Copyright © 1984, 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.

AN IC PROCESS-ORIENTED CIRCUIT SIMULATOR DEVELOPED FROM SPICE2G.6 USING THE CSIM MOSFET MODEL

by

Ralph C. L. Liu

Memorandum No. UCB/ERL M84/9

15 January 1984

AN IC PROCESS-ORIENTED CIRCUIT SIMULATOR DEVELOPED FROM SPICE2G.6 USING THE CSIM MOSFET MODEL

by

Ralph C. L. Liu

Memorandum No. UCB/ERL M84/9

15 January 1984

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

AN IC PROCESS-ORIENTED CIRCUIT SIMULATOR DEVELOPED FROM SPICE2G.6 USING THE CSIM MOSFET MODEL

Ralph C.L. Liu

Department of Electrical Engineering
and Computer Sciences
University of California, Berkeley

ABSTRACT

An IC process-oriented version of CSIM (Compact Short-channel IGFET Model) [1] has been implemented in the circuit simulation program SPICE2G.6. The model and simulator are IC process-oriented in that the electrical parameters for the active and parasitic circuit elements are internally evaluated, given the description of the circuit layout and a process file. The process file, which results from a separate parameter extraction program [2], is used by this version of the simulator for the process-related characteristics of these active and parasitic circuit elements. The process file method replaces the problem of maintaining a large number of separate electrical parameter sets with the maintenance of a few process files, one for each fabrication process. It is hoped that the format chosen for the process file will become a standard within the IC industry. Many different circuit designs employ the same IC process (process file) and differ only in the mask description (circuit layout). Since the simulator takes these two separate types of information as input, circuit design activities map in a one-to-one fashion with the standard IC manufacturing procedures. The consequence of the adoption of this methodology is expected to be a significant increase in designer productivity.

ACKNOWLEDGEMENT

I wish to express my gratitude to Professor D. O. Pederson who has given me continuous encouragement and support during the course of my research. Dr. D. L. Scharfetter has laid the foundation for this project and guided me throughout the process. His efforts and patience are greatly appreciated. The important contributions and help from B.J. Sheu are acknowledged. I am also thankful for the stimulating discussions with Dr. Tak Young, Jeff Burns, Brian Messenger and Kartikeya Mayaram. This report has been prepared with special aid from Jeff Burns.

An important part of this project was accomplished during Summer 1983 at American Microsystems, Inc. The discussions with S. Chen and A. Doganis at Xerox PARC were also very helpful to the work reported. This project has been supported by the Semiconductor Research Corporation under grant SRC-82-11-008 and by General Instruments, Inc.

TABLE OF CONTENTS

SECTION 1: INTRODUCTION 2

SECTION 2: IC PROCESS-ORIENTED CIRCUIT SIMULATION 6

SECTION 3: THE CSIM MODEL 10

SECTION 4: PROCESS FILE AND THE SIMULATION INPUT SYNTAX 23

SECTION 5: IMPLEMENTATION 27

SECTION 6: CONCLUSION 32

Appendix 1: Quadratic Interpolation for Beta0 33

Appendix 2: User Specification for CSIM1 34

Appendix 3: An Example of the Process File 35

Appendix 4: An Example of SPICE Input 40

Appendix 5: Linked List Configurations 42

Appendix 6: Subroutine Changes 45

REFERENCES 47

I. INTRODUCTION

This report describes a project directed towards the implementation of a new MOSFET model, the Compact Short-channel IGFET Model (CSIM) [1], into the circuit simulation program SPICE2G.6. In addition to the implementation of the model, a new integrated circuit *process-oriented* simulator structure has been developed using the existing SPICE2G.6 program as a starting point, and is documented in this report. The term process-oriented refers to a design methodology which encourages the designers of custom integrated circuits to mirror in their activities the procedures followed in actual IC manufacture. That is, the end product of a design effort is the *layout* of the circuit, which is used to generate a mask set. The actual integrated circuit is made from a semiconductor wafer and the mask set through a sequence of *processing* steps. In the process-oriented approach, the circuit design is carried out through the use of a description of the process and a description of the layout.

The masks of an IC are rigidly defined and unchanging. The processing steps associated with the substrate and each mask layer, however, are far from constant. Understanding the process variations, and their effective characterization and control, are keys to realizing functional circuits. Integrated circuit products which meet performance and yield goals result to a large extent from the ability of the circuit designer to evolve a design (mask set) appropriate for a given process. Hence the design activity is layout oriented, as is the input description to the simulator developed in this project. Performance evaluations over the range of the process result from circuit simulations performed for different process files which represent best, nominal, or worst case conditions, for example. Alternatively, they may represent nominal process characteristics

for different manufacturers. By contrast, to perform process-oriented circuit design utilizing previous versions of SPICE2, a designer would have to calculate for himself all the parameter values for each device model over the range of the process.

The process-oriented approach has several other benefits. For example, it emphasizes a different, more realistic way of viewing integrated circuits. Typically a circuit is considered to be composed of different devices and elements, such as transistors and resistors. However, an integrated circuit is actually composed of active areas and passive areas on a semiconductor wafer. The task of circuit simulation can be realized by analyzing the properties of these two kinds of areas. The process file contains information on the passive areas as well as the active areas. Thus, the process file approach encourages this alternative way of viewing an integrated circuit. Another issue addressed by the process file approach is that of exploiting the similarity of behavior of the devices in an IC, which occurs because they are all fabricated at the same time. This exploitation can be thought of as a means of simplifying the design process. The process file constitutes a database from which the electrical parameters of all devices of the same type are determined. The process-file orientation therefore directly addresses the similarity-of-behavior issue.

The process file is derived from measured I-V data by running a process parameter extraction program described elsewhere [2]. For one process, the parameters pertaining to all MOSFET devices as well as the parameters for the passive elements such as resistors and capacitors are obtained from one such file. The process file can be used to evaluate the process itself, the device design, and the circuit design. A goal of this research is that this process file

format will become an industry standard.

In summary, the following figure diagrams the approach.

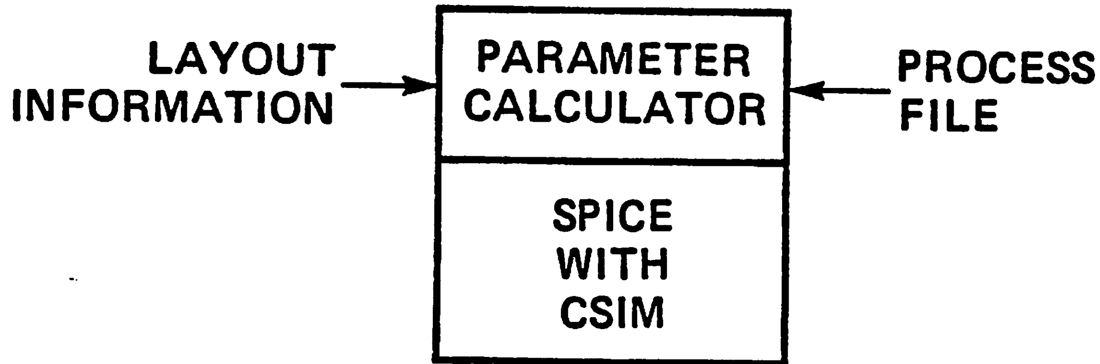

Fig. 1 Process-Oriented Circuit Simulation

Several device modeling concerns are addressed in this research. Device-size shrinkage in particular has raised challenging and possibly contradicting concerns with respect to the circuit-simulation tools used in the development of state-of-the-art IC designs. One concern is how to reduce the simulation time as the number of devices simulated becomes large. Another concern is in regards to developing models for smaller devices which assure the accuracy of the simulations. It is felt that the generation of models with increasing/decreasing levels of accuracy, corresponding to increasing/decreasing numbers of model parameters, will effectively facilitate engineering tradeoffs with respect to these concerns. The bipolar transistor model in SPICE2 has this property, but the existing MOSFET models do not.

The MOSFET model implemented in this project is an extended version of the CSIM model developed at Bell Labs [3]. The extensions result in a DC model with several levels of increasing accuracy, corresponding to the use of larger numbers of process parameters in the process file. At the higher levels, some key parameters are allowed to vary with terminal voltages. In addition, a new CSIM charge/capacitor model consistent with the DC model equations is briefly described in this report and has been installed in the simulator. Detailed descriptions of the CSIM DC and charge models are given elsewhere [4].

This report contains a detailed description of the process-oriented circuit simulator developed from SPICE2G.6 and an overview of the extended CSIM model. The format of the new process file and the new input syntax for the simulator are documented. Modifications to the internal data structure and the added subroutines required for the CSIM model and the process file are described. Finally some test results comparing the CSIM model with the present SPICE2 Level-1, Level-2, and Level-3 MOSFET models [5],[6] are presented.

II. IC PROCESS-ORIENTED CIRCUIT SIMULATION

In circuit simulation, mathematical models are used as idealizations of the physical devices in the circuit. Due to the complexity of transistor behavior, a transistor model usually has a number of *electrical* parameters ranging from 10 to 50, and they form a parameter set. To describe different parameter sets, different .MODEL commands are used in circuit simulation programs, such as SPICE2. When a circuit becomes large and contains many different types of transistors, many .MODEL commands are required. Also, different electrical parameter sets are often required for devices of the same type but differing sizes, because the parameters seldom scale well with transistor size. This large number of electrical parameter sets has led in some cases to the use of model libraries. An important activity auxiliary to circuit simulation is the characterization of devices and support of model libraries. As the size of a model library grows, there arises the problem of organizing the library in a way which is efficient for both device characterization and circuit simulation.

One solution of this problem is to organize the electrical parameter sets in the library by IC fabrication process. This can be accomplished by giving the parameter sets which characterize devices from the same IC process the same process name, with distinct subtitles for each different model. A further step is to combine all parameter sets with the same process name into a process file. For the CSIM model used in this project, the parameter sets for all devices of a single type can be reduced to a special set of parameters which are independent of transistor size. This special set of parameters is called the set of *process* parameters. Therefore, the process file contains only one set of process parameters for each device type.

A goal of this project has been to replace the maintenance of model libraries with the maintenance of process files. No standard has been established for the model library approach or the process file approach. Integrated circuit manufacturers have not provided a complete, verified characterization of their processes to the general designer community. By developing a process-oriented version of SPICE2, defining the format for the process file, and providing the software for process file generation, it is hoped that a new standard for such activities may result.

With the process parameter set from the process file and the device dimensions, the electrical parameters which are specific for a single transistor geometry are computed in a pre-processing step before the simulation. The I-V characteristics and gate-charge characteristics can then be predicted. The transistor geometries are obtained from the circuit layout, which is part of the input to the new simulator. The above methodology is shown in Fig. 1 on Page 4. The precalculation procedure is described more fully in Section III.

The behavior of the passive elements of an integrated circuit, such as resistors and capacitors, can be easily described by the proper process information and their dimensions. For example, the value of a polysilicon resistor can be determined from the polysilicon sheet-resistance per square and the effective length and width of the polysilicon line. Process information such as resistance per square and capacitance per unit area should be included in the process file, so that the effects of the passive elements of an integrated circuit can automatically be included in the simulation. The circuit topology and the appropriate layout information needed for these calculations must be provided in the input to the simulator.

A remaining issue is that of transistor parasitics. The characteristics of transistor parasitics are similar to the characteristics of specific passive elements in the same circuit. For instance, the source and drain regions (usually made on diffusion layers) of a transistor have certain resistance and capacitance values. The characteristics of the source and drain regions are similar to those of other diffusion areas fabricated by the same IC process. Therefore the process information for the passive devices can also be used for the parasitics of transistors.

In this research, the passive devices and transistor parasitics are called interconnections because these two types of elements can be viewed as the elements between active areas on an integrated circuit. The interconnections are made of polysilicon, metal or diffusion. There can be several different layers for each of the three materials on an integrated circuit. At present, only a single layer for each of them is implemented in the circuit simulation program. However, the addition of code for more interconnection layers is straightforward. The IC process data associated with the diffusion layer are:

sheet resistance per square
zero-bias bulk junction capacitance per unit area
zero-bias bulk junction sidewall capacitance per unit length
bulk junction saturation current per unit area
bulk junction potential
bulk junction bottom grading coefficient
bulk junction sidewall grading coefficient

The IC process data associated with the metal and polysilicon layers are the

same:

sheet resistance per square

capacitance per unit area

edge capacitance per unit length

To summarize, in process-oriented circuit simulation the integrated circuit is conceptually divided into active areas and interconnections (passive areas). The active areas, which are the gate regions of transistors, can be characterized by the CSIM model and its associated set of electrical parameters. A process parameter extraction program [2] uses measured I-V data from transistors of various sizes to produce the CSIM process parameter sets. The process parameters are independent of device size. The passive areas include resistors, capacitors and the source and drain regions of transistors. The size-independent process data characterizing the passive areas are also obtained from measurements. The information on the passive areas together with that on the active areas form the process file.

III. THE CSIM MODEL

The CSIM formulation was originally developed at Bell Labs by H.C. Poon [3], [7], [8]. This formulation is referred to as CSIM0 in this report. The CSIM0 current equations are built upon a set of 8 device-dependent model parameters which are called *electrical* parameters. The 8 electrical parameters of CSIM0 can be derived from a set of 18 size-independent parameters which are referred to as *process* parameters. In the work at Berkeley by B. Sheu, et al., [1] a new version of CSIM0, termed CSIM1, has been developed. Like CSIM0, CSIM1 contains 8 electrical parameters, derivable from a process file composed of 18 process parameters as shown in Figure 2. The major difference between CSIM1 and CSIM0 is that for CSIM0 the velocity saturation coefficient U1 has a piece-wise linear expression, while for CSIM1 U1 has a smooth expression.

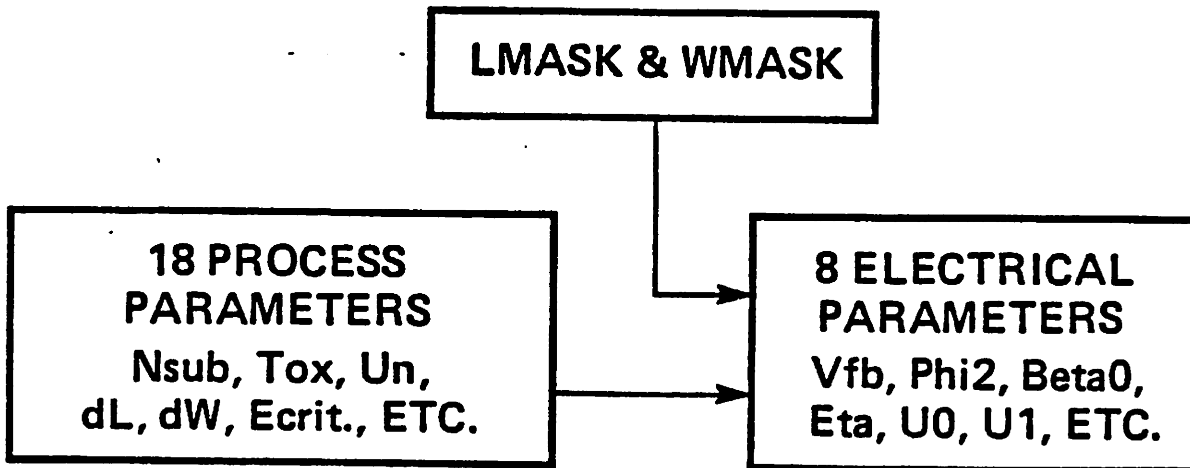

Fig.2 The Pre-Processing Phase of CSIM1

The research by Sheu, et al., also has revealed that the 8 basic electrical parameters vary with operating point. In particular the mobility degradation coefficient (U0) and the velocity saturation coefficient have a dependence on backgate bias V_{bs}. Also, the drain-induced barrier lowering coefficient (Eta) and the intrinsic-conductance coefficient (Beta0) have dependencies on both V_{bs} and V_{ds}. The operating-point variations of the parameters are accounted for in a new model called CSIM2. For a device of a particular size, the CSIM2 model contains 16 parameters. These are the original 8 plus an additional 8 parameters for the voltage dependencies. Among the additional 8 electrical parameters of CSIM2, four describe the dependence of Beta0 upon voltages V_{bs} and V_{ds}, two describe the dependence of Eta upon V_{bs} and V_{ds}, and the last two bear the effects of V_{bs} on U0 and U1. Figures 3 and 4 display some measured results showing the dependences mentioned here. The original 8 parameters show differing degrees of voltage dependence, as can be seen by noting the different numbers of added parameters associated with each original parameter.

It is worthwhile to mention specifically the five CSIM2 electrical parameters for Beta0, i.e., the original one plus the four new ones. The strong dependence of Beta0 on V_{ds} make it necessary to extract its value at two points, namely $V_{ds} = 0\,V$ and $V_{ds} = V_{dd}$, so that all the Beta0 values within this range can be estimated as accurately as possible. It has been found that the characteristics of Beta0 as shown in Fig.4 are adequately modeled by a quadratic in V_{ds}. Since quadratic interpolation requires three data values, the slope of the Beta0 characteristic at V_{dd} is chosen as the third one. The interpolation scheme is outlined more fully in Appendix 1. The Beta0 dependence on V_{bs} is approximately linear; thus, one more parameter is needed for this variation bringing the total number of added Beta0-related parameters to four.

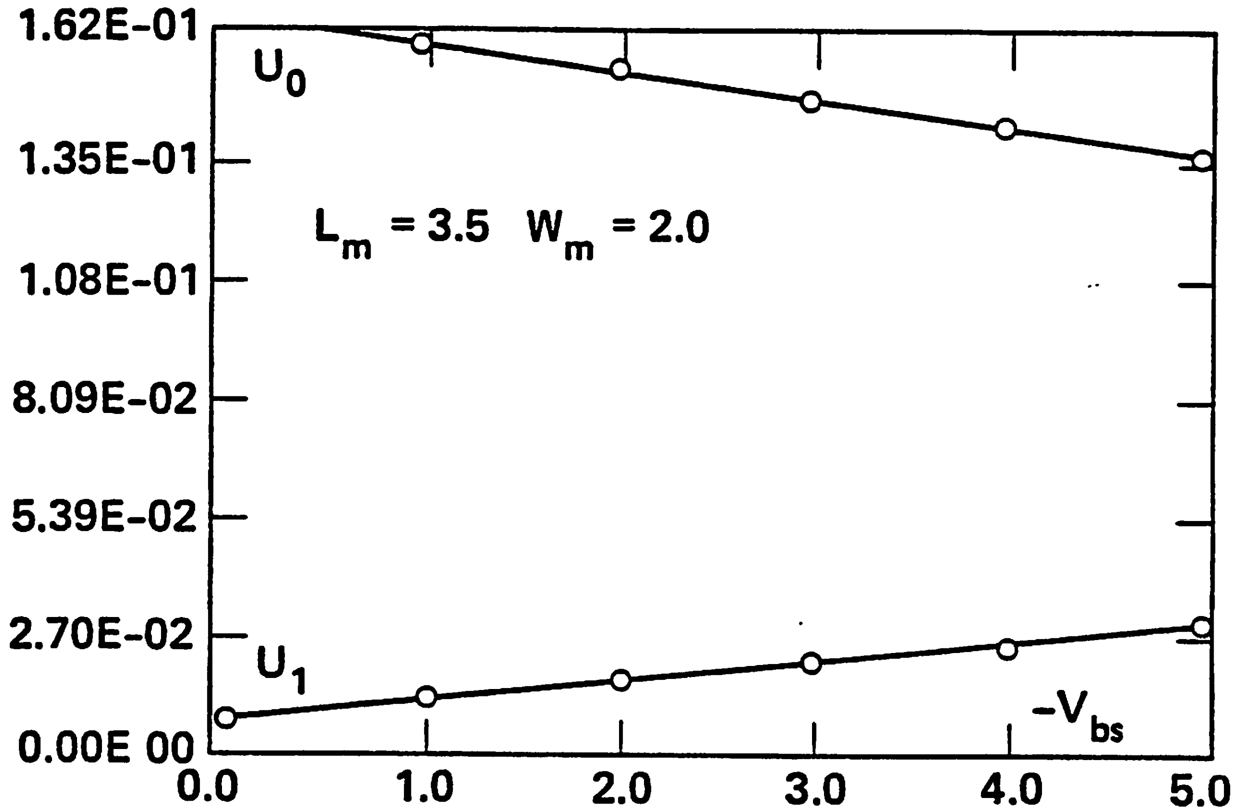

Fig. 3 U0/U1 Dependence on V_{bs}

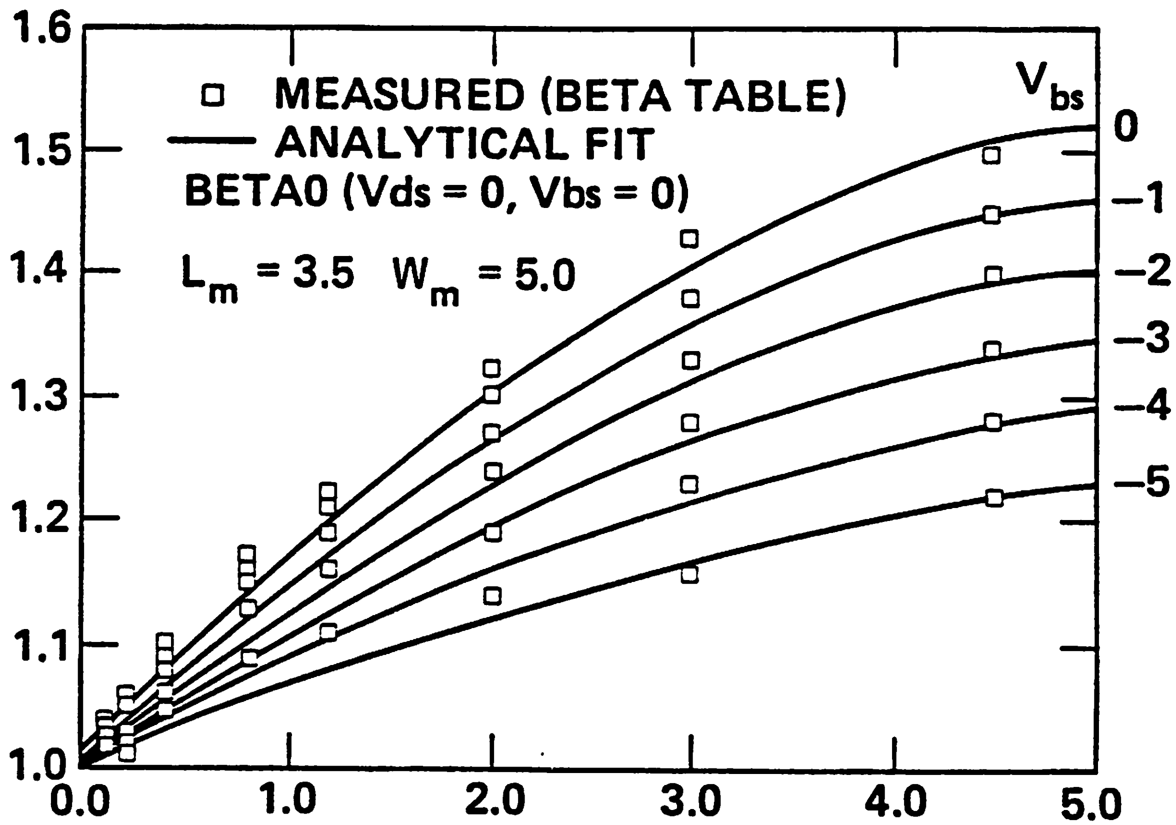

Fig. 4 Beta/Beta0 Versus V_{ds}

The methods used for the bias dependencies evolved from the algorithm developed to perform automatic model characterization. In particular, the model evolved with the constraint that all process parameters be obtainable from a fully automated parameter extraction (or process characterization) algorithm. The automated characterization project is described in a report by B. Messenger [2]. The result of the automated characterization, which utilizes I-V data from devices of different geometries, is a geometry-independent set of 51 CSIM2 process parameters. The following paragraph contains a description of the relationship between the 51 process parameters and the 16 electrical parameters.

The 16 geometry-specific CSIM2 electrical parameters, like the electrical parameters of CSIM1, are calculated from the process parameters and transistor sizes in a pre-processing step as shown in Fig. 6. The 16 CSIM2 electrical parameters in general have different dependencies on device length and width. To account for this, 32 additional parameters, 2 for each of the original 16, are included, making the total number of parameters 48. Such size dependence is typically described by the expression

$$P = P_0 (1 + P_l / L + P_w / W)$$

where W and L are the MOSFET's gate width and length, respectively. Thus, the original 16 parameter values, e.g., P_0, are the parameter values for a device with infinitely large length and width. As an example, Fig.5 shows the dependence of Beta0 on device length. There are 3 more CSIM2 parameters, none of which scale with transistor size. They are t_{ox}, temperature, and V_{dd}. Here, V_{dd} is the drain bias V_{ds} value at which Beta0 is measured (usually 5 volts). These 51 parameter numbers are obtained from the automated characterization program [2] referenced above, and they constitute the CSIM2 process parameters.

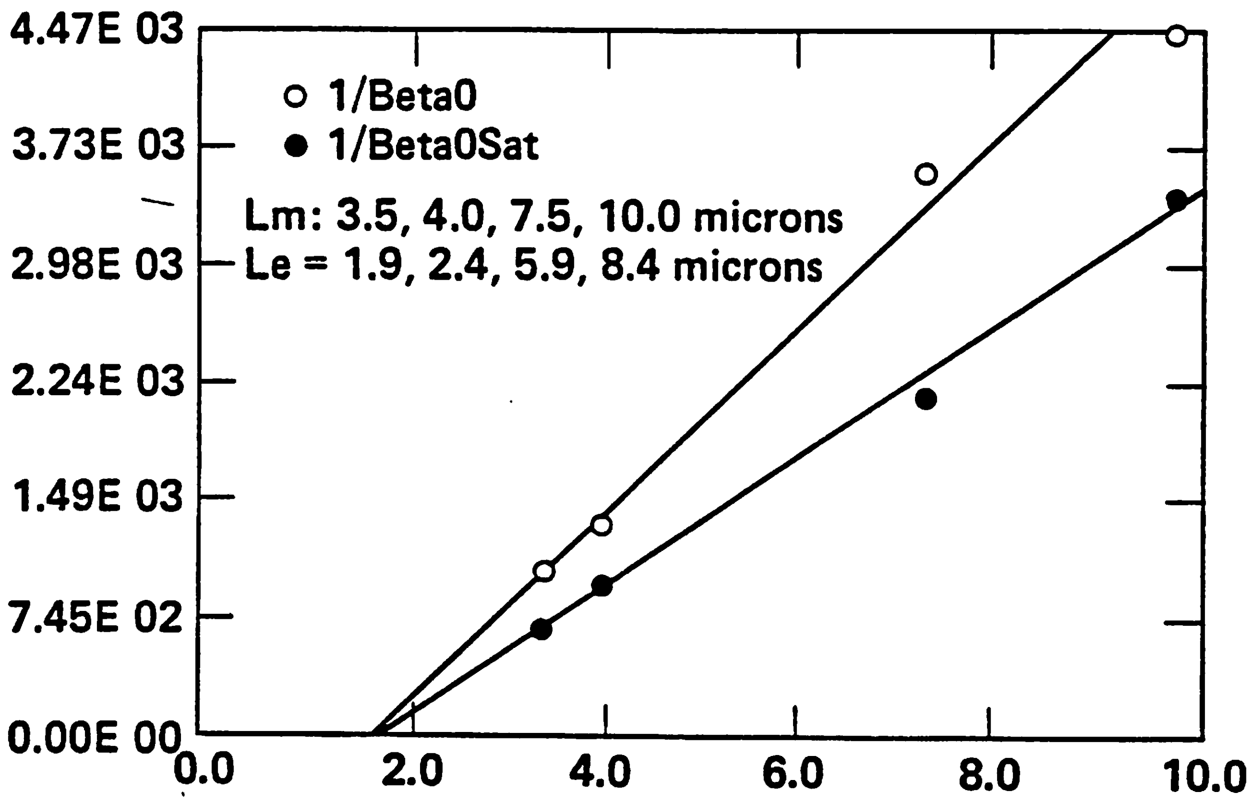

Fig. 5 1/Beta0 and 1/Beta0Sat Versus L

The CSIM1 process parameters account for length and width variation in a similar fashion.

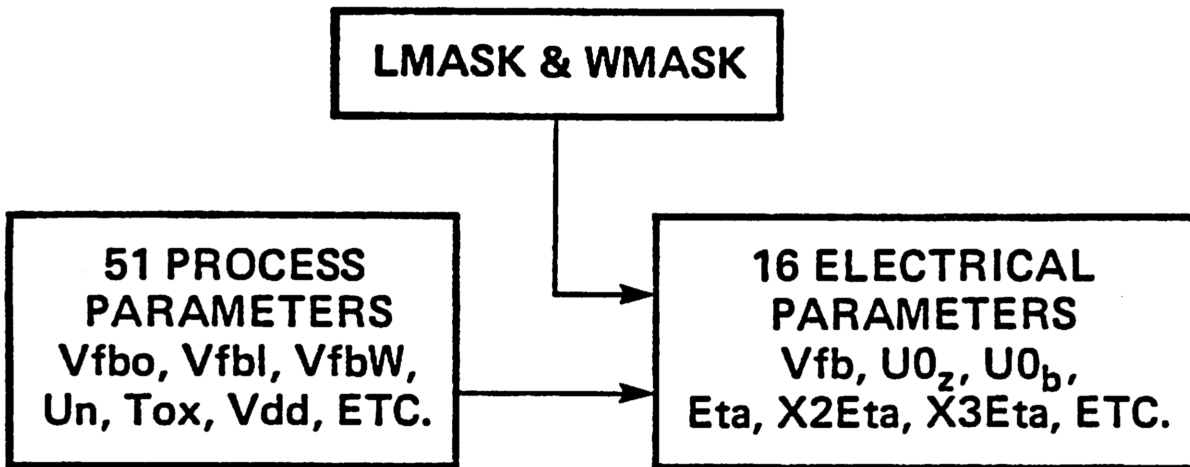

Fig.6 The Pre-Processing Phase of CSIM2

The models which have been implemented to date are CSIM2 and CSIM1. A third version of the CSIM model exists. This version has unreduced operating-point dependencies and is referred to as CSIM3 [1]. The CSIM3 form exists in anticipation of new phenomenon which will have to be accounted for as device sizes continue to shrink. When this occurs, the automated characterization procedure and the circuit simulation models can be upgraded to CSIM3, to allow characterization and simulation of more complex transistor behavior with minimal implementation delays.

It should be noted that the CSIM models have been implemented in a manner quite similar to the implementation of the SPICE2 bipolar model. In the

bipolar model, the more complex Gummel-Poon form has more model parameters than the less accurate Ebers-Moll model. In the CSIM formulation, one can take the model hierarchy to an even higher (perhaps table-driven) level than bipolar model, where the electrical parameter set is obtainable directly from measurements with little or no data reduction.

A companion CSIM charge/capacitance model [4] has been implemented in the simulator. In this model, charge is taken as the state variable [9] and the formulation guarantees charge conservation. The charge expressions are similar to those of [10]. This charge/capacitance model and its I-V characteristic counterpart, derived from the same considerations of MOS device physics, form a unified CSIM model previously unavailable.

The physical effects which are important for short-channel MOSFET devices, such as mobility degradation, velocity saturation and drain-induced barrier lowering, are taken into account in the CSIM model. Compared with the SPICE2 MOS Level-1 model, the CSIM model has higher accuracy. This is especially true for short-channel devices. Furthermore, the CSIM model equations are much simpler than those of the SPICE2 Level-2 MOS model [6]. The CSIM model and the SPICE2 Level-3 model are both semi-empirical models. They have similar equations [6]. The capacitance model derived by Meyer [11] is still used for Level-3. There is no well-developed parameter extraction program for this model; therefore, the user has to struggle with some of the rarely used model parameters. This scheme not only invites potential errors, it also wastes the user's time. In contrast, the Berkeley CSIM model has been especially formulated to allow for automated characterization in terms of a process file, and the I-V and Q-V implementation is self-consistent. Furthermore the CSIM1 model is

derivable from the same process file as CSIM2 (see Appendix 2).

At present only the DC equations of the CSIM model have been fully tested although the charge/capacitance model has also been installed in the simulator. Debugging and testing are currently in progress. The accuracy of the CSIM model has been tested by doing simulations using parameter sets from the parameter extraction program. Figures 7a, 7b, and 7c show that the CSIM1 model gives an excellent fit of the transistor current to the measured data.

The execution times for several different circuits using the CSIM1, CSIM2, and the SPICE2 Level-1, Level-2, and Level-3 models are listed in Table 1. It can be seen that in terms of CPU seconds-per-iteration the CSIM1 model is comparable with the SPICE2 Level-1 model. The CSIM2 model is as accurate as Level-2 or Level-3 but somewhat more efficient. The number of iterations for each model varies with each circuit.

The CSIM models also give good DC convergence results. Two circuits were tested, one being a three-stage CMOS ring oscillator. When calculating the operating point using the SPICE2 Level-2 model, there is no convergence even after 5000 iterations. If the CSIM1 model is used, however, convergence occurs after 12 iterations. The other circuit is a 26-transistor cascode opamp. Convergence of the simulation results after 996 iterations when using the SPICE2 Level-2 model, while with the CSIM1 model the iteration number reduces to 159.

CPU Seconds-Per-Iteration/Number of Iterations					
Circuit/Analysis	CSIM1	CSIM2	level1	level2	level3
2T inverter/DC transcurve	0.019/59	0.021/63	0.013/55	*	0.021/62
12T memcell/Op. point	0.088/17	0.092/17	0.074/55	0.22/6	0.13/14
27T opamp/Op. point	0.16/23	0.17/73	*	0.255/20	0.16/60
8T opamp/Transfer function	0.073/18	0.084/14	0.12/11	0.17/11	0.095/17

*: No convergence.

Table 1

The CSIM models perform well. However, it must be emphasized again that the most significant contribution of this work is in the process-oriented nature of the implementation.

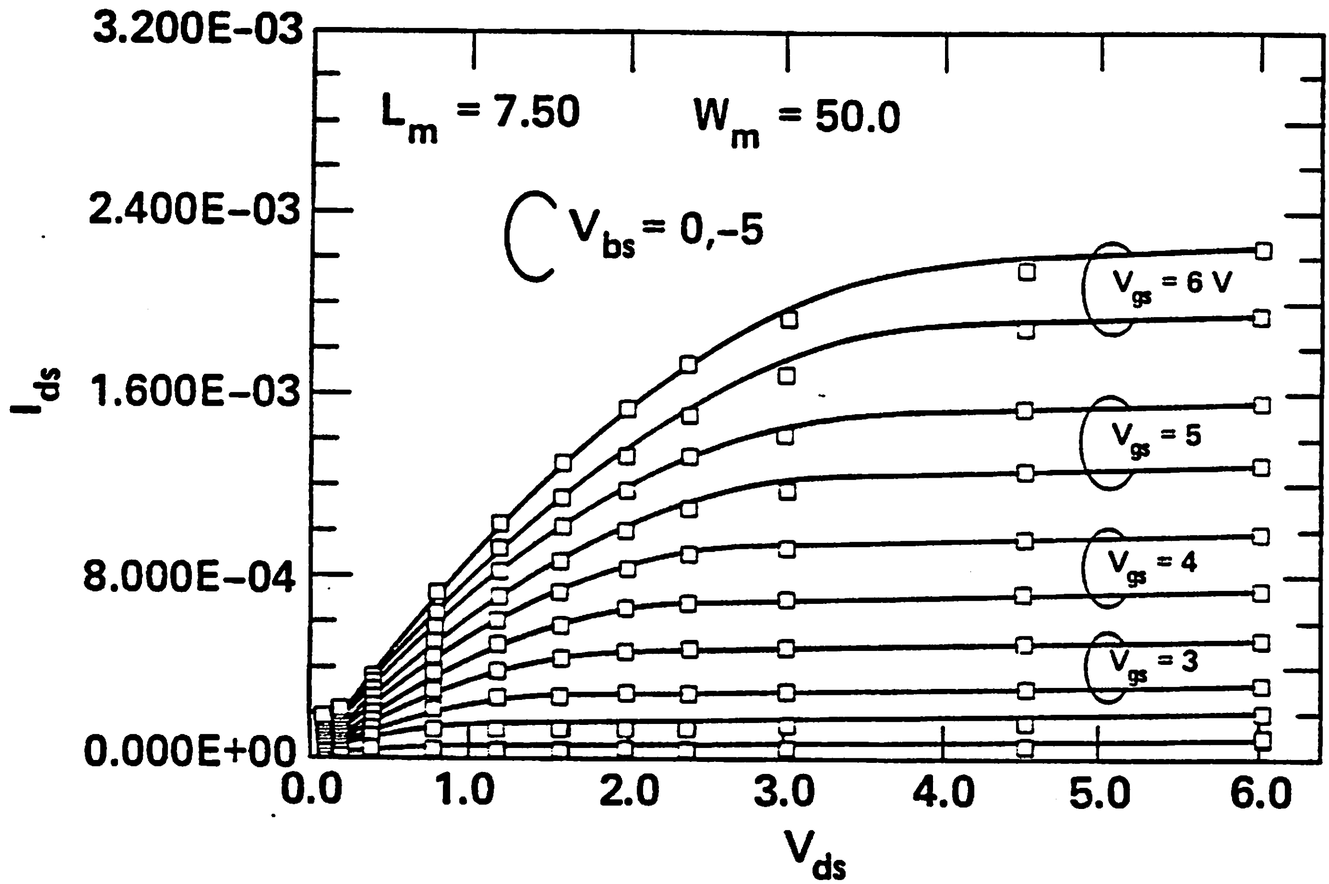

Fig. 7a Curves from CSIM1 Model and Measured Data Points

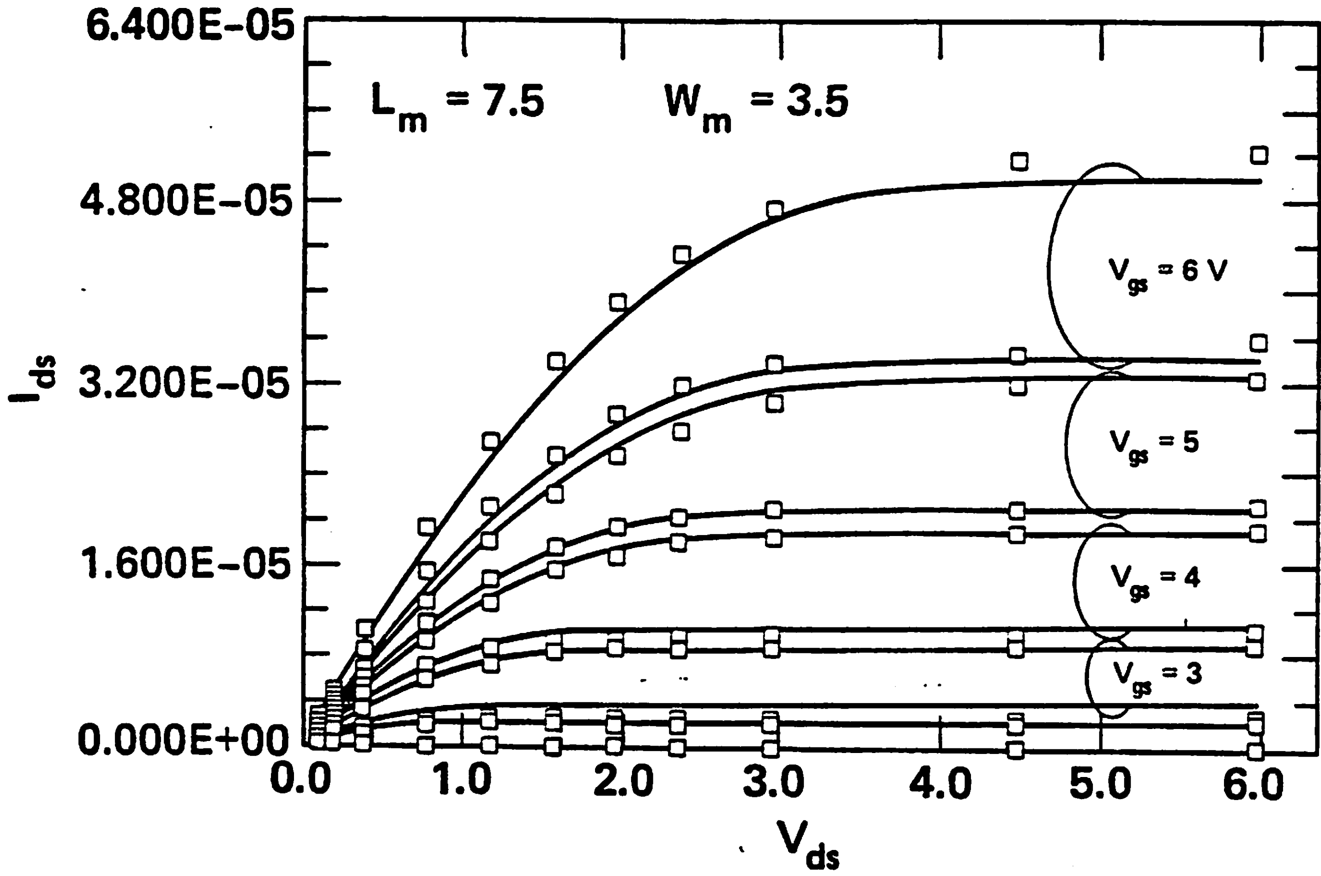

Fig. 7b Curves from CSIM1 Model and Measured Data Points

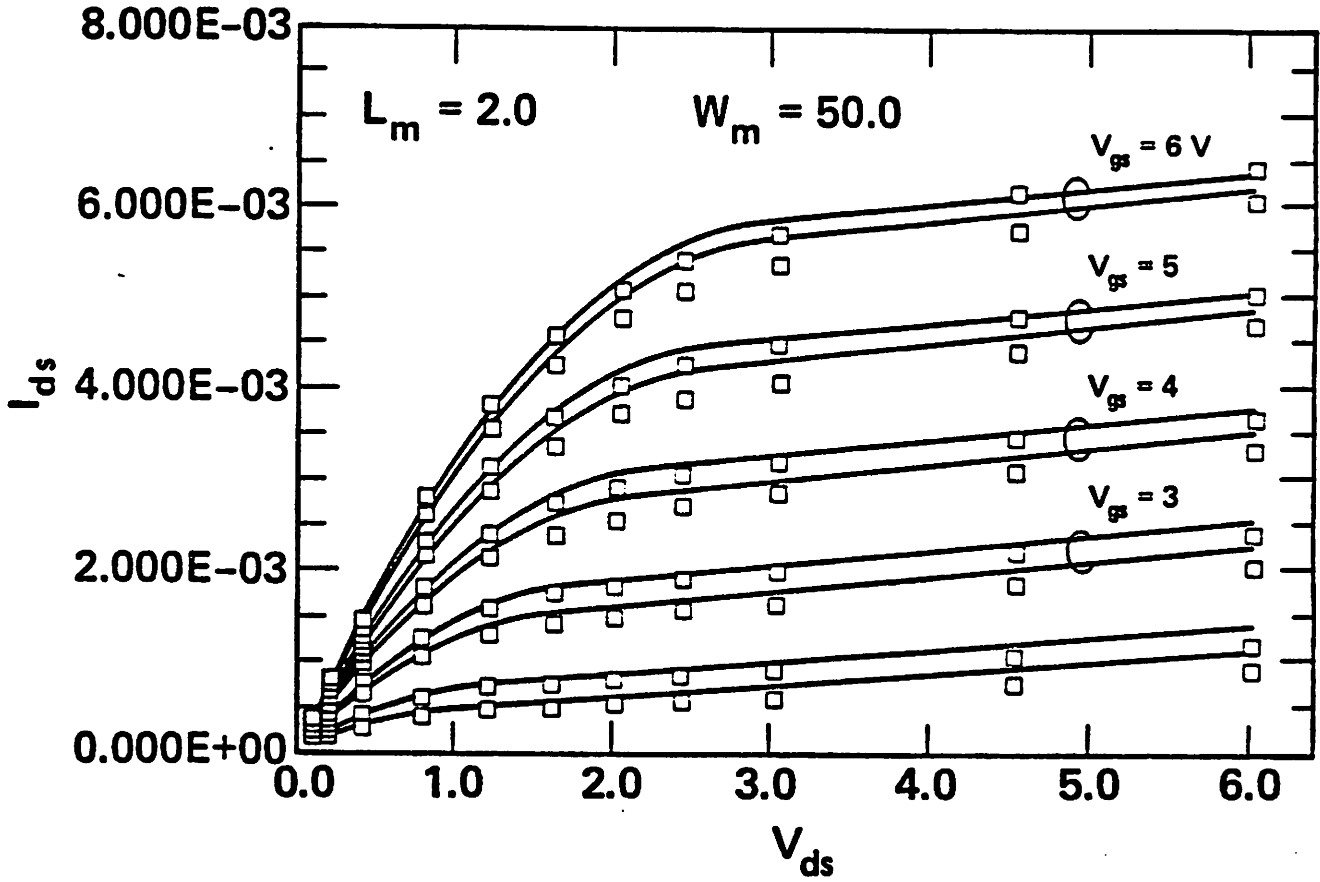

Fig. 7c Curves from CSIM1 Model and Measured Data Points

IV. PROCESS FILE AND THE SIMULATOR INPUT SYNTAX

As stated previously, the process file for the simulation program under development contains two kinds of information, the extracted CSIM process parameters appropriate for the active areas, and the process information for evaluating the parameters of the passive areas on the integrated circuit. Given the process file and the layout information, the circuit simulation program is able to calculate all the electrical parameters needed for the simulation. It is a nontrivial task to design the format for such a process file. There are several considerations regarding this issue presented in the following paragraphs.

The first requirement is that the process file be complete, i.e., it should contain all the information regarding the active and parasitic devices (MOSFETs, bipolars, diodes, resistors and capacitors) fabricated by one particular process. Different processes should have different files. This insures a one-to-one correspondence between a process and its process file. During the process characterization, all the extracted parameters from one wafer go to one process file. If it is necessary to simulate circuits composed of devices fabricated by different processes, that is, multi-chip simulation, more than one process file is used. The reduction of characterization data over many wafers to generate best or worst case process files is not addressed in the present work.

The process file should be kept like a record and should not be allowed to change arbitrarily. If for some reason the user wants to use some different process parameter values, the modified values should be specified in the circuit description as shown below. The process file itself should not be modified. This consideration is important in an industrial environment where changing the

process information in a large database is a management issue, of considerable significance and importance. It is also important to make the format of the process file unambiguous so that the simulation program can generate a unique internal data structure associated with the process information.

Based on these reasons the following format for the process file and the input of the simulator has been chosen. At the present time, there are only entries for MOSFET transistors and interconnections (polysilicon, diffusion, and metal layers). The following is a brief illustration of the format of the process file. Upper case denotes keywords in the first line. For a detailed example, see Appendix 3.

```
mos-type1  mos-type2 ... POLY  METL  DIFU
51 CSIM2 process parameters for mos-type1
51 CSIM2 process parameters for mos-type1
....
process data for polysilicon layer
process data for metal layer
process data for diffusion layer
```

The following are several notes about this format.

1. Mos-type includes keywords NMOS PMOS, and etc.

2. The line 'mos-type1 mos-type2 ... POLY ...' tells the simulation program how much information to be expected in the file. If a keyword like POLY or METL is missing, there is no such layer fabricated by the process.

3. The 51 CSIM2 parameters are the outputs of the parameter extraction program for the CSIM2 model. This program could be expanded later to produce

the process information of the polysilicon, metal, and diffusion layers as well.

The circuit input description format is as represented below:

(a) Resistor elements

RXXX node1 node2 process-name_layer-type L (W)

(b) Capacitor elements

CXXX node1 node2 process-name_layer-type L (W)

(c) MOSFET devices

SXXX nodeD nodeG nodeS nodeB process-name_mos-type L W
(AD AS PD PS IC=VDS,VGS,VBS)

(d) Process information

.PROCESS process-name FILENAME= process-file-name
.PROCESS process-name device-type par1=xxx par2=xxx ...

In the case of capacitors and resistors, the width parameter (W) should only be provided if the default value from the process file is not to be used. The process-names in a,b,c refer to the same process-name as referred to by the .PROCESS commands in (d). This is how an element or device is properly associated with its corresponding process information. AD, AS, PD, PS, IC have the same meaning as that in SPICE2 User's Manual [12]. Device-type means either mos-type or layer-type. The second .PROCESS command allows the user to change the specific values of par1, par2, etc. in the named process file. An example is given in Appendix 4.

The second .PROCESS command can also be used alone:

.PROCESS process-name device-type par1=xxx par2=xxx ...

This format gives the user the freedom to enter directly the values of par1, par2, etc. without referring to a process file. This is a convenient way to specify the parameter values of additional interconnect layers, as there are only a few interconnect layers built into the simulator at present.

V. IMPLEMENTATION DOCUMENTATION
(REQUIRED BY SPICE MAINTAINERS, ONLY)

The implementation of the CSIM model and the process-oriented simulator structure involve a large number of changes in the source code of SPICE 2G.6. These changes are described in terms of the data structure and the subroutines.

It was desired that the added CSIM model not affect the execution of the existing SPICE2 MOS models. Therefore the implementation of the former did not take the form of adding another level to the original three levels of MOS models. A new linked list is created for the CSIM device with internal ID=15. This linked list is similar to that used by the original SPICE2 MOS device [13] except in two places.

The first difference is that more entries are reserved to store the electrical parameters of each individual device. For CSIM2 there are altogether 16 such parameters; therefore, the additional entries VALUE(LOCV+16) to VALUE(LOCV+31) in the linked list are used for this purpose. The other difference is that there are two model pointers in the CSIM device linked list, while for SPICE2 there is only one. One of the two pointers of the CSIM device points to its parameters appropriate for the gate region and the other points to its parameters appropriate for the source and drain.

A new linked list with ID=25 is created to store the 51 extracted CSIM process parameters. This linked list is pointed to by all the MOS devices which use the process parameters stored in it. Another linked list with ID=26 is created for the process parameters of the interconnections. It has nine entries which

store the seven process parameters listed in Section II plus two more (default width of the layer and average variation of size due to side-etching or mask compensation). Polysilicon and metal layers also have these two data values. This linked list is pointed to by both CSIM devices and diffusion resisters, if it stores the process parameters of the diffusion layer. It is pointed to by resistor or capacitor elements if it stores the process parameters of polysilicon or metal.

The resistor and capacitor elements in the circuit formerly had no models, hence no pointers; now they have. For such an element two more entries are added to the original linked list, one for the layer type and the other for the model pointer. The layer type is represented by an integer code, +1 for polysilicon, 0 for metal and -1 for diffusion. During the read-in stage, the simulation program stores the corresponding integer for different key words (POLY, METL and DIFU) appearing on the line specifying the element. Later the integer code of the element is used in searching for the appropriate model to which the pointer entry of the element can point. The same strategy is used for CSIM MOS devices. Appendix 5 contains the configuration of the linked lists described above.

All the geometric sizes of resisters, capacitors and MOSFETs are stored in their own linked lists as before. The precalculation takes place during the error-check stage. The electrical parameters of each MOS device are calculated from the CSIM process parameters plus the length and width of the device and are stored in the linked list of each device. The resistor and capacitor values are calculated from the process information of the various layers plus the sizes of the elements. The capacitance at the bottom junction of a diffusion layer is voltage dependent; therefore, its value has to be determined during each

iteration.

There are 7 new subroutines written for the implementation. Their names and functions are described in the following:

1. Subroutine PROCHK.f:

This subroutine does a one-time pre-processing of device model parameters as described above. It also assigns default values to the parameters not specified in the input, puts limits on the process parameters if their values are not appropriate, and prints out a list of the electrical parameters of the MOS devices. Its role is similar to that of MODCHK.f for the SPICE2 Level 1, 2 and 3 MOS models.

2. Subroutine CSMEQ1.f:

The CSIM1 DC equations are implemented in this subroutine. Given all the electrical parameters and the specific terminal voltages of a MOS device, this routine evaluates the drain current and its derivatives. It also calls Subroutine CSIMQ.f to evaluate the charges associated with the gate, bulk and drain regions.

3. Subroutine CSMEQ2.f:

The CSIM2 DC equations are implemented in this subroutine. It is very similar to Subroutine CSMEQ1.f.

4. Subroutine CSIM.f:

This subroutine processes the MOS devices using the CSIM model for DC and transient analysis. It is similar to Subroutine MOSFET.f in the original SPICE

code. Only the companion CSIM charge/capacitance model is used for the charge storage effect associated with the thin oxide.

5. Subroutine CSIMQ.f:

This routine calculates the charge and capacitances associated with the thin oxide using the CSIM charge/capacitance model equations. It plays a role similar to that of Subroutine MOSQx.f in implementing the Ward-Dutton charge/capacitance model.

6. Subroutine CSMCAP.f:

Given the calculated charge and capacitances from Subroutine CSIMQ.f, this subroutine computes the equivalent conductances and total terminal charges for the CSIM devices. It is like Subroutine MOSCAP.f.

7. Subroutine FNDTYP.f:

This subroutine changes the pointers from resistors, capacitors and CSIM devices to the linked lists for their own models. It uses two keys to search for a match, the process name and the model type. It is similar to Subroutine FNDNAM.f in the original SPICE2 code.

There are about 14 other subroutines partially modified to implement the process-oriented simulator with the CSIM model. Their names are listed below and the specific changes are described in Appendix 6.

ADDELT.f	LNKREF.f	FIND.f
ERRCHK.f	READIN.f	LOAD.f

DCTRAN.f	MATPTR.f	CARD.f
MATLOC.f	NXTCHR.f	TRUNC.f
TOPCHK.f	ALTER.f	

VI. CONCLUSION

CSIM is an efficient short-channel MOSFET model whose parameter values are readily obtainable from measurements. Different levels of the model with increasing/decreasing numbers of parameters give the user the ability to exploit the trade-offs between accuracy and speed. The process-oriented approach provides an efficient, automated system for the evaluation of integrated circuit performance.

The remaining issues for future research include testing fully the CSIM charge/capacitance model. The internal data structures and efficient algorithms still need to be developed for the CSIM3 model. The issue of statistical modeling and design needs to be addressed.

The modifications to SPICE2 which resulted in the simulator described in this report have been carried out on the source code of the VAX/UNIX version of SPICE2G.6. The modified subroutines are stored in the directory '~ralph/project' on the UCBCAD computer of the Electronics Research Laboratory at UC Berkeley.

Appendix 1: Quadratic Interpolation for Beta0

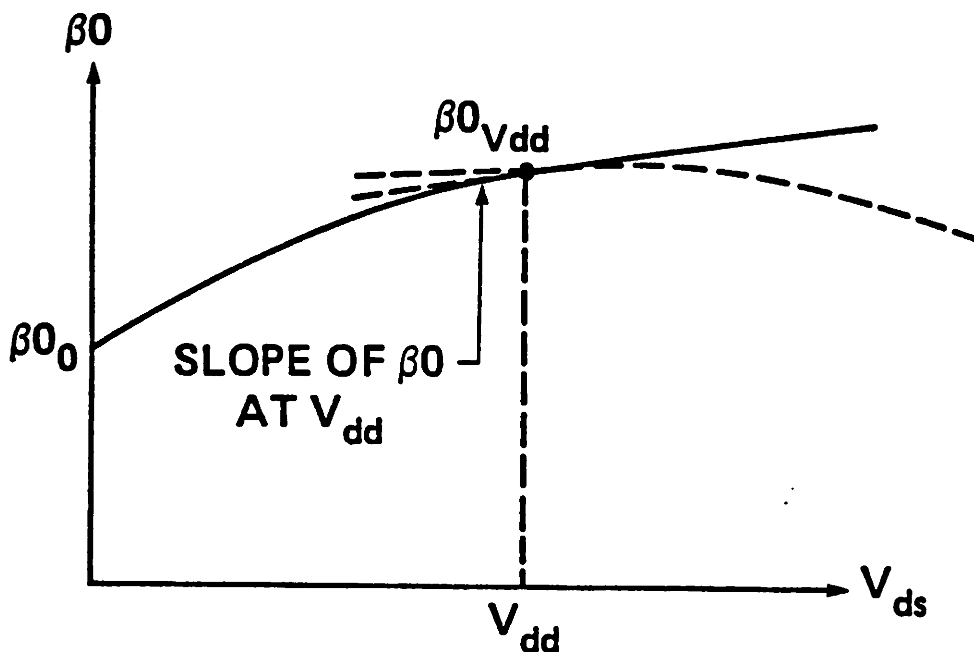

Fig.8 Quadratic Interpolation for Beta0

As shown above, the curve of Beta0 between $V_{ds}=0$ and $V_{ds}=V_{dd}$ is uniquely determined by three data: Beta0 values at 0V and V_{dd} plus the slope of Beta0 at V_{dd}. The curve may bend down in case $V_{ds} > V_{dd}$ (as shown by broken line), giving incorrect prediction for Beta0 value in that region. A linear interpolation is chosen at $V_{ds} > V_{dd}$ with constant slope = slope of Beta0 at V_{dd}, so that the undesired case is avoided. The code to realize this interpolation is in Subroutine CSIMEQ.f.

Appendix 2: User Specification for CSIM1

There are two ways the user can run SPICE with the CSIM1 formulation. One way is to give directly the CSIM1 parameter set (or a subset) in the .PROCESS command. The following is part of a sample input deck:

```
.PROCESS ETRAN NMOS (VFB=-0.034 NSUB=0.85e15 TOX=0.07 LK1=-0.75
+ WK1=0.0 K20=0.01 LK2=-1.96 WK2=0.0 ETAO=0.0 NETA=3.57 UN=817.6
+ VO=14.5 LU=-0.07 ECRIT=1.35 LV=9.58 DL=-0.5 DW=0.0)
```

The units used by the CSIM1 parameters are listed below:

PARAMETER NAME	UNIT	PARAMETER NAME	UNIT
Vfb	Volt	Nsub	10^15/Cm^3
Tox	Micron	Lk1	Micron
Wk1	Micron	K20	UnitLess
Lk2	Micron	Wk2	Micron
Etao	UnitLess	nEta	UnitLess
un	Cm^2/Volt-Sec	VO	Volt
Lu	Micron	Ecrit	Volt/Micron
Lv	Micron	dL	Micron
dW	Micron		

Another way for the user to choose the CSIM1 formulation is to specify V_{dd} = 0V in the .PROCESS command. The following line shows how this can be done:

.PROCESS process-name FILENAME=process-file-name

.PROCESS process-name device-type V_{dd}=0

Appendix 3: An Example of the Process File

This file is generated by the automated characterization program, and it contains the process information for the transistors as well as for the interconnections. For transistors, the names of the process parameters are defined by appending the italic letters '*l*' and '*w*' to the names the CSIM2 electrical parameters [1] to denote the dependence of that electrical parameter on the device length and width, respectively. For transistor and interconnection, the names of the process parameters used in the input of SPICE are contained in parentheses ( ). For example, the electrical parameter V_{fb} has 3 corresponding process parameters: Vfb (VFB), V_{fbl} (LVFB) and V_{fbw} (WVFB) and the names VFB, LVFB and WVFB are used in SPICE input. As mentioned in SECTION III, the value of an electrical parameter is obtained by the formula:

$$P = P_o(1 + P_l / L + P_w / W)$$

P_o, P_l and P_w are the process parameters associated with the eletrical parameter P. The names of the process parameters are listed as following:

TRANSISTORS

1	V_{FB} (VFB)	V_{FBl} (LVFB)	V_{FBw} (WVFB)
2	φ_S (PHI)	φ_{Sl} (LPHI)	φ_{Sw} (WPHI)
3	K_1 (K1)	K_{1l} (LK1)	K_{1w} (WK1)
4	K_2 (K2)	K_{2l} (LK2)	K_{2w} (WK2)
5	η_0 (ETA)	η_{0l} (LETA)	η_{0w} (WETA)
6	U_n (UN)	δ_l (DL)	δ_w (DW)
7	U_{0Z} (U0)	U_{0Zl} (LU0)	U_{0Zw} (WU0)
8	U_{1Z} (U1)	U_{1Zl} (LU1)	U_{1Zw} (WU1)

9	β_{ZB} (X2B)	β_{ZBl} (LX2B)	β_{ZBw} (WX2B)
10	η_B (X2E)	η_{Bl} (LX2E)	η_{Bw} (WX2E)
11	η_D (X3E)	η_{Dl} (LX3E)	η_{Dw} (WX3E)
12	U_{0B} (X2U0)	U_{0Bl} (LX2U0)	U_{0Bw} (WX2U0)
13	U_{1B} (X2U1)	U_{1Bl} (LX2U1)	U_{1Bw} (WX2U1)
14	β_S (BS)	β_{Sl} (LBS)	β_{Sw} (WBS)
15	β_{SB} (X2BS)	β_{SBl} (LX2BS)	β_{SBw} (WX2BS)
16	β_{SD} (X3BS)	β_{SDl} (LX3BS)	β_{SDw} (WX3BS)
17	T_{ox} (TOX)	T_{emp} (TEMP)	V_{dd} (VDD)

The names of the process parameters of diffusion layer are defined as following:

sheet resistance/square: Rsh

zero-bias bulk-junction capacitance/unit area: Cj

zero-bias bulk-junction sidewall capacitance/unit length: Cjw

bulk-junction saturation current/unit area: I_{js}

bulk-junction potential: Pj

bulk-junction bottom grading coefficient: Mj

bulk-junction sidewall grading coefficient: Mjw

default width of the layer: Wdf

average variation of size due to side etching or mask compensation: δ_l

The names of the process parameters of poly and metal layers are defined as following:

sheet resistance/square: Rsh

capacitance/unit area: Cj

edge capacitance/unit length: Cjw

default width of the layer: Wdf

average variation of size due to side etching or mask compensation: δ_l

INTERCONNECTIONS

1	Rsh (RSH)	Cj (CJ)	Cjw (CJW)
2	I_{js} (IJS)	Pj (PJ)	Mj (MJ)
3	Mjw (MJW)	Wdf (WDF)	δ_l (DL)

The following is an example of a process file. The lines starting with '*' are used as comments.

```
NMOS PMOS POLY METL DIFU
* NMOS TRANSISTOR
-0.572554713875,-0.35922842073,0
0.754246465933,0,0
0.930486754909,-0.57665832132,0
0.108221303676,-1.45112957046,0
0.0139077810548,-5.26375952793E-03,0
671.448801893,-0.440000740655,-2
0.127621721659,0.708395806671,0
-0.0319161798569,0.138866431887,0
1.8462602926E-06,5.42700786396E-06,0
-5.30846652441E-04,7.38801846014E-03,0
-3.9776377468E-03,0.011576566231,0
9.85803221363E-03,-0.0158729534534,0
```

-7.54488731758E-03,0.0137982152705,0

-0.917106532137,-0.974820183329,0

1.09755253947,-0.825214096472,0

0,0,0

0.03,26.84,5.0

* PMOS TRANSISTOR

-0.580856653889,-0.356603927807,0

0.759102989381,0,0

0.932744411147,-0.576542996217,0

0.108642571868,-1.44745086756,0

-0.180261543765,0.228099779629,0

171.449231504,-0.439998328651,-2

0.127622311862,0.708385270152,0

-0.0319211777796,0.138895274203,0

1.84626160916E-06,5.42719739833E-06,0

0.0574582123745,-0.0633061075104,0

-7.94337258283E-03,-4.97548601149E-03,0

9.85814641142E-03,-0.0158730914637,0

-7.54704150907E-03,0.01380235915,0

0.169583920305,0.289454336983,0

-0.0218203592984,-0.0227173315581,0

0.0129855740002,-0.0366006248674,0

0.03,26.84,5

* POLY LAYER

30,2.0E-04,1.0E-09,

0,0,0,

0,4.0,0.5

* METAL LAYER

0.5,1.5E-04,1.2E-09,

0,0,0,

0,4.0,0.5

* DIFFUSION LAYER

50,2.0E-04,1.0E-09,

1.0E-08,0.86,0.5,

0.33,4.0,0.5

Appendix 4: An Example of SPICE input

```
CASCODE OPAMP
IBIAS 34 51 4U
CLOAD1 5 0 PC1_DIFU 50U
CLOAD2 6 0 PC1_DIFU 60U
V1 1 0 AC 1
V2 2 0
V35 55 5
V75 75 5
V46 66 6
V86 86 6
S1 3 1 20 51 PC1_NMOS W=490U L=4U AD=980P AS=1800P
S2 4 2 20 51 PC1_NMOS W=490U L=4U AD=980P AS=1800P
S3 3 11 50 50 PC1_PMOS W=12U L=8U AD=48P
S4 4 11 50 50 PC1_PMOS W=12U L=8U AD=48P
S5 55 12 3 50 PC1_PMOS W=16U L=4U AD=40P AS=40P
S6 66 12 4 50 PC1_PMOS W=16U L=4U AD=40P AS=40P
S7 75 13 7 51 PC1_NMOS W=12U L=4U AD=20P AS=20P
S8 86 13 8 51 PC1_NMOS W=12U L=4U AD=20P AS=20P
S9 7 14 9 51 PC1_NMOS W=5U L=8U AD=40P AS=40P
S10 8 14 9 51 PC1_NMOS W=5U L=8U AD=40P AS=40P
S11 9 5 51 51 PC1_NMOS W=4U L=12U AD=18P AS=18P
S12 9 6 51 51 PC1_NMOS W=4U L=12U AD=18P AS=18P
S13 20 13 21 51 PC1_NMOS W=24U L=4U AD=40P AS=40P
S14 21 14 22 51 PC1_NMOS W=10U L=8U AD=2880P AS=2880P
S15 22 0 51 51 PC1_NMOS W=8U L=12U AD=180U AS=180P
```

```
S16 31 11 50 50 PC1 _PMOS W=6U  L=8U  AD=400P AS=400P
S17 32 12 31 50 PC1 _PMOS W=16U  L=4U  AD=54P AS=54P
S18 32 32 14 51 PC1 _NMOS W=3U  L=4U  AD=9P AS=9P
S19 14 14 33 51 PC1 _NMOS W=5U  L=8U  AD=128P AS=128P
S20 33 0  51 51 PC1 _NMOS W=8U  L=12U  AD=36P AS=36P
S21 11 11 50 50 PC1 _PMOS W=6U  L=8U  AD=400P AS=400P
S22 34 34 11 50 PC1 _PMOS W=4U  L=4U  AD=15P AS=15P
S23 0  32 13 51 PC1 _NMOS W=12U  L=4U  AD=36P AS=36P
S24 13 14 33 51 PC1 _NMOS W=5U  L=8U  AD=288P AS=288P
S25 12 11 50 50 PC1 _PMOS W=6U  L=8U  AD=400P AS=400P
S26 0  34 12 50 PC1 _PMOS W=16U  L=4U  AD=54P AS=54P
VDD 50 0  2.5
VSS 51 0 -2.5
*
*NOTE: "cmosproc" below could be the process file in Appendix 3.
*
.PROCESS PC1 FILENAME = cmosproc
.TF V(6,5) V1
.WIDTH OUT=80
.OPTIONS ACCT ITL6=50
.END
```

Appendix 5: Linked-List Configurations

A. The linked-list specification for original SPICE MOSFET device (ID=14). The LXi table is not included.

-1: subckt info	
LOC+0: next pointer	LOCV+0: element name
+1: LOCV	+1: channel length
+2: nd	+2: channel width
+3: ng	+3: drain diffusion area
+4: ns	+4: source diffusion area
+5: nb	+5: IC: vds
+6: nd'	+6: IC: vgs
+7: ns'	+7: IC: vbs
+8: model pointer	+8: device mode
+9: off	+9: von
+10: |	+10: vdsat
. } pointers to sparse	+11: drain perimeter
. | matrix locations &	+12: source perimeter
+32: | LXi table	+13: # square of drain diff.
+33: cycle number	+14: # square of source diff.
	+15: XQC

B. The linked list specification for CSIM MOS device (ID=15). The LXi table is not included.

-1: subckt info

LOC+0: next pointer	LOCV+0: element name
+1: LOCV	+1: channel length
+2: nd	+2: channel width
+3: ng	+3: drain diffusion area
+4: ns	+4: source diffusion area
+5: nb	+5: IC: vds
+6: nd'	+6: IC: vgs
+7: ns'	+7: IC: vbs
+8: model pointer 1	+8: device mode
+9: off	+9: von
+10: |	+10: vdsat
. |	+11: drain perimeter
. |	+12: source perimeter
. } pointers to sparse	+13: gate-source overlap cap.
. | matrix locations &	+14: gate-bulk overlap cap.
. | LXi table	+15: vto
. |	+16: |
+32: |	. |
+33: model type	. } electrical parameters
+34: model pointer 2	. |
+35: cycle number	+31: |

C. The linked list specification for resister (ID=1).

-1: subckt info

LOC+0: next pointer	LOCV+0: element name
+1: LOCV	+1: g (TEMP)

+2: |	+2: r (TNOM)
. } pointers to sparse	+3: temp.coef. 1
. | matrix locations	+4: temp.coef. 2
+7: |	+5: length
+8: model type	+6: width
+9: model pointer	
+10: cycle number	

* The changes made to the linked list of capacitor is similar to those with resistor.

D. The linked list specification for CSIM model (ID=25).

-1: subckt info	
LOC+0: next pointer	LOCV+0: process name
+1: LOCV	+1: |
+2: model type	. |
	. } 51 CSIM2 process parameters
	. |
	+51: |

E. The linked list specification for interconnection model (ID=26).

-1: subckt info	
LOC+0: next pointer	LOCV+0: process name
+1: LOCV	+1: |
+2: model type	. } 9 process parameters
	+9: |

Appendix 6: Subroutine Changes

Subroutines ADDELT.f, ALTER.f and FIND.f:

The major changes in these subroutines are made to create linked lists with ID=15, 25 and 26, and to expand the sizes of the linked lists with ID=1 and 2. Appendix 6 describes the configurations of those linked lists.

Subroutine ERRCHK.f:

The changes are to translate node initial conditions to device initial conditions when UIC is specified on the .TRAN card for devices with ID=15, and to assign default values of length, width and areas to those devices. The added Subroutine PROCHK is called by this subroutine.

Subroutine DCTRAN.f:

Subroutine CSIM.f is called by this subroutine. Line 348 is changed from: "1 +JELCNT(14)" to: "1 +JELCNT(14)+JELCNT(15)".

Subroutine LNKREF.f:

Lines 127 - 134 are inserted to resolve unsatisfied name references for the CSIM devices. Subroutine FNDTYP is called by LNKREF twice for such device to fix both model pointers (for reference, see Chapter V). Resisters and capacitors also call FNDTYP instead of calling FNDNAM as before.

Subroutine LOAD.f:

Subroutine CSIM is called by this subroutine.

Subroutines MATPTR.f, MATLOC.f and TRUNC.f:

Pieces of code are inserted in these subroutines to take care of the CSIM devices. The changes are easily found by the comments in front of the code.

Subroutine TOPCHK.f:

The line "1 1HQ,1HJ,1HM,0.0D0,0.0D0,1HT,0.0D0,0.0D0,0.0D0 /" is changed to "1 1HQ,1HJ,1HM,1HS,0.0D0,1HT,0.0D0,0.0D0,0.0D0 /".

Subroutine READIN.f:

There are more changes made in this subroutine than the previous ones. Variable strings BIDM(5), JPOLAR(5), NPROID(5), JPAR(3) and BMPAR(61) are created and initialized to accommodate the needs to read in the new CSIM and INTERCONNECTION parameters. One more entry is added to the string AIDC to allow the use of command .PROCESS in the input deck. Changes are made in the blocks of resistors, capacitors and devices to take care of the special syntax used in the input. The PROCESS command is treated like MODEL command except it is used for models with ID=25 and 26. The process file is opened and the parameters in it are read by this subroutine.

REFERENCES

[1] B. J. Sheu, D. L. Scharfetter and H. C. Poon, "Compact Short-Channel IGFET Model", to be published.

[2] B. Messenger, "A Fully Automated MOS Device Characterization System For Process-oriented Integrated Circuit Design", MS report, UCB/EECS Department, 1984.

[3] H. C. Poon, "A Compact Short-Channel IGFET Model (CSIM)", internal Bell Labs memorandum, April 2, 1976.

[4] B. J. Sheu, D. L. Scharfetter, C. Hu, and D. O. Pederson, "A New Charge Model for MOSFET Circuit Simulation", to be published.

[5] L. W. Nagel, "SPICE2: A Computer Program to Simulate Semiconductor Circuits", Memo. UCB/ERL M520, May 1975

[6] A. Vladimirescu, S. Liu, "The Simulation of MOS Integrated Circuits Using SPICE2," Memo. UCB/ERL M80/7, Oct. 1980.

[7] H. C. Poon, L. D. Yau, R. L. Johnston, and D. Beecham, "DC Model for Short-Channel IGFETs," *IEEE IEDM Technical Digest*, pp. 156- (1973).

[8] H. C. Poon, *Proceedings of Asilomar Conf. Circuits, Systems and Computers*, Pacific Grove, Ca, Dec. 1977, pp. 508-512.

[9] D. E. Ward and R. W. Dutton, "A Charge-Oriented Model for MOS Transistor Capacitances", *IEEE J. Solid-State Circuits*, vol. SC-13, No. 5, Oct. 1978.

[10] Ping Yang, B. D. Epler and P. K. Chatterjee, "An Investigation of the Charge Conservation Problem for MOSFET Circuit Simulation", *IEEE J. Solid-State Circuits*, vol. SC-18, No. 1, Feb. 1983.

[11] J. E. Meyer, "MOS Models and Circuit Simulation," *RCA Review*, vol. 32, Mar. 1971.

[12] A. Vladimirescu, K. Zhang, A. R. Newton, D. O. Pederson, A. Sangiovanni-Vincentelli, "SPICE Version 2G User's Guide",

[13] E. Cohen, "Program Reference for SPICE2", Memo. UCB/ERL M592, Jun. 1976.