I)=(256*(INP(1814))+INP(1813))/204.8:PSET(I-(I\639)*639,198-H(I)*20):IF I
NKEY$<>"" THEN 2290
2250 IF I/639 = I\639 THEN CLS
2260 NEXT I
2270 REM *** the program can now leisurely walk as a turttle
2280 GOTO 2300
2290 N = I                  'because the data collection was cut short, so is n
2300 PRINT "# of data collected = ";N
2310 GOSUB 2520
2320 REM
2330 BEEP :BEEP :BEEP
2340 PRINT "Read the development temperature ";:INPUT DEVTEMP
2350 IF DEVTEMP <=0 OR DEVTEMP > 100 THEN  2330
2360 PRINT "saving data on  ";B$+A$+D$
2370 OPEN I$ FOR OUTPUT AS #1
2380 PRINT #1,RESISTS$
2390 PRINT #1,DEVELOPERS$
2400 PRINT #1,DEVTEMP
2410 PRINT #1,USING "####.#####";N,T1,D,T2,REFL
2420 FOR I=1 TO N STEP 7
2430 PRINT #1,USING "####.#####";H(I),H(I+1),H(I+2),H(I+3),H(I+4),H(I+5),H(I+6)
2440 NEXT I
2450 CLOSE #1
2460 PRINT " "
2470 PRINT "Will you continue [y/n] ";:INPUT CON$
2480 IF CON$ <> "y" THEN 2500
2490 GOTO 200
2500 PRINT "***** THANK YOU FOR CHECKING EVERYTHING  !!! *****"
2510 END
```

```
2520 REM Subroutine GRID
2530 MG=1 :MGY=1
2540 XTIC=10 :YTIC=.5
2550 NGRID=XTIC*MG :LRESET=10*NGRID
2560 LTEN=-1 :LXTEN=1
2570 FOR L=1 TO 631 STEP NGRID
2580 PSET(L,198)
2590 LTEN=LTEN+1
2600 IF LTEN=LXTEN*10 THEN LXTEN=LXTEN+1 :PSET(L,197)
2610 NEXT L
2620 NYSTP=YTIC*20*MGY :NYRESET=2*NYSTP
2630 FOR LY=0 TO 198 STEP NYSTP
2640 PSET(1,198-LY)
2650 IF LY/NYRESET = LY\NYRESET THEN PSET(2,198-LY)
2660 IF LY/5/NYRESET = LY\(5*NYRESET) THEN PSET(3,198-LY):PSET(4,198-LY)
2670 NEXT LY
2680 RETURN
```

```
10 REM Program Name: Peaknois.bas  by Deok J. Kim ;  Version 3.1   Feb 16  '83
20 REM This program finds the maximum and minimum points faster than peakdeok
30 REM This program also remove noise and replace it with quadrtic extrapolate
40 DIM V(5000),NPV(30),SUI$(30)
50 PRINT "Which disk has the data file [a/b] ";:INPUT DISK$
60 IF DISK$ <> "a" AND DISK$ <> "b" THEN 50
70 PRINT " "
80 PRINT "Do you want to see the data files on the disk [y/n] ";:INPUT ANW$
90 IF ANW$ <> "y" AND ANW$ <> "n" THEN 80
100 PRINT " "
110 IF ANW$ = "y" AND DISK$="a" THEN FILES "a:*.dat"
120 IF ANW$ = "y" AND DISK$="b" THEN FILES "b:*.dat"
130 PRINT " "
140 PRINT "Enter the file name(only number) to be plotted.";:INPUT FILENUMBER$
150 PRINT "If the peak file is there  then  type  y ";:INPUT PF$
160 PRINT "If the noise file is there  then  type  y ";:INPUT NO$
170 FILEEXTENTION$= "dat"
180 FILENAME$ = DISK$ +":"+ FILENUMBER$ +"."+ FILEEXTENTION$
190 IF NO$="y" THEN FILENAME$=DISK$+":"+FILENUMBER$+"no."+FILEEXTENTION$
200 OPEN FILENAME$ FOR INPUT AS #1
210 ERASE V
220 DIM V(5000)
230 INPUT #1,RESIST$
240 INPUT #1,DEVELOPER$
250 INPUT #1,DEVTEMP
260 INPUT #1, N,DT,PW,TE,REFL
270 FOR I=1 TO N STEP 7
280      INPUT #1,V(I),V(1+I),V(2+I),V(3+I),V(4+I),V(5+I),V(6+I)
290 NEXT I
300 CLOSE #1
310 PRINT " "
320 PRINT "Do you want to see raw data [y/n] " ;:INPUT AN$
330 IF AN$ <> "y" AND AN$ <> "n" THEN 320
340 IF AN$ = "n" THEN GOTO 420
350 PRINT RESIST$
360 PRINT DEVELOPER$
370 PRINT "Dev Temp ";DEVTEMP
380 PRINT N,DT,PW,TE,REFL
390 FOR I=1 TO N STEP 7
400      PRINT USING "####.#####";V(I),V(I+1),V(I+2),V(I+3),V(I+4),V(I+5),V(I+6)
410 NEXT I
420 IF PF$ <> "y" THEN 550
430 PEAK$=DISK$+":"+FILENUMBER$+"p"+".dat"
440 OPEN PEAK$ FOR INPUT AS #1
450   INPUT #1, RESIST$
460   INPUT #1, DEVELOPER$
470   INPUT #1, DEVTEMP
480   INPUT #1, NF,DT,PW,TE,REFL
490   FOR KP=0 TO (NF-1)
500   INPUT #1, NPV(KP),WHATEVER
510   NEXT KP
520   NPVS=NF-2
530 CLOSE #1
540 GOTO 1800
550 IR=0
560 REM Plot Curve.
570 MG=1
580 MGY=1
```

```
590 YA=0:ARROW=0:MAN=0
600 NS=1 : NE=N  'This is the default plot range
610 CLS :KEY OFF :SCREEN 2
620 GOSUB 2540
630 NMGSET=0 :NSALL=NS :NEALL=NE
640 PRINT "Can you find all the verteces from the picture [y/n]";:INPUT APV$
650 IF APV$ <> "y" AND APV$ <> "n" THEN 640
660 IF APV$ = "n" THEN 560
670 PRINT " "
680 PRINT "If the drop point and the first vertex are not distinctive "
690 PRINT "then take the next one as the first vertex"
700 PRINT " "
710 PRINT "Type 1 for first peak or 0 for first valley ";:INPUT FIRST
720 IF FIRST <> 1 AND FIRST <> 0 THEN 710
730 PRINT "How many peaks and valleys ?  Give the total number. ";:INPUT NPVS
740 IF FIRST=1 THEN PRINT "*** First peak ***"
750 IF FIRST=0 THEN PRINT "*** First valley ***"
760 PRINT "The total # of peaks and valleys = ";NPVS
770 IF NPVS < 1 THEN 700
780 REM Biginning of arrow-work
790 IF MG < 1 THEN PRINT "Divide the picture properly. MG >= 1":GOTO 560
800 ISUM=0 :NTEST=NPVS+1
810 YA=1                  'for Arrow-Work Routine
820 PRINT " "
830 PRINT " "
840 PRINT "The picture will come up on the screen. When you decide the"
850 PRINT "arrow position an arrow begins to grow from the left side."
860 PRINT "You should stop it as soon as it passes the interested point"
870 PRINT "by hitting any key. Hit again and it grows again."
880 PRINT "Repeat the action until you find all points."
890 PRINT "Plot ";NS;:PRINT " - ";NE;:PRINT " : Mag x=";MG;:PRINT ",y=";MGY
900 PRINT "You may change mag in x and y ";:INPUT MG,MGY
910 IF MG < 1 THEN PRINT "*** Mx must be POSITIVE INTEGER ***" :GOTO 900
920 IF(MG*(NE-NS) > 600 ) THEN NE=NS+INT(600/MG)
930 PRINT "You can vary the speed of arrow. Min=1,Max=7";:INPUT SPEED
940 IF SPEED < 1 OR SPEED > 7 THEN 930
950 NSPEED=INT(8-SPEED)
960 IF (MG*(NE-NS) > 600 ) THEN PRINT "Change end-point ";:INPUT NE
970 IF (MG*(NE-NS) > 600 ) THEN PRINT "Too large !!":GOTO 960
980 CLS:KEY OFF:SCREEN 2
990 GOSUB 2540
1000 PRINT "Give arrow position. 0 < max,min < 10 ";:INPUT YMAX,YMIN
1010 IF YMAX <= 0 OR YMIN >= 10 THEN 960
1020 IF YMIN <= 0 OR YMAX >= 10 THEN 960
1030 GOSUB 3220
1040 IF ISUM > NTEST THEN 1070
1050 NS = NPV(ISUM-1) :NE=N
1060 ARROW=1 :GOTO 810
1070 SMTH$ = "n"
1080 REM Find oput the points around the arrow-stop
1090 REM Computer will search 10 points around it.
1100 REM Find drop-point first.
1110 N1=NPV(0)-5
1120 IF N1 < 1 THEN N1 = 1
1130 NDROP=NPV(0)+5
1140 IF V(NDROP-1) > V(NDROP)*1.3 THEN 1190
1150 IF NDROP=N1 THEN 1190
1160 BEEP
1170 NDROP=NDROP - 1
1180 GOTO 1140
1190 NPV(0)=NDROP+1
1200 REM Find peaks and valleys
1210 TEST=FIRST :IT=0
1220 IF TEST = 1 THEN GOSUB 3360
1230 IF IT >= NPVS THEN 1280
1240 IF TEST = 0 THEN GOSUB 3450
1250 IF IT >= NPVS THEN 1280
```

```
1260 BEEP
1270 GOTO 1220
1280 REM Find the flat point
1290 N1=NPV(NPVS+1)+1    :N2=NPV(NPVS+1)+5
1300 IF N2>N THEN N2=N
1310 NFLAT=N2
1320 IF NFLAT < N1 THEN NPV(NPVS+1)=NFLAT :GOTO 1410
1330 NFLAT1=ABS(V(NFLAT-1)-V(NFLAT))
1340 NFLAT2=ABS(V(NFLAT-2)-V(NFLAT))
1350 NFLAT3=ABS(V(NFLAT-3)-V(NFLAT))
1360 IF NFLAT1 > .003 AND NFLAT2 > .003  THEN 1390
1370 BEEP
1380 NFLAT=NFLAT-1 :GOTO 1320
1390 IF NFLAT3 > .003 THEN 1410
1400 NFLAT=NFLAT-1 :GOTO 1320
1410 NPV(NPVS+1)=NFLAT
1420 REM Check the points
1430 NS=1:NE=N:MG=1:MGY=1:YA=0
1440 CLS:KEY OFF:SCREEN 2
1450 PRINT "***** X-Mag must be POSITIVE INTEGER *****"
1460 PRINT "***** (plot-points) x (X-Mag) <= 630 *****"
1470 LIMPLOT=1
1480 GOSUB 2540
1490 IF LIMPLOT=1 THEN 1440
1500 GOSUB 3540
1510 PRINT " "
1520 PRINT "... correct [y/n] ";:INPUT ALC$
1530 IF ALC$ <> "y" AND ALC$ <> "n" THEN 1520
1540 IF ALC$ = "n" THEN 1590
1550 PRINT "Did you check all verteces [y/n] ";:INPUT ALCH$
1560 IF ALCH$ <> "y" AND ALCH$ <> "n" THEN 1550
1570 IF ALCH$ = "n" THEN 1430
1580 GOTO 1800
1590 MAN=1
1600 PRINT "How many are unbalanced ";:INPUT MH
1610 IF MH<=0 OR MH=>(NPVS+2) THEN 1600
1620 DIM WR(30)
1630 PRINT "Enter the wrong one "
1640 FOR I=1 TO MH
1650 INPUT WR(I)
1660 NEXT I
1670 FOR JW=1 TO MH
1680 NW=WR(JW)
1690 CLS:KEY OFF :SCREEN 2
1700 PRINT "******* correcting  = ";NPV(NW)
1710 NE=NPV(NW)+10 :NS=NPV(NW)-10 :IF NS <= 0 THEN NS=1
1720 MG=5
1730 GOSUB 2540
1740 GOSUB 3540
1750 PRINT "Enter the right number ";:INPUT NPV(NW)
1760 NEXT JW
1770 MAN=0
1780 ERASE WR
1790 GOTO 1430
1800 PRINT " "
1810 PRINT "**************************************************"
1820 PRINT " "
1830 PRINT "The information obtained from the plot-work"
1840 PRINT " "
1850 PRINT "**************************************************"
1860 FOR I=1 TO NTEST
1870   IF NPV(I) < NPV(I-1) THEN PRINT "Wrong order in vortex number.":GOTO 550
1880 NEXT I
1890 PRINT " "
1900 PRINT "Drop point =";:PRINT USING "#####";NPV(0)
```

```
1910 FIRST=0 :I=1 :JX=0
1920 IF V(NPV(1)) > V(NPV(2)) THEN FIRST = 1
1930 IF FIRST = 1 THEN I=0 :JX=1
1940 FOR J=1 TO NPVS
1950      I=I+(-1)^(J+JX)
1960      K=ABS(I)
1970 PRINT USING "###";K;:PRINT USING "#####";NPV(J);:PRINT   USING "###.#####";
V(NPV(J))
1980 NEXT J
1990 PRINT "End point  =";:PRINT USING "#####";NPV(NPVS+1)
2000 PRINT " "
2010 PRINT "Do you want to check [y/n] ";:INPUT COR$
2020 IF COR$ <> "y" AND COR$ <> "n" THEN 2010
2030 IF COR$ = "y" THEN 1420
2040 IF SMTH$ = "y" THEN 2250
2050 PRINT "Do you want to check noise [y/n] ";:INPUT CNS$
2060 IF CNS$="n" THEN 2250
2070 PRINT "***** noise checking *****"
2080 NOISE=.05
2090 NS=1 :NE=NPV(0)
2100 GOSUB 3780
2110 FOR NC=1 TO NPVS+1
2120  NS=NPV(NC-1) :NE=NPV(NC)
2130  GOSUB 3780
2140 IF NOI$ <> "x" THEN 2160
2150 GOSUB 3780
2160 NEXT NC
2170 NS=1 :NE=NPV(NPVS+1)
2180 CLS:KEY OFF:SCREEN 2:MG=1:MGY=1
2190 GOSUB 2540
2200 PRINT " "
2210 PRINT "All smooth out [y/n] ";:INPUT SMTH$
2220 IF SMTH$ <> "y" AND SMTH$ <> "n" THEN 2210
2230 IF SMTH$ = "y" THEN 2250
2240 GOTO  5150
2250 PRINT "On which disk do you want to save it [a/b] ";:INPUT SV$
2260 IF SV$ <> "a" AND SV$ <> "b" THEN PRINT "Type a or b ":GOTO 2250
2270 PEAK$ = SV$ + ":" + FILENUMBER$ + "p" + "." + "dat"
2280 NDPF=NPVS+2
2290 OPEN PEAK$ FOR OUTPUT AS #1
2300 PRINT #1,RESIST$
2310 PRINT #1,DEVELOPER$
2320 PRINT #1,DEVTEMP
2330 PRINT #1,NDPF,DT,PW,TE,REFL
2340 NDPF1=NPVS+1
2350 FOR IP=0 TO NDPF1
2360     PRINT #1, USING "####.#####";NPV(IP),V(NPV(IP))
2370 NEXT IP
2380 CLOSE #1
2400 FILENO$=SV$+":"+FILENUMBER$+"no"+".dat"
2410 OPEN FILENO$ FOR OUTPUT AS #1
2420 PRINT #1,RESIST$
2430 PRINT #1,DEVELOPER$
2440 PRINT #1,DEVTEMP
2450 PRINT #1,NPV(NPVS+1),DT,PW,TE,REFL
2460 FOR I=1 TO NPV(NPVS+1) STEP 7
2470 PRINT #1,USING "####.#####";V(I),V(I+1),V(I+2),V(I+3),V(I+4),V(I+5),V(I+6)
2480 NEXT I
2490 CLOSE #1
2500 END
2510 REM
2520 REM
2530 REM This is the subroutine PLOTMAG.
2540 PRINT "Total # of data  ";N
2550 PRINT "Plot range from ";NS;:PRINT " to ";NE
2560 IF YA <> 0 THEN 2740
```

```
2570 PRINT "Will you change the plot range [y/n] ";:INPUT PCH$
2580 IF PCH$="y" THEN PRINT "Give two numbers. #1,#2  ";:INPUT NS,NE
2590 IF LIMPLOT=1 AND (NE-NS)>630 THEN PRINT "Maximum points <= 630":GOTO 2570
2600 IF NS > NE THEN PRINT "You are a fool.  Nstart < Nend !!!":GOTO 2570
2610 PRINT  "Do you want to change magnification [y/n]";:INPUT MAG$
2620 IF MAG$ <> "y" AND MAG$ <> "n" THEN 2610
2630 IF MAG$ = "n" THEN 2690
2640 NMGSET=0
2650 PRINT  "                    in X-axis ";:INPUT MG
2660 IF LIMPLOT=1 AND MG < 1 THEN PRINT "Give positive integer.":GOTO 2640
2670 IF MG < 0 THEN NMGSET=1 :MGA=ABS(MG)
2680 PRINT  "                    in Y-axis ";:INPUT MGY
2690 TST=NS*DT
2700 TEND=NE*DT
2710 PRINT " "
2720 PRINT "Plot begins at ";TST;:PRINT "sec";:PRINT " ends at ";TEND;:PRINT "se
c"
2730 PRINT " "
2740 MK=1
2750 IF MG <= 0 THEN MG=1:MK=MGA:NN=INT((NE-NS)/MGA +1):GOTO 2770
2760 NN=MG*(NE-NS)+1
2770 IF LIMPLOT=1 AND NN > 638 THEN PRINT "DIVIDE PLOT-RANGE !!"
2780 GOSUB 4970
2790 IF LIMPLOT=1 AND NN > 638 THEN RETURN
2800 LIMPLOT=0
2810 K=1
2820 FOR I=1 TO NN STEP MG
2830      Y=MGY*V(K+NS-1)
2840      PSET(I-(I\639)*639,199-Y*20)
2850      IF I/639 = I\639 THEN 3150
2860 K=K+MK
2870 NEXT I
2880 IF NE > N GOTO 3210
2890 IF IR > 1 THEN 3210
2900 IF MAN <> 1 THEN 3210
2910 PRINT "Do you want to check the data in the plot [y/n] ";:INPUT PR$
2920 IF PR$ <> "y" AND PR$ <> "n" THEN 2910
2930 IF PR$ = "n" THEN 3090
2940 PRINT "Beginning number ";NS
2950 NQ=INT((NE-NS+1)/7)
2960 NEE=7*NQ+NS-1
2970 NR=NE-NEE
2980 FOR I=NS TO NEE STEP 7
2990 PRINT USING "####.#####";V(I),V(I+1),V(I+2),V(I+3),V(I+4),V(I+5),V(I+6)
3000 NEXT I
3010 IF NR = 0 THEN 3080
3020 I=0
3030 I=I+1
3040 PRINT USING "####.#####" ;V(NEE+I);
3050 IF I >= NR THEN 3070
3060 GOTO 3030
3070 PRINT "xxxxxxxxxx"
3080 PRINT "Ending    number ";NE
3090 IF IR > 1 THEN 1590
3100 REM PRINT "Do you want to see another section [y/n]";:INPUT SEE$
3110 REM IF SEE$ <> "y" AND SEE$ <> "n" THEN 2610
3120 REM IF SEE$ = "y" THEN CLS :GOTO 330
3130 REM IF SEE$ = "n" THEN 2700
3140 RETURN
3150 PRINT "Do you want yo plot further [y/n] ";:INPUT YE$
3160 IF YE$ <> "y" AND YE$ <> "n" THEN 3150
3170 IF YE$ = "n" THEN 3210
3180 CLS
3190 GOSUB 4970
3200 GOTO 2870
3210 RETURN
```

```
3220 REM Subroutine Arrow Growth
3230 FOR I=1 TO NN
3240   FOR K=1 TO NSPEED
3250   PSET(I,199-YMAX*MGY*20)
3260   PSET(I,199-YMIN*MGY*20)
3270   NEXT K
3280   IF INKEY$ <> "" THEN 3310
3290 NEXT I
3300 RETURN
3310 IF INKEY$ = "" THEN 3310
3320 NPV(ISUM)=INT((I-1)/MG)+NS
3330 IF ISUM = NTEST THEN ISUM=ISUM+1 :GOTO 3300
3340 ISUM=ISUM+1
3350 GOTO 3290
3360 REM Max-routine
3370 IT=IT+1
3380 N1=NPV(IT)-5
3390 N2=NPV(IT)+5
3400 FOR KR=N1 TO N2
3410 IF V(KR) > V(NPV(IT)) THEN NPV(IT)=KR
3420 NEXT KR
3430 TEST=0
3440 RETURN
3450 REM Min-routine
3460 IT=IT+1
3470 N1=NPV(IT)-5
3480 N2=NPV(IT)+5
3490 FOR KR=N1 TO N2
3500 IF V(KR) < V(NPV(IT)) THEN NPV(IT)=KR
3510 NEXT KR
3520 TEST=1
3530 RETURN
3540 REM Re-calculated Arrow-point plot
3550 IF NMGSET=0 THEN 3610
3560 PRINT "X-mag < 0 !! Hit any key to plot again."
3570 IF INKEY$ = "" THEN 3570
3580 GOTO 1430
3590 ICH1=0 :ICH2=0
3600 IF NE < NPV(0) THEN PRINT "... RETURN " :GOTO 3770
3610    FOR I=0 TO NPVS+1
3620      IF NPV(I) >= NS THEN ICH1=I :GOTO 3640
3630    NEXT I
3640    FOR I=0 TO NPVS+1
3650      IF NPV(NPVS+1-I) <= NE THEN ICH2=NPVS+1-I :GOTO 3680
3660    NEXT I
3670 PRINT "=== ";
3680 FOR I=ICH1 TO ICH2
3690    FOR J=1 TO 39
3700       LX=MG*(NPV(I)-NS)+1
3710       Y=199-(10-.25*J)*20
3720       PSET(LX,Y)
3730    NEXT J
3740 PRINT USING "#####";I;
3750 NEXT I
3760 PRINT " ==="
3770 RETURN
3780 REM noise out
3790 NBTW=NE-NS :MG=1
3800 IF NBTW < 319 THEN MG=2
3810 IF NBTW < 213 THEN MG=3
3820 IF NBTW < 160 THEN MG=4
3830 VMAX = V(NS)
3840 FOR I=(NS+1) TO NE
3850  IF V(I) > VMAX  THEN VMAX=V(I)
3860 NEXT I
3870 MAXY=5 :MGY=MAXY/VMAX
```

```
3880 MK=1 :K=1
3890 NN=MG*(NE-NS)+1
3900 CLS:KEY OFF:SCREEN 2
3910 GOSUB 4970
3920 FOR I=1 TO NN STEP MG
3930   Y=MGY*V(K+NS-1)
3940   PSET(I,199-Y*20)
3950   K=K+MK
3960 NEXT I
3970 NX=3 :BEEP
3980 PRINT "NX =";NX ;:PRINT "  NOISE =";NOISE
3990 PRINT "ANY SINGLE NOISE IN THE PICTURE  [x/y/n] ";:INPUT NOI$
4000 IF NOI$ = "n" THEN RETURN
4010 IF NOI$ = "x" THEN PRINT "Give NX ,NOISE  ";:INPUT NX,NOISE
4020 IF NX < 2 THEN PRINT "NX should not be smaller than 2 " :GOTO 3970
4030 K=2 :MK=1
4040 FOR I=1 TO (NN-MG) STEP MG
4050    Y1=V(K+NS-2)
4060    PSET(I,199-MGY*Y1*20)
4070    Y=V(K+NS-1)
4080    IF ABS(Y-Y1) > NOISE THEN 4170
4090       FOR J=1 TO MG
4100       Y(J)=Y1+J*(Y-Y1)/MG
4110       PSET(I+J,199-MGY*Y(J)*20)
4120       NEXT J
4130    K=K+MK
4140 NEXT I
4150 RETURN
4160 END
4170 REM noise warning ************
4180 LOCATE 3,1:PRINT "When it sounds type n for noise or p for passing  "
4190 V$="0"
4200  NTON=220/7
4210    M=220
4220    SOUND M,1.5
4230    V$=INKEY$
4240    IF V$ = "n" THEN 4280
4250    IF V$ = "p" THEN 4280
4260  IF M > 1000 THEN  4090
4270  M=M+NTON :GOTO 4220
4280 LOCATE 3,1:PRINT STRING$(60,3)
4290 IF V$ = "p" THEN 4090
4300 REM extrapolate the noise point by quadratic curve fitting
4310 NREF=NPV(0)
4320 IF NS < NPV(0) THEN NREF=0
4330 IF (K+NS-1) < (NREF+NX) THEN Y=Y1:V(K+NS-1)=Y :GOTO 4090
4340 FIT2 = 0
4350 GOSUB 4670
4360 Y=A0
4370 V(K+NS-1)=Y
4380 GOTO 4090
4390 REM BALANCING the VERTECES
4400 NBAL=3 :NQ=2*NBAL
4410 NBAL1=NBAL
4420 IF NPV(1)-NPV(0) < NBAL THEN NBAL1=NPV(1)-NPV(0)
4430 NQ1=2*NBAL1
4440 FIT2=1
4450 FOR IBV=1 TO NPVS
4460 NX=NBAL
4470 IF IBV=1 THEN NX=NBAL1
4480 IF NX < 2 THEN PRINT "NX < 2 "
4490 GOSUB 4670
4500 V(NPV(IBV))=A0
4510 NSFT=1
4520 NX=NBAL
4530 IF IBV=1 THEN NX=NBAL1
```

```
4540 FOR IX=1 TO NX
4550 IQ=IX*NSFT
4560 YR=A0+A1*IQ+A2*IQ*IQ
4570 IF A2 > 0 THEN 4620
4580 IF V(NPV(IBV)-IQ) > A0 THEN V(NPV(IBV)-IQ)=YR
4590 NEXT IX
4600 IF NSFT = -1 THEN 4660
4610 NSFT=-1 :GOTO 4540
4620 IF V(NPV(IBV)-IQ) < A0 THEN V(NPV(IBV)-IQ) = YR
4630 GOTO 4590
4640 NEXT IBV
4650 FIT2=0
4660 RETURN
4670 REM Quadratic Fitting
4680 NSUM=2*NX
4690 SX=0:SX2=0:SX3=0:SX4=0:SY=0:SXY=0:SX2Y=0
4700 NSFT=1
4710 FOR IX=1 TO NX
4720  IQ=IX*NSFT
4730  SX=SX+IQ
4740  SX2=SX2+IQ^2
4750  SX3=SX3+IQ^3
4760  SX4=SX4+IQ^4
4770  X=K+NS-1-IQ
4780  IF FIT2 = 1 THEN X=NPV(IBV)-IQ
4790  Y=V(X)
4800  SY=SY+Y
4810  SXY=SXY+Y*IQ
4820  SX2Y=SX2Y+Y*IQ*IQ
4830 NEXT IX
4840 IF NSFT = -1 THEN 4890
4850 NSFT=-1 :GOTO 4710
4860 REM
4870 REM calculate A0,A1,A2 in y=A0+A1*iq+A2*iq^2
4880 REM
4890 DET=NSUM*SX2*SX4+2*SX*SX2*SX3-SX2*SX2*SX2-SX*SX*SX4-NSUM*SX3*SX3
4900 A0=SY*SX2*SX4+SX*SX3*SX2Y+SX2*SX3*SXY-SX2*SX2*SX2Y-SX3*SX3*SY-SX*SX4*SXY
4910 A0=A0/DET
4920 A1=NSUM*SXY*SX4+SY*SX2*SX3+SX*SX2*SX2Y-SX2*SX2*SXY-SY*SX*SX4-NSUM*SX3*SX2Y
4930 A1=A1/DET
4940 A2=NSUM*SX2*SX2Y+SX*SX2*SXY+SX*SY*SX3-SX2*SX2*SY-SX*SX*SX2Y-NSUM*SX3*SXY
4950 A2=A2/DET
4960 RETURN
4970 REM GRID subroutine
4980 SCALE = MG
4990 IF NMGSET=1 THEN SCALE=1/MGA
5000 XTIC=10 :YTIC=.5
5010 NGRID=XTIC*SCALE :LRESET=10*NGRID
5020 LTEN=-1 :LXTEN=1
5030 FOR L=1 TO 631 STEP NGRID
5040 PSET(L,199)
5050 LTEN=LTEN+1
5060 IF LTEN=LXTEN*10 THEN LXTEN=LXTEN+1 :PSET(L,198)
5070 NEXT L
5080 NYSTP=YTIC*20*MGY :NYRESET=2*NYSTP
5090 FOR LY=0 TO 199 STEP NYSTP
5100 PSET(1,199-LY)
5110 IF LY/NYRESET = LY\NYRESET THEN PSET(2,199-LY)
5120 IF LY/5/NYRESET = LY\(5*NYRESET) THEN PSET(3,199-LY):PSET(4,199-LY)
5130 NEXT LY
5140 RETURN
5150 REM HEAVY NOISE REMOVAL SUBROUTINE
5160 PRINT "# of zone = "; NPVS+1
5170 PRINT "Enter the heavy noise zone number "::INPUT HEV
5180 IF HEV > NPVS+1 THEN 5160
5190 IF HEV <= 0 THEN 5570
```

```
5200 NHEV=INT(HEV)
5210 NHEVS=NHEV-1
5220 N1=NPV(NHEVS) :N2=NPV(NHEV)
5230 CLS :KEY OFF :SCREEN 2
5240 PRINT "ZONE NUMBER = ";NHEV
5250 PRINT "Plot Zone ";N1 ;:PRINT " - ";N2
5260 PRINT "Plot Range #,#" ;:INPUT NS,NE
5270 IF NS >= NE THEN 5260
5280 IF NS < N1 OR NS > N2 THEN 5260
5290 IF NE < N1 OR NE > N2 THEN 5260
5300 PRINT "Magnification MX,MY ";:INPUT MG,MGY
5310 IF MG < 1 OR MGY < 0  THEN 5300
5320 NN=MG*(NE-NS)+1
5330 K=1
5340 FOR IP=1 TO NN STEP MG
5350      Y=MGY*V(K+NS-1)
5360      PSET(IP-(IP\639)*639, 199-Y*20)
5370 K=K+1
5380 NEXT IP
5390 PRINT "Can you find the two nodal points [y/n] ";:INPUT TNP$
5400 IF TNP$ <> "y" THEN 5160
5410 PRINT "Enter two numbers ";:INPUT J1,J2
5420 IF J1 >= J2 THEN 5410
5430 IF J1 < NS OR J1 > NE THEN 5410
5440 IF J2 < NS OR J2 > NE THEN 5410
5450 FOR J=J1 TO J2
5460      V(J)=V(J1) + (V(J2)-V(J1))*(J-J1)/(J2-J1)
5470 NEXT J
5480 K=1
5490 FOR JP=1 TO NN STEP MG
5500      Y=MGY*V(K+NS-1)
5510      PSET(JP-(JP\639)*639, 199-Y*20)
5520 K=K+1
5530 NEXT JP
5540 PRINT "Hit any key to continue "
5550 IF INKEY$ = "" THEN 5550
5560 CLS : GOTO 5160
5570 GOTO 2170
```

```
10 REM Program RATEX.bas by Deok J. Kim      Version 1.6  May 25  '83
20 REM This is the extended version of RXBEST.BAS (0.9)
30 REM Find rate by fitting data with least square method
40 REM This program produces XR file which can be cenverted to RATE vs DEPTH
50 BEEP
60 DIM H(2300),XT(2300),XTF(2300),V(2300),VT(100),NT(100),NTH(100)
70 DIM KE(100),A0(100),A1(100),A2(100),X(100)
80 PI=3.141592654#          'constant
90 REM The ticks in x,y axes are given by xtic,ytic for plotting
100 XTIC=10 :YTIC=.1
110 PRINT "Which disk has the data file [a/b] ";:INPUT DISK$
120 IF DISK$ <> "a" AND DISK$ <> "b" THEN 110
130 PRINT " "
140 PRINT "Do you want to see the data files on the disk [y/n] ";:INPUT ANW$
150 IF ANW$ <> "y" AND ANW$ <> "n" THEN 140
160 PRINT " "
170 IF ANW$ = "y" AND DISK$ = "a" THEN FILES "a:*.dat"
180 IF ANW$ = "y" AND DISK$ = "b" THEN FILES "b:*.dat"
190 PRINT " "
200 FILEEXTENTION$= "dat"
210 PRINT "Enter the file name(only number) " ;:INPUT FILENUMBER$
220 FILEPEAK$ = DISK$ +":"+ FILENUMBER$ + "p" +"."+ FILEEXTENTION$
230 FILE$  = DISK$ +":"+ FILENUMBER$ + "no" + "." + FILEEXTENTION$
240 PRINT " "
250 PRINT "Open Files ... ";FILEPEAK$,FILE$
260 OPEN FILEPEAK$ FOR INPUT AS #1
270 INPUT #1,RESIST$
280 INPUT #1,DEVELOPER$
290 INPUT #1,DEVTEMP
300 INPUT #1,NF,DT,PW,TX,REFL
310 FOR I=1 TO NF
320      INPUT #1,NT(I),VT(I)
330 NEXT I
340 CLOSE #1
350 OPEN FILE$ FOR INPUT AS #1
360 INPUT #1,RESIST$
370 INPUT #1,DEVELOPER$
380 INPUT #1,DEVTEMP
390 INPUT #1,N,DT,PW,TX,REFL
400 FOR I=1 TO N STEP 7
410    INPUT #1,H(I),H(I+1),H(I+2),H(I+3),H(I+4),H(I+5),H(I+6)
420 NEXT I
430 CLOSE #1
440 PRINT " "
450 PRINT "***** peak   data *****"
460 PRINT " "
470 PRINT RESIST$
480 PRINT DEVELOPER$
490 PRINT "Dev Temp ";DEVTEMP
500 PRINT NF,DT,PW,TX,REFL
510 FOR I=1 TO NF
520      PRINT NT(I),VT(I)
530 NEXT I
540 GOTO 670
550 PRINT "Will you check intensity vs time data  [y/n] ";:INPUT AN$
560 IF AN$ <> "y" AND AN$ <> "n" THEN 550
570 IF AN$ = "n" THEN 670
580 PRINT " "
```

```
590 PRINT RESIST$
600 PRINT DEVELOPER$
610 PRINT DEVTEMP
620 PRINT N,DT,PW,TX,REFL
630 FOR I=1 TO N STEP 7
640   PRINT USING "####.#####"; H(I),H(I+1),H(I+2),H(I+3),H(I+4),H(I+5),H(I+6)
650 NEXT I
660 REM ==================================================================
670 REM find out proper MAX and MIN  for the first fractional wave depth
680 REM ==================================================================
690 VMAX=VT(1) :VMIN=VT(1)
700 FOR IP=2 TO NF
710   IF VT(IP) > VMAX  THEN VMAX=VT(IP)
720   IF VT(IP) < VMIN  THEN VMIN=VT(IP)
730 NEXT IP
740 FIRST=0 :IF VT(1) < VT(2) THEN FIRST=1
750 IF VT(1) > VT(3) THEN VP1=VMAX  :VV1=VT(2)
760 IF VT(1) < VT(3) THEN VP1=VT(3) :VV1=VT(2)
770 IF FIRST=1 AND VT(1) > VT(3) THEN VP1=VT(2) :VV1=VT(3)
780 IF FIRST=1 AND VT(1) < VT(3) THEN VP1=VT(2) :VV1=VMIN
790 REM VP1:first peak ; VV1:first valley
800 REM              ======================
810 REM              =                    =
820 REM              =  find depth vs time  =
830 REM              =                    =
840 REM              ======================
850 PRINT " "
860 PRINT "Selection of the resist type ... "
870 PRINT "   1. KODAK 820 "
880 PRINT "   2. Shipley Microposit 1470 "
890 PRINT "   3. Hunt WX-118 "
900 PRINT "Enter the number. If anything else  then return. " ;:INPUT RESI
910 IF RESI=1 THEN RINX=1.61 :GOTO 950
920 IF RESI=2 THEN RINX=1.62 :GOTO 950
930 IF RESI=3 THEN RINX=1.625 :GOTO 950
940 PRINT "What is the resist index at 6328 ";:INPUT RINX
950    WLUM=.6328
960    ANGA=30 :ANG=ANGA*PI/180
970    TNTH=SIN(ANG)/SQR(RINX*RINX-SIN(ANG)*SIN(ANG))
980    QUAT=WLUM/4/(RINX/COS(ATN(TNTH))-TNTH*SIN(ANG))
990 DAPLOT=0 :DAGPLOT=0 :XTPLOT=0 :XTFPLOT=0 :RATESET=0
1000 XS=0 :PRINT "====== ETCH-DEPTH vs TIME ======"
1010 FOR IP=1 TO NF-1
1020 BEEP :PRINT "XS =";:PRINT USING "###.#####";XS ;:PRINT "  ... IP =";IP
1030    POSPHI=0 :IF VT(IP) > VT(IP+1) THEN POSPHI=1
1040    V0=(VT(IP)+VT(IP+1))/2 :V1=ABS(VT(IP)-VT(IP+1))/2
1050    IF IP=1 THEN V0=(VP1+VV1)/2 :V1=(VP1-VV1)/2
1060    IS=NT(IP) :IE=NT(IP+1)
1070    FOR IK=IS TO IE
1080       IF H(IK)=V0 THEN PHI = PI/2
1090       Y=ABS((H(IK)-V0)/V1) :IF Y >= 1 THEN ARCCOSY=0 :GOTO 1110
1100       ARCCOSY = PI /2 - ATN(Y/SQR(1-Y*Y))
1110       IF H(IK) > V0 THEN PHI = ARCCOSY
1120      IF H(IK) < V0 THEN PHI = PI - ARCCOSY
1130      IF IK=NT(1) THEN PHI0=PHI
1140      IF POSPHI=0 THEN PHI=PI-PHI
1150      IF IK=IS THEN PHIS=PHI
1160      XT=QUAT*(PHI-PHIS)/PI
1170      XT(IK)=XS+XT
1180   NEXT IK
1190 XS=XT(IE)
1200 NEXT IP
1210 BEEP :THICK=XT(NT(NF))
1220 PRINT "The resist film thickness is ";THICK ;:PRINT " um"
1230 GOTO 1270
1240 DAPLOT=1 :XTPLOT=1 :PRINT "Plot Thickness vs Time ......"
```

```
1250 GOSUB 3310
1260 DAPLOT=0 :XTPLOT=0
1270 BEEP
1280 PRINT "=========== LEAST SQUARE FITTING =============="
1290 KE(0)=0 :KE(1)=NT(1) :MSDCOMP=0 :PRINT "MSD Compare Mode =";MSDCOMP
1300 PRINT "The drop point   =";NT(1)
1310 NDIV= 8 :PRINT "# of division    =";NDIV
1320 IWID=INT((NT(NF-2)-NT(1))/NDIV) :IDP=NDIV+1
1330 FOR IP=2 TO IDP
1340 KE(IP)=KE(IP-1)+IWID
1350 NEXT IP
1360 PRINT " "
1370 PRINT "+++++  division schedule  +++++"
1380 PRINT "           WINDOW =";IWID
1390 FOR IP=1 TO IDP
1400 PRINT USING "      #####";IP,KE(IP)
1410 NEXT IP
1420 PRINT " "
1430 PRINT "************* FITTING STARTS *****************"
1440 BEEP :XSKIP=0 :XR=.1   'XR is a retardation depth
1450 IP=1 :KP=2 :IF MSDCOMP=0 THEN KP=1
1460 IPB=0 :NTH(0)=KE(1) :SDMX=0
1470 IF IP = IDP THEN 1990
1480 IF IP = IDP-1 THEN KP=1
1490 PRINT " "
1500 GOSUB 2700
1510 IF MSDCOMP=0 THEN MIB=1 :KF=2 :GOTO 1770
1520 MINMSD=10
1530 FOR IB=1 TO KP
1540 IBS=KE(IP) :IBE=KE(IP+IB)
1550 MSD1=0
1560 MSD2=0
1570 PRINT "IP = ";IP ;:PRINT "  ...  IB =";IB :BEEP
1580 PRINT FL0(IB),FL1(IB)
1590 PRINT FP0(IB),FP1(IB),FP2(IB)
1600   FOR IM=IBS TO IBE
1610   TM=DT*IM
1620   DEL1 = TM*FL1(IB) + FL0(IB) - XT(IM)
1630   DEL2 = TM*TM*FP2(IB) + TM*FP1(IB) + FP0(IB) - XT(IM)
1640   MSD1 = MSD1 + DEL1*DEL1
1650   MSD2 = MSD2 + DEL2*DEL2
1660   PRINT USING "  #.####^^^^  ";MSD1,MSD2
1670   NEXT IM
1680 NMSD=IBE-IBS+1
1690 IF MSD1/NMSD  < MINMSD THEN MINMSD=MSD1/NMSD  :KF=1 :MIB=IB
1700 IF MSD2/NMSD  < MINMSD THEN MINMSD=MSD2/NMSD  :KF=2 :MIB=IB
1710 NEXT IB
1720 PRINT "MINIMUM MSD = ";:PRINT USING "#.###^^^^";MINMSD :
1730 IF KF=1 THEN PRINT " for LINEAR FIT ";
1740 IF KF=2 THEN PRINT " for PARABOLIC FIT ";
1750 IF MIB=1 THEN PRINT " with next first division"
1760 IF MIB=2 THEN PRINT " with next second division"
1770 IPB=IPB+1 :NTH(IPB)=KE(IP+MIB)
1780 IF KF=1 THEN A0(IPB)=FL0(MIB) :A1(IPB)=FL1(MIB) :A2(IPB)=0
1790 IF KF=2 THEN A0(IPB)=FP0(MIB) :A1(IPB)=FP1(MIB) :A2(IPB)=FP2(MIB)
1800 IF IP > 1 THEN 1830
1810 PANO=A1(1)*A1(1)-4*A2(1)*A0(1)
1812 IF PANO > 0 THEN 1830
1813 IF KE(2) < (KE(1)+5) THEN PRINT "Zero Surface Rate ! " :GOTO 2450
1815 FOR IDUM=0 TO (IDP-2)
1816     ICN=IDP-IDUM : KE(ICN+1)=KE(ICN)
1817 NEXT IDUM
1820 KE(2)=INT((KE(1)+KE(3))/2)
1822 IDP=IDP+1
1823 PRINT "..... "
1824 PRINT "mode ... automatic division "
```

```
1825 GOTO 2620
1830 JS=NTH(IPB-1) :JE=NTH(IPB)
1840 DEVS=0 :C2=A2(IPB) :C1=A1(IPB) :C0=A0(IPB)
1850 PRINT "C0,C1,C2  FROM ";JS ;:PRINT "  TO ";JE
1860 PRINT C0,C1,C2
1870 FOR IX=JS TO JE
1880    TIX=DT*IX          'etching starts from drop-point, nt(1)
1890    XTF(IX) = C0 + C1*TIX + C2*TIX*TIX
1900 IF XSKIP=1 THEN 1920
1910 IF XTF(IX) >= XR THEN TR=(IX-NT(1))*DT :XSKIP=1 :PRINT "TR =";TR
1920    DEVX=XT(IX)-XTF(IX) :DEVS=DEVS+DEVX*DEVX
1930 NEXT IX
1940 X(IPB)=XTF(JE)
1950 SDP=DEVS/(JE-JS+1) :PRINT "MSDP =";SDP
1960 IF SDP > SDMX THEN SDMX=SDP :IPBMX=IPB
1970 IP=IP+MIB
1980 GOTO 1470
1990 BEEP :PRINT " "
2000 PRINT "======== RATE PARAMETERS ========"
2010 FOR I=1 TO IPB
2020   PRINT NTH(I),:PRINT USING "  #.####^^^^  ";A0(I),A1(I),A2(I)
2030 NEXT I
2040 PRINT " "
2050 PRINT "Time for etching ";XR ;:PRINT " um = ";TR ;:PRINT " sec."
2060 PRINT "Maximum Deviation ";SDMX ;:PRINT "  at ";IPBMX
2070 PRINT "========= Plot Fitted Depth ========="
2080 XTPLOT=1 :XTFPLOT=1
2090 GOSUB 3310
2100 XTPLOT=0 :XTFPLOT=0
2110 BEEP
2120 RXPLOT=0 :XF=1 :RNORM=.1 :NORY=5
2130 PRINT "========= Rate vs Depth =========="
2140 X(0)=0 :XSUM=0
2150 FOR I=1 TO IPB
2160 BEEP
2170 IF XSUM > X(I) THEN 2270
2180 PAN=A1(I)*A1(I)-4*A2(I)*(A0(I)-XSUM) :IF PAN < 0 THEN 2250
2190 RATE=SQR(PAN) :IF XSUM=0 THEN RSUR=RATE
2200 IF RXPLOT=0 THEN 2240
2210    YR=199-NORY*RATE*20/RNORM
2220    MX=INT(500*XF*XSUM)+1
2230    PSET(MX,YR)
2240 IF RXPLOT=0 THEN PRINT USING " ##.######";RATE,XSUM
2250 XSUM=XSUM+.01 :IF XSUM > 1.25 THEN 2280
2260 GOTO 2170
2270 NEXT I
2280 RBLK=RATE :RATESET=1
2290 IF RXPLOT=1 THEN 2410
2300 RXPLOT=1 :CLS :KEY OFF :SCREEN 2
2310 PRINT "================="
2320 PRINT "= RATE vs DEPTH ="
2330 PRINT "================="
2340 PRINT "File Name = ";FILENUMBER$;:PRINT " ";RESIST$;
2341 PRINT " in ";DEVELOPER$;:PRINT " at ";DEVTEMP
2345 PRINT "DOSE = ";PW*TX
2350 PRINT "Surface Rate =";RSUR ;:PRINT "  Bulk Rate =";RBLK
2360 PRINT "Norm-Rate, Norm-Level ";:INPUT RNORM,NORY
2370 MGY=1 :NMGSET=1 :MGA=2
2380 GOSUB 4030
2390 NMGSET=0
2400 GOTO 2140
2410 PRINT "Plot again [y/n] ";:INPUT CFS$
2420 IF CFS$ = "y" THEN BEEP :GOTO 2300
2430 PRINT "Do you want to save this R vs X data [y/n] ";:INPUT SRX$
2440 IF SRX$ = "y" THEN 2960
2450 PRINT "Will you do it again with your own division [y/n] ";:INPUT QW$
```

```
2460 IF OW$ <> "y" THEN 2630
2470 ERASE XTF
2480 DIM XTF(2300)
2490 IF IP > 1 THEN 2520
2500 FOR I=1 TO IDP :PRINT I,KE(I) :NEXT I
2510 GOTO 2530
2520 FOR I=0 TO IPB :PRINT (I+1),NTH(I) :NEXT I
2530 PRINT "Enter the division. No entry means STOP THERE."
2540 PRINT "The drop point is ";KE(1)
2550 IO=2
2560 INPUT KE(IO) :IF KE(IO) = 0 THEN 2590
2570 IF KE(IO) < NT(1) THEN PRINT "entry is small. ENTER AGAIN ";:INPUT KE(IO)
2580 IO=IO+1 :BEEP :GOTO 2560
2590 IDP=IO-1
2600 IF KE(IDP) > NT(NF-1) THEN KE(IDP)=NT(NF-1) :PRINT "Last entry was cut."
2610 PRINT "MSD Compare Mode. Type 1 only for compare.  ";:INPUT MSDSKIP
2620 RATESET=0 :DAPLOT=0 :DAGPLOT=0 :GOTO 1430
2630 PRINT ">>>>>>> DATA GENERATING <<<<<<<"
2640 GOSUB 3110
2650 DAPLOT=1 :DAGPLOT=1 :PRINT "Plot Fitted-Data vs Time ......"
2660 GOSUB 3310
2670 DAPLOT=0 :DAGPLOT=0
2680 BEEP
2690 GOTO 2960
2700 REM ========== SUBROUTINE FUNCTION FIT ==============
2710 S1=0 :S2=0 :S3=0 :S4=0 :Z0=0 :Z1=0 :Z2=0
2720 IHOV1=INT((KE(IP)-KE(IP-1))/2) :IHOV2=INT((KE(IP+1)-KE(IP))/2)
2730 IQ1=KE(IP)-IHOV1 :IF IQ1 < KE(1) THEN IQ1=KE(1)
2740 T1=DT*IQ1 :IQS=IQ1
2750 FOR IA=1 TO KP
2760 IQE=KE(IP+IA)+IHOV2 :IF IQE > KE(IDP) THEN IQE=KE(IDP)
2770 FOR IQ=IQS TO IQE
2780 TQ=DT*IQ-T1
2790 S1=S1+TQ :S2=S2+TQ*TQ :S3=S3+TQ*TQ*TQ :S4=S4+TQ*TQ*TQ+TQ
2800 Y=XT(IQ)
2810 Z0=Z0+Y :Z1=Z1+Y*TQ :Z2=Z2+Y*TQ*TQ
2820 NEXT IQ
2830 NQ=IQE-IQ1+1 :IQS=IQE+1
2840 PRINT "# of sampling data = ";NQ
2850 DET=NQ*S2-S1*S1 :BEEP
2860 A0=(S2*Z0-S1*Z1)/DET :A1=(NQ*Z1-S1*Z0)/DET
2870 FL0(IA)=A0-A1*T1 :FL1(IA)=A1
2880 DET=NQ*S2*S4+2*S1*S2*S3-S2*S2*S2-S1*S1*S4-NQ*S3*S3 :BEEP
2890 A0=(Z0*S2*S4+S1*S3*Z2+S2*S3*Z1-S2*S2*Z2-S3*S3*Z0-S1*S4*Z1)/DET
2900 A1=(NQ*Z1*S4+Z0*S2*S3+S1*S2*Z2-S2*S2*Z1-Z0*S1*S4-NQ*S3*Z2)/DET
2910 A2=(NQ*S2*Z2+S1*S2*Z1+S1*Z0*S3-S2*S2*Z0-S1*S1*Z2-NQ*S3*Z1)/DET
2920 FP0(IA)=A0-A1*T1+A2*T1*T1 :FP1(IA)=A1-2*A2*T1 :FP2(IA)=A2 'time shift
2930 NEXT IA
2940 RETURN
2950 PRINT " "
2960 PRINT "want to save it on a or b ";:INPUT DISK$
2970 IF DISK$ <> "a" AND DISK$ <> "b" THEN 2960
2980 VRX$=DISK$+":"+FILENUMBER$+"RX"+"."+FILEEXTENTION$
2990 OPEN VRX$ FOR OUTPUT AS #1
3000 PRINT #1,RESIST$
3010 PRINT #1,DEVELOPER$
3020 PRINT #1,USING "####.#####";DEVTEMP
3030 PRINT #1,USING "####.#####";THICK
3040 PRINT #1,USING "####.#####";IPB,DT,PW,TX,REFL
3050 FOR I=1 TO IPB
3060 PRINT #1,USING "  #.####^^^^  ";X(I),A0(I),A1(I),A2(I)
3070 NEXT I
3080 CLOSE #1
3090 END
3100 REM   "=============================================="
3110 REM   "=          >>>> signal generating <<<<          ="
```

```
3120 REM   "=============================================="
3130 ERASE V
3140 DIM V(2000)
3150 IF VT(1) < VT(2) THEN PHIO=-PHIO
3160 MSD=O :JMSD=0
3170 FOR IP=1 TO NF-2
3180 BEEP
3190 VO=(VT(IP)+VT(IP+1))/2 :V1=ABS((VT(IP)-VT(IP+1))/2)
3200 IF IP=1 THEN VO=(VP1+VV1)/2 :V1=(VP1-VV1)/2
3210    FOR IT=NT(IP) TO NT(IP+1)
3220    DPS=XTF(IT) :IF IT > NTH(IPB) THEN 3280
3230    THETA=PHIO+PI*DPS/QUAT
3240    V(IT)=VO+V1*(COS(THETA))
3250    MSD=MSD+(V(IT)-H(IT))*(V(IT)-H(IT)) :JMSD=JMSD+1
3260    NEXT IT
3270 NEXT IP
3280 MSD=MSD/JMSD :PRINT "MSD total =";MSD
3290 RETURN
3300 REM ==================================================================
3310 REM                       PLOT simulated data
3320 REM ==================================================================
3330 MG=1 :MGY=1
3340 IF PRTP=1 THEN PRINT " # of data te be plotted   ";NPT :GOTO 3380
3350 PRINT "Total # of data ";N
3360 PRINT "plot-range    #,# ";:INPUT NS,NE :NPT=NE-NS+1
3370 NMGSET=0
3380 PRINT "******* PLOT MENUE *******"
3390 PRINT "X-mag ";:INPUT MG
3400 IF MG = 0 THEN PRINT "X-mag = 0 : not allowed !! ":GOTO 3390
3410 PRINT "Y-mag ";:INPUT MGY
3420 IF MGY <= 0 THEN PRINT "Y-mag > 0 !! ":GOTO 3410
3430 PRINT " YO   ";:INPUT YO
3440 IF MG < 0 THEN NMGSET=1 :MGA=ABS(MG)
3450 TS=NS*DT :TE=NE*DT
3460 CLS :KEY OFF :SCREEN 2
3470 REM =============== HEADING of PLOT ===============
3480 PRINT "File Name =";FILENUMBER$;:PRINT "   TR =";TR;:PRINT " sec."
3490 PRINT "Quarter Depth =";QUAT ;:PRINT "   Angle of Incidence =";ANGA
3500 IF RATESET <> 1 THEN 3520
3510 PRINT "Surface Rate =";RSUR ;:PRINT "  Bulk Rate =";RBLK :RATESET=0
3520 PRINT "Plot-Range =";NS ;:PRINT " - ";NE ;:PRINT "   dt =";DT;
3530 IF DAGPLOT=1 THEN PRINT "   MSD for all data =";MSD
3540 IF DAGPLOT <> 1 THEN PRINT " "
3550 PRINT "Plot begins at ";TS;:PRINT " , ends at ";TE;:PRINT " sec"
3560 MK=1
3570 IF MG <= 0 THEN MG=1:MK=MGA:NN=INT((NE-NS)/MGA+1) :GOTO 3590
3580 NN=MG*(NE-NS)+1
3590 GOSUB 4030
3600 IF XTFPLOT <> 1 THEN 3620
3610 REM FOR IP=0 TO IPB :PRINT IP+1,NTH(IP) :NEXT IP
3620 K=1
3630 FOR I=1  TO NN STEP MG
3635 IF (K+NS-1) < NT(1) THEN 3700
3640     DA=MGY*(H(K+NS-1)-YO)  :DAG=MGY*(V(K+NS-1)-YO)
3650     XT=MGY*(THICK-XT(K+NS-1)-YO) :XTF=MGY*(THICK-XTF(K+NS-1)-YO)
3660     IF DAPLOT =1 THEN  PSET(I-(I\639)*639,199-DA*20)
3670     IF XTPLOT =1 THEN  PSET(I-(I\639)*639,199-XT*20)
3680     IF XTFPLOT=1 THEN  PSET(I-(I\639)*639,199-XTF*20)
3690     IF DAGPLOT=1 THEN  PSET(I-(I\639)*639,199-DAG*20)
3700     IF I/639 = I\639 THEN 3960
3710 K=K+MK
3720 NEXT I
3730 IF NE > N THEN 3950
3740 PRINT "DATA-CHECK [y/n] ";:INPUT PR$
3750 IF PR$ <> "y" AND PR$ <> "n" THEN 3740
3760 IF PR$ = "n" THEN 3920
```

```
3770 PRINT "Beginning number ";NS
3780 NQ=INT((NE-NS+1)/7)
3790 NEE=7*NQ+NS-1
3800 NR=NE-NEE
3810 FOR I=NS TO NEE STEP 7
3820 PRINT USING "####.#####" ;V(I),V(I+1),V(I+2),V(I+3),V(I+4),V(I+5),V(I+6)
3830 NEXT I
3840 IF NR=0 THEN 3910
3850 I=0
3860 I=I+1
3870 PRINT USING "####.#####" ;V(NEE+I)
3880 IF I >= NR THEN 3900
3890 GOTO 3860
3900 PRINT "              "
3910 PRINT "Ending number ";NE
3920 PRINT "Want to see another section [y/n] ";:INPUT SEE$
3930 IF SEE$ <> "y" AND SEE$ <> "n" THEN 3920
3940 IF SEE$ = "y" THEN 3350
3950 GOTO 4020
3960 PRINT "Want to plot further [y/n] ";:INPUT YE$
3970 IF YE$ <> "y" AND YE$ <> "n" THEN 3960
3980 IF YE$ = "n" THEN 4020
3990 CLS
4000 GOSUB 4030
4010 GOTO 3720
4020 RETURN
4030 REM subroutine GRID
4040 SCALE = MG
4050 IF NMGSET=1 THEN SCALE = 1/MGA
4060 NGRID=XTIC*SCALE :LRESET=10*NGRID
4070 LTEN=-1 :LXTEN=1
4080 FOR L=1 TO 631 STEP NGRID
4090 PSET(L,199)
4100 LTEN=LTEN+1
4110 IF LTEN=LXTEN*10 THEN LXTEN=LXTEN+1 :PSET(L,198)
4120 NEXT L
4130 IF NORM$ = "y" THEN MGY=1
4140 NYSTP=YTIC*20*MGY :NYRESET=5*NYSTP
4150 FOR LY=0 TO 199 STEP NYSTP
4160 PSET(1,199-LY)
4170 IF LY/NYRESET = LY\NYRESET THEN PSET(2,199-LY)
4180 IF LY/2/NYRESET = LY\(2*NYRESET) THEN PSET(3,199-LY):PSET(4,199-LY)
4190 NEXT LY
4200 RETURN
```

Appendix C

Resist Parameter Table

TABLE I. R Parameters for MICROPOSIT 1470 at Various Prebake Temperatures
Prebaked for 20 min and developed in 351 1:5 at 20° C

R	80° C	100° C	75/100° C†	120° C
R_1 (μm/sec)	0.24	0.20	0.13	0.017
R_2 (μm/sec)	0.0005	0.0006	0.0006	0.0005
R_3	8.1	6.1	5.5	2.2
R_4 (μm)	0.24	0.14	0.13	0.08
R_5	0.76	0.40	0.17	0.05
R_6	0.55	0.28	0.26	0.11

† prebake at 75° C and post-exposure bake at 100° C

TABLE II. R Parameters for MICROPOSIT 1470 for Several Developers
Prebaked at 80° C for 20 min. and developed at 20° C

R	MICROPOSIT:H_2O 1:1	351:H_2O 1:5	351:H_2O 1:3	MF312:H_2O 1:1
R_1 (μm/sec)	0.27	0.24	0.44	0.31
R_2 (μm/sec)	0.0006	0.0005	0.0025	0.0033
R_3	7.4	8.1	6.2	7.2

TABLE III. A,B,C Exposure Parameters for MICROPOSIT 1470
at Various Prebake Temperatures
Exposed at 436 nm and prebaked for 20 min.

A,B,C	80° C	100° C	120° C
A (μm^{-1})	0.55	0.55	0.36
B (μm^{-1})	0.03	0.05	0.11
C (cm^2/mJ)	0.014	0.014	0.014

TABLE IV. A,B,C, and R Parameters for MICROPOSIT 1350J
Exposed at 436 nm and prebaked at 70° C for 60 min.
Developer MICROPOSIT 1:1 at 20° C

A,B,C		R_1, R_2, R_3	
A (μm^{-1})	0.55	R_1 (μm/sec)	0.27
B (μm^{-1})	0.045	R_2 (μm/sec)	0.0005
C (cm^2/mJ)	0.013	R_3	7.4

TABLE V. A B C Parameters for KODAK 820
Prebaked at 100° C for 30 min

A (μm^{-1})	B (μm^{-1})	C (cm^2/mJ)
0.51	0.031	0.013

TABLE VI. R Parameters for KODAK 820
Prebaked at 100° C for 30 min.

Developer	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8	R_9	R_{10}
1 KODAK 809 3 water at 20° C	0.23	0.0016	5.6	0.25	0.62	0.08	-	-	-	-
1 KODAK 932 1 water at 25° C	0.33	0.0015	10	0.25	0.45	0.02	0.13	0.55	0.02	0.62

R_1 and R_2 are in μm/sec
R_4 is in μm
R_3 and R_5-R_{10} are dimensionless

Appendix D

Example of Resist Profile Simulation (SAMPLE)

i. Input Example

```
trial 35 96 ;
linespace 1.0 1.0 ;
trial 22 1.0 .5 ;
run 1 ;
dose 90.0 ;
resmodel (.4358 ) (.5 .03 .0125)
                  (1.68 -.017  1.03762) ;
run 3 ;
devrate 2 0.23 0.0016 5.6 0.25 0.62 0.08 ;
trial 2 0 1 1 ;
trial 38 1 ;
devtime  15 75 5 ;
run 4 ;
```

```
1
1-----------------------------------------------------------------------------8
  *****                           SAMPLE                                *****
*****          Simulation and Modelling of Profiles in                    *****
*                       Lithography and Etching                               *
                          (ERL, EECS, UCB)

                   (Version  1.6        Jun 19, 1984)
               (VAX/UNIX version 0.0         Jun 19, 1984)
                        Wed Jul  4 23:08 10 1984
1-----------------------------------------------------------------------------8

  Input =trial 35 96 ;

  Input =linespace 1.0 1.0 ;

The mask is a grating with a periodic pattern of
     line/space     1.00000    1.00000 micrometers wide

  Input =trial 22 1 0 .5 ;

The window is  1.0 micrometers wide
     The mask edge is  0.5 micrometers from the left edge of the window

  Input =run 1 ;

Run the imaging subsystem to get
     the normalized horizontal energy distribution
     in the image of the mask resulting from
     a uniform illumination on the mask with a
     total of 1.0 mJ/cm2


***************
*  Run image  *
***************


                         Image parameter values


   Wavelength     Relative    Numerical   Aperture   Filling     Defocus
  (micrometers)   intensity   aperture     shape     factor   (micrometers)
     0.4358        1.0000      0.2800      circle     0.70        1.50


A periodic mask pattern with 1.0000 um wide lines and 1.0000 um wide spaces
Intensity window is  1.0000 um wide
Mask edge (L/S) is located at  0.5000 um from the left window boundary
    Parco2 used for partial coherent intensity computation

          Slope at the mask edge is          1.137 (1/um )
          Contrast by image min.max is       0.869
          Window contrast is                 0.869

1

  Input =dose 90 0 ;

Single exposure at the intensity of    90.00000 millijoules per sq  centimeter

  Input =resmodel ( 4358 ) ( .5 .03 .0125)

  Input =                  (1.68 -.017  1.03762) ;


At lambda =    0.43580 micrometers  the resist ABC parameters are
     A =    0.50000 (1/um),  B =    0.03000 (1/um),  C =    0.01250 (sq cm)/mJ
     the unexposed refractive index is (    1.68000,  -0.01700) and
     the thickness is     1.03762 micrometers.

  Input =run 3 ;

Find out the actual bleaching in the resist
```

```
****************
*  Run Expose  *
****************

Exposure parameters :

     Dose =  90.0 mJ/cm**2

Resist parameters :

     Wavelength um        A 1/um       B 1/um       C cm**2/mJ
        .4358             0.5000       0.0300         0.0125

Wafer parameters :

     Layer no.  1 is photoresist, and its extinction coefficient
          values, k, given below are at the start of expsosure.

     Layer no.  1     thickness = 1.0376 um
     Layer no.  2     thickness = 0.0741 um

     Wavelength:  .4358 um
     Vertical standing wave period in the photoresist is 0.1297 um

     Layer no.  1     index(n+ik) =  1.68  -0.018
     Layer no.  2     index(n+ik) =  1.47   0.
     Substrate        index(n+ik) =  4.73  -0.136

Intermediate results :

     Photoresist has  96 vertical and  49 horizontal grid divisions

        . Thickness of vertical   grid divisions is .0108 um
          Width     of horizontal grid divisions is .0204 um

Exposure results :

     Overall fractional power reflected

          Dose       (mJ/cm**2):      0.       7.1      14.2      21.3      28.4      35.6
          Wavelength .4358 um:   0.0070   0.0118   0.0177   0.0245   0.0319   0.0397

          Dose       (mJ/cm**2):     42.7     49.8      57.3      65.8      75.7      87.1
          Wavelength .4358 um:   0.0477   0.0557   0.0637   0.0723   0.0812   0.0900

          Dose       (mJ/cm**2):    100.1    115.2     132.4
          Wavelength .4358 um:   0.0981   0.1052   0.1111
1

   Input =devrate 2 0.23 0.0016 5.6 0.25 0.62 0.08 ;

 The development rate is given by an analytic function in M and z as.

   Rate(M,z) = f(M,z)*Rb(M)
         Where Rb(M) is bulk rate
               f(M,z) is rate-retardation factor near surface

     Rb(M) = 1.0/((1.0-M*exp(-R3*(1.0-M)))/R1 + M*exp(-R3*(1.0-M))/R2) um/sec
         Where  R1 = 0.23000 um/sec, R2 = 0.00160 um/sec, R3 =   5.60

     f(M,z) = 1-(1-(R5-(R5-R6)*M))*exp(-z/R4)
         Where  R4 = 0.25000 um, R5 = 0.62000, R6 = 0.08000

   Input =trial 2 0 1 1 ;

 this trial-stmt sets the flags
      idevfl(1)=   0.     idevfl(2)=   1.    idevfl(3)=   1

   Input =trial 38 1 ;

   Input =devtime  15 75 5 ;

 Develope the resist from   15.00000 to   75.00000 seconds in     5 steps

   Input =run 4 ;

 Find the developed profiles of the photoresist
```

```
*****************
*  Run develop  *
*****************
```

Parameter values

The first 3, 6, 8, or 10 parameters are used if all parameters must be positive and R5 to R10 must not be greater than one

R1 = 0 2300 um/sec   R2 = 0 0016 um/sec   R3 = 5 06

R4 = 0 2900 um   R5 = 0 6200   R6 = 0 0800

First development output = 15 0 sec
Time increment between profile outputs = 15 0 sec
Final development output = 75 0 sec
Maximum develop rate = 0 230000 um/sec , at m = 0
Extra string points requested
Initial development run

Background develop rate(bulk) = 0 00166 um/sec
m= 75   develop rate(bulk) = 0 00834 um/sec
m= 50   develop rate(bulk) = 0 04307 um/sec

The developer has broken through the resist in 42 9 seconds

---- Developed pattern ----

[illegible]	X	[illegible] Depth	Wave diff
15 0	0 4820	0 3935	0 6993
30 0	0 4893	0 7479	0 2792
45 0	0 2534	1 0901	-0 0566
60 0	-0 0992	1 0743	-0 0353
75 0	-0 2035	1 0738	-0 0349

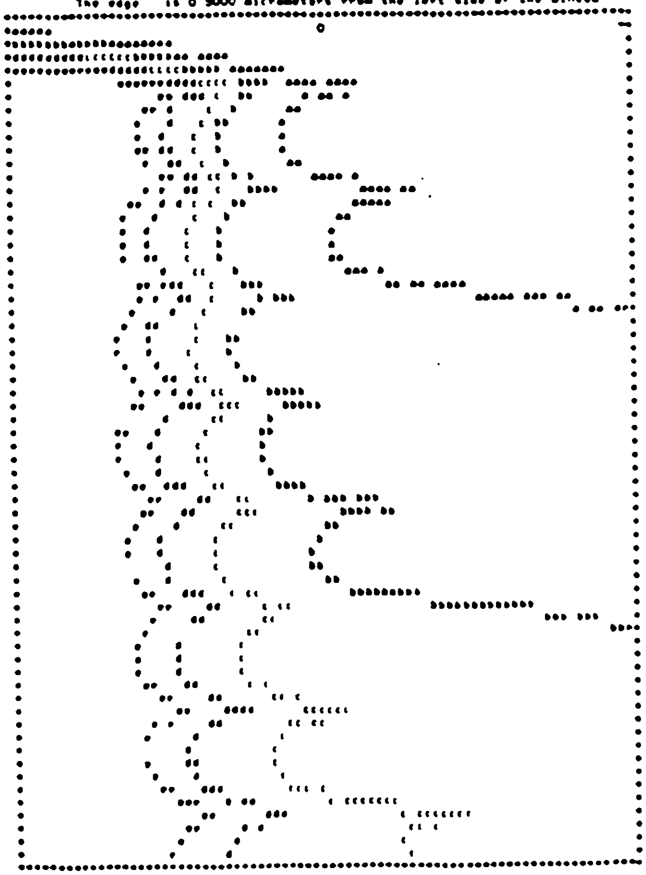
time	CD min	height	CD max	height	slope (deg)	CD at top	height
45 0	-0 0873	0 1311	0 2347	0 0512	-78 2	-0 2210	0 7709
60 0	-0 2291	0 1285	-0 0755	0 0536	-85 2	-0 2824	0 7692
75 0	-0 2980	0 1263	-0 1980	0 0550	-87 4	-0 3295	0 7812

CD = Critical Dimension i e the resist opening width
The slope above is computed using a CDmin in a range from 0 015 and 0 135 micrometers above the bottom of the resist and another CDmin in the range from 0 15 and 0 27 micrometers below the top of resist

x left = -0 5000 micrometers
x right = 0 5000 micrometers
z top = 0 micrometers
z bottom = 1 0376 micrometers

Symbol	time	resist-substrate intersection	sidewall angle estimate (by a straight line fit to all the CDmins)
a	15 0 sec		
b	30 0 sec		
c	45 0 sec	x = 0 1344 micrometers	82 9 degrees
d	60 0 sec	x = -0 1612 micrometers	-79 1 degrees
e	75 0 sec	x = -0 2606 micrometers	-83 3 degrees

The window is 1 0000 micrometers wide in x
The edge is 0 5000 micrometers from the left side of the window



Appendix E

The FORTRAN Program for the Design Algorithm

The FORTRAN code for the design algorithm has been added to the SAMPLE. The name of the subroutine is tral12. This also contains the subroutines RATEZX and DEVLOP. The following is the program codes as a part of mod02x2.f in the SAMPLE.

```
c
c       /* if argl = 12 call tral12 routine (added by deok kim, July,'84,
10      if(itest.ne.12) goto 15
        negin=0
        jdeok = nminst - 1
        if(jdeok .ge. 1) goto 11
        write(iprint,12)
        return
11      dostar=stnmls(2)
        ddos  =stnmls(3)
        devbrk=stnmls(4)
        resiw=stnmls(5)
        slope=stnmls(6)
        if(dostar .le. 0 ) negin=1
        if(ddos   .le. 0 ) negin=1
        if(devbrk .le. 0 ) negin=1
        if(slope  .le. 0 ) negin=1
        if(resiw  .le. 0 ) negin=1
        if(negin  .eq. 0)  goto 14
        write(iprint,13)
        return
12      format(/,49h***** At least one input must be specified *****,/)
13      format(/,39h***** All inputs must be positive *****,/)
14      continue
        call tral12(jdeok,dostar,ddos,devbrk,resiw,slope)
        return
c
```

```
c
      subroutine trall2(jdeok,dostar,ddos,devbrk,resiw,slope)
c
c     Program DESIGN F (Deok Kim, July, 1984)
c     contains subroutines RATEZX and DEVLOP, and
c     finds the process for a desired profile
c     by reading HURINT and RMZDOS tables of SAMPLE
c     and by calculating resit linewidth
c     and slope of the resist wall
c     with vaying dose
c
      dimension rr(500),pac(500),x(500)
      dimension rrh(101,50),mzlev(50),zt(50)
      dimension depth(50),xz(50),rps(20)
c
      common /comvar/ vardos,mzlev(50)
      common /objms2/ rlw,rsw,rlw2,rsw2
      common /cbwino/ window,edge,wndorg
      common /horimc/ deltx,mnhpts,nmhpts,horint(50)
      common /exptbl/ expos(21),rmzdos(501,21)
      common /simpar/ nprlyrs,nprpts,nendiv,deltm,deltz
      common /dvpar3/ inflor,kim,r1,r2,r3,r4,r5,r6,r7,r8,r9,r10
      common /rrpacx/ rr(500),pac(500),x(500),jxsum
      common /horizn/ nhflg,depth(50),zt(50)
      common /dvtiml/ jhordiv,xlim,time,xsum,devlim
      common /iol   / iterms,ibulk,iprout,ireswl,iin,iprint,ipunch
c
      if(ddos .lt. 0.2) ddos=10.0
      if(jdeok .le. 2) devbrk=100.0
      if(jdeok .le. 3) resiw=0.8*rlw
      if(jdeok .le. 4) slope=82.0
      resiwmax=rlw+rsw/2.0
      if(resiw .le. resiwmax) goto 11
      write(iprint,10)
   10 format(/54h***** resist linewidth on the wafer is too large *****,/)
      return
   11 if(slope .lt. 180.0) goto 13
      write(iprint,12)
   12 format(/51h***** maximum slope is limited to 180 degrees *****,/)
      return
   13 continue
c
      delslop=0.5
      slophi=slope+delslop
      sloplo=slope-delslop
c
      winox=wndorg+window
      resix=wndorg+resiw/2
      hormin=horint(1)
      hormax=horint(1)
      jmin=1
      jmax=1
      jhordiv=nmhpts
      do 17 i=1,jhordiv
      if (horint(i) .le. hormax) goto 16
      hormax=horint(i)
      jmax=i
   16 if (horint(i) .ge. hormin) goto 17
      hormin=horint(i)
      jmin=i
   17 continue
      hxmin=(jmin-1)*deltx
      hxmax=(jmax-1)*deltx
      devwid=abs(hxmax-hxmin)
```

```
      if (window .ge. devwid) goto 18
      write(iprint,21)
   18 continue
c
c     DETERMINE the direction of lateral develop
      latdevflg=0
      if (hxmax .gt. hxmin) latdevflg=-1
      if (hxmin .gt. hxmax) latdevflg=1
      if (latdevflg .ne. 0) goto 19
      write(iprint,21)
      goto 2000
   19 continue
   21 format('warning ... CHECK THE POSITION OF WINDOW')
      dirsign=-1.0
      if (latdevflg .gt. 0) dirsign=1.0
c
      dosmax=expos(1)
      do 20 i=1,nendiv
      if (expos(i) .gt. dosmax) dosmax=expos(i)
   20 continue
c
c     FIND THE Z-LEVEL FOR PEAK ENERGY COUPLED
      halfdiv=nendiv/2
      ntest=int(halfdiv)
      i=1
      mzlev(i)=2
      do 25 izpos=3,nprpts-1
      sigt1=rmzdos(izpos+1,ntest)-rmzdos(izpos,ntest)
      sigt0=rmzdos(izpos,ntest)-rmzdos(izpos-1,ntest)
      if ((sigt1 .le. 0.0) .or. (sigt0 .gt. 0.0)) goto 25
      i=i+1
      mzlev(i)=izpos
   25 continue
      ilast=i+1
      mzlev(ilast)=nprpts
      do 28 k=1,ilast
      depth(k)=deltz*(mzlev(k)-2)
c     write(iprint,119) depth(k),rmzdos(mzlev(k),ntest)
   28 continue
c 119 format(2f10.5)
c
      write(iprint,392)
  392 format(/'========== START SEARCHING ==========')
c
      doslim=500.0
      dose=dostar-ddos
  400 dose=dose+ddos
      if (dose .le. doslim) goto 405
      write(iprint,402) doslim
  402 format(/'Dose is limited arbitrarily at ',f10.3)
  405 write(iprint,407) dose
  407 format(/'******* INCIDENT DOSE (mJ/cm2) =',f7.3)
c
c     CALCULATES BREAKTHROUGH TIME
      nhflg=0
      vardos=dose*hormax
      if (hormax .le. 0.0) goto 1001
      if (vardos .gt. dosmax) goto 1000
      call ratezx
      devlim=devbrk
      call devlop
      if (xsum .ge. depth(ilast)) goto 33
      write(iprint,35) time
      goto 400
   33 tbreak=time
      write(iprint,34) tbreak
```

```
   34 format('break time =',f10.5)
   35 format('RESIST IS NOT BROKEN THROUGH IN TIME LIMIT =',f6.2)
c
c     CALCULATES DEVTIME FOR RESIST LINEWIDTH
      nhflg=1
      do 42 k=1,jhordiv
      vardos=dose*horint(k)
      call ratezx
      do 40 kv=1,ilast
      rrh(k,kv)=rr(kv)
   40 continue
   42 continue
      do 48 k=1,jhordiv
      kn=k
      if (latdevflg .gt. 0) goto 47
      kn=jhordiv-k+1
   47 rr(k)=rrh(kn,ilast)
      x(k)=(k-1)*deltx-wndorg
   48 continue
c     xlim is the accumulative development length
      xlim=abs(resix-hxmax)
      call devlop
      xz(ilast)=hxmax+dirsign*xsum
      tline=time+tbreak
      write(iprint,129) tline,resiw
  129 format(9hdevtime =,f6.2,2x,20h, resist linewidth =,f6.3)
c
c     CALCULATES RESIST TOP LOSS
      nhflg=-1
      vardos=dose*hormin
      call ratezx
      devlim=tline
      call devlop
      toploss=xsum
      write(iprint,139) toploss
  139 format('resist top loss =',f10.5)
c
c     CALCULATES LATERAL DEVELOPMENT LENGTH
c     now xlim is limited to window length
      xlim=window
      nhflg=1
      do 73 klat=1,ilast-1
      do 66 k=1,jhordiv
      kn=k
      if (latdevflg .gt. 0) goto 65
      kn=jhordiv-k+1
   65 rr(k)=rrh(kn,klat)
      x(k)=(k-1)*deltx-wndorg
   66 continue
c     now devlim is limited to the devtime for resist linewidth
      devlim=tline-zt(klat)
      call devlop
      xz(klat)=hxmax+dirsign*xsum
   73 continue
      write(iprint,145)
      write(iprint,146) wndorg,resix,windx
  145 format(/10h WNDORG(x),10h RESIST(x),10h WINDOW(x))
  146 format(3(f7.3,3x))
      write(iprint,148)
  148 format(/10h     Z(um),10h     X(um),10h    Z-TIME)
      xdmax=-1000.0
      do 83 k=1,ilast
      xdif=xz(ilast)-xz(k)
      if (xdif .le. xdmax) goto 82
      xdmax=xdif
      kmax=k
```

```
 82 if (depth(k) .le. toploss) ktop=k+1
    zdepth=-1*depth(k)
    write(iprint,150) zdepth,xz(k),zt(k)
 83 continue
150 format(3f10.5)
    xdiftop=xz(ilast)-xz(ktop)
    xdmax=xz(ilast)-xz(kmax)
    xdifmod=xz(ilast-1)-xz(ktop)
    if (xdiftop .eq. 0.) goto 400
    if (xdmax .eq. 0.) goto 400
    if (xdifmod .eq. 0.) goto 400
    slopetop=(depth(ilast)-depth(ktop))/xdiftop
    slopebot=(depth(ilast)-depth(kmax))/xdmax
    slopemod=(depth(ilast-1)-depth(ktop))/xdifmod
    angletop=atan(slopetop)*180/3.14
    if (angletop .lt. 0.) angletop=180+angletop
    anglebot=atan(slopebot)*180/3.14
    if (anglebot .lt. 0.) anglebot=180+anglebot
    anglemod=atan(slopemod)*180/3.14
    if (anglemod .lt. 0.) anglemod=180+anglemod
    sloptest=anglemod
    write(iprint,155)
155 format(/10h     ANGLE,10h      Z(A),10h      Z(B))
    zhigh=-1*depth(ktop)
    zlow=-1*depth(ilast)
    write(iprint,150) angletop,zhigh,zlow
    zhigh=-1*depth(kmax)
    zlow=-1*depth(ilast)
    write(iprint,150) anglebot,zhigh,zlow
    zhigh=-1*depth(ktop)
    zlow=-1*depth(ilast-1)
    write(iprint,150) anglemod,zhigh,zlow
    if (jdeok .eq. 1) goto 2000
    if (sloptest .ge. sloplo) goto 500
    goto 400
500 if ((sloptest .ge. sloplo) .and. (sloptest .le. slophi)) goto 550
    if (ddos .lt. 0.2) goto 1020
    dose=dose-ddos
    ddos=ddos/2.5
    goto 400
550 write(iprint,560) sloptest,resiw,dose,tline
560 format(//9hThe slope,1x,f6.2,1x,9h(degrees),1x,3hand,1x,
   * 16hresist linewidth,1x,f4.1,1x,4h(um),/
   * 12hare obtained,1x,
   * 9hwith dose,1x,f4.1,1x,8h(mJ/cm2),1x,
   * 11hand devtime,1x,f5.1,1x,5h(sec))
    goto 2000
1000 dosair=dosmax/hormax
    write(iprint,1002) dosair
1001 write(iprint,1004) hormax
    write(iprint,1006) dosmax
1002 format('LIMIT of Incident Dose =',f6.2)
1004 format('Horizontal Image Maximum =',f6.3)
1006 format('imum Dose in RMZDOS table =',f6.2)
1020 write(iprint,1022) slope,delslop,resiw
1022 format(/1x,18hNo match for slope,1x,f6.2,1x,2h+-,1x,f3.1,1x,
   *3hand,1x,16hresist linewidth,1x,f4.1)
2000 continue
    rps(1)=r1
    rps(2)=r2
    rps(3)=r3
    rps(4)=r4
    rps(5)=r5
    rps(6)=r6
    rps(7)=r7
    rps(8)=r8
```

```
      rps(9)=r9
      rps(10)=r10
      write(iprint,2010)
      write(iprint,2020) (rps(ik), ik=1,kim)
 2010 format(46hby the process for the following R parameters )
 2020 format(6(f8.5))
      return
      end
c
      subroutine ratezx
c     produces an array of rate vs z at any x
c
      common /comvar/ vardos,mzlev(50)
      common /exptbl/ expos(21),rmzdos(501,21)
      common /simpar/ nprlyr,nprpts,nendiv,deltm,deltz
      common /dvpar3/ inflgr,kim,r1,r2,r3,r4,r5,r6,r7,r8,r9,r10
      common /rrpacx/ rr(500),pac(500),x(500),jxsum
      common /horizn/ nhflg,depth(50),zt(50)
      common /io1   / iterm1,ibulk,iprout,ireswl,iin,iprint,ipunch
c
      do 205 jk=1,nendiv-1
      if((expos(jk).le.vardos).and.(vardos.le.expos(jk+1))) goto 205
  204 continue
  205 j=0
      kol=1
      do 250 i=2,nprpts
      zsum=deltz*(i-2)
      if (nhflg .ne. 1) goto 248
      if (i .ne. mzlev(kol)) goto 250
      kol=kol+1
  248 coff=(rmzdos(i,jk+1)-rmzdos(i,jk))/(expos(jk+1)-expos(jk))
      res=rmzdos(i,jk)+coff*(vardos-expos(jk))
      temp=res*exp(-r3*(1.0-res))
      rblk=1.0/((1.0-temp)/r1 + temp/r2)
      surf=1.0
      zsum=deltz*(i-2)
c
      rm=res
      zsum=r4
      sr0=r5
      sr1=r6
      sre1=r7
      sme1=r8
      sre2=r9
      sme2=r10
      kdeok=kim-3
      if (kdeok .lt. 1) goto 314
      if (kdeok .ne. 3) goto 308
      rom = sr0 - (sr0-sr1)*rm
      goto 313
  308 continue
      if (kdeok .gt. 6) goto 310
      if (rm .gt. sme1) goto 307
      rom = sr0 - (sr0-sre1)*rm/sme1
      goto 313
  307 rom = sre1 - (sre1-sr1)*(rm-sme1)/(1-sme1)
      goto 313
  310 continue
      if (rm .gt. sme1) goto 311
      rom = sr0 - (sr0-sre1)*rm/sme1
      goto 313
  311 continue
      if (sme1 .ge. sme2) goto 313
      if (rm .gt. sme2) goto 312
      rom = sre1 - (sre1-sre2)*(rm-sme1)/(sme2-sme1)
      goto 313
```

```
  312 continue
      rom = sre2 - (sre2-sr1)*(rm-sme2)/(1-sme2)
  313 continue
      surf = 1 - (1-rom)*exp(-xsum/zsur)
  314 rate=surf*rblk
      j=j+1
      rr(j)=rate
      pac(j)=res
      x(j)=xsum
      if (nhflg .eq. 1) goto 250
c     write(iprint,320) rate,res,xsum
  250 continue
c 320 format(3f10.5)
      jxsum=j
      return
      end
c
      subroutine devlop
c     generates depth vs time
c
      common /rrpacx/ rr(500),pac(500),x(500),jxsum
      common /horizn/ nhflg,depth(50),zt(50)
      common /dvtiml/ jhordiv,xlim,time,xsum,devlim
      common /iol   / itermi,ibulk,iprout,iresul,iin,iprint,ipunch
c
c     ndevt . increament of dev time dt=dx/rate/ndevt
      ndevt=20
      time=0.0
      xsum=x(1)
c     write(iprint,710) rr(1),xsum,time
      jend=jxsum-1
      if (nhflg .eq. 1) jend=jhordiv-1
      kal=1
      do 650 i=1,jend
      ratest=rr(i)
      if (rr(i+1) .gt. ratest ) ratest=rr(i+1)
      dt=(x(i+1)-x(i))/ratest/ndevt
  615 rate=rr(i)+(rr(i+1)-rr(i))*(xsum-x(i))/(x(i+1)-x(i))
      if (nhflg .ne. 0) goto 620
      if (xsum .lt. depth(kal)) goto 620
      zt(kal)=time-(xsum-depth(kal))/rate
      kal=kal+1
  620 if (time .ge. devlim) goto 700
      if (nhflg .ne. 1) goto 630
      if (xsum .ge. xlim) goto 705
  630 if (xsum .ge. x(i+1)) goto 640
      xsum=xsum+rate*dt
      time=time+dt
      goto 615
  640 time=time-(xsum-x(i+1))/rate
      xsum=x(i+1)
c     write(iprint,710) rr(i+1),xsum,time
  650 continue
      goto 750
  700 xsum=xsum-rate*(time-devlim)
      time=devlim
      goto 750
  705 time=time-(xsum-xlim)/rate
      xsum=xlim
  710 format(3f10.5)
  750 continue
      return
      end
c
```

i. Input Examples

In the first input example, the optical constants should be given to run the subroutines IMAGE and EXPOS of the SAMPLE as described in Chapter 5. Run 1 and run 3 provide the horizontal image and RMZDOS tables, respectively. A maximum dose should be given before run 3 in order to produce the RMZDOS table. The devrate 2 must be specified for the R parameters. The trial 12 statement executes the design algorithm. The first two variables are a starting dose and a dose increment, respectively. The routine calculates the slope for the given linewidth with varying dose. The third variable is the time limit for the resist break-through, because long development time for resist break-through is not desirable in practice. The fourth and the fifth variables are the resist linewidth on the wafer and the slope of the resist wall, respectively.

If only the first variable(dose) is given as shown in the second input example the slope is still calculated for the resist linewidth with 20% bias. If the resist is not broken-through within 100 seconds dose is automatically increased by 10 mJ per square centi-meter and the resist profile is calculated.

In the third input example a search of several processes for a desired feature is illustrated. A, B, and C parameters for different resists (or same resist but different bake condition) are followed by the corresponding R parameters. More than one process may be obtained for a desired profile.

Input Example 1

```
linespace 1 0 1.0 ;
trial 35 76 ;
run 1 ;
dose 200.0 ;

resmodel ( 4358 ) (.51 .031 .0125)
                  (1.68 -0.017  1.03762) ;
run 3 ;
devrate 2 0.23 0.0016 5.6 0.25 0.62 0.08 ;
trial 12 30.0 10.0 100.0 0.8 82.0 ;
```

Input Example 2

```
linespace 1 0 1.0 ;
trial 35 76 ;
run 1 ;
dose 200.0 ;


resmodel (.4358 ) (.51 .031 .0125)
                  (1.68 -0.017  1.03762) ;
run 3 ;
devrate 2 0.23 0.0016 5.6 0.25 0.62 0.08 ;
trial 12 90.0 ;
```

Input Example 3

```
linespace 1 0 1.0 ;
trial 35 76 ;
run 1 ;
dose 200.0 ;


resmodel (.4358 ) (.51 .031 .0125)
                  (1.68 -0.017  1.03762) ;
run 3 ;
devrate 2 0.23 0.0016 5.6 0.25 0.62 0.08 ;
trial 12 30.0 10.0 100.0 0.8 75.0 ;

devrate 2 0.33 0.0015 9.9 0.25 0.45 0.02 0.13 0.55 0.02 0.62 ;
trial 12 30.0 10.0 100.0 0.8 75.0 ;


resmodel (.4358) (.58 .03 .014)
                 (1.68 -0.017 1.03762) ;
run 3 ;
devrate 2 0.24 0.0005 8.1 ;
trial 12 30.0 10.0 100.0 0.8 75.0 ;

devrate 2 0.27 0.0006 7.4 ;
trial 12 30.0 10.0 100.0 0.8 75.0 ;

devrate 2 0.44 0.0025 8.2 ;
trial 12 30.0 10.0 100.0 0.8 75.0 ;

devrate 2 0.31 0.0034 7.2 ;
trial 12 30.0 10.0 100.0 0.8 75.0 ;
```

ii. Output Examples

In the output examples the ordinary outputs for image and exposure are not shown and only the output for the trial 12 statement is shown. WNDORG, RESIST, and WINDOW are the origin of the window, the coordinate of resist linewidth, and the coordinate of the window, respectively. Z, X, and Z-TIME are the maximum energy coupled position in the Z direction, the position of the lateral development, and the break-through time with depth for the maximum horizontal intensity. The angles are evaluated for three different sets of depths. The angle for the search is the last one as defined in Chapter 5. The top of the resist is still shown but, in fact, the resist top is lost. The resist top loss is also shown.

```
   Input =trial 12 30.0 10.0 100.0 0.8 82 0 ;

========== START SEARCHING ==========

******* INCIDENT DOSE (mJ/cm2) = 30.000
RESIST IS NOT BROKEN THROUGH IN TIME LIMIT =100.00

******* INCIDENT DOSE (mJ/cm2) = 40.000
RESIST IS NOT BROKEN THROUGH IN TIME LIMIT =100.00

******* INCIDENT DOSE (mJ/cm2) = 50.000
RESIST IS NOT BROKEN THROUGH IN TIME LIMIT =100.00

******* INCIDENT DOSE (mJ/cm2) = 60.000
break time =  78.55730
devtime =100.29  ; resist linewidth = 0.800
resist top loss =   0.04003

 WNDORG(x) RESIST(x) WINDOW(x)
  0          0.400      1.000

     Z(um)        X(um)      Z-TIME
   0.           0.31790     0
  -0.12973      0.22423    12.67510
  -0.25940      0.19673    21.72366
  -0.38911      0.19751    30.13230
  -0.51881      0.21400    39.62886
  -0.64851      0.24133    47.56185
  -0.77821      0.27874    57.13709
  -0.90792      0.32835    67.47599
  -1.02681      0.40000    78.55730

     ANGLE        Z(A)        Z(B)
  78.95437   -0.12970    -1.02681
  75.21965   -0.25940    -1.02681
  82.42017   -0.12970    -0.90792


 The slope  82.42 (degrees) and resist linewidth  0.8 (um) are obtained
with dose 60.0 (mJ/cm2) and devtime 100.3 (sec)
by the process for the following R parameters:
 0.23000 0.00160 5.60000 0.25000 0.62000 0.08000

 **********      End of lab session      **********
 ---------------------------------------------------
 Exec times   13.500u,    3.183s seconds     01:16:31
```

```
   Input =trial 12 90.0 ,

=========== START SEARCHING ==========

******* INCIDENT DOSE (mJ/cm2) = 90 000
break time =  40 58775
devtime = 50 14  ; resist linewidth = 0 800
resist top loss =   0 02605

 WNDORG(x) RESIST(x) WINDOW(x)
  0          0 400      1 000

     Z(um)       X(um)      Z-TIME
   0           0 29003     0
  -0 12973     0 22600     5 85598
  -0 25943     0 20839    10 36309
  -0 38911     0 21137    14 63603
  -0 51881     0 22667    19 03791
  -0 64851     0 25013    23 73862
  -0 77821     0 28251    28 85426
  -0 90792     0 32765    34 46778
  -1 02681     0 40000    40 58995

     ANGLE         Z(A)       Z(B)
  79 06682   -0 12970   -1 02681
  76 04797   -0 25940   -1 02681
  82 60331   -0 12970   -0 90792
by the process for the following R parameters
 0 23000 0 00160 5 60000 0 25000 0 62000 0 08000

 **********      End of lab session      **********
 ------------------------------------------------------
 Exec time:   12 6170,     2.517s seconds      01 20 16
```