SIMULATION OF DEVELOPED RESIST PROFILES FOR MASKED ION BEAM LITHOGRAPHY (MIBL)

by

G. M. Atkinson

Memorandum No. UCB/ERL M85/51

21 June 1985

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

SIMULATION OF DEVELOPED RESIST PROFILES FOR MASKED ION BEAM LITHOGRAPHY (MIBL)

Gary M. Atkinson

Department of Electrical Engineering and Computer Sciences
and the Electronics Research Laboratory
University of California Berkeley, California 94720

ABSTRACT

A Masked Ion Beam Lithography (MIBL) simulator has been developed as part of the SAMPLE process simulation program at the University of California, Berkeley. The program uses analytical models of the mask scattering effects. The modeling approach is discussed and complete software documentation is included.

(Version 1.1 April 7, 1985)

Acknowledgments

The author would like to thank the entire SAMPLE group at the University of California, Berkeley. In particular, I would like to mention the thoughtful guidance of Dr. Andrew R. Neureuther and Dr. Nathan W. Cheung. I am grateful to both Charles W. Slayman and John L. Bartelt of Hughes Research Laboratory for their financial intellectual and moral support. Also the generous help from Dr. Sharad N. Nandgaonkar, Dr. Michael G. Rosenfield, Gino Addiego, Albert J.D. Chen, Wing-Yu Leung and the late John L. Reynolds. All these people made this simulation effort a lot easier. Finally I would like to thank my industrial sponsors which include AT&T Bell Laboratories, IBM and Hughes Aircraft Co.. This research was partially funded through the MICRO program (grant #532411-19900) and an IBM fellowship.

1. PHYSICS AND MODELING

1.1. *Introduction*

Masked ion beam lithography is a relatively new lithography technique that attempts to achieve both submicrometer resolution and high throughput. The basic concept is illustrated in Fig. 1. A collimated ion beam passes through a partially transmitting mask and then exposes the resist. The limited extent of the scattering of the ions in the mask and resist means that submicrometer pattern definition can be achieved. In order to study this technique a new simulation program was developed.

The simulation can be broken down into two parts, the exposure and the development. The result of the exposure is that energy is deposited in the resist. This distribution is described by a two dimensional array (elin(82,1002). This array is then fed to the ion beam developer which develops the resist and outputs the resist contours.

The physical processes that take place in the ion mask are indeed very complex and very difficult to model in a general fashion. To make this problem tractable a two-fold approach has been taken. The program can run in two modes. In the simplest mode the program makes all the necessary scattering calculations. In this mode there are many assumptions and approximations (These will be discussed later in further detail). In the second mode the user inputs the scattering parameters himself and the approximations are entirely up to the user. This allows the user to simulate MIBL exposures in the regions where the programs scattering approximations may not be valid. The input format also allows the user to mix and match these two modes, using the program model for one variable and his own values for another.

The program models used are based on three basic assumptions. First of all, it is assumed that a hydrogen beam is used. Secondly, a <100> channeling silicon membrane technology is assumed as the support structure of the mask. The last assumption is that the absorber patterns on the membrane are either tungsten, gold or silicon. The reasons underlying this set of assumptions are twofold. The modeling of the physical processes is much simpler with these assumptions and the primary interest of this author is in a hydrogen ion/silicon channeling mask technology. As mentioned previously, the input structure allows the program models to be bypassed so that any ion beam and/or mask set-up can be used. The following description of the simulation models is based on the above three assumptions.

1.2. *Modeling*

In order to model the resist exposure by the scattered beam (ie. after passing through the mask), the scattered beam is divided into three components as illustrated in Fig. 2. These are the channeled, dechanneled and background components. The exposure problem then is reduced to determining the individual dose, energy and angular divergence of each beam component.

The dose, energy and angular divergence of the dechanneled and background beams can be readily calculated since they correspond to ion interactions with amorphous materials. For the energy loss and dose calculations of these two components the data and models compiled by Andersen and Ziegler are used.[6] The angular scattering calculations follow that of Meyer.[5]

For the channeled component of the beam, there exist no simple empirical or analytical models. Therefore, in order to model the channeled beam component three assumptions were made. The first is that energy loss of channeled ions is approximately 65% of the nonchanneled ion loss (due to their localization to regions of low electron density). Secondly, it is assumed that a delta function input beam along a channeling direction yields a gaussian shaped beam upon exit of the crystal, as illustrated in Fig. 3. It is also assumed that the $\psi_{1/2}$ of this gaussian impulse response scales with the critical angle as developed by Lindhard :[3]

$$\psi_{1/2} = K\psi_{crit} . \tag{1}$$

A semi-empirical value can be found for K by calculating the critical angle and measuring $\psi_{1/2}$ experimentally. $\psi_{1/2}$ for a 200 keV hydrogen beam passing through a 0.7 μm <100> silicon membrane has been measured to be about 0.4°.[19] This corresponds to an exit energy of $\approx .135 keV$. The scaling factor is then:

$$K = \frac{\left[\psi_{1/2}\right]_{135keV}}{\left[\psi_{crit}\right]_{135keV}} = 0.243^\circ . \tag{2}$$

Assuming K is constant with respect to energy, the $\psi_{1/2}$ of a beam of arbitrary energy is determined as follows: First the energy lossed in the support membrane is calculated. Then the critical angle corresponding to the exit energy is evaluated. Then $\psi_{1/2}$ is given by Eqn. (1). Using this approach, a complete picture of the beam after the mask can be calculated readily from the beam parameters and mask geometry. In mathematical terms: The gaussian delta function response is convolved with the absorber pattern (a series of step functions) which gives an error function distribution which represents the lateral spreading of the ions after passing through the mask. For ease of calculation, each beam component is calculated separately. The three beam components for two 1 μm spaces separated by a 1μm line are shown in Fig. 4. Here the plots have been exaggerated to illustrate all three beam components A flowchart of this calculation is shown in Fig. 5.

Once the lateral distribution, energy and dose of each of the beam components is known, the resist can be exposed. For each component the energy loss data of Andersen and Ziegler gives the amount of energy deposited in the resist.[8] Each component is calculated individually, the extent of its exposure being determined by its lateral distribution, energy and dose. The components are then summed together to form the total exposure (which is represented by a 2-D array, elin(82,1002) in Joules/cm**2).

The development is then done by the ion beam developer which is a modified version of the e-beam developer (which is a modified version of the optical developer). Basically the development uses a string algorithm with the advancement of the string points based on the formula:

$$R(D) = R_1\left[c_m + \frac{D}{D_o}\right]^a \tag{3}$$

For more information on the development code, the user is referred to the e-beam and optical lithography documentation of the SAMPLE program.[14,15]

1.3. *Tapered Absorber Modeling*

The modeling of a tapered absorber edge is accomplished using a staircase step approximation (Fig. 6). The user specifies how many steps are used. The program then calculates the dose that gets through and the lateral spreading of each of the steps. However, it would be quite involved to calculate the transmitted energy of each of the steps so an approximate method is used. The energy of transmission is calculated at three equidistant points along the taper. The energy of the ions exiting each third of the taper is assumed to be constant at the value for the middle of that section.

1.4. *Damage Etching Option*

The Ion Beam Machine has been modified in order to investigate the effects of ion beam induced damage on the etching of silicon dioxide. For this case the development assumes that SiO2 is the resist and that its development rate is given by an equation of the form:

$$1/R = U/R_1+(1-U)/R_2 \qquad 1$$

where,

$$U = exp(E/E_0) \qquad 2$$

Here R is the development rate (micrometers/sec), E is the energy deposited in $joules/cm^3$, E_0 is a characteristic energy, R_1 is the undamaged etch rate and R_2 is the maximum damage etch rate. E_0 is approximately the point where $R = 2 \times R_1$.

This model is a first pass approximation that is intended to be fitted by the user. The basic form of the model follows the experimental results of Monfret and Bernard.[16]

1.5. *Additional Approximations and Limitations*

In addition to the approximations already discussed there are some which haven't been mentioned. In order to be complete they are listed here:

1) No thermal distortion of the mask is taken into account.
2) Each beam component is assumed to be monoenergetic.
3) The beam is assumed to be collimated at the surface of the resist.
4) The channeled and dechanneled components of the beam are lumped together when they pass through the absorbers.
5) The spreading of the beam is assumed to increase linearly with depth in the resist.
6) The number of ions dechanneled when the beam misalignment equals the critical angle is assumed to be 50%.
7) The ions escaping through the sides of the absorbers are ignored.

Finally, the string development algorithms have some difficulty when the mask scattering ($\psi_{1/2}$) is reduced to zero. Therefore, open stencil masks can only be simulated if the beam has a finite ψ/χ.

REFERENCES

1 J.L. Bartelt, C.W. Slayman, J.E. Wood, J.Y. Chen, C.M. McKenna, C.P. Minning, J.F. Coakley, R.E. Holman, and C.M. Perrygo J. Vac. Sci. Technol. **19**, 1166, (1981).

2 D.V. Morgan, *Channeling: Theory, Observation and Applications* (Wiley-Interscience, New York, 1973).

3 J. Lindhard, Mat. Fys. Medd. Dan. Vid. Selsk. **34**, 3, (1965).

5 L. Meyer, Phys. Stat. Sol. **44**, 253 (1971).

6 J.N. Randall and J.C. Wolfe, Appl. Phys. Lett. **39**, 742, (1981).

7 J.N. Randall and J.C. Wolfe, Appl. Phys. Lett. **41**, 247, (1982).

8 H.H. Andersen and J.F. Ziegler, *Hydrogen Stopping Powers and Ranges in All Elements*, (Pergamon, New York, 1977), vol. 3.

10 L. Karapiperis, I. Adesida, C.A. Lee, and E.D. Wolf, J. Vac. Sci. Technol., **19**, 1259 (1981).

11 J.E. Jensen and C.W. Slayman, unpublished.

12 R.G. Wilson, H.L. Dunlap, D.M. Jamba and D.R. Meyers, *Semiconductor Measurement Technology:* Angular Sensitivity of Controlled Implanted Doping Profiles, (U.S. Government Printing Office, 1978).

13 C.W. Slayman, J.L. Bartelt, C.M. McKenna, and J.Y. Chen, Opt. Eng. **22**, 208, (1983).

14 M.G. Rosenfield, Ph.D. Thesis, Department of Electrical Engineering and Computer Science, University of California, Berkeley, 1984.

15 M.M. O'Toole, Ph.D. Thesis, Department of Electrical Engineering and Computer Science, University of California, Berkeley, 1979.

16 G.M. Atkinson and A.R. Neureuther, J. Vac. Sci. Technol , Jan/Feb (1985).

17 G.M. Atkinson and N.W. Cheung, Nucl. Inst. Meth., Jan/Feb (1985).

18 A. Monfret and J. Bernard, Proc. Int. Conf. Ion Imp. Semi., Bavaria, Germany, 1971.

FIGURE CAPTIONS

Fig. 1. The basic MIBL concept.

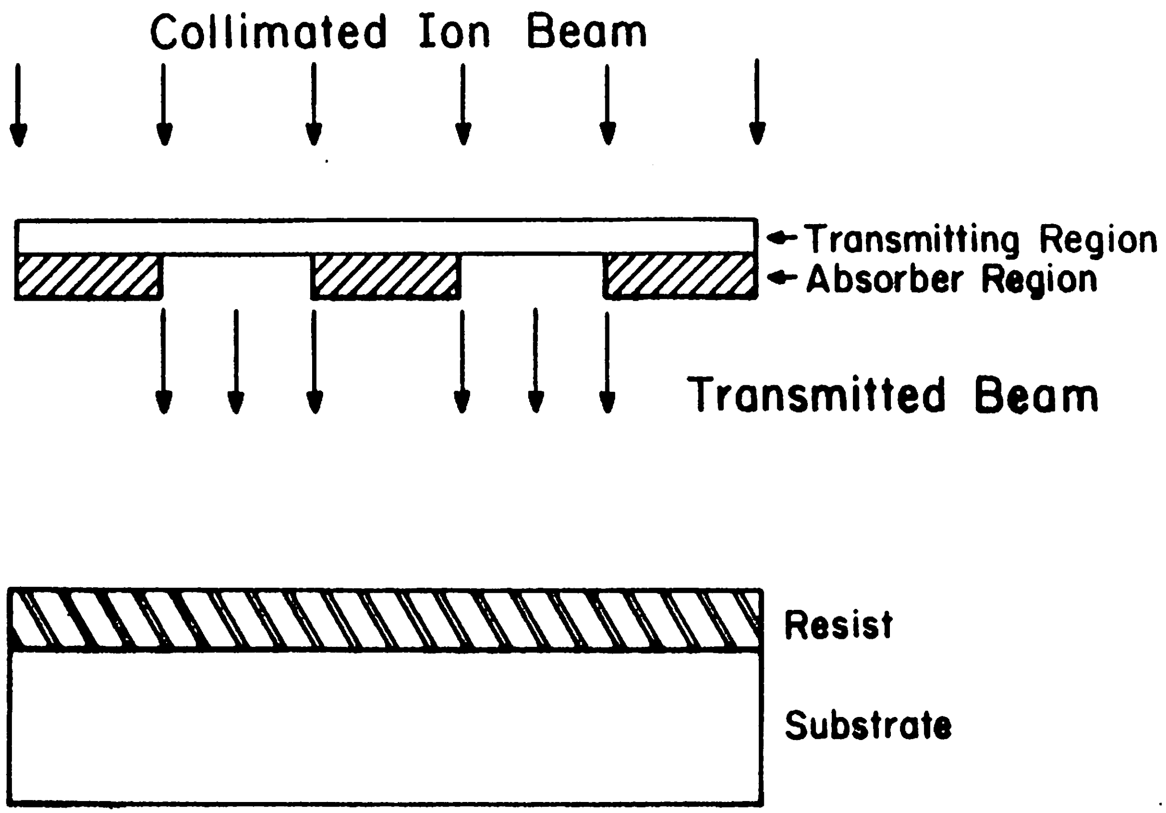
Fig. 2. The scattering effects in the MIBL exposure.

Fig. 3. Delta function input beam response.

Fig. 4. The three beam components after passing through the mask: a) the channeled beam, b) the background beam and c) the dechanneled beam (exaggerated for illustration).

Fig. 5. Flowchart of MIBL simulator.

Fig. 6. Tapered absorber approximation.



Collimated Ion Beam

Transmitting Region

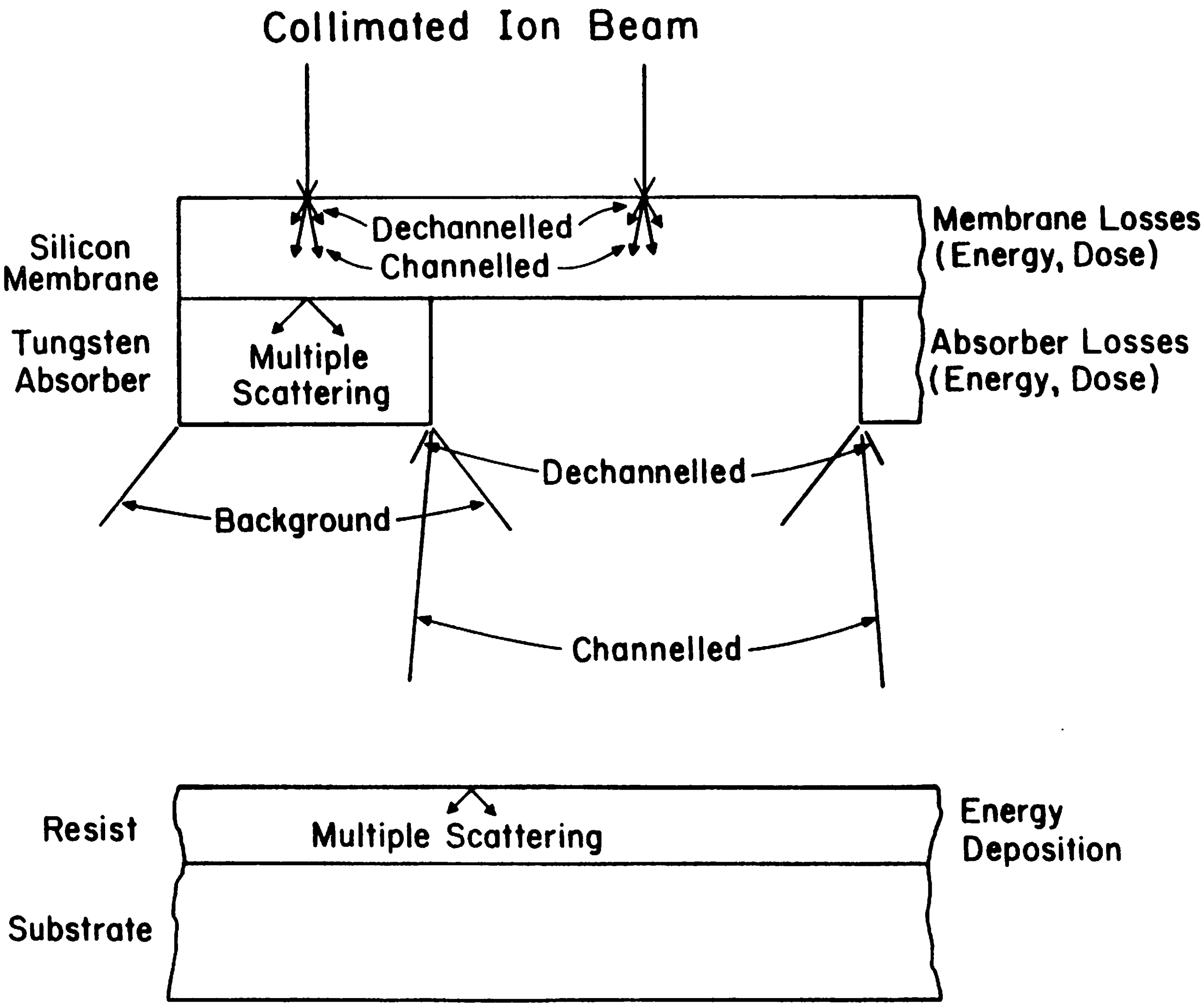
Absorber Region

Transmitted Beam

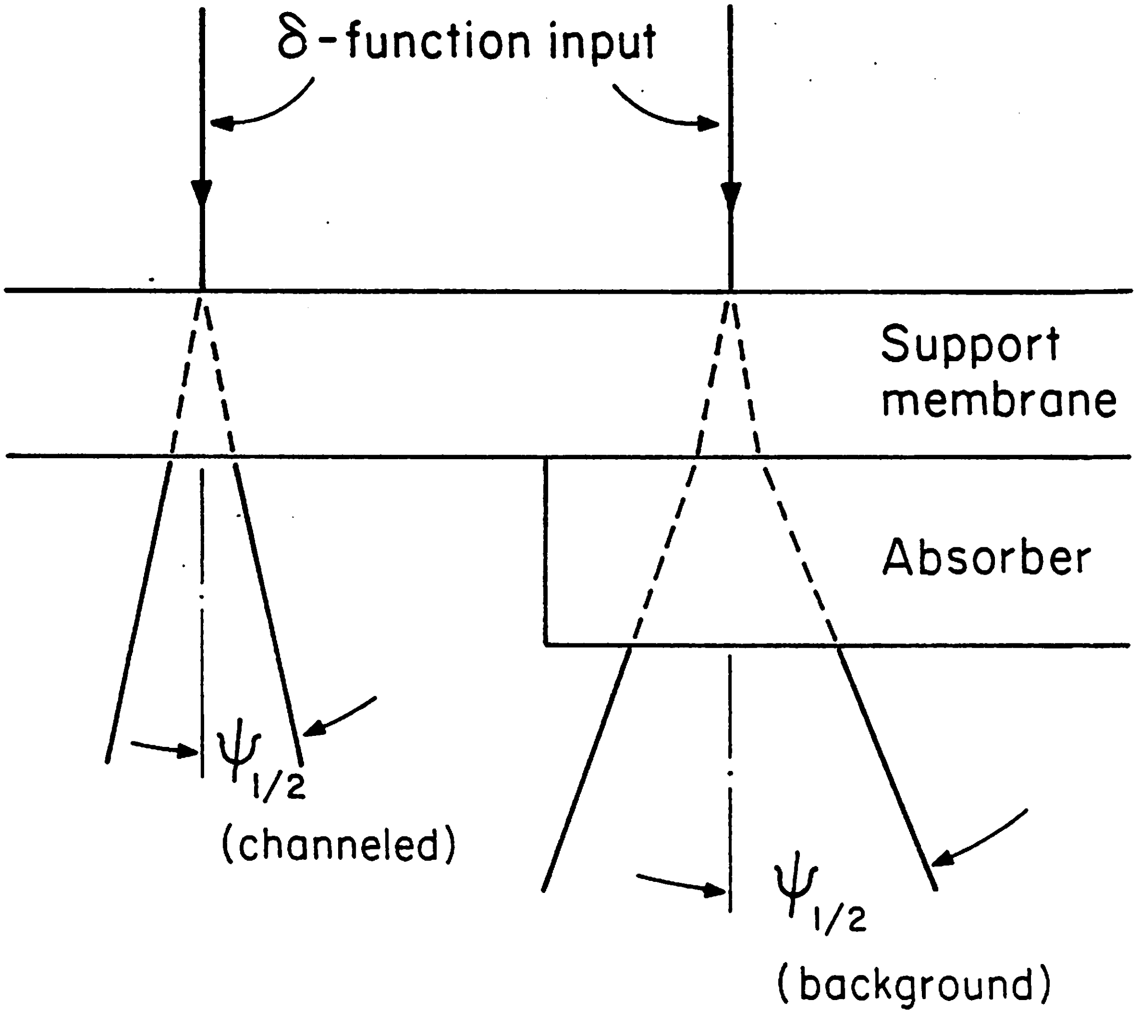
Resist

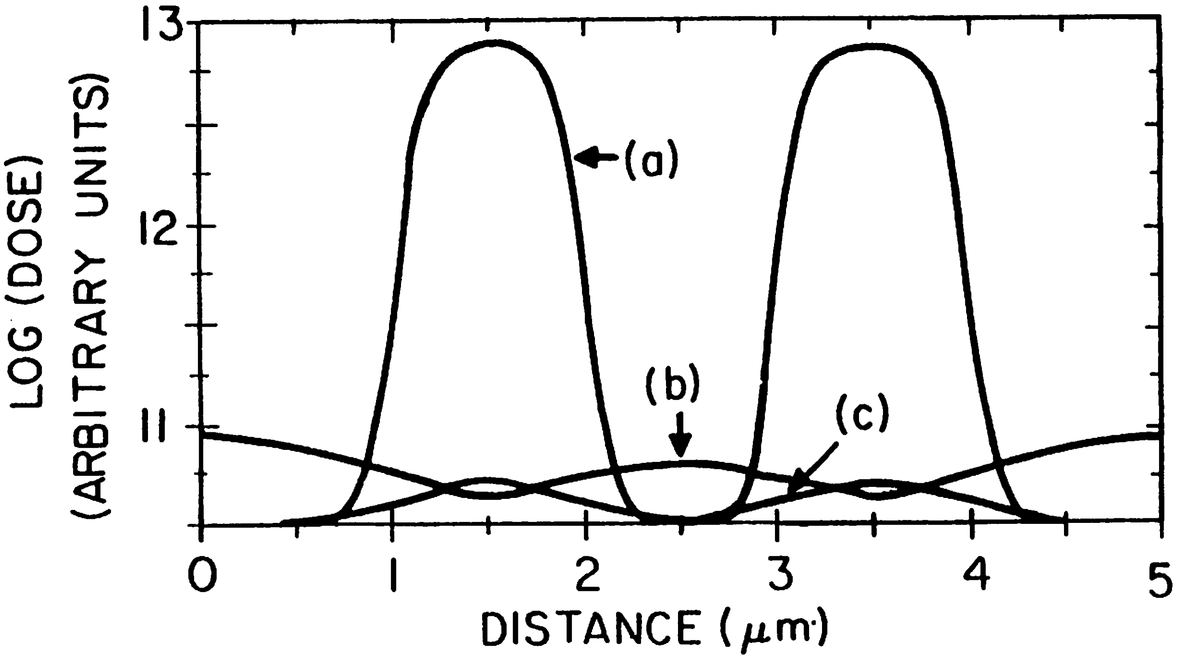
Substrate

Masked Ion Beam Lithography

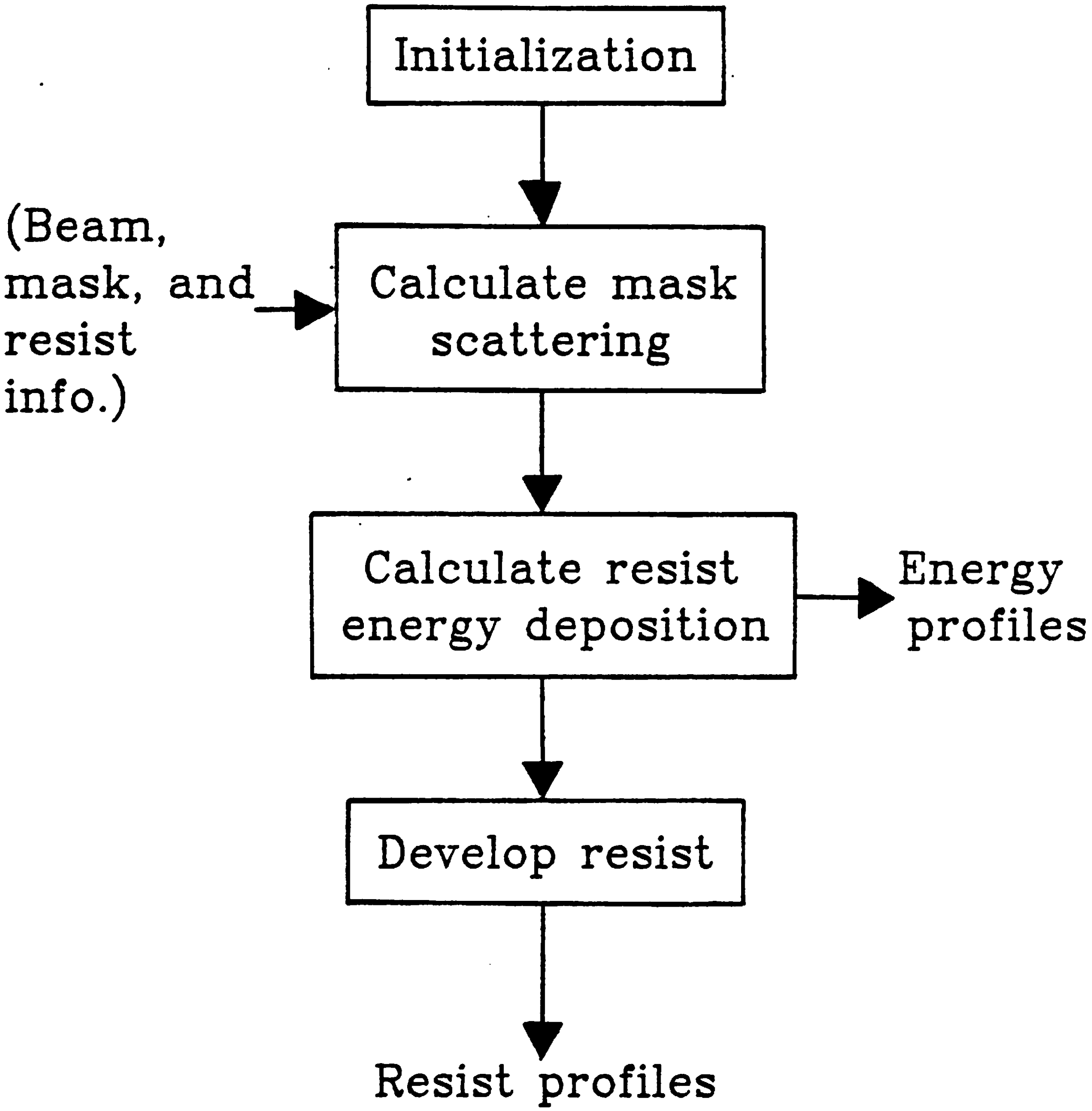


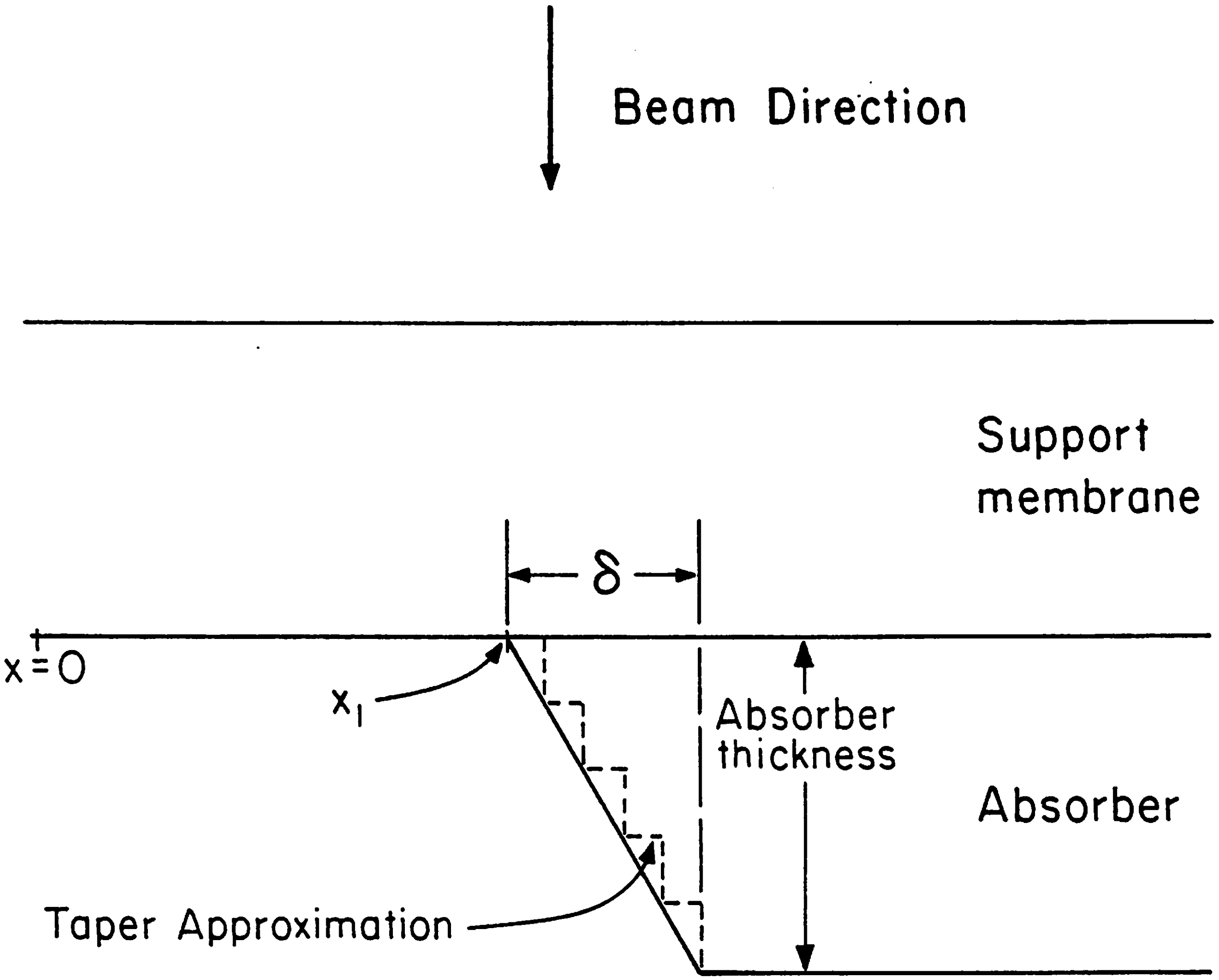
Scattering Effects


δ-function input
Support membrane
Absorber
$\psi_{1/2}$
(channeled)
$\psi_{1/2}$
(background)


13
12
11
LOG (DOSE)
(ARBITRARY UNITS)
(a)
(b)
(c)
0
1
2
3
4
5
DISTANCE (μm)

Flowchart


Initialization
(Beam, mask, and resist info.)
Calculate mask scattering
Calculate resist energy deposition
Energy profiles
Develop resist
Resist profiles


Beam Direction
Support membrane
δ
x = 0
x_1
Absorber thickness
Absorber
Taper Approximation

2. USER'S GUIDE

2.1. *Introduction*

A MIBL (Masked Ion Beam Lithography) machine has been incorporated into the SAMPLE (Simulation and Modeling of Profiles in Lithography and Etching) computer program. This ion beam machine incorporates the ion beam exposure and time evolution development of resist exposed in the MIBL format. The program assumes a hydrogen beam and a <100> silicon channeling mask. The absorber patterns are either tungsten, gold or silicon. The resist is specified by an etch rate vs. dose curve. The ion beam machine uses the beam parameters and the mask geometry to form an image of the beam after passing through the mask. This image exposes the resist which is then developed using the SAMPLE string algorithm. The input is through a series of keyword statements which also allow the user to bypass the program models and input the scattering parameters that are used to form the post mask beam image. In this way other ion beams and masking technologies can be simulated.

2.2. *Program Flow*

The program begins by initializing itself to a set of default parameters. The user's input is then read in via the keyword statements in the following manner: The user inputs the beam parameters and the mask geometry (Ionbeam and ionmask). At this point the user can specify that either the program will calculate the scattering parameters or the user inputs them (Ionscat). Either way, the scattering parameters are determined. Next the resist thickness and window of interest are specified (Ionreswin). Then the energy deposition in the resist is either input by the user or calculated by the program (Ionedep). Now the ion beam calculates the exposure of the resist (Ionexpose). After exposure, the development rate parameters are input for the resist (Ionresist). Next the resist is developed and the devlopment contours are output in a file called f77punch7 (Iondevlp). Finally the energy deposition contours in the resist (the exposure) can be printed into a file called engpts (Ionecntr).

2.3. *Keywords and Trial Statements for the Masked Ion Beam Machine*

ionprint ipflgs(1)...ipflgs(8) ; (trial 301)

Ionprint inputs the printing flags for arrays and information. There are a total of 7 flags which control the printing as follows: Print energy deposition arrays (ipflgs(1)=1)), Print lateral ion distribution at resist surface (ipflgs(2)=1), Print axial energy depositon arrays (ipflgs(3)=1), Print distance calculation parameters (ipflgs(4)=1), Print mask geometry (ipflgs(5)=1), Print mask scattering data (ipflgs(6)=1), Print resist data (ipflgs(7) =1) and Print range data (ipflgs(8)=1). This statement is optional. The default case prints all the scattering parameters but not the arrays used in the calculations.

ionbeam itype,e0,dose(x1013), bangle ; (trial 303)**

Ionbeam inputs the beam parameters. These are the ion type (itype), the initial beam energy (e0,keV), the dose (which is multiplied be a factor of 1e13 after it is input, 1/cm**2) and the beam angle (bangle,degrees). The beam angle is measured relative to the line perpendicular to the mask/wafer plane. This statement is optional. The default case is itype=1 (H+, which is all that is implemented so far), e0=200 keV, dose=1.3e13, and bangle=0. To keep the default value simply enter -1 for that value.

ionmask spce,absthk,delta,supthk,xray(1)...xray(9) ; (trial 304)

Ionmask sets the geometry of the exposure mask. All units are in micrometers. This consists of the mask/resist spacing (spce), the absorbing layer thickness (absthk), the tapered edge length (for tapered absorber runs, delta equals the distance it takes the absorber to go from zero thickness to its maximum thickness.) the membrane thickness (supthk), and the absorber positions. (xray) xray(1) indicates the x posistion of the first rising edge of the absorber pattern, xray(2) the next falling edge etc... This statement is optional. The defaults are spce=25 micrometers, absthk=.85 micrometers, delta=0, supthk=.85 micrometers and xray is set for periodic 1.0 micrometer lines and spaces. To keep the default value simply enter -1 for that value. Note that if the first absorber rising edge is specified at 0, the program assumes the absorber extends to negative infinity. Also the absorbers can not overlap|

ionscat abstyp,dele,psihlf,dosthr,cntrst,psibak,delew; (trial 305)

Ionscat sets mask scattering parameters. These are the abscrber type (abstype=74 for tungsten or gold, 14 for amorphous silicon absorbers and 15 for channeling silicon absorbers), the energy loss in the support membrane (dele,keV), the half angle of the angular distribution of the ions exiting the support membrane (psihlf,degrees), the ion dose that makes it through the support layer to expose the resist (dosthr, 1e13/cm**2), the log of the ratio of the exposing dose to the dose leaking through the absorber regions, ie. the mask contrast (cntrst), the half angle of the background angular ion distribution exiting through the absorber regions (psibak,degrees) and the energy loss in the absorbers (delew,keV). Note that psibak includes the effect of the beam spreading in the support membrane (psihlf) since the ions pass through the membrane first. This statement is optional. The default values correspond to the beam parameters and mask geometry specified above in ionbeam and ionmask. To use a default value simply enter -1 for that value. If you want the program to calculate the value enter -10 for that value.

ionreswin resthk, reswin, shift, sgres1, sgres2 ; (trial 306)

Ionreswin sets resist geometry and scattering parameters. There are the resist thickness (resthk,micrometers), the resist window (reswin, micrometers), the distance from zero the resist window is shifted (shift,micrometers) and the sigma of the lateral spreading of the ions in the resist in the exposed regions (sgres1,micrometers) and the unexposed regions (sgres2,micrometers). This statement is optional. The default assumes resthk=0.5 micrometers, reswin=5.0 micrometers, shift=0, sgres1=0.011 micrometers and sgres2=0.011 micrometers. To use a default value enter -1 for that value.

ionedep axepts axbpts; (trial 308)

Ionedep determines the axial energy distribution deposited in the resist. The variable axepts indicates that a file called axiale.dat (set up by the user) will be input and that the number of data points in it is axepts. Similarly for axbpts and axialb.dat. These data files then should contain the energy deposited in the resist for the exposure regions (axiale.dat) and the background or absorber regions (axialb.dat). The units are eV/angstrom. If you want the program to calculate this data, insert -10 for these values. The statement is optional and the default case will use data appropriate for the default beam, mask and resist parameters (axepts=30, axbpts=30).

ionexpose horpts, que ; (trial 309)

Ionexpose exposes the resist, ie. it forms energy deposition matrix in the resist. The number of horizontal point used in the energy calculations is horpts. The variable que specifies how many steps are used in the tapered absorber calculations. The default values are horpts=100 and que=10. This statement is required for the program to run any exposure (The arrays for the energy deposition are initialized to zero). Use -1 to indicate default values desired.

ionfrac frac ; (trial 310)

Ionfrac sets the anisotropic rate fraction. This is a parameter that was found necessary in the e-beam machine to insure that the developing rates in the simulation matched those in experiment. The horizontal developing rate is multiplied by frac so that frac<1 would give enhanced anisotropic developing. The default value is 1.0 and the statement is optional.

ionresist r1, cm, d0, alph ; (trial 311)

Ionresist sets the development rate eqn constants for the resist. The default are those for PMMA, r1=1.0, cm=1.0, d0=174 and alpha=1.9. To indicate default values are desired insert -1 for that value. This statement is optional.

For the damage etching option the format for the ionresist keyword input is:

ionresist 99999 R1 E0 R2 ; (trial 311)

The 99999 parameter in the input is a flag that invokes the damage etching model. The default values are R1=0.0008 (micrometers/sec), E0=1.0E07, and R2=0.0024. Note: E0 is multiplied by 1.0E07 automatically in the program in order to simplify the input. Use -1 to indicate default values. This statement is optional.

iondevlp devsrt, devend, devinc, npts ; (trial 312)

# TRL312 develops the resist. Resist contours are plotted every 'devinc' seconds starting 'devsrt' seconds into the development and stopping after 'devend' seconds of development. The number of points in the development string is npts. This statement is required to achieve development of the resist. The default values give contours every 20 seconds starting at 10 seconds and stopping at 90 seconds. To indicate default values use -1 for each value.

ionecntr engmax,idep,iskip,ityplt ; (trial 313)

Ionecntr causes the program to output the energy contour data. This data is then output to a file called engpts. The vertical scale of the plot data can be set by engmax. If it is desired to have the program set the plotting scale, set engmax=1.0. To set the scale set engmax= 'number' (J/cm**3). The rows of the energy depositon array are output starting at row 'idep' and every 'iskip' rows after that (idep=1 is the top of the resist). The type of plot is determined by ityplt (ityplt=1 to 6). If ityplt=1, the contribution from each beam component is plotted separately. These components are the channeled beam, the background beam leaking through the absorbers, the beam that is transmitted through the tapered edges of an absorber and the dechannelled beam. If ityplt=2, all these components are added together before plotting. If just one component is desired then ityplt=3 (channelled beam), ityplt=4 (background beam), ityplt=5 (tapered edge beam) and ityplt=6 (dechannelled beam).

Default Parameter Listing

```
ipflgs(1)=0
ipflgs(2)=0
ipflgs(3)=0
ipflgs(4)=1
ipflgs(5)=1
ipflgs(6)=1
ipflgs(7)=1
ipflgs(8)=0
r1=1.0
cm=1.0
d0=174.
alph=1.9
are1=.0008
are2=.0024
see0=1.0e7
itype  = 1
e0     = 200.
dele   =  57.
delew  = 141.7
rangm  = 2.8396
strglm = .21743
ranga  = .45839
strgla  = .15315
dose   = 1.3e13
bangle = 0.
dosthr = 1.299e13
dosbak = 6.86e10
chimin = 9.43e-4
chidos = 1.1e10
que    = 10
chnfct = .65
psihlf =0.36
sigthe = 0.0054
psibak = 25.74
sigbak = 9.55
sigmsk  = 0.1335
sigtot  = 0.1340
absthk  = 0.85
delta   = 0.
spce    = 25.
cntrst = 2.3
fact    = 1.0
supthk = 0.85
sigbig = 9.56
sigchi = 0.17
sigtap = 0.
h200=.02609
dee=5.431
ay=1.085e-9
eh=1.8e-9
rhocrt=.08919
abdens=19.3
```

```
abstyp=74.
em1=1.008
em2=184.
eprim=67933.4
sgres1  = 0.011
sgres2  = 0.011
reswin = 5.0
shift = 0.
resthk= 0.5
data (xray(i),i=1,9)/0,1.0,2.0,3.0,4.0,4*0./
sysray(j)=0.
bakray(j)=0.
tapray(j)=0.
chiray(j)=0.
elin(k,l)=0.
tmat(k,l)=0.
bmat(k,l)=0.
cmat(k,l)=0.
horpts=100
axepts=80
axbpts=80
cellx=.0255
cellz=.00641
idevfl(1)=0
idevfl(2)=1
idevfl(3)=0
idevfl(4)=0
idevfl(5)=0
devsrt=10.0
devend=90.0
devinc=20.0
npts=100
frac=1.0
idep=1
iskip=9
ityplt=1
```

2.4. *Examples of SAMPLE Input Files for Ion Beam*

These examples are designed to illustrate the use of the ion beam simulator in SAMPLE. The first is the simplest file, the default exposure and development:

```
**expose the resist
ionexpose -1 -1
**develop the resist
iondevlp -1 -1 -1 -1
```

The above input file runs the default simulation example. Ionexpose exposes the resist using the default number of points in the energy array. Iondevlp develops the resist, and outputs the resist contours. Fig. 1 shows the developed profiles as plotted on an HP 2648 graphics terminal. Ionexpose is required in the input file in order to get an exposure and iondevlp is required in order to get the development.

The next input file illustrates how the ion beam machine can be used to calculate scattering parameters without doing the development. This file basically runs just the exposure part of the simulation:

```
**set printing flags
ionprint 0 0 0 1 1 1 1 1
** input beam parameters
ionbeam -1 190 2. 0
** set mask topography
ionmask 25 0.75 0 0.75 0 0.5 1. 1.5 2.
** calculate mask scattering
ionscat 74 -10 -10 -10 -10 -10 -10
** set resist geometry
ionreswin 1 2.5 0 .01 .01
** calculate axial energy deposition in resist
ionedep -10 -10
** expose the resist
ionexpose 200 -1
** output the energy contours in the resist
ionecntr 1 1 27 1
```

In this example ionprint sets the printing flags to print out the scattering information which includes the ion ranges (ipflgs(8)). Ionbeam sets the hydrogen beam parameters to 190 keV at 2e13 /cm**2. Ionmask sets the absorber thickness to .75 micrometer as well as the support membrane. The mask/resist spacing is set to 25 micrometer and the tapered edge width to 0. The last five numbers set the absorber edges form an alternating 1 micron absorber pattern from 0 to 2.5 micrometers. Ionscat sets the absorbers to tungsten and tells the program to calculate each of the five scattering parameters. Ionreswin inputs the resist thickness (1 micrometer), the resist window of interest (2.5 micrometer), the distance this window is shifted (0 micrometer) and the lateral scattering of the channeled ions and the background ions in the resist (.01 micrometer). Ionedep tells the program to calculate the axial energy deposition in the resist. The first -10 indicates for the channeled and dechanneled ions, the second -10 for the background ions. Ionexpose exposes the resist using 200 points in the lateral direction of the energy array (perpendicular to the ion directon). Finally, ionecntr outputs the energy deposition distribution into a file called engpts. It specifies to print the 1st and 28th rows of the energy array, row 1 is the resist surface and 28 is near the resist/substrate interface. A plot of the engpts file is shown in Fig. 2. Note that the energy deposited at the bottom of the resist is almost identical to that at the top, it is actually slightly higher.

This next example adds the development option:

```
**set beam parameters
ionbeam -1 190 2. 0
**set mask geometry
ionmask 25 0.7 0 0.8 0 0.5 1. 1.5 2.
**set scattering parameters
ionscat 14 110 0.8 1.25 -10 30 70
** set resist geometry
ionreswin .4 2.5 0 .01 .01
** input axial energy deposition
ionedep 35 16
**expose the resist
ionexpose -1 -1
**develop the resist
iondevlp 15 60 15 200
```

This file is very similar to the previous example with the following exceptions: In ionscat the user is setting the absorber type to silicon, the energy lossed in the support membrane to 110 keV, the angular spread of the channeled beam to 0.8 degrees, the dose that makes it through the membrane to 1.25e13, the mask contrast is to be calculated, the angular spread of the background beam to 30 degrees and the energy lossed in the absorbers to 70 keV. Note that these numbers are merely for illustration and do not correspond to any particular experiment. Also in ionedep the user is specifying that the axial energy deposition data for the channeled ions is in a file 'axiale.dat' and contains 35 points and the background axial deposition data is in a file 'axialb.dat' and contains 16 points. In iondevlp the development contours are set to every 15 seconds for 60 seconds using 200 points on the development string to start. A plot of the f77punch7 file is shown in Fig. 3. This is what we in the business would call a bad exposure, but the parameters are for illustration only and do not represent a typical exposure.

This final example illustrates the use of the tapered absorber option:

```
** set beam parameters
ionbeam 1 250 2.5 0
set tapered mask geometry
ionmask 25 1.1 1 1.25 1.
** calculate mask scattering
ionscat 74 -10 -10 -10 -10 -10 -10
** set resist parameters
ionreswin 1 2. 0 .01 .01
**calculate axial energy deposition
ionedep -10 -10
**expose resist
ionexpose -1 10
**develop resist
iondevlp 60 300 60 100
**output energy contours
ionecntr 1 1 27 1
```

In ionmask a 25 micrometer mask/resist spacing is specified, a 1.1 micrometer thick absorber with a tapered edge that is 1 micrometer wide forming a 47 degree slope. The support membrane is 1.25 micrometer thick and the absorber edge begins at 1.0 micrometers. Ionexpose tells the program to expose the resist and use 10 steps as an approximation of the sloped absorber edge. Ionecntr prints out the 1st and 28th rows of each of the contributions to the exposure energy density array. The f77punch7 file is plotted in Fig. 4. The engpts file is illustrated in Fig. 5. The solid lines indicate the surface of the resist, the dotted lines the bottom. The channeled component and tapered edge components are the only two contributions large enough to show up at this scale.

2.5. *Subroutine Explanations*

INITIB initializes default parameters

SUBROUTINE IBARRA forms the energy deposition distribution matrix

SUBROUTINE MTFRMA forms the 2-D energy deposition distribution in the resist window of interest. (using analytical data)

ANASYS calculates part of the lateral ion distribution after passing through the mask, the result is a 1-D array in the resist window of interest. this routine gives the contribution of the ions which pass through the open membrane regions (channeled ions).

BAKSYS calculates part of the lateral ion distribution after passing through the mask, the result is a 1-D array in the resist window of interest. this routine calculates the contribution of the ions that pass through the Absorber regions of the mask.

CHISYS calculates the lateral distribution of the non- channelled ions, the result is a 1-D array in the resist window of interest. these ions pass through the open membrane regions of the mask only.

TAPSYS calculates the lateral distribution of ions that pass through a tapered absorber region, the result is a 1-D array in the resist window of interest. The tapered absorber is approximated as a series of steps. (specified by que.

FUNCTION ZIRATE(cz) finds development rate in micrometers/sec zirate lets the string develop outside the boundary at a much reduced rate, in order to keep the string length down. chkr deletes the points outside of (0.,xmax).

IBMSG(numb) is the message subr for ion-beam

SUBROUTINE PRARRA prints arrays according to which flags are set

SUBROUTINE EDEPAX inputs the axial energy deposition

SUBROUTINE ESTRAT lines up axial energy arrays

SUBROUTINE MSKATR sets scattering parameters of the mask

FUNCTION RERF(x) This routine calculates the error function (Erf(x) for all real values. (As opposed to erf(y) which calculates the statistical error function and gaussn calculates Erfc(x) for positive arguments. ie rerf is the integral from -oo to x of (1/sqrt(2pi))*exp(-t**2/2)*dt

IBDEV is the sub-controller for the pr develop routines. idevfl(3)=1 for publication runs, which are more costly.

SUBROUTINE ABSCAL calculates the psibak scattering angle for different ions, energies, given the absorber thickness. This calculation adds the effect of the membrane scattering to the absorber scattering to produce sigtap, the total sigma for an absorber of thickness 'tee' on a membrane of thickness 'supthk'. The absorber calculation follows that of Meyer.

FUNCTION G1TAU(x) calculates the g1(tau) function used in the calculation of Meyer for plural scattering. A two part polynomial fit is used, each of the form: y = a + b*x + c*x**2 + c*x**3 ...

FUNCTION G2TAU(x) calculates the g2(tau) function used in the calculation of Meyer for plural scattering. A two part polynomial fit is used, each of the form: y = a + b*x + c*x**2 + c*x**3 ...

SUBROUTINE SIGCAL calculates the sigthe scattering angle for different ions, energies, and crystal directions. This calculation follows that of Wilson et al.

SUBROUTINE CRTCAL calculates the critical angle and chimin for the incoming ions.

SUBROUTINE ENCALC(isub,ichn,iabs) calculates the energy loss by an H+ ion of energy e0 (keV) passing through a silicon or tungsten layer of thickness x (microns). The energy loss is reduced by the channeling factor (chnfct) to account for reduced electronic stopping in a channeling mode.

FUNCTION SECALC2(e,z2) calculates the electronic stopping power for H+ according to the formula of Varelas and Biersack, as modified by, and using the parameters determined by Anderson Ziegler in "Hydrogen Stopping Powers and Ranges in All Elements" vol. 3 1977. The four parameter fit is:

$$se = (slo*shi) / (slo + shi)$$

where: slo=a1*e**.45 , shi=(a2/e)*log(1+a3/e+a4*e) a1,a2,a3,a4 are fitting parameters, e the energy in keV/amu(H+) conv is a conversion factor to get se in keV/micron (se is originally in ev-cm2/10e15 atoms) for e < 10 keV, se=alo*e**.5, where alo is the low energy fitting parameter

SUBROUTINE MEMDOS calculates the dose passing through the membrane (or absorber if it's silicon) based on e0, Rp and Straggle info. Data used is a polynomial fit to Andersen and Ziegler data.

SUBROUTINE ABSDOS calculates the dose passing through the absorbers based on e0, dele, Rp and Straggle info. Data used is a polynomial fit to Andersen and Ziegler data.

SUBROUTINE IPLOT will print, into a file, the absorbed energy density of the convolved, arrayed profile, in the window of interest--at specified depths in the resist.

SUBROUTINE ELOSS calculates the energy deposited in a 'restthk' layer of pmma using Braggs rule (PMMA=C5-H8-02). The energy under the transparent regions is put into axe(80) and the enrgy deposited under the absorbers is put in axb(80) 30 points are calculated

SUBROUTINE ENCALC2(e0,x,z,elos) calculates the energy lost by an H+ ion of energy e0 (keV) passing through resist of thickness x (microns)

SUBROUTINE TLOSS(e,ax) calculates energy deposited in in resist under a tapered absorber, in a similar fashion to eloss.

SUBROUTINE TECALC(thcknss,efin) calculates the energy lost by an H+ ion of energy e0 (keV) passing through a tapered tungsten layer of thickness x (microns).

SUBROUTINE TRL301 sets flags or printing arrays/info

SUBROUTINE TRL303 inputs beam parameters

SUBROUTINE TRL304 inputs mask topography

SUBROUTINE TRL305 sets mask scattering parameters

SUBROUTINE TRL306 sets resist window of interest and scattering parameters.
SUBROUTINE TRL308 determines the axial energy deposition

SUBROUTINE TRL309 calculaates the system response and calculates the 2-D energy dep matrix SUBROUTINE TRL310 sets anrate fraction

SUBROUTINE TRL311 sets rate eqn constants

SUBROUTINE TRL312 runs the develop routines

SUBROUTINE TRL313 this initializes and runs the energy contour option

2.6. *Additional Common Block Descriptions*

Not all of the variables used in the MIBL simulator are accessible through the keyword statements. These "hidden variables" might be useful to someone who wants to modify this program or is interested in exactly how the calculations are made. For this reason, the variables ***not already detailed in the keyword statement section 2.3*** are described here.

```
      common /sistem/ sysray(1000), horpts, fact, bakray(1000),
     *          tapray(1000)
```

The lateral distribution of the ions at the resist surface is described by sysray, bakray and tapray for the channeled, background and tapered edge contributions respectively. These are normalized distributions.

```
      common /sysflg/ iflgbk,iflgtp,iflgch
```

The flags iflgbk, iflgtp and iflgch signal the program to make the background, taper and dechanneled calculations respectively.

```
      common /mscatr/ sigmsk, psihlf, sigthe, spce, sigtot, dele,
     *          dosthr,dosbak,cntrst,psibak,sigbak,sigbig,delew
```

The variable sigthe is equal to psihlf in radians. The variables sigmsk and sigbak are sigma (in micrometers) of a delta function input beam when it reaches the resist surface of the channeled and background components. sigtot and sigbig are the sigmas of a delta function input beam in the resist of the channeled and background components (micrometers). dosbak = dosthr/10**cntrst.

```
      common /mskdat/ rangm,strglm,que,ranga,strgla
```

rangm and strglm are the range and straggle of the channeled ions in the membrane support region. ranga and strgla are the range and straggle of the ions in the absorbers.

```
      common /msktop/ xray(9), xrmax, absthk, delta,supthk
```

xrmax is the number of points specified in xray(9).

```
      common /angdat/ eh,dee,rhocrt,eprim,psicrt,h200,psi200,sig200,
                crtang
```

eh, dee, rhocrt, and eprim are the variables used in the calculation of the critical angle according to Wilson.[12] psicrt is the critical angle that corresponds to the exit energy of the channeled ions and h200 is the critical angle that corresponds to the exit energy of the channeled ions when the initial energy is 200 keV. psi200 and sig200 are the half angle and sigma for an initial energy of 200 keV. crtang is the calculated critical angle for the incoming ions.

```
      common /absdat/ abstyp,abdens,ay,em1,em2
```

absdens is the density of the absorber (g/cm**3) and ay, em1 and em2 are parameters used in the multiple scattering calculations of Meyer [5]

common /cnvlv2/ emat(80,500), numspt, wght0, ncemat

This common block was left in from the e-beam program for use in the monte-carlo option which is not yet implemented.

common /cell/ cellx, cellz

cellx and cellz are the cell sizes (micrometers) of the energy matrix elin(82,1002).

common /chndat/ chnfct

chnfct is the percentage of the electronic energy loss that channeled ions see relative to dechanneled ions.

common /chidat/ cmat(82,1002),chiray(1000),sigchi,chidos,chimin

cmat is the energy deposition matrix contribution due to the dechanneled ions (J/cm**3). chiray is their lateral distribution (normalized). sigchi is the sigma of the dechanneled beam at the resist surface (micrometers), chidos is the total dose of the dechanneled component (1/cm**2) and chimin is the percentage of the total dose that was dechanneled.

common /sigdat/ sigtap,dostap,itap,rangt,strglt

sigtap is the sigma of the beam traveling through the tapered edge when it reaches the resist surface. dostap is the dose of the tapered beam component, itap is a flag used in the tapered edge calculations, and rangt and strglt are the range and straggle of the ions in the tapered edge.

common /io1/ itermi, ibulk, iprout, iresvl, iin, iprint, ipunch

these parameters are i/o variables that are dependent on the computer system in use.

common /flags/ iflags(20)

debugging flags that indicated that certain calculations were finished.

common /edep/ axe(80), axepts
common /edep2/ axb(80), axbpts

axe(80) is the energy loss of the channeled and dechanneled ions in the resist and axepts is the number of points used in axe(80). axe is input in eV/angstrom but is converted to J/cm. axb(80) and axbpts are the same variables but for the background beam component.

common /impulse/ impray(1000)

Put in for montecarlo calculations but not implemented.

```
common /line1/ elin (82,1002),lcou,lincou,elnwgt (1999)
```

elin is the total energy deposition distribution in the resist. the other variables are leftover from the e-beam code and aren't used here.

```
common /mattmp/ bmat(82,1002),tmat(82,1002),amat(82,1002)
```

bmat is the energy deposition distribution in the resist for the background ions, tmat for the tapered edges and amat is the contribution from the channeled ions.

```
common /devflg/ idevfl(5)
```

flags for the developing machine. idevfl(1)-print no. of string points, etc...idevfl(2)- print out points for each profile, and idevfl(3)- print more points for more accuracy (takes more time also).

```
common /dvelp1/ cxzl,cxzr,xz(1000),xmax,zmax,npts,nadchk,nckout
common /dvelp2/ tadv,tchk,ttot,iflag,smaxx,sminx,smaxz
common /dvelp3/ nzflg,ttotsv
common /dvelp4/ break,maxpts,nadsav,ncksv1,ncksv2,nout
common /horimg/ deltx,mnhpts,nmhpts,horint(50)
common /simpar/ nprlyr,nprpts,nendiv,deltm,deltz
```

These common blocks have the same function as in the e-beam code and are used in the development algorithm.

FIGURE CAPTIONS

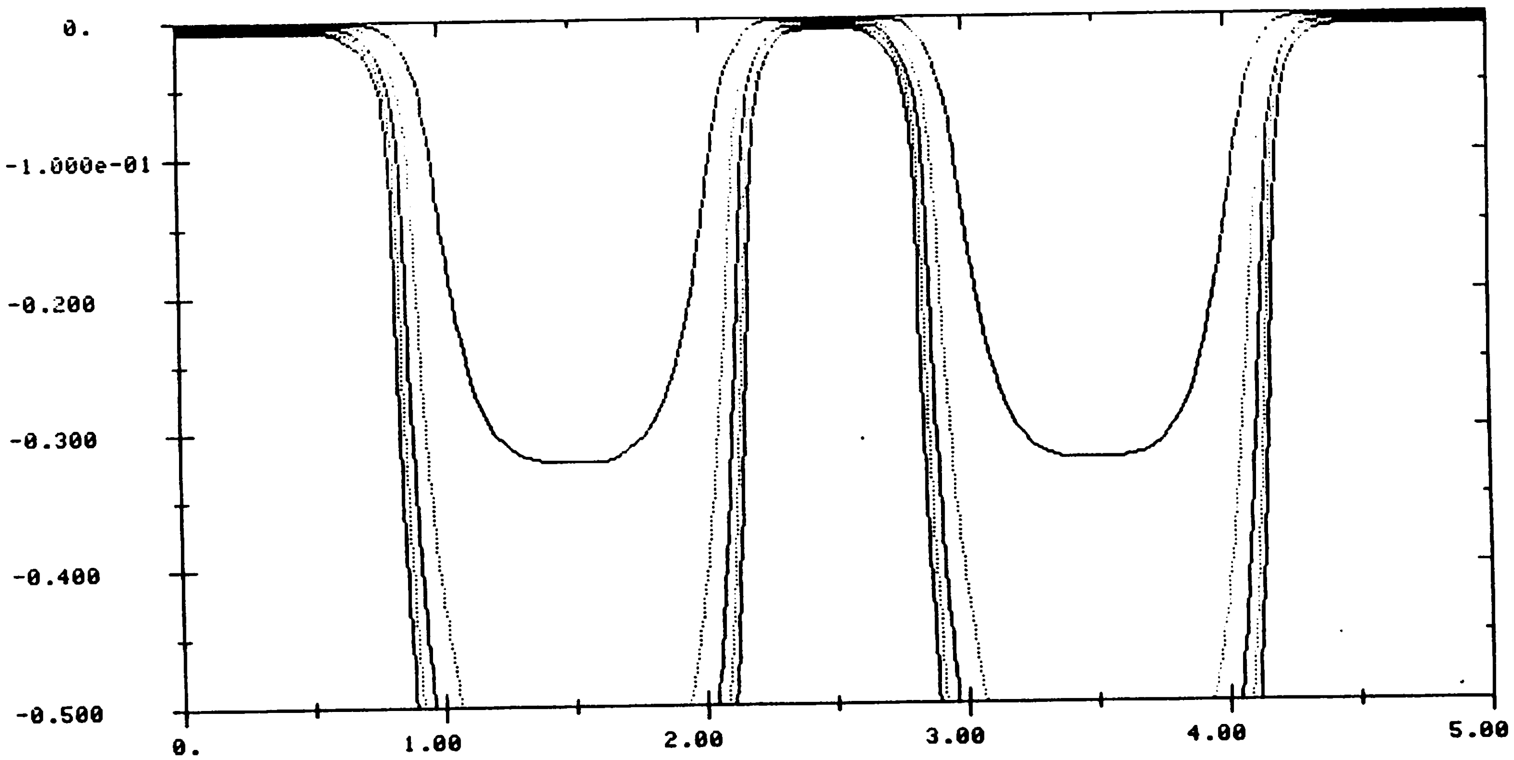
Fig. 1. The default exposure resist profiles.

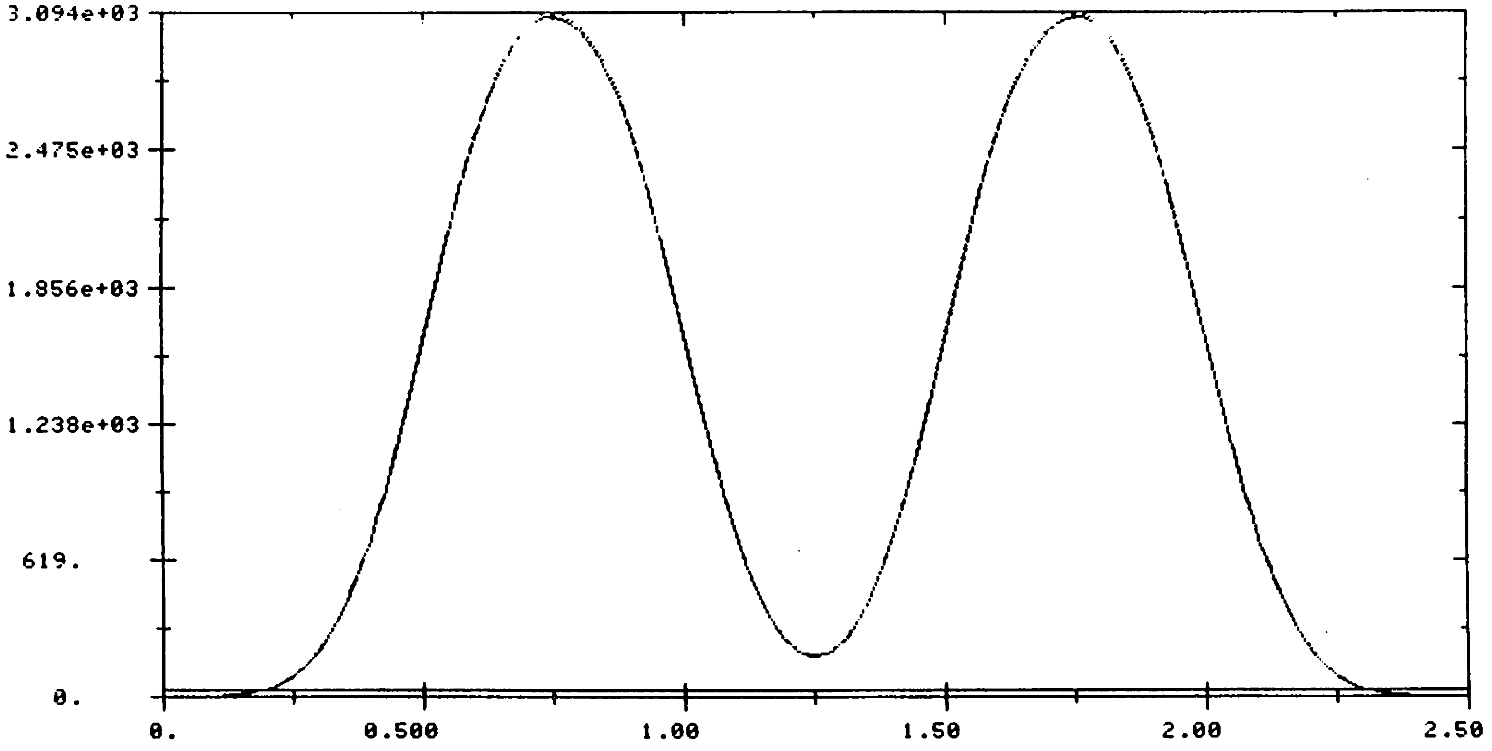
Fig. 2. Energy contours for example 2.

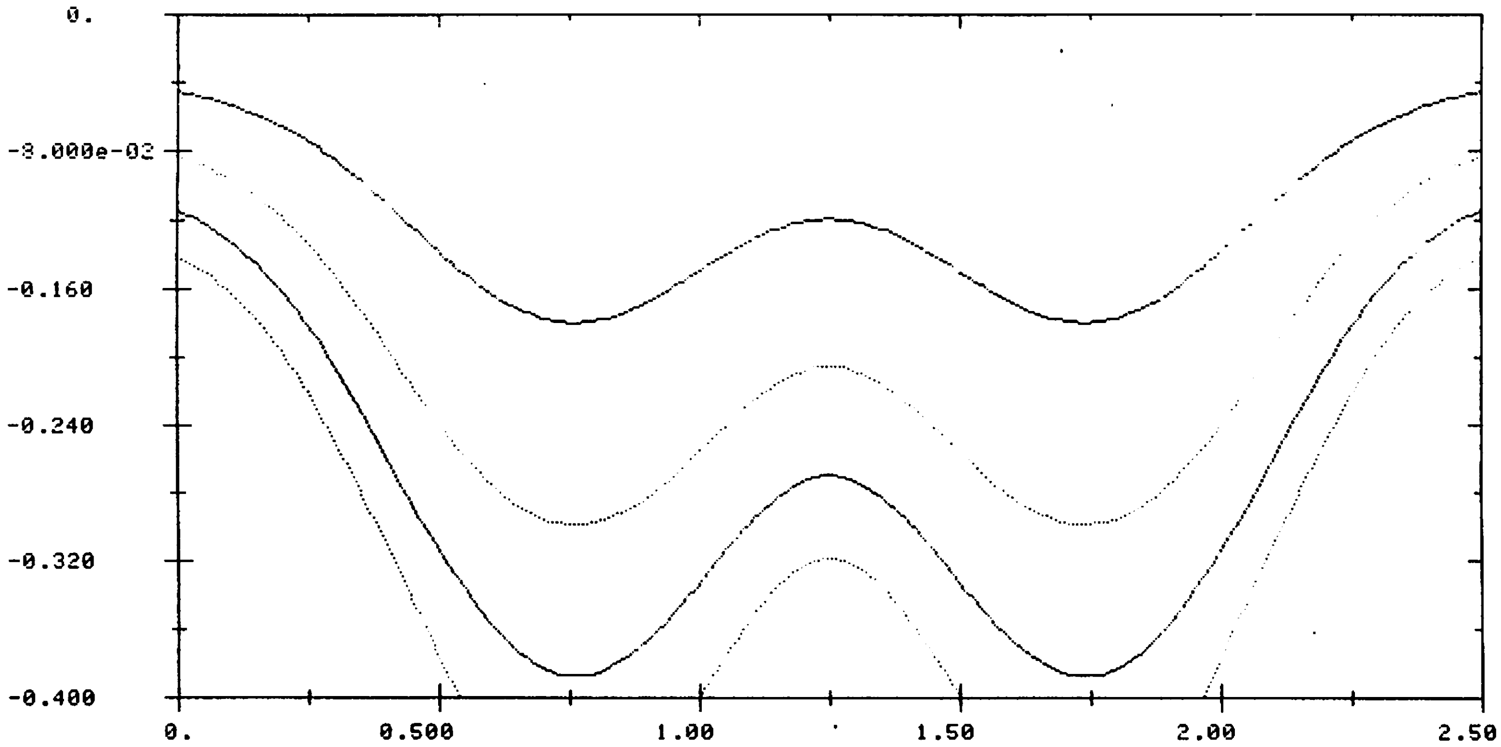
Fig. 3. The resist profiles for example 3.

Fig. 4. Tapered absorber resist profiles.

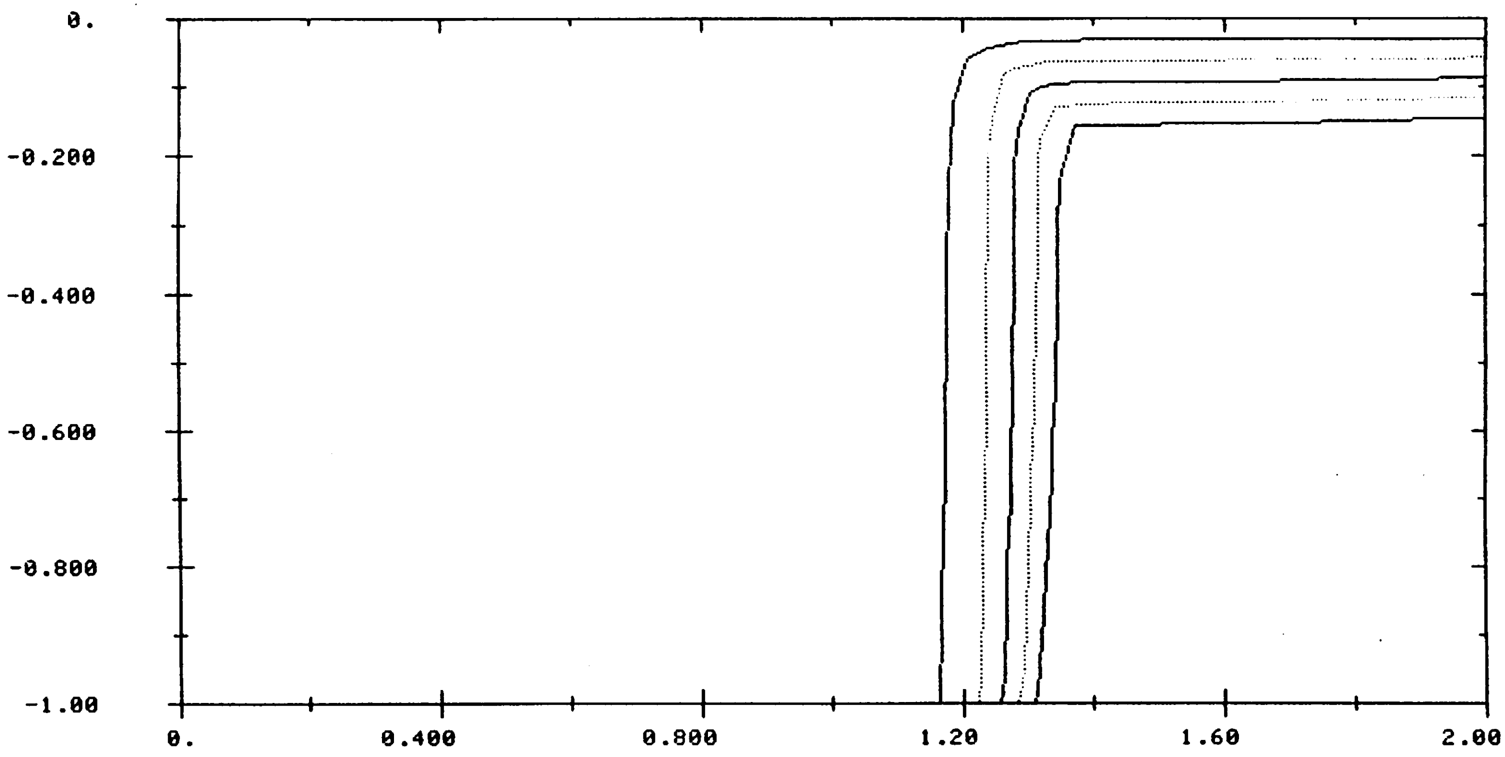
Fig. 5. Tapered absorber energy contours.

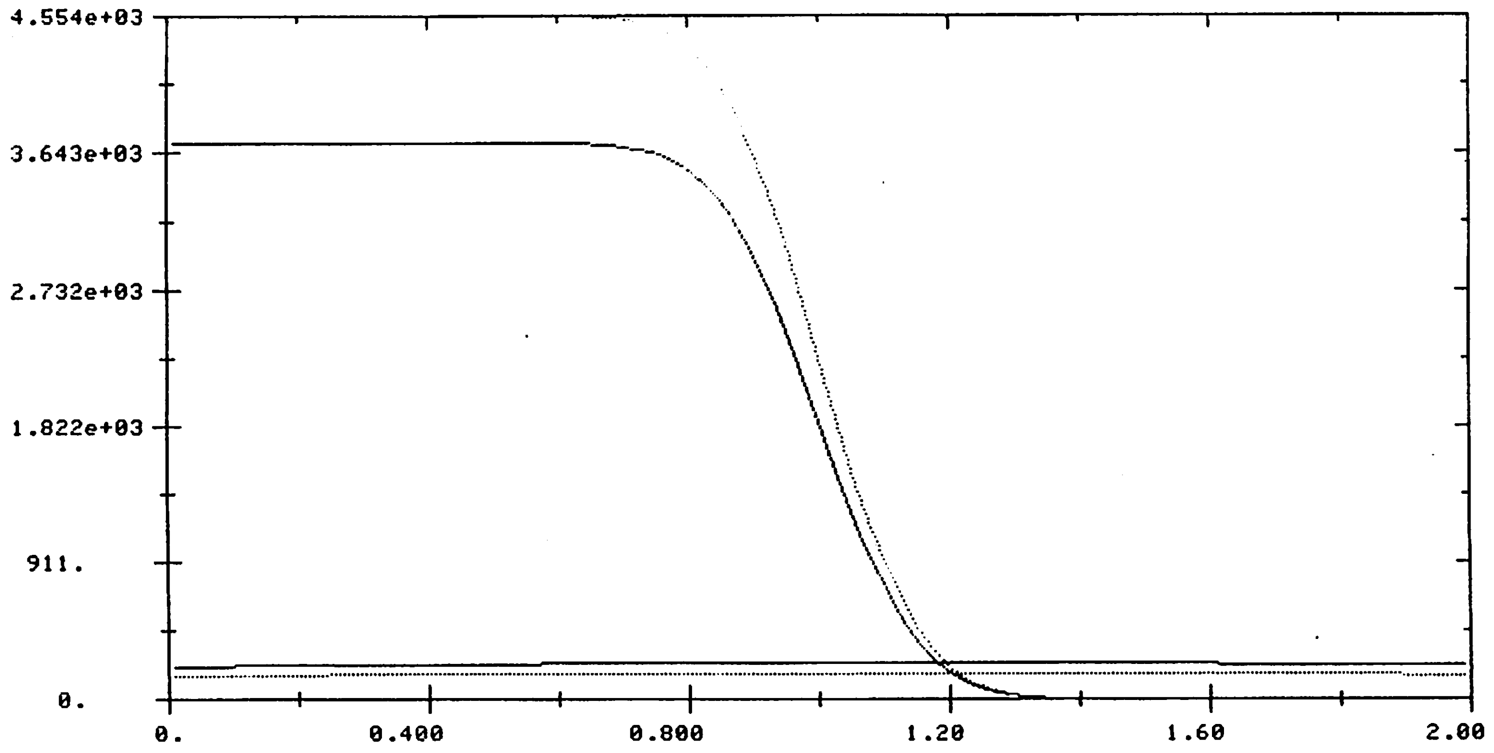

0.
-1.000e-01
-0.200
-0.300
-0.400
-0.500
0.
1.00
2.00
3.00
4.00
5.00


3.094e+03
2.475e+03
1.856e+03
1.238e+03
619.
0.
0.
0.500
1.00
1.50
2.00
2.50



0.
-8.000e-02
-0.160
-0.240
-0.320
-0.400
0.
0.500
1.00
1.50
2.00
2.50


0.
-0.200
-0.400
-0.600
-0.800
-1.00
0.
0.400
0.800
1.20
1.60
2.00


4.554e+03
3.643e+03
2.732e+03
1.822e+03
911.
0.
0.
0.400
0.800
1.20
1.60
2.00