Copyright © 1985, by the author(s).
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

SIMPL-2 (SIMULATED PROFILES FROM LAYOUT - VERSION 2)

by

K. Lee

Memorandum No. UCB/ERL M85/55

3 July 1985

title page

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

SIMPL-2
(*SIM*ulated *P*rofiles from the *L*ayout - version *2)*

Keunmyung Lee

Ph.D. Dissertation

Department of Electrical Engineering and Computer Sciences
University of California, Berkeley, California 94720, U.S.A.

ABSTRACT

SIMPL is a computer-aided design (CAD) tool for simulating the cross sectional profile of integrated circuits along an arbitrary 'cut-line' drawn on the layout. The second generation of SIMPL. SIMPL-2 is capable of displaying in color two-dimensional process effects such as the "bird's-beak," lateral diffusion, undercut in etching and sidewall coverage in deposition, based on its linked-polygonal and grid type databases.

The device profile from the composite process can be generated rapidly using elementary internal physical process models. More rigorous external process simulators can also be invoked through an interface for profile data and transfer of control. An external process simulator, SAMPLE, is successfully linked to SIMPL-2 for the deposition process. As of now, the other processes are done by internal routines. However, these modules are easily detachable and the other external simulators can be linked for more rigorous simulation. SIMPL-2 thus acts not only as a color display tool for other simulators but also provides a here-to-fore missing link between layout based CAD tools and process and device simulators.

For the graphics interface, SIMPL-2 uses a device-independent graphics package, MFB (Model Frame Buffer). The color and pattern information for each layer of layout and cross section is stored in a readable file and can be modified.

SIMPL-2 consists of about 12,500 lines of C codes excluding MFB and the external process simulator. Currently, SIMPL-2 runs on a VAX11/780 with Berkeley UNIX 4.2BSD.

Committee Chairman

SIMPL-2

(*SIM*ulated *P*rofiles from the *L*ayout - version 2)

Copyright © 1985

by

Keunmyung Lee

Dedicated to
my parents,
brother and sister

Acknowledgements

I would like to thank my advisor Professor Andrew R. Neureuther for his guidance. encouragement. and friendship throughout this research. I also wish to thank Professors W. G. Oldham. D. A. Hodges. J. R. Whinnery. and E. J. Pinney for serving on my dissertation committee.

I wish to thank my colleagues in the SAMPLE group. It has been a unforgettable experience to work with them. I thank Gino Addiego. Sharad Nandgaonkar. Paul Carey. Mike Grimm. Yoshi Sakai. Mike Rosenfield. the late John Reynolds. Wendy Fong. Hudy Damar. Wing Leung. and Deok Kim for their support and friendship.

It is a pleasure to acknowledge the contributions of Peter Moore and Professor A. R. Newton on the last stage of this work. I wish to thank Soo-Young Oh and John Moll at Hewlett-Packard Laboratories for their advice during my work at the laboratory which influenced this research.

It was joyful to make friends with many people in Berkeley. I thank everyone of them for their help on my life.

The financial support from SRC is gratefully acknowledged.

I dedicate this dissertation to my parents. Ye Min and Hee Kyung Lee. sister. Myung Sun. and brother. Keun Jae for their love.

Table of Contents

Dedication i

Acknowledgements ii

Table of Contents iii

CHAPTER 1 INTRODUCTION 1

1.1 Definition of SIMPL 1

1.2 History of SIMPL 1

1.3 Dissertation Outline 2

CHAPTER 2 SIMPL-2 : A CAD Tool for IC CAD 3

2.1 IC CAD Environment 3

2.2 Functional Role of SIMPL-2 3

CHAPTER 3 PROGRAM STRUCTURE OF SIMPL-2 6

3.1 Program Structure 6

3.1.1 General 6

3.1.2 Layout 6

3.1.3 Process 6

3.1.4 Model Frame Buffer (MFB) 8

3.1.5 Running without Graphics 8

3.2 Layout Data Data Structure 8

3.3 Cross Section Data Structure 10

3.3.1 General 10

3.3.2 Linked-Polygonal Data Structure 10

3.3.3 Grid Data Structure 14

3.4 Graphics Interface 14

3.4.1 MFB 14

3.4.2 Routines for Graphics Interface 14

CHAPTER 4 PROCESS SIMULATION 16

4.1 General 16

4.2 Internal Process Simulators 16

4.3 Interaction with External Process Simulators 16

4.3.1 Interface with SAMPLE 17

4.3.2 Interface with the Other External Process Simulators 17

CHAPTER 5 INTERNAL PROCESS SIMULATORS 20

5.1 General 20

5.2 Photo-Lithography Process 20

5.2.1 General 20

5.2.2 Photo-Resist Coating 20

5.2.3 Exposure 21

5.2.4 Development 23

5.2.5 Shortcut for the Photo-Lithography Process 23

5.3 Deposition Process 26

5.4 Etching Process 26

5.5 Oxidation Process 27

5.6 Ion Implantation Process 29

CHAPTER 6 FUTURE EXTENSIONS 32

6.1 General 32

6.2 Interaction with Other Simulators 32

6.3 User Interface 33

CHAPTER 7 EXAMPLE RUNS 34

7.1 Berkeley CMOS Inverter 34

7.1.1 Run Sheet of the Berkeley CMOS IC 34

7.1.2 Simulation by SIMPL-2 41

7.2 DRAM Cells 50

References 53

Appendix : SIMPL-2 User Guide 57

CHAPTER 1 INTRODUCTION

§ 1.1 Definition of SIMPL

SIMPL (SIMulated Profiles from the Layout) is a computer-aided-design tool for integrated circuits which automatically simulates the cross sectional view of integrated circuits along an arbitrary 'cut-line' drawn on the layout. For a given layout, SIMPL extracts the mask information from the intersections of the layout geometries with the specified cut-line. Using this layout information, SIMPL generates the evolution of cross sectional structure according to the process steps given to SIMPL interactively or in batch mode.

§ 1.2 History of SIMPL

The first generation of SIMPL, SIMPL-1 [Grim83], was developed in 1983. The main object of SIMPL-1 was to give rapid on-line visual feedback to the circuit designer as to the topographical features being created during layout. For this application, SIMPL-1 approximates the cross section with rectangles based on the linked-rectangular database, and uses very simple one-dimensional process models. SIMPL-1 takes the layout information in the CIF [Mead80] format and outputs the cross section in CIF. Thus, it is possible to use a layout graphics editor which understands CIF file format to draw the cross section output such as KIC [Kell81].

The second generation, SIMPL-2, which is the topic of this dissertation, simulates the cross section with two-dimensional effects such as the "bird's beak," lateral diffusion, undercut in etching, and sidewall coverage in deposition. To include the two-dimensional effects, a quantum leap from SIMPL-1 was necessary. A new data structure has been devised and implemented. SIMPL-2 supports two kinds of databases: a linked-polygonal database and grid type database. Another feature of SIMPL-2 is that its database and modular program structure make it possible to connect external process and device simulators. As of now, the process simulation program, SAMPLE [Oldh79],

[Oldh80]. is connected to SIMPL-2 for the deposition process. A convenient user interface is also implemented using the device independent graphics package MFB (Model Frame Buffer) [Bill83].

§ 1.3 Dissertation Outline

The position of SIMPL-2 in the environment of IC CAD is discussed in Chapter 2. Chapter 3 discusses the program and data structure of SIMPL-2 including the graphics interface. Chapter 4 describes how the process simulations are and can be done with the internal and external simulators. Each of the internal process simulation modules are described in detail in Chapter 5. The directions of future extension of SIMPL are discussed in Chapter 6. Example runs of SIMPL-2 are presented in Chapter 7. The appendix includes the user guide of SIMPL-2.

CHAPTER 2 SIMPL-2 : A CAD Tool for IC CAD

§ 2.1 IC CAD Environment

As integrated circuit technology advances, computer-aided-design and computer simulation become imperative in the area of the device and process design as well as circuit and system design. Fig.2-1 shows the overall CAD environment for integrated circuits. When a system specification is given, the system is partitioned into several blocks according to their functions. Circuit designers implement the functional blocks with circuit elements. The circuits are simulated and modified through two loops: at first without layout consideration, and then with layout. These two iteration loops are shown in Fig.2-1 on the left and right, respectively.

Another necessary CAD effort is necessary in device design which starts from the bottom in Fig.2-1. This path determines the physical characteristics of each element fabricated in the integrated circuits. As a process sequence is given, the process is simulated and modified along with the experimental feedback. The devices fabricated from the process are characterized and the resulting models are supplied to the circuit simulator as shown in Fig.2-1 by means of the dotted line. The final layout and process steps are determined by these two main paths: one which starts from the system specification and the other from the process specification.

§ 2.2 Functional Role of SIMPL-2

As discussed in § 2.1, the process and device simulation provides the electrical characteristics for the circuit simulation. Most of the two-dimensional process and device simulators such as SAMPLE, SUPRA [Chin82], MINIMOS [Selb80], and GEMINI [Gree80] do not utilize layout information for their simulations. However, as the integration level becomes denser, layout affects the circuit performance drastically and must be taken into consideration. To include the effects of layout into the process and device simulation, a full three-dimensional approach is necessary. OYSTER [Kopp83]

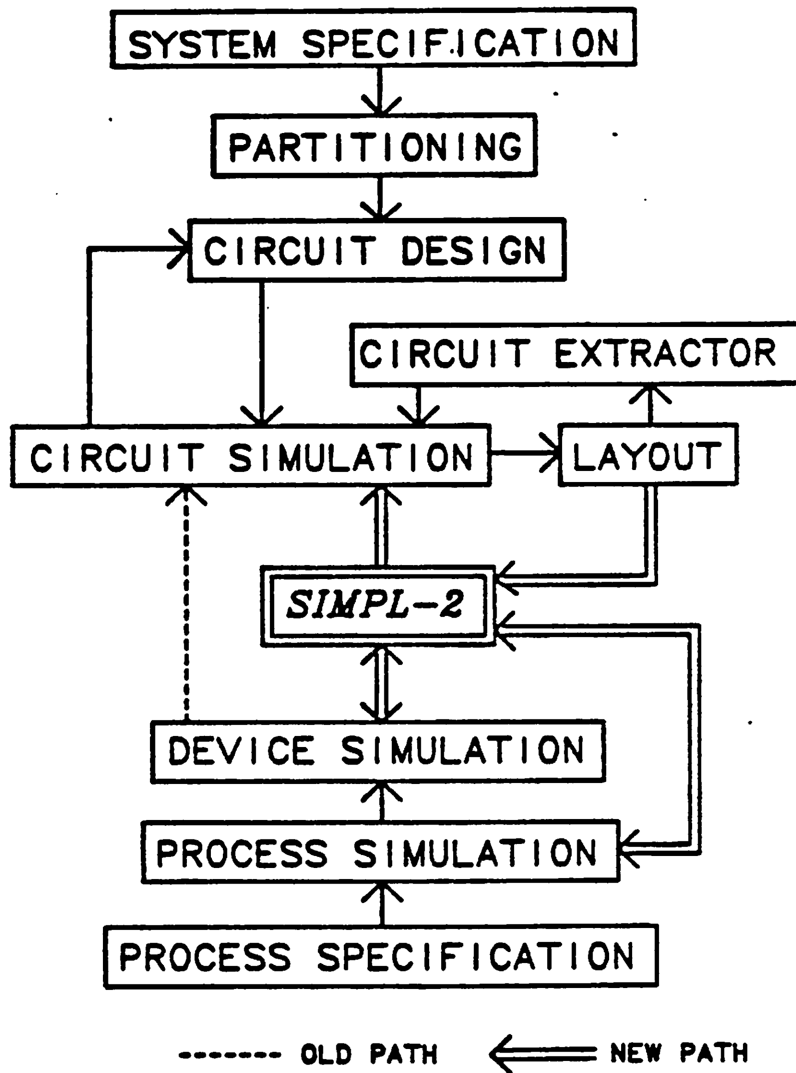


Fig.2-1 Role of SIMPL-2 in the IC CAD environment.

attempts three-dimensional simulation using solid-modeling techniques. but due to over-simplified geometries and large amount of computation time it is an unrealistic design aid at this point.

SIMPL-2 tries a pseudo-three-dimensional approach. Instead of full a three-dimensional simulation. SIMPL-2 extracts the mask information along an arbitrary cut-line and provides "mask" information to the process simulators. With several cuts in the critical areas, an IC designer can obtain valuable information which includes both the horizontal (layout) and vertical (process) information. SIMPL-2 generates the device profile from the composite process rapidly using elementary internal physical process models. Also, more rigorous external process simulators can be invoked through an interface for profile data and transfer of control. SIMPL-2 can thus act not only as a color display tool for other simulators but can also provide a here-to-fore missing link between layout based CAD tools and process and device simulators.

The concept of SIMPL is to provide the circuit designer with a physical view of the structures being produced in conjunction with the layout. Also, the communication capabilities in SIMPL-2 provide two additional benefits. First, SIMPL-2 permits mask information to be automatically transmitted to process and device simulators. Second, SIMPL-2 can also provide a 'conduit' amongst process and device simulation programs.

CHAPTER 3 PROGRAM STRUCTURE OF SIMPL-2

§ 3.1 Program Structure

§ 3.1.1 General

The program structure of SIMPL-2 is shown in Fig.3-1. SIMPL-2 consists of modules managing layouts, process simulation, and the graphics interface. The program language, "C" [Kern78], is employed, since C is appropriate for the ease of implementing data structures and gives straightforward compatibility with other CAD programs, such as MFB which is used for the graphics interface. The first version of SIMPL-2 consists of 12,500 lines of C and runs on VAX11/780 with UNIX 4.2BSD operating system.

§ 3.1.2 Layout

SIMPL-2 processes two kinds of inputs, layout data and process information. The routines related to the layout do the following functions. First, a layout file is read in and transferred to the graphics routines. The transferred layout data is displayed on the upper half screen of a color graphics terminal. Second, when the user specifies a 'cut-line' with a pointing device such as a mouse or light pen, the layout cutting routines check every element in the layout and collects the mask edges which are intersected by the cut-line. This mask information is stored for use in subsequent processing steps. § 3.2 discusses the data structure for the layout in detail.

§ 3.1.3 Process

The physical process modules are the core of SIMPL-2. SIMPL-2 maintains the database which contains the cross sectional structure along the cut-line. Each process module acts on the current database and rebuilds the database for the specified process step. The profile can be displayed or saved into a permanent file at any point in the sequence of processes. The process information can be fed into SIMPL-2 interactively or in batch mode. During an execution of SIMPL-2 in interactive mode, the sequence of

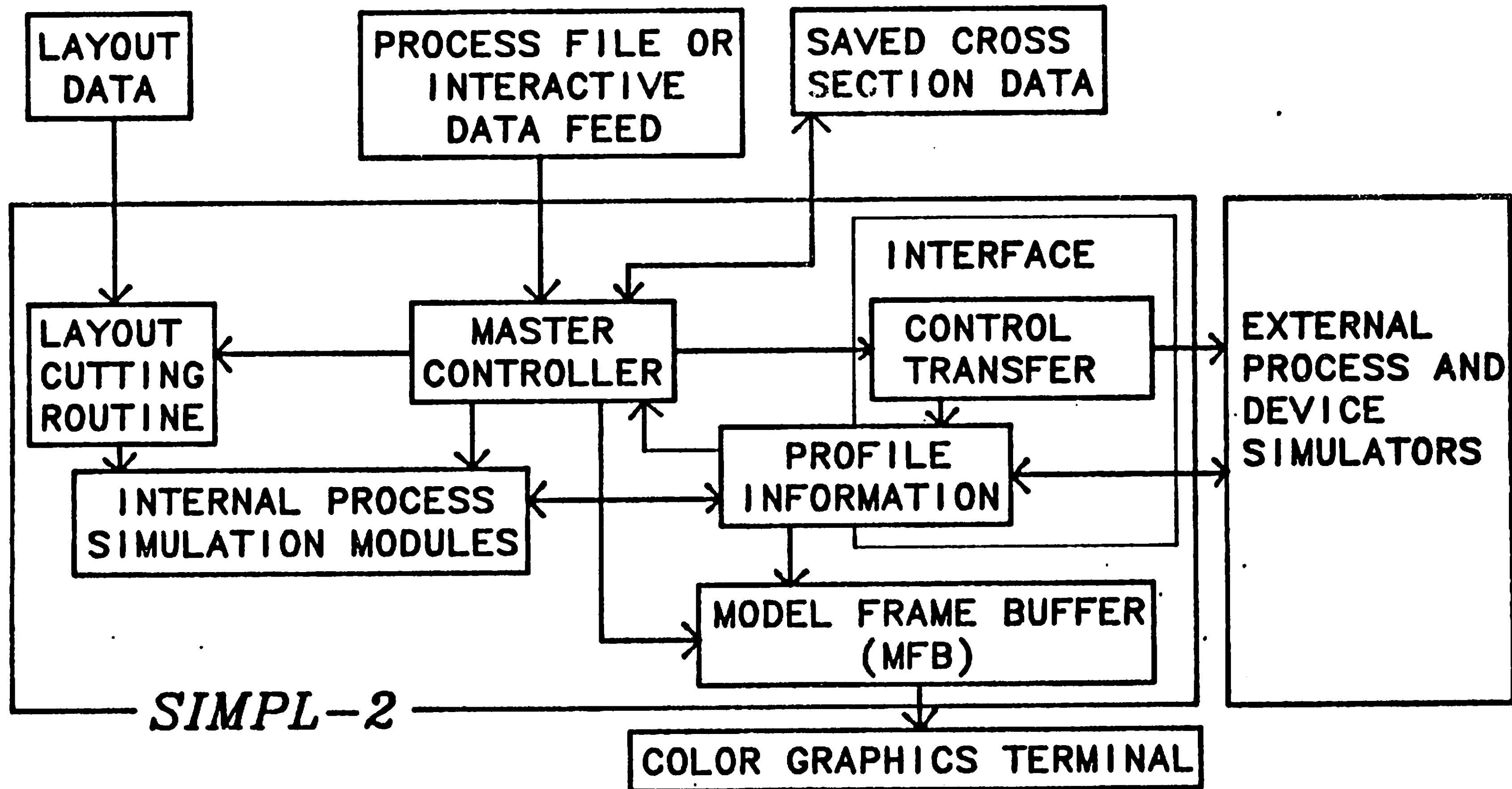


Fig.3-1 Program structure of SIMPL-2.

process inputs can be stored into a file for later use in batch mode with the same process and a different cut-line.

The process routines are fully modularized and can be either internal or external. Currently, all the process modules except the SAMPLE deposition module are internal. However, these modules can easily be detached and external simulators can easily be linked. The data structure for the cross section is dealt with in § 3.3 and process modules are discussed in detail in chapter 4 and chapter 5.

§ 3.1.4 Model Frame Buffer (MFB)

As shown in Fig.3-1, all graphic data is controlled by Model Frame Buffer (MFB). Also, the interactive commands given by the user are first channeled through the MFB. The graphics interface with the MFB is explained in § 3.4.

§ 3.1.5 Running without Graphics

SIMPL-2 also supports a non-display mode. In this mode, all the physical processes are executed without displaying the layout and cross sectional profiles. The profiles can be saved at each step in a permanent file for subsequent processing or displayed later on a graphics device. The non-graphics mode can enhance the speed of simulation.

§ 3.2 Layout Data Data Structure

When a cut-line is specified, the layout parsing routine reads through the layout file written in CIF. CIF defines four kinds of elements: box, polygon, flash, wire. The semantics for the geometries are defined in Introduction to VLSI Systems [Mead80]. Each element of the layout is tested against the cut-line. This test is done in a general way in order to handle the cases like a slanted cut-line or rotated element. If the cut-line crosses an element, the layer name of the element and crossing points are stored in the string shown in Fig.3-2. The string is defined in C as follows.

```
#define NAME_LENGTH 5
```

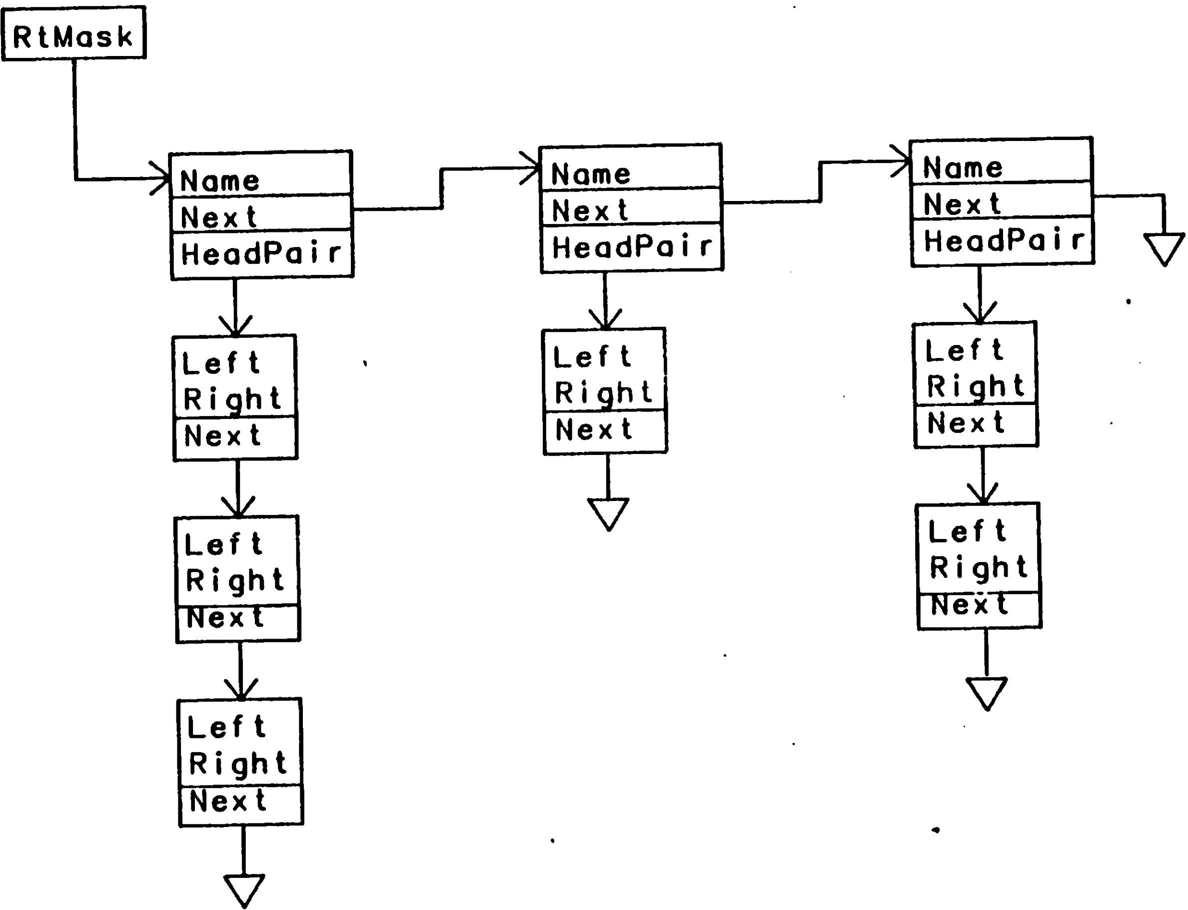


Fig.3-2 Data structure for the mask information along a cut-line.

```
typedef struct Blockpair {
    float Left;
    float Right;
    struct Blockpair *Next;
} BLOCKNODE, *BLOCKPTR;

typedef struct Masknode {
    char Name[NAME_LENGTH];
    BLOCKPTR HeadPair;
    struct Masknode *Next;
} MASKNODE, *MASKPTR;

MASKPTR        RtMask;
```

When the photo-lithography step occurs, the matching mask name is searched and the exposure is done with respect to the mask edges. Mask inversion is implemented to take into account the polarity of the photo-resist and mask as explained in § 5.2.

§ 3.3 Cross Section Data Structure

§ 3.3.1 General

To store and manipulate the cross section, SIMPL-2 defines two data structures, linked-polygonal and grid data structure.

§ 3.3.2 Linked-Polygonal Data Structure

The linked-polygonal data structure is devised to handle arbitrary shapes of materials which are formed during the process. The linked-polygonal data structure is shown in Fig.3-3 and Fig.3-4. Fig.3-3 shows a typical doubly-linked polygon string. Each node represents a polygon. It contains the name of the polygon, number of vertices, pointers to the next and previous polygons, and pointer to a vertex node of the polygon. Fig.3-4 shows the vertex string. Each node represents a vertex of a polygon and contains the coordinates of the vertex. There are three pointers to the null-terminated character strings. These character strings contain the names of the materials which are separated by the vertex. The third string is needed for the branching nodes. The three pointers at the bottom of each node point to the next vertex of the corresponding material boundary

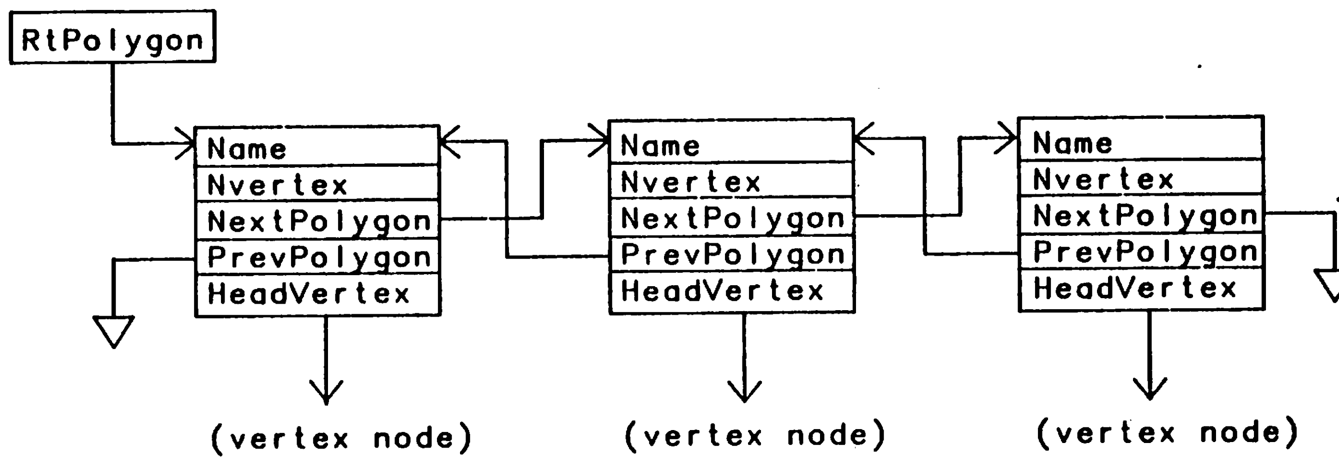


Fig.3-3 Doubly-linked polygon string for the cross section.

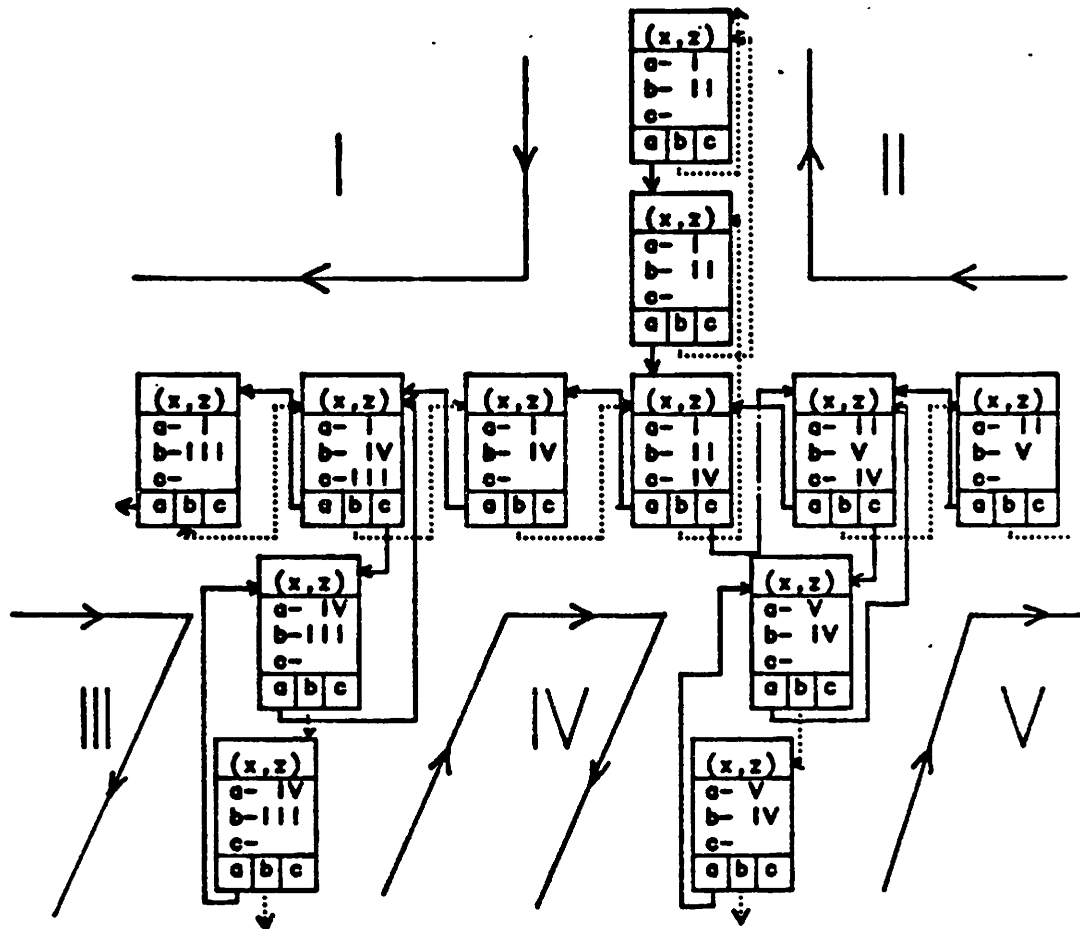


Fig.3-4 Linked polygonal data structure for the cross section.

when traversed in a clockwise fashion. In other words, you can trace the boundary of a material as follows. First, find the name of the material in the node. Then, if the name is in 'a', then go to the node pointed by the pointer 'a'. With this data structure, one can easily keep track of the relative position of materials which is essential for the physical processing. The data structure defined in C is as follows:

```
#define NAME_LENGTH 5

typedef struct Polygon {
    char Name[NAME_LENGTH];
    int Nvertex;
    struct Polygon *NextPolygon;
    struct Polygon *PrevPolygon;
    struct Vertex *HeadVertex;
} POLYGONNODE, *POLYGONPTR;

typedef struct Coordinates {
    float x;
    float z;
} COORDINATES;

typedef struct Vertex {
    struct Coordinates xz;
    char aMtrl[NAME_LENGTH];
    char bMtrl[NAME_LENGTH];
    char cMtrl[NAME_LENGTH];
    struct Vertex *a;
    struct Vertex *b;
    struct Vertex *c;
} VERTEXNODE, *VERTEXPTR;

POLYGONPTR    RtPolygon;
```

There are two special names for VERTEXNODE's. The name, 'AIR', is reserved for the nodes which are exposed to air. The other name, 'BNDR' (boundary), is written on the nodes which are located on the left and right boundary of the cross section simulation window. These two names are useful for searching for the top layer during processing. The memory locations for these polygon and vertex nodes are allocated dynamically.

§ 3.3.3 Grid Data Structure

For a non-homogeneous area such as doped silicon and oxide region, a non-uniform rectangular grid is used. The grid is generated and deleted automatically as the process evolves. SIMPL-2 initializes only a horizontal grid when a new simulation is started. During the etching, photo-resist development, and oxidation steps, new grid lines are added on the new vertices generated. And the grid lines which become void after the etching processes are deleted. The grid is described as follows in C.

```
#define NX 150
#define NZ 100

float impu[NZ][NX];
float x[NX], z[NZ];
float dx[NX-1], dz[NZ-1];
```

§ 3.4 Graphics Interface

§ 3.4.1 MFB

A device-independent graphics package, Model Frame Buffer (MFB), is employed for the graphics interface. The MFB package provides the user with a virtual graphics interface. MFB performs the terminal dependent task of encoding/decoding graphics code, thereby allowing the user to write graphics programs to execute on any graphics device.

MFBCAP is a database describing graphics terminals and is used to adapt the output of MFB to a particular terminal. Terminals are described in MFBCAP by defining a set of capabilities that each terminal has, and by describing how operations are to be performed. Currently MFBCAP entries for the terminals such as Tektronix 4014, Tektronix 4113, AED 512, AED 767, HP 2648, HP 9872, HP 150, DEC VT125 are established. SIMPL-2 has been developed on HP150 and Tektronix 4113 terminals.

§ 3.4.2 Routines for Graphics Interface

To execute SIMPL-2 in graphics mode, a file named 'SIMPL_pattern' is required. This file contains the fill pattern and color information for each material and mask layer

which will be displayed on the screen. The layers described in this file are listed with their name on both sides of the screen. The format for each layer in the file is as follows.

```
NAME NWEL
RGB  1000 1000 0
FILL 128 64 32 16 8 4 2 1
```

'NAME', 'RGB', and 'FILL' are the keywords for the material name, color code, and fill pattern, respectively. 'NAME' is limited to four characters. The three numbers following 'RGB' are the intensity of red, green, and blue colors normalized to 1000. For example, 1000 1000 1000 makes the color white. The fill pattern is defined with 8X8 bit pattern. Each decimal number followed by 'FILL' stands for the bit pattern of the corresponding row. When SIMPL-2 is executed, SIMPL-2 uses the file, SIMPL_pattern, to load the color and pattern information in an internal string.

The file, SIMPL_pattern, also contains the color information for the impurity doping concentration. A doubly linked string defined as follows in C is created when SIMPL-2 reads the file, SIMPL_pattern.

```
typedef struct doping_spectrum {
      int color;
      int fill;
      float low;
      float high;
      struct doping_spectrum *to_low;
      struct doping_spectrum *to_high;
} DOPINGNODE, *DOPINGPTR;

DOPINGPTR  RtDoping;
```

Assigning the color for every cell in the grid can be done fast with this string since the doping change is gradual in most cases except across the junctions.

CHAPTER 4 PROCESS SIMULATION

§ 4.1 General

The main difference between SIMPL-2 and other process simulators is that SIMPL-2 is capable of simulating the entire IC process including the affects of masking. While all the simulations are done by the internal simulation module in SIMPL-1 [Grim83], SIMPL-2 is designed to simulate the process with external process simulators. The basic idea is when a process step is specified, SIMPL-2 transfers the current profile to the external simulator which will simulate the process step, then SIMPL-2 takes the resultant profile back and rebuilds a new data structure accordingly.

To carry out this job, the control of the external simulators should be transferred along with the profile data. In short, SIMPL-2 acts like the controller over the simulators as well as the manager of the cross section database. SIMPL-2 can simulate deposition, oxidation, etching, ion-implantation, and photo-lithography.

§ 4.2 Internal Process Simulators

Even though the overall design of SIMPL-2 is based on the philosophy described in the previous section, SIMPL-2 maintains its own simulators due to the lack of suitable external simulators which can handle a general simulation window size. The internal process simulators are part of the program, however, these modules are separate from the SIMPL-2 itself. Thus, when an external process simulator is available for a process step, SIMPL-2 can be easily modified to incorporate the external simulator. Even though the internal process simulators are based on elementary models, they still provide good cross sections. The physical models and management of the database of the internal simulators are discussed in Chapter 5.

§ 4.3 Interaction with External Process Simulators

§ 4.3.1 Interface with SAMPLE

Among the process simulation modules, the deposition module is fully external except for the deposition with the vertical deposition option explained in § 5.3. When the deposition of a material is need during the process, SIMPL-2 extracts the top contour of the cross section as shown in Fig.4-1. SIMPL-2 builds a SAMPLE input file with the extracted profile and other parameters and then executes SAMPLE by a UNIX system call, "SYSTEM". SIMPL-2 supplies one of the two kinds of SAMPLE input parameters: one for the anisotropic deposition by sputtering source and the other for isotropic deposition. The etching machine in SAMPLE is called with negative etch rate for the isotropic deposition.

SAMPLE generates two kinds of outputs. One is the standard output which contains the input information, rough resultant profile, and the run-time statistics. The other is the string point output of the deposited contour needed by SIMPL-2. SAMPLE writes this graphics data into a file named 'f77punch7.' When the deposition is completed by SAMPLE, SIMPL-2 opens the f77punch7 file and reads the final contour. Then, the database is rebuilt with respect to the contour. To rebuild the database, first, a new POLYGON node is generated and then the contour from SAMPLE is stitched with the initial top contour to make the new polygon. The overall procedure is shown in Fig.4-1. The detailed pointer manipulation for the deposition is explained in the program source code of deposition process and is not described in this dissertation.

§ 4.3.2 Interface with the Other External Process Simulators

In this section, a systematic way to connect the external simulators is discussed. Every simulation program has its own peculiarity for its specific application. This peculiarity includes the internal data structure, input and output format. This peculiarity deters the communication between simulation tools. SIMPL-2 attempts to be the central hub which connects the simulators in an organized fashion.

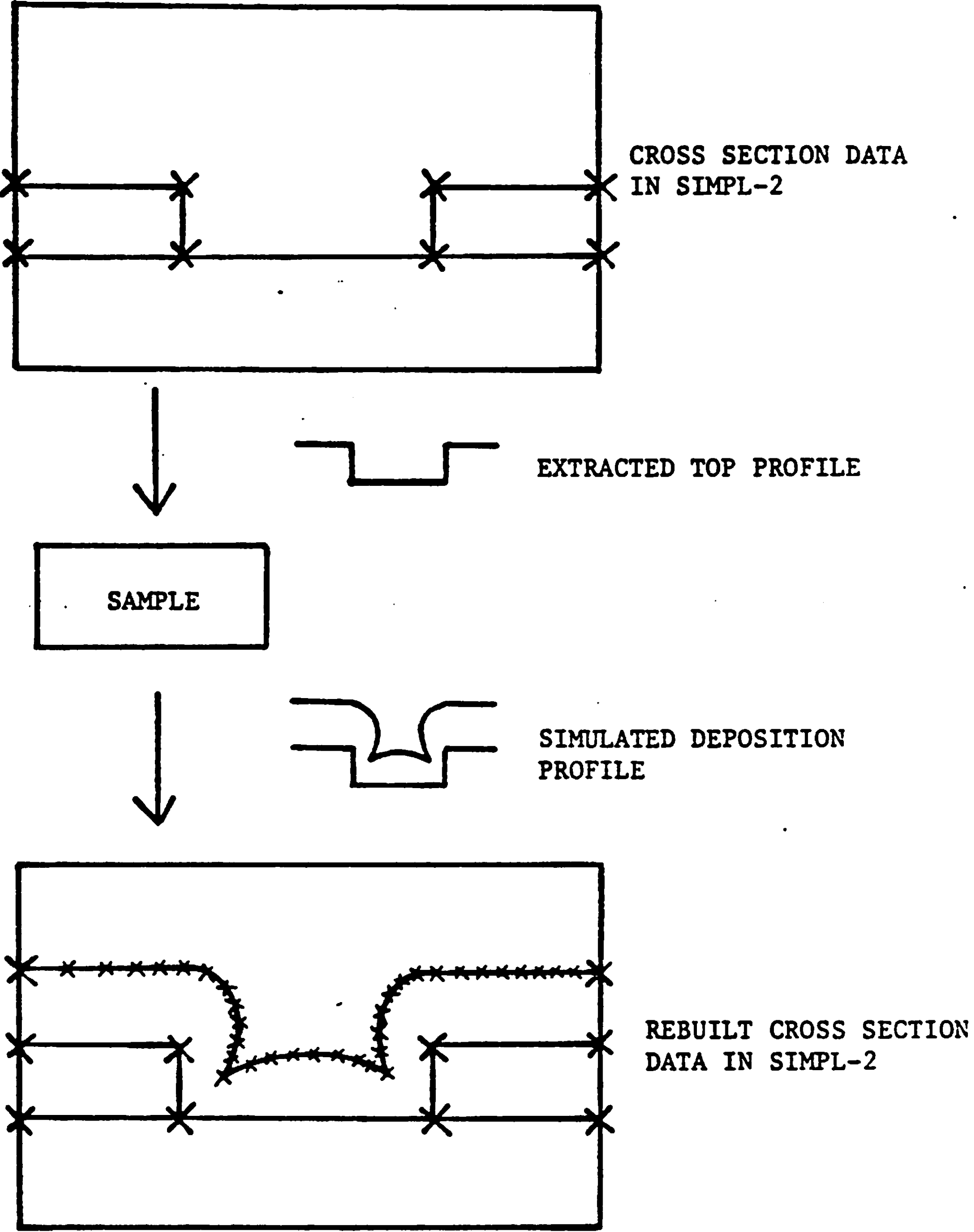


Fig.4-1 Data flow for the deposition simulation by SAMPLE.

Most of the simulation tools have grid type database or string type database. To communicate with these programs, SIMPL-2 has two kinds of data structures as discussed in § 3.3. SIMPL-2 is fully compatible with the external simulators which have the string type database such as SAMPLE. The polygons of SIMPL-2 database can be easily formed from the strings by linking them. The typical case is the deposition case in § 4.3.1.

More severe problems occur for the grid type database. Most simulation programs with grid type database have grids which are suitable for their own specific need which is to increase computational efficiency and accuracy. To connect a simulator with the grid type database, SIMPL-2 supports non-uniform rectangular grid. When the connection between SIMPL-2 and the grid type simulator is made, it is desirable to set up the grid in SIMPL-2 to resemble the grid the external simulator as much as possible. However, some data interpolation is unavoidable when the data in SIMPL-2 is to be transferred to another external simulator.

Besides the structural data stored in SIMPL-2, generally more information such as simulator commands and coefficients must be supplied to the external simulator. These can easily be handled by implementing a direct channel through SIMPL-2. The channel, however, should be exclusively designed for each external simulator.

For the kinds of data described above, it is better to transfer data through files, and not in an internal data array. The file-to-file interface is the best way to communicate. The file-to-file interface also gives the freedom to replace external simulators without complication.

CHAPTER 5 INTERNAL PROCESS SIMULATORS

§ 5.1 General

In this chapter, the internal process modules are discussed in detail. The physical models used by the internal simulation are rather primitive. However, the string and grid management algorithms which are used to implement those physical models are applicable to more elaborate physical models. Along with the physical models for the processes, the string and grid management routines are also discussed in each section.

§ 5.2 Photo-Lithography Process

§ 5.2.1 General

In a real lab environment, the photo-lithography process consists of photo-resist coating, exposure and development. SIMPL-2 simulates these three process steps using 'DEPO', 'EXPO', and 'DEVL' modules. Each process simulation step is explained in the following sections.

§ 5.2.2 Photo-Resist Coating

When a photo-lithography step occurs during a sequence of processes, photo-resist is first coated on the wafer. The photo-resist coating is carried out by the deposition module with the isotropic deposition option. In interactive mode, the photo-resist coating is done as follows.

```
WHICH PROCESS ? DEPO
NAME OF THE MATERIAL ? RST
THICKNESS OF THE MATERIAL (micro-meter) ? 1.0
ISOTROPIC, ANISOTROPIC, OR VERTICAL (I, A, or V)" ? I
DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes
```

In the example above, the photo-resist named 'RST' is coated isotropically with the thickness of $1\mu m$. The polarity of the photo-resist will be taken into consideration in the development process.

§ 5.2.3 Exposure

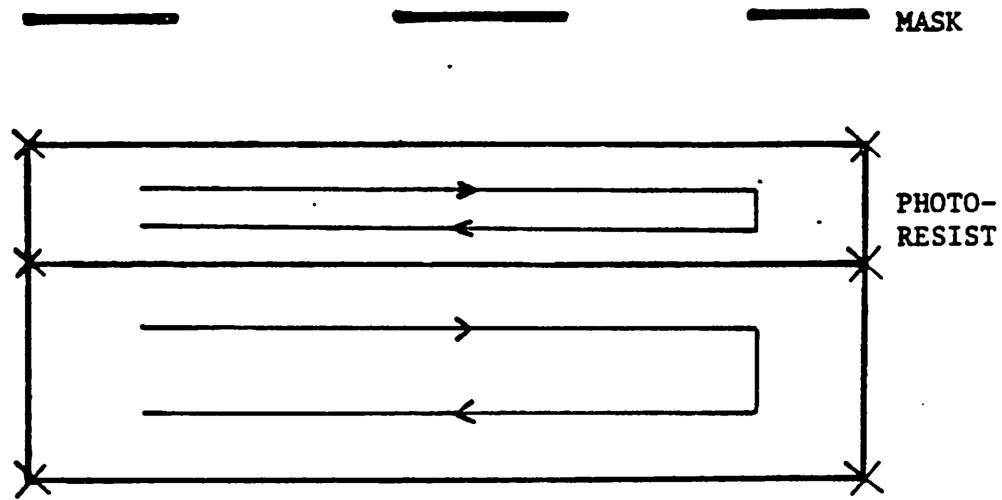
In the real physical exposure process, the exposed areas of the photo-resist to the unexposed region have a different development rate. Basically, the exposed photo-resist becomes a different material which characterizes itself when immersed into development solution. The photo-lithography machine in SAMPLE simulates the exposure and development process rigorously. However, connecting SAMPLE for the photo-lithography process is only possible by assuming the topography is locally uniform at each mask edge. This is because SAMPLE does not presently allow photo-lithography process on non-planar layers.

SIMPL-2 simulates the photo-lithography step by changing the material name for portions of the photo-resist which are exposed. In interactive mode, the exposure process is executed as follows.

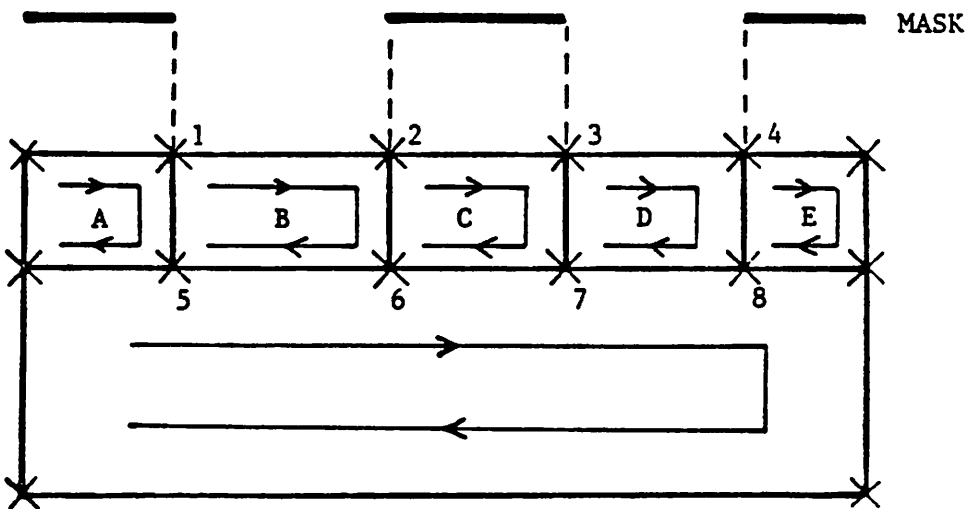
```
WHICH PROCESS ? EXPO
WHICH MASK ? NWEL
INVERT THE MASK (yes or no) ? no
NAME OF THE EXPOSED RESIST ? ERST
DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes
```

The exposure step above is a CMOS process step described in Chapter 7. The exposure of the photo-resist is done with the mask NWEL(n-well) without inversion. The exposure module searches the top layer (photo-resist) and the top polygon is divided along the mask edges. The exposed portions are named as 'ERST' (exposed-resist). This name is needed for the subsequent development process.

A simple example of string management for the exposure step is shown in Fig.5-1. The original photo-resist is split into five polygons, in this case into rectangles, A, B, C, D, and E. The eight vertex nodes, 1 through 8, are generated and inserted. For each polygon, the vertices are linked together along the arrow shown in Fig.5-1(b). The polygon A, C, and E keep the original photo-resist name, and B and D are named as specified by user, 'ERST' in the exposure example above.



(a)



(b)

Fig.5-1 Simulation of exposure process.

§ 5.2.4 Development

The development process of photo-resist is simulated by removing the appropriate polygons generated in the exposure step. The following example shows the development process in interactive mode.

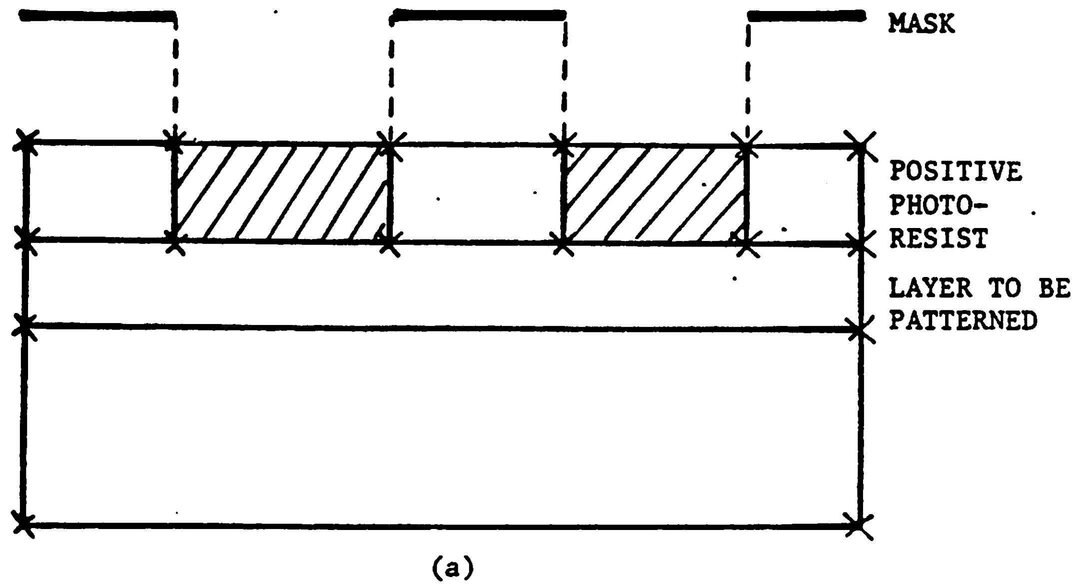
```
WHICH PROCESS ? DEVL
NAME OF THE LAYER TO BE DEVELOPED ? ERST
DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes
```

The example above is the continuation of the examples in § 5.2.2 and § 5.2.3. The polygons named 'ERST' are deleted and the reserved word, AIR is written in the vertex nodes which are exposed to air by the development process as shown in Fig.5-2. Since the photo-resist used in the example is positive resist, the exposed areas are developed away in the development process. If the photo-resist happened to be negative resist in the previous example, the answer to the second question would be 'RST' instead of 'ERST' and the unexposed areas of photo-resist would be developed away. The polarity of the photo-resist is taken in account in this way. New grid lines are added on the vertices which do not have grid lines crossing on them after the string management.

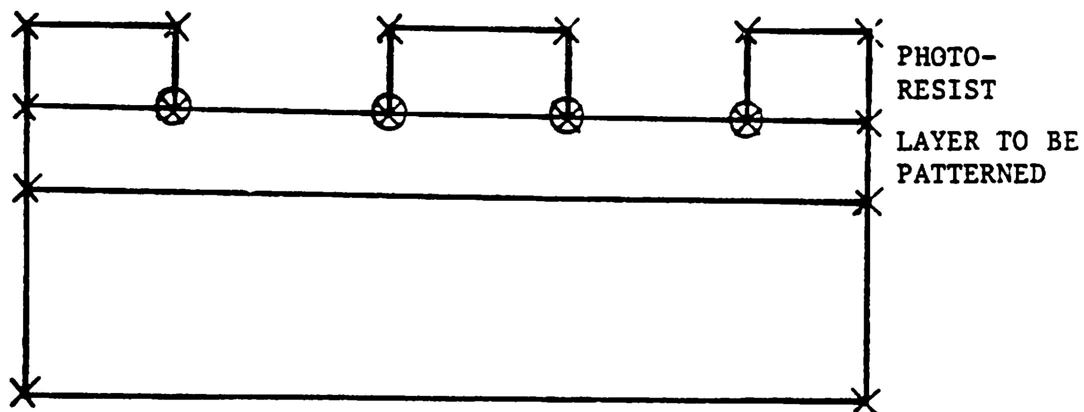
§ 5.2.5 Shortcut for the Photo-Lithography Process

When the patterned photo-resist is used as the mask for etching without bias or undercutting, the overall process can be simplified. The processes, resist coating → exposure → development → etching → resist stripping can be substituted by exposure (of the material to be etched) → etching as explained below.

Since the exposure module changes the material name of the exposed area delineated by the mask, the module can be used on the layer to be etched without using photo-resist. Now, the mask pattern is transferred directly to the layer. By simply etching the patterned layer, the same effect as the original processes can be obtained. Fig.5-3(a) and (b) show the equivalence.



(a)



⊗ VERTEX NODES EXPOSED TO THE AIR BY DEVELOPMENT PROCESS

(b)

Fig.5-2 Simulation of development process.

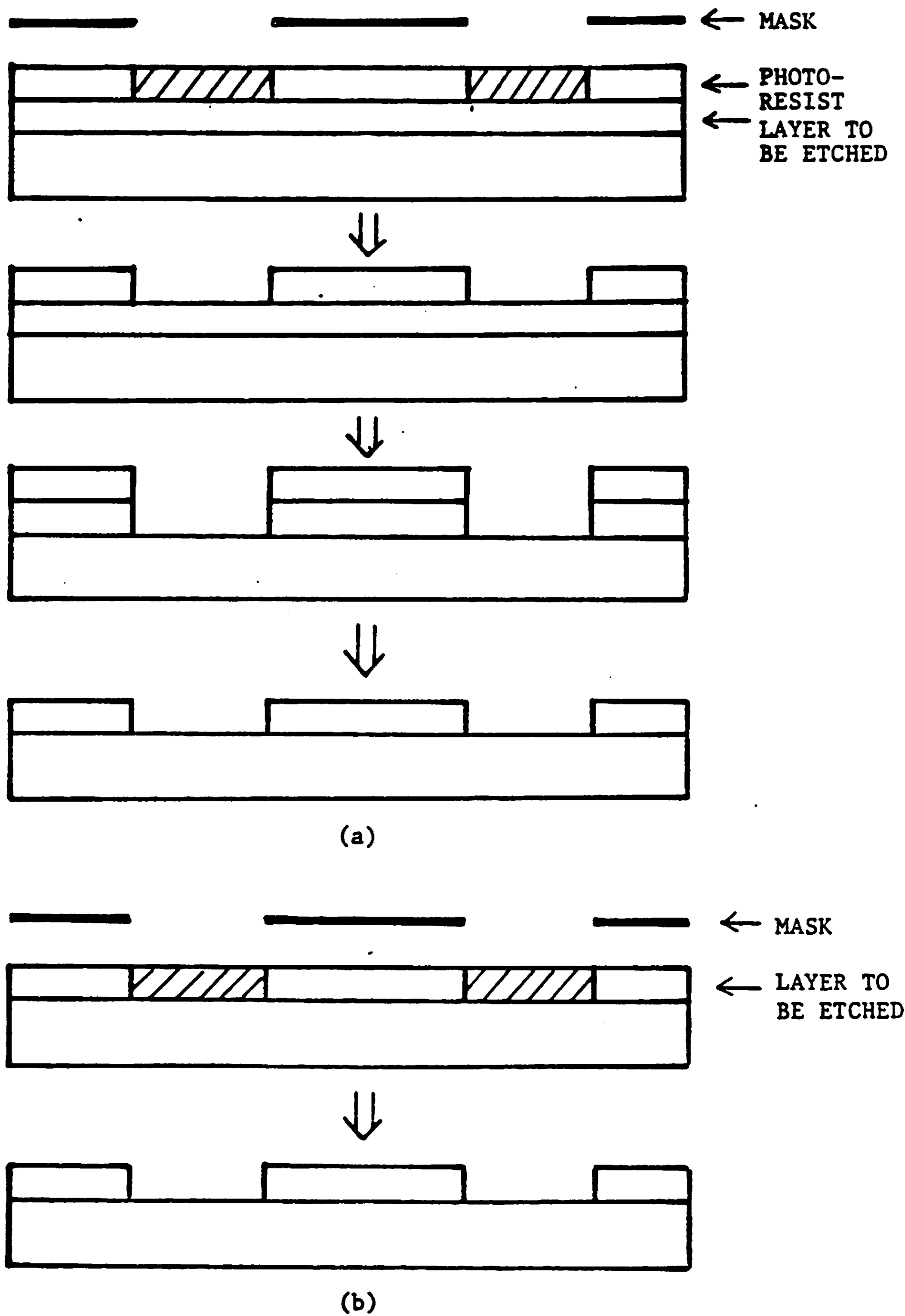


Fig.5-3 A Shortcut for the photo-lithography step.

§ 5.3 Deposition Process

When SAMPLE is not available for the deposition process, the deposition process can be simulated by the elementary internal model of SIMPL-2. The following is a deposition example with the internal model.

```
WHICH PROCESS ? DEPO
NAME OF THE MATERIAL ? NTRD
THICKNESS OF THE MATERIAL (micro-meter) ? 0.08
ISOTROPIC, ANISOTROPIC, OR VERTICAL (I, A, or V) ? V
DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes
```

In the example above, the internal deposition model is called by typing "V" for the question, "ISOTROPIC, ANISOTROPIC, OR VERTICAL (I, A, or V) ?" The specified material is deposited only vertically.

§ 5.4 Etching Process

The etching process is another process which SAMPLE is able to simulate [Reyn79]. As in the case of photo-lithography, the etching machine in SAMPLE works with planar cases which is true for small simulation windows. For a large simulation window, the cross sectional structure is planar only for first few steps of the process before the layer become non-planar. For this reason, a elementary model for the etching process has been devised. The following examples show the two kinds of etching processes.

```
WHICH PROCESS ? ETCH
WHICH LAYER DO YOU WANT ETCH ? RST
ETCH ALL (yes or no) ? yes
DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes
```

The first example shows the case of unmasked etching. The layer, 'RST', is completely etched. In this case, the string management of the polygon string and vertex string is simple. The polygons with the specified name are deleted if they are exposed to air. Then the reserved word, AIR, is written in the vertex nodes which are now exposed to air.

```
WHICH PROCESS ? ETCH
```

```
WHICH LAYER DO YOU WANT ETCH ? NTRD
ETCH ALL (yes or no) ? no
AMOUNT OF VERTICAL ETCH (micro_meter) ? 0.1
RATIO X/Z OF ETCHING (0.0 <= RATIO <= 1.0) ? 0.5
DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes
```

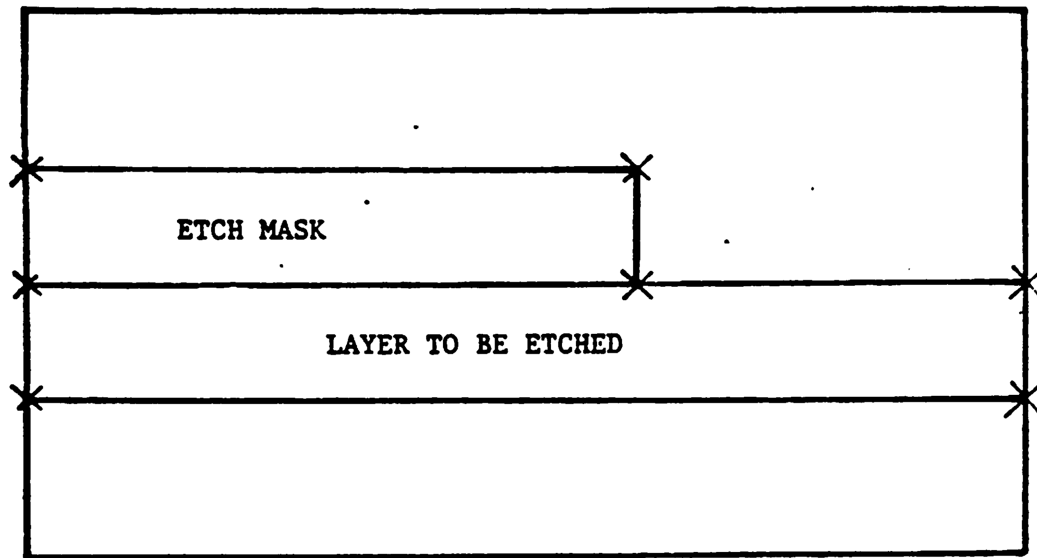
The second example is for the case where etching is done in conjunction with a mask : Fig.5-4 illustrates the case. The etching is controlled by two parameters to simulate the etching with undercut. The amount of vertical etch is defined by Z in Fig.5-4(b). The ratio X/Z describes the lateral etch. With these two parameters, the etch profile can be approximated with straight segments. The new vertex nodes, a and b in Fig.5-4(b) are inserted. New grid lines are added on the vertices which do not have grid lines crossing on them after the string management.

§ 5.5 Oxidation Process

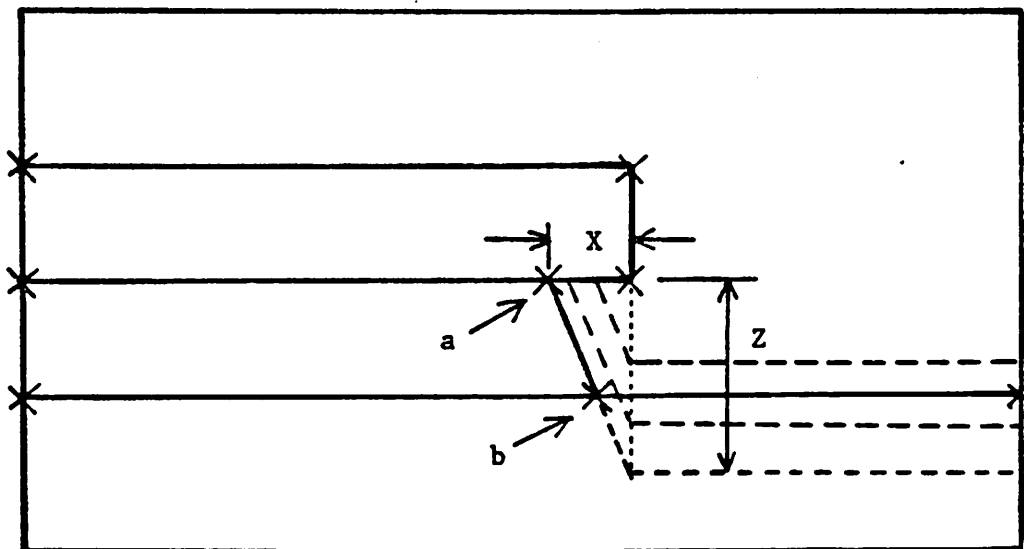
The oxidation process on a planar substrate is basically a one-dimensional process and the model for the oxidation has been reported [Deal65]. Also, the two-dimensional analyses and simulations of the oxidation process have been completed [Chin83a], [Chin83b]. The most accurate way to simulate the oxidation process would be by calling the simulation programs which are based on the models above. However, the programs currently available are only for detailed simulations in small window areas. Thus, SIMPL-2 simulates the oxidation by its own module. This module is only concerned with the resulting geometrical structure from the oxidation process. This structure can be supplied from the experiments or from rigorous simulations.

The oxidation process is carried out interactively as follows.

```
WHICH PROCESS ? OXID
OXIDE THICKNESS (micro-meter) ? 0.058
Xt (micro-meter) ? 0.058
Xe (micro-meter) ? 0.029
u1 ? 0.1
u2 ? 0.5
u3 ? 0.9
d1 ? 0.1
d2 ? 0.5
d3 ? 0.9
```



(a)



(b)

Fig.5-4 Simulation of etching process.

DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes

The OXIDE THICKNESS is the thickness of oxide resultant from the oxidation process on the bare substrate. If there is an oxide layer before oxidation. Ut and Dt in Fig.5-5 are approximated by

$$U_t = \sqrt{U_i^2+(\alpha T_{ox})^2} \quad (5.4a)$$

$$D_t = \sqrt{D_i^2+(\beta T_{ox})^2} \quad (5.4b)$$

where T_{ox} is the given OXIDE THICKNESS on the bare substrate. α and β are the fractions of oxide thicknesses grown upward and downward from the initial substrate surface, respectively.

Xt is the total bird's beak length and Xe is the length of encroachment as shown in Fig.5-5. u_1, u_2, u_3, d_1, d_2, and d_3 are defined as follows:

$$u_1 = U_1/U_t \quad (5.4c)$$

$$u_2 = U_2/U_t \quad (5.4d)$$

$$u_3 = U_3/U_t \quad (5.4e)$$

$$d_1 = D_1/D_t \quad (5.4f)$$

$$d_2 = D_2/D_t \quad (5.4g)$$

$$d_3 = D_3/D_t \quad (5.4h)$$

where U_1, U_2, U_3, D_1, D_2, and D_3 are defined in Fig.5-5.

The new vertex nodes generated during the oxidation process are marked by x's in Fig.5-5. There can be previously existing vertex nodes between these new nodes. In this case the existing nodes are moved to points calculated by linear interpolation of the two adjacent nodes. New grid lines are added on the vertices which do not have grid lines crossing on them after the string management.

§ 5.6 Ion Implantation Process

The rigorous simulations for the doping process are available [Dutt81], [Ryss80]. However, SIMPL-2 supports its own internal ion implantation module because of the limited window size of those simulators. The ion implantation process is simulated with

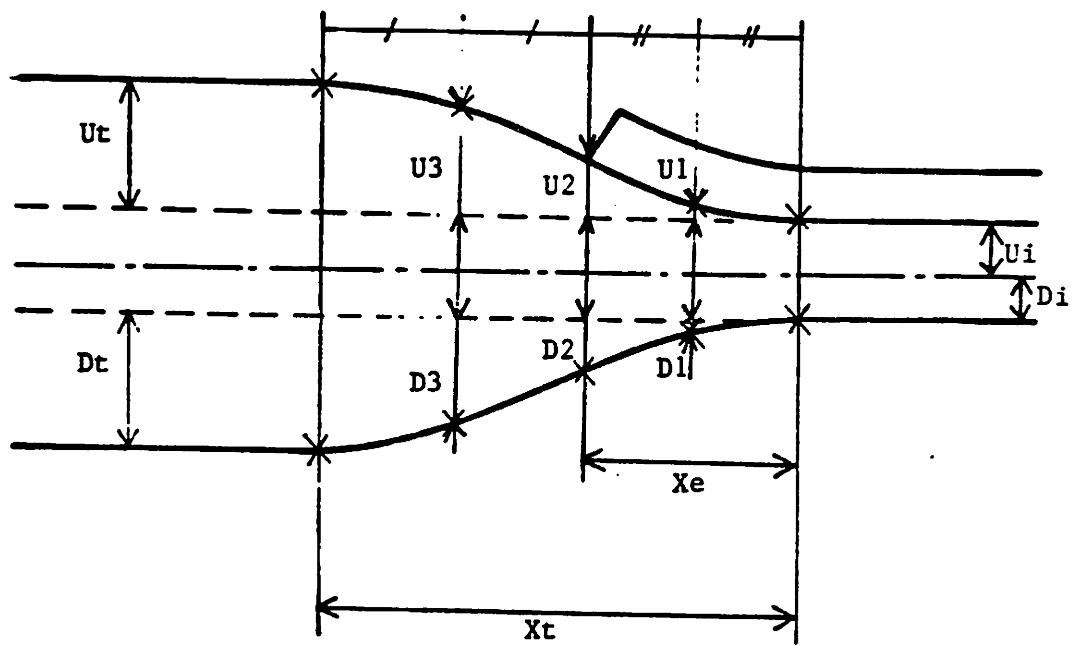


Fig.5-5 Simulation of local oxidation process.

the following information given by the user.

WHICH PROCESS ? IMPL
IMPLANTATION TYPE (p or n) ? n
DOSE (cm**(-2)) ? 1.0e+12
STANDARD DEVIATION (micro-meter) ? 1.5
PEAK DEPTH (micro-meter) ? 0.0
BLOCK THICKNESS (micro-meter) ? 0.1
DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes

The model used in SIMPL-2 is a Gaussian distribution. The implanted profile is approximated by

$$C(x) = (N / \sqrt{2\pi}\Delta R_p) exp(-(x - R_p)^2 / 2\Delta R_p^2) \quad (5.5a)$$

where N is the implanted dose, R_p is the projected range, and ΔR is the projected standard deviation. Since SIMPL-2 does not have a module for thermal annealing, the final profile after any thermal steps should be approximated by a Gaussian.

The BLOCK THICKNESS is the thickness of oxide that implanted ions cannot fully penetrate. For a given BLOCK THICKNESS, SIMPL-2 searches the areas where the bare substrate is exposed or where the oxide is thinner than the BLOCK THICKNESS. Also, SIMPL-2 assumes a complete 'block' by other materials covering the surface. Before the impurity concentration is calculated for the implanted region, vertical grid lines are added near the implantation mask edges. After the new grid is formed, the implanted dose is super-imposed on the original doping concentration.

Lateral diffusion is approximated by the same Gaussian distribution decaying radially from the implantation mask edges. This is a crude approximation to be used until a suitable two-dimensional simulator which can treat large simulation windows is available.

CHAPTER 6 FUTURE EXTENSIONS

§ 6.1 General

The first form of SIMPL-2 has been established. However, there are numerous ways to expand and improve the simulation capabilities of SIMPL-2. The main directions are connecting rigorous process simulators, device simulators, and electrical parameter extractors to SIMPL-2. Another area of interest is the improvement of the user interface.

§ 6.2 Interaction with Other Simulators

As indicated through out this dissertation, the internal process simulation modules are to be exchanged with external ones. There are some prime candidates to be linked to SIMPL-2. The photo-lithography and etching processes can be simulated by SAMPLE when SAMPLE extends its capabilities to simulate non-planar cases. SAMPLE may exploit the data structure of SIMPL-2 for the non-planar problems. The separation of SAMPLE into individual processes and the file-to-file interface are desirable [Nand84] for the connection to SIMPL-2. For two-dimensional oxidation and diffusion process simulations, programs such as BICEPS [Penu83], SOAP [Chin83b], "A Universal Two-Dimensional Process-Simulation Program [Lore84]" are good candidates. However, these programs should be able to simulate large areas, before they can optimally interact with SIMPL-2. When these connections are made, SIMPL-2 will become an indispensable tool to establish new ground rules for new integrated circuits at their initial stage of development.

Electrical parameter extraction is of great importance in the overall CAD effort for integrated circuits. Some preliminary work has been done for parasitic capacitance and resistance [Lee82a], [Lee82b], [Lee83]. The extension of these works to the level of SIMPL-2 is of great interest. The extraction of electrical characteristic of active devices is usually done on a single device configured by a designer. The automation of this pro-

cedure through SIMPL-2 will be possible by cutting a single device, when SIMPL-2 is linked to rigorous simulators.

It is desirable to set a standard format for the cross section data file used by the simulators. The standard format provides smoother file-to-file communication between the simulators. EDIF (Electronic Design Interchange Format) [EDIF85] may be extended to take into account the cross section data.

§ 6.3 User Interface

There is some room for improvements in the user interface of SIMPL-2. For now, the commands are entered by typing in the key words in graphics mode as well as in non-graphics mode. The menu-driven command input is desirable in the graphics mode. The current command window at the bottom of the screen can be used for the menu display.

SIMPL-2 understands only layout files written in non-hierarchical CIF, because the sizes of layouts used by SIMPL-2 are small enough to be written in non-hierarchical CIF. However, hierarchical parsing of CIF files is needed to simulated a portion of large layout generally.

CHAPTER 7 EXAMPLE RUNS

§ 7.1 Berkeley CMOS Inverter

§ 7.1.1 Run Sheet of the Berkeley CMOS IC

This section lists the CMOS process sequence developed by Y. Sakai, P. Carey, and K. Terrill. The simulation of cross section along a horizontal cut-line over an inverter layout with this process is presented in § 7.1.2.

THE NEW BERKELEY CMOS PROCESS

June 6, 1982 Rev.1 Yoshi Sakai

June 6, 1983 Rev.2 Paul Carey

Aug 20, 1983 Rev.3 Kyle Terrill
(Implant doses determined)

Jan 5, 1984 Rev.4 Kyle Terrill
(New PRIST technique added)

A. INITIAL WAFER PREPARATION

1. Standard Wafer Cleaning (TCA Scrub , acetone, methanol)
2. Piranha Clean 15 min.

B. INITIAL OXIDATION (1) (TCA clean furnace before using)

1. Water Break Test HF:DI/1:10
2. 1000°C Dry O2 51 min. 58 nm (push in O2)
3. Switch to N2 for 10 min.
4. Oxide Thickness Measurement ~ 500 A

C. N-WELL DEFINITION

1. Nitride Deposition 80 nm
2. Nitride thickness measurement
3. Piranha 5 min (optional)

4. Standard Photo-Lithography [Mask~ MNWL(Mask N-WeL1)]

5. Nitride Dry Etch

PLASMA THERM ELECTRODE ON TOP PLATE
POWER 100 Watts
FLOW SF6/O2 20/2
PRESSURE (plasma on) 50 mT
TEMP ~ 65° C
RATE ~ 0.022 micron per min.

6. Photoresist Removal Acetone 15 min. Piranha 15 min.

D. LOCOS 1 (TCA clean furnace before using)

1. Oxide Dip HF:DI/1:10 ~ 10 sec. to remove 40A

2. Oxidation (5min push) 1000 C steam 60 min. (0.4 um)

3. Switch to N2 for 10 min. (5min pull)

4. Oxide thickness measurement ~ 4000 A

5. Oxide Dip HF:DI/1:10 30 sec.

6. Nitride Removal Phosphoric/sulfuric Acid 20/1 155 C ~ 90 min.

(The use of a REFLUX system is required to maintain the proper temperature during etching.)

7. Oxide thickness measurement over NWELL areas (need ~ 50 nm)

8. N-Well(I) Implantation Phos/125 Kev/(1e12) (noise comp.)

9. Piranha 15 min.

10. Oxide Dip HF:DI/1:10 10 sec.

11. Drive In (TCA clean furnace before)

(7min push) 1150 C N2 16 hrs.

12. N-Well(II) Implantation Phos/200 Kev 1.5e11 (noise comp.)

13. Piranha 15 min.

E. ACTIVE REGION DEFINITION (N & P REGIONS)

1. Oxide Etch HF:DI/1:5 until surface beads then 20 sec. more

2. Initial Oxidation (II) (TCA before) 1000 C Dry O2 55min. 55 nm

3. Switch to N2 for 10 min.

4 Oxide thickness measurement ~ 550 A

5. Nitride Deposition ~ 120 nm

6. Nitride thickness measurement

7. piranha 5min (optional)

8. Standard Photo-Lithography [Mask~ MAA]

9. Nitride Dry Etching (see C. 5)

10. Photoresist Removal Acetone 15 min. Piranha 10 min.

F. P-FIELD DEFINITION

1. Standard Photo-Lithography (Mask~ NWELL)

2. P-Field Implantation Boron/25 Kev 3.5e13

3. Strip off Photoresist Acetone 15 min. Piranha 10 min.

G. BACKSIDE IMPLANT

1. Spin on frontside protection PR 3000 rpm. 30 sec.

2. Bake 120 C. 20 min.

3. Plasma Etch backside in barrel etcher

until backside nitride is gone.

SF6/O2 15 watts .1 torr 65 C .

4. BACKSIDE Implantation BF2 200kev. 2e15.

5. Remove PR Acetone 15 min . Piranha 15 min .

H. LOCOS 2 (TCA clean furnace)

1. Clean again piranha 15 min.

2. HF DIP HF/DI 1/10 10 sec.

3. 1000 C (5min push/pull) steam 230min. 0.9 um

4. Oxide thickness measurement

I. CAPACITOR DEFINITION/NPN BIPOLAR BASE DEFINITON

1. This step is not necessary.

J. Nitride Removal

1. Oxide Dip HF:DI/1:10 40 sec.

2. Nitride Removal Phosphoric Acid with 5% sulfuric at 155 C

(If some etching is not observed after 15 min then the oxide on the nitride was not fully removed.)

K. Gate Oxidation (TCA clean furnace before using)

1. Oxide Etch HF:DI/1:10 until active area is oxide free ~ 1.8 min.

2. Oxidation (I) (5min push/pull) 1000 C Dry O2 ~ 53 min. 55 nm + 10min N2 anneal

3. Oxide Etch HF:DI/1:10 until active area is oxide free ~ 1.8 min.

4. Oxidation (II) (5min push/pull) 1000 C Dry O2 ~ 53 min. 55 nm + 10min N2 anneal

5. Oxide thickness measurement (want ~ 550 A)

L. THRESHOLD IMPLANTATION Boron/30 keV 4e11 (noise comp)

M. POLY DEPOSITION

1. Oxide Dip HF:DI/1:10 to get 50 nm oxide ~ 10 sec.

2. Poly Deposition 0.39 um

3. Poly Thickness Measurement

4. Poly Doping PUSH N2 5min

O2/N2 3min

predep POCL3/O2/N2 20min

drive-in N2 15min

PULL 5min

(Do not TCA clean predep furnace. This tube may reflow at high temperatures.)

5. PSG Removal HF:DI/1:10 1 min.

6. Poly Sheet Resistance Measurement

Using Four Point Probe. (desire <25 ohm/square)

N. GATE DEFINITION (N & P Regions)

1. Standard Photo-Lithography [Mask~ MSI]

2. Poly Dry Etching USING RIE ETCHING

i) Descum 100 Watts O2 1 min

ii) Poly Etch RIE Mode

PLASMA THERM ELECTRODE ON BOTTOM PLATE

POWER	100 watts
FLOW	SF6/O2 20/2
PRESSURE (plasma on)	50 mT
TEMP.	~ 65°C
RATE	~ .4 micron per min

(This is a little fast. The rate can be reduced by decreasing the power. This needs to be characterized.)

3. Photoresist Removal Acetone 15 min., Piranha 10 min.

4. Gate Length Measurement 2.5 - 3.5 um

using the vickers

O. Oxide Regrowth for As Implantation (TCA clean furnace)

1. Clean again piranha 15 min.

2. Oxide Removal HF:DI/1:10 ~ 1.5 min. (remove 55 nm)

3. Reoxidation (TCA clean furnace)

PUSH O2 4 min

800°C Steam 18 min.

Tox on Active Region 20~ 25 nm

Tox on n+Poly Si 80~ 100 nm

P. N+ DEFINITION AND As DRIVE IN

1. Standard Photo-Lithography [Mask~ MIIN]

2. PLASMA HARDEN RESIST (Prist tech.)

i) Post bake 90°C 10 min.

ii) Descum O2 100 Watts 30°C 1 min

iii) CF4 with 4% O2 Plasma 5.5 sec., 100 mT 50 watts 65°C.

3. Hard Bake 180°C for 30 min.

(If PRIST fails, you can see reflow occur.)

(Implant should not be more than 2 hours after hard bake)

4. Implant As 80 Kev/5e15

5. Photoresist Removal

i) Ash O2 250 Watts plasma mode 140°C ~ 30 min.

ii) Piranha 15 min.

6. Oxide Dip HF:DI/1:50 5 sec.

7. Drive-In (TCA clean furnace) 5min push/pull

1000°C N2 1hr

Q. P+ DEFINITION

1. Standard Photolithgraphy [Mask~ MIIP] (inverse of MIIN)

2. PLASMA HARDEN RESIST (Prist tech.) see P. 2,3 above

3. Implantation Boron/30 Kev/3E15

4. Photoresist Removal see P. 5

R. PASSIVATION

1. Oxide Dip HF:DI/1:50 5 sec

2. Oxidation 850°C Steam 10 min. (~200A)

3. PSG CVD Deposition 0.25 um undoped Oxide

N2 11/11

O2 0.179 15 min

SiH4 0.010

0.5 um 10% Phosphorous doped PSG

N2 11/11

O2 0.179 30 min

SiH4 0.010

PH3 0.095

4. PSG Thickness Measurement (nanospect)

(This measurement is not very accurate.)

(Note appearance of PSG around poly edges.)

5. Reflow and Densification 950°C N2 15 min.

6. Look for reflow around poly edges.

S. CONTACT HOLE DEFINITION

1. Standard Photo-Lithography [Mask~ MCC]

2. PSG Etch (Wet) ~ 5-6 min (oxide free over S/D + 50sec)

(This step is the most critical in this process.)

(It is important to make sure that the contact edge does not reach the poly edge or a metal-poly short will occur.)

3. Photoresist Removal acetone 10 min Piranha 15 min.

T. METALIZATION

1. Oxide Dip BHF 1/10 20 sec.

2. Al-Si(1.0%) Sputtering

   Argon flow = 45 sccm

   Pressure = 2.8 mT

   Current = 2 amps ~ 390V

   Do 8. 15 sec exposures with 7.5 min cooling periods

   between exposures.

3. Al Thickness Measurement (desire 0.8um-1.0um)

U. METAL RUN DEFINITION

1. Standard Photo-Lithography [Mask~ MME]

2. Al Etch Aluminum Etchant Type A 45°C

3. Bake (optional) 115°C 20 min. + Si Dry Etch or

   Si Wet Etch

   (No harmful effects will occurred if Si is not removed

   as long as only 1% Si is in Al)

4. Photoresist Removal Acetone 15 min. (Heat acetone)

§ 7.1.2 Simulation by SIMPL-2

The process information data given interactively to SIMPL-2 to simulate the CMOS process in § 7.1.1 is as follows.

```
WHICH PROCESS ? OXID
OXIDE THICKNESS (micro-meter) ? 0.058
Xt (micro-meter) ? 0.058
Xe (micro-meter) ? 0.029
u1 ? 0.1
u2 ? 0.5
u3 ? 0.9
d1 ? 0.1
d2 ? 0.5
d3 ? 0.9
DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes
```

```
WHICH PROCESS ? DEPO
NAME OF THE MATERIAL ? NTRD
THICKNESS OF THE MATERIAL (micro-meter) ? 0.08
ISOTROPIC, ANISOTROPIC, OR VERTICAL (I, A, or V) ? I
DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes

WHICH PROCESS ? DEPO
NAME OF THE MATERIAL ? RST
THICKNESS OF THE MATERIAL (micro-meter) ? 1.0
ISOTROPIC, ANISOTROPIC, OR VERTICAL (I, A, or V) ? I
DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes

WHICH PROCESS ? EXPO
WHICH MASK ? NWEL
INVERT THE MASK (yes or no) ? no
NAME OF THE EXPOSED RESIST ? ERST
DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes

WHICH PROCESS ? DEVL
NAME OF THE LAYER TO BE DEVELOPED ? ERST
DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes

WHICH PROCESS ? ETCH
WHICH LAYER DO YOU WANT ETCH ? NTRD
ETCH ALL (yes or no) ? no
AMOUNT OF VERTICAL ETCH (micro_meter) ? 0.1
RATIO X/Z OF ETCHING (0.0 <= RATIO <= 1.0) ? 0.5
DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes

WHICH PROCESS ? ETCH
WHICH LAYER DO YOU WANT ETCH ? RST
ETCH ALL (yes or no) ? yes
DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes

WHICH PROCESS ? OXID
OXIDE THICKNESS (micro-meter) ? 0.4
Xt (micro-meter) ? 0.4
Xe (micro-meter) ? 0.2
u1 ? 0.1
u2 ? 0.5
u3 ? 0.9
d1 ? 0.1
d2 ? 0.5
d3 ? 0.9
DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes
```

```
WHICH PROCESS ? ETCH
WHICH LAYER DO YOU WANT ETCH ? NTRD
ETCH ALL (yes or no) ? yes
DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes


WHICH PROCESS ? IMPL
IMPLANTATION TYPE (p or n) ? n
DOSE (cm**(-2)) ? 1.0e+12
STANDARD DEVIATION (micro-meter) ? 1.5
PEAK DEPTH (micro-meter) ? 0.0
BLOCK THICKNESS (micro-meter) ? 0.1
DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes


WHICH PROCESS ? ETCH
WHICH LAYER DO YOU WANT ETCH ? OXID
ETCH ALL (yes or no) ? yes
DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes


WHICH PROCESS ? OXID
OXIDE THICKNESS (micro-meter) ? 0.055
Xt (micro-meter) ? 0.055
Xe (micro-meter) ? 0.028
u1 ? 0.1
u2 ? 0.5
u3 ? 0.9
d1 ? 0.1
d2 ? 0.5
d3 ? 0.9
DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes


WHICH PROCESS ? DEPO
NAME OF THE MATERIAL ? NTRD
THICKNESS OF THE MATERIAL (micro-meter) ? 0.12
ISOTROPIC, ANISOTROPIC, OR VERTICAL (I, A, or V) ? I
DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes


WHICH PROCESS ? EXPO
WHICH MASK ? ACTV
INVERT THE MASK (yes or no) ? no
NAME OF THE EXPOSED RESIST ? ERST
DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? no

WHICH PROCESS ? ETCH
WHICH LAYER DO YOU WANT ETCH ? ERST
ETCH ALL (yes or no) ? y
DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes


WHICH PROCESS ? DEPO
```

NAME OF THE MATERIAL ? RST
THICKNESS OF THE MATERIAL (micro-meter) ? 1.0
ISOTROPIC, ANISOTROPIC, OR VERTICAL (I, A, or V) ? I
DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes

WHICH PROCESS ? EXPO
WHICH MASK ? NWEL
INVERT THE MASK (yes or no) ? no
NAME OF THE EXPOSED RESIST ? ERST
DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes

WHICH PROCESS ? DEVL
NAME OF THE LAYER TO BE DEVELOPED ? ERST
DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes

WHICH PROCESS ? IMPL
IMPLANTATION TYPE (p or n) ? p
DOSE (cm**(-2)) ? 3.5e+13
STANDARD DEVIATION (micro-meter) ? 0.15
PEAK DEPTH (micro-meter) ? 0.05
BLOCK THICKNESS (micro-meter) ? 0.1
DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes

WHICH PROCESS ? ETCH
WHICH LAYER DO YOU WANT ETCH ? RST
ETCH ALL (yes or no) ? y
DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes

WHICH PROCESS ? OXID
OXIDE THICKNESS (micro-meter) ? 0.9
Xt (micro-meter) ? 0.9
Xe (micro-meter) ? 0.45
u1 ? 0.1
u2 ? 0.5
u3 ? 0.9
d1 ? 0.1
d2 ? 0.5
d3 ? 0.9
DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes

WHICH PROCESS ? ETCH
WHICH LAYER DO YOU WANT ETCH ? NTRD
ETCH ALL (yes or no) ? y
DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes

WHICH PROCESS ? IMPL
IMPLANTATION TYPE (p or n) ? p

DOSE (cm**(-2)) ? 4e11
STANDARD DEVIATION (micro-meter) ? 0.05
PEAK DEPTH (micro-meter) ? 0.0
BLOCK THICKNESS (micro-meter) ? 0.1
DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes

WHICH PROCESS ? DEPO
NAME OF THE MATERIAL ? POLY
THICKNESS OF THE MATERIAL (micro-meter) ? 0.39
ISOTROPIC, ANISOTROPIC, OR VERTICAL (I, A, or V) ? I
DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes

WHICH PROCESS ? EXPO
WHICH MASK ? POLY
INVERT THE MASK (yes or no) ? n
NAME OF THE EXPOSED RESIST ? ERST
DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? no

WHICH PROCESS ? DEVL
NAME OF THE LAYER TO BE DEVELOPED ? ERST
DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes

WHICH PROCESS ? DEPO
NAME OF THE MATERIAL ? RST
THICKNESS OF THE MATERIAL (micro-meter) ? 1.0
ISOTROPIC, ANISOTROPIC, OR VERTICAL (I, A, or V) ? I
DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes

WHICH PROCESS ? EXPO
WHICH MASK ? PSD
INVERT THE MASK (yes or no) ? no
NAME OF THE EXPOSED RESIST ? ERST
DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes

WHICH PROCESS ? DEVL
NAME OF THE LAYER TO BE DEVELOPED ? ERST
DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes

WHICH PROCESS ? IMPL
IMPLANTATION TYPE (p or n) ? n
DOSE (cm**(-2)) ? 5.0e+15
STANDARD DEVIATION (micro-meter) ? 0.1
PEAK DEPTH (micro-meter) ? 0.0
BLOCK THICKNESS (micro-meter) ? 0.1
DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes

WHICH PROCESS ? ETCH

```
WHICH LAYER DO YOU WANT ETCH ? RST
ETCH ALL (yes or no) ? y
DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes

WHICH PROCESS ? DEPO
NAME OF THE MATERIAL ? RST
THICKNESS OF THE MATERIAL (micro-meter) ? 1.0
ISOTROPIC, ANISOTROPIC, OR VERTICAL (I, A, or V) ? I
DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes

WHICH PROCESS ? EXPO
WHICH MASK ? PSD
INVERT THE MASK (yes or no) ? y
NAME OF THE EXPOSED RESIST ? ERST
DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes

WHICH PROCESS ? DEVL
NAME OF THE LAYER TO BE DEVELOPED ? ERST
DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes

WHICH PROCESS ? IMPL
IMPLANTATION TYPE (p or n) ? p
DOSE (cm**(-2)) ? 3.0e+15
STANDARD DEVIATION (micro-meter) ? 0.1
PEAK DEPTH (micro-meter) ? 0.0
BLOCK THICKNESS (micro-meter) ? 0.1
DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes

WHICH PROCESS ? ETCH
WHICH LAYER DO YOU WANT ETCH ? RST
ETCH ALL (yes or no) ? y
DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes

WHICH PROCESS ? DEPO
NAME OF THE MATERIAL ? PSG
THICKNESS OF THE MATERIAL (micro-meter) ? 0.75
ISOTROPIC, ANISOTROPIC, OR VERTICAL (I, A, or V) ? I
DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes

WHICH PROCESS ? EXPO
WHICH MASK ? CONT
INVERT THE MASK (yes or no) ? yes
NAME OF THE EXPOSED RESIST ? ERST
DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes

WHICH PROCESS ? DEVL
```

```
NAME OF THE LAYER TO BE DEVELOPED ? ERST
DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes


WHICH PROCESS ? ETCH
WHICH LAYER DO YOU WANT ETCH ? OXID
ETCH ALL (yes or no) ? no
AMOUNT OF VERTICAL ETCH (micro_meter) ? 0.700000
RATIO X/Z OF ETCHING (0.0 <= RATIO <= 1.0) ? 0.000000
DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes


WHICH PROCESS ? DEPO
NAME OF THE MATERIAL ? METL
THICKNESS OF THE MATERIAL (micro-meter) ? 0.8
ISOTROPIC, ANISOTROPIC, OR VERTICAL (I, A, or V) ? A
SPUTTERING SOURCE ANGLE (degrees) ? 45.0
DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes


WHICH PROCESS ? EXPO
WHICH MASK ? CONT
INVERT THE MASK (yes or no) ? yes
NAME OF THE EXPOSED RESIST ? ERST
DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes


WHICH PROCESS ? DEVL
NAME OF THE LAYER TO BE DEVELOPED ? ERST
DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes


WHICH PROCESS ? QUIT
```

Fig.7-1 shows the layout of a inverter and the $50\mu m$-long cut-line on the layout. The simulated cross-section is shown in the bottom half of Fig.7-1. Fig.7-1 was taken from a Tektronix 4113 color graphics terminal. It took 8 minutes of CPU time in graphics mode and 5 minutes in non-graphics mode to generate the cross section of the CMOS inverter on VAX11/780 with UNIX 4.2BSD.

Fig.7-2 shows the profile along the $10\mu m$-long cut-line around n-channel source region on the same CMOS layout. The 'bird's-beak', lateral diffusion, and aluminum deposited by sputtering are clearly shown.

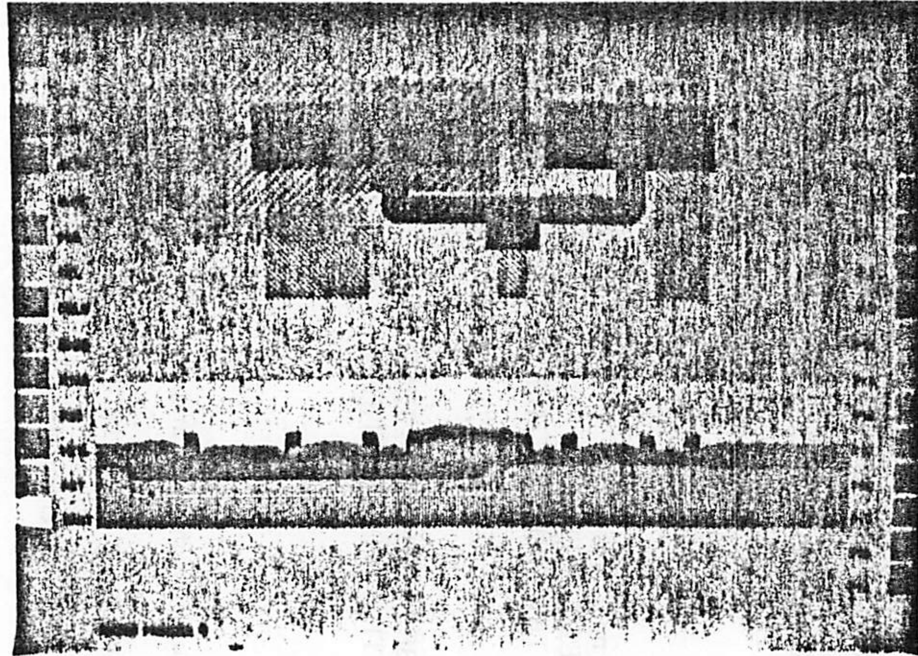


Fig.7-1 Layout of a CMOS inverter and the cross sectional structure along the $50\mu m$-long cut-line simulated by SIMPL-2.

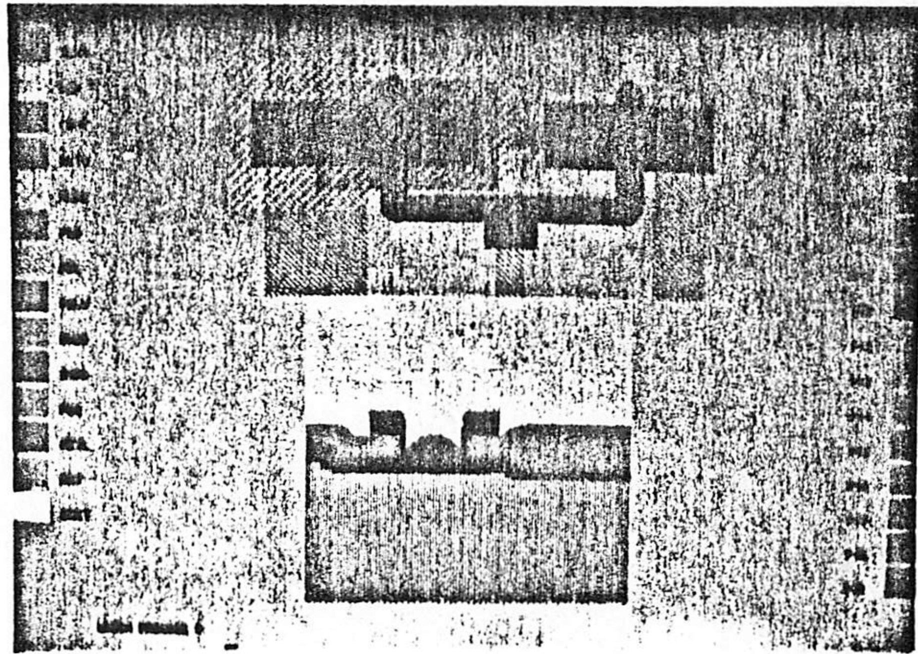


Fig.7-2 Layout of a CMOS inverter and the cross sectional structure along the 10μm-long cut-line around the N-channel source simulated by SIMPL-2.

§ 7.2 DRAM Cells

The profile of a NMOS DRAM cell along the 20μm-long cut-line is shown in Fig.7-3. The metal line across the profile is the bit line and the left polysilicon is the word line. The polysilicon on the right side of the cross section makes the storage capacitance.

Fig.7-4 shows the layout of CMOS DRAM cells and cross section along the 7.92μm-long diagonal cut-line drawn on the layout. This example has double level polysilicon and double level metal process. On the cross section, the first level metal makes contact with the second level polysilicon on the left and the second level metal is connected to the first level metal through the via on the right.

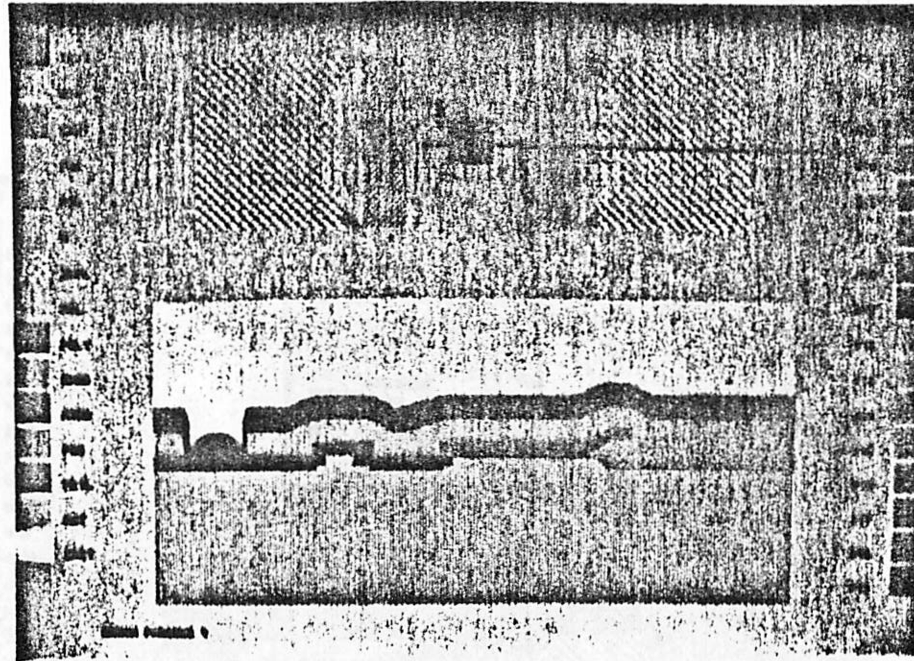


Fig.7-3 Layout of a NMOS DRAM cells and the cross sectional structure along the $20\mu m$-long cut-line.

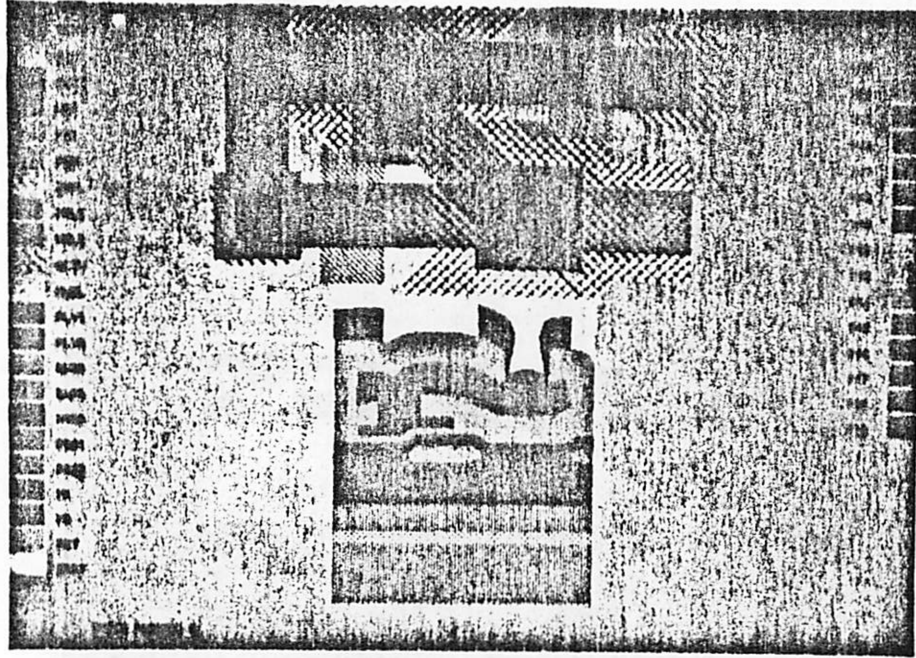


Fig.7-4 Layout of a CMOS DRAM cells and the cross sectional structure along the 7.92μm-long diagonal cut-line.

References

[Bill83] G.C. Billingsley, "Program Reference for KIC," Memorandum no. UCB/ERL M83/62, Electronics Research Laboratory, U.C. Berkeley, Oct. 1983.

[Chin82] D. Chin, M.R. Kump, H.-G. Lee, and R.W. Dutton, "Process Design Using Two-Dimensional Process and Device Simulations," IEEE Trans. Electron Devices, vol. ED-29, no. 2, pp. 336-340, Feb. 1982.

[Chin83a] D. Chin, S.Y. Oh, S.M. Hu, R.W. Dutton, and J.L. Moll, "Two-Dimensional Oxidation," IEEE Trans. Electron Devices, vol. ED-30, no. 7, pp. 744-749, July 1983.

[Chin83b] D. Chin, S.Y. Oh, and R.W. Dutton, "A General Solution Method for Two-Dimensional Nonplanar Oxidation," IEEE Trans. Electron Devices, vol. ED-30, no. 9, pp. 993-998, Sept. 1983.

[Deal65] B.E. Deal and A.S. Grove, "General Relationship for the Thermal Oxidation of Silicon," J. Appl. Phys., vol. 36, no. 12, Dec. 1965.

[Dutt81] R.W. Dutton and S.E. Hansen, "Process Modeling of Integrated Circuit Device Technology," Proceedings of the IEEE, vol. 69, no. 10, pp. 1305-1320, Oct. 1981.

[EDIF85] Electronic Design Interchange Format Steering Committee, "EDIF, Electornic Design Interchange Format, Version 1 0 0," The EDIF Users' Group, Design Automation Department, Texas Instruments, P.O. Box 225474, MS3668, Dallas, Texas 75265., 1985.

[Gree80] J.A. Greenfield and R.W. Dutton, "Nonplanar VLSI Device Analysis Using the

Solution of Poisson's Equation." IEEE Trans. Electron Devices, vol. ED-27, no. 8, pp. 1520-1532, Aug. 1980.

[Grim83] M.A. Grimm, K. Lee, and A.R. Neureuther, "SIMPL-1 (SIMulated Profiles from the Layout - version 1)," International Electron Device Meeting, Dec. 1983.

[Kell81] K.H. Keller, "KIC, A Graphics Editor for Integrated Circuits," Masters Report, Department of Electrical Engineering and Computer Science, University of California, Berkeley, CA, June 1981.

[Kern78] B.W. Kernighan and D.M. Ritchie, "The C Programming Language," Prentice-Hall, Englewood Cliffs, New Jersey, 1978.

[Kopp83] G.M. Koppelman, and M.A. Wesley, "OYSTER: A Study of Integrated Circuits as Three-Dimensional Structures," IBM J. Res. Develop., vol. 27, no. 2, Mar. 1983.

[Lee82a] K. Lee, Y. Sakai and A.R. Neureuther, "Topography Dependent Step Coverage Resistance Simulation for VLSI Design," 1982 Symposium on VLSI Technology, Proceedings pp. 61-62, Oiso, Japan, Sept. 1982.

[Lee82b] K. Lee, Y. Sakai, and A.R. Neureuther "Topography Dependent Electrical Parameter Simulation," International Electron Device Meeting, Dec. 1982.

[Lee83] K. Lee, Y. Sakai, and A.R. Neureuther, "Topography-Dependent Electrical Parameter Simulation for VLSI Design," IEEE Trans. Electron Devices, vol. ED-30, no. 11, pp. 1469-1474, Nov. 1983.

[Lore84] J. Lorenz. K. Haberger. G. Prinke. H. Ryssel. A. Seidel. J. Pelka. A. Sacks. and M. Svoboda. "A Universal Two-Dimensional Process-Simulation Program." Second Conference on Numerical Simulation of VLSI Devices. Boston. MA.. Nov. 1984.

[Mead80] C.A Mead and L.A Conway. "Introduction to VLSI Systems." Addison-Wesley. 1980.

[Nand84] S.N. Nandgaonkar. "A Family of Simulation Programs for IC Fabrication Processes (Their Structure. Design and Implementation)." Memorandum no. UCB/ERL M84/90. Electronics Research Laboratory. University of California. Berkeley. CA 94720. Oct. 1984.

[Oldh79] W. G. Oldham. S. N. Nandgankar. A. R. Neureuther. and M. O'Toole. "A General Simulator for VLSI Lithography and Etching Processes: PartI-Application to Projection Lithography." IEEE Trans. Electron Devices. vol. ED-26. no. 4. pp. 717-722. Apr. 1979.

[Oldh80] W. G. Oldham. A. R. Neureuther. C. Sung. J. L. Reynolds and S. N. Nandgaonkar. "A General Simulator for VLSI Lithography and Etching Processes: PartII-Application to Deposition and Etching." IEEE Trans. Electron Devices. vol. ED-27. no. 8. pp. 1455-1459. Aug. 1980.

[Penu83] "A Comprehensive Two-Dimensional VLSI Process Simulation Program. BICEPS." B.R. Penumalli. IEEE Trans. Electron Devices. vol. ED-30. no. 9. pp. 986-992. Sept. 1983.

[Reyn79] J.L. Reynolds. A.R. Neureuther. and W.G. Oldham. "Simulation of Dry Etched

Line Edge Profiles." J. Vac. Sci. Technology. 16(6). Nov./Dec. 1979.

[Ryss80] H. Ryssel. K. Haberger. K. Hoffmann. G. Prinke. R. Duemcke. and A. Sachs. "Simulation of Doping Processes." IEEE Trans. Electron Devices. vol. ED-27. no. 8. pp. 1484-1492. Aug. 1980.

[Selb80] S. Selberherr. A. Schuetz. H.W. Poetzl. "MINIMOS - a Two-Dimensional MOS Transistor Analyzer." IEEE Trans. Electron Devices. vol. ED-27. no. 8. pp. 1540-1550. Aug. 1980.

SIMPL-2 User Guide

1. Introduction

SIMPL-2 (SIMulated Profiles from the Layout - version 2) is a computer-aided-design tool for integrated circuits which automatically simulates the cross sectional view of integrated circuits along an arbitrary 'cut-line' drawn on the layout. For a given layout, SIMPL-2 extracts the mask information from the intersections of the layout geometries with the specified cut-line. Using this layout information, SIMPL-2 generates the evolution of cross sectional structures according to the process steps given to SIMPL-2 interactively or in batch mode.

The device profile from the composite process can be generated rapidly using elementary internal physical process models. More rigorous external process simulators can also be invoked through an interface for profile data and transfer of control. As of now, the SAMPLE program is linked to SIMPL-2 to simulate the deposition process.

The first version of SIMPL-2 consists of 12,500 lines of C and runs on VAX11/780 with UNIX 4.2BSD operating system.

2. How to load SIMPL-2

The MFB package and SAMPLE program are necessary to run SIMPL-2 and some modifications should be made according to the system before SIMPL-2 is loaded. Since SIMPL-2 uses MFB package for its graphics interface, SIMPL-2 should be loaded together with MFB. The line in the 'makefile' of SIMPL-2 which specifies the path to MFB may be modified. The line is currently

```
MFB = /oc/cad/lib/mfb.a
```

Also the line in mfb.h

```
#define DEFAULTMFBCAP "~cad/lib/mfbcap"
```

should be modified if 'mfbcap' is located somewhere else.

There is another line to be modified depending on the system. As described above, the SAMPLE program is called for the deposition process (except for the vertical deposition option). So the line,

```
system("/od/sample/samsoft/release/ucb/sample.out <SAMPLE_Input >
SAMPLE_Output");
```

in the source file 'Deposition.c' should be modified according to the location of SAMPLE.

Now, SIMPL-2 is ready to be compiled and loaded.

```
% make
```

compiles the source codes of SIMPL-2 and load the object files. An executable file 'simpl' is generated by the command.

3. How to run SIMPL-2

3.1 Preparing 'MFBCAP'

When the file 'mfbcap' does not contain an entry for the graphics terminal. a new entry can be written with the key words specified in MFBCAP(5) in the UNIX Programmer's Manual. The terminal should support the 'replace (jam) mode' to run SIMPL-2.

3.2 Preparing 'SIMPL_pattern'

'SIMPL_pattern' file is required to execute SIMPL-2. This file has the pattern and color information for each material and mask layer which will be drawn on the screen. All the layers described in this file are listed with their name on both sides of the screen. The format for each layer in the file is as follows.

```
NAME NWEL
RGB  1000 1000 0
FILL 128 64 32 16 8 4 2 1
```

'NAME'. 'RGB'. and 'FILL' are the keywords for the material name. color code. and fill pattern. respectively. 'NAME' is limited to four characters. The three numbers following 'RGB' are the intensity of red. green. and blue colors normalized to 1000. For example. 1000 1000 1000 makes the color white. The fill pattern is defined with 8X8 bit pattern. Each decimal number followed by 'FILL' stands for the bit pattern of the corresponding row. When SIMPL-2 is executed. SIMPL-2 uses the file. SIMPL_pattern. to load the color and pattern information in an internal string.

The file. SIMPL_pattern. also has the color information for the impurity doping concentration written in the same format. Here is an example.

```
NAME NP
RGB 600 500 500
FILL 255 255 255 255 255 255 255 255
NAME N12
RGB 0 200 0
FILL 136  0  0  0 34  0  0  0
NAME N15
RGB 0 200 0
FILL 170  85 170  85 170  85 170  85
NAME N18
RGB 0 200 0
FILL 255 255 255 238 255 255 255 238
NAME P12
RGB 400 40 0
FILL 136  0  0  0 34  0  0  0
NAME P15
RGB 400 40 0
FILL 170  85 170  85 170  85 170  85
NAME P18
RGB 400 40 0
FILL 255 255 255 238 255 255 255 238
```

It begins with the name 'NP' which is intrinsic substrate (not exactly intrinsic: NP covers the doping concentration less than the lowest concentration specified.) Following 'NP', the pattern information for the n-type should come. The number after 'N' means the exponent of doping concentration. For the above example, 'N15' covers 10^{15} to 10^{18} and 'N18' covers 10^{18} and above. After Nxx's, the p-type color spectrum is defined in the same way. This pattern information for the doping concentration should come after those for all the other cross section and layout layers.

3.3 Commands

SIMPL-2 can be invoked by typing

% simpl -n

or

% simpl -(terminal type)

The first command runs SIMPL-2 in non-graphics mode and the second runs SIMPL-2 in graphics mode. In non-graphics mode, SIMPL-2 runs exactly the same way as in graphics mode except that no graphics output is drawn on the screen. The terminal type is specified by a key word in the MFBCAP file. For example, one of the key words t3, T3, 4113D, or T4113D can be used for Tektroix 4113 color graphics terminal.

When SIMPL-2 is invoked, it searches the 'SIMPL_pattern' file in the present working directory and displays the patterns in the file on both sides of the screen. The layers for the layout and cross section are displayed on the left and color spectrum of doping concentration on the right.

After the initialization described above, SIMPL-2 prompts questions on the command window at the bottom. The first question is

LAYOUT FILE NAME ?

A CIF file name may be given for this question and SIMPL-2 draws the layout on the upper half screen. The next question is

START A NEW SIMULATION ? (yes or no)

When 'y(es)' is typed, SIMPL-2 asks for a location of cut-line by giving the graphics cursor on the screen. For a specified cut-line, the mask information is stored in the file 'Mask_Edges'. A substrate for the cross section is initialized by answering the following questions.

SUBSTRATE TYPE ? (p or n)
DOPING CONCENTRATION (cm**(-3)) ?

If 'n(o)' is typed to the question, "START A NEW SIMULATION ? (yes or no)", SIMPL-2 asks the savefile name which contains the cross section data which was generated by a previous run. At this time the 'Mask_Edges' file is read and the cut-line is drawn on the layout. Thus, the 'Mask_Edges' file should have the corresponding cut-line and mask information with the savefile.

Before SIMPL-2 draws the cross section, it asks

VERTICAL DIMENSION SCALE FACTOR ?

The vertical scaling is needed since the vertical dimensions are usually smaller than the horizontal dimension. A scaling factor between 2 and 3 are usually used.

Now, SIMPL-2 is ready to simulate the process steps. On the command line at the bottom,

WHICH PROCESS ?

is displayed. The available answers to these questions are

DEPO
DEVL
ETCH
EXPO
IMPL
NEW_CUT
NEW_LAYOUT
OPEN_PC, CLOSE_PC
OXID
PROCESS_FILE
REDRAW
SAVE
WAIT
QUIT

In the rest of this section, each command is explained in alphabetic order.

*** DEPO ***

This is the command for the deposition process. This command is followed by the questions below.

WHICH PROCESS ? DEPO
NAME OF THE MATERIAL ? NTRD
THICKNESS OF THE MATERIAL (micro-meter) ? 0.08
ISOTROPIC, ANISOTROPIC, OR VERTICAL (I, A, or V) ? I
DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes

The name of the material to be deposited should be different from that of any material which is exposed to air before the deposition process. The deposition process is carried out by the SAMPLE program when I or A is given for the question "ISOTROPIC, ANISOTROPIC, OR VERTICAL (I, A, or V) ?" For the isotropic deposition, SIMPL-2 calls the etching machine of SAMPLE with a negative etch rate and for the anisotropic deposition SIMPL-2 calls the deposition machine with the sputtering source. For the anisotropic deposition the following question is asked.

SPUTTERING SOURCE ANGLE (degrees) ?

The source angle is the angle between the center of the source and the edge of it seen from a point on the wafer.

When DEPO is done by SAMPLE, three new files, 'SAMPLE_Input', 'SAMPLE_Output', and 'f77punch7' are generated in the present working directory. SAMPLE_Input is the input file for SAMPLE, SAMPLE_Output is the standard output of SAMPLE, and f77punch7 is the string points output for plot. The f77punch7 is used to rebuild the data base of SIMPL-2.

When V is given for the question "ISOTROPIC, ANISOTROPIC, OR VERTICAL (I, A, or V) ?", the material is deposited only vertically without running SAMPLE.

*** DEVL ***

This is the command for the development process in photo-lithography. The development process of photo-resist is simulated by removing the appropriate polygons generated by the exposure step. In interactive mode, the development process is carried out as follow.

WHICH PROCESS ? DEVL

NAME OF THE LAYER TO BE DEVELOPED ? ERST

DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes

For the example above, the 'ERST' layers which are exposed to air are removed.

***** ETCH *****

This is the command for the etching process. The etching process is simulated interactively as follows.

WHICH PROCESS ? ETCH

WHICH LAYER DO YOU WANT ETCH ? RST

ETCH ALL (yes or no) ? yes

DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes

The first example shows the case of unmasked etching. The layer, 'RST', is totally etched. The polygons with the specified name are deleted if they are exposed to air.

WHICH PROCESS ? ETCH

WHICH LAYER DO YOU WANT ETCH ? NTRD

ETCH ALL (yes or no) ? no

AMOUNT OF VERTICAL ETCH (micro_meter) ? 0.1

RATIO X/Z OF ETCHING (0.0 <= RATIO <= 1.0) ? 0.5

DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes

The second example is for the case where etching is done in conjunction with a mask. The etching is controlled by two parameters to simulate the etching with undercut.

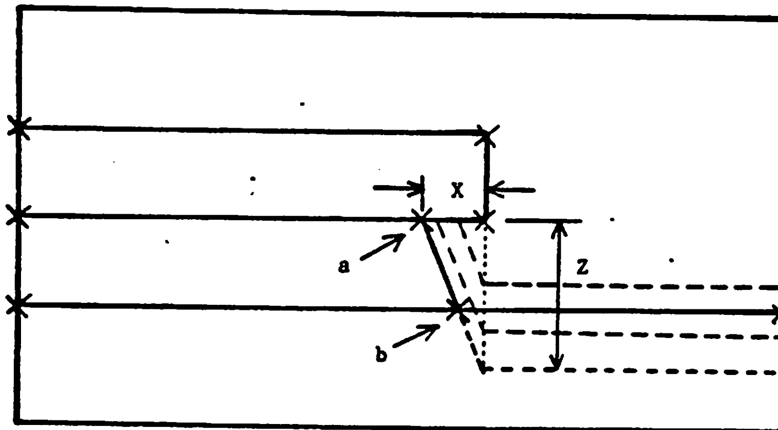


Fig.3-1

The amount of vertical etch, Z, and the lateral etch ratio, X/Z, are defined in Fig.3-1. With these two parameters, the etch profile can be approximated with a straight segment.

***** EXPO *****

This is the command for the exposure process in photo-lithography. SIMPL-2 approximates the photo-lithography step by changing the material name for portions of the photo-resist which are exposed. In interactive mode, the exposure process is executed as follows.

WHICH PROCESS ? EXPO
WHICH MASK ? NWEL
INVERT THE MASK (yes or no) ? no
NAME OF THE EXPOSED RESIST ? ERST
DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes

The inverted mask is used to expose the photo-resist if 'y(es)' is typed for the question 'INVERT THE MASK (yes or no) ?'.

***** IMPL *****

This is the command for the ion-implantation process. The ion implantation process is simulated with the following information given by the user.

```
WHICH PROCESS ? IMPL
IMPLANTATION TYPE (p or n) ? n
DOSE (cm**(-2)) ? 1.0e+12
STANDARD DEVIATION (micro-meter) ? 1.5
PEAK DEPTH (micro-meter) ? 0.0
BLOCK THICKNESS (micro-meter) ? 0.1
DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes
```

The model used in SIMPL-2 is a Gaussian distribution. The implanted profile is approximated by

$$C(x) = (N / \sqrt{2\pi}\Delta R_p) exp(-(x - R_p)^2 / 2\Delta R_p^2)$$

where N is the implanted dose. R_p is the projected range. and ΔR is the projected standard deviation. Since SIMPL-2 does not have a module for the thermal annealing. the final profile after any thermal steps should be approximated by a Gaussian.

The BLOCK THICKNESS is the thickness of oxide that implanted ions cannot fully penetrate. For a given BLOCK THICKNESS. SIMPL-2 searches the areas where the bare substrate is exposed or the oxide is thinner than the BLOCK THICKNESS. Also. SIMPL-2 assumes a complete 'block' by other materials covering the surface. Before the impurity concentration is calculated for the implanted region. vertical grid lines are added near the implantation mask edges. After the new grid is formed. the implanted dose is super-imposed on the original doping concentration.

The lateral diffusion is approximated by the same Gaussian distribution decaying radially on the implantation mask edges. This is a crude approximation to be used until a suitable two-dimensional simulator which can treat large simulation windows is available.

***** NEW_CUT *****

When this command is typed for the question 'WHICH PROCESS ?', SIMPL-2 stops the current simulation and prompts the graphics cursor for a new cut-line in graphics mode or asks the coordinates of cut-line in non-graphics mode.

***** NEW_LAYOUT *****

This command is used to call a new layout. The question,

LAYOUT FILE NAME ?

follows this command.

***** OPEN_PC, CLOSE_PC *****

These are commands to prepare a process file for the batch-mode run. When 'OPEN_PC' is typed for the question 'WHICH PROCESS ?', SIMPL-2 asks

PROCESS FILE NAME TO BE WRITTEN ?

After a file name is given, every question given by SIMPL-2 and the user's answer are stored in the designated file until 'CLOSE_PC' is typed for the question 'WHICH PROCESS ?'. 'CLOSE_PC' command puts 'END' in the process file at the end. This file is to be used to run SIMPL-2 in batch-mode by the command 'PROCESS_FILE'.

***** OXID *****

This is the command for the oxidation process. The oxidation process is carried out interactively as follows.

WHICH PROCESS ? OXID
OXIDE THICKNESS (micro-meter) ? 0.058
Xt (micro-meter) ? 0.058
Xe (micro-meter) ? 0.029
u1 ? 0.1
u2 ? 0.5
u3 ? 0.9
d1 ? 0.1
d2 ? 0.5
d3 ? 0.9
DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes

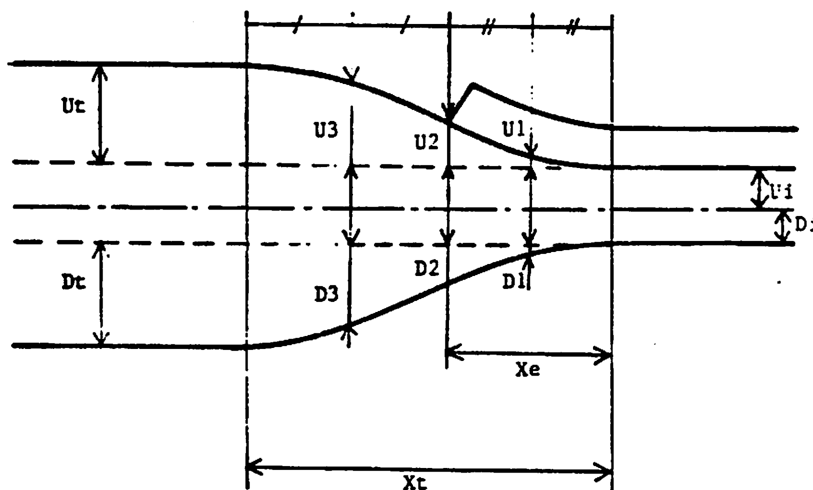


Fig.3-2

The OXIDE THICKNESS is the thickness on the bare substrate. If there is oxide layer before oxidation Ut and Dt in Fig.3-2 are approximated by

$$U_t = \sqrt{U_i^2 + (\alpha T_{ox})^2}$$

$$D_t = \sqrt{D_i^2 + (\beta T_{ox})^2}$$

where T_{ox} is the given OXIDE THICKNESS on the bare substrate. α and β are the fractions of oxide thicknesses grown upward and downward from the initial substrate surface, respectively.

Xt is the total bird's beak length and Xe is the length of encroachment as shown in Fig.3-2. u_1, u_2, u_3, d_1, d_2, and d_3 are defined as follows.

$$u_1 = U_1 / U_t$$

$$u_2 = U_2 / U_t$$

$$u_3 = U_3 / U_t$$

$$d_1 = D_1 / D_t$$

$$d_2 = D_2 / D_t$$

$$d_3 = D_3 / D_t$$

where U_1, U_2, U_3, D_1, D_2, and D_3 are defined in Fig.3-2.

***** PROCESS_FILE *****

When this command is typed for the question 'WHICH PROCESS ?', SIMPL-2 asks

PROCESS FILE NAME ?

SIMPL-2 does the process simulation according to the file until 'END' is read in the file. This process file may be written by the commands, 'OPEN_PC' and 'CLOSE_PC'.

***** REDRAW *****

This command is used to redraw the cross section with a different vertical scaling factor. The question

VERTICAL DIMENSION SCALE FACTOR ?

follows to set a new scaling factor.

*** SAVE ***

This command saves the current cross section data in a file. The file name is given to SIMPL-2 by answering the question

SAVEFILE NAME ?

which comes after the 'SAVE' command. The savefile is to be used to display the cross section or to continue the process later.

*** WAIT ***

When this command is given. SIMPL-2 sleeps until the 'return' key is hit.

*** QUIT ***

This command terminates the session.

4. Example

This section presents an example layout and the first few process steps for the layout. The following is the CIF description of a CMOS inverter. When this file is given to SIMPL-2, the layout is drawn on the upper half screen of a terminal.

```
(CIF file of symbol hierarchy rooted at cmos.k);
DS 1 1 1;
9 cmos.k;
L PSD;
B 1400 850 -800 -25;
L ACTV;
B 1300 500 900 0;
B 1750 500 -1125 0;
L POLY;
B 400 200 0 -800;
W 200 -900 350 -900 -600 900 -600 900 350;
L MTL;
B 400 400 0 -700;
B 400 400 -500 0;
B 400 400 500 0;
B 600 200 0 0;
B 800 2500 -1500 -50;
B 200 400 0 -1100;
B 400 2500 1300 -50;
B 200 1100 0 650;
L CONT;
B 200 200 -500 0;
B 600 200 -1500 0;
B 200 200 500 0;
B 200 200 1300 0;
B 200 200 0 -700;
L NWEL;
B 2200 1200 -1100 0;
DF;
C 1;
E
```

When a cut-line is given, the mask infomation is stored in the file, 'Mask_Edges.' For example, if the coordinates of cut-line end points are (-25.0 ,0.0), (25.0 ,0.0) (unit : μm), the stored data in the file, 'Mask_Edges', is as follows.

cut line : -2500 0 2500 0

number of masks = 6

PSD 1

10.000 24.000

ACTV 2

5.000 22.500

27.500 40.500

POLY 2

15.000 17.000

33.000 35.000

MTL 3

6.000 14.000

18.000 32.000

36.000 40.000

CONT 4

7.000 13.000

19.000 21.000

29.000 31.000

37.000 39.000

NWEL 1

3.000 25.000

The process sequence of the CMOS inverter can be carried out step by step with this mask information. The process inputs up to the local oxidation for N-well formation are given below as an example.

WHICH PROCESS ? OXID

OXIDE THICKNESS (micro-meter) ? 0.058

Xt (micro-meter) ? 0.058

Xe (micro-meter) ? 0.029

u1 ? 0.1

u2 ? 0.5

u3 ? 0.9

d1 ? 0.1

d2 ? 0.5

d3 ? 0.9

DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes

WHICH PROCESS ? DEPO

NAME OF THE MATERIAL ? NTRD

THICKNESS OF THE MATERIAL (micro-meter) ? 0.08

ISOTROPIC, ANISOTROPIC, OR VERTICAL (I, A, or V) ? I

DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes

WHICH PROCESS ? DEPO

NAME OF THE MATERIAL ? RST

THICKNESS OF THE MATERIAL (micro-meter) ? 1.0

ISOTROPIC, ANISOTROPIC, OR VERTICAL (I, A, or V) ? I

DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes

WHICH PROCESS ? EXPO

WHICH MASK ? NWEL

INVERT THE MASK (yes or no) ? no

NAME OF THE EXPOSED RESIST ? ERST

DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes

WHICH PROCESS ? DEVL

NAME OF THE LAYER TO BE DEVELOPED ? ERST

DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes

WHICH PROCESS ? ETCH

WHICH LAYER DO YOU WANT ETCH ? NTRD

ETCH ALL (yes or no) ? no

AMOUNT OF VERTICAL ETCH (micro_meter) ? 0.1

RATIO X/Z OF ETCHING (0.0 <= RATIO <= 1.0) ? 0.5

DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes

WHICH PROCESS ? ETCH

WHICH LAYER DO YOU WANT ETCH ? RST

ETCH ALL (yes or no) ? yes

DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes

WHICH PROCESS ? OXID

OXIDE THICKNESS (micro-meter) ? 0.4

Xt (micro-meter) ? 0.4

Xe (micro-meter) ? 0.2

u1 ? 0.1

u2 ? 0.5

u3 ? 0.9

d1 ? 0.1

d2 ? 0.5

d3 ? 0.9

DO YOU WANT TO DRAW THE CROSS SECTION (yes or no) ? yes

5. SIMPL-2 Structure

This section contains the names of functions in the source files.

CIF_Actions.c :

ABeginCall(SymbolNum)
ABeginSymbol(SymbolNum,A,B)
ABox(Length,Width,X,Y,XDirection,YDirection)
AComment(Text)
ADeleteSymbol(SymbolNum)
AEnd()
AEndCall()
AEndSymbol()
ALayer(Technology,Mask)
APolygon(Path)
ARoundFlash(Width,X,Y)
AT(Type,X,Y)
AUserExtension(Digit,Text)
AWire(Width,Path)

CIF_Parser.c :

PBox()
PCIF(CIFFileName,StatusString,StatusInt, WhichAction)
PCall()
PCharacter(WhiteSpaceControl,EOFControl)
PComment()
PDeleteSymbol()
PEnd()
PError(PErrorMessage)
PInteger(WhiteSpaceControl,EOFControl)
PLayer()
PLookAhead(For)
PPath(Path)
PPoint(X,Y)
PPolygon()

PPrimitiveCommand()
PRoundFlash()
PSymbol()
PUserExtension()
PWhiteSpace(WhiteSpaceControl,EOFControl)
PWire()

Deposition.c :

Deposition()
Join_Depo(left, right, left_v, right_vertex)
Read_f77punch7(top, bottom, pn_top, pn_bottom)
Rebuild_Depo()
Run_SAMPLE()
Stitch()
Vertical_Depo()
Work_Out_Top()
Write_SAMPLE_Input(type)
Write_Top(fp)
float Find_ztop()

Develop.c :

Develop()

Do_Process.c :

Do_Process()
Draw()

Etching.c :

Do_Etch()
Etch_All()
Etch_Case1()
Etch_Case2()

Etch_Case3()

Etch_Case4()

Etching()

Get_Etching_Info()

Scan_Etch()

Set_LT_RT(pLT, pRT)

Set_lt()

Set_rt()

Exposure.c :

Expo_Case1(left, right)

Expo_Case2(left, right)

Expo_Case3(left, right)

Expo_Case4(left, right)

Exposure()

Get_Corners()

Invert_Mask(pp_pair, p_pair)

MASKPTR Get_Mask()

React(left, right)

Set_Current_Polygon()

Set_Resist_Name()

Grid.c :

Add_Column(xx)

Add_Grid()

Add_Row(zz)

Delete_Row()

Flush_Grid()

Get_num_x()

Get_num_z()

Is_There_x(xx)

Is_There_z(zz)

Remove_Dense_Row()

Set_dx_dz()

float Get_Top(name)

Implantation.c :

Get_Implant_Info()
Impl_Set_left_right()
Implant()
Implantation()
Left_Add_Columns()
Left_Diffusion()
Right_Add_Columns()
Right_Diffusion()
Set_r(p_r)
Vertical_Implant()
float Gaussian(x. total. deviation. peak)

Initialize.c :

Initialize_Cutline_Profile()
Initialize_Extern()
Initialize_Grid()
Initialize_Layout()
Initialize_String()
New_Initialize()
Old_Initialize()
Parse_Command_Line(argc. argv)

Mask_Info.c :

Add_Block(x1. x2)
Cut_Circle(x. y. r. pc1. pc2)
Cut_Polygon(path. pp_cut_pairs)
Cut_Rectangle(x. y. pc1. pc2)
Get_Mask_Info()
MBox(Length.Width.X.Y.XDirection.YDirection)
MLayer(Technology.Mask)

MPolygon(Path)
MRoundFlash(Width,X,Y)
MWire(Width,Path)
Read_Mask_Info()
TRS(px, py)
Write_Mask_Edges()

Misc.c :

Answer_Float(answer)
Answer_Float_e(answer)
Answer_Integer(answer)
Answer_String(answer)
Ask(s)
Exit(s)
Message(s)
Round(x)
float Interpolate(x1, x2, y1, y2, xx)
float Power(x, n)

Oxidation.c :

Append_Thick(x, t)
Get_Oxidation_Info()
Make_One_Polygon()
Oxid_Case1()
Oxid_Case2()
Oxid_Case3()
Oxid_Case4()
Oxid_Set_left_right()
Oxidation()
Oxidize()
Set_Up_Left()
Set_Up_Right()
float Original_Thick(x)

Read_Data.c :

Read_Data()
Read_Grid(fp)
Set_Data_Structure(fp)
Set_Polygon(fp, n)
Set_Vertex(p, fp, n)
WriteAIR()

SIMPL.c :

main(argc, argv)

SIMPL_MFB.c :

Clear_Screen()
Color_Id(name)
DBox(Length,Width,X,Y,XDirection,YDirection)
DLayer(Technology,Mask)
DPolygon(Path)
DRoundFlash(Width,X,Y)
DWire(Width,Path)
Draw_Cross_Section()
Draw_Cross_Section_AIR()
Draw_Cross_Section_Grid()
Draw_Cross_Section_Polygons()
Draw_Cutline(x1, y1, x2, y2)
Draw_Layout()
Draw_Mask(p)
Draw_Patterns()
Exist_Color_Fill(name)
Free_color(p)
Free_fill(p)
Get_Draw_Cutline(px1, py1, px2, py2)
Get_Layout_Size()
Get_Pattern_Info()

Gprintf(s)
Gscanf(s)
Init_current_d()
Initialize_MFB()
Make_Doping_Spectrum()
Point(pointX.pointY.Key.Mask)
SetDebounceTime()
Set_Color_Fill(name)
Set_Cross_Section_z_Scale()
Set_Cursor_Color()
Set_Doping_Color_Fill(x)
Strlen(s)
Viewport_Allocation()
WBox(Length.Width.X.Y.XDirection.YDirection)
WLayer(Technology.Mask)
WPolygon(Path)
WRoundFlash(Width.X.Y)
WWire(Width.Path)

String.c :

AIRize_String(p. q. name)
Append_Polygon(p)
Count_Polygons(Rt_p)
Count_Vertices(polygon)
Delete_Mtrl(name. p)
Delete_Polygon(p_polygon)
Flush_Vertex(p)
Free_Block(p)
Free_Mask(p)
Free_Polygon(p)
Full(p)
Insert(x. p. q. name)
Insert_Polygon(p_polygon. prev_polygon)
Inside_Polygon(new_polygon. Polygon. x. z)
Is_There(name.p)

Link(p, name, q)

Move(PPVertex,Name)

Move_Backward(PPVertex,Name)

One_D_Search_or_Insert(ppvertex, name, x_position)

POLYGONPTR Find_Polygon(p, name)

Rename(from_vertex, to_vertex, from_name, to_name)

Separate_Polygon(p_polygon)

Substitute(p, from_string, to_string)

Two_D_Search_or_Insert(pp_vertex, x1, z1, x2, z2, p, q, name)

Which_Node(p, pp, name)

Write_a_AIR(p)

float Thickness(p, name)

struct Vertex *Get_LeftTop()

struct Vertex *Get_RightTop()

Write_Data.c :

Write_savefile()