NIST PUBLICATIONS

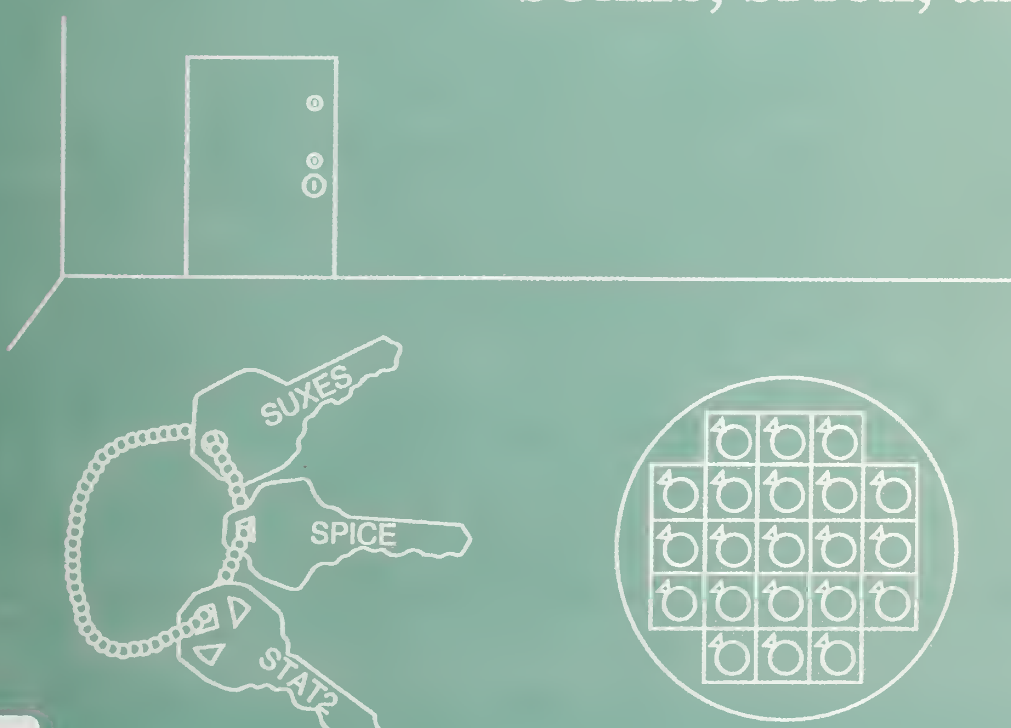
UNITED STATES OF AMERICA

NIST SPECIAL PUBLICATION **400-90**

U.S. DEPARTMENT OF COMMERCE/Technology Administration
National Institute of Standards and Technology

Semiconductor Measurement Technology:

Evaluating a Chip, Wafer, or Lot Using SUXES, SPICE, and STAT2



J. C. Marshall and R. L. Mattis

Semiconductor Measurement Technology:

Evaluating a Chip, Wafer, or Lot Using SUXES, SPICE, and STAT2

J. C. Marshall and R. L. Mattis

Semiconductor Electronics Division
Electronics and Electrical Engineering Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899

April 1992

U.S. DEPARTMENT OF COMMERCE, Barbara Hackman Franklin, Secretary
TECHNOLOGY ADMINISTRATION, Robert M. White, Under Secretary for Technology
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY, John W. Lyons, Director

National Institute of Standards and Technology Special Publication 400-90
Natl. Inst. Stand. Technol. Spec. Publ. 400-90, 145 pages (Apr. 1992)
CODEN: NSPUE2

U.S. GOVERNMENT PRINTING OFFICE
WASHINGTON: 1992

For sale by the Superintendent of Documents, U.S. Government Printing Office, Washington, DC 20402-9325

TABLE OF CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

Semiconductor Measurement Technology:
EVALUATING A CHIP, WAFER, OR LOT USING SUXES, SPICE, AND STAT2

Janet C. Marshall and Richard L. Mattis
Semiconductor Electronics Division
National Institute of Standards and Technology
Gaithersburg, MD 20899

Abstract

The computer procedure KEYS (linKing softwarE to analYze waferS) links SUXES (Stanford University eXtractor of modEl parameterS), SPICE (a Simulation Program with Integrated Circuit Emphasis), and STAT2. Given data points for individual devices, SUXES obtains the model parameters for SPICE. SPICE predicts the behavior of an individual device or an entire circuit. After analyzing each test chip on a wafer, STAT2 determines correlation coefficients and generates wafer maps of selected parameters. These wafer maps are valuable to the designer, modeler, and process engineer.

The entire package accomplishes the following: (1) standardizes the technique of running SUXES and SPICE in an integrated mode; (2) simulates and plots the characteristic curves; (3) simulates and plots the results of an optional dynamic circuit (for example, a ring oscillator); (4) performs steps (2) and (3) for every test chip on each wafer; (5) summarizes the results from each chip, each wafer, and the lot; (6) rank-orders the model parameters for each wafer according to their correlation coefficients with respect to chosen circuit parameters; and (7) generates wafer maps of several quantities. A CMOS 19-stage ring oscillator is used to illustrate the capabilities of KEYS.

Key words: circuit simulator; CMOS; ring oscillator; SPICE; STAT2; SUXES; wafer maps

NOMENCLATURE

Symbol	*Units*	*Identification*
ϵ_o	F/m	Permittivity of free space 8.85×10^{-12} F/m
ϵ_{ox}	F/m	Static dielectric constant of SiO_2 $3.9 \times \epsilon_o = 34.5 \times 10^{-12}$ F/m
ϵ_s	F/m	Static dielectric constant of silicon $11.7 \times \epsilon_o = 103.5 \times 10^{-12}$ F/m
ERR	%	The average error is the sum of the absolute value of the normal errors divided by the number of points. The normal error is the percent difference between the measured data points and SUXES's predicted data points.
I_{ds}	A	Drain-to-source current
k	J/(K)	Boltzmann's constant 1.38×10^{-23} J/(K)
L_n	m	Designed channel length for an *n*-channel MOSFET
L_p	m	Designed channel length for a *p*-channel MOSFET
MCKT	MHz	The measured circuit response, e.g., frequency
PDIFF	%	The percent difference between the measured and simulated circuit response is the absolute value of $[(\mathrm{SCKT} - \mathrm{MCKT})/(\mathrm{MCKT})] \times 100\%$ where it is assumed that the measured circuit response is the reference response.
q	C	Electronic charge = 1.602×10^{-19} C
RMS	V	The root mean square deviation between the measured threshold voltage curve and its linear least-squares fit when plotted against $\sqrt{\lvert V_{bs}\rvert + \mathrm{tphif}} - \sqrt{\mathrm{tphif}}$ where tphif=0.6 V.
SCKT	MHz	The simulated circuit response, e.g., frequency
T	K	Temperature
V_{bs}	V	Body-to-source voltage
V_{ds}	V	Drain-to-source voltage
V_{gs}	V	Gate-to-source voltage
W_n	m	Designed channel width for an *n*-channel MOSFET
W_p	m	Designed channel width for a *p*-channel MOSFET

1. INTRODUCTION

This report describes a computer procedure called KEYS (linKing softwarE to analYze waferS) that was developed at the National Institute of Standards and Technology. KEYS links SUXES (Stanford University eXtractor of modEl parameterS) [1],* SPICE (a Simulation Program with Integrated Circuit Emphasis) [2], and STAT2 [3] to facilitate integrated circuit evaluation. Other examples of packages for integrated circuit evaluation have been described [4,5]. A simplified block diagram for KEYS is given in figure 1.

SPICE predicts currents and voltages at chosen nodes in a circuit and requires a models file which contains input parameters for the devices in the circuit. The accuracy of these input parameters and the device models determines how well the simulated currents and voltages predict the measured currents and voltages. Since developing a models file for SPICE is a difficult task, a computer program called SUXES was developed. SUXES obtains most of these model parameters and adapts to any model. Given initial estimates for the parameter values, upper and lower bounds for these values, and measured data points for each device, SUXES optimizes the parameter values by varying them slightly, according to an initial estimates file, an options file, and a strategy file, so as to decrease the root mean square (rms) deviation between the predicted and measured data points. To aid process engineers, modelers, and designers, STAT2 determines correlation coefficients with respect to selected parameters and generates wafer maps of selected parameters. These maps provide a visual inspection of the positional variations of parameter values over a wafer.

The flowchart for the computer procedure KEYS is given in figure 2. To perform these tasks, additional FORTRAN programs as given in Appendix A were written. Command files connect these programs so that they may run as a single entity. The software used in KEYS is listed in table 1.

In the computer procedure KEYS, there are three sets of parameters: (1) the estimated input parameters for SUXES, (2) SUXES's optimized parameters, and (3) the model input parameters for SPICE. Given the data points for individual devices, SUXES obtains the model parameters for SPICE. The combination of SUXES and SPICE gives a procedure for parameterizing experimental data from circuits. SUXES's optimized parameters may not be physically meaningful. Nonetheless, in this report, the assumption is made that SUXES's optimized parameters have the same definition as the model input parameters for SPICE. Hence, the software to be presented directly inserts these parameters into the models file of SPICE.

The computer procedure KEYS characterizes a chip, wafer, or lot. In this report, a 19-stage

---

* Certain commercial equipment, instruments, and computer programs are identified in this paper in order to specify the experimental procedure adequately. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the equipment or program identified are necessarily the best available for the purpose.

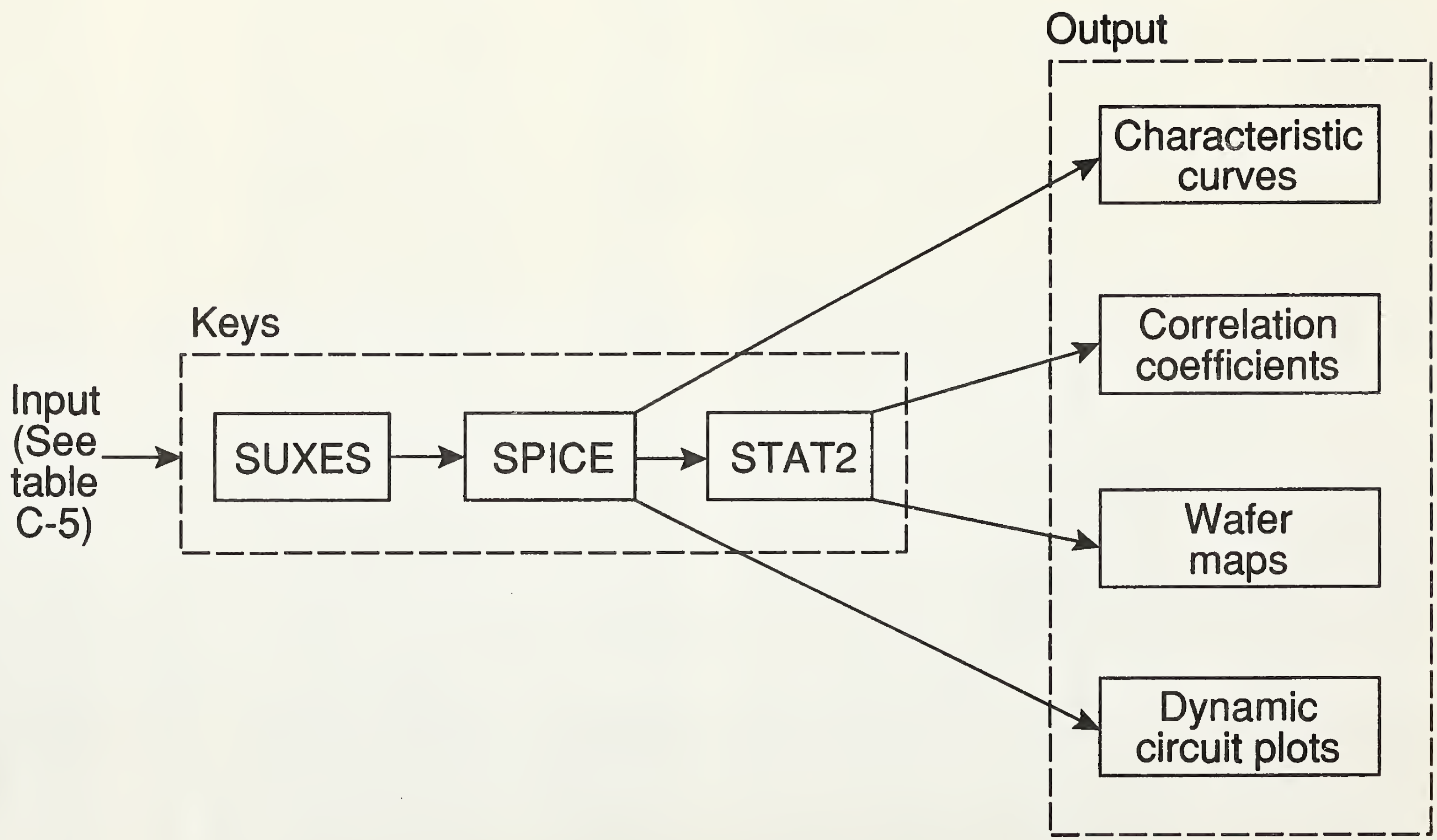


Figure 1. Simplified block diagram for the computer procedure KEYS.

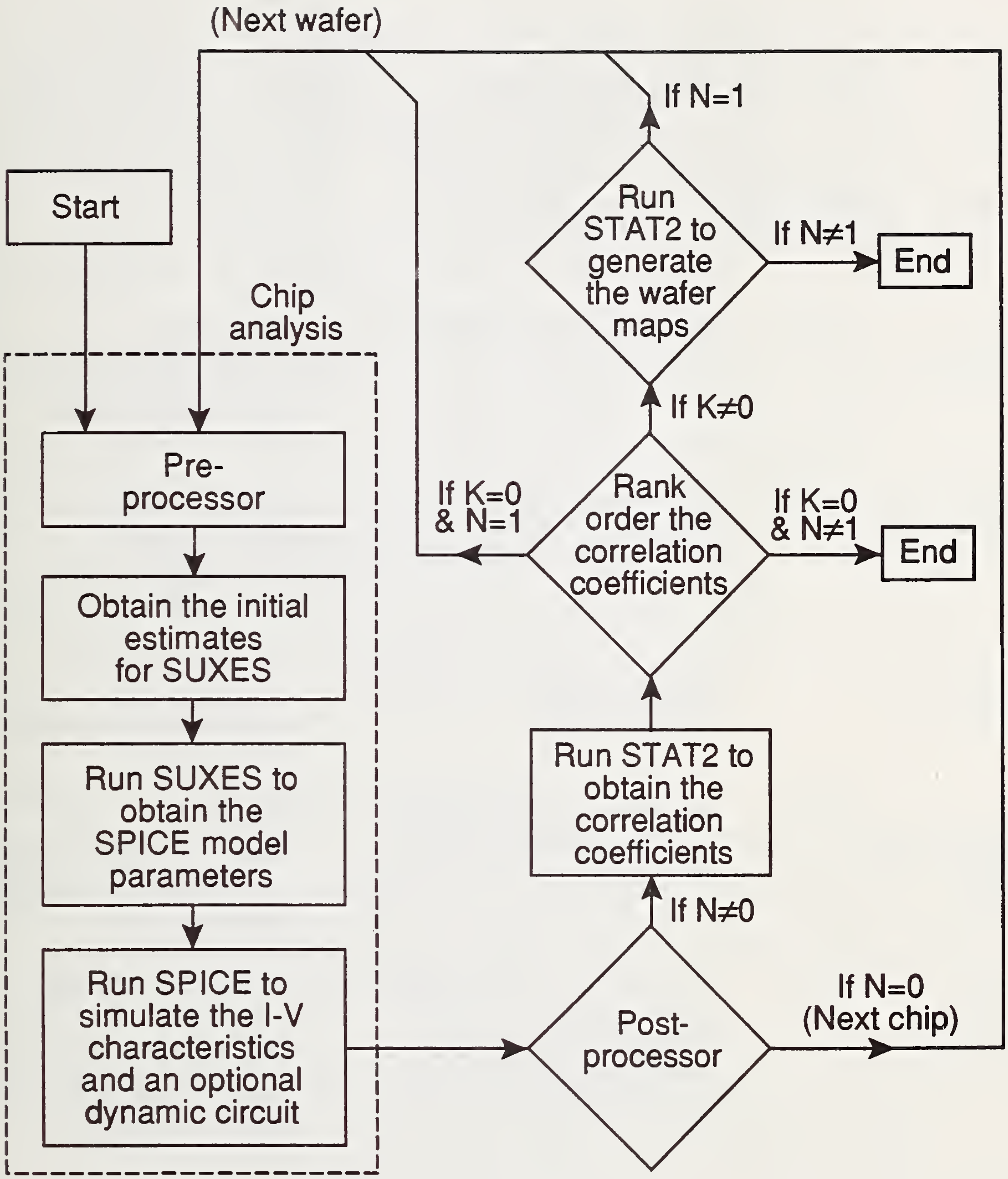


Figure 2. Flowchart for the computer procedure KEYS.

Table 1. A Description of the Programs Used in the Computer Procedure KEYS

PROGRAM NAME	# OF SOURCE PROGRAM STATEMENTS	PURPOSE	AUTHOR
SUXES (version I)	3792	Obtains the model parameters for SPICE	K. Doganis Electrical Engineering Software Santa Clara, Calif. (408) 296-8151
SPICE (version 2G.5)	43414	The circuit simulator which obtains the I-V characteristics and simulates a dynamic circuit	Univ. of Calif. Berkeley, Calif. with revisions by R. Mattis (301) 975-2235
NCAR Graphics MCVAX UTILITY	1764	Makes plots on the specified output devices	NCAR Graphics Information NCAR, Boulder, Colo. (303) 497-1201
STAT2 (version 1.31) CRDB CMLIB	16212	Determines the correlation coefficients and generates wafer maps	R. Mattis Electronics and Electrical Engineering Laboratory NIST, Gaithersburg, Md. (301) 975-2235
Misc. Files (version I)	1620	These include the command procedures, the editor files, the data files, and the FORTRAN files needed to link the software	J. Marshall, R. Mattis, and M. Gaitan Electronics and Electrical Engineering Laboratory NIST, Gaithersburg, Md. (301) 975-2052

ring oscillator is used to illustrate the procedure. This example takes approximately 11 CPU minutes per chip on a VAX-11/780. The entire package accomplishes the following: (1) standardizes the technique of running SUXES and SPICE in an integrated mode; (2) simulates and plots the characteristic curves; (3) simulates and plots the waveforms of an optional dynamic circuit (for example, a ring oscillator); (4) performs steps (2) and (3) for every test chip on each wafer if desired; (5) summarizes the results from each chip, each wafer, and the lot; (6) rank-orders the model parameters for each wafer according to their correlation coefficients with respect to chosen circuit parameters; and (7) generates wafer maps of selected parameters. The wafer maps are typically made for the most highly correlated parameters. Therefore, a careful selection of the chosen parameters with regard to the correlation coefficients will improve the usefulness of the wafer maps to process engineers, modelers, and designers.

This report contains three parts: Motivation and General Principles, An Example Using CMOS Technology, and the Conclusions. In the first part, Section 2.1 describes the linking of SUXES, SPICE, and STAT2. It is assumed that the reader is familiar with SUXES, SPICE, and STAT2, so these programs are only briefly described in sections 2.2, 2.3, and 2.4. In the second part, section 3.1 describes the test structure measurement methods, and section 3.2 discusses the output which consists of plots, tabular listings, and wafer maps. Appendix A contains the listing of the computer procedure KEYS. Appendix B presents a method of modifying SUXES's strategy file to decrease the rms deviation between the measured data points and SUXES's predicted data points while still providing realistic curves. Finally, Appendix C describes the procedure to run KEYS.

2. MOTIVATION AND GENERAL PRINCIPLES

2.1 Linking SUXES, SPICE, and STAT2

SUXES, SPICE, and STAT2 are the programs described in the following sections. SUXES supplies model parameters for SPICE, SPICE simulates circuits, and STAT2 obtains correlation coefficients and generates wafer maps. KEYS can be used to provide a smooth transition between these programs. The following three paragraphs demonstrate this and the last paragraph describes how it is done.

Typically, the estimated input parameters for SUXES are manually inserted into SUXES's initial estimates file. The estimated input parameters vary, for example, with oxide thickness. Hence, these values need to be calculated for each value of oxide thickness. KEYS can be used to perform these calculations and eliminate manual intervention.

Typically, the model input parameters for SPICE are manually inserted into the models file of SPICE. This becomes a task which is burdensome since each chip has its own set of parameters. KEYS can be used to insert directly SUXES's optimized parameters into the models file of SPICE. The values of the model input parameters for SPICE are also stored for future correlation coefficient analyses and possible wafer maps.

Typically, the values for the model input parameters for SPICE are manually inserted into files suitable for STAT2. KEYS can be used to place these values into the appropriate files to eliminate manual intervention. Thus, analyzing chips, wafers, and lots has become an easy task.

In smoothing the transitions between SUXES, SPICE, and STAT2, KEYS uses the EDT editor (copyright, Digital Equipment Corporation). Some editors can be imbedded in automated programs and can actually edit a file while the program is executing. EDT is an example of such an editor. This eliminates human intervention and error. A command procedure calls the file to be edited, and uses the editor commands in a specified file. Thus, it places the parameter values from SUXES into SPICE, and it modifies command procedures to reference the data files for the next chip. The linking of the three software packages is accomplished in this manner. The procedure to run KEYS is given in Appendix C.

2.2 SUXES

SUXES [1] obtains the model parameters for SPICE [2,6,7]. A model-independent Levenberg-Marquardt algorithm is used to minimize the rms deviation between the measured and predicted data points. This minimization makes it possible to perform a SPICE analysis more accurately. Given initial estimates for the parameter values, upper and lower bounds for these values, and measured data points for each device, the parameter values are optimized by varying them slightly, according to an initial estimates file, an options file, and a strategy file, so as to decrease the rms deviation between the predicted and measured data points. In the computer procedure KEYS, the parameters optimized by SUXES are indicated in table 2.

In KEYS the following input files are needed to run SUXES: the data file(s), the initial estimates file, the options file, and the strategy file. After the optimization, an extracted parameters file and a plot file are created. The extracted parameters file contains the optimized values of the parameters, and the plot file contains the predicted data points.

The data file(s) contain(s) the measured data points. The proper polarities are used for the devices. There is an exception: for *p*-channel MOSFETs, a positive current must be recorded, or the device must be treated as an *n*-channel MOSFET. The automated procedure ensures that the proper polarities are used in SUXES.

The initial estimates file includes the initial guesses for the parameter values, the lower bounds, and the upper bounds for these parameters. The parameter values are free to vary within these specified boundaries. The upper and lower bounds used in this analysis were obtained using the technique described in the SUXES report [1] or from the method described in Appendix B to find an appropriate strategy file. No attempt was made to quantify the values of these bounds (say $\pm 10\%$ or $\pm\sigma$ about the initial estimate), since each parameter has its own optimization sensitivity. The user may alter these values to increase the number of acceptable chips. In KEYS, an acceptable chip is one whose optimized SUXES values are not pegged at the lower or upper bounds. In the example with CMOS technology, there are two initial estimates files: NINPUT.DAT is for the *n*-channel

Table 2. A Brief Description of the Parameters Considered in this Analysis

PARAMETER	*UNITS*	*PARAMETER DESCRIPTION*
LEVEL	–	model index
TYPE	–	type of device: +1 *n*-channel device −1 *p*-channel device
*LD	m	lateral diffusion
*TOX	m	oxide thickness
*NSUB	cm^{-3}	substrate doping
*VTO	V	zero-bias threshold voltage
*KP	A/V^2	intrinsic transconductance parameter
*GAMMA	$V^{0.5}$	bulk threshold parameter
*PHI	V	surface potential at strong inversion
*UO	$cm^2/V\text{-}s$	surface mobility at low gate voltages
*DELTA	–	width effect on threshold voltage
*VMAX	m/s	maximum drift velocity of carriers
*XJ	m	metallurgical junction depth
*KAPPA	–	saturation field factor
*NFS	cm^{-2}	effective fast surface state density
*ETA	–	static feedback effect parameter
*THETA	V^{-1}	empirical mobility modulation parameter
CJSW	F/m	zero-bias bulk junction sidewall capacitance per meter of junction perimeter
CGDO	F/m	gate-drain overlap capacitance per meter channel width
CGSO	F/m	gate-source overlap capacitance per meter channel width
CGBO	F/m	gate-bulk overlap capacitance per meter channel length
RSH	$\Omega/\square$	sheet resistance of the drain and source diffusions
JS	A/m^2	bulk junction saturation current per square meter of junction area
TPG	–	type of gate material: +1 opposite to substrate −1 same as substrate 0 Al gate
MJ	–	bulk junction bottom grading coefficient
MJSW	–	bulk junction sidewall grading coefficient
PB	V	bulk junction potential
XQC	–	thin-oxide capacitance model flag and coefficient of channel charge share attributed to drain (0.0 – 0.5)
LAMBDA	V^{-1}	channel length modulation parameter
UEXP	–	critical field exponent
UCRIT	V/m	critical field for mobility degradation
NEFF	–	total channel charge coefficient
NSS	cm^{-2}	effective surface charge density

* The parameters found via SUXES, in this analysis.

MOSFETs, and PINPUT.DAT shown in table 3 is for the *p*-channel MOSFETs.

This list contains all the parameters that SUXES can optimize. In this analysis a subset of these parameters is used. The methods used to obtain the initial estimates for this subset of parameters are given in table 4. The other parameter values are initialized to the default values specified in this table.

The options file selects the desired data points from the data file(s), determines any weighting of the data, and determines the method of convergence or termination. In the example with CMOS technology, all the data points are used in SUXES. The options file for the *n*-channel MOSFETs is called NOPT.FIL, and the options file for the *p*-channel MOSFETs, shown in table 5, is called POPT.FIL. This table also provides the definitions of the convergence/termination criteria used in this file. Additional information can be found in the original report on SUXES [1].

The strategy file selects the order and groups of parameter values to be optimized. In the example with CMOS technology, the parameters whose values are optimized are: LD, TOX, NSUB, VTO, KP, GAMMA, PHI, UO, DELTA, VMAX, XJ, KAPPA, NFS, ETA, and THETA. Refer to table 2 for the parameter descriptions. The strategy file for the *n*-channel MOSFETs is called NSTRAT.FIL, and the strategy file for the *p*-channel MOSFETs, shown in table 6, is called PSTRAT.FIL. A parameter's value is optimized if a "1" appears in the column corresponding to the iteration of interest. A procedure is described in Appendix B which modifies the strategy file in order to decrease the rms deviation between the measured data points and SUXES's predicted data points. This procedure provides curves that are consistent with the measured data.

After the optimization is complete, SUXES generates a plot file. In the example with CMOS technology, the I_{ds} values are predicted for all combinations of V_{gs} and V_{ds} values. Therefore, if the voltages are measured, a tremendous number of I-V curves are generated. To save memory, to decrease the computer computation time, and to have a plot file of a manageable size, the measured voltage values in the data file(s) should be rounded or truncated before running SUXES. The automated procedure does this.

The capacitive parameters (CGDO, CGSO, and CGBO) are calculated after the optimization. The equations given in table 7 are used with the appropriate values from the optimization.

2.3 SPICE

SPICE performs dc, transient, or ac small-signal analyses of circuits [2,6,7]. Dynamic circuits are typically simulated using a transient analysis which predicts branch currents and node voltages as a function of time. A dc analysis determines the circuit's operating point, making it possible to obtain the characteristic curves. The ac small-signal analysis determines the ac variables as a function of frequency. In the example given, the I-V characteristics for the *p*-channel MOSFETs and the response of the optional dynamic circuit (the ring oscillator) were simulated using the SPICE files listed in tables 8 and 9, respectively. The files used to get the I-V characteristics for the *n*-channel MOSFETs are

Table 3. A *P*-Channel MOSFET Initial Estimates File of SUXES

*
* FILENAME: PINPUT.DAT
*

* parameters	initial	lowerbound	upperbound
*			
LEVEL	3	1	3
TYPE	−1.0	−1.0	1.0
LD	0.45	0.1	2.0
TOX	635.0	250.0	1000.0
NSUB	1.2×10^{16}	1.0×10^{13}	1.0×10^{17}
VTO	−0.85	−3.0	−0.1
KP	14.02×10^{-6}	1.0×10^{-6}	$100. \times 10^{-6}$
GAMMA	1.16	0.1	2.0
PHI	0.3618	0.3	0.9
UO	500.0	0.1	1250.0
UEXP	0.0	0.001	2.3
UCRIT	0.0	1.0×10^{4}	1.0×10^{6}
DELTA	0.6749	-1.0×10^{2}	1.0×10^{2}
VMAX	3.24×10^{5}	1.0×10^{4}	9.9×10^{7}
XJ	0.66	0.1	1.5
LAMBDA	0.0	1.0×10^{-6}	1.0×10^{-1}
KAPPA	2.0	0.0	999.0
NFS	1.17×10^{12}	1.0×10^{10}	4.0×10^{14}
NEFF	0.0	1.0×10^{-2}	1.0×10^{1}
NSS	1.0×10^{11}	0.0	1.0×10^{14}
TPG	−1.0	−1.0	1.0
ETA	0.1210	−10.0	10.0
THETA	0.2396	−10.0	10.0

Table 4. Methods Used to Obtain Initial Estimates for Input Parameters of SUXES

PARAMETER	*DEFAULT VALUE USED OR HOW OBTAINED*
LEVEL	Default value = 3
TYPE	*N*-channel MOSFET default value = 1.0 *P*-channel MOSFET default value = −1.0
LD	*N*-channel MOSFET default value = 0.38 μm *P*-channel MOSFET default value = 0.45 μm
TOX	Default value = 63.5 nm
NSUB	$NSUB = (\frac{1}{2(\epsilon_s)q})[\frac{GAMMA(\epsilon_{ox})}{TOX}]^2$
VTO	(1) This is measured at $V_{ds} = 0.2$ V and $V_{bs} = 0.0$ V for the *n*-channel devices. (2) This is measured at $V_{ds} = -0.2$ V and $V_{bs} = 0.0$ V for the *p*-channel devices. (3) A five-point maximum slope technique is used. (4) When the maximum slope is found, a straight line is extrapolated to the x-axis and 0.5 V_{ds} is subtracted from the V_{gs} value to obtain the threshold voltage. (5) VTO = the threshold voltage
KP	(1) This is measured at $V_{ds} = 0.2$ V and $V_{bs} = 0.0$ V for the *n*-channel devices. (2) This is measured at $V_{ds} = -0.2$ V and $V_{bs} = 0.0$ V for the *p*-channel devices. (3) $KP = \frac{maxslope(L)}{V_{ds}(W)}$ (4) An unreasonable value of KP causes the computer program to abort.
GAMMA	(1) GAMMA = |slope| of the threshold voltage vs. $[\sqrt{\|V_{bs}\| + tphif} - \sqrt{tphif}]$ (2) Assume tphif = 0.6 V. (3) For the *n*-channel MOSFETs, measurements are taken at $V_{bs} = -3.0$, −6.0, and −9.0 V for $V_{ds} = 0.2$ V. (4) For the *p*-channel MOSFETs, measurements are taken at $V_{bs} = 3.0$, 6.0, and 9.0 V for $V_{ds} = -0.2$ V. (5) A five-point maximum slope technique is used. When the maximum slope is found, a straight line is extrapolated to the x-axis and 0.5 V_{ds} is subtracted from the V_{gs} value to obtain the threshold voltage. (6) A recorded default value for a measured threshold voltage causes the computer program to abort.
PHI	$PHI = 2(\frac{kT}{q})\ln(\frac{NSUB}{n_i})$ where $n_i = 1.45 \times 10^{10}$ cm^{-3} at 300 K The simulated ring oscillator frequencies vary by less than 0.2 % for values of n_i between 1.00×10^{10} cm^{-3} and 1.45×10^{10} cm^{-3}.

UO	$UO = \frac{KP(TOX)}{\epsilon_{ox}}$
UEXP	This is not a MOS3 SPICE input parameter. Default value = 0.0
UCRIT	This is not a MOS3 SPICE input parameter. Default value = 0.0
DELTA	The value of this parameter is preset to be the value it typically is from previous SUXES runs. *N*-channel MOSFET default value = 2.685 *P*-channel MOSFET default value = 0.6749
VMAX	The value of this parameter is preset to be the value it typically is from previous SUXES runs. *N*-channel MOSFET default value = 7.9×10^5 m/s *P*-channel MOSFET default value = 3.24×10^5 m/s
XJ	$XJ = \frac{LD}{0.8}$
LAMBDA	This is not a MOS3 SPICE input parameter. Default value = 0.0
KAPPA	The value of this parameter is preset to be the value that helps the theory match the experiment. *N*-channel MOSFET default value = 0.80 *P*-channel MOSFET default value = 0.20
NFS	The value of this parameter is preset to be the value it typically is from previous SUXES runs. *N*-channel MOSFET default value = 2.28×10^{12} cm^{-2} *P*-channel MOSFET default value = 1.17×10^{12} cm^{-2}
NEFF	This is not a MOS3 SPICE input parameter. Default value = 0.0
NSS	Default value = 10^{11} cm^{-2}
TPG	*N*-channel MOSFET default value = 1.0 *P*-channel MOSFET default value = −1.0
ETA	The value of this parameter is preset to be the value it typically is from previous SUXES runs. *N*-channel MOSFET default value = 0.1306 *P*-channel MOSFET default value = 0.1210
THETA	The value of this parameter is preset to be the value it typically is from previous SUXES runs. *N*-channel MOSFET default value = 0.3612 V^{-1} *P*-channel MOSFET default value = 0.2396 V^{-1}

Table 5. A *P*-Channel MOSFET Options File of SUXES

```
*
*  FILENAME:    POPT.FIL
*
*  Nsign = Number of significant digits
*  Maxfn = Maximum number of function evaluations
*  Delta = Norm of gradient bound
*  Epsil = Sum of squares difference
*  Linit = Marquardt parameter initial value
*  Lscal = Marquardt parameter scale
*  Luppr = Marquardt parameter upper value
*  Fcdsw = Forward/central difference switch
*
  Criteria
  Nsign=4 Maxfn=200 Delta=1.0 × 10^-9 Epsil=1.0 × 10^-6
  Linit=0.1 Lscal=2.0 Luppr=1000 Fcdsw=0.1
*
  RangVd    min=-5.5    max=0.00    incr=0.0
*
  RangVg    min=-5.5    max=0.00    incr=0.0
*
  RangVb    min=0.0    max=0.0    incr=1.0
*
*  Weight
*
```

Table 6. A *P*-Channel MOSFET Strategy File of SUXES

```
*
*  FILENAME:    PSTRAT.FIL
*
*
*  parameters     first       second      third       fourth      fifth
*                 iteration   iteration   iteration   iteration   iteration
   LEVEL          0           0           0           0           0
   TYPE           0           0           0           0           0
   LD             0           0           1           0           0
   TOX            0           1           0           0           1
   NSUB           0           0           1           0           0
   VTO            0           1           0           0           1
   KP             0           0           1           0           0
   GAMMA          0           1           0           0           1
   PHI            0           0           0           1           0
   UO             0           0           0           1           0
   UEXP           0           0           0           0           0
   UCRIT          0           0           0           0           0
   DELTA          1           1           0           0           1
   VMAX           0           0           0           1           0
   XJ             0           0           1           0           0
   LAMBDA         0           0           0           0           0
   KAPPA          0           0           0           1           0
   NFS            0           1           0           0           1
   NEFF           0           0           0           0           0
   NSS            0           0           0           0           0
   TPG            0           0           0           0           0
   ETA            1           0           0           0           0
   THETA          1           0           0           0           0
```

Table 7. The Methods Used to Obtain Several Input Parameters for SPICE

PARAMETER	*DEFAULT VALUE USED OR HOW OBTAINED*
CJSW	Default value = 0.6×10^{-9} F/m
CGDO	CGDO = $\frac{\epsilon_{ox} LD}{TOX}$ where LD is the lateral diffusion
CGSO	CGSO = CGDO
CGBO	CGBO = $\frac{\epsilon_{ox} GO}{2TOX}$ where GO is the design rule of the gate overlap = 4 μm
RSH	*N*-channel MOSFET default value = 11.63 $\Omega/\square$ *P*-channel MOSFET default value = 61.57 $\Omega/\square$
JS	Default value = 5.0×10^{-5} A/m^2
MJ	Default value = 0.5
MJSW	Default value = 0.5
PB	Default value = 0.9 V
XQC	Default value = 0.45

Table 8. A SPICE File to Obtain the I-V Characteristics for a *P*-Channel MOSFET

```
*
*
* FILENAME = PMODEL.FIL
*
*
.MODEL MODP PMOS LEVEL=3 LD=0.450UM TOX=0.0631UM NSUB=0.789E+15
+ KP=10.30U GAMMA=0.186 PHI=0.766 UO=205.7 DELTA=3.025 VMAX=315790.
+ XJ=0.874UM KAPPA=25.8 NFS=0.300E+12 ETA=0.0798 THETA=0.1628
+ CJSW=0.6E-9 CGDO=0.246E-09 CGSO=0.246E-09 CGBO=0.109E-08
+ RSH=61.57 JS=5.0E-5 TPG=-1.0 MJ=0.5 MJSW=0.5 PB=0.9 XQC=0.45
+ VTO=-0.715V
*
*
*
* FILENAME = IVCHARP.FIL
* NOTE:    Use with PMODEL.FIL to make IVCHARP.CEL
*
* P-CHANNEL MOS OUTPUT CHARACTERISTICS
*
M2 1 2 0 0 MODP L=2.33UM W=6.33UM AD=42.25P AS=42.25P PD=26U
+ PS=26U NRD=1.0 NRS=1.0
*
VD 4 0
VG 2 0
VIDS 4 1
*
.OPTIONS NODE NOPAGE
.DC VD 0 -5 -.5 VG 0 -5 -1
.PLOT DC I(VIDS)
.END
```

Table 9. A SPICE File for a 19-Stage CMOS Ring Oscillator

```
*
* FILENAME = NMODEL.FIL
*
.MODEL MODN NMOS LEVEL=3 LD=0.408UM TOX=0.0558UM NSUB=0.778E+15
+ KP=29.87U GAMMA=0.302 PHI=0.591 UO=574.5 DELTA=1.659 VMAX=794207.
+ XJ=0.475UM KAPPA=24.2 NFS=0.228E+13 ETA=0.0939 THETA=0.1767
+ CJSW=0.6E-9 CGDO=0.252E-09 CGSO=0.252E-09 CGBO=0.124E-08
+ RSH=11.63 JS=5.0E-5 TPG=1.0 MJ=0.5 MJSW=0.5 PB=0.9 XQC=0.45
+ VTO=0.784V
*
* FILENAME = PMODEL.FIL
*
.MODEL MODP PMOS LEVEL=3 LD=0.450UM TOX=0.0631UM NSUB=0.789E+15
+ KP=10.30U GAMMA=0.186 PHI=0.766 UO=205.7 DELTA=3.025 VMAX=315790.
+ XJ=0.874UM KAPPA=25.8 NFS=0.300E+12 ETA=0.0798 THETA=0.1628
+ CJSW=0.6E-9 CGDO=0.246E-09 CGSO=0.246E-09 CGBO=0.109E-08
+ RSH=61.57 JS=5.0E-5 TPG=-1.0 MJ=0.5 MJSW=0.5 PB=0.9 XQC=0.45
+ VTO=-0.715V
*
*
* FILENAME = INVERT.FIL
*
.SUBCKT INVERT 1 2 3
M1 100 1 0 0 MODN L=2.33UM W=2.33UM AD=12.5P AS=12.5P PD=15U
+ PS=15U NRD=2.0 NRS=2.0
M2 100 1 3 3 MODP L=2.33UM W=6.33UM AD=42.25P AS=42.25P PD=26U
+ PS=26U NRD=1.0 NRS=1.0
R1 100 2 1332
C1 2 0 0.043P
.ENDS INVERT
*
*
* FILENAME = X19.FIL
* NOTE:    Use with NMODEL.FIL, PMODEL.FIL and INVERT.FIL to make X19.CEL
*
X1 1 2 3 INVERT
X2 2 4 3 INVERT
X3 4 5 3 INVERT
X4 5 6 3 INVERT
X5 6 7 3 INVERT
X6 7 8 3 INVERT
X7 8 9 3 INVERT
X8 9 10 3 INVERT
X9 10 11 3 INVERT
REND 11 200 1110
CEND 200 0 .044P
X10 200 12 3 INVERT
X11 12 13 3 INVERT
X12 13 14 3 INVERT
X13 14 15 3 INVERT
X14 15 16 3 INVERT
```

```
X15 16 17 3 INVERT
X16 17 18 3 INVERT
X17 18 19 3 INVERT
X18 19 20 3 INVERT
X19 20 400 3 INVERT
RCIR 400 1 334
CCIR 1 0 .073P
RPP 1 101 1.175
CPP 101 0 .322P
CPROBE 101 0 .1P
*
.IC V(1)=5 V(2)=0 V(4)=5 V(5)=0 V(6)=5 V(7)=0 V(8)=5
.IC V(9)=0 V(10)=5 V(11)=0 V(12)=5 V(13)=0 V(14)=5 V(15)=0
.IC V(16)=5 V(17)=0 V(18)=5 V(19)=0 V(20)=5
.IC V(3)=5
*
VCC 3 0 5V
*
.OPTIONS VNTOL=10UV LIMPTS=500 ITL4=50 ITL5=0
.TRAN 1.0NS 200NS UIC
.PLOT TRAN V(1)
.PLOT TRAN V(2)
.PLOT TRAN V(4)
.END
```

---

similar to those of the p-channel MOSFETs. MOS3 was the model used in SPICE 2G.5 for all the simulations.

To perform the simulations, a models file is needed to characterize the devices. It is assumed that SUXES's optimized parameters have the same definition as the input model parameters for SPICE. Hence, these parameters are directly inserted into the models file of SPICE. SUXES supplies most of the model parameter values. Several additional parameters (CGDO, CGSO, and CGBO) are calculated solely from the optimized SUXES values according to the equations given in table 7. In the computer procedure KEYS, the parameters LEVEL and TYPE are predetermined (as specified in table 4) and not allowed to be changed by SUXES. Several parameters are used in SPICE but cannot be optimized with the current version of SUXES. They are: CJSW, RSH, JS, MJ, MJSW, PB, and XQC. These parameters are considered noncritical [8]. The parameters mentioned above are briefly described in table 2, and the default values used in this analysis are listed in table 7.

In addition to the models file, SPICE requires that the user supply the circuit elements and connections, any voltage or current sources, the initial conditions, the type of analysis, the output desired, and other pertinent SPICE information.

To obtain realistic simulations, especially as device dimensions are reduced, it becomes necessary to include parasitic elements in the simulations. The parasitic elements included in the example given in this report are the source and drain diffusion resistances, the polysilicon resistances and capacitances, the metal resistances and capacitances, and the probe capacitance.

The two most common errors that occur when simulating a circuit are the inability of the values of the node potentials in the circuit to converge and the internal timestep being too small in the transient analysis. To avoid these two problems, the following can be done:

(1) Initialize each node in the circuit to an appropriate starting potential.
(2) Use the options on the .OPTIONS card as shown in table 9.
(3) If all else fails, modify the models file. Perhaps a different set of parameters is needed or the values for the existing parameters are unacceptable. For comparison, refer to the models file included in table 9.
(4) For additional help, consult reference [9].

2.4 STAT2

STAT2 is used to determine correlation coefficients [3,10,11] and to generate wafer maps for KEYS. The sample correlation coefficient, r, is a measure of the similarity of the spatial variation of two sets of data. Consider the paired observations (x_1, y_1), (x_2, y_2), ..., (x_n, y_n) on two quantities. If a large value of x_i implies a large value of y_i, then the quantities are said to be positively correlated, and r is positive. If a large value of x_i implies a small value of y_i, then the quantities are said to be negatively correlated, and r is negative. If a large value of x_i implies nothing about y_i, then x and y are said to be uncorrelated, and r is near zero. The statistic r is given as follows:

$$r = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{[\sum_{i=1}^{n}(x_i - \overline{x})^2][\sum_{i=1}^{n}(y_i - \overline{y})^2]}}, \tag{1}$$

where $\overline{x}$ and $\overline{y}$ are the arithmetic means of the x_i and y_i values of the sample summed over the n points. Note that r must take on values in the range $[-1,1]$. Stronger positive or negative correlation is indicated by r approaching $+1$ or -1, respectively.

The output of SUXES produces a set of files readable by STAT2. There is one file for each parameter of interest. An example of such a STAT2-readable file is shown in table 10. Site location information can be understood with the help of figure 3. The first 16 records of the file contain the location information for the data values which follow. The first of the 16 records gives the location of the first sites in rows 1 through 4. The second record gives the location of the last sites in rows 1 through 4. In the example of table 10, the first row of the wafer has test sites in columns 4 through 6, the second row of the wafer has test sites in columns 2 through 8, etc. The third and fourth records give the location of the first and last site in row 5. When all rows have been characterized, remaining positions within the first 16 records of the file contain zeroes. The data values are given in records 17 through the end of the file. They are ordered by rows, such that values from the first row of sites are first, followed by values from the second row of sites, and so on for as many rows of sites as are present.

In order for STAT2 to calculate correlation coefficients, it is necessary to create a database and enter the processed output of SUXES into the database. The database is created by the CRDB program (a program which is adjunct to and supplied with STAT2) [3] such that data from all test sites are included in the database. A sequence of commands such as that shown in table 11 directs STAT2 to enter the processed output of SUXES into the database. Each of the STAT2-readable files is assigned and read (STAT2 commands ASG and REA), values less than 10^{-12} are excluded (XLT), statistics are calculated for the remaining sites (PRS), replacement values are calculated for the excluded sites (AXP), and the data set is written to the database (WDB). A total of 43 parameters are thus processed. Because the first record in the database serves as a directory, writing of actual data begins at record 2 or entry number 2 and ends at entry number 44.

Having constructed the database, correlation coefficients are calculated using the STAT2 GET and SDB commands. As a default, the measured frequency, simulated frequency, and the percent difference are used as reference data sets (database entries 4, 5, and 6) with which all other data sets are correlated. The resulting correlation coefficients are sorted in KEYS by magnitude. Table 12 contains an example of the output. This table shows at a glance the parameters which are most influential in determining the measured frequency, simulated frequency, and percent difference.

It is often useful to see the spatial variation of one or more parameters over the wafer. Using a sequence of commands such as that shown in table 13, STAT2 can produce gray-tone maps similar to the one shown in figure 4.

Table 10. Example of a File Suitable for Input to STAT2*

4 2 1 2	! Starting column locations for rows 1 through 4
6 8 9 8	! Ending column locations for rows 1 through 4
4 0 0 0	! Starting column location for row 5
6 0 0 0	! Ending column location for row 5
0 0 0 0	! Zero values indicate the array of test sites
0 0 0 0	! does not go beyond row 5. The 64
0 0 0 0	! numbers in the first 16 records of
0 0 0 0	! the file provide for site locations
0 0 0 0	! in an array having up to 32 rows
0 0 0 0	! and 32 columns.
0 0 0 0	
0 0 0 0	
0 0 0 0	
0 0 0 0	
0 0 0 0	
0 0 0 0	
0.000E+00	! Data value (for example, MCKT) for site at row 1, column 4
0.000E+00	! Data value for site at row 1, column 5
0.000E+00	! Data value for site at row 1, column 6
0.000E+00	! Data value for site at row 2, column 2
0.990E+01	! Data value for site at row 2, column 3
0.940E+01	
0.930E+01	
0.960E+01	
0.100E+02	
0.103E+02	
0.000E+00	
0.105E+02	
0.930E+01	
0.970E+01	
0.950E+01	
0.930E+01	
0.940E+01	
0.980E+01	
0.112E+02	
0.000E+00	
0.900E+01	
0.930E+01	
0.990E+01	
0.104E+02	
0.109E+02	
0.113E+02	! Data value for site at row 4, column 8
0.000E+00	! Data value for site at row 5, column 4
0.990E+01	! Data value for site at row 5, column 5
0.112E+02	! Data value for site at row 5, column 6

*Text beginning with an exclamation point is explanatory and not part of the actual file. The original file had nine rows of sites. In the interest of brevity the table shows an example having five rows and nine columns as shown in figure 3.

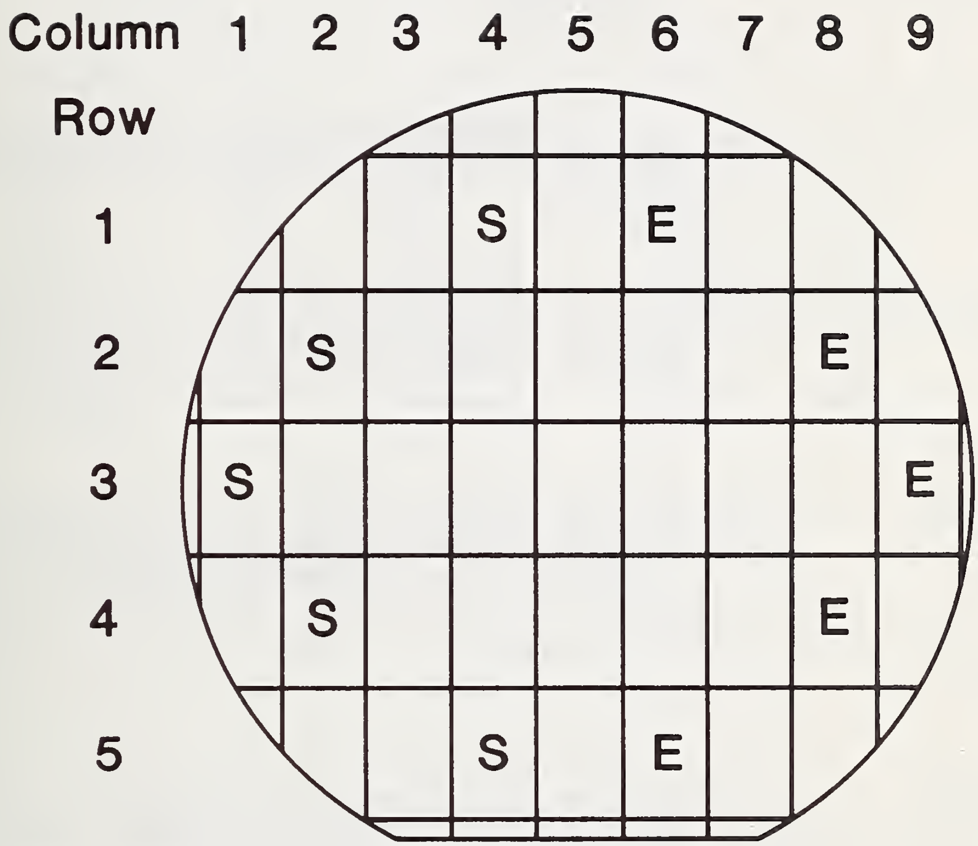


Figure 3. Schematic grid for STAT2 nomenclature regarding site locations.

Table 11. Example of a STAT2 Command Sequence Used to Build a Database and Calculate Correlation Coefficients*

```
ASG W.DAT              ! Process the file W.DAT as explained in the text. W.DAT is
REA 1 0                !       of the type described in table 10. This becomes
XLT 1.0E-12            !       entry number 2 in the database.
PRS
AXP
WDB,1,1,1,1,2
ASG L.DAT              ! Process the file L.DAT in a similar manner to produce
REA 1 0                !       database entry number 3.
XLT 1.0E-12
PRS
AXP
WDB,1,1,1,1,3
ASG MCKT.DAT           ! Process the file MCKT.DAT. This contains the measured
REA 1 0                !       frequency information and becomes database entry
XLT 1.0E-12            !       number 4.
PRS
AXP
WDB,1,1,1,1,4
ASG SCKT.DAT           ! Process the file SCKT.DAT. This contains the simulated
REA 1 0                !       frequency information and becomes database entry 5.
XLT 1.0E-12
PRS
AXP
WDB,1,1,1,1,5
ASG PDIFF.DAT          ! Process the file PDIFF.DAT. This contains the percent
REA 1 0                !       differences between the measured and simulated
XLT 1.0E-12            !       frequencies and becomes database entry 6.
PRS
AXP
WDB,1,1,1,1,6
...
...
...
ASG PRMS2.DAT          ! Continue processing data files in a similar manner, concluding
REA 1 0                !       with the file PRMS2.DAT which becomes database
XLT 1.0E-12            !       entry number 44.
PRS
AXP
WDB,1,1,1,1,44
GET 4                  ! Calculate the sample correlation coefficients for database
SDB ALL COEF 0.0       !       entries 4 (the measured frequency), 5 (the
GET 5                  !       simulated frequency), and 6 (the percent
SDB ALL COEF 0.0       !       difference), against all other entries in
GET 6                  !       the database.
SDB ALL COEF 0.0
END
```

*Text beginning with an exclamation point is explanatory and not part of the actual file.

Table 12. A Rank-Ordering of the Magnitudes of the Ten Highest Correlated Parameters with Respect to (a) the Percent Difference Between the Measured and Simulated Ring Oscillator Frequencies, (b) the Measured Frequency, and (c) the Simulated Frequency. An *n*- or *p*-channel device designation appears before most of the parameters to distinguish between the device types.

	(a) The Percent Difference		(b) The Measured Frequency		(c) The Simulated Frequency	
	Parameter*	Corr. Coef.	Parameter**	Corr. Coef.	Parameter	Corr. Coef.
1	PDIFF	1.000	MCKT	1.000	SCKT	1.000
2	NKP	0.915	SCKT	0.745	NKP	0.979
3	NUO	0.909	NKP	0.660	NUO	0.972
4	SCKT	0.886	NUO	0.659	PDIFF	0.886
5	PKP	0.809	NKAPPA	0.618	PKP	0.877
6	NLD	0.730	PKP	0.616	MCKT	0.745
7	NCGBO	0.706	PTOX	0.511	NSUB	0.713
8	NSUB	0.701	NSUB	0.451	NLD	0.698
9	NCGDO	0.698	NGAMMA	0.404	NCGBO	0.678
10	NGAMMA	0.672	NLD	0.385	NCGDO	0.670

*Remodeling these parameters can change the percent difference between the measured and simulated ring oscillator frequencies.

**Wafer maps of these parameters may suggest strategies to improve process control.

Table 13. Example of a STAT2 Command Sequence Used to Make a Gray-Tone Map

```
ASG MCKT.DAT
REA 1 0
PRS
XLT 1.0E-12
AXP
PRS
MP4,8,1,1,0,0,0
MEASURED FREQUENCY (MHz) – LOT YY027 WAFER Y
END
```

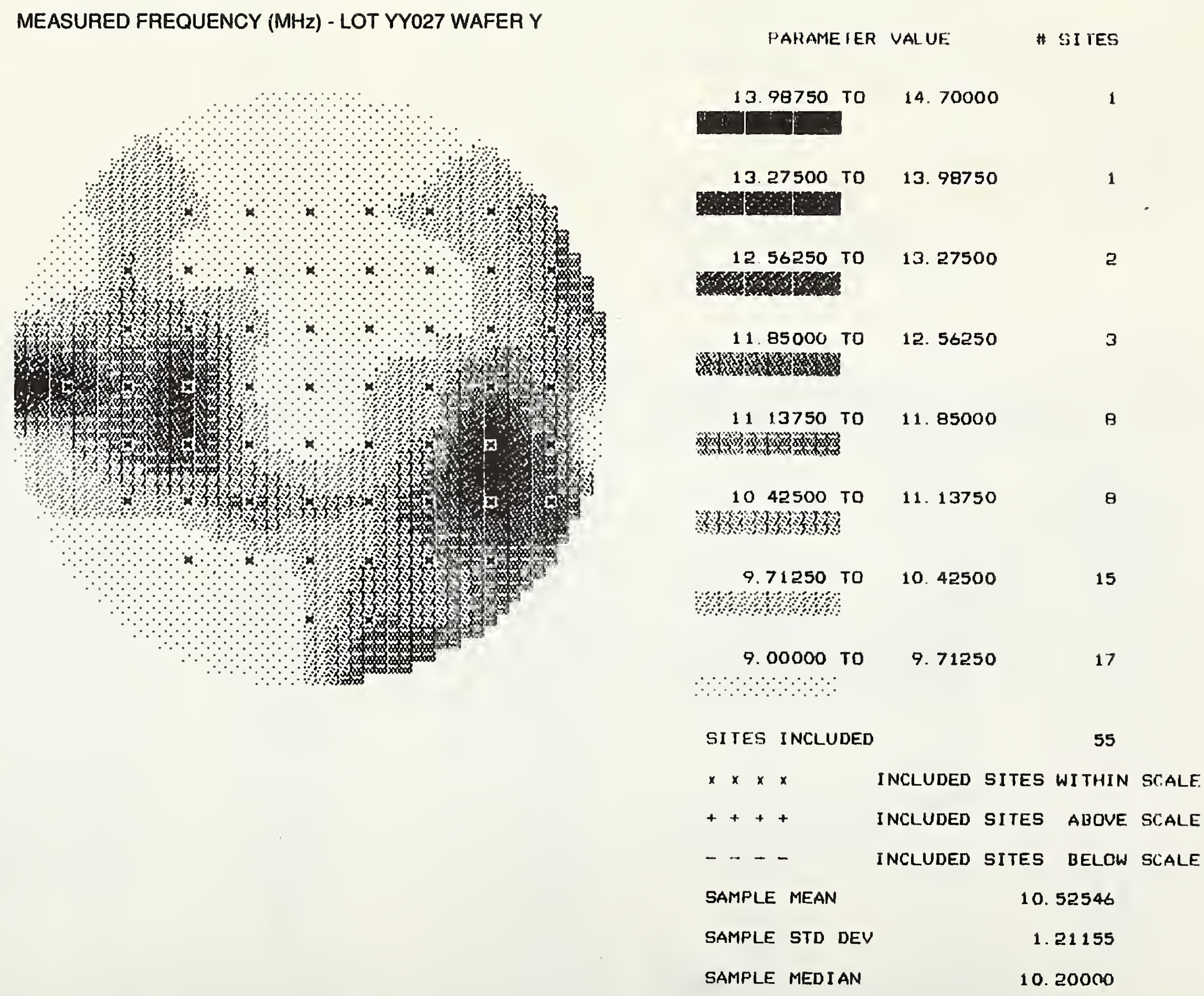


Figure 4. Wafer map produced by STAT2 of the measured ring oscillator frequencies (MCKT).

3. AN EXAMPLE USING CMOS TECHNOLOGY

In this part of this report, an example of the application of KEYS is presented. In particular, a twin-tub, *n*-substrate, CMOS bulk process using four-micrometer design rules was used in this example to manufacture 65 test chips on a 4-in. (10.16-cm) wafer. A 19-stage ring oscillator was selected as the demonstration circuit. MOSFETs, both *n*- and *p*-channel, with the same dimensions as the ring oscillator MOSFETs, were used to measure the I-V data points for SUXES.

This part is divided into two sections. Section 3.1 describes the test structure measurement methods and section 3.2 discusses the output which consists of plots, tabular listings, and wafer maps.

3.1 Test Structure Measurement Methods

Three types of test structures were probed on each test chip: a ring oscillator, cross-bridge resistors [12,13], and *n*- and *p*-channel MOSFETs with the same dimensions as the ring oscillator MOSFETs. A manual probe station and oscilloscope were used to probe the ring oscillators, and the measurements for the other test structures were taken with an Accutest Model 3000 parametric test system.

The ring oscillator was selected as the demonstration circuit. It is a simple circuit and can be used to compare frequencies among chips. Since the frequency is sensitive to the model parameters, it is also a measure of how "reasonable" the critical parameters are to the user. Critical parameters are defined here to be those parameters which have the greatest effect on the ring oscillator frequency [8]. If the user is not satisfied with the percent difference between the simulated and measured frequencies, then the value of a critical parameter is considered "unreasonable" to the user.

The 19-stage ring oscillator is shown in figure 5. The designed channel lengths of the *n*- and *p*-channel MOSFETs and the designed channel widths of the *n*-channel MOSFETs are 4 μm. The designed channel widths of the *p*-channel MOSFETs are 8 μm. A cross-sectional view of an inverter is shown in figure 6. The *n*-wells are not contacted on the top side of the wafer, creating a small voltage drop (approximately 0.1 V). This is due to the doping differences between the substrate, set at 5.0 V, and the *n*-well. A crude estimate of this voltage drop assumes an abrupt change in the doping concentrations between the *n*-type well and *n*-type substrate. This estimate is:

$$\text{voltage drop} = \left(\frac{\mathrm{kT}}{\mathrm{q}}\right)\left[\ln\left(\frac{\mathrm{NWELL}}{\mathrm{SUB}}\right)\right], \tag{2}$$

where NWELL is the *n*-well doping concentration, and SUB is the *n*-substrate doping concentration. This estimate for the voltage drop decreases the simulated ring oscillator frequency by 0.3%. To minimize interference with the actual signal, low capacitance probes (0.01 pF) are used to probe the output of the ring oscillator. Care was taken to minimize effects that could alter the measured results. A 5-V voltage swing was obtained. Start-up circuitry is not incorporated into the design of this ring oscillator; however, a comparison

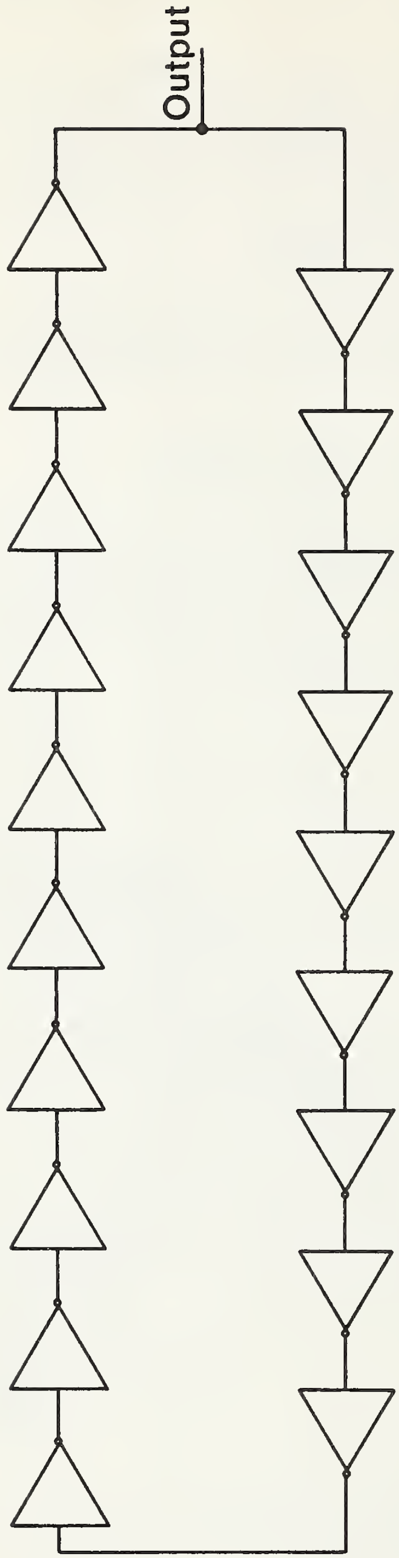


Figure 5. Block diagram of the 19-stage CMOS ring oscillator.

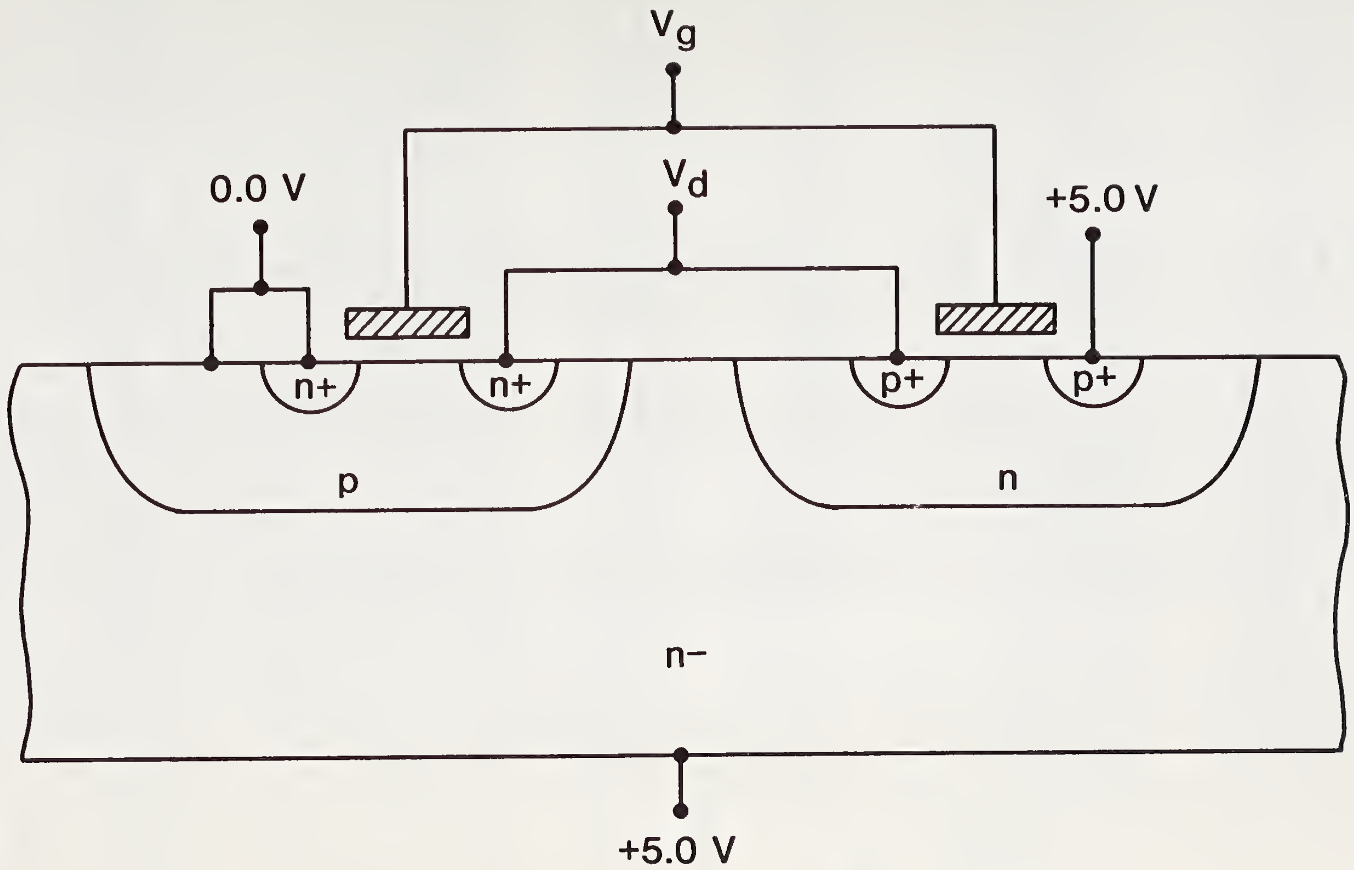


Figure 6. Cross-sectional view of an inverter using a twin-tub, *n*-substrate, CMOS bulk process.

of the measured waveforms with the simulated waveforms reveals that the fundamental oscillation frequency is obtained.

Four-micrometer cross-bridge resistors were probed automatically to determine sheet resistance and linewidth values for n^+ diffusion, poly, and metal. A cross-bridge resistor is shown in figure 7. For details on how to obtain the sheet resistance and linewidth values from these test structures, see references [12,13]. The polysilicon linewidth values obtained from this measurement are representative of the magnitude and changes in the MOSFET channel-lengths used in SPICE.

The third group of test structures are the *n*- and *p*-channel MOSFETs with the same dimensions as the ring oscillator MOSFETs. These devices, of which the *p*-channel MOSFET is shown in figure 8, were used to measure the I-V data points for SUXES and to obtain the initial estimates for the parameters NSUB, VTO, KP, GAMMA, PHI, and UO. These parameters are defined in table 2. The voltage values used to obtain these measurements are given in table 14.

3.2 The Output

KEYS can produce output in the form of plots, tabular listings, and wafer maps. Each form is discussed in the subsections that follow.

3.2.1 The Output Plots

For each chip, plots are made for the *n*- and *p*-channel MOSFET I-V characteristics and a simulated ring oscillator node potential as a function of time. These plots verify that the parameter values are satisfactory.

The I-V characteristics include three separate sets of curves: the measured I-V data, SUXES's predicted I-V data, and SPICE's simulated I-V data. Any deviations will be visually apparent to the circuit designer. A plot for a *p*-channel MOSFET is shown in figure 9. The open circles refer to the measured I-V data points. SUXES predicts the I-V data points using its optimized parameters in SPICE, and the symbol for these data is a filled circle. Finally, a solid line refers to the simulated I-V data points obtained from SPICE using the models file discussed in section 2.2.

For the whole wafer, good agreement is seen among the three curves for the *p*-channel MOSFETs. This is also the case for the *n*-channel MOSFETs. For the wafer, the average percent deviation, for all the acceptable chips, between the measured I-V data points and SUXES's predicted I-V data points for the *p*-channel MOSFETs is 3.2%. This value is 3.4% for the *n*-channel MOSFETs. An acceptable chip is one whose optimized SUXES values are not pegged at the lower or upper bounds.

In KEYS, SPICE is also used to simulate the response of an optional dynamic circuit. The measured waveforms are compared with the SPICE simulated waveforms. Figure 10 displays a simulated ring oscillator waveform. The frequency and period are also recorded on these plots. The ring oscillator is oscillating in its fundamental mode [14] and the node

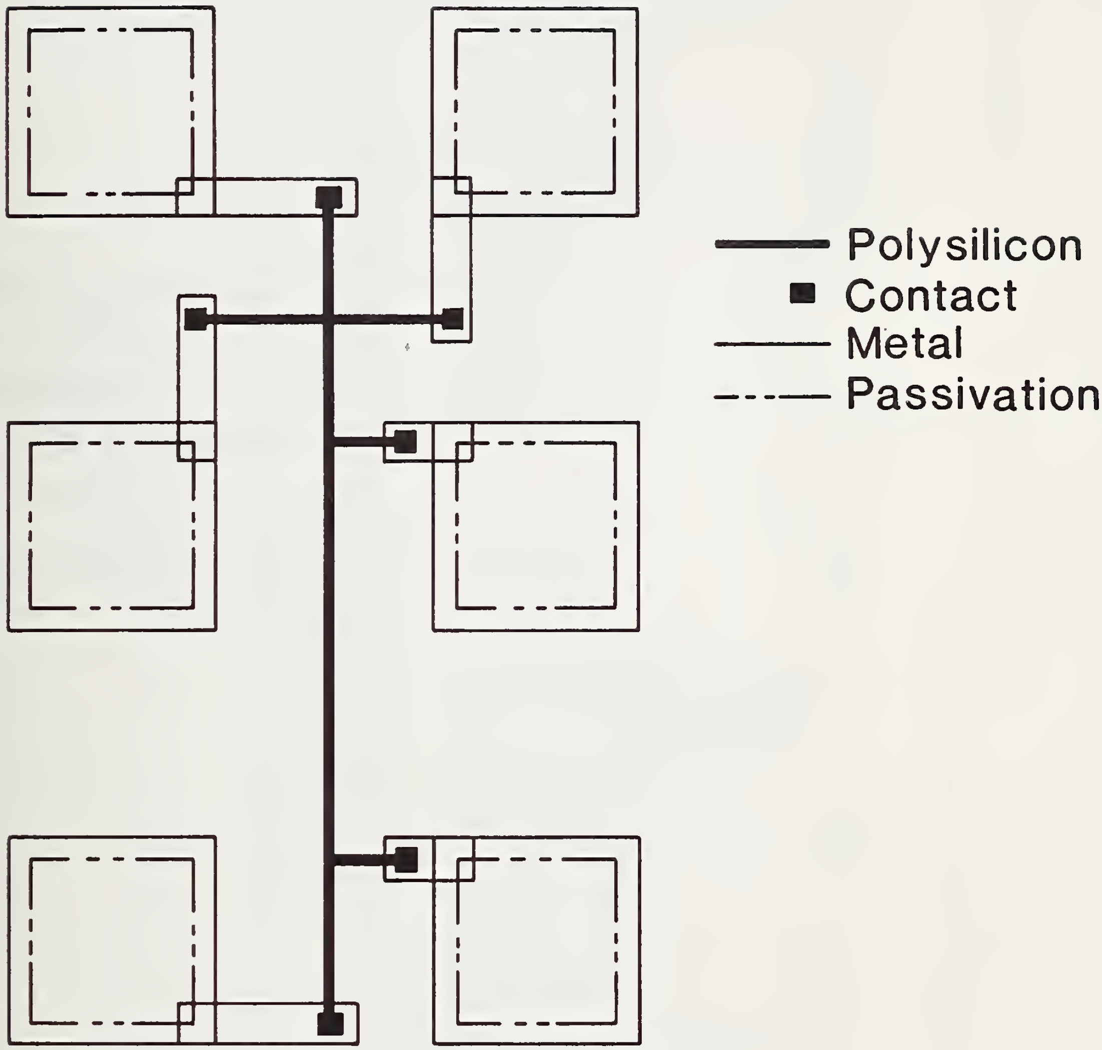


Figure 7. Four-μm cross-bridge resistor for determining the polysilicon linewidth.

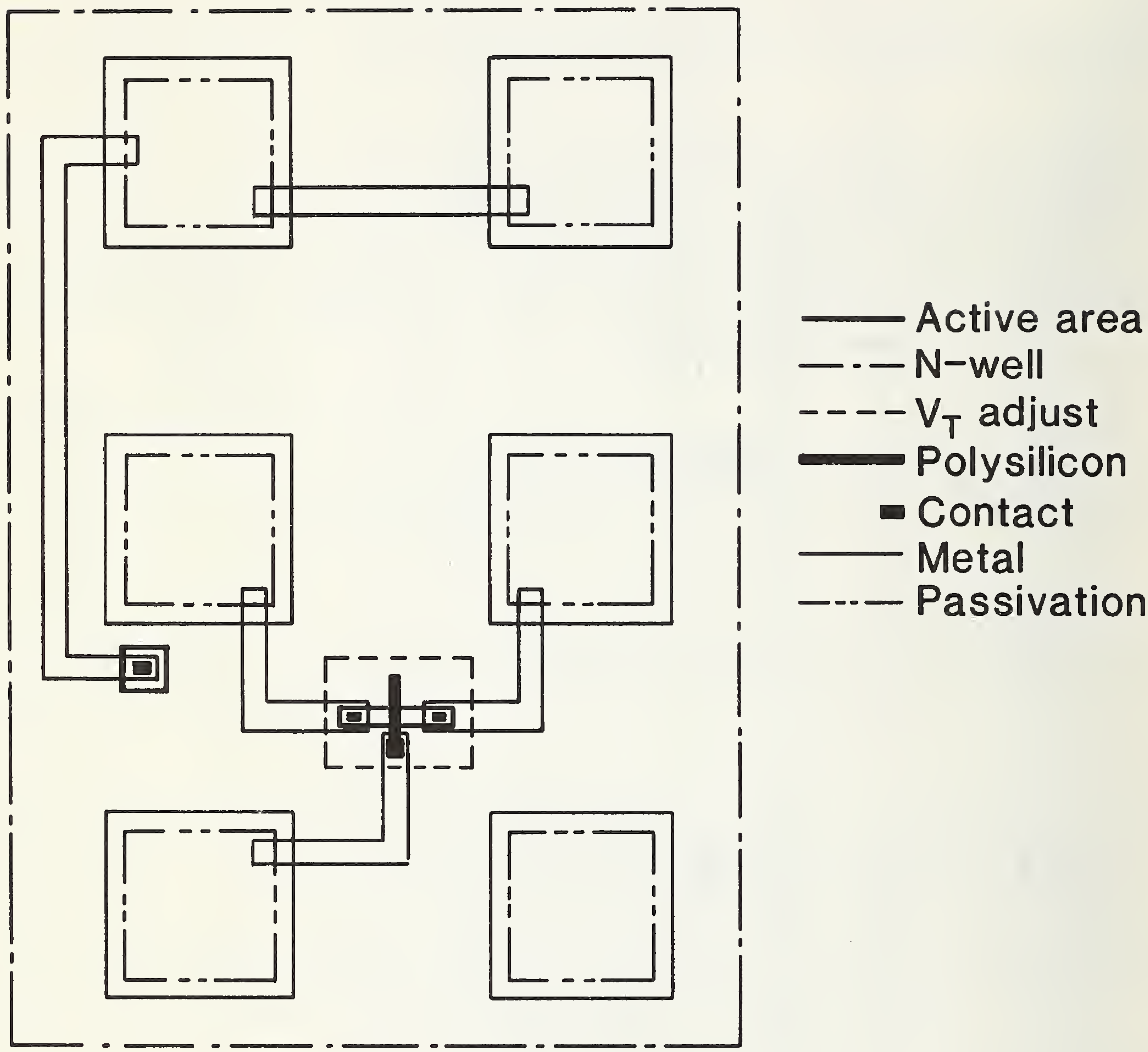


Figure 8. *P*-channel MOSFET of the same dimensions as the *p*-channel ring oscillator MOSFETs.

Table 14. The Voltage Values Used to Measure the *N*- and *P*-Channel MOSFETS

	I-V Data Point Measurements		Measurements Needed for the Initial Estimates	
	n-channel	*p*-channel	*n*-channel	*p*-channel
Device dimensions	L_n=4 μm W_n=4 μm	L_p=4 μm W_p=8 μm	L_n=4 μm W_n=4 μm	L_p=4 μm W_p=8 μm
V_{bs}	0.0 V	0.0 V	0,−3,−6,−9 V	0,3,6,9 V
V_{gs}	0.1,0.2,0.3, 0.5,1,1.5, 2,2.5,3, 3.5,4,4.5, and 5 V	−0.1,−0.2, −0.3,−0.5, −1,−1.5, −2,−2.5, −3,−3.5, −4,−4.5, and −5 V	not applicable	not applicable
V_{ds}	0.1,0.2,0.3, 0.5,1,1.5, 2,2.5,3, 3.5,4,4.5, and 5 V	−0.1,−0.2, −0.3,−0.5, −1,−1.5, −2,−2.5, −3,−3.5, −4,−4.5, and −5 V	0.2 V	−0.2 V
V_{chuck}	0.0 V	0.0 V	0.0 V	V_{bs}

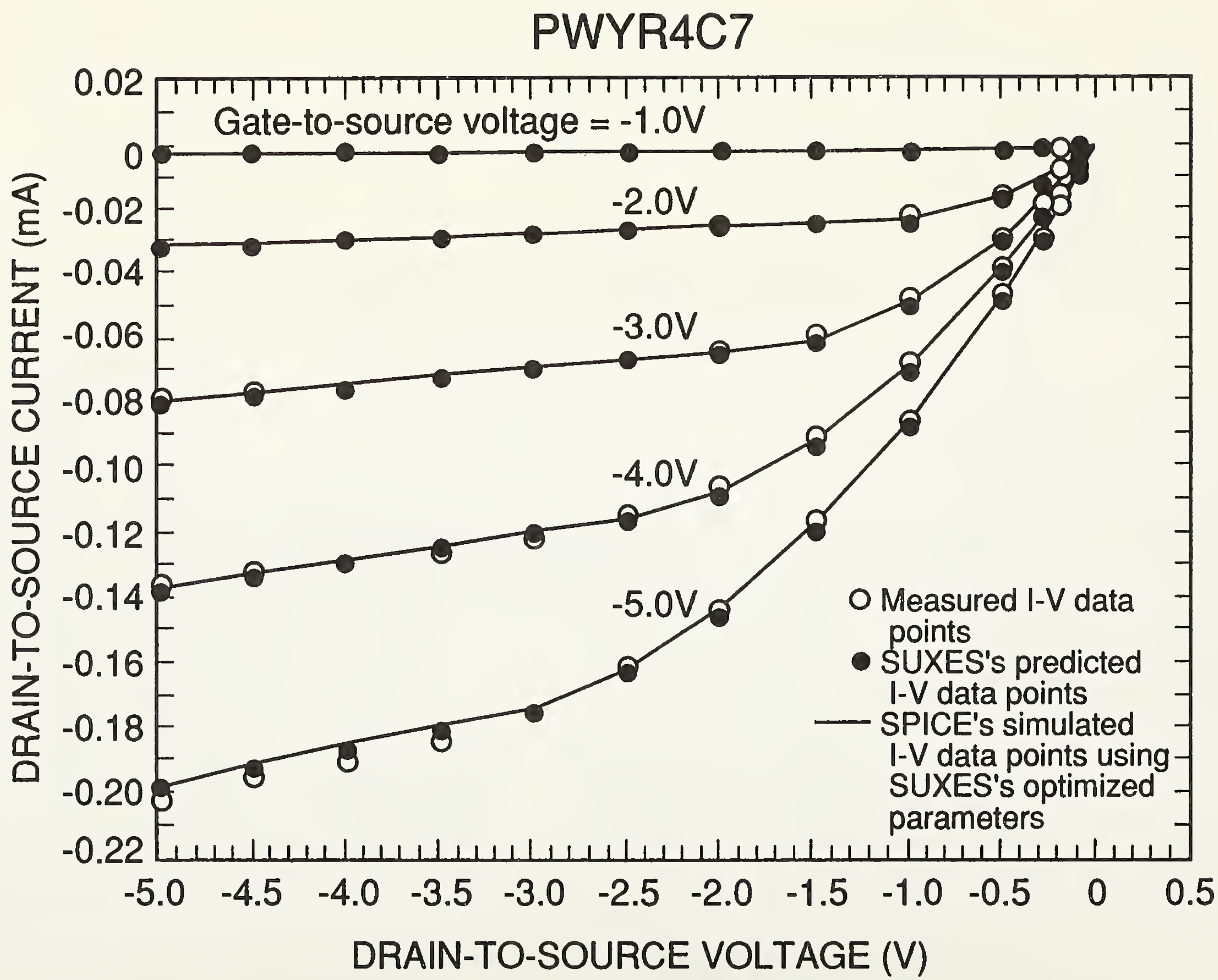


Figure 9. Plot of the I-V characteristics for a *p*-channel MOSFET on the test chip located on wafer Y at row 4 column 7.

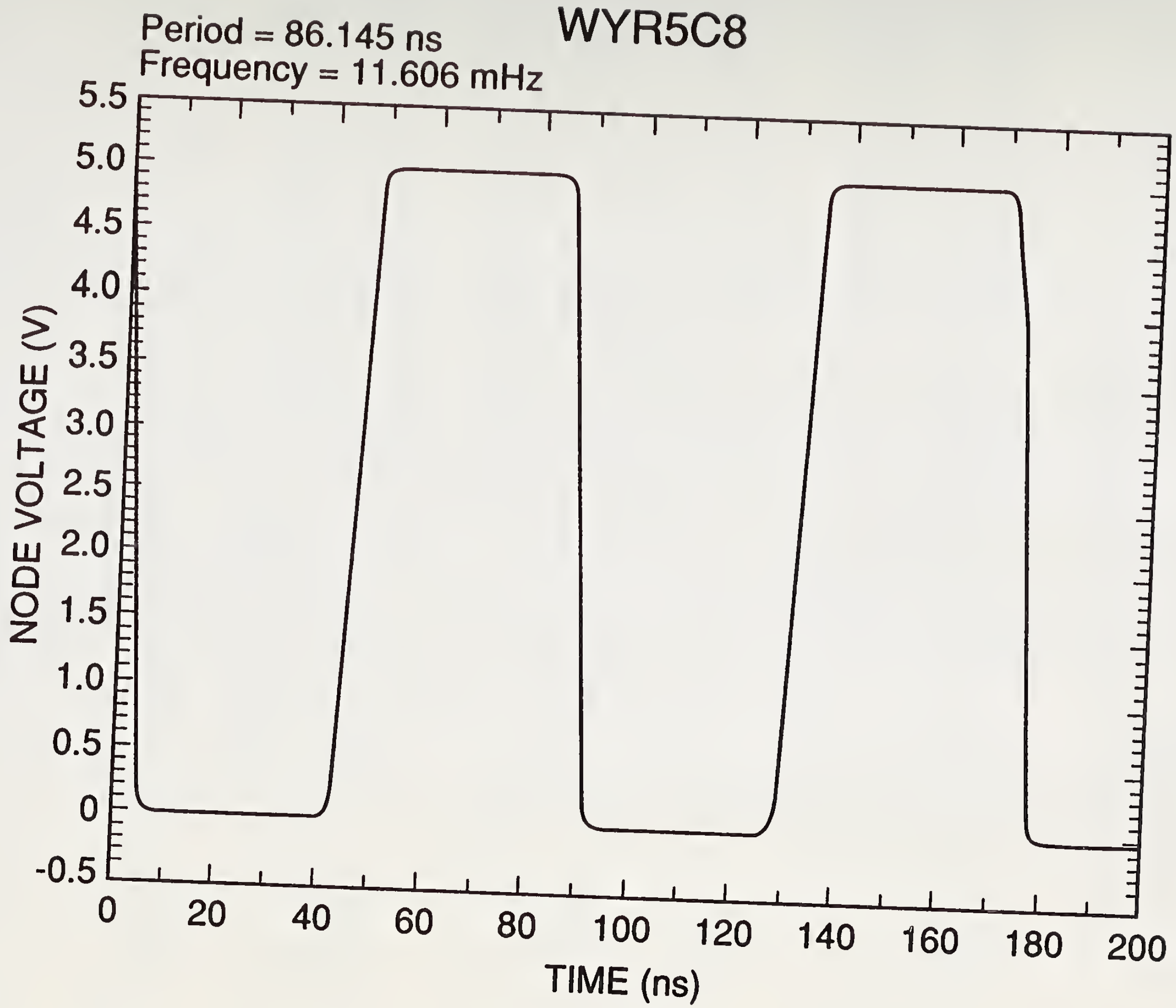


Figure 10. Plot of a simulated node potential in the 19-stage CMOS ring oscillator as a function of time for the ring oscillator on the test chip located on wafer Y at row 5 column 8.

voltages undergo a 5-V voltage swing. For the wafer, the average percent difference, for all the acceptable chips, between the simulated and measured ring oscillator frequencies was 8.6%.

The titles for figures 9 and 10 indicate the wafer and the chip location on the wafer. For the I-V characteristics, an *n*- or *p*-channel device designation is included. For example, PWYR4C7 is a plot (fig. 9) of the I-V characteristics for a *p*-channel MOSFET on the chip located at row 4 column 7 on wafer Y, and WYR5C8 is a plot (fig. 10) of a ring oscillator node potential on the chip located at row 5 column 8 on wafer Y.

3.2.2 The Tabular Listings

KEYS generates two types of tabular listings: a comprehensive summary and rank-orderings of the correlation coefficients. The comprehensive summary is shown in table 15. This table provides the measurement results, the optimized values of SUXES's parameters, the simulation results, the calculated parameters for each chip, the wafer averages, and the lot averages when appropriate. The comprehensive summary in combination with the plots verifies that the I-V data, SUXES, and SPICE are working properly.

Preceding the results from each chip, a "+" or "−" sign appears. A chip's results are either accepted or rejected. If the values for the *n*- and *p*-channel MOSFETs lie between specified lower and upper bounds in SUXES's initial estimates file, it is considered an "acceptable" chip as defined by the user. Preceding the results of an "acceptable" chip, a "+" sign appears. If a parameter's value is pegged at one of these boundaries, then preceeding the results for that chip, a "−" sign appears. Assuming the I-V data look reasonable to the user, a rejected chip can become an accepted chip by modifying the strategy file. This is described in Appendix B. Chips with unreasonable linewidth and frequency measurements as determined by the user are removed from the database.

The wafer averages follow the results from all the chips. The number of acceptable chips out of the total number of chips is recorded. The format of the wafer averages is similar to the format for a single chip. The values are the averages of the corresponding values from the acceptable chips. Throughout this table, the *n*-channel MOSFET information is listed before the *p*-channel MOSFET information.

The lot averages, when appropriate, follow the results from the last wafer. The total number of wafers is recorded. The format for the lot averages is similar to that of the wafer averages. The values are the corresponding average values from the wafer averages.

For brevity, table 15 provides the results from five chips as opposed to all the chips. Hence, in the complete listing, the average values would correspond to the wafer averages. For the entire wafer, the average percent difference between the simulated and measured ring oscillator frequencies was 8.6%. Different strategy files can change this average percent difference.

The second type of tabular listing is a rank-ordering of the spatial correlation coefficients with respect to chosen parameters. The rank-orderings for this example are given in

Table 15. A Tabular Summary of Each Chip's Results Similar to the Wafer Summary. For brevity, this listing was made for five chips. The n-channel MOSFET information is listed before the p-channel MOSFET information.

#									
*	FILENAME = RESULTS.DAT								
*									
*	Lot Number = YY027								
*									
*	Wl	R	C	Wn	Lp	Wp	Lp		
*				MCKT	SCKT	PDIFF			
*									
*				CGDO	CGSO	CGBO	RMS		
*				LD	TOX	NSUB	VTO	KP	
*				GAMMA	PHI	UO	DELTA	VMAX	
*				XJ	KAPPA	NFS	ETA	THETA	ERR
+									
	Y	2	7	2.45	2.45	6.45	2.45		
				10.00	10.30	3.00			
				0.159×10^{-9}	0.159×10^{-9}	0.795×10^{-9}	0.005		
				0.401	868.2	0.122×10^{16}	0.812	27.50	
				0.380	0.650	504.5	1.618	0.750×10^{6}	
				0.415	26.7	0.239×10^{13}	0.0985	0.171	0.056
				0.291×10^{-9}	0.291×10^{-9}	0.129×10^{-8}	0.007		
				0.449	533.2	0.963×10^{15}	−0.788	10.34	
				0.234	0.836	224.2	3.004	0.291×10^{6}	
				0.793	22.8	0.369×10^{12}	0.0842	0.160	0.047
+									
	Y	3	2	2.18	2.18	6.18	2.18		
				10.50	10.56	0.57			
				0.254×10^{-9}	0.254×10^{-9}	0.125×10^{-8}	0.008		
				0.405	550.6	0.167×10^{16}	0.771	26.18	
				0.432	0.619	520.1	1.709	0.789×10^{6}	
				0.475	14.7	0.228×10^{13}	0.0646	0.186	0.066
				0.271×10^{-9}	0.271×10^{-9}	0.120×10^{-8}	0.001		
				0.450	573.3	0.142×10^{16}	−0.758	9.34	
				0.279	0.871	225.5	3.472	0.443×10^{6}	
				0.811	33.4	0.390×10^{12}	0.0558	0.184	0.054
+									
	Y	3	4	2.56	2.56	6.56	2.56		
				9.70	10.17	4.85			
				0.253×10^{-9}	0.253×10^{-9}	0.124×10^{-8}	0.007		
				0.409	556.9	0.186×10^{16}	0.718	26.78	
				0.457	0.625	520.5	1.706	0.783×10^{6}	
				0.475	20.0	0.228×10^{13}	0.1266	0.148	0.052
				0.278×10^{-9}	0.278×10^{-9}	0.125×10^{-8}	0.009		
				0.446	552.7	0.650×10^{15}	−0.730	9.96	

		0.167	0.739	204.5	3.280	0.298×10^{6}	
		0.880	28.3	0.386×10^{12}	0.0942	0.149	0.042
+							
	Y 3 5	2.55	2.55	6.55	2.55		
		9.50	9.95	4.74			
		0.257×10^{-9}	0.257×10^{-9}	0.125×10^{-8}	0.007		
		0.411	550.7	0.161×10^{16}	0.721	25.64	
		0.425	0.619	499.2	1.678	0.786×10^{6}	
		0.475	20.8	0.228×10^{13}	0.1158	0.148	0.053
		0.273×10^{-9}	0.273×10^{-9}	0.123×10^{-8}	0.008		
		0.444	560.5	0.448×10^{15}	−0.698	9.63	
		0.110	0.708	194.5	3.520	0.308×10^{6}	
		1.030	32.2	0.417×10^{12}	0.0903	0.147	0.040
−							
	Y 3 6	2.75	2.75	6.75	2.75		
		9.30	12.22	31.40			
		0.987×10^{-10}	0.987×10^{-10}	0.690×10^{-9}	0.006		
		0.286	1000.0	0.100×10^{14}	1.344	58.00	
		0.100	0.621	878.6	6.808	0.814×10^{6}	
		0.268	78.9	0.273×10^{13}	0.7921	0.162	0.483
		0.226×10^{-9}	0.226×10^{-9}	0.100×10^{-8}	0.003		
		0.449	686.8	0.420×10^{15}	−0.693	11.41	
		0.100	0.724	242.1	3.017	0.281×10^{6}	
		0.994	22.5	0.246×10^{12}	0.1567	0.134	0.044
*							
*	* * WAFER AVERAGES * * *						
*							
*							
*	Wl	Wn	Ln	Wp	Lp		
*		MCKT	SCKT	PDIFF			
*							
*		CGDO	CGSO	CGBO	RMS		
*		LD	TOX	NSUB	VTO	KP	
*		GAMMA	PHI	UO	DELTA	VMAX	
*		XJ	KAPPA	NFS	ETA	THETA	ERR
	Num chips/Tot chips = 4/ 5						
	Y	2.43	2.43	6.44	2.43		
		9.93	10.24	3.29			
		0.231×10^{-9}	0.231×10^{-9}	0.113×10^{-8}	0.007		
		0.407	631.6	0.159×10^{16}	0.756	26.52	
		0.424	0.628	511.1	1.678	0.777×10^{6}	
		0.460	20.6	0.231×10^{13}	0.1014	0.163	0.057
		0.278×10^{-9}	0.278×10^{-9}	0.124×10^{-8}	0.006		
		0.447	554.9	0.870×10^{15}	−0.743	9.82	
		0.198	0.789	212.2	3.319	0.335×10^{6}	
		0.878	29.2	0.391×10^{12}	0.0811	0.160	0.046

table 12. These listings are with respect to the following three chosen parameters: the percent difference between the measured and simulated ring oscillator frequencies (PDIFF), the measured ring oscillator frequency (MCKT), and the simulated ring oscillator frequency (SCKT). In these tables, an *n*- or *p*-channel device designation appears before most of the parameters to distinguish between the device types. The exceptions are the parameters which do not require a designation. They are: PDIFF, MCKT, and SCKT.

The parameters at the top of the rank-ordering in table 12(a), if reoptimized, can change PDIFF. To reoptimize the parameters, the strategy file may be altered according to the procedure given in Appendix B. Remodeling these same parameters can also change PDIFF. This could have the effect of decreasing the rms deviation between the measured and predicted I-V data points, and it could increase the number of acceptable chips.

Table 12(b) lists the correlation coefficient of each parameter with respect to MCKT. Given a minimum number of acceptable chips (as predetermined by the user), a wafer map of MCKT and wafer maps of the highly correlated parameters with respect to MCKT may suggest strategies to improve process control. An example of a wafer map is given in figure 4 for MCKT. This is the third form of output.

3.2.3 The Wafer Maps

Wafer maps may be generated from the results of the correlation coefficient rank-orderings. If a minimum number of acceptable chips (as predetermined by the user) are obtained, the wafer maps are generated by STAT2. In this example, wafer maps are made of the following parameters: (1) MCKT; (2) SCKT; (3) PDIFF; (4) the two highest correlated parameters with respect to PDIFF; excluding the maps already generated; and (5) the two highest correlated parameters with respect to MCKT, excluding the maps already generated. An example of a wafer map is given in figure 4 for the parameter MCKT. This map along with wafer maps of the highly correlated parameters with respect to MCKT may suggest strategies to improve process control.

The user determines the minimum number of acceptable chips. If this value is too low, meaningless wafer maps can be generated. Altering the strategy file (using the procedure in Appendix B) for the parameters at the top of table 12(a) may increase the number of acceptable chips. Other techniques may be just as viable. Meaningful wafer maps are an invaluable tool in evaluating and improving the process. Distinctive patterns may exist which may be caused by limitations in the processing equipment or the manner in which such equipment is used to handle wafers. For example, the doping is at a minimum at the center of the wafer when it is placed in the middle of a fully loaded, slotted quartz diffusion boat. In this case, the flow of gas is maintained parallel to the face of the wafer. When this flow is not parallel to the face of the wafer, other patterns may result. One example is two doping minima [15].

4. CONCLUSIONS

This report describes a computer procedure called KEYS that can be used to characterize a chip, wafer, or lot. KEYS links SUXES, SPICE, and STAT2 according to the flowchart in figure 2. The software used in KEYS is given in table 1.

The benefits of this procedure are as follows: (1) the technique of running SUXES and SPICE becomes standardized; (2) a considerable amount of time is saved; (3) manual intervention and error are reduced; (4) the procedure can be modified for different dynamic circuits; (5) the positional information and the comprehensive output are valuable to process engineers, modelers, and designers; and (6) modifying KEYS for updated versions of SUXES, SPICE, and STAT2 is relatively easy as long as the authors of these programs do not modify their input-output formats.

An example using CMOS technology is also given. Some of the limitations of this procedure for the example are as follows: (1) the program aborts if the I-V data are not characteristic of a typical MOSFET; (2) the boundary values in SUXES are arbitrary; (3) the user determines the minimum number of acceptable chips, and meaningless wafer maps are an indication this value is too low; (4) considerable effort is required to modify the strategy file, according to the procedure given in Appendix B, to increase the number of acceptable chips; and (5) the user helps to determine, via the strategy file, how closely the simulated results match the measured results.

The output of this software addresses problems with simulating, modeling, and processing. With regard to the simulations, the wafer summary, a portion of which is shown in table 15, determines which chips are "acceptable." The I-V characteristics and the ring oscillator plots of these chips verify that the parameter values are realistic. For the unacceptable chips, this information along with the parameter values help determine if the I-V data, SUXES, or SPICE need reexamination. Also, this information helps determine if it is worthwhile modifying the strategy file to make these chips acceptable. With regard to modeling efforts, the parameters at the top of table 12(a), if remodeled properly, would be responsible for decreasing the percent difference between the measured and simulated ring oscillator frequencies. These same parameters, if reoptimized, may be responsible for increasing the number of acceptable chips. With regard to the processing efforts, wafer maps reveal distinctive patterns of nonuniformity in semiconductor processes. A wafer map of the measured frequency and wafer maps of the parameters at the top of table 12(b) may assist to develop processes which minimize nonuniformity. For example, ring oscillator frequency variations may be minimized by providing tighter controls on the polysilicon linewidth variations, which may be due to variations in film thickness [15].

5. ACKNOWLEDGMENTS

Many thanks go to J. A. Horst who measured the threshold voltages and I-V data points, developed the Accutest computer software to obtain these measurements, and transferred these data to the VAX. We appreciate the software made available to us by M. Gaitan. For comments on this report, we thank H. S. Bennett, A. R. Hefner, and L. W. Linholm. The editorial assistance of J. A. Gonzalez was invaluable. The help or support of all others involved, including J. Albers and C. L. Wilson, is appreciated.

6. REFERENCES

[1] Doganis, K., and Dutton, R. W., SUXES: Stanford University Extractor of Model Parameters (Users Manual), U.S. Army Research Office Contract No. DAAG-29-80-K-C-0013 and DARPA Contract No. MDA903-79-C-0257, Integrated Circuits Laboratory, Stanford University, Stanford, CA (November 1982).

[2] Nagel, L. W., SPICE2: A Computer Program to Simulate Semiconductor Circuits, Memorandum No. ERL-M520, Electronics Research Laboratory, University of California, Berkeley, CA (May 1975).

[3] Mattis, R. L., *Semiconductor Measurement Technology*: A FORTRAN Program for Analysis of Data from Microelectronic Test Structures, NBS Special Publication 400-75 (July 1983); Mattis, R. L., and Zucker, R., Release Notes for STAT2 Version 1.31: An Addendum to NBS Special Publication 400-75, NBSIR 83-2779 (November 1983); Mattis, R. L., Release Notes for STAT2 Version 1.7: An Addendum to NBS Special Publication 400-75, NBSIR 86-3333 (March 1986).

[4] Nye, B., Sangiovanni-Vincentelli, A., Spoto, J., and Tits, A., DELIGHT.SPICE: An Optimization-Based System for the Design of Integrated Circuits, Proc. 1983 IEEE Custom Integrated Circuits Conf., pp. 233-238, May 1983.

[5] Scharfetter, D. L., Tremain Jr., R. E., Oki, T. J., Doganis, K., and Chen, S., An Integrated IC Process Characterization Facility, *IEDM Tech. Dig.*, 246-249 (December 1983).

[6] Vladimirescu, A., and Liu, S., The Simulation of MOS Integrated Circuits Using SPICE2, Memorandum No. ERL-M80/7, Electronics Research Laboratory, University of California, Berkeley, CA (February 1980).

[7] Alexander, D. R., Antinone, R. J., and Brown, G. W., SPICE2 MOS Modeling Handbook, BDM/A-77-071-TR, The BDM Corporation, Albuquerque, NM (May 1977).

[8] Cassard, J. M., A Sensitivity Analysis of SPICE Parameters Using an Eleven-Stage Ring Oscillator, *IEEE Trans. Electron Devices* ED – 31 (2), 264-269 (1984) and *IEEE J. Solid State Circuits* SC – 19 (1), 130-135 (February 1984).

[9] Unit Software & Consulting Inc., SPICE2 Error Messages and Explanations, (November 1982).

[10] Snedecor, G. W., and Cochran, W. G., Statistical Methods, (The Iowa State University Press, Ames, IA, 1967), pp. 172-198.

[11] Mendenhall, W., Introduction to Probability and Statistics, (Duxbury Press, Belmont, CA, 1971), pp. 276-279.

[12] Buehler, M. G., Grant, S. D., and Thurber, W. R., Bridge and van der Pauw Sheet Resistors for Characterizing the Linewidth of Conducting Layers, *J. Electrochem.*

Soc. 125 (4), 650-654 (April 1978).

[13] Buehler, M. G., and Linholm, L. W., Role of Test Chips in Coordinating Logic and Circuit Design and Layout Aids for VLSI, *Solid State Technology* 24 (9), 68-74 (September 1981).

[14] Sasaki, N., Higher Harmonic Generation in CMOS/SOS Ring Oscillators, *IEEE Trans. Electron Devices* ED – 29 (2), 280-283 (February 1982).

[15] Perloff, D. S., Wahl, F. E., and Reimer, J. D., Contour Maps Reveal Non-Uniformity In Semiconductor Processing, *Solid State Technology* 20 (2), 31-36, 42 (February 1977).

[16] Doganis, K., and Scharfetter, D. L., General Optimization and Extraction of IC Device Model Parameters, *IEEE Trans. Electron Devices* ED – 30 (9), 1219-1228 (September 1983).

Appendix A - The Program Listing for the Computer Procedure KEYS

This appendix contains the computer procedure KEYS. It is ordered by file type. The four different file-types specified are command procedures (with a .COM or .XXX extension), FORTRAN files (with a .FOR extension), data files (with a .FIL, .DAT, or .XXX extension), and editor files (with a .WAY extension). They are listed in this order and the files within each file-type are listed in the order of their occurrence. The command procedures are given first and they are:

(1) KEYS.COM

(2) LAST3.COM

(3) INKEY.XXX

(4) SUXES.XXX

(5) SPICEIV.XXX

(6) RMSKEY.COM

(7) STKEY.COM

(8) CORKEY.XXX

(9) MAPKEY.XXX

The FORTRAN files included in this Appendix are listed second and they are:

(1) RCKEY.FOR

(2) SUKEY.FOR

(3) CAPKEY.FOR

(4) IVKEY.FOR

(5) SUMKEY.FOR

(6) RANKEY.FOR

The data files are listed third and they are:

(1) NLW.FIL

(2) RC.FIL

(3) LINE.FIL

(4) LOTNUMBER.FIL

(5) NPARIN.FIL

(6) PPARIN.FIL

(7) NINPUT.XXX

(8) PINPUT.XXX

(9) NOPT.FIL

(10) POPT.FIL

(11) NSTRAT.FIL

(12) PSTRAT.FIL

(13) IVCHARN.XXX

(14) IVCHARP.XXX

(15) NMODEL.XXX

(16) PMODEL.XXX

(17) SUBCKT.XXX

(18) CIRCUIT.FIL

(19) PCHIPS.FIL

(20) STEND.FIL

(21) COREF.FIL

The editor files that are listed fourth are:

(1) EDNO.WAY

(2) EDSUXN.WAY

(3) EDSUXP.WAY

(4) EDPAR.WAY

(5) EDSPICEN.WAY

(6) EDSPICEP.WAY

(7) EDLAST.WAY

(8) EDNRMS.WAY

(9) EDPRMS.WAY

(10) EDRES.WAY

(11) EDSKEY.WAY

(12) EDMAP.WAY

(13) EDSIMF.WAY

(14) EDPDIFF.WAY

(15) EDMEAF.WAY

(16) EDMAPS.WAY

Refer to table C-4 for a more complete listing of the file names.

KEYS.COM

```
$ !
$ !  To run KEYS type:  SUBMIT/noprinter/notify KEYS
$ !
$ !  FILENAME = KEYS.COM
$ !
$ !
$ ! CURDIR == F$ENVIRONMENT("DEFAULT")
$ SET DEFAULT 'KEYSDIR'
$ !
$ copy nlw.fil in.dat
$ !
$ edit/edt/command=edno.way/output=blank.fil in.dat
$ copy blank.fil switchl.dat
$ copy blank.fil plw.dat
$ copy blank.fil nguest.dat
$ copy blank.fil pguest.dat
$ copy blank.fil nparspice.dat
$ copy blank.fil pparspice.dat
$ copy blank.fil nlwl.dat
$ copy blank.fil plwl.dat
$ copy blank.fil mapl.dat
$ !
$ !  Determine which chip to consider
$ !
$ run RCKEY
$ !
$ !
$ !  If the last chip is being considered, set the switch in RMSKEY.COM
$ !
$ copy/concat blank.fil,rmskey.com,switchl.dat in.yyy
$ !
$ edit/edt/command=edlast.way/output=rmskey.com in.yyy
$ !
$ !  Modify and run the command procedure INKEY for the appropriate chip
$ !
$ copy/concat blank.fil,inkey.xxx,switchl.dat,nlw.fil,plw.dat in.yyy
$ !
$ @last3 2 in.yyy LLL WWW
$ edit/edt/command=edt3.com/output=out.yyy in.yyy
$ @last3 3 out.yyy DDD BBB AAA
$ edit/edt/command=edt3.com/output=in.yyy out.yyy
$ @last3 2 in.yyy EEE III
$ edit/edt/command=edt3.com/output=out.yyy in.yyy
$ @last3 3 out.yyy QQQ QQQ QQQ
$ edit/edt/command=edt3.com/output=in.yyy out.yyy
$ @last3 3 in.yyy QQQ PPP NNN
$ edit/edt/command=edt3.com/output=inkey.com in.yyy
$ !
$ @INKEY
$ !
```

LAST3.COM

```
$ !
$ ! Procedure LAST3.COM -- Reads the last 1, 2 or 3 records of a file
$ ! and generates an EDT command file which makes substitutions based
$ ! on the contents of those last three records.  P1 contains the
$ ! number 1, 2, or 3 and indicates how many substitute commands are
$ ! to be produced.  P2 contains the name of the file to be edited.
$ ! The output file is always EDT3.COM.  P3 contains the string which
$ ! is to be replaced by the string in the last line.  P4 contains the
$ ! string which is to be replaced by the next to the last line.  P5
$ ! contains the string which is to be replaced by the second to last
$ ! line.
$ !
$ ! On any of the last three lines which are processed, if there are any
$ ! leading spaces, these spaces are removed and the text on that line
$ ! is left justified in the EDT3.COM file.
$ !
$ WS :== "WRITE SYS$OUTPUT "
$ IF (P1.EQ.1) .OR. (P1.EQ.2) .OR. (P1.EQ.3) THEN GOTO L0
$ WS "Argument missing or not in range 1 to 3"
$ EXIT
$ !
$ L0:
$ OPEN/READ INFILE 'P2'
$ !
$ ! Branch depending on how many records are to be examined.
$ !
$ IF P1.EQ.1 THEN GOTO L1
$ IF P1.EQ.2 THEN GOTO L2
$ IF P1.EQ.3 THEN GOTO L3
$ !
$ ! Case 1: Generate EDT3.COM based on just the last record in the file.
$ !
$ L1:
$ READ/END_OF_FILE=TOO_FEW INFILE RECORD1
$ L11:
$ READ/END_OF_FILE=L12 INFILE RECORD1
$ GOTO L11
$ !
$ L12:
$ CLOSE INFILE
$ OPEN/WRITE OUTFILE EDT3.COM
$ !
$ ! Left justify the last record.
$ !
$ LENGTH := 'F$LENGTH(RECORD1)'
$ IF LENGTH .LE. 1 THEN GOTO L14
$ FIRST_SPACE := 'F$LOCATE(" ",RECORD1)'
$ IF FIRST_SPACE .NE. 0 THEN GOTO L14
$ RECORD1 := 'F$EXTRACT(1,LENGTH,RECORD1)'
$ !
```

```
$ L14:
$ WRITE OUTFILE "SUBS /''P3'/''RECORD1'/ 1:END"
$ WRITE OUTFILE "T END-1"
$ WRITE OUTFILE "D .:END"
$ WRITE OUTFILE "EXIT"
$ CLOSE OUTFILE
$ GOTO AFTER_WRITE
$ !
$ ! Case 2: Construct EDT3.COM using the last two records in the file.
$ !
$ L2:
$ NUM=0
$ READ/END_OF_FILE=TOO_FEW INFILE RECORD2
$ NUM=1
$ READ/END_OF_FILE=TOO_FEW INFILE RECORD1
$ NUM=2
$ L21:
$ READ/END_OF_FILE=L22 INFILE RECORD2
$ NUM=1
$ READ/END_OF_FILE=L22 INFILE RECORD1
$ NUM=2
$ GOTO L21
$ !
$ L22:
$ CLOSE INFILE
$ OPEN/WRITE OUTFILE EDT3.COM
$ !
$ ! Left justify the RECORD1.
$ !
$ LENGTH := 'F$LENGTH(RECORD1)'
$ IF LENGTH .LE. 1 THEN GOTO L24
$ FIRST_SPACE := 'F$LOCATE(" ",RECORD1)'
$ IF FIRST_SPACE .NE. 0 THEN GOTO L24
$ RECORD1 := 'F$EXTRACT(1,LENGTH,RECORD1)'
$ !
$ L24:
$ !
$ ! Left justify RECORD2.
$ !
$ LENGTH := 'F$LENGTH(RECORD2)'
$ IF LENGTH .LE. 1 THEN GOTO L25
$ FIRST_SPACE := 'F$LOCATE(" ",RECORD2)'
$ IF FIRST_SPACE .NE. 0 THEN GOTO L25
$ RECORD2 := 'F$EXTRACT(1,LENGTH,RECORD2)'
$ !
$ L25:
$ IF NUM.EQ.1 THEN WRITE OUTFILE "SUBS /''P3'/''RECORD2'/ 1:END"
$ IF NUM.EQ.1 THEN WRITE OUTFILE "SUBS /''P4'/''RECORD1'/ 1:END"
$ IF NUM.EQ.2 THEN WRITE OUTFILE "SUBS /''P3'/''RECORD1'/ 1:END"
$ IF NUM.EQ.2 THEN WRITE OUTFILE "SUBS /''P4'/''RECORD2'/ 1:END"
$ WRITE OUTFILE "T END-2"
$ WRITE OUTFILE "D .:END"
```

```
$ WRITE OUTFILE "EXIT"
$ CLOSE OUTFILE
$ GOTO AFTER_WRITE
$ !
$ L3:
$ !
$ ! Case 3: Construct EDT3.COM using last 3 records of file.
$ !
$ NUM=0
$ READ/END_OF_FILE=TOO_FEW INFILE RECORD3
$ NUM=1
$ READ/END_OF_FILE=TOO_FEW INFILE RECORD2
$ NUM=2
$ READ/END_OF_FILE=TOO_FEW INFILE RECORD1
$ NUM=3
$ L31:
$ READ/END_OF_FILE=L32 INFILE RECORD3
$ NUM=1
$ READ/END_OF_FILE=L32 INFILE RECORD2
$ NUM=2
$ READ/END_OF_FILE=L32 INFILE RECORD1
$ NUM=3
$ GOTO L31
$ !
$ L32:
$ CLOSE INFILE
$ OPEN/WRITE OUTFILE EDT3.COM
$ !
$ ! Left justify RECORD1.
$ !
$ LENGTH := 'F$LENGTH(RECORD1)'
$ IF LENGTH .LE. 1 THEN GOTO L33
$ FIRST_SPACE := 'F$LOCATE(" ",RECORD1)'
$ IF FIRST_SPACE .NE. 0 THEN GOTO L33
$ RECORD1 := 'F$EXTRACT(1,LENGTH,RECORD1)'
$ !
$ L33:
$ !
$ ! Left justify RECORD2.
$ !
$ LENGTH := 'F$LENGTH(RECORD2)'
$ IF LENGTH .LE. 1 THEN GOTO L34
$ FIRST_SPACE := 'F$LOCATE(" ",RECORD2)'
$ IF FIRST_SPACE .NE. 0 THEN GOTO L34
$ RECORD2 := 'F$EXTRACT(1,LENGTH,RECORD2)'
$ !
$ L34:
$ !
$ ! Left justify RECORD3.
$ !
$ LENGTH := 'F$LENGTH(RECORD3)'
$ IF LENGTH .LE. 1 THEN GOTO L35
```

```
$ FIRST_SPACE := 'F$LOCATE(" ",RECORD3)'
$ IF FIRST_SPACE .NE. 0 THEN GOTO L35
$ RECORD3 := 'F$EXTRACT(1,LENGTH,RECORD3)'
$ !
$ L35:
$ IF NUM.EQ.1 THEN WRITE OUTFILE "SUBS /''P3'/''RECORD3'/ 1:END"
$ IF NUM.EQ.1 THEN WRITE OUTFILE "SUBS /''P4'/''RECORD1'/ 1:END"
$ IF NUM.EQ.1 THEN WRITE OUTFILE "SUBS /''P5'/''RECORD2'/ 1:END"
$ IF NUM.EQ.2 THEN WRITE OUTFILE "SUBS /''P3'/''RECORD2'/ 1:END"
$ IF NUM.EQ.2 THEN WRITE OUTFILE "SUBS /''P4'/''RECORD3'/ 1:END"
$ IF NUM.EQ.2 THEN WRITE OUTFILE "SUBS /''P5'/''RECORD1'/ 1:END"
$ IF NUM.EQ.3 THEN WRITE OUTFILE "SUBS /''P3'/''RECORD1'/ 1:END"
$ IF NUM.EQ.3 THEN WRITE OUTFILE "SUBS /''P4'/''RECORD2'/ 1:END"
$ IF NUM.EQ.3 THEN WRITE OUTFILE "SUBS /''P5'/''RECORD3'/ 1:END"
$ WRITE OUTFILE "T END-3"
$ WRITE OUTFILE "D .:END"
$ WRITE OUTFILE "EXIT"
$ CLOSE OUTFILE
$ GOTO AFTER_WRITE
$ !
$ AFTER_WRITE:
$ EXIT
$ !
$ TOO_FEW:
$ WS "Too few records in the file 'P2'"
$ EXIT
```

INKEY.XXX

```
$ !
$ !  FILENAME:  INKEY.COM
$ !
$ !            In the program SUKEY, the width is asked for first then
$ !            the length.
$ !
$ !
$ SET DEFAULT 'KEYSDIR'
$ copy WAAARBBBCDDDVB.DAT vbin.dat
$ !-----------------------------------------------------------------------------
$ !
$ ! Consider the n-channel MOSFETs.
$ !
NNN
$ IF L .EQS. 0 THEN GOTO PCHAN
$ copy nWAAARBBBCDDD.dat in.dat
$ copy nparin.fil in3.dat
$ !
$ ! Gather the initial estimates for SUXES.
$ !
$ run SUKEY
III
EEE
$ copy guest.dat nguest.dat
$ copy out.dat nWAAARBBBCDDDF.dat
$ copy out2.dat nWAAARBBBCDDDS.dat
$ !
$ !  Insert the parameters in the SUXES input file.
$ !
$ copy/concat blank.fil,ninput.xxx,nguest.dat in.yyy
$ !
$ @last3 3 in.yyy VVV VVV VVV
$ edit/edt/command=edt3.com/output=out.yyy in.yyy
$ @last3 3 out.yyy VVV VVV VVV
$ edit/edt/command=edt3.com/output=in.yyy out.yyy
$ @last3 2 in.yyy VVV VVV
$ edit/edt/command=edt3.com/output=out.yyy in.yyy
$ @last3 2 out.yyy HHH VVV
$ edit/edt/command=edt3.com/output=in.yyy out.yyy
$ @last3 2 in.yyy KKK VVV
$ edit/edt/command=edt3.com/output=out.yyy in.yyy
$ @last3 2 out.yyy JJJ VVV
$ edit/edt/command=edt3.com/output=in.yyy out.yyy
$ @last3 2 in.yyy MMM VVV
$ edit/edt/command=edt3.com/output=out.yyy in.yyy
$ @last3 2 out.yyy OOO VVV
$ edit/edt/command=edt3.com/output=in.yyy out.yyy
$ @last3 2 in.yyy RRR VVV
$ edit/edt/command=edt3.com/output=out.yyy in.yyy
$ @last3 2 out.yyy SSS VVV
```

```
$ edit/edt/command=edt3.com/output=in.yyy out.yyy
$ @last3 2 in.yyy TTT VVV
$ edit/edt/command=edt3.com/output=out.yyy in.yyy
$ @last3 2 out.yyy UUU VVV
$ edit/edt/command=edt3.com/output=ninput.dat out.yyy
$ !
$ ! Modify and run the command procedure for SUXES for the appropriate
$ !     chip and device type.
$ !
$ copy/concat blank.fil,suxes.xxx,plw.dat in.yyy
$ !
$ @last3 2 in.yyy VVV VVV
$ edit/edt/command=edt3.com/output=out.yyy in.yyy
$ @last3 3 out.yyy ggg aza ccc
$ edit/edt/command=edt3.com/output=in.yyy out.yyy
$ edit/edt/command=edsuxn.way/output=suxes.com in.yyy
$ !
$ copy ninput.dat input.dat
$ copy nopt.fil opt.dat
$ copy nstrat.fil strat.dat
$ !
$ @SUXES
$ !
$ ! Gather the extracted parameters.
$ !
$ copy expar.dat in.dat
$ !
$ edit/edt/command=edpar.way/output=exparn.dat in.dat
$ !
$ copy exparn.dat expar.dat
$ run CAPKEY
$ !
$ copy parspice.dat nparspice.dat
$ !
$ ! Insert the width and length into the SPICE file which obtains the
$ !     IV characteristics.
$ !
$ copy/concat blank.fil,ivcharn.xxx,nlw.fil in.yyy
$ !
$ @last3 2 in.yyy BZB FFF
$ edit/edt/command=edt3.com/output=out.yyy in.yyy
$ @last3 3 out.yyy VVV VVV VVV
$ edit/edt/command=edt3.com/output=ivchar.fil out.yyy
$ !
$ copy ivchar.fil ivcharn.fil
$ !
$ ! Put together the n-channel MOSFET models file for SPICE.
$ !
$ copy/concat blank.fil,nmodel.xxx,nparspice.dat in.yyy
$ !
$ @last3 3 in.yyy vvv aqa bqb
$ edit/edt/command=edt3.com/output=out.yyy in.yyy
```

```
$ @last3 3 out.yyy cqc dqd eqe
$ edit/edt/command=edt3.com/output=in.yyy out.yyy
$ @last3 3 in.yyy fqf gqg hqh
$ edit/edt/command=edt3.com/output=out.yyy in.yyy
$ @last3 3 out.yyy iqi jqj kqk
$ edit/edt/command=edt3.com/output=in.yyy out.yyy
$ @last3 3 in.yyy lql mqm nqn
$ edit/edt/command=edt3.com/output=out.yyy in.yyy
$ @last3 3 out.yyy oqo pqp rqr
$ edit/edt/command=edt3.com/output=in.yyy out.yyy
$ @last3 1 in.yyy sqs
$ edit/edt/command=edt3.com/output=nmodel.fil in.yyy
$ !
$ COPY/CONCAT BLANK.FIL,NMODEL.FIL,IVCHARN.FIL IVCHARN.CEL
$ copy ivcharn.cel ivchar.cel
$ !
$ ! Modify the command procedure which obtains the IV characteristics.
$ !
$ copy/concat blank.fil,spiceiv.xxx,nlwl.dat in.yyy
$ edit/edt/command=edspicen.way/output=out.yyy in.yyy
$ @last3 2 out.yyy gzg fzf
$ edit/edt/command=edt3.com/output=in.yyy out.yyy
$ @last3 3 in.yyy eze dzd czc
$ edit/edt/command=edt3.com/output=spiceiv.com in.yyy
$ !
$ !  Simulate the n-channel MOSFET IV characteristics.
$ !
$ @SPICEIV
$ !
$ !----------------------------------------------------------------------------
$ !
$ ! Repeat the above for the P-CHANNEL MOSFETs.
$ !
$ PCHAN:
PPP
$ IF M .EQS. 0 THEN GOTO CKT
$ copy pWAAARBBBCDDD.dat in.dat
$ copy pparin.fil in3.dat
$ !
$ ! Gather the initial estimates for SUXES.
$ !
$ run SUKEY
WWW
LLL
$ copy guest.dat pguest.dat
$ copy out.dat pWAAARBBBCDDDF.dat
$ copy out2.dat pWAAARBBBCDDDS.dat
$ !
$ ! Insert the parameters in the SUXES input file.
$ !
$ copy/concat blank.fil,pinput.xxx,pguest.dat in.yyy
$ !
```

```
$ @last3 3 in.yyy VVV VVV VVV
$ edit/edt/command=edt3.com/output=out.yyy in.yyy
$ @last3 3 out.yyy VVV VVV VVV
$ edit/edt/command=edt3.com/output=in.yyy out.yyy
$ @last3 2 in.yyy VVV VVV
$ edit/edt/command=edt3.com/output=out.yyy in.yyy
$ @last3 2 out.yyy HHH VVV
$ edit/edt/command=edt3.com/output=in.yyy out.yyy
$ @last3 2 in.yyy KKK VVV
$ edit/edt/command=edt3.com/output=out.yyy in.yyy
$ @last3 2 out.yyy JJJ VVV
$ edit/edt/command=edt3.com/output=in.yyy out.yyy
$ @last3 2 in.yyy MMM VVV
$ edit/edt/command=edt3.com/output=out.yyy in.yyy
$ @last3 2 out.yyy OOO VVV
$ edit/edt/command=edt3.com/output=in.yyy out.yyy
$ @last3 2 in.yyy RRR VVV
$ edit/edt/command=edt3.com/output=out.yyy in.yyy
$ @last3 2 out.yyy SSS VVV
$ edit/edt/command=edt3.com/output=in.yyy out.yyy
$ @last3 2 in.yyy TTT VVV
$ edit/edt/command=edt3.com/output=out.yyy in.yyy
$ @last3 2 out.yyy UUU VVV
$ edit/edt/command=edt3.com/output=pinput.dat out.yyy
$ !
$ ! Modify and run the command procedure for SUXES for the appropriate
$ !     chip and device type.
$ !
$ copy/concat blank.fil,suxes.xxx,plw.dat in.yyy
$ !
$ @last3 2 in.yyy VVV VVV
$ edit/edt/command=edt3.com/output=out.yyy in.yyy
$ @last3 3 out.yyy ggg aza ccc
$ edit/edt/command=edt3.com/output=in.yyy out.yyy
$ edit/edt/command=edsuxp.way/output=suxes.com in.yyy
$ !
$ copy pinput.dat input.dat
$ copy popt.fil opt.dat
$ copy pstrat.fil strat.dat
$ !
$ @SUXES
$ !
$ ! Gather the extracted parameters.
$ !
$ copy expar.dat in.dat
$ !
$ edit/edt/command=edpar.way/output=exparp.dat in.dat
$ !
$ copy exparp.dat expar.dat
$ run CAPKEY
$ copy parspice.dat pparspice.dat
$ !
```

```
$ ! Insert the width and length into the SPICE file which obtains the
$ !     IV characteristics.
$ !
$ copy/concat blank.fil,ivcharp.xxx,plw.dat in.yyy
$ !
$ @last3 2 in.yyy BZB FFF
$ edit/edt/command=edt3.com/output=out.yyy in.yyy
$ @last3 3 out.yyy VVV VVV VVV
$ edit/edt/command=edt3.com/output=ivchar.fil out.yyy
$ !
$ copy ivchar.fil ivcharp.fil
$ !
$ ! Put together the p-channel MOSFET models file for SPICE.
$ !
$ copy/concat blank.fil,pmodel.xxx,pparspice.dat in.yyy
$ !
$ @last3 3 in.yyy vvv aqa bqb
$ edit/edt/command=edt3.com/output=out.yyy in.yyy
$ @last3 3 out.yyy cqc dqd eqe
$ edit/edt/command=edt3.com/output=in.yyy out.yyy
$ @last3 3 in.yyy fqf gqg hqh
$ edit/edt/command=edt3.com/output=out.yyy in.yyy
$ @last3 3 out.yyy iqi jqj kqk
$ edit/edt/command=edt3.com/output=in.yyy out.yyy
$ @last3 3 in.yyy lql mqm nqn
$ edit/edt/command=edt3.com/output=out.yyy in.yyy
$ @last3 3 out.yyy oqo pqp rqr
$ edit/edt/command=edt3.com/output=in.yyy out.yyy
$ @last3 1 in.yyy sqs
$ edit/edt/command=edt3.com/output=pmodel.fil in.yyy
$ !
$ COPY/CONCAT BLANK.FIL,PMODEL.FIL,IVCHARP.FIL IVCHARP.CEL
$ copy ivcharp.cel ivchar.cel
$ !
$ ! Modify the command procedure which obtains the IV characteristics.
$ !
$ copy/concat blank.fil,spiceiv.xxx,plwl.dat in.yyy
$ edit/edt/command=edspicep.way/output=out.yyy in.yyy
$ @last3 2 out.yyy gzg fzf
$ edit/edt/command=edt3.com/output=in.yyy out.yyy
$ @last3 3 in.yyy eze dzd czc
$ edit/edt/command=edt3.com/output=spiceiv.com in.yyy
$ !
$ ! Simulate the p-channel MOSFET IV characteristics.
$ !
$ @SPICEIV
$ !
$ !---------------------------------------------------------------------
$ !
$ !  Simulate the optional dynamic circuit.
$ !
$ CKT:
```

```
$ !
$ IF (L .EQS. 0) .AND. (M .EQS. 0) THEN GOTO END
$ !
$ !
$ !  Insert the correct widths and lengths into the SPICE file
$ !
$ copy/concat blank.fil,subckt.xxx,plw.dat,nlw.fil in.yyy
$ !
$ @last3 2 in.yyy hzh izi
$ edit/edt/command=edt3.com/output=out.yyy in.yyy
$ @last3 3 out.yyy VVV VVV VVV
$ edit/edt/command=edt3.com/output=in.yyy out.yyy
$ @last3 2 in.yyy jzj kzk
$ edit/edt/command=edt3.com/output=out.yyy in.yyy
$ @last3 3 out.yyy VVV VVV VVV
$ edit/edt/command=edt3.com/output=subckt.fil out.yyy
$ !
$ !
$ ! Simulate the dynamic circuit.
$ !
$ copy/concat -
 blank.fil,nmodel.fil,pmodel.fil,subckt.fil,circuit.fil circuit.cel
$ !
$ RUN SPICENBS
CIRCUIT.CEL
CIRCUIT.DAT
CIRCUIT.PLT
$ !
$ ! Plot the results of the dynamic circuit.
$ !
$ ASSIGN CIRCUIT.PLT FOR011
$ RUN S5PLOT
G
WAAARBBBCDDD$
TIME (sec)$
NODE VOLTAGE (V)$
$ RUN MCVAX
DEVICE LPA0: P300
READ IOP020.DAT
PLOT
EXIT
$! SET TERM/BRO
$ END:
$ !
$ ! Sum up the results for this chip.
$ !
$ submit/noprinter/notify RMSKEY
$ !
```

SUXES.XXX

```
$ !
$ ! FILENAME:  SUXES.COM
$ !
$ ! This command procedure runs SUXES which obtains the values for most
$ ! of the SPICE parameters.
$ !
$ purge *.*
$ RUN SSUXES
6,5
N
Y
1
INPUT
2
77WcccRazaCgggF
77WcccRazaCgggS
OPT
STRAT
N
F
22
N
EXPAR
Q
$ !
```

SPICEIV.XXX

```
$ !
$ ! FILENAME:  SPICEIV.COM
$ !
$ SET DEFAULT 'KEYSDIR'
$ !
$ ! Run SPICE to obtain the IV characteristics.
$ !
$ RUN SPICENBS
IVCHAR.CEL
IVCHAR.DAT
IVCHAR.PLT
$ !
$ ! Massage the data for the plotting program PLOTMY.
$ !
$ copy 777WczcRdzdCezeF.DAT in.dat
$ !
$ copy 777WczcRdzdCezeS.DAT in2.dat
$ !
$ copy 77722fzfgzgP.DAT in3.dat
$ !
$ copy ivchar.plt in4.dat
$ !
$ RUN IVKEY
$ !
$ !
$ ! Plot the data where:
$ !
$ !  + = IV data
$ !  * = SUXES data
$ !  - = SPICE data
$ !
$ !
$ purge *.*
$ !
$ run plotmy
n
8
out.dat
n
1
out2.dat
n
1
out3.dat
n
1
out5.dat
n
1
out6.dat
```

```
n
1
out7.dat
n
1
out8.dat
n
1
out9.dat
n
1
1
777WczcRdzdCeze$
VDS (V)$
IDS (A)$
2
n
n
y
+
+
*
-
-
-
-
-
$ RUN MCVAX
DEVICE LPA0: P300
READ IOP020.DAT
PLOT
EXIT
$ !
```

RMSKEY.COM

```
$ !
$ ! FILENAME:  RMSKEY.COM
$ !
$ !  Find the average error for the SUXES runs and gather the parameter
$ !      values for the file RESULTS.DAT.
$ !
$ SET DEFAULT 'KEYSDIR'
$ !
$ copy/concat blank.fil,results.dat,switchl.dat,freq.dat,sfreq.dat in2.dat
$ copy sys$login:keys.log in.dat
$ !
$ L=1
$ IF L .EQS. 0 THEN GOTO PCHAN
$ !
$ edit/edt/command=ednrms.way/output=nrms.dat in.dat
$ !
$ copy/concat blank.fil,in2.dat,nparspice.dat,nrms.dat in3.dat
$ copy in3.dat in2.dat
$ !
$ PCHAN:
$ M=1
$ IF M .EQS. 0 THEN GOTO CONTINUE
$ !
$ edit/edt/command=edprms.way/output=prms.dat in.dat
$ !
$ copy/concat blank.fil,in2.dat,pparspice.dat,prms.dat in3.dat
$ copy in3.dat in2.dat
$ !
$ CONTINUE:
$ !
$ edit/edt/command=edres.way/output=in.dat in2.dat
$ !
$ ! Put the parametric results into an appropriate format for each chip,
$ !      each wafer, and each lot.
$ !
$ purge *.*
$ !
$ run SUMKEY
$ !
$ copy out.dat results.dat
$ !
$ purge *.*
$ purge sys$login:*.LOG
$ !
$ N=1
$ IF N .EQS. 3 THEN GOTO LAST
$ IF N .EQS. 0 THEN GOTO NEXT
$ !
$ ! The last chip on a wafer was just analyzed.  Place the switches into the
$ !      command procedure CORKEY which determine whether or not the wafer
```

```
$ !     maps are to be made.
$ !
$ copy/concat blank.fil,corkey.xxx,switch1.dat,switch2.dat in.yyy
$ !
$ edit/edt/command=edmap.way/output=corkey.com in.yyy
$ !
$ print results.dat
$ !
$ submit/noprinter/notify STKEY
$ GOTO END
$ !
$ NEXT:
$ !
$ ! Consider the next chip.
$ !
$ submit/noprinter/notify KEYS
$ GOTO END
$ !
$ LAST:
$ print results.dat
$ !
$ END:
$ !
```

STKEY.COM

```
$ !
$ ! FILENAME:  STKEY.COM
$ !
$ SET DEFAULT 'KEYSDIR'
$ !
$ ! Prepare the parametric data files for STAT2.
$ !
$ edit/edt/command=edskey.way/output=in.dat w.dat
$ copy/concat blank.fil,stend.fil,in.dat w.dat
$ !
$ edit/edt/command=edskey.way/output=in.dat l.dat
$ copy/concat blank.fil,stend.fil,in.dat l.dat
$ !
$ edit/edt/command=edskey.way/output=in.dat wp.dat
$ copy/concat blank.fil,stend.fil,in.dat wp.dat
$ !
$ edit/edt/command=edskey.way/output=in.dat lp.dat
$ copy/concat blank.fil,stend.fil,in.dat lp.dat
$ !
$ edit/edt/command=edskey.way/output=in.dat mckt.dat
$ copy/concat blank.fil,stend.fil,in.dat mckt.dat
$ !
$ edit/edt/command=edskey.way/output=in.dat sckt.dat
$ copy/concat blank.fil,stend.fil,in.dat sckt.dat
$ !
$ edit/edt/command=edskey.way/output=in.dat pdiff.dat
$ copy/concat blank.fil,stend.fil,in.dat pdiff.dat
$ !
$ !
$ edit/edt/command=edskey.way/output=in.dat ncgdo.dat
$ copy/concat blank.fil,stend.fil,in.dat ncgdo.dat
$ !
$ edit/edt/command=edskey.way/output=in.dat ncgbo.dat
$ copy/concat blank.fil,stend.fil,in.dat ncgbo.dat
$ !
$ edit/edt/command=edskey.way/output=in.dat nrms.dat
$ copy/concat blank.fil,stend.fil,in.dat nrms.dat
$ !
$ edit/edt/command=edskey.way/output=in.dat nld.dat
$ copy/concat blank.fil,stend.fil,in.dat nld.dat
$ !
$ edit/edt/command=edskey.way/output=in.dat ntox.dat
$ copy/concat blank.fil,stend.fil,in.dat ntox.dat
$ !
$ edit/edt/command=edskey.way/output=in.dat nsub.dat
$ copy/concat blank.fil,stend.fil,in.dat nsub.dat
$ !
$ edit/edt/command=edskey.way/output=in.dat nvto.dat
$ copy/concat blank.fil,stend.fil,in.dat nvto.dat
$ !
```

```
$ edit/edt/command=edskey.way/output=in.dat nkp.dat
$ copy/concat blank.fil,stend.fil,in.dat nkp.dat
$ !
$ edit/edt/command=edskey.way/output=in.dat ngamma.dat
$ copy/concat blank.fil,stend.fil,in.dat ngamma.dat
$ !
$ edit/edt/command=edskey.way/output=in.dat nphi.dat
$ copy/concat blank.fil,stend.fil,in.dat nphi.dat
$ !
$ edit/edt/command=edskey.way/output=in.dat nu0.dat
$ copy/concat blank.fil,stend.fil,in.dat nu0.dat
$ !
$ edit/edt/command=edskey.way/output=in.dat ndelta.dat
$ copy/concat blank.fil,stend.fil,in.dat ndelta.dat
$ !
$ edit/edt/command=edskey.way/output=in.dat nvmax.dat
$ copy/concat blank.fil,stend.fil,in.dat nvmax.dat
$ !
$ edit/edt/command=edskey.way/output=in.dat nxj.dat
$ copy/concat blank.fil,stend.fil,in.dat nxj.dat
$ !
$ edit/edt/command=edskey.way/output=in.dat nkappa.dat
$ copy/concat blank.fil,stend.fil,in.dat nkappa.dat
$ !
$ edit/edt/command=edskey.way/output=in.dat nnfs.dat
$ copy/concat blank.fil,stend.fil,in.dat nnfs.dat
$ !
$ edit/edt/command=edskey.way/output=in.dat neta.dat
$ copy/concat blank.fil,stend.fil,in.dat neta.dat
$ !
$ edit/edt/command=edskey.way/output=in.dat ntheta.dat
$ copy/concat blank.fil,stend.fil,in.dat ntheta.dat
$ !
$ edit/edt/command=edskey.way/output=in.dat nerr.dat
$ copy/concat blank.fil,stend.fil,in.dat nerr.dat
$ !
$ purge *.*
$ !
$ edit/edt/command=edskey.way/output=in.dat pcgdo.dat
$ copy/concat blank.fil,stend.fil,in.dat pcgdo.dat
$ !
$ edit/edt/command=edskey.way/output=in.dat pcgbo.dat
$ copy/concat blank.fil,stend.fil,in.dat pcgbo.dat
$ !
$ edit/edt/command=edskey.way/output=in.dat prms.dat
$ copy/concat blank.fil,stend.fil,in.dat prms.dat
$ !
$ edit/edt/command=edskey.way/output=in.dat pld.dat
$ copy/concat blank.fil,stend.fil,in.dat pld.dat
$ !
$ edit/edt/command=edskey.way/output=in.dat ptox.dat
$ copy/concat blank.fil,stend.fil,in.dat ptox.dat
```

```
$ !
$ edit/edt/command=edskey.way/output=in.dat psub.dat
$ copy/concat blank.fil,stend.fil,in.dat psub.dat
$ !
$ edit/edt/command=edskey.way/output=in.dat pvto.dat
$ copy/concat blank.fil,stend.fil,in.dat pvto.dat
$ !
$ edit/edt/command=edskey.way/output=in.dat pkp.dat
$ copy/concat blank.fil,stend.fil,in.dat pkp.dat
$ !
$ edit/edt/command=edskey.way/output=in.dat pgamma.dat
$ copy/concat blank.fil,stend.fil,in.dat pgamma.dat
$ !
$ edit/edt/command=edskey.way/output=in.dat pphi.dat
$ copy/concat blank.fil,stend.fil,in.dat pphi.dat
$ !
$ edit/edt/command=edskey.way/output=in.dat pu0.dat
$ copy/concat blank.fil,stend.fil,in.dat pu0.dat
$ !
$ edit/edt/command=edskey.way/output=in.dat pdelta.dat
$ copy/concat blank.fil,stend.fil,in.dat pdelta.dat
$ !
$ edit/edt/command=edskey.way/output=in.dat pvmax.dat
$ copy/concat blank.fil,stend.fil,in.dat pvmax.dat
$ !
$ edit/edt/command=edskey.way/output=in.dat pxj.dat
$ copy/concat blank.fil,stend.fil,in.dat pxj.dat
$ !
$ edit/edt/command=edskey.way/output=in.dat pkappa.dat
$ copy/concat blank.fil,stend.fil,in.dat pkappa.dat
$ !
$ edit/edt/command=edskey.way/output=in.dat pnfs.dat
$ copy/concat blank.fil,stend.fil,in.dat pnfs.dat
$ !
$ edit/edt/command=edskey.way/output=in.dat peta.dat
$ copy/concat blank.fil,stend.fil,in.dat peta.dat
$ !
$ edit/edt/command=edskey.way/output=in.dat ptheta.dat
$ copy/concat blank.fil,stend.fil,in.dat ptheta.dat
$ !
$ edit/edt/command=edskey.way/output=in.dat perr.dat
$ copy/concat blank.fil,stend.fil,in.dat perr.dat
$ !
$ purge *.*
$ !
$ !  Create the database for STAT2.
$ !
$ run crdb
SUXES'S VALUES
4
$ !
$ !  Run STAT2 to fill the database and obtain the correlation coefficients.
```

```
$ !
$ run STAT2
asg w.dat
rea 1 0
xlt 1.0E-12
prs
axp
wdb,1,1,1,1,2
ASG 1.DAT
REA 1 0
XLT 1.0E-12
PRS
AXP
WDB,1,1,1,1,3
asg wp.dat
rea 1 0
xlt 1.0E-12
prs
axp
wdb,1,1,1,1,4
ASG 1p.DAT
REA 1 0
XLT 1.0E-12
PRS
AXP
WDB,1,1,1,1,5
asg mckt.dat
rea 1 0
xlt 1.0E-12
prs
axp
wdb,1,1,1,1,6
ASG sckt.DAT
REA 1 0
XLT 1.0E-12
PRS
AXP
WDB,1,1,1,1,7
asg pdiff.dat
rea 1 0
xlt 1.0E-12
prs
axp
wdb,1,1,1,1,8
asg ncgdo.dat
rea 1 0
xlt 1.0E-12
prs
axp
wdb,1,1,1,1,9
asg ncgbo.dat
rea 1 0
```

```
xlt 1.0E-12
prs
axp
wdb,1,1,1,1,10
asg nrms.dat
rea 1 0
xlt 1.0E-12
prs
axp
wdb,1,1,1,1,11
asg nld.dat
rea 1 0
xlt 1.0E-12
prs
axp
wdb,1,1,1,1,12
asg ntox.dat
rea 1 0
xlt 1.0E-12
prs
axp
wdb,1,1,1,1,13
asg nsub.dat
rea 1 0
xlt 1.0E-12
prs
axp
wdb,1,1,1,1,14
asg nvto.dat
rea 1 0
xlt 1.0E-12
prs
axp
wdb,1,1,1,1,15
asg nkp.dat
rea 1 0
xlt 1.0E-12
prs
axp
wdb,1,1,1,1,16
asg ngamma.dat
rea 1 0
xlt 1.0E-12
prs
axp
wdb,1,1,1,1,17
asg nphi.dat
rea 1 0
xlt 1.0E-12
prs
axp
wdb,1,1,1,1,18
```

```
asg nu0.dat
rea 1 0
xlt 1.0E-12
prs
axp
wdb,1,1,1,1,19
asg ndelta.dat
rea 1 0
xlt 1.0E-12
prs
axp
wdb,1,1,1,1,20
asg nvmax.dat
rea 1 0
xlt 1.0E-12
prs
axp
wdb,1,1,1,1,21
asg nxj.dat
rea 1 0
xlt 1.0E-12
prs
axp
wdb,1,1,1,1,22
asg nkappa.dat
rea 1 0
xlt 1.0E-12
prs
axp
wdb,1,1,1,1,23
asg nnfs.dat
rea 1 0
xlt 1.0E-12
prs
axp
wdb,1,1,1,1,24
asg neta.dat
rea 1 0
xlt 1.0E-12
prs
axp
wdb,1,1,1,1,25
asg ntheta.dat
rea 1 0
xlt 1.0E-12
prs
axp
wdb,1,1,1,1,26
asg nerr.dat
rea 1 0
xlt 1.0E-12
prs
```

```
axp
wdb,1,1,1,1,27
asg pcgdo.dat
rea 1 0
xlt 1.0E-12
prs
axp
wdb,1,1,1,1,28
asg pcgbo.dat
rea 1 0
xlt 1.0E-12
prs
axp
wdb,1,1,1,1,29
asg prms.dat
rea 1 0
xlt 1.0E-12
prs
axp
wdb,1,1,1,1,30
asg pld.dat
rea 1 0
xlt 1.0E-12
prs
axp
wdb,1,1,1,1,31
asg ptox.dat
rea 1 0
xlt 1.0E-12
prs
axp
wdb,1,1,1,1,32
asg psub.dat
rea 1 0
xlt 1.0E-12
prs
axp
wdb,1,1,1,1,33
asg pvto.dat
rea 1 0
xgt -1.0E-12
prs
axp
wdb,1,1,1,1,34
asg pkp.dat
rea 1 0
xlt 1.0E-12
prs
axp
wdb,1,1,1,1,35
asg pgamma.dat
rea 1 0
```

```
xlt 1.0E-12
prs
axp
wdb,1,1,1,1,36
asg pphi.dat
rea 1 0
xlt 1.0E-12
prs
axp
wdb,1,1,1,1,37
asg pu0.dat
rea 1 0
xlt 1.0E-12
prs
axp
wdb,1,1,1,1,38
asg pdelta.dat
rea 1 0
xlt 1.0E-12
prs
axp
wdb,1,1,1,1,39
asg pvmax.dat
rea 1 0
xlt 1.0E-12
prs
axp
wdb,1,1,1,1,40
asg pxj.dat
rea 1 0
xlt 1.0E-12
prs
axp
wdb,1,1,1,1,41
asg pkappa.dat
rea 1 0
xlt 1.0E-12
prs
axp
wdb,1,1,1,1,42
asg pnfs.dat
rea 1 0
xlt 1.0E-12
prs
axp
wdb,1,1,1,1,43
asg peta.dat
rea 1 0
xlt 1.0E-12
prs
axp
wdb,1,1,1,1,44
```

```
asg ptheta.dat
rea 1 0
xlt 1.0E-12
prs
axp
wdb,1,1,1,1,45
asg perr.dat
rea 1 0
xlt 1.0E-12
prs
axp
wdb,1,1,1,1,46
GET 6
SDB ALL COEF 0.0
GET 7
SDB ALL COEF 0.0
GET 8
SDB ALL COEF 0.0
end
$ !
$ ! Gather and rank-order the correlation coefficient information.
$ !
$ submit/noprinter/notify CORKEY
$ !
```

CORKEY.XXX

```
$ !
$ ! FILENAME:  CORKEY.COM
$ !
$ SET DEFAULT 'KEYSDIR'
$ !
$ copy sys$login:stkey.log in2.dat
$ !
$ ! Gather and rank-order the correlation coefficients with respect to the
$ !     simulated frequency.
$ !
$ edit/edt/command=edsimf.way/output=in.dat in2.dat
$ !
$ run RANKEY
$ print out.dat
$ !
$ ! Gather and rank-order the correlation coefficients with respect to the
$ !     percent difference between the measured and simulated frequencies.
$ !
$ edit/edt/command=edpdiff.way/output=in.dat in2.dat
$ !
$ run RANKEY
$ copy out2.dat mapl.dat
$ print out.dat
$ !
$ ! Gather and rank-order the correlation coefficients with respect to the
$ !     measured frequency.
$ !
$ edit/edt/command=edmeaf.way/output=in.dat in2.dat
$ !
$ run RANKEY
$ copy out2.dat map2.dat
$! copy out.dat save.dat
$ print out.dat
$ !
$ K=1
$ N=2
$ IF K .EQS. 0 THEN GOTO MID
$ !
$ ! Modify the command procedure for MAPKEY to include the appropriate lot
$ !     and wafer.
$ !
$ copy/concat blank.fil,mapkey.xxx,lotnumber.fil,nlw.fil in.zzz
$ copy/concat blank.fil,in.zzz,map2.dat,mapl.dat,switchl.dat in.yyy
$ !
$ edit/edt/command=edmaps.way/output=out.yyy in.yyy
$ @last3 3 out.yyy EEE DDD CCC
$ edit/edt/command=edt3.com/output=in.yyy out.yyy
$ @last3 3 in.yyy BBB QQQ QQQ
$ edit/edt/command=edt3.com/output=out.yyy in.yyy
$ @last3 3 out.yyy QQQ QQQ ZZZ
```

```
$ edit/edt/command=edt3.com/output=in.yyy out.yyy
$ @last3 1 in.yyy AAA
$ edit/edt/command=edt3.com/output=mapkey.com in.yyy
$ !
$ submit/noprinter/notify MAPKEY
$ GOTO END
$ !
$ MID:
$ IF N .EQS. 2 THEN GOTO FINISH
$ !
$ ! Consider the first chip on the next wafer.
$ !
$ submit/noprinter/notify KEYS
$ GOTO END
$ !
$ FINISH:
$ copy results.dat oresults.dat
$ END:
```

MAPKEY.XXX

```
$ !
$ ! FILENAME:  MAPKEY.COM
$ !
$ !  Generate the FIRST wafer map for the measured frequency.
$ !
$ SET DEFAULT 'KEYSDIR'
$ run crdb
SUXES'S VALUES
4
$ !
$ purge *.*
$ !
$ run STAT2
asg mckt.dat
rea 1 0
prs
xlt 1.0E-12
axp
prs
MP4,8,1,1,0,0,0
MEASURED FREQUENCY (MHz) - LOT AAA WAFER ZZZ
END
$ PRINT/NOFEED STAT2.LOG
$ !
$ !  Generate the SECOND wafer map for the simulated frequency.
$ !
$ run STAT2
asg sckt.dat
rea 1 0
prs
xlt 1.0E-12
axp
prs
MP4,8,1,1,0,0,0
SIMULATED FREQUENCY (MHz) - LOT AAA WAFER ZZZ
END
$ PRINT/NOFEED STAT2.LOG
$ !
$ !  Generate the THIRD wafer map for the percent difference.
$ !
$ run STAT2
asg pdiff.dat
rea 1 0
prs
xlt 1.0E-12
axp
prs
MP4,8,1,1,0,0,0
PERCENT DIFFERENCE (%) - LOT AAA WAFER ZZZ
END
```

```
$ PRINT/NOFEED STAT2.LOG
$ !
$ !  Generate the FOURTH wafer map for the highest correlated parameter
$ !      with respect to the measured frequency.
$ !
$ run STAT2
asg BBB.dat
rea 1 0
prs
xlt 1.0E-12
axp
prs
MP4,8,1,1,0,0,0
BBB - LOT AAA WAFER ZZZ
END
$ PRINT/NOFEED STAT2.LOG
$ !
$ !  Generate the FIFTH wafer map for the second highest correlated parameter
$ !      with respect to the measured frequency.
$ !
$ run STAT2
asg CCC.dat
rea 1 0
prs
xlt 1.0E-12
axp
prs
MP4,8,1,1,0,0,0
CCC - LOT AAA WAFER ZZZ
END
$ PRINT/NOFEED STAT2.LOG
$ !
$ !  Generate the SIXTH wafer map for the highest correlated parameter
$ !      with respect to the percent difference.
$ !
$ run STAT2
asg DDD.dat
rea 1 0
prs
xlt 1.0E-12
axp
prs
MP4,8,1,1,0,0,0
DDD - LOT AAA WAFER ZZZ
END
$ PRINT/NOFEED STAT2.LOG
$ !
$ !  Generate the SEVENTH wafer map for the second highest correlated parameter
$ !      with respect to the percent difference.
$ !
$ run STAT2
asg EEE.dat
```

```
rea 1 0
prs
xlt 1.0E-12
axp
prs
MP4,8,1,1,0,0,0
EEE - LOT AAA WAFER ZZZ
END
$ PRINT/NOFEED STAT2.LOG
$ N=2
$ IF N .EQS. 2 THEN GOTO FINISH
$ !
$ ! Consider the first chip on the next wafer.
$ !
$ submit/noprinter/notify KEYS
$ GOTO END
$ !
$ FINISH:
$ copy results.dat oresults.dat
$ END:
$ !
$ ! THE END !
$ !
```

RCKEY.FOR

```
C
C       FILENAME:  RCKEY.FOR
C
C       This program chooses the next chip to consider for SUXES
C               and SPICE.
C
C
        DIMENSION IW(25), IR(100), IC(100)
C
        character*4 lotnum1
        character*4 lotnum2
C
        open(unit=20, file='rc.fil', status='old')
        open(unit=30, file='line.fil', status='old')
        open(unit=40, file='in.dat', status='old')
        open(unit=45, file='nmodel.fil', status='unknown')
        open(unit=46, file='pmodel.fil', status='unknown')
        open(unit=50, file='plw.dat', status='unknown')
        open(unit=55, file='plwl.dat', status='unknown')
        open(unit=60, file='nlw.fil', status='unknown')
        open(unit=61, file='nlwl.dat', status='unknown')
        open(unit=70, file='freq.dat', status='unknown')
        open(unit=90, file='switchl.dat', status='unknown')
        open(unit=98, file='lotnumber.fil', status='old')
        open(unit=99, file='results.dat', status='unknown')
        ifirst=1
        assign=0.0
C
C       READ THE FILE
C
        MO=0
        DO 378 MO=1,7
        READ (20,377)IDUMMY
377     FORMAT (A1)
378     CONTINUE
C
        MO=0
        DO 478 MO=1,5
        READ (30,377)IDUMMY
478     CONTINUE
C
C       READ THE PREVIOUS ROW AND COLUMN
C
        read (40,11) IWLW
11      format (A1)
C       write (90,21) iwlw
21      format (1x,A1)
        READ (40,41) IRLW
        READ (40,41) ICLW
        if (irlw .ne. 0) go to 102
```

```
      assign=1.0
      read (98,734) lotnuml, lotnum2
734   format (2A4)
C-----------------------------------------------------------------------
C
C     WRITE THE HEADING
C
      WRITE (99,1)
1     format(1x,'#')
      WRITE (99,2)
2     format(1x,'*  FILENAME = RESULTS.DAT')
      WRITE (99,4)
      WRITE (99,3) lotnuml, lotnum2
3     format(1x,'*  Lot Number = ',1x,2A4)
      WRITE (99,4)
4     format(1x,'*')
      WRITE (99,5)
5     format(1x,'*  W1   R   C    Wn        Ln          Wp ',
     +'       Lp ')
      WRITE (99,667)
667   format(1x,'*                MCKT      SCKT        PDIFF')
      WRITE (99,4)
      WRITE (99,6)
6     format (1x,'*               CGDO      CGSO        CGBO',
     +'      RMS  ')
      WRITE (99,7)
7     format (1x,'*                LD        TOX        SUB ',
     +'      VTO       KP')
      WRITE (99,8)
8     format(1x,'*               GAMMA      PHI         UO ',
     +'      DELTA    VMAX')
      WRITE (99,9)
9     format(1x,'*                XJ       KAPPA       NFS ',
     +'      ETA      THETA       ERR')
      write(99,666)
666   format(1x,'****************************************************',
     + '*********************')
C-----------------------------------------------------------------------
C
C     FIND THIS ROW AND COLUMN IN RC.FIL
C
102   I=0
800   I=I+1
      READ (20,12,END=231) IW(I), IR(I), IC(I)
12    format (A1,1x,I2,1x,I2)
C     write (90,22) IW(I), IR(I), IC(I)
22    format (1x,A1,2x,I1,2x,I1)
      if (assign .eq. 0.0) go to 222
      iwlw=iw(I)
      irlw=ir(I)
      iclw=ic(I)
      iwaf=iw(I)
```

```
      IROW=IR(I)
      ICOL=IC(I)
      go to 880
222   if (IW(I) .ne. iwlw) go to 800
      IF ((IR(I) .EQ. IRLW) .AND. (IC(I) .EQ. ICLW)) GO TO 700
      GO TO 800
C
C     CHIP NOT FOUND THEREFORE START FROM SCRATCH
C
231   iwaf=iw(1)! Therefore first chip
      IROW=IR(1)
      ICOL=IC(1)
      ind=1
      ifirst=0
      GO TO 880
C
C     THE CURRENT ROW AND COLUMN HAS BEEN FOUND...FIND THE NEXT CHIP
C
700   READ (20,12,END=900) IW(I), IR(I), IC(I)
C     write (90,22) IW(I), IR(I), IC(I)
      iwaf=IW(I)
      IROW=IR(I)
      ICOL=IC(I)
C
C     FIND THE LENGTH AND WIDTH
C
880   READ (30,15, END=900) IIW, IIR, IIC, Wn, ALn, Wp, ALp, FREQ, IFLG
15    FORMAT (2x,A1,2x,I2,1X,I2,1X,F5.2,1X,F5.2,1X,F5.2,1x,F5.2,
     +1x,F5.2,1x,I3)
C     write(90,25) iiw, iir,iic,wn,aln,wp,alp,freq,iflg
25    FORMAT (3x,A1,2x,I2,1X,I2,1X,F5.2,1X,F5.2,1X,F5.2,1x,F5.2,
     +1X,F5.2,1x,I3)
      if(iwaf .ne. iiw) go to 880
      IF ((IROW .EQ. IIR) .AND. (ICOL .EQ. IIC)) GO TO 770
      GO TO 880
C
C     Reasonable W, L, and FREQ ?
C
770   IF((((WN.GE.0.00).AND.(ALN.GE.0.00)) .AND. ((WP.GE.0.00).AND.
     +(ALP.GE.0.00))) .AND. (FREQ.GE.0.00)) GO TO 777
C
C     GO TO THE NEXT CHIP
C
      if (ifirst .eq. 1) GO TO 700
      ind = ind + 1
      iwaf = iw(ind)
      irow = ir(ind)
      icol = ic(ind)
      go to 880
C
C     WRITE THE VALUES IN NLW.FIL, PLW.DAT, FREQ.DAT, NLW1.DAT, and PLW1.DAT
C
```

```
777     write (60,10) IWAF
        WRITE (60,41) IROW
        WRITE (60,41) ICOL
        WRITE (60,51) WN
        WRITE (60,51) ALN
        write (50,10) iwaf
        WRITE (50,41) IROW
        WRITE (50,41) ICOL
        WRITE (50,51) WP
        WRITE (50,51) ALP
        write (70,10) iwaf
        WRITE (70,41) IROW
        WRITE (70,41) ICOL
        WRITE (70,51) WN
        WRITE (70,51) ALN
        WRITE (70,51) WP
        WRITE (70,51) ALP
        WRITE (70,51) FREQ
C
        LN = IFIX(ALN*10)
        LP = IFIX(ALP*10)
        MWN = IFIX(WN*10)
        IF (WN .GE. 10.0) MWN=IFIX(WN)
        MWP = IFIX(WP*10)
        IF (WP .GE. 10.0) MWP=IFIX(WP)
C
        write (61,53) IWAF
        WRITE (61,42) IROW
        WRITE (61,42) ICOL
        WRITE (61,42) MWN
        WRITE (61,42) LN
        write (55,53) iwaf
        WRITE (55,42) IROW
        WRITE (55,42) ICOL
        WRITE (55,42) MWP
        WRITE (55,42) LP
41      FORMAT (I2)
42      FORMAT (1x,I2)
51      FORMAT (F5.2)
10      FORMAT (A1)
53      FORMAT (1x,A1)
        GO TO 888
C
900     WRITE (6,*) ' NO MORE CHIPS LEFT OR NO MATCH'
        IROW=0
        ICOL=0
        ALN=0.0
        ALP=0.0
        WN=0.0
        WP=0.0
        FREQ=0.0
        write (60,10) iwaf
```

```
      WRITE (60,41) IROW
      WRITE (60,41) ICOL
      WRITE (60,51) WN
      WRITE (60,51) ALN
      write (50,10) iwaf
      WRITE (50,41) IROW
      WRITE (50,41) ICOL
      WRITE (50,51) WP
      WRITE (50,51) ALP
      write (70,10) iwaf
      WRITE (70,41) IROW
      WRITE (70,41) ICOL
      WRITE (70,51) WN
      WRITE (70,51) ALN
      WRITE (70,51) WP
      WRITE (70,51) ALP
      WRITE (70,51) FREQ
888   CONTINUE
C
C     IFLG=1    JUST N-CHANNEL
C     IFLG=2    JUST P-CHANNEL
C     IFLG=3    BOTH
C
      IF (IFLG .EQ. 3) GO TO 503
      IF (IFLG .EQ. 2) GO TO 502
      WRITE (90,511)
      WRITE (90,512)
      write(46,39)
39    format(1x,'*')
      GO TO 1000
502   WRITE (90,510)
      WRITE (90,513)
      write(45,39)
      GO TO 1000
503   WRITE (90,511)
      WRITE (90,513)
1000  CONTINUE
510   FORMAT('$ L=0')
511   FORMAT('$ L=1')
512   FORMAT('$ M=0')
513   FORMAT('$ M=1')
C
C     should the maps be generated ?
C
      if (ifirst .eq. 0) go to 995
994   read (20,12,end=999) iwaf2, irow2, icol2
997   read (30,15,end=999) iiw, iir, iic, wn, aln, wp, alp, freq, iflg
      if (iwaf2 .ne. iiw) go to 997
      if ((irow2 .eq. iir) .and. (icol2 .eq. iic)) go to 996
      go to 997
996   if ((((wn .gt. 0.0) .and. (aln .gt. 0.0)) .and. ((wp .gt. 0.0)
     +.and. (alp .gt. 0.0))) .and. (freq .gt. 0.0)) go to 993
```

```
      go to 994
C
993   if(iwaf2 .ne. iwaf) go to 992
995   write(90,61)!  Same wafer...
61    format('$ N=0')
      go to 998
C
992   write(90,81)!  Different wafer...
81    format('$ N=1')
      go to 998
C
999   if (assign .eq. 1.0) write(90,72)
72    format('$ N=3')!  One chip...
      if (assign .eq. 1.0) go to 998
C
      write (90,71)!  End of lot...
71    format('$ N=2')
C
998   continue
      END
```

SUKEY.FOR

```
C
C      FILENAME:  SUKEY.FOR
C
C      This program takes the IV data from the ACCUTEST and
C      makes it suitable for the SUXES program.
C
C
       DIMENSION VDS(1000), VGS(1000), VBS(1000), CIDS(1000)
       DIMENSION X1(100), Y1(100), VBSZ(100)
       DIMENSION VDSBE(1000), VGSBE(1000), CIDSBE(1000)
       real x(250),y(250),vds2(500,50),beta,vt
       integer n2(50),np(50)
C
       open(unit=20, file='in.dat', status='old')
       open(unit=31, file='vbin.dat', status='old')
       open(unit=32, file='in3.dat', status='old')
       open(unit=40, file='out.dat', status='unknown')
       open(unit=50, file='out2.dat', status='unknown')
       open(unit=80, file='guest.dat', status='unknown')
       open(unit=90, file='vbout.dat', status='unknown')
C
       ES = 103.5E-12                  ! in F/m
       EOX = 34.5E-12                  ! in F/m
       Q = 1.602E-19                   ! in C
       u0=0.0
       xkp=0.0
       vto=0.0
C
       MO=0
       DO 178 MO=1,16
       READ (20,177)IDUMMY
177    FORMAT (A1)
178    CONTINUE
C
       WRITE (40,2)
       WRITE (50,2)
2      FORMAT ('*',5x,'Data from the ACCUTEST')
       write (40,3)
       write (50,3)
3      format ('*',5x,'VDS',10x,'VGS',15x,'VBS',11x,'IDS')
C
       write (6,4)
4      format(' Width = ')
       read (5,*) W
C
       write (6,5)
5      format(' Length = ')
       read (5,*) AL
C
       write (40,6) W, AL
```

```
      write (50,6) W, AL
6     format (10x,'Width/Length=',F5.2,'/',F5.2)
C
C     READ THE DATA AND WRITE IT INTO THE OUTPUT FILES
C
      I=0
230   I=I+1
      KEY=40
C*600  READ (20,*,END=231) VDS(I), VGS(I), VBS(I), CIDS(I)
600   READ (20,666,END=231) IDUM, IONE, VDS(I)
      READ (20,666,END=231) IDUM, ITWO, VGS(I)
      READ (20,666,END=231) IDUM, ITHREE, VBS(I)
      READ (20,666,END=231) IDUM, IFOUR, CIDS(I)
666   format (A4, I6, 1PE13.6)
      VDS(I) = (AINT(100.*VDS(I)))/100
      VGS(I) = (AINT(100.*VGS(I)))/100
      VBS(I) = (AINT(100.*VBS(I)))/100
      IF (VDS(I) .LT. 0.00) CIDS(I)=-CIDS(I)
      IF (ABS(VDS(I)) .GT. 0.35) KEY=50
      WRITE (KEY,401) VDS(I), VGS(I), VBS(I), CIDS(I)
401   FORMAT (5X, F5.2, 8X, F5.2, 10X, F5.2, 5X, 1PE15.5)
      GO TO 230
231   CONTINUE
      NUM=I-1
C
C     ORDER ALL THE DATA FOR THE BETA PROGRAM
C
450   I=0
      KC=0
      VDSLAST=0
400   VDSY=100.0
      K=0
      J=0
      DO 88 K=1,NUM
      IF ((ABS(VDS(K)) .LT. ABS(VDSY)) .AND.
     + (ABS(VDS(K)) .GT. ABS(VDSLAST))) VDSY=VDS(K)
88    CONTINUE
      IF (ABS(VDSY) .EQ. VDSLAST) GO TO 927 ! LAST VDS VALUE ALREADY USED
      I=I+1                                 ! CURVE NUMBER
C
C     FOUND ONE VDS VALUE NOW FIND THE VGS VALUES
C
      VGSLAST=0.0
86    VGSY=100.0
      KB=0
      KA=0
      DO 89 KA=1,NUM
      IF (ABS(VDS(KA)) .NE. ABS(VDSY)) GO TO 89
      IF ((ABS(VGS(KA)) .LT. ABS(VGSY)) .AND.
     + (ABS(VGS(KA)) .GT. ABS(VGSLAST))) GO TO 82
      GO TO 89
82    VGSY=VGS(KA)
```

```
      KB=KA
89    CONTINUE
C
C     FOUND A VGS VALUE
C
      IF (KB .EQ. 0) GO TO 76          ! CURVE COMPLETE - NO VGS VALUE
      VGSLAST=VGSY
      VDSLAST=VDSY
      KC=KC+1
      J=J+1                            ! POINTS PER CURVE INCREMENT
      VDS2(J,I)=VDSY
      X(J)=VGSY
      Y(J)=CIDS(KB)
      N2(I)=J                          ! POINTS PER CURVE
      IF (NUM .EQ. KC) GO TO 927       ! OUT OF POINTS
      GO TO 86
C
76    CONTINUE
      WRITE (90,103) I, N2(I), VDS2(1,I)
103   FORMAT (' Number of points for curve ',i2,' is ',i3,
     + ' with Vds = ',f10.4)
      CALL GAIN (Y,X,N2(I),5,VDS2(1,I),BETA,VT)
      XKP=BETA*AL/W
      IF (ABS(VDS2(1,I)) .NE. 0.2) GO TO 801     ! WRONG VDS
      XKPO=XKP                                   ! IN (UA/V**2)
      VTO=VT
801   CONTINUE
      WRITE (90,*) ' KP= ',XKP,' VT= ',VT
      GO TO 400                                  ! NEXT CURVE
C
C     DETERMINE:
C             SUB (IN CM-3)
C             PHI (IN V)
C             GAMMA (IN V**.5)
C             XJ (IN UM)
C             CGDO (IN F/M)
C             CGBO (IN F/M)
C             UO (IN CM**2/VSEC)
C     FOR THE INPUTS:
C             TOX (IN A)
C             LD (IN UM)
C
C
C
C     READ AND MASSAGE THE DATA
C
927   MO=0
      do 91 MO=1,14
      read (31,177)idummy
91    continue
C
      phif2=.6
```

```
      KO=0
      do 37 KO=1,4
      read (31,566,end=131) IDUM, IONE, VBS(KO)
566   format (A4, I6, 1PE13.6)
      read (31,566,end=131) IDUM, IONE, Y1(KO)
C*      READ (31,*,END=131) VBS(KO), Y1(KO)
      X1(KO)=sqrt(abs(VBS(KO))+phif2) - sqrt(phif2)
      Y1(KO) = ABS(Y1(KO))
      write (90,*) VBS(KO), X1(KO), Y1(KO)
37    continue
C
      call lnfit(3,2,X1,Y1,slopep,xint,rmspsub)
      write (90,*) 'slopep',slopep
      KO=0
      do 47 KO=5,8
      read (31,566,end=131) IDUM, ITWO, VBS(KO)
      read (31,566,end=131) IDUM, ITWO, Y1(KO)
C*      READ (31,*,END=131) VBS(KO), Y1(KO)
      X1(KO)=sqrt(abs(VBS(KO))+phif2) - sqrt(phif2)
      Y1(KO) = ABS(Y1(KO))
      write (90,*) VBS(KO), X1(KO), Y1(KO)
47    continue
      call lnfit(3,6,X1,Y1,slopen,xint,rmsnsub)
      write (90,*) 'slopen',slopen
C
C     READ THE DATA
C
      READ (32,177) IDUMMY
C
      IKEY=32
      OKEY=80
130   READ (IKEY,*,END=131) tox, xld
C
      if(vdslast .ge. 0.0) gamma=abs(slopen)
      if(vdslast .lt. 0.0) gamma=abs(slopep)
      if(vdslast .ge. 0.0) rmssub=rmsnsub
      if(vdslast .lt. 0.0) rmssub=rmspsub
C      write (90,*) 'ikey=',ikey
C      write (90,*) 'gamma=',gamma
      sub = (1.0e-6/(2.*es*q))*(gamma*eox/(tox*1.0e-10))**2   ! IN (CM-3)
C      write (90,*) 'sub=',sub
      phi = 2.0*.02585*alog(sub/1.45e10)
      xj = xld/.8
      cgdo = eox*xld*1.0e-6/(tox*1.0e-10)                  ! IN (F/M)
      cgso = cgdo
      cgbo = eox*4.*1.0e-6/(2.*tox*1.0e-10)
      u0 = (xkpo*tox*1.0e-8*1.0e-6)/(eox*1.0e-2)           ! IN ((CM**2)/V-SEC)
C
      WRITE (OKEY,375)
375   FORMAT (1X, 'LD')
      WRITE (OKEY,376) xld
376   FORMAT (1X, F5.3)
```

```
C
      WRITE (OKEY,373)
373   FORMAT (1X, 'TOX')
      WRITE (OKEY,374) tox
374   FORMAT (1X, F5.1)
C
      WRITE (OKEY,377)
377   FORMAT(1X,'SUB')
      WRITE (OKEY,378) sub
378   FORMAT (1X,1E9.3)
C
      write (okey,903)
903   format(1x, 'vto')
      write (okey,904) vto
904   format (1x,f6.3)
C
      write (okey,905)
905   format(1x, 'xkp')
      write (okey,906) xkpo
906   format (1x,F6.3)
C
      WRITE (OKEY,577)
577   FORMAT(1X,'GAMMA')
      WRITE (OKEY,578) gamma
578   FORMAT (1X,F5.3)
C
      WRITE (OKEY,477)
477   FORMAT(1X,'PHI')
      WRITE (OKEY,478) phi
478   FORMAT (1X,F5.3)
C
      write (okey,901)
901   format(1x, 'u0')
      write (okey,902) u0
902   format (1x,F6.1)
C
      WRITE (OKEY,677)
677   FORMAT(1X,'XJ')
      WRITE (OKEY,678) xj
678   FORMAT (1X,F5.3)
C
      WRITE (OKEY,777)
777   FORMAT(1X,'CGDO')
      WRITE (OKEY,778) cgdo
778   FORMAT (1X,1E9.3)
C
      WRITE (OKEY,877)
877   FORMAT(1X,'CGSO')
      WRITE (OKEY,878) cgso
878   FORMAT (1X,1E9.3)
C
      WRITE (OKEY,977)
```

```
977     FORMAT(1X, 'CGBO')
        WRITE (OKEY,978) cgbo
978     FORMAT (1x,1E9.3)
C
        WRITE (OKEY,107)
107     FORMAT(1X, 'RMSSUB')
        WRITE (OKEY,108) rmssub
108     FORMAT (1x,F5.3)
C
131     CONTINUE
        END
C
C
        SUBROUTINE LNFIT (N,M,X,Y,SLOPE,XINT,RMS)
C
C
        DIMENSION X(1),Y(1)
        DIMENSION YINT(101),YCALC(101)
C
        I=0
        SUMX=0
        SUMY=0
        SUMXY=0
        SUMXX=0
C
        DO 10 I=M,M+N-1
C         write (6,*) 'X=',X(I),'Y=',Y(I)
        SUMX=SUMX+X(I)
        SUMY=SUMY+Y(I)
        SUMXY=SUMXY+X(I)*Y(I)
        SUMXX=SUMXX+X(I)*X(I)
10      CONTINUE
C
        ANUMS=N*SUMXY-SUMX*SUMY
C         write (6,*) 'anums=',anums
        ANUMI=SUMXX*SUMY-SUMX*SUMXY
C         write (6,*) 'anumi=',anumi
        DEN=N*SUMXX-SUMX*SUMX
C         write (6,*) 'den=',den
        SLOPE=ANUMS/DEN
C         write (6,*) 'slope=',slope
        YINT(M)=ANUMI/DEN
        XINT=-ANUMI/ANUMS
C
C       DETERMINE THE RMS DEVIATION IN THE Y DIRECTION
C
        J=0
        SQU=0
        DO 20 J=M,M+N-1
        YCALC(J)=SLOPE*X(J)+YINT(M)
        SQU=SQU+(Y(J)-YCALC(J))**2
20      CONTINUE
```

```
C
       RMS=SQRT(SQU/N)
C
       RETURN
       END
C
C-----------------------------------------------------------------
C
      subroutine gain(Id,Vg,icount,NDER,Vds,beta,vt)
C
      real Id(250),Vg(250),Vds,beta,vt
      integer icount,nu,np,NDER
C
      slopem=0.0
      slopen=0.0
      np=NDER/2
C
      do 1000 I=np+1,icount-np
      x=0.0
      y=0.0
      xy=0.0
      xsq=0.0
      nu=0
      iflag=1                            ! n-channel
      if (vg(2).lt.vg(1)) iflag=-1       ! Therefore p-channel
C
      do 391 MO=I-np,I+np
      x=x+Vg(MO)
      y=y+Id(MO)*1.0E+06
      xy=xy+Id(MO)*Vg(MO)*1.0E+06
      xsq=xsq+Vg(MO)*Vg(MO)
      nu=nu+1
  391 continue
C
      delta=nu*xsq-x*x
      if (delta .eq. 0.0) go to 360
      slopen=(nu*xy-x*y)/delta
      if (iflag .eq. 1 .and. slopen .gt. slopem) go to 999
      if (iflag .eq. -1 .and. slopen .lt. slopem) go to 999
      go to 1000
  999 slopem=slopen                            ! slopem = max slope
      yintc=(xsq*y-x*xy)/delta
      xintc=-yintc/slopem
      if (Vds .gt. .90) go to 1000
      Vt=xintc-(Vds/2.0)
 1000 continue
C
      beta=slopem/Vds
      return
  360 write(6,*) ' *** Delta equals 0 ***'
      stop
      end
```

CAPKEY.FOR

```
C
C       FILENAME:  CAPKEY.FOR
C
C       This program recalculates the capacitances given the
C       optimized parameters from SUXES.
C
C
        open(unit=20, file='guest.dat', status='old')
        open(unit=40, file='expar.dat', status='old')
        open(unit=60, file='parspice.dat', status='unknown')
C
        EOX = 34.5E-12                  ! in F/m
C
C       READ THE PRESUXES DATA
C
        KEY = 20
        k=0
        do 100 k=1,25
        read (key,377,end=331) dummy
377     format (A1)
100     continue
C
        READ (KEY,*,END=331) DRMSSUB
C
C       READ THE AFTSUXES DATA
C
331     IKEY = 40
        READ (IKEY,*,END=231) XLD,trash1,trash2
        READ (IKEY,*,END=231) TOX,trash1,trash2
        READ (IKEY,*,END=231) XNSUB,trash1,trash2
        READ (IKEY,*,END=231) VTO,trash1,trash2
        READ (IKEY,*,END=231) XKP,trash1,trash2
        XKP = XKP*1.0E+6
        READ (IKEY,*,END=231) GAMMA,trash1,trash2
        READ (IKEY,*,END=231) PHI,trash1,trash2
        READ (IKEY,*,END=231) UO,trash1,trash2
        READ (IKEY,*,END=231) DELTA,trash1,trash2
        READ (IKEY,*,END=231) VMAX,trash1,trash2
        READ (IKEY,*,END=231) XJ,trash1,trash2
        READ (IKEY,*,END=231) XKAPPA,trash1,trash2
        READ (IKEY,*,END=231) XNFS,trash1,trash2
        READ (IKEY,*,END=231) ETA,trash1,trash2
        READ (IKEY,*,END=231) THETA,trash1,trash2
C
C       RECALCULATE THE CAPACITANCES
C
        cgdo = eox*xld*1.0e-6/(tox*1.0e-10)
        cgso = cgdo
        cgbo = eox*4.*1.0e-6/(2.*tox*1.0e-10)
        ATOX = TOX*1e-4
```

```
C
C      WRITE THE PERTINENT DATA INTO PARSPICE.DAT
C
231    MKEY = 60
       WRITE (MKEY,60) XLD
       WRITE (MKEY,69) ATOX
       WRITE (MKEY,62) XNSUB
       WRITE (MKEY,63) VTO
       WRITE (MKEY,64) XKP
       WRITE (MKEY,60) GAMMA
       WRITE (MKEY,60) PHI
       WRITE (MKEY,61) UO
       WRITE (MKEY,65) DELTA
       WRITE (MKEY,66) VMAX
       WRITE (MKEY,60) XJ
       WRITE (MKEY,61) XKAPPA
       WRITE (MKEY,62) XNFS
       WRITE (MKEY,67) ETA
       WRITE (MKEY,67) THETA
       WRITE (MKEY,68) CGDO
       WRITE (MKEY,68) CGSO
       WRITE (MKEY,68) CGBO
       WRITE (MKEY,60) DRMSSUB
60     FORMAT (1x,F5.3)
61     FORMAT (1x,F6.1)
62     FORMAT (1x,1E10.3)
63     FORMAT (1x,F6.3)
64     FORMAT (1x,F6.2)
65     FORMAT (1x,F8.3)
66     FORMAT (1x,F9.0)
67     FORMAT (1x,F8.4)
68     FORMAT (1x,1E9.3)
69     FORMAT (1x,F6.4)
       CONTINUE
       END
```

IVKEY.FOR

```
C
C       FILENAME:  IVKEY.FOR
C
C       PURPOSE:  THIS PROGRAM MASSAGES THE DATA FROM SUXES AND SPICE SO
C                 IT IS SUITABLE FOR THE PLOTTING PACKAGE
C
C
        DIMENSION VDS(1000), VGS(1000), VBS(1000)
        DIMENSION CIDS(1000), CNVBS(10)
        DIMENSION VD(1000), CID(1000)
        open(unit=10, file='in.dat', status='old')
        open(unit=20, file='in2.dat', status='old')
        open(unit=30, file='in3.dat', status='old')
        open(unit=40, file='in4.dat', status='old')
        open(unit=11, file='out.dat', status='unknown')
        open(unit=21, file='out2.dat', status='unknown')
        open(unit=31, file='out3.dat', status='unknown')
        open(unit=41, file='out4.dat', status='unknown')
        open(unit=42, file='out5.dat', status='unknown')
        open(unit=43, file='out6.dat', status='unknown')
        open(unit=44, file='out7.dat', status='unknown')
        open(unit=45, file='out8.dat', status='unknown')
        open(unit=46, file='out9.dat', status='unknown')
C
C       READ THE DATA
C
        JKEY=0
600     JKEY=JKEY+10
        J=0
        DO 378 J=1,3
        READ (JKEY,377)IDUMMY
377     FORMAT (A1)
378     CONTINUE
        IF (JKEY .LT. 30) GO TO 600
        READ (40,377) IDUMMY
        READ (40,*) N
        N=N/6
C
C       MASSAGE THE DATA FILES AND THE FILES FROM SUXES
C
        IKEY=0
500     L=0
        IKEY=IKEY+10
        I=0
230     I=I+1
        READ (IKEY,*,END=231) VDS(I), VGS(I), VBS(I), CIDS(I)
        AN=0.0
100     AN=AN+1.0
        IF (AN .GT. 5.5) GO TO 230
        IF (ABS(VGS(I)) .LT. ((AN - 1.0)+.99)) GO TO 230
```

```
      IF (ABS(VGS(I)) .GT. (AN + .01)) GO TO 100
C
C     CORRECT VGS VALUE FOUND
C
      IF (ABS(VDS(I)) .LT. (.09)) GO TO 230
      IF (ABS(VDS(I)) .LT. (.11)) GO TO 777
      IF (ABS(VDS(I)) .LT. (.29)) GO TO 230
      IF (ABS(VDS(I)) .LT. (.31)) GO TO 777
      PF=0.0
240   PF=PF+0.5
      IF (PF .GT. 5.1) GO TO 230
      IF (ABS(VDS(I)) .LT. (PF-.005)) GO TO 230
      IF (ABS(VDS(I)) .LT. (PF+.005)) GO TO 777
      GO TO 240
C
C     CORRECT VDS VALUE FOUND
C
777   L=L+1
      A=1.0
      VD(L)=VDS(I)
      IF (VD(L) .LT. 0.0) A=-1.0
      CID(L)=A*CIDS(I)
      GO TO 230
231   WRITE(IKEY+1,4) L
4     FORMAT(5X,I5)
      K=0
      DO 24000 K=1,L
      WRITE (IKEY+1,*) VD(K), CID(K)
24000 CONTINUE
      IF (IKEY .LT. 30) GO TO 500
C
C     CONSIDER THE PLOT FILES FROM SPICE
C
440   I=0
      J=0
330   J=J+1
      IF (J .GE. 7) GO TO 331
      WRITE (40+J,*) N
550   I=I+1
      READ (40,*,END=331) VDS(I), CIDS(I)
      WRITE (40+J,*) VDS(I), CIDS(I)
      IF (I .EQ. J*N) GO TO 330
      GO TO 550
331   CONTINUE
      END
```

SUMKEY.FOR

```
C*****************************************************************************
C*****************************************************************************
C
C       FILENAME:  SUMKEY.FOR
C
C       PURPOSE:  THIS PROGRAM ADDS TO THE 'RESULTS.DAT' FILE AND CALCULATES
C                 THE AVERAGES.
C
C
        dimension chip(50), wave(50), wsum(50)
        dimension alave(50), alsum(50)
C
        common /plomi/ MARK, IWAF(100), lotnum1, lotnum2
        common map(10,10)
C
        character*1 IWAF
        character*1 IW
        character*1 IWAFCUR
        character*1 MARK
        character*4 lotnum1
        character*4 lotnum2
C
        open(unit=10, file='map1.DAT', status='unknown')
        open(unit=20, name='in.dat', status='old')
        open(unit=24, file='nlw.fil', status='old')
        open(unit=25, file='line.fil', status='old')
        open(unit=30, name='out.dat', status='unknown')
        open(unit=49, file='wp.dat', status='unknown')
        open(unit=50, file='lp.dat', status='unknown')
        open(unit=51, file='w.dat', status='unknown')
        open(unit=52, file='l.dat', status='unknown')
        open(unit=53, file='mckt.dat', status='unknown')
        open(unit=54, file='sckt.dat', status='unknown')
        open(unit=55, file='pdiff.dat', status='unknown')
        open(unit=56, file='nvto.dat', status='unknown')
        open(unit=57, file='nkp.dat', status='unknown')
        open(unit=58, file='ntox.dat', status='unknown')
        open(unit=59, file='nld.dat', status='unknown')
        open(unit=60, file='nsub.dat', status='unknown')
        open(unit=61, file='nrms.dat', status='unknown')
        open(unit=62, file='nphi.dat', status='unknown')
        open(unit=63, file='ngamma.dat', status='unknown')
        open(unit=64, file='nxj.dat', status='unknown')
        open(unit=65, file='ncgdo.dat', status='unknown')
        open(unit=66, file='ncgso.dat', status='unknown')
        open(unit=67, file='ncgbo.dat', status='unknown')
        open(unit=68, file='nerr.dat', status='unknown')
        open(unit=69, file='ntheta.dat', status='unknown')
        open(unit=70, file='nkappa.dat', status='unknown')
        open(unit=71, file='ndelta.dat', status='unknown')
```

```
      open(unit=72, file='neta.dat', status='unknown')
      open(unit=73, file='nu0.dat', status='unknown')
      open(unit=74, file='nvmax.dat', status='unknown')
      open(unit=75, file='nnfs.dat', status='unknown')
      open(unit=76, file='pvto.dat', status='unknown')
      open(unit=77, file='pkp.dat', status='unknown')
      open(unit=78, file='ptox.dat', status='unknown')
      open(unit=79, file='pld.dat', status='unknown')
      open(unit=80, file='psub.dat', status='unknown')
      open(unit=81, file='prms.dat', status='unknown')
      open(unit=82, file='pphi.dat', status='unknown')
      open(unit=83, file='pgamma.dat', status='unknown')
      open(unit=84, file='pxj.dat', status='unknown')
      open(unit=85, file='pcgdo.dat', status='unknown')
      open(unit=86, file='pcgso.dat', status='unknown')
      open(unit=87, file='pcgbo.dat', status='unknown')
      open(unit=88, file='perr.dat', status='unknown')
      open(unit=89, file='ptheta.dat', status='unknown')
      open(unit=90, file='pkappa.dat', status='unknown')
      open(unit=91, file='pdelta.dat', status='unknown')
      open(unit=92, file='peta.dat', status='unknown')
      open(unit=93, file='pu0.dat', status='unknown')
      open(unit=94, file='pvmax.dat', status='unknown')
      open(unit=95, file='pnfs.dat', status='unknown')
      open(unit=96, file='switch2.dat', status='unknown')
      open(unit=97, file='pchips.fil', status='old')
      open(unit=98, file='lotnumber.fil', status='old')
C
      bust=0.0                    ! the program's doing fine
      idump=0                     ! the number of zero locations
      imark=1                     ! the last MARK
      irow=0                      ! the row number
      icol=0                      ! the column number
      jrow=1
      jcol=4
      nodata=0                    ! no more data or error reading data
      zfil=0.0                    ! new chip AND wafer ?
C
      write (10,342)
342   format(1x,'AAAAAA')
      write (10,342)
C
      read (24,327) IWAFCUR
327   format (A1)
C
      CALL INTRO(chip,wave,wsum,alave,alsum,perchip,itotchip)
C
C-----------------------------------------------------------------------
C
C     READ THE DATA
C
      lot=0                       ! the number of wafers
```

```
      N=0                          ! the number of acceptable chips
      I=0                          ! the total number of chips
500   I=I+1
501   krow=irow
      kcol=icol
      read (20,477,ERR=700,END=700) MARK
      if (MARK .eq. '#') go to 507                 ! new data
      if (MARK .eq. '*') go to 588                 ! average titles
      READ (20,66,ERR=700,END=700)IWAF(I),IROW,ICOL,CHIP(1),CHIP(2),
     + CHIP(46), CHIP(47)
      READ (20,67,ERR=700,END=700) CHIP(3),CHIP(4),CHIP(5)
      READ (20,477) IDUMMY
C
C     N- or P-channel or both ?
C
      if (I .le. 1) go to 503
      M=I
      if (iwaf(I) .ne. iwaf(I-1)) go to 937
511   if ((iwaf(I) .ne. iwaf(I-1)) .and. (N .eq. 0)) I=0
      if (iwaf(I) .ne. iwaf(I-1)) CALL WAFER(imark,n,i,chip,
     + wave,wsum,alave,alsum,nodata)
      if(i .eq. 0) i=i+1
      iwaf(I) = iwaf(M)
503   read (25,69,end=502)iw,ir,ic,xndum,yndum,xpdum,ypdum,zdum,idev
      if (iwaf(I) .ne. iw) go to 503
      if ((ir .eq. irow) .and. (ic .eq. icol)) go to 505
      go to 503
502   write (6,*) ' The correct chip was not located in LINE.FIL-1'
      go to 909
505   inc=0
      if(idev .eq. 2) inc=20                       ! just p-channel
C
506   read (20,*,end=700) chip(15+inc),chip(16+inc),
     + chip(17+inc),chip(11+inc)
      read (20,*,end=700) chip(9+inc),chip(8+inc),chip(10+inc),
     + chip(6+inc),chip(7+inc)
      read (20,*,end=700) chip(13+inc),chip(12+inc),chip(23+inc),
     + chip(21+inc),chip(24+inc)
      READ (20,*,END=700) chip(14+inc),chip(20+inc),chip(25+inc),
     + chip(22+inc),chip(19+inc),chip(18+inc)
      if ((idev .eq. 1) .or. (inc .ge. 20)) go to 920   ! just n-channel
      read (20,477) IDUMMY
      inc = 20
      go to 506
C
507   if (mark .eq. '#') CALL NEWCHIP(n,i,iw,irow,icol,chip,
     + wave,wsum,alave,alsum,idev,bust,nodata,imark,zfil)
      if (zfil .ne. 0.0) go to 937
      if (BUST .eq. 1.0) go to 937
C------------------------------------------------------------------------
C
C     WRITE THE DATA INTO OUT.DAT
```

```
C
920    zfil=0.0
       if (mark .eq. '-') imark=-1
       if (mark .eq. '+') imark=1
       write (30,477) MARK
       WRITE (30,41) IWAF(I),IROW,ICOL,chip(1),chip(2),chip(46),chip(47)
       WRITE (30,67) chip(3),chip(4),chip(5)
       WRITE (30,*) ' '
       inc=0
       if (idev .eq. 2) inc=20
538    write (30,42) chip(15+inc),chip(16+inc),chip(17+inc),chip(11+inc)
       write (30,43) chip(9+inc),chip(8+inc),chip(10+inc),
     + chip(6+inc),chip(7+inc)
       WRITE (30,44) chip(13+inc),chip(12+inc),chip(23+inc),
     + chip(21+inc),chip(24+inc)
       WRITE (30,45) chip(14+inc),chip(20+inc),chip(25+inc),
     + chip(22+inc),chip(19+inc),chip(18+inc)
       if ((idev .eq. 1) .or. (inc .ge. 20)) go to 283
       write (30,47)
       inc=20
       go to 538
477    format(1x, A1)
C      For the row and column - row #1
41     format(5x,A1,1x,I3,I4,1x,F5.2,6x,F5.2,6x,F5.2,6x,F5.2)
C      For row #2
42     format(15x,1e10.3,1x,1e10.3,1x,1e10.3,2x,F5.3)
C      For row #3
43     format(16x,F5.3,3x,F6.1,7x,1e10.3,1x,F6.3,2x,F6.2)
C      For the SUXES parameters - row #4
44     FORMAT (16X,F5.3,6x,F5.3,3x,F6.1,5x,F8.3,3x,1e10.3)
C      For the last row
45     format (16x, F5.3,3x,F6.1,7x,1e10.3,F8.4,F8.3,4x,F6.2)
C      Similar to 41
46     format (5x,A1,2x, F5.2, 2x, F5.2, 6x,F5.2,6x,F5.2,5x,F6.2)
47     format (1x,' ')
49     format (8x, F5.2, 2x, F5.2, 6x,F5.2,6x,F5.2,5x,F6.2)
66     format(5x,A1,1x,I3,I4,1x,F5.2,6x,F5.2,6x,F5.2,6x,F5.2)
67     format(15x,F5.2,6x,F5.2,5x,F6.2)
69     format(2x,A1,2x,I2,1x,I2,1x,F5.2,1x,F5.2,1x,F5.2,1x,
     + F5.2,1x,F5.2,1x,I3)
C---------------------------------------------------------------------------
C
C      ignore the averages from last time
C
588    IO=0
       do 578 IO=1,24
       read (20,477) IDUMMY
578    continue
       go to 501
C---------------------------------------------------------------------------
700    nodata=1
       GO TO 707
```

```
283    if (MARK .eq. '-') go to 537
C
708    if (MARK .eq. '+') N=N+1
       NM=N
707    CALL WAFER(imark,n,i,chip,wave,wsum,alave,alsum,nodata)
       if (nodata .eq. 1) go to 937
C-----------------------------------------------------------------------
C
C      fill the wafer map files
C
C      correct row and column ?
C
537    if (IWAFCUR .NE. iw) GO TO 168
       if (map(jrow,jcol) .eq. map(irow,icol)) go to 184
       idump=map(irow,icol)-map(jrow,jcol)
C
C   fill with zeros
C
       jfill=1
182    ifill=0
       zero=0.0
       do 183 ifill=1,idump
       izer=0
       do 186 izer=1,47
       write(48+izer,189) zero
186    continue
183    continue
       if(jfill .eq. 0) go to 168
C
184    jfill=0
       if (mark .eq. '+') go to 187
       idump=1
       go to 182
C
C-----------------------------------------------------------------------
187    INC=0
       DO 600 INC=1,45
       write(50+inc,189) CHIP(INC)
189    format (1x,1e10.3)
600    CONTINUE
       write (49,189) chip(46)
       write (50,189) chip(47)
C
C-----------------------------------------------------------------------
C
C   determine the next jrow and jcol
168    jrow=irow
       jcol=icol
       if((jrow .eq. 1) .and. (jcol .eq. 6)) go to 170
       if(((jrow.eq.2).or.(jrow.eq.8)) .and. (jcol .eq. 8)) go to 171
       if (jcol .eq. 9) go to 172
       jcol=jcol+1
```

```
      go to 500
170   jrow=jrow+1
      jcol=2
      go to 500
171   if (jrow .eq. 2) jcol=1
      if (jrow .eq. 8) jcol=4
      jrow=jrow+1
      go to 500
172   jcol=1
      if (jrow .eq. 7) jcol=2
      jrow=jrow+1
      GO TO 500
C-------------------------------------------------------------------------------
C
C     fill the rest of the data files
C
937   IF (IWAFCUR .NE. iw) GO TO 427
      zero=0.0
      idump=map(9,6)-map(krow,kcol)
      kfill=0
      do 910 kfill=1,idump
      ink=0
      do 908 ink=1,47
      write(48+ink,189) zero
      if(kfill .ne. idump) go to 908
      write(48+ink,281)
281   format(' *')
908   continue
910   continue
427   jrow=1
      jcol=4
      if(zfil .ne. 0.0) go to 920
      if ((nodata .eq. 0) .and. (BUST .ne. 1.0)) go to 511
C-------------------------------------------------------------------------------
C
C     Will wafer maps be generated ?
C
909   if ((NM*100./itotchip) .gt. perchip) go to 810
      write(96,807)             !  No maps
807   format('$ K=0')
      go to 911
C
810   write(96,808)             !  Generate maps
808   format('$ K=1')
C
911   continue
      END
C*******************************************************************************
C*******************************************************************************
C
C     FILENAME:  INTRO.SUB
C
```

```
C
      SUBROUTINE INTRO (chip,wave,wsum,alave,alsum,perchip,
     + itotchip)
C
      dimension chip(1), wave(1), wsum(1)
      dimension alave(1), alsum(1)
C
      common /plomi/ MARK, IWAF(100), lotnum1, lotnum2
      common map(10,10)
C
      character*1 IWAF
      character*1 IW
      character*1 MARK
      character*4 lotnum1
      character*4 lotnum2
C
C---------------------------------------------------------------------
C
C     FILL THE MAP DIMENSION TO NUMBER THE CHIPS
C
      jmap=0
      do 160 jmap=4,6
      map(1,jmap)=jmap-3
      map(9,jmap)=jmap+59
160   continue
      jmap=0
      do 161 jmap=2,8
      map(2,jmap)=jmap+2
      map(8,jmap)=jmap+54
161   continue
      jmap=0
      do 162 jmap=1,9
      map(3,jmap)=jmap+10
      map(4,jmap)=jmap+19
      map(5,jmap)=jmap+28
      map(6,jmap)=jmap+37
      map(7,jmap)=jmap+46
162   continue
C---------------------------------------------------------------------
      read (97,477) IDUMMY
      read (97,*) perchip, itotchip
C
C     READ THE DUMMY DATA
C
      MO=0
      DO 378 MO=1,13
      READ (20,477)IDUMMY
477   format (1x, A1)
378   CONTINUE
C
      mo=0
      do 478 mo=1,5
```

```
      read (25,477) IDUMMY
478   continue
C
      read (98,734) lotnum1, lotnum2
734   format(2A4)
C
      ink=0
      do 908 ink=1,47
      write(48+ink,281)
281   format(' *')
908   continue
C
C--------------------------------------------------------------------------------
C
C     WRITE THE HEADING
C
      WRITE (30,1)
1     format(' #')
      WRITE (30,2)
2     format(' *  FILENAME = RESULTS.DAT')
      WRITE (30,4)
      WRITE (30,3) lotnum1, lotnum2
3     format(' *  Lot Number = ',1x,2A4)
      WRITE (30,4)
4     format(' *')
      WRITE (30,5)
5     format(' *  W1   R   C   Wn        Ln          Wp ',
     +'       Lp')
      WRITE (30,1006)
1006  format (' *             MCKT      SCKT        PDIFF')
      WRITE (30,*) '*'
      WRITE (30,6)
6     format (' *             CGDO      CGSO        CGBO',
     +'      RMS  ')
      WRITE (30,7)
7     format (' *              LD        TOX        SUB ',
     +'      VTO       KP')
      WRITE (30,8)
8     format(' *             GAMMA      PHI         UO ',
     +'      DELTA   VMAX')
      WRITE (30,9)
9     format(' *              XJ       KAPPA       NFS ',
     +'      ETA      THETA       ERR')
      write(30,666)
666   format(' ***************************************************',
     + '**********************')
C--------------------------------------------------------------------------------
C
C     INITIALIZE THE DIMENSIONS
C
      INC=0
      DO 100 INC=1,47
```

```
      CHIP(INC)=0.0
      WAVE(INC)=0.0
      WSUM(INC)=0.0
      ALAVE(INC)=0.0
      ALSUM(INC)=0.0
100   CONTINUE
C--------------------------------------------------------------------------------
      RETURN
      END
C********************************************************************************
C********************************************************************************
C
C     FILENAME:  NEWCHIP.SUB
C
C
      SUBROUTINE NEWCHIP(N,I,IW,IROW,ICOL,CHIP,WAVE,WSUM,
     + ALAVE,ALSUM,IDEV,BUST,NODATA,IMARK,ZFIL)
C
      dimension chip(1)
      dimension wave(1), wsum(1), alave(1), alsum(1)
      common /plomi/ MARK, IWAF(100), lotnum1, lotnum2
C
      character*1 IWAF
      character*1 IW
      character*1 MARK
      character*4 lotnum1
      character*4 lotnum2
C
C--------------------------------------------------------------------------------
C
C     READ THE NEW DATA.....IS IT GOOD OR BAD ? (- = bad)
C
      read(20,77) IWAF(I)
77    format(A1)
      READ (20,*,END=512) IROW,ICOL,CHIP(1),CHIP(2),CHIP(46),CHIP(47)
      READ (20,*,END=512) CHIP(3),CHIP(4)
      M=I
      if(I .le. 1) go to 513
      if(IWAF(I) .ne. iwaf(I-1)) zfil=1.0
      if((IWAF(I) .ne. iwaf(I-1)) .and. (N .eq. 0)) I=0
      if(IWAF(I) .ne. iwaf(I-1)) CALL WAFER(imark,n,i,chip,
     + wave,wsum,alave,alsum,nodata)
C
C     DOES IT CONTAIN N- or P-channel DATA or both ?
C
513   read (25,69,end=512)iw,ir,ic,xndum,yndum,xpdum,ypdum,zdum,idev
69    format(2x,A1,2x,I2,1x,I2,1x,F5.2,1x,F5.2,1x,F5.2,
     + 1x,F5.2,1x,F5.2,1x,I3)
      if(I .eq. 0) i=i+1
      iwaf(I)=iwaf(m)
C     write (30,69)iw,ir,ic,xndum,yndum,zdum,idev
C     write (30,68) mark, iwaf(I),i
```

```
C68    format(1x, A1, 1x, A1,1x,I2)
       if (iwaf(I) .ne. iw) go to 513
       if ((ir .eq. irow) .and. (ic .eq. icol)) go to 515
       go to 513
512    write (6,*) ' The correct chip was not located in LINE.FIL-2'
       BUST=1.0
       RETURN
515    inc=0
       if(idev .eq. 2) inc=20                        ! just p-channel
C
516    read (20,*,end=512) chip(9+inc),chip(8+inc),chip(10+inc),
     + chip(6+inc),chip(7+inc)
       chip(8+inc)=chip(8+inc)*1.e4
       read (20,*,end=512) chip(13+inc),chip(12+inc),chip(23+inc),
     + chip(21+inc),chip(24+inc)
       read (20,*,end=512) chip(14+inc),chip(20+inc),chip(25+inc),
     + chip(22+inc),chip(19+inc)
       READ (20,*,END=512) chip(15+inc),chip(16+inc),chip(17+inc),
     + chip(11+inc),trash,chip(18+inc),trash
       if ((idev .eq. 1) .or. (inc .ge. 20)) go to 517   ! just n-channel
       inc=20
       go to 516
C
C-------------------------------------------------------------------------------
C
517    IFLG=0
       IF (chip(3) .EQ. 0.0) IFLG=1                  ! 3 = MCKT
       if (IFLG .GE. 1) go to 607
       chip(5) = 100.*(ABS(chip(3) - chip(4)))/chip(3)   ! 4 = SCKT
C
       if ((chip(5) .LT. 0.0) .OR. (chip(5) .GT. 100.0)) IFLG=2
       if (IFLG .GE. 1) go to 607                    ! 5 = PDIFF
C
       if (idev .eq. 2) go to 556      ! just p-channel  6 = NVTO
       if ((chip(6) .LE. 0.3) .OR. (chip(6) .GE. 1.5)) IFLG=3
       if (IFLG .GE. 1) go to 607                    ! 7 = NKP
       if ((chip(7)  .LE. 1.0) .OR. (chip(7) .GE. 100.0)) IFLG=4
       if (IFLG .GE. 1) go to 607                    ! 8 = NTOX
       if ((chip(8) .LE. 250.0) .OR. (chip(8) .GE. 1000.0)) IFLG=5
       if (IFLG .GE. 1) go to 607                    ! 9 = NLD
       if ((chip(9) .LE. 0.0) .OR. (chip(9) .GE. 5.0)) IFLG=6
       if (IFLG .GE. 1) go to 607                    ! 10 = NSUB
       if ((chip(10) .LE. 1.0E13) .OR. (chip(10) .GE. 1.0E17)) IFLG=7
       if (IFLG .GE. 1) go to 607                    ! 11 = NRMS
       if ((chip(11) .LT. 0.0) .OR. (chip(11) .GT. 0.20)) IFLG=7
       if (IFLG .GE. 1) go to 607                    ! 12 = NPHI
C
       if ((chip(12) .LE. 0.3) .OR. (chip(12) .GE. 0.9)) IFLG=8
       if (IFLG .GE. 1) go to 607                    ! 13 = NGAMMA
       if ((chip(13) .LE. 0.1) .OR. (chip(13) .GE. 3.0)) IFLG=9
       if (IFLG .GE. 1) go to 607                    ! 14 = NXJ
       if ((chip(14) .LE. 0.0) .OR. (chip(14) .GE. 1.5)) IFLG=10
```

```
      if (IFLG .GE. 1) go to 607                    ! 15 = NCGDO
      if ((chip(15) .LE. 0.0) .OR. (chip(15) .GE. 1.0E-7)) IFLG=11
      if (IFLG .GE. 1) go to 607                    ! 16 = NCGSO
      if ((chip(16) .LE. 0.0) .OR. (chip(16) .GE. 1.0E-7)) IFLG=12
      if (IFLG .GE. 1) go to 607                    ! 17 = NCGBO
      if ((chip(17) .LE. 0.0) .OR. (chip(17) .GE. 1.0E-7)) IFLG=13
      if (IFLG .GE. 1) go to 607                    ! 18 = NERR
      if ((chip(18) .LT. 0.0) .OR. (chip(18) .GT. 10.0)) IFLG=13
      if (IFLG .GE. 1) go to 607                    ! 19 = NTHETA
c
      if ((chip(19) .LE. -2.0) .OR. (chip(19) .GE. 2.0)) IFLG=14
      if (IFLG .GE. 1) go to 607                    ! 20 = NKAPPA
      if ((chip(20) .LE. 0.0) .OR. (chip(20) .GE. 999.0)) IFLG=15
      if (IFLG .GE. 1) go to 607                    ! 21 = NDELTA
      if ((chip(21) .LE. -100.0) .OR. (chip(21) .GE. 100.0)) IFLG=16
      if (IFLG .GE. 1) go to 607                    ! 22 = NETA
      if ((chip(22) .LE. -2.0) .OR. (chip(22) .GE. 2.0)) IFLG=17
      if (IFLG .GE. 1) go to 607                    ! 23 = NUO
      if ((chip(23) .LE. 0.1) .OR. (chip(23) .GE. 1250.0)) IFLG=18
      if (IFLG .GE. 1) go to 607                    ! 24 = NVMAX
      if ((chip(24) .LE. 1.0e4) .OR. (chip(24) .GE. 9.9e7)) IFLG=19
      if (IFLG .GE. 1) go to 607                    ! 25 = NNFS
      if ((chip(25) .LE. 0.0) .OR. (chip(25) .GE. 4.0e14)) IFLG=20
      if (IFLG .GE. 1) go to 607                    ! 26 = PVTO
c
      if (idev .eq. 1) go to 557                    ! just n-channel
556   if ((chip(26) .LE. -3.0) .OR. (chip(26) .GE. -0.1)) IFLG=21
      if (IFLG .GE. 1) go to 607                    ! 27 = PKP
      if ((chip(27) .LE. 1.0) .OR. (chip(27) .GE. 100.0)) IFLG=22
      if (IFLG .GE. 1) go to 607                    ! 28 = PTOX
      if ((chip(28) .LE. 250.0) .OR. (chip(28) .GE. 1000.0)) IFLG=23
      if (IFLG .GE. 1) go to 607                    ! 29 = PLD
      if ((chip(29) .LE. 0.0) .OR. (chip(29) .GE. 5.0)) IFLG=24
      if (IFLG .GE. 1) go to 607                    ! 30 = PSUB
      if ((chip(30) .LE. 1.0E13) .OR. (chip(30) .GE. 1.0E17)) IFLG=25
      if (IFLG .GE. 1) go to 607                    ! 31 = PRMS
      if ((chip(31) .LT. 0.0) .OR. (chip(31) .GT. 0.20)) IFLG=25
      if (IFLG .GE. 1) go to 607                    ! 32 = PPHI
c
      if ((chip(32) .LE. 0.3) .OR. (chip(32) .GE. 0.9)) IFLG=26
      if (IFLG .GE. 1) go to 607                    ! 33 = PGAMMA
      if ((chip(33) .LE. 0.1) .OR. (chip(33) .GE. 3.0)) IFLG=27
      if (IFLG .GE. 1) go to 607                    ! 34 = PXJ
      if ((chip(34) .LE. 0.0) .OR. (chip(34) .GE. 1.5)) IFLG=28
      if (IFLG .GE. 1) go to 607                    ! 35 = PCGDO
      if ((chip(35) .LE. 0.0) .OR. (chip(35) .GE. 1.0E-7)) IFLG=29
      if (IFLG .GE. 1) go to 607                    ! 36 = PCGSO
      if ((chip(36) .LE. 0.0) .OR. (chip(36) .GE. 1.0E-7)) IFLG=30
      if (IFLG .GE. 1) go to 607                    ! 37 = PCGBO
      if ((chip(37) .LE. 0.0) .OR. (chip(37) .GE. 1.0E-7)) IFLG=31
      if (IFLG .GE. 1) go to 607                    ! 38 = PERR
      if ((chip(38) .LT. 0.0) .OR. (chip(38) .GT. 10.0)) IFLG=31
```

```
      if (IFLG .GE. 1) go to 607                    ! 39 = PTHETA
C
      if ((chip(39) .LE. -10.0) .OR. (chip(39) .GE. 10.0)) IFLG=32
      if (IFLG .GE. 1) go to 607                    ! 40 = PKAPPA
      if ((chip(40) .LE. 0.0) .OR. (chip(40) .GE. 999.0)) IFLG=33
      if (IFLG .GE. 1) go to 607                    ! 41 = PDELTA
      if ((chip(41) .LE. -100.0) .OR. (chip(41) .GE. 100.0)) IFLG=34
      if (IFLG .GE. 1) go to 607                    ! 42 = PETA
      if ((chip(42) .LE. -10.0) .OR. (chip(42) .GE. 10.0)) IFLG=35
      if (IFLG .GE. 1) go to 607                    ! 43 = PUO
      if ((chip(43) .LE. 0.1) .OR. (chip(43) .GE. 1250.0)) IFLG=36
      if (IFLG .GE. 1) go to 607                    ! 44 = PVMAX
      if ((chip(44) .LE. 1.0e4) .OR. (chip(44) .GE. 9.9e7)) IFLG=37
      if (IFLG .GE. 1) go to 607                    ! 45 = PNFS
      if ((chip(45) .LE. 1.0e10) .OR. (chip(45) .GE. 1.0e14)) IFLG=38
      if (IFLG .GE. 1) go to 607
C
C-----------------------------------------------------------------------
C
557   MARK = '+'
      go to 968
607   MARK = '-'
C      write (30,*) 'IFLG=', IFLG
C-----------------------------------------------------------------------
968   continue
C     write(30,9999) MARK
C9999 format(1x,A1)
      RETURN
      END
C***********************************************************************
C***********************************************************************
C
C     FILENAME:  WAFER.SUB
C
C
      SUBROUTINE WAFER (IMARK,N,I,CHIP,WAVE,WSUM,ALAVE,
     + ALSUM,NODATA)
C
      dimension chip(1), wave(1), wsum(1), alave(1), alsum(1)
      common /plomi/ MARK, IWAF(100), lotnum1, lotnum2
C
      character*1 IWAF
      character*1 MARK
      character*4 lotnum1
      character*4 lotnum2
C-----------------------------------------------------------------------
C
C     DETERMINE THE WAFER AVERAGES
C
      IF (N .EQ. 0) RETURN
      IF ((NODATA .EQ. 1) .and. (I .LE. 2)) RETURN
      IF (NODATA .EQ. 1) GO TO 791
```

```
      IF (I .EQ. 1) GO TO 201
      IF (IWAF(I) .NE. IWAF(I-1)) GO TO 309
201   if (imark .le. 0) go to 309
      J=0
      DO 200 J=1,47
      WSUM(J)=WSUM(J)+CHIP(J)
      WAVE(J)=WSUM(J)/N
200   CONTINUE
C
C-----------------------------------------------------------------------------
309   if (I .EQ. 1) RETURN
      if (IWAF(I) .eq. IWAF(I-1)) RETURN
791   WRITE (30,4)
      WRITE (30,*) '*** WAFER AVERAGES ***'
      WRITE (30,4)
      WRITE (30,4)
      WRITE (30,*) '*  W1           Wn        Ln          Wp ',
     +'       Lp '
      WRITE (30,*) '*               MCKT      SCKT        PDIFF'
      WRITE (30,*) '*'
      WRITE (30,6)
      WRITE (30,7)
      WRITE (30,8)
      WRITE (30,9)
      WRITE (30,806) N, I-1
806   format (1x,' Num chips/Tot chips =',I2,'/',I2)
C
      WRITE (30,46)IWAF(I-1),WAVE(1),WAVE(2),WAVE(46),WAVE(47)
      WRITE (30,48) WAVE(3),WAVE(4),WAVE(5)
      WRITE (30,*) ' '
      inc=0
      if (idev .eq. 2) inc=20                    ! just p-channel
821   write (30,42) wave(15+inc),wave(16+inc),wave(17+inc),
     + wave(11+inc)
      write (30,43) wave(9+inc),wave(8+inc),wave(10+inc),
     + wave(6+inc),wave(7+inc)
      WRITE (30,44) wave(13+inc),wave(12+inc),wave(23+inc),
     + wave(21+inc),wave(24+inc)
      WRITE (30,45) wave(14+inc),wave(20+inc),wave(25+inc),
     + wave(22+inc),wave(19+inc),wave(18+inc)
      if ((idev .eq. 1) .or. (inc .ge. 20)) go to 972
      write (30,47)
      inc=20
      go to 821
C-----------------------------------------------------------------------------
C
C     THE FORMAT CARDS
C
4     format(' *')
6     format (' *              CGDO      CGSO        CGBO',
     +'      RMS  ')
7     format (' *               LD        TOX         SUB ',
```

```
     +'      VTO       KP')
8      format(' *                 GAMMA      PHI         U0 ',
     +'      DELTA    VMAX')
9      format(' *                  XJ       KAPPA       NFS ',
     +'      ETA      THETA       ERR')
C      For row #2
42     format(15x,1e10.3,1x,1e10.3,1x,1e10.3,2x,F5.3)
C      For row #3
43     format(16x,F5.3,3x,F6.1,7x,1e10.3,1x,F6.3,2x,F6.2)
C      For the SUXES parameters - row #4
44     FORMAT (16X,F5.3,6x,F5.3,3x,F6.1,5x,F8.3,3x,1e10.3)
C      For the last row
45     format (16x, F5.3,3x,F6.1,7x,1e10.3,F8.4,F8.3,4x,F6.2)
C      Similar to 41
46     format (5x,A1,9x, F5.2, 6x, F5.2, 6x,F5.2,6x,F5.2)
47     format (1x,' ')
48     format (15x, F5.2, 6x, F5.2, 8x, F6.2)
C-----------------------------------------------------------------------
972    CONTINUE
       write(30,666)
666    format(' **************************************************',
     + '**********************')
       CALL LOTS(lot,n,i,wave,wsum,alave,alsum,idev,nodata)
       RETURN
       END
C***********************************************************************
C***********************************************************************
C
C      FILENAME:  LOTS.SUB
C
       SUBROUTINE LOTS(lot,n,i,wave,wsum,alave,alsum,idev,nodata)
C
       dimension wave(1), wsum(1), alave(1), alsum(1)
       common /plomi/ MARK, IWAF(100), lotnum1, lotnum2
C
       character*1 IWAF
       character*1 MARK
       character*4 lotnum1
       character*4 lotnum2
C
C
C-----------------------------------------------------------------------
C
C      DETERMINE THE LOT AVERAGES
C
       lot=lot+1
       K=0
       DO 300 K=1,47
       ALSUM(K)=ALSUM(K)+WAVE(K)
       ALAVE(K)=ALSUM(K)/LOT
300    CONTINUE
C
```

```
C       REINITIALIZE THE WAFER DIMENSIONS
C
        N=0
        I=0
        INC=0
        DO 1000 INC=1,47
        WAVE(INC)=0.0
        WSUM(INC)=0.0
1000    CONTINUE
        IF (LOT .LT. 2) RETURN
        IF (nodata .eq. 0) RETURN

C-------------------------------------------------------------------------------
C
C       SUM UP THE RESULTS FOR THE LOT
C
        WRITE (30,4)
        WRITE (30,*) '*** LOT AVERAGES ***'
        WRITE (30,4)
        WRITE (30,4)
        WRITE (30,*) '*  Lot          Wn         Ln          Wp ',
     +'       Lp '
        WRITE (30,*) '*              MCKT      SCKT        PDIFF'
        WRITE (30,*) '*'
        WRITE (30,6)
        WRITE (30,7)
        WRITE (30,8)
        WRITE (30,9)
        WRITE (30,836) lot
836     format (1x,' Num wafers =',I2)
        WRITE (30,50) lotnum1,lotnum2,alave(1),alave(2),alave(46),
     + alave(47)
        WRITE (30,48) alave(3),alave(4),alave(5)
        WRITE (30,*) ' '
        inc=0
        if (idev .eq. 2) inc=20                     ! just p-channel
936     write (30,42) alave(15+inc),alave(16+inc),alave(17+inc),
     + alave(11+inc)
        write (30,43) alave(9+inc),alave(8+inc),alave(10+inc),
     + alave(6+inc),alave(7+inc)
        WRITE (30,44) alave(13+inc),alave(12+inc),alave(23+inc),
     + alave(21+inc),alave(24+inc)
        WRITE (30,45) alave(14+inc),alave(20+inc),alave(25+inc),
     + alave(22+inc),alave(19+inc),alave(18+inc)
        if ((idev .eq. 1) .or. (inc .ge. 20)) go to 937
        write (30,47)
        inc=20
        go to 936
C-------------------------------------------------------------------------------
937     CONTINUE
        write(30,666)
666     format(' **************************************************',
```

```
     + '**********************')
C-----------------------------------------------------------------------------
C
C      THE FORMAT CARDS
C
4      format(' *')
6      format (' *             CGDO      CGSO        CGBO',
     +'      RMS  ')
7      format (' *              LD        TOX         SUB ',
     +'      VTO        KP')
8      format(' *             GAMMA      PHI         UO ',
     +'      DELTA    VMAX')
9      format(' *              XJ       KAPPA       NFS ',
     +'      ETA      THETA       ERR')
C      For row #2
42     format(15x,1e10.3,1x,1e10.3,1x,1e10.3,2x,F5.3)
C      For row #3
43     format(16x,F5.3,3x,F6.1,7x,1e10.3,1x,F6.3,2x,F6.2)
C      For the SUXES parameters - row #4
44     FORMAT (16X,F5.3,6x,F5.3,3x,F6.1,5x,F8.3,3x,1e10.3)
C      For the last row
45     format (16x, F5.3,3x,F6.1,7x,1e10.3,F8.4,F8.3,4x,F6.2)
47     format (1x,' ')
48     format (15x, F5.2, 6x, F5.2, 8x, F6.2)
50     format (5x,2A4,2x, F5.2, 6x, F5.2, 6x,F5.2,6x,F5.2)
C-----------------------------------------------------------------------------
       return
C*****************************************************************************
C*****************************************************************************
       END
```

RANKEY.FOR

```
C
C      FILENAME:  RANKEY.FOR
C      .
C      PURPOSE:  THIS PROGRAM ORDERS THE CORRELATION COEFFICIENT DATA
C                AND DETERMINES WHICH WAFER MAPS TO MAKE.
C
       dimension numfile(50), filnl(50), filn2(50)
       dimension number(50), cor(50), nchips(50)
       dimension nummax(50), cormax(50), nchipmax(50)
C
       character*1 iwaf
       character*4 filnl
       character*4 filn2
       character*4 filela
       character*4 file2a
       character*4 filelb
       character*4 file2b
       character*4 lotnuml
       character*4 lotnum2
C
       open(unit=10, file='in.dat', status='old')
       open(unit=11, file='lotnumber.fil', status='old')
       open(unit=12, file='nlw.fil', status='old')
       open(unit=15, file='coref.fil', status='old')
       open(unit=20, file='out.dat', status='unknown')
       open(unit=30, file='out2.dat', status='unknown')
       open(unit=40, file='mapl.dat', status='old')
C       open(unit=50, file='out4.dat', status='unknown')
C
C      read the lotnumber and wafer letter
C
       read (11,6) lotnuml, lotnum2
6      format(2A4)
       read (12,7) iwaf
7      format(Al)
C
C      read the coref file
C
       j=0
       do 131 j=2,46
       read (15,132,end=733) numfile(j), filnl(j), filn2(j)
132    format (I3,2A4)
131    continue
C
C      read the mapl file
C
733    read (40,833) filela, filelb
833    format (2A4)
       read (40,833) file2a, file2b
C
```

```
C      read the input data file
C
       j=0
       do 133 j=2,46
       read (10,*,err=135,end=135) number(j), itrash1, itrash2, itrash3,
     + itrash4, itrash5, itrash6, itrash7, cor(j), nchips(j)
133    continue
135    jmax=j-1
C
C      order the input data according to the correlation coefficient
C
       j=0
       nummax(1)=0
       do 126 j=2,jmax
       cormax(j)=0.0
       nummax(j)=0
126    continue
C
       j=0
       do 136 j=2,jmax
       if (abs(cor(j)) .gt. abs(cormax(2))) nummax(2)=number(j)
       if (abs(cor(j)) .gt. abs(cormax(2))) nchipmax(2)=nchips(j)
       if (abs(cor(j)) .gt. abs(cormax(2))) cormax(2)=cor(j)
136    continue
C
       i=2
137    i=i+1
       if (i .gt. jmax) go to 800
       j=0
       do 138 j=2,jmax
       if ((abs(cor(j)) .le. abs(cormax(i))) .or. (abs(cor(j)) .gt.
     + abs(cormax(i-1)))) go to 138
       if ((number(j) .eq. nummax(i-1)) .or. (number(j) .eq.
     + nummax(i-2))) go to 138
       if (i .lt. 5) go to 602
       if ((number(j) .eq. nummax(i-3)) .or. (number(j) .eq.
     + nummax(i-4))) go to 138
602    nummax(i)=number(j)
       nchipmax(i)=nchips(j)
       cormax(i)=cor(j)
138    continue
       go to 137
C
C      print the results in order
C
800    write (20,701) lotnum1, lotnum2, iwaf
701    format(/,1x,'Lot ',2A4,'Wafer ',A1)
       write (20,702)
702    format(/,13x,'Corr.')
       write (20,700)
700    format(1x,'Parameter',3x,'Coef.',3x,'#Chips',/)
C
```

```
      L=0
      m=50
      filnl(50)='xxxx'
      n=0
      k=1
802   k=k+1
      if (k .gt. jmax) go to 900
C
C     find the proper value for j
C
      j=0
      i=0
      do 803 i=2,46
      if (nummax(k) .eq. number(i)) j=i
803   continue
      n=n+1
C
      write (20,801) filnl(j), filn2(j), cormax(k), nchipmax(k)
801   format(2x,2A4,2x,F6.3,5x,I2)
      if ((n .eq. 1) .and. (filnl(j) .eq. 'MEAC')) m=j
C
C     DETERMINE WHICH MAPS TO GENERATE
C
      if (L .GT. 1) GO TO 802
      if (filnl(j) .EQ. 'MEAC') GO TO 802
      if (filnl(j) .EQ. 'PDIF') GO TO 802
      if (filnl(j) .EQ. 'SIMC') GO TO 802
      if ((filnl(m) .eq. 'MEAC') .and. (filnl(j) .eq. filela))
     + go to 802
      if ((filnl(m) .eq. 'MEAC') .and. (filnl(j) .eq. file2a))
     + go to 802
C     write (50,808) filela, filelb
C     write (50,808) file2a, file2b
C     write (50,808) filnl(j), filn2(j)
C     write (50,808) filnl(m), filn2(m)
      write (30,808) filnl(j), filn2(j)
808   format(2A4)
      L=L+1
      go to 802
C
900   CONTINUE
      END
```

NLW.FIL

```
Y
 0
 4
 2.46
 2.46
```

RC.FIL

```
*
*  FILENAME = RC.FIL
*   THIS IS A FILE OF CHIPS OF WHICH SUXES AND SPICE ARE TO BE
*   RUN
*
* Wafer Row Column
*
Y  2  4
```

LINE.FIL

```
*
*  Filename = LINE.FIL
*  Xerox wafer
*  Wafer Row  Column  Wn   Ln   Wp   Lp  (Measured Freq) NorPorBOTH
*  FORMAT (2x,A1,2x,I2,1x,I2,1x,F5.2,1x,F5.2,1x,F5.2,1x,F5.2,1x,F5.2,1x,I3)
  Y   2  4  2.46  2.46  6.46  2.46  9.40   3
```

LOTNUMBER.FIL

```
YY027
```

NPARIN.FIL

```
TOX     LD
635.
.38
```

PPARIN.FIL

```
TOX     LD
635.
.45
```

NINPUT.XXX

```
*
*  FILENAME:  NINPUT.DAT
*
*  PURPOSE:   THIS IS THE N-CHANNEL INPUT DATA FOR SPICE
*             WHICH GIVES AN INITIAL GUESS AND THE RANGE IN
*             WHICH TO EXPECT THE PARAMETER VALUE.
*             SUXES IS RESTRICTED TO THIS RANGE.
*             Date generated 6-20-83
*
*
*  parameters  initial   lowbound   upperbound
     Level          3        1          3
     Type           1.0     -1.0        1.0
     Xld            UUU      0.1        1.0
     Tox            TTT    250.0     1000.0
     Nsub           SSS      1.d13      1.d17
     Vto            RRR      0.3        1.5
     Xkp            OOOd-6   1.0d-6   100.d-6
     Gamma          MMM       .1        2.0
     Phi            JJJ       .3         .9
     Uo             KKK      0.1     1250.0
     Uexp           0.0      0.001      2.3
     Ucrit          0.0      1.d4       1.d6
     Delta          2.685   -1.d2       1.d2
     Vmax           7.9d5    1.d4       9.9d7
     Xj             HHH       .1        1.5
     Lamda          0.0      1.d-6      1.d-1
     Kappa          .8000    0.0      999.0
     Nfs          2.28d12    0.0        4e14
     Neff           0.0      1.d-2      1.dl
     Nss            1e11     0.0        1e14
     Tgate          1.0      0.0        1.0
     Eta            .1306   -2.0        2.0
     Theta          .3612   -2.0        2.0
```

PINPUT.XXX

```
*
*  FILENAME:  PINPUT.DAT
*
*  PURPOSE:  THIS IS THE P-CHANNEL INPUT DATA FOR SPICE
*            WHICH GIVES AN INITIAL GUESS AND THE RANGE IN
*            WHICH TO EXPECT THE PARAMETER VALUE.
*            SUXES IS RESTRICTED TO THIS RANGE.
*            Date generated 6-20-83
*
*
*  parameters  initial   lowbound   upperbound
     Level          3        1          3
     Type         -1.0     -1.0         1.0
     Xld           UUU      0.1         2.0
     Tox           TTT    250.0      1000.0
     Nsub          SSS      1.d13       1.d17
     Vto           RRR     -3.0        -0.1
     Xkp           OOOd-6   1.0d-6    100.d-6
     Gamma         MMM       .1          2.0
     Phi           JJJ       .3           .9
     Uo            KKK      0.1      1250.0
     Uexp           0.0     0.001        2.3
     Ucrit          0.0     1.d4         1.d6
     Delta        .6749    -1.d2         1.d2
     Vmax        3.24d5     1.d4         9.9d7
     Xj            HHH       .1          1.5
     Lamda          0.0     1.d-6        1.d-1
     Kappa          .20     0.0        999.0
     Nfs         1.17e12    1e10         4e14
     Neff           0.0     1.d-2        1.d1
     Nss           1e11     0.0          1e14
     Tgate         -1.0    -1.0          1.0
     Eta          .1210   -10.0         10.0
     Theta        .2396   -10.0         10.0
```

NOPT.FIL

```
*
*  FILENAME:  NOPT.FIL
*
*  PURPOSE:  THIS IS AN OPTIONS FILE FOR AN N-CHANNEL TRANSISTOR
*            IN WHICH THE DESIRED BIASES ARE SELECTED FROM THE IV DATA
*            Date generated 6-20-83
*
*
    Criteria
    Nsign=4 Maxfn=200 Delta=1.0d-9 Epsil=1.0d-6
    Linit=0.1 Lscal=2.0 Luppr=1000 Fcdsw=0.1
*
    RangVD    min=0.00   max=5.5   incr=0.0
*
    RangVG    min=0.00   max=5.5   incr=0.0
*
    RangVB    min=0.0    max=0.0   incr=1.0
*
*    Weight
*
```

POPT.FIL

```
*
*  FILENAME:  POPT.FIL
*
*  PURPOSE:  THIS IS AN OPTIONS FILE FOR A P-CHANNEL TRANSISTOR
*            IN WHICH THE DESIRED BIASES ARE SELECTED FROM THE IV DATA
*            Date generated 6-20-83
*
*
    Criteria
    Nsign=4 Maxfn=200 Delta=1.0d-9 Epsil=1.0d-6
    Linit=0.1 Lscal=2.0 Luppr=1000 Fcdsw=0.1
*
    RangVD    min=-5.5   max=0.00  incr=0.0
*
    RangVG    min=-5.5   max=0.00  incr=0.0
*
    RangVB    min=0.0    max=0.0   incr=1.0
*
*    Weight
*
```

NSTRAT.FIL

```
*
*  FILENAME:  NSTRAT.FIL
*
*  PURPOSE:  THIS IS THE STRATEGY FILE TO HELP ALLEVIATE THE
*            REDUNDANT PARAMETER PROBLEM.
*            Date generated 6-20-83
*
*
*  parameters  first  second  third  fourth  fifth
     Level       0      0       0      0       0
     Type        0      0       0      0       0
     Xld         0      0       1      0       0
     Tox         0      1       0      0       1
     Nsub        0      0       1      0       0
     Vto         0      1       0      0       1
     Xkp         0      0       1      0       0
     Gamma       0      1       0      0       1
     Phi         0      0       0      1       0
     Uo          0      0       0      1       0
     Uexp        0      0       0      0       0
     Ucrit       0      0       0      0       0
     Delta       1      1       0      0       1
     Vmax        0      0       0      1       0
     Xj          0      0       1      0       0
     Lamda       0      0       0      0       0
     Kappa       0      0       0      1       0
     Nfs         0      1       0      0       1
     Neff        0      0       0      0       0
     Nss         0      0       0      0       0
     Tgate       0      0       0      0       0
     Eta         1      0       0      0       0
     Theta       1      0       0      0       0
```

PSTRAT.FIL

```
*
*  FILENAME:  PSTRAT.FIL
*
*  PURPOSE:  THIS IS THE STRATEGY FILE TO HELP ALLEVIATE THE
*            REDUNDANT PARAMETER PROBLEM.
*            Date generated 6-20-83
*
*
*  parameters  first second third fourth fifth
     Level       0      0      0     0      0
     Type        0      0      0     0      0
     Xld         0      0      1     0      0
     Tox         0      1      0     0      1
     Nsub        0      0      1     0      0
     Vto         0      1      0     0      1
     Xkp         0      0      1     0      0
     Gamma       0      1      0     0      1
     Phi         0      0      0     1      0
     Uo          0      0      0     1      0
     Uexp        0      0      0     0      0
     Ucrit       0      0      0     0      0
     Delta       1      1      0     0      1
     Vmax        0      0      0     1      0
     Xj          0      0      1     0      0
     Lamda       0      0      0     0      0
     Kappa       0      0      0     1      0
     Nfs         0      1      0     0      1
     Neff        0      0      0     0      0
     Nss         0      0      0     0      0
     Tgate       0      0      0     0      0
     Eta         1      0      0     0      0
     Theta       1      0      0     0      0
```

IVCHARN.XXX

```
*
*
* FILENAME = IVCHARN.FIL
* NOTE: Use with MODEL.FIL to make IVCHARN.CEL
*
* N-CHANNEL MOS OUTPUT CHARACTERISTICS
*
M1 1 2 0 0 MODN L=BZBUM W=FFFUM AD=12.5P AS=12.5P PD=15U
+ PS=15U NRD=2.0 NRS=2.0
*
VD 3 0
VG 2 0
VIDS 3 1
*
.OPTIONS NODE NOPAGE
.DC VD 0 5 .5 VG 0 5 1
.PLOT DC I(VIDS)
.END
```

IVCHARP.XXX

```
*
*
* FILENAME = IVCHARP.FIL
* NOTE:  Use with MODEL.FIL to make IVCHARP.CEL
*
* P-CHANNEL MOS OUTPUT CHARACTERISTICS
*
M2 1 2 0 0 MODP L=BZBUM W=FFFUM AD=42.25P AS=42.25P PD=26U
+ PS=26U NRD=1.0 NRS=1.0
*
VD 4 0
VG 2 0
VIDS 4 1
*
.OPTIONS NODE NOPAGE
.DC VD 0 -5 -.5 VG 0 -5 -1
.PLOT DC I(VIDS)
.END
```

NMODEL.XXX

```
*
*
* FILENAME = NMODEL.FIL
* NOTE:  This file uses parameters from the process and SUXES.
*
.MODEL MODN NMOS LEVEL=3 LD=sqsUM TOX=rqrUM NSUB=pqp VTO=oqoV
+ KP=nqnU GAMMA=mqm PHI=lql UO=kqk DELTA=jqj VMAX=iqi
+ XJ=hqhUM KAPPA=gqg NFS=fqf ETA=eqe THETA=dqd
+ CJSW=0.6E-9 CGDO=cqc CGSO=bqb CGBO=aqa
+ RSH=11.63 JS=5.0E-5 TPG=1.0 MJ=.5 MJSW=.5 PB=.9 XQC=.45
*
```

PMODEL.XXX

```
*
*
* FILENAME = PMODEL.FIL
* NOTE:  This file uses parameters from the process and SUXES.
*
.MODEL MODP PMOS LEVEL=3 LD=sqsUM TOX=rqrUM NSUB=pqp VTO=oqoV
+ KP=nqnU GAMMA=mqm PHI=lql UO=kqk DELTA=jqj VMAX=iqi
+ XJ=hqhUM KAPPA=gqg NFS=fqf ETA=eqe THETA=dqd
+ CJSW=0.6E-9 CGDO=cqc CGSO=bqb CGBO=aqa
+ RSH=61.57 JS=5.0E-5 TPG=-1.0 MJ=.5 MJSW=.5 PB=.9 XQC=.45
*
```

SUBCKT.XXX

```
*
*
* FILENAME = INVERT.FIL
*
.SUBCKT INVERT 1 2 3
*
M1 100 1 0 0 MODN L=hzhUM W=iziUM AD=12.5P AS=12.5P PD=15U
+ PS=15U NRD=2.0 NRS=2.0
M2 100 1 3 3 MODP L=jzjUM W=kzkUM AD=42.25P AS=42.25P PD=26U
+ PS=26U NRD=1.0 NRS=1.0
R1 100 2 1332
C1 2 0 .043P
*
.ENDS INVERT
*
```

CIRCUIT.FIL

```
*
* FILENAME = CIRCUIT.FIL
* NOTE:  Use with NMODEL.FIL, PMODEL.FIL, and SUBCKT.XXX to make CIRCUIT.CEL
*
X1 1 2 3 INVERT
X2 2 4 3 INVERT
X3 4 5 3 INVERT
X4 5 6 3 INVERT
X5 6 7 3 INVERT
X6 7 8 3 INVERT
X7 8 9 3 INVERT
X8 9 10 3 INVERT
X9 10 11 3 INVERT
REND 11 200 1110
CEND 200 0 .044P
X10 200 12 3 INVERT
X11 12 13 3 INVERT
X12 13 14 3 INVERT
X13 14 15 3 INVERT
X14 15 16 3 INVERT
X15 16 17 3 INVERT
X16 17 18 3 INVERT
X17 18 19 3 INVERT
X18 19 20 3 INVERT
X19 20 400 3 INVERT
RCIR 400 1 334
CCIR 1 0 .073P
RPP 1 101 1.175
CPP 101 0 .322P
CPROBE 101 0 .1P
*X20 6 50 30 51 52 OUTBUF3
*
.IC V(1)=5 V(2)=0 V(4)=5 V(5)=0 V(6)=5 V(7)=0 V(8)=5
.IC V(9)=0 V(10)=5 V(11)=0 V(12)=5 V(13)=0 V(14)=5 V(15)=0
.IC V(16)=5 V(17)=0 V(18)=5 V(19)=0 V(20)=5
*.IC V(30)=5 V(50)=0 V(51)=0 V(52)=5
.IC V(3)=5
*
VCC 3 0 5V
*VCCO 30 0 5.0V
*VA 24 0 PULSE(0V 5V 0NS 5NS 5NS 1000NS 2000NS)
*
.OPTIONS VNTOL=10UV LIMPTS=500 ITL4=50 ITL5=0
.TRAN 1.0NS 200NS UIC
*.PLOT TRAN V(101)
*.PLOT TRAN V(1)
*.PLOT TRAN V(2)
.PLOT TRAN V(4)
*.PLOT TRAN V(50)
.END
```

PCHIPS.FIL

```
* percent acceptable chips to make wafer maps ... number of test sites/wafer 30.0
65
```

STEND.FIL

```
  4  2  1  1
  6  8  9  9
  1  1  1  2
  9  9  9  8
  4  0  0  0
  6  0  0  0
  0  0  0  0
  0  0  0  0
  0  0  0  0
  0  0  0  0
  0  0  0  0
  0  0  0  0
  0  0  0  0
  0  0  0  0
  0  0  0  0
  0  0  0  0
```

COREF.FIL

```
 2 W
 3 L
 4 WP
 5 LP
 6 MCKT
 7 SCKT
 8 PDIFF
 9 NCGDO
10 NCGBO
11 NRMS
12 NLD
13 NTOX
14 NSUB
15 NVTO
16 NKP
17 NGAMMA
18 NPHI
19 NUO
20 NDELTA
21 NVMAX
22 NXJ
23 NKAPPA
24 NNFS
25 NETA
26 NTHETA
27 NERR
28 PCGDO
29 PCGBO
30 PRMS
31 PLD
32 PTOX
33 PSUB
34 PVTO
35 PKP
36 PGAMMA
37 PPHI
38 PUO
39 PDELTA
40 PVMAX
41 PXJ
42 PKAPPA
43 PNFS
44 PETA
45 PTHETA
46 PERR
```

EDNO.WAY

```
!
! filename = edno.way
!
delete 1:end-1
exit
```

EDSUXN.WAY

```
!
! filename = EDSUXN.WAY
!
s/77/N/1:end
s/22/N22/1:end
exit
```

EDSUXP.WAY

```
!
! filename = EDSUXP.WAY
!
s/77/P/1:end
s/22/P22/1:end
exit
```

EDPAR.WAY

```
!
! FILENAME = EDPAR.WAY
!
! delete unnecessary rows
1
t 'Xld'
delete 1:.-1
t 'Uo'
delete .+1:.+2
t 'Xj'
delete .+1
t 'Xnfs'
delete .+1:.+3
!
! delete first column
!
sub/  Xld          //1:end-1
sub/  Tox          //1:end-1
sub/  Nsub         //1:end-1
sub/  Vto          //1:end-1
sub/  Xkp          //1:end-1
sub/  Gamma        //1:end-1
sub/  Phi          //1:end-1
sub/  Uo           //1:end-1
sub/  Delta        //1:end-1
sub/  Vmax         //1:end-1
sub/  Xj           //1:end-1
sub/  Xkappa       //1:end-1
sub/  Xnfs         //1:end-1
sub/  Theta        //1:end-1
sub/  Eta          //1:end-1
!
resequence
exit
```

EDSPICEN.WAY

```
!
! FILENAME = EDSPICEN.WAY
!
s/ //end-5:end
s/777/N/1:end
exit
```

EDSPICEP.WAY

```
!
! FILENAME = EDSPICEP.WAY
!
s/ //end-5:end
s/777/P/1:end
exit
```

EDLAST.WAY

```
!
! filename = EDLAST.WAY
!
t 'N='
move end-1 to .
delete
1
t 'M='
move end-1 to .
delete
1
t 'L='
move end-1 to .
delete
resequence
exit
```

EDNRMS.WAY

```
!
! filename = ednrms.way
!
t 'a g a i n'
delete .:end-1
t end-50
t 'optimization'
.-3
delete 1:.-1
delete .+1:end-1
resequence
1
s/max err=//
s/% average err=//
s/% rms=//
exit
```

EDPRMS.WAY

```
!
! filename = edprms.way
!
t 'a g a i n'
t 'a g a i n'
delete .:end-1
t end-50
t 'optimization'
.-3
delete 1:.-1
delete .+1:end-1
resequence
1
s/max err=//
s/% average err=//
s/% rms=//
exit
```

EDRES.WAY

```
!
! FILENAME = EDRES.WAY
!
! add an '#' to denote new data
1
t '$ N='
delete .-2:.
copy 1 to .
resequence
exit
```

EDSKEY.WAY

```
!
! FILENAME = EDSKEY.WAY
!
delete end-1
!
1
t '*'
t '*'
t '*'
t '*'
t '*'
t '*'
t '*'
t '*'
t '*'
t '*'
t '*'
t '*'
t '*'
t '*'
t '*'
t '*'
t '*'
t '*'
t '*'
t '*'
t '*'
t '*'
t '*'
t '*'
t '*'
t '*'
delete 1:.-1
!
resequence
exit
```

EDMAP.WAY

```
!
! FILENAME = EDMAP.WAY
!
1
t 'K='
delete
move end-1 to .
!
1
t 'N='
delete
move end-1 to .
!
delete end-2:end-1
resequence
exit
```

EDSIMF.WAY

```
!
! FILENAME = EDSIMF.WAY
!
1
t 'GET 7'
t 'SDB'
t 'ENTRY'
delete 1:+
t 'NUMBER'
delete .:end-1
!
s/-JAN-//1:end-1
s/-FEB-//1:end-1
s/-MAR-//1:end-1
s/-APR-//1:end-1
s/-MAY-//1:end-1
s/-JUN-//1:end-1
s/-JUL-//1:end-1
s/-AUG-//1:end-1
s/-SEP-//1:end-1
s/-OCT-//1:end-1
s/-NOV-//1:end-1
s/-DEC-//1:end-1
!
s/://1:end-1
s/://1:end-1
!
resequence
exit
```

EDPDIFF.WAY

```
!
! FILENAME = EDPDIFF.WAY
!
1
t 'GET 8'
t 'SDB'
t 'ENTRY'
delete 1:+
t 'NUMBER'
delete .:end-1
!
s/-JAN-//1:end-1
s/-FEB-//1:end-1
s/-MAR-//1:end-1
s/-APR-//1:end-1
s/-MAY-//1:end-1
s/-JUN-//1:end-1
s/-JUL-//1:end-1
s/-AUG-//1:end-1
s/-SEP-//1:end-1
s/-OCT-//1:end-1
s/-NOV-//1:end-1
s/-DEC-//1:end-1
!
s/://1:end-1
s/://1:end-1
!
resequence
exit
```

EDMEAF.WAY

```
!
! FILENAME = EDMEAF.WAY
!
1
t 'GET 6'
t 'SDB'
t 'ENTRY'
delete 1:+
t 'NUMBER'
delete .:end-1
!
s/-JAN-//1:end-1
s/-FEB-//1:end-1
s/-MAR-//1:end-1
s/-APR-//1:end-1
s/-MAY-//1:end-1
s/-JUN-//1:end-1
s/-JUL-//1:end-1
s/-AUG-//1:end-1
s/-SEP-//1:end-1
s/-OCT-//1:end-1
s/-NOV-//1:end-1
s/-DEC-//1:end-1
!
s/://1:end-1
s/://1:end-1
!
resequence
exit
```

EDMAPS.WAY

```
!
! FILENAME = EDMAPS.WAY
!
1
t 'N='
delete
move end-1 to .
!
delete end-2:end-1
s/   //end-4:end-1
s/  //end-4:end-1
s/ //end-4:end-1
!
resequence
exit
```

Appendix B - Modifying SUXES's Strategy File

The strategy file needs modification if: (1) SUXES's predicted I-V characteristics do not resemble the measured I-V characteristics, (2) the user does not accept the percent deviation between these curves, and (3) the user is not satisfied with the percent difference between the simulated and measured ring oscillator frequencies. One such modification technique is to determine the parameter(s) which is(are) causing the deviation in the I-V characteristics. To do this, plot the measured I-V data along with the I-V characteristics using the initial estimates for the parameter values. If this plot is not qualitatively satisfactory to the user, the values for the initial estimates must be reexamined. If using the initial estimates for the parameter values gives I-V characteristics which resemble the measured I-V characteristics, then plot the I-V characteristics after the values of the first iteration of parameters are optimized.

If the user accepts these plots, continue until an iteration of parameters fails to produce an acceptable plot. In the set of parameters which produces the unsatisfactory plot, eliminate one parameter at a time in the strategy file to determine which one(s) cause(s) the deviation between the measured I-V data and the simulated I-V data. By using this method, the parameter(s) causing the deviation can be found. The parameter KAPPA (the saturation field factor) is an example of the above. Assume the user is not satisfied with the upward bending of the I-V characteristics in the saturation region, and assume KAPPA is causing this bending after the first iteration, while the I-V characteristics using the initial estimates do not have this trait. A technique to eliminate the effect of KAPPA is to obtain an accurate initial estimate of the value of this parameter and then avoid optimizing this parameter's value until the values of the other parameters have been optimized. To obtain an initial estimate for the value of KAPPA, a parameter value must be optimized which has very little influence on the result (for example, the value of NFS). Therefore, using about four or five different initial estimates for the value of KAPPA, SUXES can be run to optimize only the value of NFS while using the initial estimates for the values of the other parameters. An appropriate value of KAPPA is found when the user is satisfied with the resemblance of the simulated I-V characteristics to the measured I-V data. Using the technique described above, the simulated I-V characteristics will have a realistic shape even though the rms deviation might be lower with other, less realistic optimization techniques.

After a reasonable initial estimate for the value of KAPPA is found, an appropriate strategy file must be found. SUXES can operate in an automated or a manual mode. Up until now it did not matter in which mode it was operating. At this point, it is best to use the manual mode which includes a sensitivity analysis. To obtain an appropriate strategy file, several parameters (perhaps three) can be randomly selected and optimized. These are potential candidates for the first iteration of parameters. These parameters can be optimized as a group if no errors occur and high values in the sensitivity analysis are obtained. Low values in the sensitivity analysis indicate the existence of redundant parameters, and the directions of the eigenvectors indicate which parameters are redundant [1,16]. The boundary values may be altered to decrease the number of errors due to the pegging of a parameter at a boundary. If the above criteria have been met, and if a plot of the measured I-V data resembles the simulated I-V data, then the first three parameters have been found.

After the values of the first three parameters are found, the process is repeated for the second, third, etc., iteration of parameters until the values of all the parameters are optimized and the user is satisfied with the rms deviation. The values of the parameters can be optimized more than once.

The optimization technique for the *p*-channel MOSFETs, which follows, was found in the manner described above. This optimization technique is also used for the *n*-channel MOSFETs.

(i) DELTA, ETA, THETA (the first iteration parameters)
(ii) TOX, VTO, GAMMA, DELTA, NFS (the second iteration parameters, etc.)
(iii) LD, NSUB, KP, XJ
(iv) PHI, UO, VMAX, KAPPA
(v) TOX, VTO, GAMMA, DELTA, NFS

Appendix C - The Procedure to Run KEYS

The computer procedure KEYS was written on a VAX-11/780 using VAX/VMS. Data are initially received from the ACCUTEST, then transferred to the VAX subdirectory which contains the computer procedure KEYS.

The ACCUTEST data files have a format similar to that shown in table C-1. On the line beginning with POS, the chip position is recorded. The following two lines are ignored. The data values are recorded on the remaining lines starting with VAL. First, the V_{ds} value is listed, followed on consecutive lines by the V_{gs} value, the V_{bs} value, and then the I_{ds} value. This sequence is repeated for the remainder of the data sets. KEYS converts these data into a format, as shown in table C-2, suitable for SUXES. Also, KEYS provides the plotting package, PLOTMY, with the V_{ds} and I_{ds} data points needed to plot the I-V characteristics. The format for the plot file is similar to that given in table C-3 (without the date and time). The other plotting package, S5PLOT, plots the ring oscillator node potential as a function of time and records the period and frequency. It is provided data similar to those shown in table C-3.

The files listed in the order of their occurrence for the computer procedure KEYS are given in table C-4. The files that may be edited before KEYS is run are given in table C-5. After editing, KEYS is evoked by typing “submit/noprinter/notify KEYS.”

Table C-1. Example of a Data File from the ACCUTEST

```
TPN C01N
STA 1
DAT 19-NOV-90
BTI 11:11:10
ETI
OPE
LOT
DEV
WFT
TSN
DSN    25
WFN     1
DIE    25
POS 8.000000E+00 5.000000E+00
VAL    1    0.000000E+00
VAL    2    1.491000E-10
VAL    1    1.000000E-01
VAL    2    5.000000E-01
VAL    3    0.000000E+00
VAL    4    1.491000E-10
VAL    1    2.000000E-01
VAL    2    5.000000E-01
VAL    3    0.000000E+00
VAL    4    1.609000E-10
VAL    1    3.000000E-01
VAL    2    5.000000E-01
VAL    3    0.000000E+00
VAL    4    1.681000E-10
VAL    1    5.000000E-01
VAL    2    5.000000E-01
VAL    3    0.000000E+00
VAL    4    1.806000E-10
...
...
...
VAL    1    5.000000E+00
VAL    2    5.000000E+00
VAL    3    0.000000E+00
VAL    4    1.849000E-04
BIN    0
EOF
```

Table C-2. Example of a SUXES Data File

*	Data from the ACCUTEST		
* VDS	VGS	VBS	IDS
	Width/Length= 2.44/ 2.44		
0.10	0.50	0.00	1.49100E−10
0.20	0.50	0.00	1.60900E−10
0.30	0.50	0.00	1.68100E−10
0.10	1.00	0.00	6.12000E−07
0.20	1.00	0.00	9.11000E−07
0.30	1.00	0.00	1.03100E−06
0.10	1.50	0.00	2.23000E−06
0.20	1.50	0.00	4.04000E−06
0.30	1.50	0.00	5.47000E−06
0.10	2.00	0.00	3.82000E−06
0.20	2.00	0.00	7.24000E−06
0.30	2.00	0.00	1.02300E−05
0.10	2.50	0.00	5.34000E−06
0.20	2.50	0.00	1.02700E−05
0.30	2.50	0.00	1.47600E−05
0.10	3.00	0.00	6.78000E−06
0.20	3.00	0.00	1.31700E−05
0.30	3.00	0.00	1.91300E−05
0.10	3.50	0.00	8.13000E−06
0.20	3.50	0.00	1.59000E−05
0.30	3.50	0.00	2.33000E−05
0.10	4.00	0.00	9.45000E−06
0.20	4.00	0.00	1.85900E−05
0.30	4.00	0.00	2.73000E−05
0.10	4.50	0.00	1.07500E−05
0.20	4.50	0.00	2.11000E−05
0.30	4.50	0.00	3.11000E−05
0.10	5.00	0.00	1.19700E−05
0.20	5.00	0.00	2.36000E−05
0.30	5.00	0.00	3.48000E−05

Table C-3. Example of a Plot File

```
19-NOV-90 05:15:55
201
0.000E+00   1.158E+00
1.000E-09   6.697E-01
2.000E-09   5.665E-01
3.000E-09   4.710E-01
4.000E-09   3.861E-01
5.000E-09   3.147E-01
6.000E-09   2.560E-01
7.000E-09   2.082E-01
8.000E-09   1.683E-01
9.000E-09   1.362E-01
1.000E-08   1.101E-01
1.100E-08   8.813E-02
1.200E-08   7.098E-02
1.300E-08   5.754E-02
1.400E-08   4.680E-02
1.500E-08   3.767E-02
...         ...
...         ...
...         ...
1.900E-07   1.040E+00
1.910E-07   2.005E+00
1.920E-07   3.091E+00
1.930E-07   4.029E+00
1.940E-07   4.770E+00
1.950E-07   4.915E+00
1.960E-07   4.977E+00
1.970E-07   4.993E+00
1.980E-07   4.997E+00
1.990E-07   4.999E+00
2.000E-07   5.000E+00
```

Table C-4. The Files Listed in the Order of Their Occurrence in the Computer Procedure KEYS. This list does not include files that are created during the program. The FORTRAN files are provided data files which can be edited by the editor files. The command procedures are connected together creating the computer procedure KEYS.

POSSIBLE COMMAND PROCEDURES	FORTRAN FILES	DATA FILES	EDITOR FILES
(1) KEYS.COM			
			(1) EDNO.WAY
		+ (1) NLW.FIL	
		+ (2) RC.FIL	
		+ (3) LINE.FIL	
		+ (4) LOTNUMBER.FIL	
	(1) RCKEY.FOR		
(2) LAST3.COM			
(3) INKEY.XXX			
		(5) W*R*C*VB.DAT	
		(6) NW*R*C*.DAT	
		(7) PW*R*C*.DAT	
		+ (8) NPARIN.FIL	
		+ (9) PPARIN.FIL	
	(2) SUKEY.FOR		
		(10) NINPUT.XXX	
		(11) PINPUT.XXX	
		+ (12) NOPT.FIL	
		+ (13) POPT.FIL	
		+ (14) NSTRAT.FIL	
		+ (15) PSTRAT.FIL	
(3A) SUXES.XXX			(2) EDSUXN.WAY
			(3) EDSUXP.WAY
	* (3) SSUXES.FOR		
			(4) EDPAR.WAY
	(4) CAPKEY.FOR		
		+ (16) IVCHARN.XXX	
		+ (17) IVCHARP.XXX	
		+ (18) NMODEL.XXX	
		+ (19) PMODEL.XXX	
(3B) SPICEIV.XXX			(5) EDSPICEN.WAY
			(6) EDSPICEP.WAY
	(5) SPICENBS.FOR		
	(6) IVKEY.FOR		
	(7) PLOTMY.FOR		
		+ (20) SUBCKT.XXX	
		+ (21) CIRCUIT.FIL	

	(8) S5PLOT.FOR (9) MCVAX.FOR		
(4) RMSKEY.COM	(10) SUMKEY.FOR	+ (22) PCHIPS.FIL	(7) EDLAST.WAY (8) EDNRMS.WAY (9) EDPRMS.WAY (10) EDRES.WAY
(5) STKEY.COM	(11) CRDB.FOR (12) STAT2.FOR	+ (23) STEND.FIL	(11) EDSKEY.WAY
(6) CORKEY.XXX	(13) RANKEY.FOR	(24) COREF.FIL	(12) EDMAP.WAY (13) EDSIMF.WAY (14) EDPDIFF.WAY (15) EDMEAF.WAY
(7) MAPKEY.XXX			(16) EDMAPS.WAY

* To obtain this program, consult K. Doganis. Refer to table 1 for details.
\+ The user may edit these files according to table C-5.

Table C-5. The Files to be Edited at the Discretion of the User

FILENAME	POSSIBLE MODIFICATIONS	FORMAT
(1) NLW.FIL	The wafer letter, the row, and the column of the first chip to consider are entered into this file on the first three lines. If the row number is zero, the comprehensive summary is started from scratch. This is done before each lot is analyzed. The fourth and fifth lines should not be modified.	(A1) (I2) (I2)
(2) RC.FIL	KEYS considers the chips (denoted by wafer letter, row, and column) in this file.	(A1,1X,I2, 1X, I2)
(3) LINE.FIL	The data lines of this file contain the wafer letter, the row, the column, the width in micrometers of the *n*-channel MOSFETs, the length of the *n*-channel MOSFETs, the width of the *p*-channel MOSFETs, the length of the *p*-channel MOSFETs, the measured ring oscillator frequency in MHz, and a designation indicating NMOS technology (1), PMOS technology (2), or CMOS technology (3).	(2X,A1,2X, I2,1X,I2, 1X,F5.2,1X, F5.2,1X,F5.2, 1X,F5.2,1X, F5.2,1X,I3)
(4) LOTNUMBER.FIL	The lotnumber is entered into this file.	(2A4)
(5) NPARIN.FIL	The *n*-channel MOSFET oxide thickness in angstroms is followed by the lateral diffusion in micrometers.	free format
(6) PPARIN.FIL	The *p*-channel MOSFET oxide thickness in angstroms is followed by the lateral diffusion in micrometers.	free format
(7) NOPT.FIL	The *n*-channel MOSFET options file for SUXES.	see reference [1]
(8) POPT.FIL	The *p*-channel MOSFET options file for SUXES.	see reference [1]
(9) NSTRAT.FIL	The *n*-channel MOSFET strategy file for SUXES.	see reference [1]
(10) PSTRAT.FIL	The *p*-channel MOSFET strategy file for SUXES.	see reference [1]
(11) IVCHARN.XXX	For the *n*-channel MOSFETs, modifications to the SPICE device parameters (AD, AS, PD, PS, NRD, and NRS) and output characteristics are made here. The program will input the widths and lengths; therefore, do not modify this portion of the file.	see references [2,6]

(12) IVCHARP.XXX	For the *p*-channel MOSFETs, modifications to the SPICE device parameters (AD, AS, PD, PS, NRD, and NRS) and output characteristics are made here. The program will input the widths and lengths; therefore, do not modify this portion of the file.	see references [2,6]
(13) NMODEL.XXX	For the *n*-channel MOSFETs, modifications to the SPICE model parameters LEVEL, CJSW, RSH, JS, TPG, MJ, MJSW, PB, and XQC are made here. The program will input the other parameters; therefore, do not modify this portion of the file.	see references [2,6]
(14) PMODEL.XXX	For the *p*-channel MOSFETs, modifications to the SPICE model parameters LEVEL, CJSW, RSH, JS, TPG, MJ, MJSW, PB, and XQC are made here. The program will input the other parameters; therefore, do not modify this portion of the file.	see references [2,6]
(15) SUBCKT.XXX	Modifications to the SPICE subcircuit file are made here. The program will input the widths and lengths; therefore, do not modify this portion of the file.	see references [2,6]
(16) CIRCUIT.FIL	Modifications to the chosen SPICE file are made here.	see references [2,6]
(17) PCHIPS.FIL	The percentage of acceptable chips needed to make wafer maps is the first data value, and the number of test sites per wafer is the second data value.	free format
(18) STEND.FIL	Modifications to the STAT2 stend array are made here.	(1X,4(I2,1X))

NIST-114A
(REV. 3-90)

U.S. DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY

BIBLIOGRAPHIC DATA SHEET

1. PUBLICATION OR REPORT NUMBER
NIST/SP-400/90

2. PERFORMING ORGANIZATION REPORT NUMBER

3. PUBLICATION DATE
April 1992

4. TITLE AND SUBTITLE

Evaluating a Chip, Wafer, or Lot Using SUXES, SPICE, and STAT2

5. AUTHOR(S)

J. C. Marshall and R. L. Mattis

6. PERFORMING ORGANIZATION (IF JOINT OR OTHER THAN NIST, SEE INSTRUCTIONS)

U.S. DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
GAITHERSBURG, MD 20899

7. CONTRACT/GRANT NUMBER

8. TYPE OF REPORT AND PERIOD COVERED
Final

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (STREET, CITY, STATE, ZIP)

Same as item #6

10. SUPPLEMENTARY NOTES

11. ABSTRACT (A 200-WORD OR LESS FACTUAL SUMMARY OF MOST SIGNIFICANT INFORMATION. IF DOCUMENT INCLUDES A SIGNIFICANT BIBLIOGRAPHY OR LITERATURE SURVEY, MENTION IT HERE.)

The computer procedure KEYS (linKing softwarE to analYze waferS) links SUXES (Stanford University eXtractor of modEl parameterS), SPICE (a Simulation Program with Integrated Circuit Emphasis), and STAT2. Given data points for individual devices, SUXES obtains the model parameters for SPICE. SPICE predicts the behavior of an individual device or an entire circuit. After analyzing each test chip on a wafer, STAT2 determines correlation coefficients and generates wafer maps of selected parameters. These wafer maps are valuable to the designer, modeler, and process engineer.

The entire package accomplishes the following: (1) standardizes the technique of running SUXES and SPICE in an integrated mode; (2) simulates and plots the characteristic curves; (3) simulates and plots the results of an optional dynamic circuit (for example, a ring oscillator); (4) performs steps (2) and (3) for every test chip on each wafer; (5) summarizes the results from each chip, each wafer, and the lot; (6) rank-orders the model parameters for each wafer according to their correlation coefficients with respect to chosen circuit parameters; and (7) generates wafer maps of several quantities. A CMOS 19-stage ring oscillator is used to illustrate the capabilities of KEYS.

12. KEY WORDS (6 TO 12 ENTRIES; ALPHABETICAL ORDER; CAPITALIZE ONLY PROPER NAMES; AND SEPARATE KEY WORDS BY SEMICOLONS)

circuit simulator; CMOS; ring oscillator, SPICE, STAT2, SUXES, wafer maps

13. AVAILABILITY

X	UNLIMITED
	FOR OFFICIAL DISTRIBUTION. DO NOT RELEASE TO NATIONAL TECHNICAL INFORMATION SERVICE (NTIS).
X	ORDER FROM SUPERINTENDENT OF DOCUMENTS, U.S. GOVERNMENT PRINTING OFFICE, WASHINGTON, DC 20402.
X	ORDER FROM NATIONAL TECHNICAL INFORMATION SERVICE (NTIS), SPRINGFIELD, VA 22161.

14. NUMBER OF PRINTED PAGES

145

15. PRICE

ELECTRONIC FORM * U.S. G.P.O.:1992-311-891:60457

NIST *Technical Publications*

Periodical

Journal of Research of the National Institute of Standards and Technology—Reports NIST research and development in those disciplines of the physical and engineering sciences in which the Institute is active. These include physics, chemistry, engineering, mathematics, and computer sciences.
Papers cover a broad range of subjects, with major emphasis on measurement methodology and the basic technology underlying standardization. Also included from time to time are survey articles on topics closely related to the Institute's technical and scientific programs. Issued six times a year.

Nonperiodicals

Monographs—Major contributions to the technical literature on various subjects related to the Institute's scientific and technical activities.

Handbooks—Recommended codes of engineering and industrial practice (including safety codes) developed in cooperation with interested industries, professional organizations, and regulatory bodies.

Special Publications—Include proceedings of conferences sponsored by NIST, NIST annual reports, and other special publications appropriate to this grouping such as wall charts, pocket cards, and bibliographies.

Applied Mathematics Series—Mathematical tables, manuals, and studies of special interest to physicists, engineers, chemists, biologists, mathematicians, computer programmers, and others engaged in scientific and technical work.

National Standard Reference Data Series—Provides quantitative data on the physical and chemical properties of materials, compiled from the world's literature and critically evaluated. Developed under a worldwide program coordinated by NIST under the authority of the National Standard Data Act (Public Law 90-396). NOTE: The Journal of Physical and Chemical Reference Data (JPCRD) is published bimonthly for NIST by the American Chemical Society (ACS) and the American Institute of Physics (AIP). Subscriptions, reprints, and supplements are available from ACS, 1155 Sixteenth St., NW., Washington, DC 20056.

Building Science Series—Disseminates technical information developed at the Institute on building materials, components, systems, and whole structures. The series presents research results, test methods, and performance criteria related to the structural and environmental functions and the durability and safety characteristics of building elements and systems.

Technical Notes—Studies or reports which are complete in themselves but restrictive in their treatment of a subject. Analogous to monographs but not so comprehensive in scope or definitive in treatment of the subject area. Often serve as a vehicle for final reports of work performed at NIST under the sponsorship of other government agencies.

Voluntary Product Standards—Developed under procedures published by the Department of Commerce in Part 10, Title 15, of the Code of Federal Regulations. The standards establish nationally recognized requirements for products, and provide all concerned interests with a basis for common understanding of the characteristics of the products. NIST administers this program as a supplement to the activities of the private sector standardizing organizations.

Consumer Information Series—Practical information, based on NIST research and experience, covering areas of interest to the consumer. Easily understandable language and illustrations provide useful background knowledge for shopping in today's technological marketplace.

Order the **above** *NIST publications from: Superintendent of Documents, Government Printing Office, Washington, DC 20402.*

Order the **following** *NIST publications—FIPS and NISTIRs—from the National Technical Information Service, Springfield, VA 22161.*

Federal Information Processing Standards Publications (FIPS PUB)—Publications in this series collectively constitute the Federal Information Processing Standards Register. The Register serves as the official source of information in the Federal Government regarding standards issued by NIST pursuant to the Federal Property and Administrative Services Act of 1949 as amended, Public Law 89-306 (79 Stat. 1127), and as implemented by Executive Order 11717 (38 FR 12315, dated May 11, 1973) and Part 6 of Title 15 CFR (Code of Federal Regulations).

NIST Interagency Reports (NISTIR)—A special series of interim or final reports on work performed by NIST for outside sponsors (both government and non-government). In general, initial distribution is handled by the sponsor; public distribution is by the National Technical Information Service, Springfield, VA 22161, in paper copy or microfiche form.

U.S. Department of Commerce
National Institute of Standards and Technology
Gaithersburg, MD 20899

Official Business
Penalty for Private Use $300