

NOTICE WARNING CONCERNING COPYRIGHT RESTRICTIONS:

The copyright law of the United States (title 17, U.S. Code) governs the making of photocopies or other reproductions of copyrighted material. Any copying of this document without permission of its author may be prohibited by law.

**EXTENSIONS TO AN INTEGRATED
CIRCUIT MANUFACTURING PROCESS SIMULATOR**

by

Sani R. Nassif

DRC-01-10-82

April, 1982

EXTENSIONS TO AN INTEGRATED CIRCUIT MANUFACTURING PROCESS SIMULATOR

Sani R. Nassif

**Department of Electrical Engineering
Carnegie-Mellon University
Pittsburgh, PA 15213**

Abstract

A simulator for the bipolar integrated circuit manufacturing process, *FABRICS*, written at CMU, was extended to include MOS manufacturing operations. Interfaces to circuit analysis programs, and an interactive user interface, were added.

Submitted in partial fulfillment of the requirements for the degree of Master of Science.

UNIVERSITY LIBRARIES
CARNegie-MELLON UNIVERSITY
PITTSBURGH, PENNSYLVANIA 15213

TABLE OF CONTENTS

INTRODUCTION	2
1. BASIC METHODOLOGY OF SIMULATION	3
1.1 Introduction	3
1.2 Statistical Simulation	3
1.3 Models	5
1.3.1 Decomposing the Process model	5
1.3.2 Types of models	6
1.3.3 Mixed Model	7
1.3.4 Basic Model Structure	9
1.4 Disturbances	10
1.5 Applications of Simulation Methodology, FABRICS	11
1.5.1 Structure of FABRICS	11
1.5.2 Implementation	11
2. EXTENSIONS FOR MOS PROCESS SIMULATION	14
2.1 Introduction	14
2.2 Ion-Implantation Model	14
2.2.1 Background	14
2.2.2 Model Implemented	17
2.2.2.1 Implantation	17
2.2.2.2 Annealing	18
2.2.2.3 Redistribution during subsequent high-temperature processes	18
2.3 Extending the Photolithography Model	21
2.3.1 A Model for Equivalent Dimensions	22
2.3.1.1 Photolithography	23
2.3.1.2 Corners	23
2.3.1.3 Sidewalls of Diffused Resistors	24
2.3.2 A Model for Contact Resistance to Diffused Impurity Layers	25
2.4 Model for Oxidation	27
3. INTERFACES TO FABRICS	29
3.1 Introduction	29
3.2 The Design Environment	29
3.3 Interface to Circuit Simulators	30
3.3.1 Possible modes of simulation	30
3.3.2 Interface implemented	32
3.4 User Interface	33
3.4.1 INPUT Interface	33
3.4.2 OUTPUT Interface	34

4. APPLICATIONS, LIMITATIONS AND FUTURE EXPANSION	35
4.1 Introduction	35
4.2 Applications	35
4.3 Limitations	36
4.4 Further Extensions and Possibilities	36
Appendix A Tables	38
A.1 Approximation to Corners	38
A.2 Approximation to Penetration data	39
A.3 Constants of Oxidation Model	40

LIST OF FIGURES

Figure 1-1:	Basic model structure	9
Figure 1-2:	Multi-Level RNG	10
Figure 1-3:	Structure of FABRICS	12
Figure 1-4:	DATA flow diagram of FABRICS	13
Figure 2-1:	Comparison between theory and experiments (after [Furukawa 74])	16
Figure 2-2:	Definition of R_p and σ_p	16
Figure 2-3:	A typical resistor	21
Figure 2-4:	Equivalent Resistor	22
Figure 2-5:	Illustrating the effect of photolithography disturbances	23
Figure 2-6:	Equivalent length of corners, ([Hall 67]).	24
Figure 2-7:	Sidewalls of Diffused Resistors	25
Figure 3-1:	A possible design environment	30
Figure 3-2:	Interface for a chip-by-chip mode	31
Figure 3-3:	Interface for a mean-and-statistics mode	32
Figure 4-1:	Possible scheme for a complete MOS design environment	37

ACKNOWLEDGEMENT

I would like to acknowledge the help and assistance, as well as guidance and, perhaps most importantly, patience, of my advisor Dr. Steve Director. Invaluable coaching and encouragement came from Andrzej Strojwas, to whom I am deeply indebted.

Thanks also to Stephen J. Conroy, Carole M. Johnson and Frank J. Sweeney, who helped with programming parts of *FABRICS*.

INTRODUCTION

As integrated circuits grow in complexity and size, there is an urgent need for CAD tools to help the designer in his task. The designer needs, in addition to tools that help him in layout and design-rule checking, tools to simulate circuit response and performance. These tools require, of course, knowledge of element parameters on the IC. In most design work, a knowledge of the *nominal* values of these parameters is sufficient.

If, however, the designer requires information about the *statistical distribution* of IC parameters; as would be required, for example, to optimize yield, then a better understanding of the process is required. In particular, a tool is needed that can relate the random disturbances in the process to the variations in element parameters, in a meaningful way. Such a tool is may be realised as a *Statistical Process Simulator* capable of simulating a large number of ICs, while accounting for the differences between them due to the random disturbances in the process that were mentioned above.

In the first chapter, the basic methodology of simulation is described. *FABRICS*, a simulator of the bipolar IC manufacturing process, is also presented. The processes required to simulate the MOS manufacturing process, along with the models implemented, are described in Chapter 2. Chapter 3 describes the interface between *FABRICS* and the circuit simulator *SPICE* [Nagel 75], as well as the interactive user interface which was developed for *FABRICS* to facilitate its use. Chapter 4 outlines some applications and limitations of the simulator, concluding with possible directions for further work.

CHAPTER 1

BASIC METHODOLOGY OF SIMULATION

1.1 Introduction

This chapter describes the basic model structure required for statistical simulation, as well as the important features that are peculiar to the semiconductor manufacturing processes, and useful when considering the task of simulation. The application of the methodology to FABRICS is discussed.

In this and the following chapters, where meaningful, we will use upper case letters to denote *Random Variables*, while lower case letters will be used to represent a sample of that random variable, e.g. G is a random variable and g is a sample of G . Subscripts on values of random variables (e.g. g_3 , g_n) will be used for different samples of the random variable.

1.2 Statistical Simulation

A Statistical Simulator [Maly 79] must generate sets of IC parameters, which fluctuate according to simulated disturbances in the process. These sets x_1, x_2, \dots, x_n can be thought of as samples of a random variable X . The sets x_i may contain values of IC elements as well as inline measurements, and are a function of:

- z_1 a vector of process parameters (e.g. times and temperatures in diffusion furnaces)
- z_2 a vector of layout parameters (e.g. mask dimensions for the various devices)
- d_1 a vector of *Random Disturbances* (e.g. disturbances due to crystal defects)
- X a vector of constants (e.g. Boltzmann's constant k)

Thus, x_i may be expressed as:

$$x_i = P(z_1, z_2, d_i, X) \quad (1.1)$$

Where P is a model of the process, and will be discussed in the next section. The generation of disturbances d_i is considered in section 1.4.

The accuracy of a statistical simulator depends on how well the Joint Probability Density Function, *JPDF*, of X (a random variable, values of which constitute the set x_i) relates to the *JPDF* of an equivalent set X^* measured on the real manufacturing line.

Inaccuracy arises due to three factors:

- i. *Inaccuracy of modeling*, meaning inaccuracy due to the inability of the model to account correctly for all physical effects. This may be due to simplifications in the model itself, or because the process itself is not well characterised.
- ii. *Inaccuracy of identification*, which is inaccuracy in model calculations due to the inaccuracy in the *identification of model parameters*. This, as we shall mention later, sometimes counteracts the added accuracy of a complicated model due to the larger number of parameters required.
- iii. *Inaccuracy of solving*, whenever numerical methods are used to solve equations, there is an inherent inaccuracy due to the finite number of steps involved, for example, in integration.

Since the statistical approach to simulation requires the generation of a large set of data, then we can expect the model P to be solved many times in the course of one simulation run. For this approach to be viable, the process model must be economical, meaning solvable within a small CPU time.

1.3 Models

1.3.1 Decomposing the Process model

In general, P in eq. (1.1) is a collection of sets of partial differential equations modeling the process. These equations are usually complex and highly non-linear. SUPREM [Antoniadis 78] is a process simulator which employs advanced numerical algorithms to solve such equations. This approach, however, is not viable for statistical simulation. In section 1.2 we stipulated that the statistical simulator must generate a large set of IC element parameter data, which is related to the disturbances in the process. Having to solve complicated sets of equations repeatedly to do such a simulation will result in prohibitively high CPU computation cost. Hence the need for some other approach to statistical process simulation.

Towards this end, let us consider simplifying the solution of P by decomposing it into two separate models:

1. A model of element and inline parameters:

$$M_E(\mu, X, X) = 0 \quad (1.2)$$

Where μ is a set of *physical* parameters resulting from the process, e.g. concentration profiles and window dimensions.

2. A model for physical parameters resulting from the process:

$$M_P(z_1, z_2, d, X) = 0 \quad (1.3)$$

Where z_1 , z_2 , d , and X are as defined for eq. (1.1). The reason that equation (1.3) is not written in the form:

$$\mu = M_P(z_1, z_2, d, X) \quad (1.4)$$

is that the outcome of each manufacturing step depends on the various manufacturing steps performed before it, e.g. base diffusion depends on the epitaxial layer parameters.

The advantage of such a decomposition lies in the simpler models involved.

This is because relating impurity profiles to process conditions, and device parameters to impurity profiles, is a lot easier than relating device parameters to process conditions directly. Considering the case of diffusion, models for impurity profiles in terms of process conditions and some constants are readily available [Cave 65]. The same applies for models of, say transistors, in terms of impurity profiles (see for example [Chou 71]). Note, however, that we have not defined M_p and M_n yet, and that, in the general case, these functions may also be sets of partial differential equations, and thus, would still require a significant computational effort. In the following section we consider possible alternatives to expressing models in the form of sets of partial differential equations.

1.3.2 Types of models

Consider now a model of the form:

$$X \ll Fte^{z_2, D, A} \quad (1.5)$$

where F is an *analytical* expression which approximates the solution of P , an example is to approximate the profile of implanted impurities in silicon by a gaussian distribution, as in section 2.2.2.1; but in reality, the profile is gaussian only near the peak of the distribution. This kind of model would, at best, approximate P over some limited range of process conditions. Improving its accuracy usually means considering higher order (i.e. less significant) effects, with the corresponding need for more parameters, which, as noted in section 1.2, may not increase over-all accuracy.

Although a model in the form of eq. (1.5) is computationally economical, in the sense that it does not require the solution of differential equations, it may not be accurate enough for our purposes. In order to achieve such a computational advantage, while trying to maintain accuracy, consider an approximation to the solution of the model P of the form:

$$X = A_0 \cdot A^{\wedge} \cdot A_2 2A_j \cdot _ \quad (1.6)$$

where:

$$z = (z_1, z_2)^T$$

and thus,

$$x_i \ll a_{jj} \cdot a^z \cdot a^i z (a^i)^T \cdot ,, \quad (1.7)$$

Where the A matrices are now random variables, varying in a way to simulate the disturbances in the process.

In theory, a model such as eq. (1.6) can be fitted to any relation between X and z. But accuracy can only be increased by increasing the order of eq. (1.6), and, consequently, more parameters would have to be identified before simulation can be performed. One must also note, at this point, that the A matrices do not have any real physical meaning, that is, they cannot be directly related to measurable physical quantities. This distinction makes identification of the parameters that will control the generation of these random variables difficult, since it will have to be done using polynomial fitting techniques, without prior knowledge about the range of values possible.

The main strength of eq. (1.6), however, lies in the fact that it can be *tuned* to a specific relationship between X and z by correctly choosing the parameters that generate the A matrices. Such tuning can be accomplished by comparing X and X*, as defined in section 1.2.

1.3.3 Mixed Model

At this point, we observe that the analytical approximation to P in eq. (1.1), eq. (1.5), has the essential physical information but is inaccurate. While the polynomial approximation, eq. (1.6) can be *made* accurate by *tuning*, but is a "black box" model without any physical interpretation attached to its parameters. In the following, we will try to combine these two models into one that has the advantages of both.

Assuming, in eq. (1.5), that D is a vector of *statistically independent physical parameters*, the variations of which constitute the disturbances in the process, we can attempt to tune eq. (1.5) in much the same way as eq. (1.6). If we also expand eq. (1.5) as in section 1.3.1, we can then rewrite it as:

$$x_i \ll F \left(\begin{matrix} \cdot \\ \cdot \\ \cdot \end{matrix} \right) \quad (1.8)$$

and

$$*j = F_p \langle z_1, z_2, D, A \rangle \quad (1.9)$$

Where F_ϵ is an analytical model of elements and inline parameters, and F_p is an analytical model of the process. $*$, as in eq. (1.2), is a set of physical parameters resulting from the process. By doing this, we now have a *simple* model, by virtue of the decomposition, which has the essential physical information about the process, and which can be *tuned* by correctly identifying the parameters that will generate the physical parameters used in that model, i.e. D .

The tuning can be accomplished in the same way as for eq. (1.6), that is, minimizing the difference between X generated by eq. (1.5) and an X^* measured in the process. We note here that the *statistical independency* of components of the random vector D makes this task a lot easier. The minimization is done over the parameters that control the RNGs generating D . This may be expressed in the following way:

$$\min_f | X - X^* | \quad (1.10)$$

Where

$$X = F_\epsilon(t^*, X)$$

$$* \ll F_p(z_1, z_2, D, A)$$

$$D = \text{RNG}(f) \quad (1.11)$$

f is the set of parameters controlling the generation of the random vector D . We note that since the A matrices *do not* have any physical interpretation, they are restricted to the process conditions under which the identification has been done. Since physical parameters, however, depend on process conditions in a known manner¹, then a model described by eqs. (1.8) and (1.9) may be easily adjusted to varying process conditions.

Next, we consider a possible implementation of such a model.

¹ .0. tte <topepd*nc« of diffusivity on temperature

1.3.4 Basic Model Structure

The basic model structure is shown in Figure 1*1.

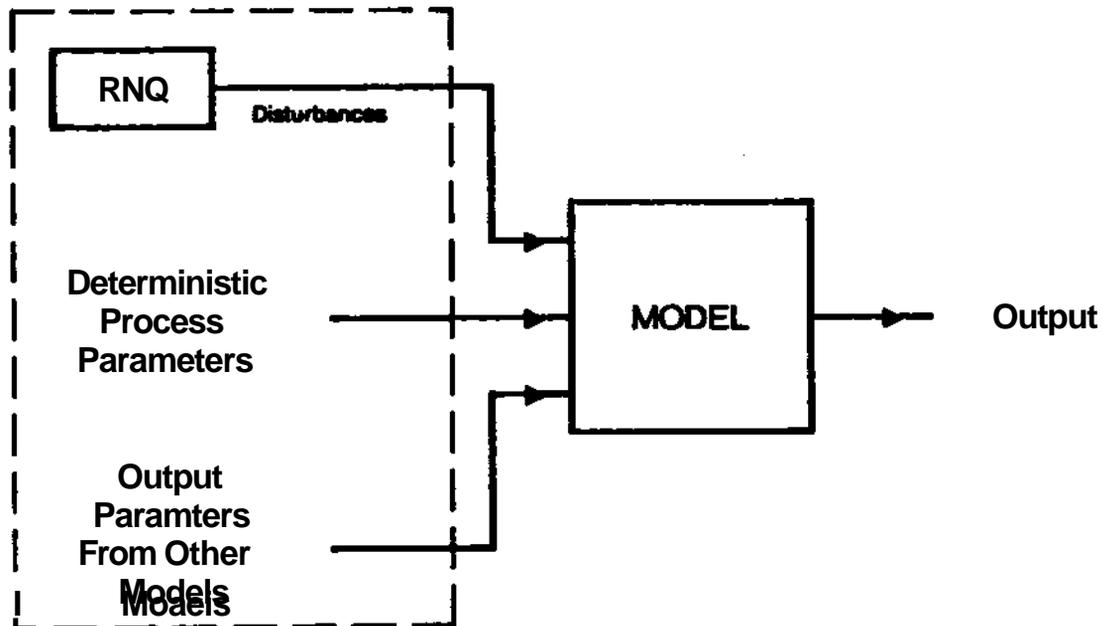


Figure 1-1: Basic model structure

The inputs to the model are of three main types:

- e *deterministic parameters* these represent quantities that are fixed by the process engineer, such as times for various processes, fluxes of gases through diffusion chambers etc... Constants, such as, for example, Boltzmann's constant, can be thought of as being deterministic parameters as well.
- e *random variables* that are outputs from previous process modeling steps, e.g. impurity concentration of the epitaxial layer for the base diffusion model. This set of variables is random because previous models had some random variables as input, and thus, their output will also be random.
- e *random variables* that are outputs from random number generators (RNGs), and that represent the random disturbances of the process.

This model structure is very powerful, it will allow us to have a simple model that can be solved very fast, but which will be accurate because it has been tuned to the particular process via the parameters that control the RNGs.

1.4 Disturbances

Real process disturbances come at different levels. Crystal defects, for example, are local, and so effect a small area of a chip. Whereas operations such as photolithography, which are performed one wafer at a time, tend to effect the chips on one wafer in much the same manner, but can cause differences between wafers. Still at a higher level an operation such as diffusion is performed on a number of wafers at the same time, thus, it would cause little variability from wafer to wafer, but causes significant differences between lots.

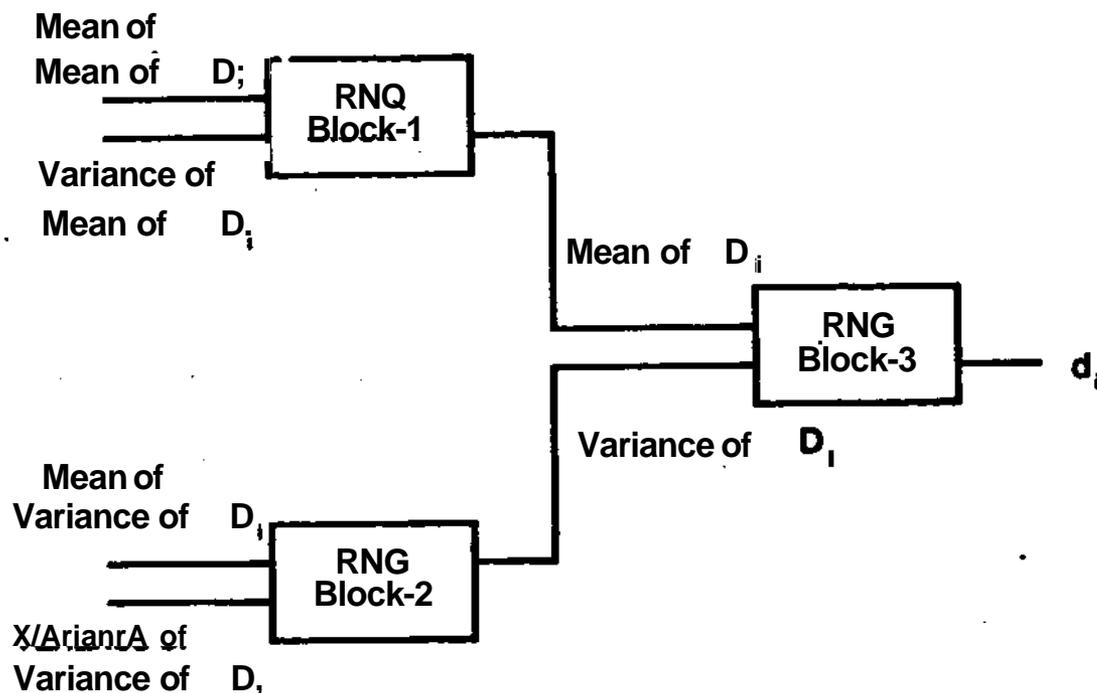


Figure 1-2: Multi-Level RNG

Since the disturbances have this hierarchy, then the RNGs generating them must be generated by some form of multi-level generator. An example of such a generation process is shown in Figure 1-2.

Each RNG block accepts a mean and variance of a variable, producing a value for that variable. So RNG block 1 accepts the *mean* and *variance* of the *mean* of D_i on the *wafer*, and produces a mean of D_i for a *chip*. Similarly, RNG block 2 accepts the *mean* and *variance* of the *variance* of D_i on the *wafer*.

producing a variance of D_i for the chip. And now the mean and variance of D_i on the chip can produce values of D_i for points on the chip.

So for a given simulation level (e.g. wafer with 100 chips), we can use an identifying process to estimate the means and variances, and use those values to generate the values of the physical parameters anywhere on a chip [Maly 81].

1.5 Applications of Simulation Methodology, FABRICS

FABRICS [Maly 79] [Strojwas 81] is a process simulator built around the simulation technology suggested above. This simulator incorporates device and process models for a typical bipolar process. The RNG control parameters were identified for a particular process, and thus the simulator was able to provide meaningful data for that process.

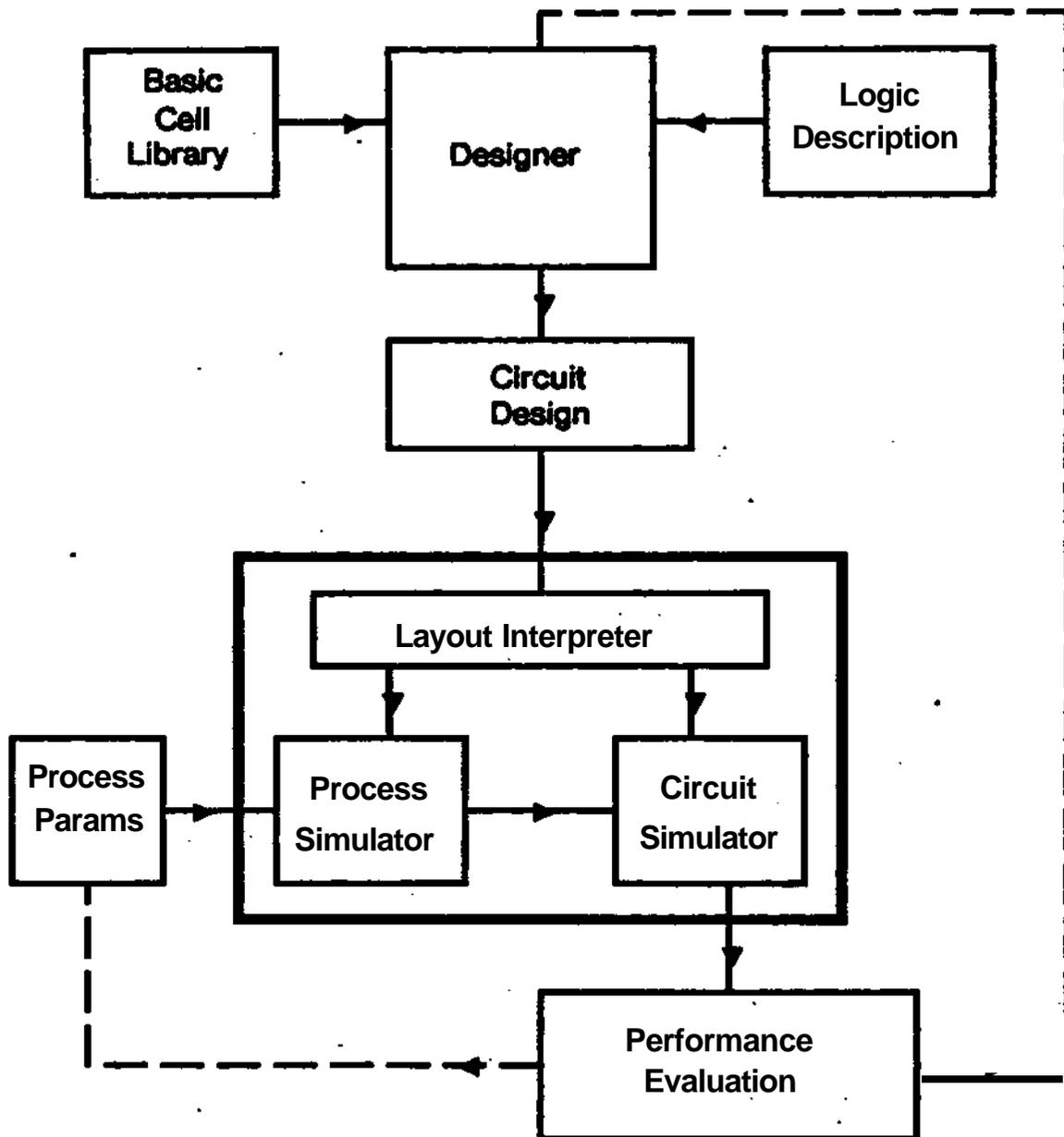
1.5.1 Structure of FABRICS

Figure 1-3 clearly shows how the basic model stipulated above is implemented. The interactive version of FABRICS [Strojwas 81] is shown in Figure 1-4. The process models are contained in the *PROCESS* block while the circuit and in-line element models are in the *CALPAR* block. The *MAIN* block generates the chip RNG parameters and controls the overall simulation.

The output for each chip simulated is stored in *ELPAR* and statistical routines are applied to it to produce meaningful displays such as histograms and scatter-plots and compute correlation coefficients.

1.5.2 Implementation

FABRICS is currently written in DEC-FORTRAN-20, and runs on CMU's TOPS-20. A more detailed description is given in Chapter 3, as well in the FABRICS MANUAL [Strojwas 81].



Figur« 1-1 Structure of FABRICS

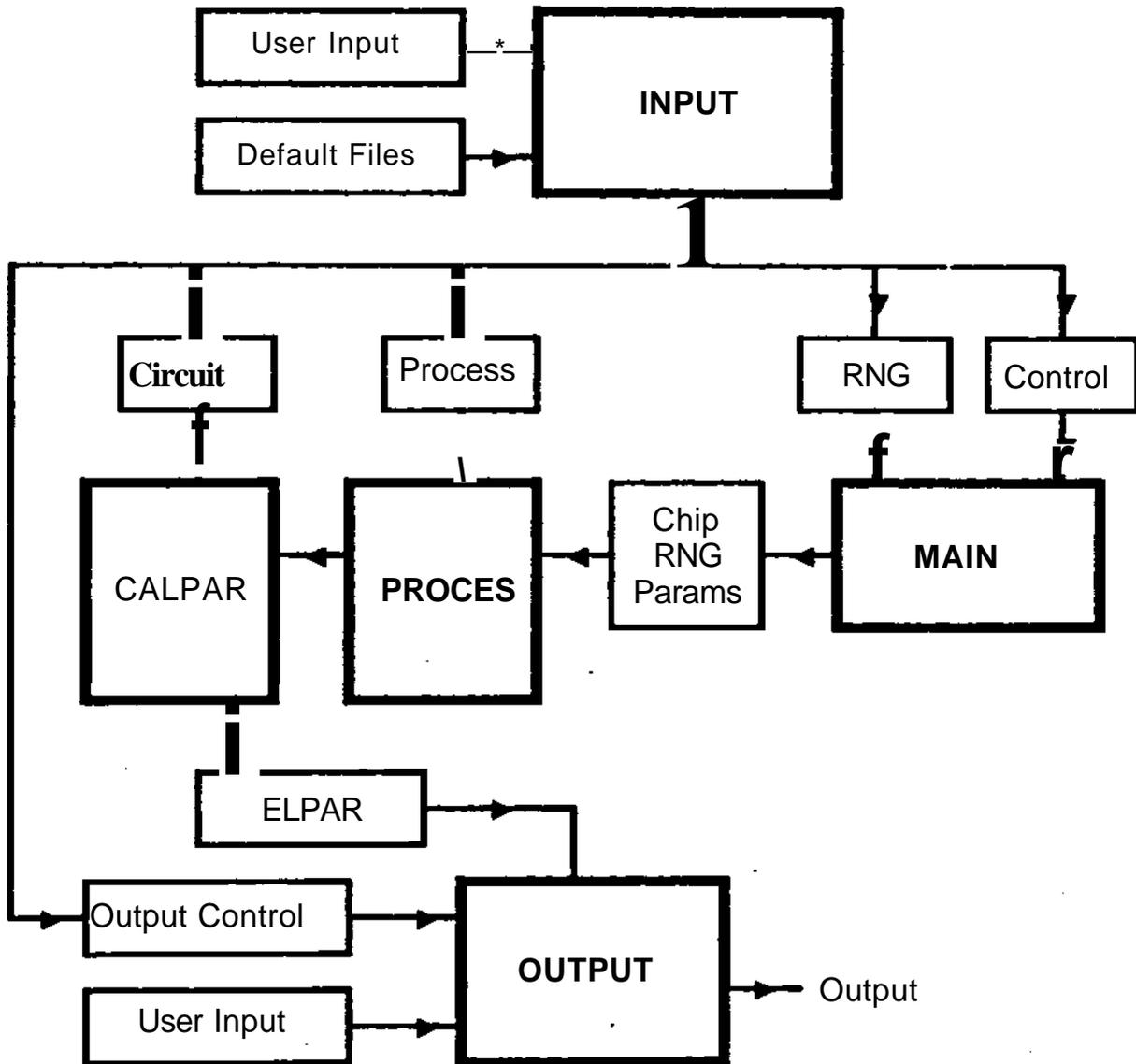


Figure 1-4: DATA flow diagram of FABRICS

CHAPTER 2

EXTENSIONS FOR MOS PROCESS SIMULATION

2.1 Introduction

This chapter begins with a brief review of some important MOS processing steps. The models used to characterize these steps are described, with some physical background for each. We describe extensions, to the resistor model in FABRICS, these extensions are also applicable to inter-connection resistances for MOS circuits.

2.2 Ion-Implantation Model

Ion-Implantation is an important part of the MOS process. It affords the process engineer precise control of the resulting impurity distribution, as well as control over the total amount of impurities present. In diffusion, each new layer must have a surface concentration significantly² higher than the one preceding it, so there is a limit on the number of layers, as well as on the the resulting values of sheet resistance. In Ion-Implantation, the designer has independent control of the *depth* of the impurities, and their *density*. Since the depth is controlled by the *energy* of the implant, whereas the density is controlled by the *dose* or current of implantation.

2.2.1 Background

When a substrate is bombarded by energetic ions, it will lose some of it's own ions by sputtering, and retain some of the incident ions. Those ions retained, are said to have been *implanted*. Interest in ion-implantation goes as early as 1954, in a patent by Shockley [Shockley 54]. Much work was done

²typically 2 or 3 orders of magnitude higher

on characterizing the resulting impurity profiles [Furukawa 74], and a unified theory for amorphous substrates was proposed by Linhard, Scharff and Schiott [Linhart 63], and generally referred to as *LSS* theory. *LSS* theory divides the forces stopping ions to two parts:

1. *ELECTRONIC*, due to interaction of the incident ion with the electrons in the host, S_e .
2. *NUCLEAR*, due to collisions between the incident ion and the nuclei of the host atoms, S_n .

and so the total energy loss per unit distance is expressed as:

$$\frac{dE}{dx} = N[S_n(E) + S_e(E)] \quad (2.1)$$

where N is the average number of host atoms per unit volume. Integrating to E_0 , the incident energy, gives total range R :

$$R = \frac{1}{N} \int_0^{E_0} \frac{dE}{S_n(E) + S_e(E)} \quad (2.2)$$

Different approximations to S_n and S_e lead to different solutions of the problem [Gibbons 68]. Computer solutions of the differential equations [Gibbons 75] [Brice 75] have been tabulated. Measurements of physical profiles, using such methods as Capacitance-Voltage characteristics, backscattering, junction staining and others [Hofker 75] [Fair 76] [Ziegler 72], agree well with theory, see Figure 2-1. But it is important to note that this theory was developed for *amorphous* materials, and that implantation along some crystal directions allows the incident ions to *channel* through the crystal to great depth [Pavlov 72]. Implantation is performed, usually, at a small angle with respect to the normal, typically 7° , to avoid this effect.

The implanted profiles are described by a range R_p , which is the *projected range*³, as well as σ_p , the variance of the distribution (see Figure 2-2). Higher order moments [Mylroi 72] can be used to characterize the distribution more fully.

³. i.e. range projected on the normal to the silicon surface

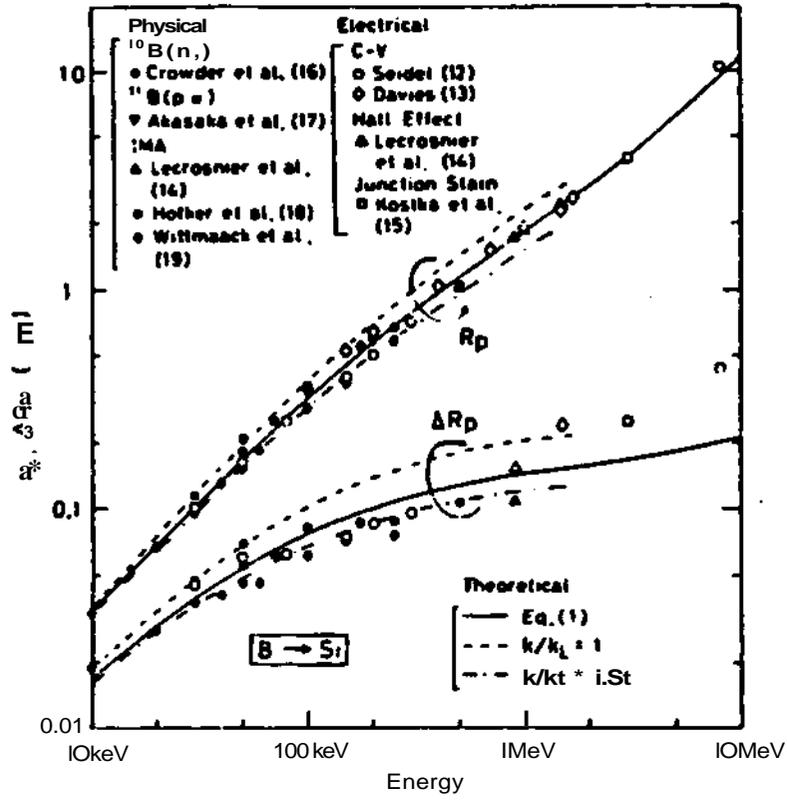


Figure 2-1: Comparison between theory and experiments (after [Furukawa 74])

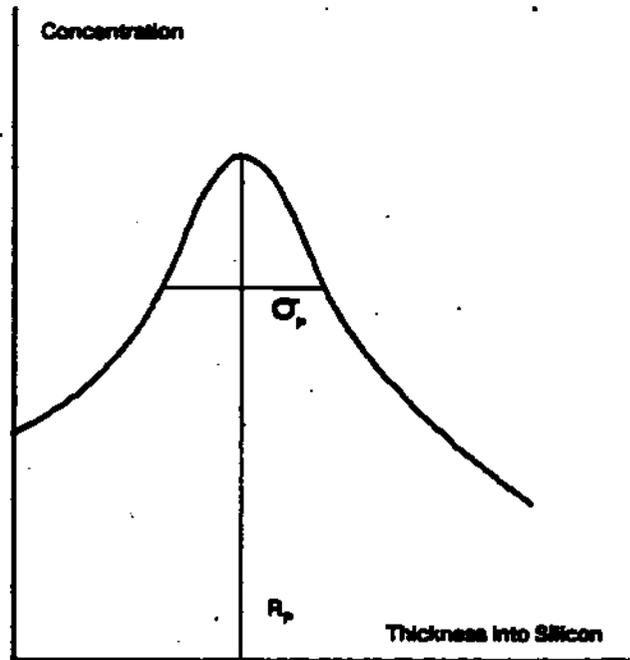


Figure 2-2: Definition of R_p and σ_p

2.2.2 Model Implemented

There are many ways of estimating the distribution of the implanted ions in a substrate. A simple gaussian distribution [Ziegler 72], a double gaussian distribution [Gibbons 73], or a Pearson distribution [Hofker 75]. We chose to use the simple gaussian distribution, mainly because work has been done on an analytical solution of the problem of rediffusion during subsequent high-temperature processes [Huang 70] [Lee 79] [Wu 76], see also section 2.2.2.3. We also limited our concern to one impurity. Boron, which is widely used in n-MOS technology.

2.2.2.1 Implantation

The profile after implantation was modeled as a gaussian distribution, with a mean R_p and a variance σ_p^2 . Data for Boron means and variances in Silicon, from [Gibbons 75], for the energy range 10-150 KeV was fitted to linear and parabolic equations of the form:

$$R_p \ll aE \cdot b \quad (2.3)$$

$$\sigma_p \ll cE^* \cdot d \quad (2.4)$$

Where E is the Energy in KeV, R_p is the mean and σ_p is the variance of penetration, both in //m. a , b , and c were calculated using a least-squares fit. The results are presented in Appendix A.2, along with a comparison to the data from [Gibbons 75]. d in eq. (2.4) was modeled as a random variable, to account for disturbances in the process due to the quality of the silicon.

The profile, thus, would be expressed as:

$$n(x) * \frac{f}{\sigma_p(2w)*} \exp \frac{- (x - R_p)^2}{2\sigma_p^2} \quad (2.5)$$

Where f is the dose in cm^{-2} .

2.2.2.2 Annealing

Annealing is a low-temperature process which follows implantation. It is performed to allow the crystal structure, which has been damaged by the implantation, to reform.

Since the diffusion length $(Dt)^{1/2}$, where D is the diffusivity and t is time, is small compared to the mean of penetration R_p [Wagner 72], then we can expect the profile to retain its initial shape, with some broadening to account for damage-enhanced diffusion. In other words, R_p will remain the same, the profile will still be gaussian, but a_p will increase.

We model this in the following way:

$$\sigma_p^2 = \sigma_p^2 + \Delta\sigma_p^2 \quad (2.6)$$

$$A \ll r_p \ll r(Dt)^{1/2} \quad (2.7)$$

$$D \ll D_0 e^{E/Arr} \quad (2.8)$$

Where:

- $E = 2.4$ eV, and is the activation energy.
- k is Boltzmann's constant.
- T is temperature of annealing in °K.

t is time of annealing in seconds.

γ is a random variable identified to fit equation (2.6) to the real physical situation.

2.2.2.3 Redistribution during subsequent high-temperature processes

The determination of impurity profiles, after other high temperature processes were performed on the wafer, is important to the process engineer. [Huang 70] suggested dividing the redistribution into two components:

1. redistribution due to diffusion without taking oxidation into account

2. redistribution due to the moving Si-SiO₂ boundary.

Starting from a gaussian distribution, and building on the theory of [Huang 70], [Wu 76] derived a closed form solution to the problem. The assumptions for the [Huang 70] model were:

- i. constant boron diffusivity within the Silicon
- ii. negligible boron diffusivity in the oxide
- iii. parabolic growth rate for the oxide
- iv. the initial profile is a delta function

The last two assumptions are relaxed in the [Wu 76] model, and later in the [Lee 79] model, which is the one implemented. The improvements of the [Lee 79] model over the [Wu 76] model are; a better approximation to the redistribution due to oxidation, and accounting for the fact that the oxide is not a reflective boundary, i.e. that impurities may diffuse from the silicon to the oxide.

Thus, the model is expressed in the following manner:

$$n_1(x,t) = \frac{n_{\max} \sigma_p}{[8(1+r)Dt]^{1/2}} [\Omega(x+mx_0,t) + \Omega(x-mx_0,t)] \quad (2.9)$$

where:

$$\Omega(y,t) = \exp \frac{(y-R_p)^2}{4(1+r)Dt} \left[1 + \operatorname{erf} \frac{ry+R_p}{\sigma_p(2+2r)^{1/2}} \right] \quad (2.10)$$

$$n_{\max} = \frac{\phi}{\sigma_p(2\pi)^{1/2}}$$

and,

$$r = \frac{\sigma_p}{2Dt} \quad (2.11)$$

x_0 is the thickness of the oxide, and m is the ratio between oxide thickness and the corresponding silicon consumed in forming it ($m = 0.44$).

$$n_1(x,t) \approx A \operatorname{erfc} \frac{x - mx_0}{\sqrt{2(Dt)^*}} \quad (2.12)$$

with:

$$A \approx \frac{f \cdot D \cdot n_0 \cdot \{Q(mx_{A,t}) - CX - mx_{A,t}\} - n_0 (O \cdot t U^{\wedge} \cdot e)}{2 \cdot \left[\frac{K_p}{2 \cdot r p (Dt) h} + \exp(\mu^2 (D/\pi t)^{\frac{1}{2}}) \right]} \quad (2.13)$$

and.

$$t = \frac{r}{(1+r)^{\frac{1}{2}}} \frac{\%}{2 \cdot r p (Dt) h}$$

$$\alpha = \frac{mx_0}{2(1+rt)}$$

$$\mu = \frac{mx_0}{2(Dt)^{\frac{1}{2}}}$$

$$\beta = \frac{1-mK}{2K} \frac{K_p^{\frac{1}{2}}}{(t \cdot K_p/4K_L^2)^{\frac{1}{2}}}$$

Where:

e K_p^4 is the coefficient of parabolic oxide growth in //m.sec².

e K_L is the coefficient of linear oxide growth in /i.sec¹.

• K is the segregation coefficient.

The erf function was approximated as [Cave 65]:

$$\operatorname{erf}(x) \approx 1 - \frac{1}{x} + \frac{1}{2x^3} - \frac{1}{4x^5} + \dots$$

and,

$$\operatorname{erf}(x) = 1 - \operatorname{erfc}(x)$$

2.3 Extending the Photolithography Model

Photolithography effects determine the dimensions of devices and interconnections in the silicon. These dimensions, along with the impurity profiles, determine the electrical characteristics of the device. Below we describe models that account for the differences between the *mask* dimensions and the *actual physical dimensions* in the silicon. Although these models are described with references to diffused resistors, they can equally well be used for all dimensions (e.g. channel width for MOS devices).

The resistance of a resistor, like the one shown in Figure 2-3, can be calculated from an expression of the following form:

$$R = \frac{R_s \cdot L}{W} \quad (2.14)$$

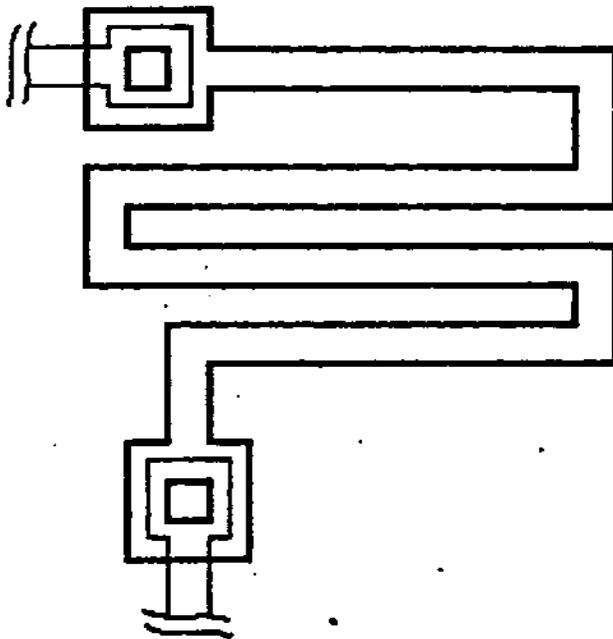


Figure 2-3: A typical resistor

Where: v

R_s is the sheet resistance of the layer (in Ω/\square)

- L_{eq} is the equivalent length in μm .
- W_{eq} is the equivalent width in μm .
- R_C is the resistance per contact in Ω

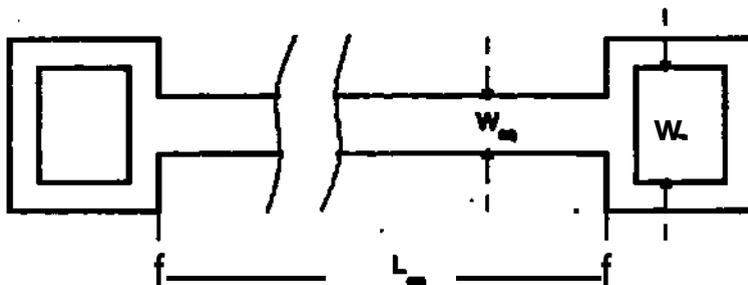


Figure 2-4: Equivalent Resistor

Such a resistor may be modeled by a "straight" resistor, such as the one shown in Figure 2-4. The sheet resistance R_s , being an inline measurement, is calculated from the physical profile using the appropriate inline model. The inline models, as well as other device models we describe in the FABRICS MANUAL [Strojwas 81]. The other parameters WB calculated as follows.

2.3.1 A Model for Equivalent Dimensions

Three factors are taken into account:

1. *Photolithography*, accounting for the difference between mask dimensions and actual dimensions in the silicon.
2. *Geometry*, replacing "corners" by their equivalent lengths.
3. *Sidewall Conductances*, accounting for the lateral diffusion of impurities.

2.3.1.1 Photolithography

The effect of Photolithography is modeled in *FABRICS* as a random disturbance in both the X and Y directions. See Figure 2-5.

Thus a dimension L in the mask will become L^* , in silicon, with:

$$L^* = L + \Delta F \quad (2.15)$$

Where ΔF is a random variable simulating photolithographic disturbances.

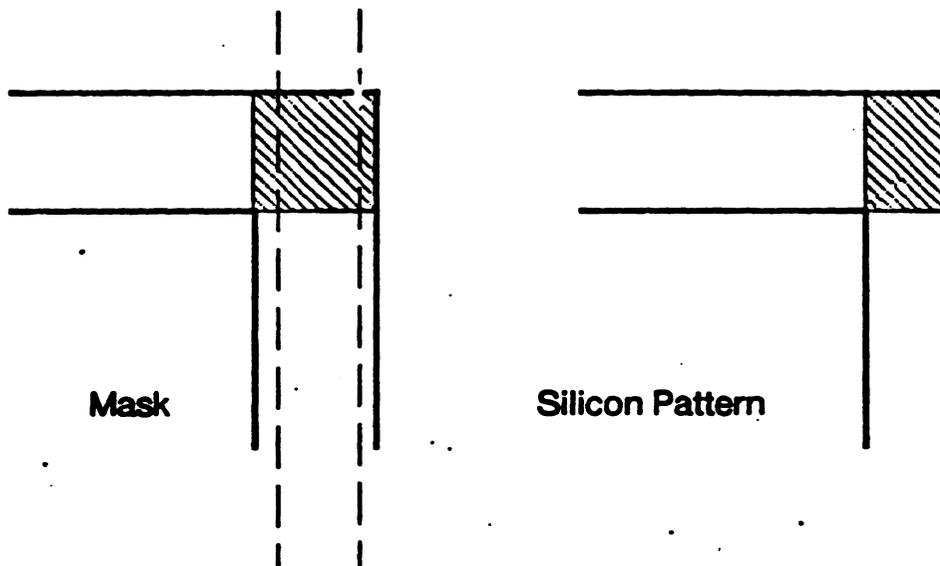


Figure 2-5: Illustrating the effect of photolithography disturbances

2.3.1.2 Corners

Transforming corners into straight length is based on a scheme described in [Hall 67], and is shown in Figures 2-6.

The formula presented by [Hall 67] for a corner such as in Figure 2-6 is:

$$N_{eq} = a^{-1} - \frac{2}{\pi} \ln \frac{4a}{a^2 + 1} + \frac{a^2 - 1}{\pi a} \cos^{-1} \frac{a^2 - 1}{a^2 + 1} \quad (2.16)$$

Where N_{eq} is the number of squares equivalent to a corner (e.g. shaded

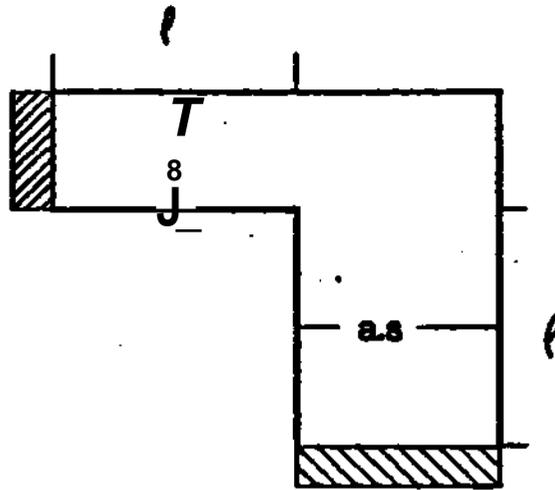


Figure 2-6: Equivalent length of corners. ([Hall 67]).

portions in Figure 2-5). Since a , the ratio between the two widths connecting at the corner, will always be near one, eq. (2.16) can be approximated by the quadratic:

$$N = 0.13837 a^2 - 0.27674 a + 0.6971 \quad (2.17)$$

The agreement between the two formulas is good for a $\pm 1.5\%$ (see table A.1).

2.3.1.3 Sidewalls of Diffused Resistors

Sidewall conductances (Figure 2-7) can be thought of as two resistors in parallel with the original resistor, decreasing the total resistance of the resistor. This effect can be modeled [Kuzmicz 78] by an increase in the width of the resistor. Thus, for base resistors:

$$AW = 2R_{SB}/k \quad (2.18)$$

Where AW is measured in μm and R_{SB} is the sheet resistance of the base layer.

$$k = 1.83 \times 10^{12} \frac{\ln(N_{SB}/N_{EP1})}{X_{JC}^2 N_{SB}^{0.754}} \quad (2.19)$$

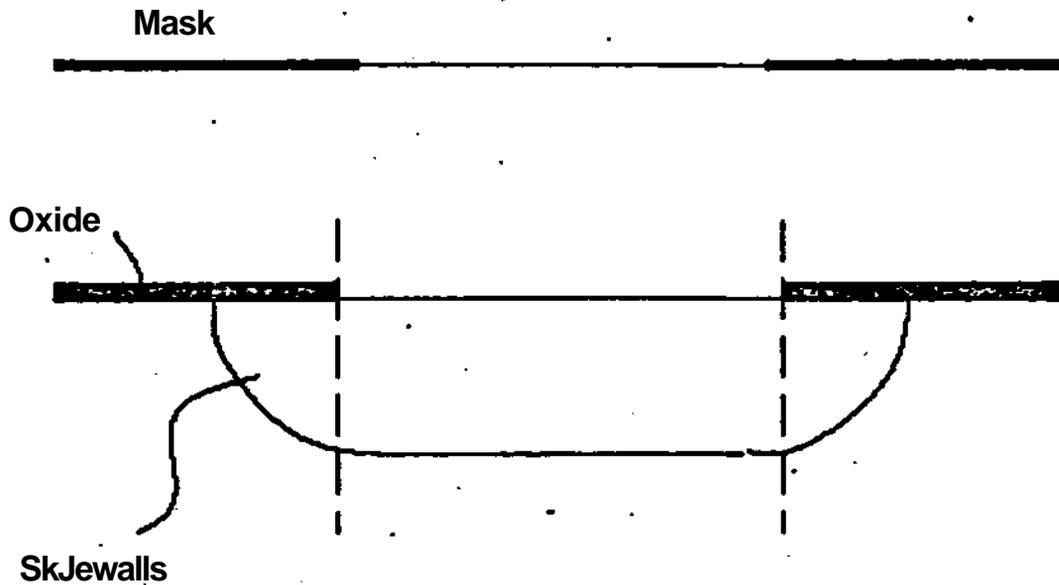


Figure 2-7: Sidewalls of Diffused Resistors

in which:

- # $N_{M_{SB}}$ the surface concentration of the base layer in cm^{-2} .
- e N_{cPI} the concentration in the EPI layer in cm^{-3} .
- X_{JC} the depth of the base-epi junction in μm .

k will be in $\mu\text{m}\cdot\Omega^{-1}$.

2.12 A Model for Contact Resistance to Diffused Impurity Layers

The contact resistance to diffused impurity layers is calculated using the following formula [DAndrea 70]:

$$R_{C'} = (R_S \rho_C)^{1/2} / W_C \quad (2.20)$$

Where:

- R_c is the resistance per contact in (Ω)
- R_s is the sheet resistance of the diffused layers underneath the contact in Ω/\square .
- p_c is the *specific contact resistance*, and is modeled by fitting a linear function to data extracted from [Fang 79], with the intercept treated as a random variable to account for processing disturbances.
- W_c is the *width* of the contact in μm .

2.4 Model for Oxidation

Oxidation is important in semiconductor processes in general, for its use as a masking medium for diffusion, and in MOS technology in particular for its use as an insulator between the gate and the channel. Therefore, it was important to have a model capable of providing accurate data for all types of oxidation conditions (e.g. wet, with water vapor, and dry, with oxygen).

The thermal oxidation of silicon is well characterized by an equation of the form: [DEAL 65]

$$X_0^2 + AX_0 = B(t + \tau) \quad (2.21)$$

where:

- X_0 is the oxide thickness
- τ is a constant related to the initial oxide thickness
- t is time

The constants A and B depend on temperature and can be expressed as:

$$B = C_1 e^{-E_1/kT} \quad (2.22)$$

$$B/A = C_2 e^{-E_2/kT} \quad (2.23)$$

Where:

- E_1 and E_2 are activation energies for parabolic and linear oxidation rates, respectively.
- C_1 and C_2 are constants depending on the orientation of the crystal. T is temperature in °K.
- k is Boltzmann's constant.

The values for C_1 , C_2 , E_1 and E_2 for dry and wet oxidation of <111> and <100> silicon, these orientations are the most commonly used for MOS and Bipolar technology, respectively, are given in table A.3, as reported by [Deal 78]. Equation (2.21) is solved for increments of oxide thickness in each process as follows:

$$AX_Q \ll \% \{ (A - X_Q) \cdot [(A \cdot X_0)^2 \cdot ABt]^* \} \quad (2.24)$$

CHAPTER 3

INTERFACES TO FABRICS

3.1 Introduction

In this chapter, we describe the interfaces between *FABRICS* and circuit analysis programs. An interface to the circuit analysis program *SPICE* is described. We also discuss the interactive user front end.

3.2 The Design Environment

Any CAD tool is only as useful as it is easy to use. A program such as *FABRICS* must be able to communicate with the designer in a manner that will aid, rather than hinder, productivity. Hence the need for interfaces between *FABRICS* and other programs, as well as an interface to the user.

A possible design environment is shown in Figure 3-1. The package of programs might include a process simulator, a circuit simulator, and appropriate interfaces. The layout of an IC is analyzed and split into two data segments:

1st- Element Layout, i.e. dimensions of individual elements

2nd- Interconnections

The first portion is used in the process simulator to simulate the elements, while the second is used in the circuit simulator.

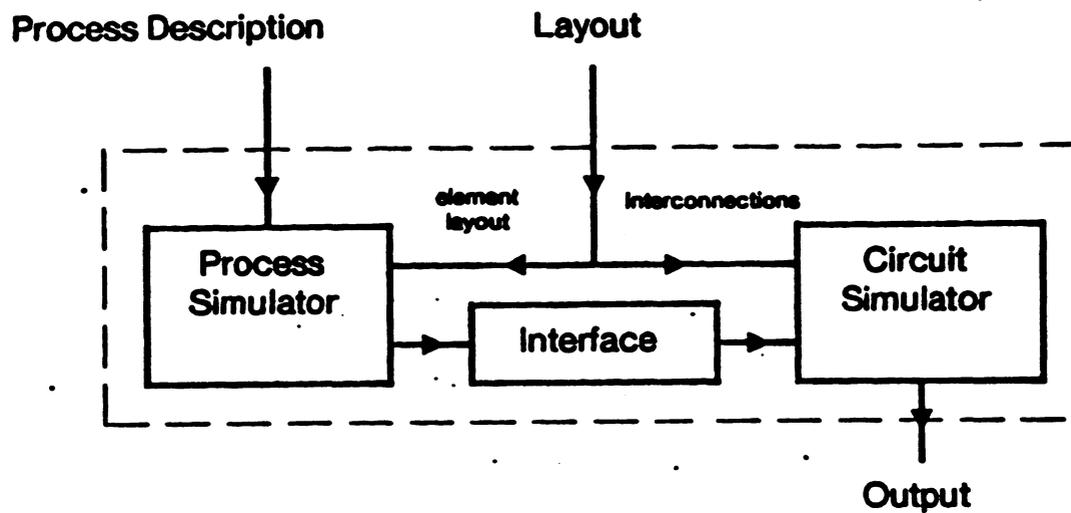


Figure 3-1: A possible design environment

3.3 Interface to Circuit Simulators

Since no package to extract element layout and interconnection information from the IC layout description was available, the interface between FABRICS and circuit simulators was built in such a way as to allow the user to interactively enter the node numbers for the elements simulated. A file is then generated which is compatible with the circuit simulator SPICE.

3.3.1 Possible modes of simulation

Simulation of a circuit, where elements are to be calculated using a statistical process simulator, can be done in two ways:

1. on a chip-by-chip basis, i.e. taking the output for every chip and performing the simulation with those values. Since many simulators take advantage of the fact that circuit layout (and consequently interconnections) is unchanged, and that only the values of the elements change, the cost of such a simulation technique may be reduced significantly.

2. Simulation using the mean values, and statistics on the variations of the elements, as well as the correlation coefficients. This method may not be as accurate as the first, but would cost less for large samples.

The structure of possible interfaces for the two approaches above is shown in Figures 3-2 and 3*3.

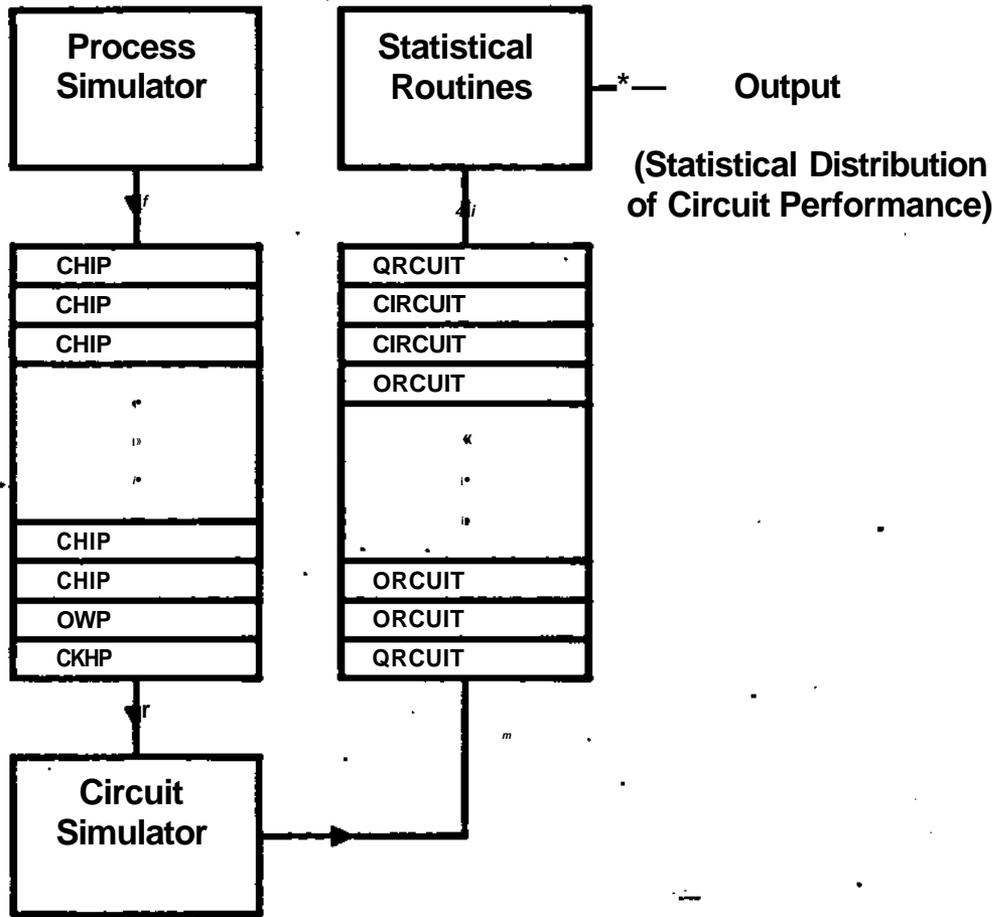


Figure 3-2: Interface for a chip-by-chip mode

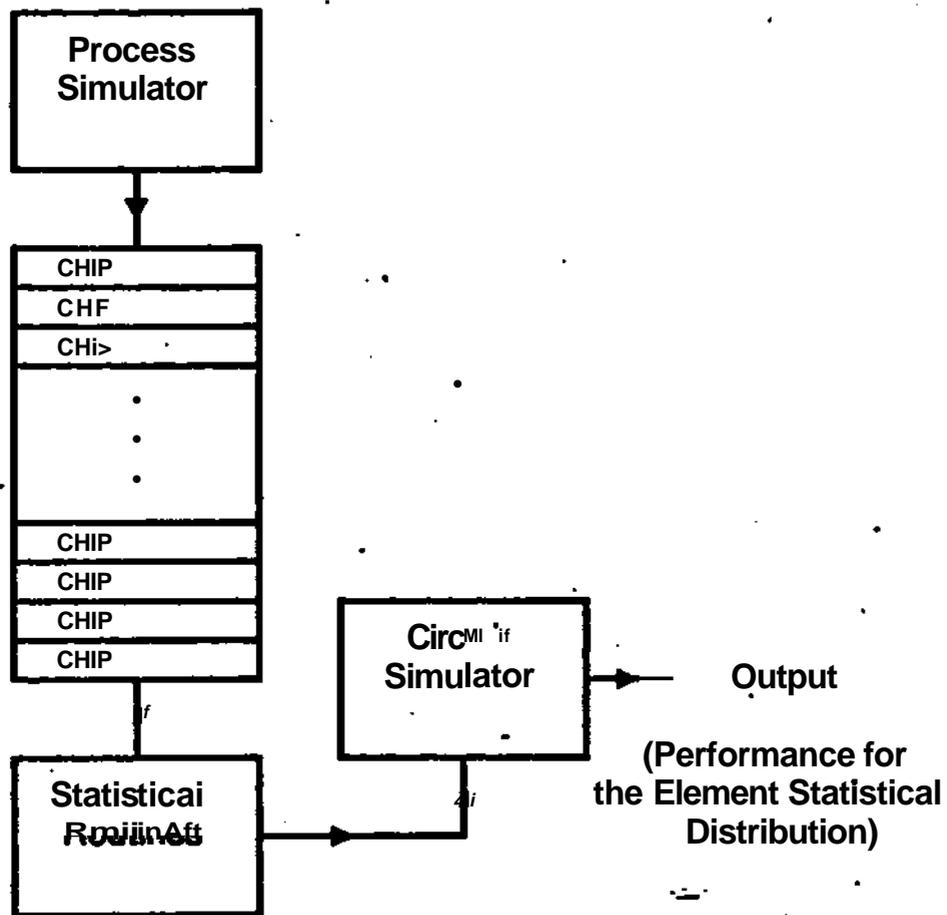


Figure 3-3: Interface for a mean-and-statistics mode

3.3.2 Interface implemented

As mentioned above, a package to extract features from a layout description and deliver element dimensions was not available, so the interface between FABRICS and SPICE was built to allow FABRICS to prompt for the nodes of elements simulated. And a SPICE compatible file is then generated.

To allow for future expansion to the second method of simulation, i.e. the chip-by-chip method, provision was made to write the chip element

parameters, chip by chip, into a file; which will eventually be used by a circuit simulator with the required capabilities.

3.4 User Interface

The user interface implemented in FABRICS is built around a *hidden menu* structure. By hidden menu we mean that the menu is not on display all the time, so the experienced user, who knows what data is needed at any time, need not be bothered. But the novice can have the menu displayed at any time by hitting a help character.

Extensive help, error detection, and defaulting features are implemented, as well as rudimentary edit commands for sections that require them. The interface is written in DEC-FORTRAN, and runs on a DEC-20 machine. A PASCAL version is under development.

A more detailed view of the interactive user interface is presented in the FABRICS MANUAL, [Strojwas 81], which has examples of the use of the interface.

3.4.1 INPUT Interface

In Figure 1-3, the input interface accepts user input, as well as default files, to produce five data segments:

1. *CONTROL* which defines the type of simulation being performed, i.e. simulation at the *chip*, *wafer* or *lot* level. With information about how many simulations will be performed, number of chips per wafer, number of elements per chip etc—
2. *RNG*, this data segment contains the parameters that control the RNGs that will generate the data needed by the RNGs for the chip. They depend on the level of simulation. For a wafer level simulation, the RNG segment would contain⁵:
 - mean of mean on wafer
 - variance of mean on wafer

⁵ see Figure 1-2

- mean of variance on wafer
 - variance of variance on wafer
3. **CIRCUIT**, defines the dimensions of all the elements that will be simulated, as well as information about which element parameters to calculate, which in-line parameters to calculate, as well as information about which devices will share the same diffusion profiles.
 4. **PROCESS**, has the process-engineer-defined parameters of the process, such as times and temperatures, as well as oxidation data.
 5. **OUTPUT CONTROL** is the data segment passed to the output, it contains names of parameters, as well as the order in which they are calculated.

3.4.2 OUTPUT Interface

The output interface allows the user to display and store several basic forms of output:

- a. Statistical Information, including means, variances, and extrema, as well as correlation coefficients.
- b. Histograms, which can be displayed on any TTY.
- c. Plot Files, which are:
 - Scatter plot files, which can be displayed on an HP-2647 Graphics terminal or entered into a general scatter plot program.
 - Beta plot files, which plot transistor β vs. current and voltage on an HP-2647 Graphics terminal.
- d. Store all elements of all chips, with information to help retrieve the elements in an orderly fashion.
- e. Store all correlation coefficients, to aid in factor analysis.

Output may be directed to disk alone, or displayed on the screen as well. The program that controls the graphics terminal is not yet an integral part of FABRICS, but will be when the PASCAL version is completed.

CHAPTER 4

APPLICATIONS, LIMITATIONS AND FUTURE EXPANSION

4.1 Introduction

We now discuss the application areas of a statistical simulator, and describe some future expansion possibilities, as well as possible limitations of the simulator.

4.2 Applications

The statistical process simulator described so far, FABRICS, may be used for several design tasks. The capabilities of a process simulator are not limited to its use as such, but when communicating with other programs, may be used as a useful design tool. Some applications are outlined below:

- As in section 3.3, a process simulator interfaced with a circuit simulator may be used as an IC design tool when the design of an IC is to be optimized. The optimization may be over device geometries or process conditions, and may be done interactively, with the designer viewing the results and deciding on any necessary compromise, or automatically, with predefined circuit performance and yield.
- A process simulator may be used as a diagnostic tool for process disturbances. In this mode, faulty lots are analyzed and the data is used to simulate possible reasons for the drop in yield. It may also be used to reject faulty wafers early in the production run, thus minimizing process cost.
- By itself, a process simulator may be used for process design, i.e. setting the process parameters, such as times and temperatures. It may also be used for optimizing the process for a specific applications, if the required process conditions are defined.

4.3 Limitations

The basic limitation on the accuracy of FABRICS is imposed by the basic model structure described in section 1.3.4. The identification of the RNG parameters simulating disturbances in the process is done for some set of process conditions. These conditions may be thought of as being a quiescent point of operation for the process. Any radical change in these conditions will lead to results that deviate from comparable physical measurements.

The analytical physical model does account for some of the changes that will occur due to variations in process conditions. But, as mentioned in section 1.3.2, since this analytical model can, at best, be accurate over some *limited* range of process conditions, then it cannot possibly account for radical process parameter changes.

One approach for avoiding such difficulties is to solve the model differential equations for the process conditions, and then use the results to identify a new set of RNG control parameters. This will be cheaper than performing experiments and measurements on the process with the new conditions.

4.4 Further Extensions **and** Possibilities

Now that the processes used in MOS technology have been added to FABRICS, we need MOS device models, as well as data from MOS processes to generate a set of RNG control parameters. With such a capability, the applications described in section 4.2 may be applied to the MOS technology as well.

As suggested in section 3.2, a package that can extract features from some IC graphic design language (e.g. *CIF* [Carver 80]), may be used in a design environment to partition IC layout information into device layout and interconnections. Hence, a process simulator communicating with a circuit simulator may form a part of a complete design environment that uses standard MOS cells, some form of *logic* description, and some design package. The design package should be able to generate an IC layout, given the logic description of the circuit, by suitably interconnecting the basic cells. This may be a design engineer, or some CAD tool. A possible scheme is shown in Figure 4-1.

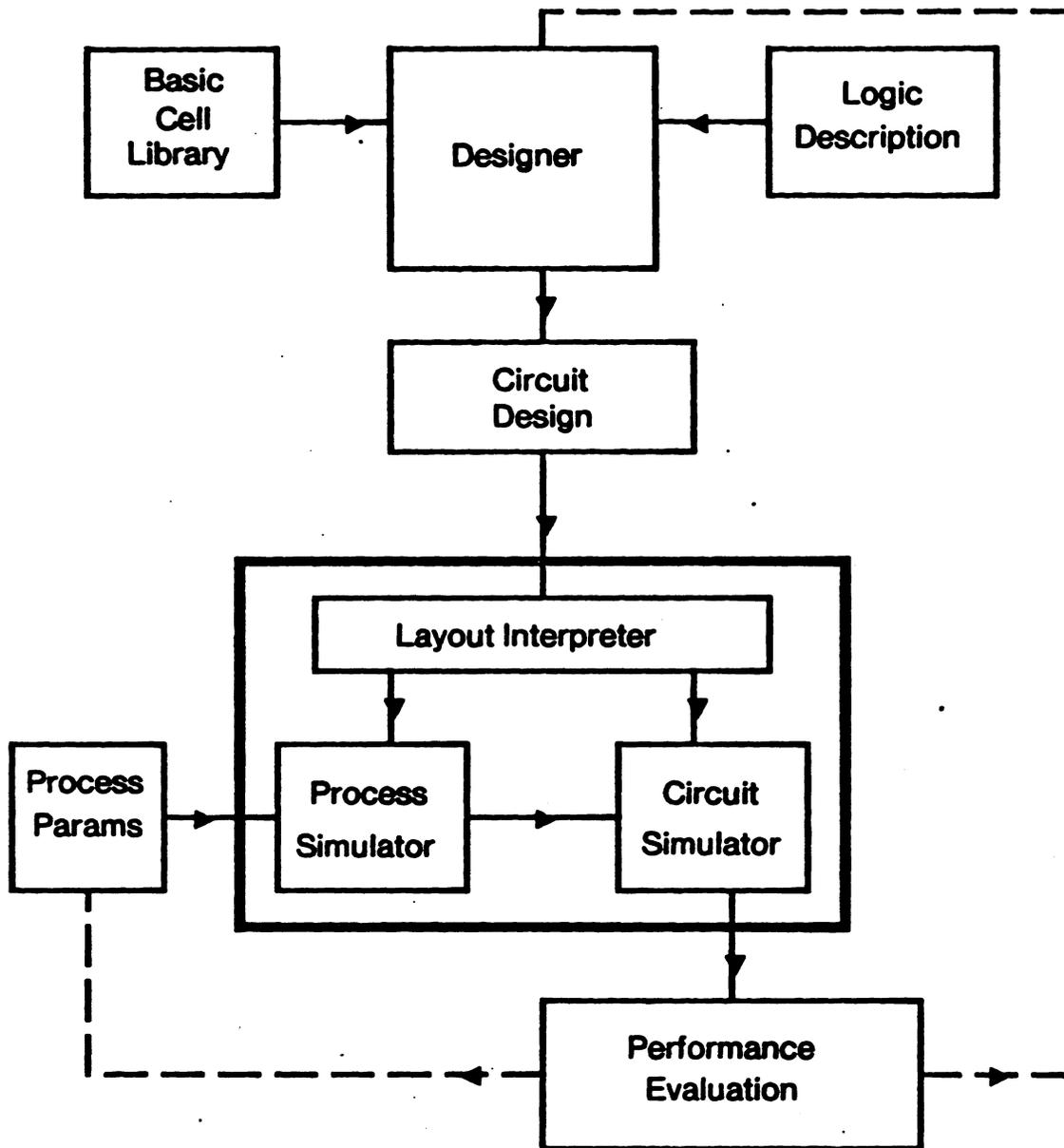


Figure 4-1: Possible scheme for a complete MOS design environment

APPENDIX A TABLES

A.1 Approximation to Corners

Below is a comparison between values of corner resistance calculated from eq. (2.16) [Hall 67], and eq. (2.17), an approximation to eq. (2.16). See section 2.3.1.2. The two equations are repeated below.

$$N_c \approx a^{-1} \frac{2}{\pi} \ln \frac{4a}{a^2 + 1} + \frac{a^2 - 1}{\pi a} \cos^{-1} \frac{a^2 - 1}{a^2 + 1}$$

$$N_c \approx 0.13837 a^2 - 0.27674 a + 0.6971$$

a	eq. (2.16)	eq. (2.17)
1.000	0.5587	0.5587
1.050	0.5591	0.5592
1.100	0.5601	0.5604
1.150	0.5618	0.5623
1.200	0.5643	0.5648
1.250	0.5674	0.5677
1.300	0.5712	0.5712
1.350	0.5757	0.5750
1.400	0.5809	0.5792
1.450	0.5867	0.5836
1.500	0.5933	0.5883

A.2 Approximation to Penetration data

The ion implantation model is described by two equations (see section 2.2.2.1h)

$$R_p \ll aE \cdot b$$

$$e_p \ll cE^* \cdot d$$

Where:

- $a \ll 2.7553 \times 10^{13}$
- $b \ll 1.8474 \times 10^{-2}$
- $c \ll 7.2655 \times 10^{13}$
- d is a random variable with means and variances defined to *fit* the model to the real situation.

Following is a comparison between data from [Gibbons 75] and the data generated using the above equations and constants.

Energy (KeV)	R_p (/im)	σ_p (/im)	V from eq. (2.3) {(/im)}
10	0.0333	0.0171	0.0460
20	0.0662	0.0283	0.0736
30	0.0987	0.0371	0.1011
40	0.1302	0.0443	0.1287
50	0.1608	0.0504	0.1562
60	0.1903	0.0556	0.1838
70	0.2188	0.0601	0.2113
80	0.2465	0.0641	0.2389
90	0.2733	0.0677	0.2665
100	0.2994	0.0710	0.2940
110	0.3248	0.0739	0.3216
120	0.3496	0.0766	0.3491
130	0.3737	0.0790	0.3767
140	0.3974	0.0813	0.4042
150	0.4205	0.0834	0.4318

A.3 Constants of Oxidation Model

As reported in [Deal 78]:

Oxidation Ambient

	DRY		WET	
Silicon Orientation	<100>	<111>	<100>	<111>
C_1	0.1215	0.215	1.07	1.07
C_2	1.03×10^2	1.73×10^2	0.270×10^5	0.453×10^5
E_1	0.78	1.23×10^1	0.78	1.23×10^1
E_2	0.205×10^1	0.2×10^1	0.205×10^1	0.2×10^1

References

- [Antoniadis 78] D. A. Antoniadis, S. E. Hansen and R. W. Dutton.
SUPREM II - A program for IC Process Modeling and Simulation
Integrated Circuit Laboratory, Stanford Electronics Laboratories,
Stanford University, Stanford, CA 94305., 1978.
Technical Report No. 5019-2.
- [Brice 75] D. K. Brice.
Ion Implantation Range and Energy Deposition Distributions.
IFI/PLENUM, 1975.
- [Carver 80] C. Mead and L. Conway.
Introduction to VLSI Systems.
Addison Wesley, 1980.
- [Cave 65] K. J. S. Cave.
The Base Diffusion Profile Arising from Boron Redistribution
in the Oxide - A useful approximation.
Solid State Elec. 8(12):991-993, 1965.
- [Chou 71] S. Chou.
An Investigation of Lateral Transistors - D.C. Characteristics.
Solid State Elec. 14(9):811-826, 1971.
- [DAndrea 70] G.DAndrea and H.Murrman.
Correction Terms for Contacts to Diffused Resistors.
IEEE Trans. Elect. Dev. :484-485, June, 1970.
- [DEAL 65] B. E. DEAL AND A. S. GROVE.
General Relationships for the Thermal Oxidation of Silicon.
J. APPL. PHYS. 36(12):3770-3778, DECEMBER, 1965.
- [Deal 78] B.E.Deal.
Thermal Oxidation Kinetics of Silicon in Pyrogenic H₂O and
5% HCl/H₂O Mixtures.
J. Electrochem. Soc. 125(4):576-579, April, 1978.
- [Fair 76] R.B.Fair and C.C.Tsai.
Profile Parameters of Implanted-Diffused Arsenic Layers in
Silicon.
J. Electrochem. Soc. 123(4):583-585, April, 1976.
- [Fang 79] Y. K. Fang, C. Y. Chang and Y. K. Su.
Contact Resistance in Metal-Semiconductor Systems.
Solid-St. Electron. 22:933, 1979.
- [Furukawa 74] S. Furukawa and H. Ishiwara.
How Well Can Experimental Results be Explained by Theories?
In S. Namba (editor), *Ion-Implantation in Semiconductors*, pages
143-154. PLENUM press, August, 1974.

- [Gibbons 68] J.F.Gibbons.
Ion-Implantation in Semiconductors-Part I. Range Distribution
Theory and Experiments.
Proc. IEEE 56(3h295-319. March, 1968.
- [Gibbons 73] J.F.Gibbons and S.Mylroie.
Estimation of Impurity Profiles in Ion-Implanted Amorphous
Targets Using Half-Gaussian Distributions.
Appl. Phys. Lett. 22(11h568-569. June. 1973.
- [Gibbons 75] J. F. Gibbons, W. S. Johnson and S. W. Mylroie.
Projected Range Statistics.
Dowden, Hutchinson & Ross, Inc.. Stroudsburg, Pennsylvania,
1975.
- [Hall 67] P. M. Hall.
Resistance Calculations for Thin Film Patterns.
Thin Solid Films :277-295, 1967.
- [Hofker 75] W.I.Hofker. D.P.Oosthoek, N.J.Koeman and H.A.M.DeGrefte.
Concentration Profiles of Boron Implantations in Amorphous
and Polycrystalline Silicon.
Radiation Effects 24:223-231, 1975.
- [Huang 70] J.S.T.Huang and LC.Welliver.
On the Redistribution of Boron in the Diffused Layer During
Thermal Oxidation.
J. Electrochem. Soc. 117(12):1577-1580. December, 1970.
- [Kuzmicz 78] Kuzmicz.
Private Communication to Andrzej Strojwas.
- [Lee 79] KLee, R.Dutton and D.Antoniadis.
On Redistribution of Boron During Thermal Oxidation of
Silicon.
J. Electrochem. Soc. 126(11):2001-2007, November, 1979.
- [Linhard.63] J. Linhard, M. Scharff, and H. Schiott.
Range Concepts and Heavy Ion Ranges.
Mat. Fys. Medd. Dan. Vid. Sclsk. 33:1-39, 1963..
- [Maly 79] W.Maly and A.Strojwas.
Simulation of Bipolar IC Elements for Statistical Circuit
Design.
In *Proceedings of 1979 ISC AS*. pages 788. ISCAS, 1979.
- [Maly 81] W. Maly and A. J. Strojwas.
Statistical Simulation of IC Manufacturing Process.
1981.

- [Mylroi 72] S. Mylroi and J. F. Gibbons.
Computation of Third Central Moments for Projected Range Distributions for Common Ion-Implanted Dopants in Silicon. In B. L. Crowder (editor). *Ion-Implantation in Semiconductors and other Materials*, pages 243-253. PLENUM press, December, 1972.
- [Nage! 75] L. W. Nagel.
SPICE, A computer program to simulate semiconductor circuits. PhD thesis. University of California, Berkeley, 1975.
- [Pavlov 72] P. V. Pavlov.
The influence of Channeling and Anomalous Diffusion on the Profiles of implanted Phosphorus and Boron in Silicon. *Radiat. Eff.* 13:153-155, April, 1972.
- [Shockley 54] W. Shockley.
Forming Semiconductor Devices by Ion c Bombardment. U.S. Patent 2 787 564.
- [Strojwas 81] A. J. Strojwas and S. R. Nassif.
FABRICS Manual
Design Research Center, Carnegie-Mellon Univ. Pittsburgh, PA 15213, 1981.
Research Report No. DRC-18-30-81.
- [Wagner 72] S.Wagner.
Diffusion of Boron from Shallow Ion Implants in Silicon. *J. Electrochem. Soc.* 119d1h1570-1576, November, 1972.
- [Wu 76] C.P.Wu, E.C.Douglas and C.W.Mueller.
Redistribution of Ion-Implanted Impurities in Silicon During Diffusion in Oxidizing Ambients. *IEEE Trans. Elect. Dev.* :1095-1097, September, 1976.
- [Ziegler 72] J.F.Ziegler, B.LXrowder, CW.Cole, J.E.E. Baglin, and B.J.Masters.
Boron Atom Distributions in Ion-Implanted Silicon by the $(i(n),^4\text{He})$ Nuclear Reaction. *Appl. Phys. Lett.* 2i(1h16-i7, July, 1972.