```
zzzzzzzzzz  OOOOOOOOOO  MMM     MMM  BBBBBBBBB   IIIIIIII  EEEEEEEEEE
zzzzzzzzzz  OOOOOOOOOO  MMM     MMM  BBBBBBBBB   IIIIIIII  EEEEEEEEEE
      zzz   OO      OO  MMMM   MMMM  BB      BB     II     EE
      zzz   OO      OO  MMMM   MMMM  BB      BB     II     EE
    zzz     OO      OO  MM  MM  MM  BBBBBBBBB      II     EEEEEEEE
    zzz     OO      OO  MM  MM  MM  BBBBBBBBB      II     EEEEEEEE
  zzz       OO      OO  MM      MM  BB      BB     II     EE
  zzz       OO      OO  MM      MM  BB      BB     II     EE
zzz         OO      OO  MM      MM  BB      BB     II     EE
zzz         OO      OO  MM      MM  BB      BB     II     EE
zzzzzzzzzz  OOOOOOOOOO  MM      MM  BBBBBBBBB   IIIIIIII  EEEEEEEEEE
zzzzzzzzzz  OOOOOOOOOO  MM      MM  BBBBBBBBB   IIIIIIII  EEEEEEEEEE
```

V E R S I O N 1.1

by Werner JÜNGLING

INSTITUT FÜR ALLGEMEINE ELEKTROTECHNIK UND ELEKTRONIK

ABTEILUNG FÜR COMPUTER UNTERSTÜTZES KONSTRUIEREN

ABTEILUNG FÜR PHYSIKALISCHE ELEKTRONIK

TECHNICAL UNIVERSITY OF VIENNA

GUSSHAUSSTRASSE 27-29

A-1040 VIENNA

AUSTRIA

C O N T E N T S

1. I N T R O D U C T I O N

The program package ZOMBIE is a one dimensional general solver for

* systems of coupled parabolic differential equations with non constant coefficients
* Systems of coupled elliptic differential equations with non constant coefficients
* systems of coupled ordinary differential equations of first order with non constant coefficients and
* systems of coupled nonlinear algebraic equations.

ZOMBIE has been designed for the simulation of complete IC-fabrication steps and the determination of the electric behavior of these devices. ZOMBIE is thexefore supported by models which enable the simulation of ion implantations, diffusions and predepositions of typical impurities in silicon. Furthermore the transient electrical behavior of diodes can be simulated.

The development of new and improved physical models for process simulation is the main concern of the program package. ZOMBIE offers therefore the possibility to set up these models just by specifying the main functions describing the differential equations. Most of the numerical work (e.g. discretization, solving and grids in space and time) is performed by the program.

ZOMBIE uses the method of finite differences to discretize the spatial Operators in the partial differential equations. An exponential fitting factor is used to discretize the current relation. For the transient integration a backward differentiation formula of up to 6^{th} order is used including automatic time step and order control.

1.1 Initial Solutions

Several possibilities are offered to obtain the initial solutions.

* the variables can be set to a constant value by the COPY or the VARIABLE command
* the initial profile can be obtained by the Simulation of an ion implantation using the IMPLANT command
* up to ten user provided subroutines PROF* (*=0...9) can be called to compute the initial solution
* the initial solution can be computed from measured profiles stored on files using the GET and the CV command.
* the initial solution can be computed from distributions of other variables using the COPY and CV command.

Any combination of these possibilities can be chosen to determine the initial solution. During the execution of an IMPLANT, a PROFILE, a SOLVE or a TRANSIENT command ZOMBIE uses the specified information to install an optimal spatial grid for the simulation.

1.2 Partial Differential Equations

Equations (1) to (2) denote the system of 'na' coupled partial differential equations which can be solved by ZOMBIE. S_j denotes an independent variable, F_i the corresponding flux, x the depth and t the time.

$$\sum_{j=1}^{na} T_{ij} \cdot \frac{\delta S_j}{\delta t} + \frac{\delta F_i}{\delta x} = G_i \qquad (1)$$

$$F_i = \sum_{j=1}^{na} -D_{ij} \cdot \frac{\delta S_j}{\delta x} + \sum_{j=1}^{na} -U_{ij} \cdot S_j \cdot \frac{\delta \phi}{\delta x} \qquad (2)$$

The functions T, G, D and U may depend on depth, time, any of the variables or any of the specified fluxes as specified by equations (3) to (6).

$$T_{ij} = T_{ij}(x,t) \qquad (3)$$

$$G_i = G_i(x,t,S_k,F_l) \qquad (4)$$

$$D_{ij} = D_{ij}(x,t,S_k,F_l) \qquad (5)$$

$$U_{ij} = U_{ij}(x,t,S_k,F_l) \qquad (6)$$

The electrostatic potential ϕ can be any of the specified variables S_j. It must be set by the VARIABLE command (EL=POTENTIAL).

1.3 Boundary Conditions

Equation (7) to Equation (10) denote 'na' coupled boundary conditions:

$$\sum_{j=1}^{na} A_{ij} \cdot F_j + \sum_{j=1}^{na} B_{ij} \cdot S_j = C_i \qquad (7)$$

$$A_{ij} = A_{ij}(x,t) \qquad (8)$$

$$B_{ij} = B_{ij}(x,t) \qquad (9)$$

$$C_i = C_i(x,t,S_k,F_l) \qquad (10)$$

If the i^{th} differential equation does not have spatial operators (and therefore no boundary conditions) set

$$A_{ii} = 1 \qquad (11)$$

$$A_{ij} = 0 \quad (\text{for } i<>j) \qquad (12)$$

$$B_{ij} = 0 \qquad (13)$$

$$C_i = 0 \qquad (14)$$

If none of the differential equations has a spatial operator set (SPATIAL=.FALSE.) in the TRANSIENT command to speed up the simulation. In this case the subroutines DEF*FL and DEF*BC will not be called.

1.4 Units and Sign Conventions

The program performs no scaling of the partial differential equations. Therefore the user has to provide that no overflow occurs during the computation. Since most of the Computers can handle real numbers in an absolute range from 10^{-38} to 10^{+38} the values of S_j and F_j should not enceed the range from 10^{-19} to 10^{+19}. The implemented models for process and device simulation use the units [µm, sec, A, K] instead of [cm, sec, A, K].

The sign of the fluxes F_j is denoted positive with increasing x-values. At a lower boundary a positive flux is flowing into the simulation domain, at an upper boundary a positive flux is flowing out of the simulation domain.

2. C O M M A N D S

This chapter gives a short summary about the possible commands which can be specified in the INPUT-DECK of ZOMBIE. If the short explanation of keys specified in chapter 3 does not seem to be sufficient additional information will be given. The commands are listed in functional groups. An alphabetical listing of all commands with all possible keys can be found in chapter 3.

Specification of a Simulation Domain

LAYER-command:

It is used to set the lower and the upper boundary of the spatial Simulation domain. The specification of the material (MATERIAL-key) affects only the IMPLANT command.

Specification of Variables

VARIABLE-command:

This command specifies the position, kind, range and requested numerical accuracies for a variable during the simulation. If the variable on position PO is specified, it is necessary that the variables on the positions 1 to PO-1 are also specified before an IMPLANTATION, a PROFILE, a SOLVE or a TRANSIENT command is executed.

If the values of the variables can change their sign set (LI=.TRUE.) otherwise set (LI=.FALSE.). This will avoid division by zero during the computation of the spatial and transient discretization errors.

INITIALIZE-command:
The command specifies a variable on a new position with values which have already been specified by a VARIABLE or an INITIALITE command. It can be used to quickly change the ordering of the variables if it should be required. INITIALIZE copies all data set by a VARIABLE-command and the spatial distribution of the variable. It does not copy the flux.

DELETE-command:
It is used to erase a variable specification on a certain position. If the variable on the position PO is deleted all variables on a higher position should also be deleted before executing a TRANSIENT, SOLVE, PROFILE or IMPLANT command.

Setting up a Grid

A grid is set up as soon as a LAYER, an IMPLANT, a PROFILE, a SOLVE or a TRANSIENT command is executed.

A LAYER command sets up an eleven point equidistant initial grid.

An IMPLANT command uses an equidistant eleven point inital grid, the maximum number of specified grid points and enlarges the grid if necessary.

The GR-key in the PROFILE, SOLVE or TRANSIENT command controls how the grid will be modified during the simulation.

GR=RIGID: means that the grid will not be changed during the execution of the command.

GR=ENLARGE: means that grid points will be inserted if it is required. No grid point will be removed.

GR=MODIFY: means that a completely new grid will be set up starting with the equidistant initial grid. This will be done once in a PROFILE command, after every solving during a SOLVE command and during a

TRANSIENT command after a certain number of time steps have elapsed.

Setting up the Initial Solution

VARIABLE-command:
The initial solution can be set to a constant value (VA-key)

COPY-command:
The values of variable can be set to a constant value (VA-key) or the distribution can be computed by a linear combination of existing solutions of other variables.

CV-command:
The solution can be updated from information copied from a position of the internal storage area.

PROFILE/SOLVE/TRANSIENT-command:
The values of a variable can be computed by user/system specified subroutines PROF* (*=0..9), DEVIPR during a PROFILE, a SOLVE or a TRANSIENT command. During the PROFILE, SOLVE and TRANSIENT command the solution can also be updated from information in the storage area (E* and S* keys, *=1..5)

IMPLANT-command:
The initial profile can be computed by simulating an ion implantation.

Solving Differential Equations

TRANSIENT-command:
The transient command is used to solve a (partial) differential equation with transient operators. If you set SPATIAL=.TRUE. the differential equation is a partial differential equation (process simulation, transient device simulation, Example 1, 2, 3 and 5) otherwise it is an

ordinary differential equation of first order (Example 6).

The choice of the first time step can be critical. If the partial differential is a typical diffusion equation without any generation and/or recombination terms the program makes usually a good choice (e.g. Example 3) In all other cases the user should specify the TT-key. If you cannot estimate the initial time step specify a very small one, the program will speed up fast by itself.

SOLVE-command

It is used to solve elliptic differntial equations of second order. Device Simulation is a typical application for this case (Example 4A, 4B and 4C). If you set up your own models, check the number of Newton iterations necessary to solve the nonlinear equation System. Provided that all derivatives can be computed exactly the number of iterations should usually not exceed the value 3 to 5. Furthermore the L2-norm of the "Right-Hand-Side" should decrease superlinear.

Printing Information

PRINT-command:

The PRINT command is used to write results to output files. The information can be written to the standard result file (default: unit=4) or to one of the data output files (unit=11..20). (The unit numbers depend upon installation) Every PRINT command writes the following record:

```
     NC,       NX, PTX(11), PTX(12)
PTX(1),  PTX(2),  PTX(3),  PTX(4),  PTX(5)
PTX(6),  PTX(7),  PTX(8),  PTX(9),  PTX(10)
     7, KSYS(1), KSYS(2), KSYS(3), .......  KSYS(NC)
  X(1),  S(1,1),  S(2,1),  S(3,1), .......   S(NC,1)
    .        .        .        .   .......      .
    .        .        .        .   .......      .
    .        .        .        .   .......      .
    .        .        .        .   .......      .
 X(NX), S(1,NX), S(2,NX), S(3,NX), .......   S(NC,NX)
```

NC is the number of columns, NX is the number of spatial grid points. KSYS is the number which was assigned to a variable by the program in the VARIABLE command. X contains the spatial grid. If S denotes a distribution of a variable, KSYS is a positive integer otherwise S denotes a flux of the variable.

Plotting of Results

PLOT-command:

It is used to print graphs to the standard result file (unit=4) or to one of the data output files (unit=11..20). These unit numbers depend upon installation (see subroutine ZZZZOC, variables IIIIII(21) and IIIIII(22)).

Copying Information Within the Program

COPY-command:

Copies values of the distribution of a variable on position 'PS' to the position 'PD'. The final distribution can be a linear combination of the two distributions.

$$S_{pd} = fd \cdot S_{pd} + fs \cdot S_{ps} + va$$

CS-command:

Copies values of a distribution of a variable from position 'PV' to the position 'PS' in the storage area. The final distribution on the variable can be a linear combination of

the distribution on postion 'PV' and the old distribution on postion 'PS'.

$$S_{ps} = fv \cdot S_{pv} + fs \cdot S_{ps} + va$$

CV-command:
Copies values of a distribution from position 'PS' in the storage area to the position 'PV' of a variable. The final distribution on the variable can be a linear combination of the distribution on postion 'PS' and the old distribution on postion 'P'.

$$S_{pv} = fv \cdot S_{pv} + fs \cdot S_{ps} + va$$

Linear interpolation applies if the spatial grids of the two distributions are different.

Storing Data On and Reading Data From External Files

SAVE-command:
It is used to write data on the storage area to a specified output file. The output format is:

```
    2,        NX, PTX(11),  0.
    7, KSYS(IE)
 X(1),  S(IE,1)


 .      .
X(NX), S(IE,NX)
```

GET-command:
It is used to read data written by a SAVE-command. If the number of grid points in the data file exceeds the actual memory in the program, the first nx-1 and the last grid point are read and a warning is issued.

Commands Specifying Loops

A loop starts with the LOOP*-command an ends with a ELOOP*-command. The values of * (*=0..9) must coincide. Up to 10 loops can be active simultaneously.
Example:

```
LOOP5
...
LOOP2
...
...
ELOOP2
...
LOOP7
...
ELOOP7
...
ELOOP5
```

The loop Parameter is given to the user provided routines via the PTX array. **(X1=PTX(1), X2=PTX(2), ... X9=PTX(9), X0=PTX(10)).**

End of the INPUT-DECK

END-command:
This command must be the last command in the INPUT-DECK.

3. Alphabetic Listing of Commands and Keys

The following pages contain a complete alphabetic listing of all commands and keys which can be specified within the INPUT-DECK of ZOMBIE 1.0.

The command names can be abbreviated in the INPUT-DECK but they must be unique within all possible command names. (e.g. PRO denotes the PROFILE command uniquely, but PR could denote the PRINT as well as the PROFILE command)

For each command all possible keys, their characteristics and meanings are listed in eight columns. The eight columns contain the following information:

* name of the key: up to 10 characters can be specified. The name of the key can be abbreviated but it must be uniquely specified within the command.
* obligatory or optional key:
  T:=the key must be set otherwise a fatal error message is issued.
  F:=the key can be set otherwise the default value is assumed.
* the type of the value of the "key=value" sequence is specified.
  'S ' or 'L ': value is of type logical (boolean) (.TRUE., .FALSE., .ON., .OFF., .YES., .NO.)
  ' S ' or ' I ': value is of type integer (e.g. 1, 0, -14)
  ' S ' or ' R ': value is of type real (e.g. 1, 1.4, -1.E+13, 1E4 (upper case letters only))
  ' S' or ' C': value is of type character (any character except '*', ';', ',', ' '. Up to 63 characters can be set)

' S C' means that the key may have an integer or a character value. The default value is of type integer. (e.g. X1=0. is default. The user may also specify X1=PARA7)

* default logical (boolean) value
* default integer value
* default real value
* default character value (the first 30 of 63 characters are specified only)
* a short descritpion of the meaning of the key

KEYS OF THE 'COPY " COMMAND IN ALPHABETICAL ORDER

KEY-NAHE	OBL	TYPE	LV	IV	RV	CV	MEANING
FD	F	S	F	0	0.00E+00		VALUE OF "FD" IN THE TRANSFOWTION: S(PD)=FD*S(PD)+FS*S(PS)+VA
FS	F	S	F	0	0.10E+01		VALUE OF "FS" IN THE TRANSFORWTION: S(PD)=FD*S(PD)+FS*S(PS)+VA
PD	T	I	F	0	0.00E+00		VALUE OF "PO" IN THE TRANSFORWTION: S(PD)=FD*S(PD)+FS*S(PS)+VA
PS	F	S	F	0	0.00E+00		VALUE OF "PS" IN THE TRANSFOWTION: S(PD)=FD*S(PD)+FS*S(PS)+VA
VA	F	S	F	0	0.00E+00		VNUE OF "VA" IN THE TRANSFORWTION: S(PD)=FD*S(PD)+FS*S(PS)+VA

KEYS OF THE "CS " COMMAND IN ALPHABETICAL ORDER.

KEY-NAHE	OBL	TYPE	LV	IV	RV	CV	HEANING
FS	F	S	F	0	0.00E+00		VALUE OF "FS" IN THE TRANSFORMATION: S(PS)=FV*S(PV)+FS*S(PS)+VA
FV	F	S	F	0	0.10E+01		VALUE OF "FV" IN THE TRANSFOWTION: S(PS)=FV*S(PV)+FS*S(PS)+VA
PS	T	I	F	0	0.00E+00		VALUE OF "PS" IN THE TRANSFOFMTION: S(PS)=FV*S(PV)+FS*S(PS)+VA
PV	T	I	F	0	0.00E+00		VALUE OF "PV" IN THE TRANSFORMATION: S(PS)=FV*S(PV)+FS*S(PS)+VA
VA	F	S	F	0	0.00E+00		VALUE OF "VA" IN THE TRANSFOWTION: S(PS)=FV*S(PV)+FS*S(PS)+VA

KEYS OF THE "CV " COMMAND IN ALPHABETICAL ORDER

KEY-NAHE	OBL	TYPE	LV	IV	RV	CV	MEANING
FS	F	S	F	0	0.10E+01		VALUE OF "FS" IN THE TRANSFORMATION: S(PV)=FV*S(PV)+FS*S(PS)+VA
FV	F	S	F	0	0.00E+00		VALUE OF "FV" IN THE TRANSFOFMTION: S(PV)=FV*S(PV)+FS*S(PS)+VA
PS	T	I	F	0	0.00E+00		VALUE OF 'PS" IN THE TRANSFOFMTION: S(PV)=FV*S(PV)+FS*S(PS)+VA
PV	T	I	F	0	0.00E+00		VALUE OF "PV" IN THE TRANSFOFMTION: S(PV)=FV*S(PV)+FS*S(PS)+VA
VA	F	S	F	0	0.00E+00		VALUE OF "VA" IN THE TRANSFORMATION: S(PV)=FV*S(PV)+FS*S(PS)+VA

KEYS OF THE "DELETE " COMMAND IN ALPWETICAL ORDER

KEY-NAHE	OBL	TYPE	LV	IV	RV	CV	MEANING
PO	T	I	F	0	0.00E+00		POSITION OF THE VARIABLE TO BE DELETED FOR SIMULATION

NO KEYS IN THE "ELOOP* " COMMANDS (-0. .9)

HO KEYS IN THE "END " COMMAND

KEYS OF THE "EXCHANGE " COMMAND IN ALPWETICAL ORDER.

KEY-NAME	OBL	TYPE	LV	IV	RV	CV	NEANING
PA	T	I	F	0	0.00E+00		POSITION OF A VARIABLE TO BE EXCHANGED: S(PA) <--> S(PB)
PB	T	I	F	0	0.00E+00		POSITION OF A VARIABLE TO BE EXCHANGED: S(PA) <--> S(PB)

KEYS OF TIIE 'GET " COMMAND IN ALPHABETICAL ORDER.

KEY-NAME	OBL	TYPE	LV	IV	RV	CV	MEANING
CYCLE	T	I	F	0	0.00E+00		SPECIFIES THE CYCLE NUMBER TO BE READ FROM THE DATA FILE
IFILE	T	I	F	0	0.00E+00		SPECIFIES THE TAPE NUMBER OF THE INPUT FILE
POSITION	T	I	F	0	0.00E+00		THE POSITION TO WHICH OATA ARE READ INTO THE STORAGE AREA

KEYS OF THE "IMPLANT " COMMAND IN ALPHABETICAL ORDER.

KEY-NAME	OBL	TYPE	LV	IV	RV	CV	MEANING
DOSIS	T	R	F	0	0.00E+00		SPECIFIES THE IMPLANTED DOSIS - UNITS IN [MICROMETERS**-2]
ELEMENT	T	C	F	0	0.00E+00		SPECIFIES THE ELEMENT HHICH WILL BE IMPLANTED
ENERGY	T	R	F	0	0.00E+00		SPECIFIES THE IMPLANTATION ENERGY - UNITS IN [KEV]
FRONTSIOE	F	S	T	0	0.00E+00		.TRUE.-.FALSE.: IMPLANTATION IS OONE FROM THE FRONT-BACK SIOE
STATISTICS	F	S	F	0	0.00E+00	LSS	SPECIFIES THE DISTRIBUTION STATISTICS TO BE USED (E.G. LSS)
TYPE	F	S	F	0	0.00E+00	PEARSON4	SPECIFIES THE TYPE OF THE DISTRIBUTION FUNCTION (E.G. PEARSON4)

KEYS OF THE "INITIALIZE" COMMAND IN ALPHABETICAL ORDER.

KEY-NAME	OBL	NPE	LV	IV	RV	CV	MEANING
PD	T	I	F	0	0.00E+00		POSITION OF THE MSTINATION-VARIABLE TO BE COPIEO: S(PD)=S(PS)
PS	T	I	F	0	0.00E+00		POSITION OF THE SOURCE-VARIABLE TO BE COPIED : S(PD)=S(PS)

KEYS OF THE 'LAYER " COMMAND IN ALPHABETICAL ORDER.

KEY-NAHE	OBL	TYPE	LV	IV	RV	CV	MEANING
LBOUNDARY	T	R	F	0	0.00E+00		LOWER BOUNOARY OF THE SPECIFIED LAYER
MATERIAL	F	S	F	0	0.00E+00	SILICON	UATERIAL OF THE SPECIFIED LAYER
UBOUNDARY	T	R	F	0	0.00E+00		UPPER BOUNDARY OF THE SPECIFIEO LAYER

KEYS OF THE "LOOP* " COMMANDS IN ALPHABETICAL ORDER. (*=0..9)

KM-NAHE	OBL	TYPE	LV	IV	RV	CV	MEANING
ET	T	R	F	0	0.00E+00		SPECIFIES THE FINAL VALUE OF THE LOOP PARAMETER
NL	T	I	F	0	0.00E+00		SPECIFIES THE NUMBER OF STEPS BETHEEN -ST- AND -ET- IN THE LWP
ST	F	S	F	0	0.00E+00		SPECIFIES THE INITIAL VALUE OF THE LOOP PARAMETER

KEYS OF THE 'PLOT " COMMAND IN ALPHABETICAL ORDER

KEY-NAME	OBL	TYPE	LV	IV	RV	CV	MEANING
LONGX-AXIS	F	S	T	0	0.00E+00		T OR F: THE PICTURE IS PLOTTED HITH A 60 OR 24 CHAR/LINE X-AXIS
LONGYZAXIS	F	S	T	0	0.00E+00		TR OR FA:THE PICTURE IS PLOTTEO ON A 132 OR 80 CHAR/LINE DEVICE
OFILE	F	S	T	4	0.00E+00		SPECIFIES THE OUTPUT FILE TO WHICH THE FIGURE WILL BE WRITTEN
PO	F	I	F	0	0.00E+00		SPECIFIES A POSITION OF A VARIABLE OR A FLUX TO BE PLOTTEO
P1	F	I	F	0	0.00E+00		SPECIFIES A POSITION OF A VARIABLE OR A FLUX TO BE PLOTTED
P2	F	I	F	0	0.00E+00		SPECIFIES A POSITION OF A VARIABLE OR A FLUX TO BE PLOTTEO
P3	F	I	F	0	0.00E+00		SPECIFIES A POSITION OF A VARIABLE OR A FLUX TO BE PLOTTEO
P4	F	I	F	0	0.00E+00		SPECIFIES A POSITION OF A VARIABLE OR A FLUX TO BE PLOTTED
P5	F	I	F	0	0.00E+00		SPECIFIES A POSITION OF A VARIABLE OR A FLUX TO BE PLOTTEO
P6	F	I	F	0	0.00E+00		SPECIFIES A POSITION OF A VARIABLE OR A FLUX TO BE PLOTTEO
P7	F	I	F	0	0.00E+00		SPECIFIES A POSITION OF A VARIABLE OR A FLUX TO BE PLOTTED
P8	F	I	F	0	0.00E+00		SPECIFIES A POSITION OF A VARIABLE OR A FLUX TO BE PLOTTEO
P9	F	I	F	0	0.00E+00		SPECIFIES A POSITION OF A VARIABLE OR A FLUX TO BE PLOTTED
VO	F	S	T	0	0.00E+00		.TRUE./.FALSE. MEANS THAT THE PO-KEY SPECIFIES A VARIABLEIFLUX
V1	F	S	T	0	0.00E+00		.TRUE./.FALSE. MEANS THAT THE P1-KEY SPECIFIES A VARIABLEIFLUX
V2	F	S	T	0	0.00E+00		.TRUE./.FALSE. MEANS THAT THE P2-KEY SPECIFIES A VARIABLE/FLUX
V3	F	S	T	0	0.00E+00		.TRUE./.FALSE. MEANS THAT THE P3-KEY SPECIFIES A VARIABLEIFLUX
V4	F	S	T	0	0.00E+00		.TRUE./.FALSE. MEANS THAT THE P4-KEY SPECIFIES A VARIABLE/FLUX
V5	F	S	T	0	0.00E+00		.TRUE./.FALSE. MEANS THAT THE P5-KEY SPECIFIES A VARIABLEIFLUX
V6	F	S	T	0	0.00E+00		.TRUE./.FALSE. MEANS THAT THE P6-KEY SPECIFIES A VARIABLEIFLUX
V7	F	S	T	0	0.00E+00		.TRUE./.FALSE. MEANS THAT M E P7-KEY SPECIFIES A VARIABLE/FLUX
V8	F	S	T	0	0.00E+00		.TRUE./.FALSE. MEANS THAT THE P8-KEY SPECIFIES A VARIABLE/FLUX
V9	F	S	T	0	0.00E+00		.TRUE./.FALSE. MEANS THAT THE P9-KEY SPECIFIES A VARIABLEIFLUX
XB	F	S	F	0	0.00E+00		VALUE CORRESPONDING TO THE BEGIN OF THE X-AXIS
XE	F	S	F	0	0.20E+01		VALUE CORRESPONDING TO THE END OF THE X-AXIS
XL	F	S	T	0	0.00E+00		.TRUE./.FALSE. MEANS THE X-SCALE IS A LINEAR/LOGARITHMIC ONE
XH	F	S	F	2	0.00E+00		NUMBER OF UAIN INTERVALLS ON THE X-AXIS
XT	F	S	F	0	0.00E+00DEPTH [MICROMETERS]		TEXT WRITTEN ON THE X-AXIS
YO	F	S	T	0	0.00E+00		.TRUE./.FALSE. UEANS THAT THE PO-KEY RELATES TO THE YIZ-AXIS
Y1	F	S	T	0	0.00E+00		.TRUE./.FALSE. MEANS THAT THE P1-KEY RELATES TO THE YIZ-AXIS
Y2	F	S	T	0	0.00E+00		.TRUE./.FALSE. MEANS THAT THE P2-KEY RELATES TO THE Y/Z-AXIS
Y3	F	S	T	0	0.00E+00		.TRUE./.FALSE. MEANS THAT M E P3-KEY RELATES TO THE YIZ-AXIS
Y4	F	S	T	0	0.00E+00		.TRUE./.FALSE. UEANS THAT THE P4-KEY RELATES TO THE YIZ-AXIS
Y5	F	S	T	0	0.00E+00		.TRUE./.FALSE. MEANS THAT THE PS-KEY RELATES TO THE Y/Z-AXIS
Y6	F	S	T	0	0.00E+00		.TRUE./.FALSE. MEANS THAT THE P6-KEY RELATES TO THE Y/Z-AXIS
Y7	F	S	T	0	0.00E+00		.TRUE./.FALSE. MEANS THAT THE P7-KEY REIATES TO THE YIZ-AXIS
Y8	F	S	T	0	0.00E+00		.TRUE./.FALSE. MEANS THAT THE PB-KEY RELATES TO M E Y/Z-AXIS
Y9	F	S	T	0	0.00E+00		.TRUE./.FALSE. MEANS THAT M E P9-KEY RELATES TO THE YIZ-AXIS
YB	F	S	F	0	0.10E+01		VALUE CORRESPONDING TO THE BEGIN OF THE Y-AXIS
YE	F	S	F	0	0.10E+10		VALUE CORRESWNDING TO THE END OF THE Y-AXIS
YL	F	S	F	0	0.00E+00		.TRUE./.FALSE. MEANS THE Y-SCALE IS A LINEAR/LOGARITHMIC ONE
YM	F	S	F	9	0.00E+00		NUMBER OF UAIN INTERVALLS ON THE Y-AXIS
YT	F	S	F	0	0.00E+00CONCENTRATION [ATOMS PER CUBE		TEXT WRITTEN ON THE Y-AXIS
ZB	F	S	F	0	0.10E+13		VALUE CORRESPONDING TO THE BEGIN OF THE Z-AXIS
ZE	F	S	F	0	0.10E+22		VALUE CORRESPONDING TO THE END OF THE Z-AXIS
ZL	F	S	F	0	0.00E+00		.TRUE./.FALSE. MEANS THE Z-SCALE IS A LINEAR/LOGARITHMIC ONE
ZM	F	S	F	9	0.00E+00		NUMBER OF MAIN INTERVALLS ON THE Z-AXIS
ZT	F	S	F	0	0.00E+00CONCENTRATION [ATOMS PER CUBE		TEXT WRITTEN ON THE Z-AXIS

KEYS OF THE "PRINT COMMAND IN ALPHABETICAL ORDER

KEY-NAHE	OBL	NPE	LV	IV	RV	CV	MEANING
COMMENT	F	S	F	0	0.00E+00		SPECIFIES THE COMMENT MITTEN TO THE OUTPUT FILE
FBEGIN	F	S	F	1	0.00E+00		SPECIFIES THE FIRST FLUX PRINTED TO THE OUTPUT FILE
FEND	F	S	F	0	0.00E+00		SPECIFIES THE LAST FLUX PRITNED TO THE OUTPUT FILE
OFILE	F	S	F	4	0.00E+00PRINTED DISTRIBUTION OF VARIAB		SPECIFIES THE TAPE NUMBER FOR THE OUTPUT FILE (OF=4 OR 11..20)
VBEGIN	F	S	F	1	0.00E+00		SPECIFIES THE FIRST VARIABLE PRINTED TO THE OUTPUT FILE
VENO	F	S	F	0	0.00E+00		SPECIFIES THE LAST VARIABLE PRINTED TO THE OUTPUT FILE

KEYS OF THE "PROFILE " COMMAND IN ALPHABETICAL ORDER.

KEY-NAME	OBL	TYPE	LV	IV	RV	CV	MEANING
OP	F	S	F	0	0.00E+00		CALL "DEVIPR" TO GET THE INITIAL SOLUTION (PSI-N-P) FOR A DIODE
E1	F	I	F	0	0.00E+00		COPY STORED VARIABLE ON POSITION S1 TO POSITION E1: S(S1)=E(E1)
E2	F	I	F	0	0.00E+00		COPY STORED VARIABLE ON POSITION S2 TO POSITION E2: S(S2)=E(E2)
E3	F	I	F	0	0.00E+00		COPY STORED VARIABLE ON POSITION S3 TO POSITION E3: S(S3)=E(E3)
E4	F	I	F	0	0.00E+00		COPY STORED VARIABLE ON POSITION S4 TO POSITION E4: S(S4)=E(E4)
E5	F	I	F	0	0.00E+00		COPY STORED VARIABLE ON POSITION S5 TO POSITION E5: S(S5)=E(E5)
GR	F	S	F	0	0.00E+00	MODIFY	SPECIFIES POSSIBLE GRID MODIFICATIONS: RIGID, ENLARGE, MODIFY
MG	F	S	F	0	0.00E+00		SPECIFIES THE MINIMUM GRID WIDTH FOR ADDITIONAL GRID REFINEMENT
NG	F	S	F	11	0.00E+00		SPECIFIES THE NUMBER OF EQUIDISTANT POINTS IN THE INITIAL GRID
NX	F	S	F	201	0.00E+00		SPECIFIES THE MAXIMUM NUMBER OF GRID POINTS DURING THIS COMMAND
P0	F	S	F	0	0.00E+00		CALL THE USER PROVIDED SUBROUTINE "PROF0"
P1	F	S	F	0	0.00E+00		CALL THE USER PROVIDED SUBROUTINE "PROF1"
P2	F	S	F	0	0.00E+00		CALL THE USER PROVIDED SUBROUTINE "PROF2"
P3	F	S	F	0	0.00E+00		CALL THE USER PROVIDED SUBROUTINE "PROF3"
P4	F	S	F	0	0.00E+00		CALL THE USER PROVIDED SUBROUTINE "PROF4"
P5	F	S	F	0	0.00E+00		CALL THE USER PROVIDED SUBROUTINE "PROF5"
P6	F	S	F	0	0.00E+00		CALL THE USER PROVIDED SUBROUTINE "PROF6"
P7	F	S	F	0	0.00E+00		CALL THE USER PROVIDED SUBROUTINE "PROF7"
P8	F	S	F	0	0.00E+00		CALL THE USER PROVIDED SUBROUTINE "PROF8"
P9	F	S	F	0	0.00E+00		CALL THE USER PROVIDED SUBROUTINE "PROF9"
PP	F	S	F	0	0.00E+00		CALL "PROCPR" TO GET THE INITIAL SOLUTION (PSI) FOR PROCESS SIM
S1	F	I	F	0	0.00E+00		COPY STORED VARIABLE ON POSITION S1 TO POSITION E1: S(S1)=E(E1)
S2	F	I	F	0	0.00E+00		COPY STORED VARIABLE ON POSITION S2 TO POSITION E2: S(S2)=E(E2)
S3	F	I	F	0	0.00E+00		COPY STORED VARIABLE ON POSITION S3 TO POSITION E3: S(S3)=E(E3)
S4	F	I	F	0	0.00E+00		COPY STORED VARIABLE ON POSITION S4 TO POSITION E4: S(S4)=E(E4)
S5	F	I	F	0	0.00E+00		COPY STORED VARIABLE ON POSITION S5 TO POSITION E5: S(S5)=E(E5)
ST	F	S	F	0	0.00E+00		SIMULATION-START-TIME GIVEN TO THE USER SUBROUTINES BY PTX(11)
TE	T	RC	F	0	0.00E+00		SPECIFIES THE VALUE OF THE TEMPERATURE GIVEN TO FUNCTION TEMPER
U0	F	R	F	0	0.00E+00		THE 1-ST INTERVAL BOUNDARY FOR A USER REFINED GRID
U1	F	R	F	0	0.00E+00		THE 2-ND INTERVAL BOUNDARY FOR A USER REFINED GRID
U2	F	R	F	0	0.00E+00		THE 3-RD INTERVAL BOUNDARY FOR A USER REFINED GRID
U3	F	R	F	0	0.00E+00		THE 4-TH INTERVAL BOUNDARY FOR A USER REFINED GRID
U4	F	R	F	0	0.00E+00		THE 5-TH INTERVAL BOUNDARY FOR A USER REFINED GRID
U5	F	R	F	0	0.00E+00		THE 6-TH INTERVAL BOUNDARY FOR A USER REFINED GRID
U6	F	R	F	0	0.00E+00		THE 7-TH INTERVAL BOUNDARY FOR A USER REFINED GRID
U7	F	R	F	0	0.00E+00		THE 8-TH INTERVAL BOUNDARY FOR A USER REFINED GRID
U8	F	R	F	0	0.00E+00		THE 9-TH INTERVAL BOUNDARY FOR A USER REFINED GRID
U9	F	R	F	0	0.00E+00		THE 10-TH INTERVAL BOUNDARY FOR A USER REFINED GRID
X0	F	SC	F	0	0.00E+00		SETS THE VALUE OF THE PARAMETER PTX(10) IN THE USER SUBROUTINES
X1	F	SC	F	0	0.00E+00		SETS THE VALUE OF THE PARAMETER PTX(1) IN THE USER SUBROUTINES
X2	F	SC	F	0	0.00E+00		SETS THE VALUE OF THE PARAMETER PTX(2) IN THE USER SUBROUTINES
X3	F	SC	F	0	0.00E+00		SETS THE VALUE OF THE PARAMETER PTX(3) IN THE USER SUBROUTINES
X4	F	SC	F	0	0.00E+00		SETS THE VALUE OF THE PARAMETER PTX(4) IN THE USER SUBROUTINES
X5	F	SC	F	0	0.00E+00		SETS THE VALUE OF THE PARAMETER PTX(5) IN THE USER SUBROUTINES
X6	F	SC	F	0	0.00E+00		SETS THE VALUE OF THE PARAMETER PTX(6) IN THE USER SUBROUTINES
X7	F	SC	F	0	0.00E+00		SETS THE VALUE OF THE PARAMETER PTX(7) IN THE USER SUBROUTINES
X8	F	SC	F	0	0.00E+00		SETS THE VALUE OF THE PARAMETER PTX(8) IN THE USER SUBROUTINES
X9	F	SC	F	0	0.00E+00		SETS THE VALUE OF THE PARAMETER PTX(9) IN THE USER SUBROUTINES

KEYS OF THE "SAVE " COMMAND IN ALPHABETICAL ORDER.

KEY-NAME	OBL	TYPE	LV	IV	RV	CV	MEANING
OFILE	T	I	F	0	0.00E+00		SPECIFIES THE TAPE NUMBER OF THE OUPUT FILE
POSITION	T	I	F	0	0.00E+00		SPECIFIES THE POSITION OF THE STORED VARIABLE TO BE SAVED
REWINDFILE	T	L	F	0	0.00E+00		THE OUTPUT FILE IS REWOUND (TRUE) OR NOT (FALSE) BEFORE WRITING

Alphabetic Listing of Commands and Keys

KEYS OF THE "SETPROCESS" COMMAND IN ALPHABETICAL ORDER.

KEY-NAME	OBL	TYPE	LV	IV	RV	CV	MEANING
A1	F	S	F	0	0.10E+01		LOWER BOUNDARY : BORON : VALUE OF "A" IN "A*F+B*S=C+D*(S-E)"
A2	F	S	F	0	0.10E+01		LOWER BOUNDARY : PHOSPHOR: VALUE OF "A" IN "A*F+B*S=C+D*(S-E)"
A3	F	S	F	0	0.10E+01		LOWER BOUNDARY : ANTIMONY: VALUE OF "A" IN "A*F+B*S=C+D*(S-E)"
A4	F	S	F	0	0.10E+01		LOWER BOUNDARY : ARSENIC : VALUE OF "A" IN "A*F+B*S=C+D*(S-E)"
B1	F	S	F	0	0.00E+00		LOWER BOUNDARY : BORON : VALUE OF "B" IN "A*F+B*S=C+D*(S-E)"
B2	F	S	F	0	0.00E+00		LOWER BOUNDARY : PHOSPHOR: VALUE OF "B" IN "A*F+B*S=C+D*(S-E)"
B3	F	S	F	0	0.00E+00		LOWER BOUNDARY : ANTIMONY: VALUE OF "B" IN "A*F+B*S=C+D*(S-E)"
B4	F	S	F	0	0.00E+00		LOWER BOUNDARY : ARSENIC : VALUE OF "B" IN "A*F+B*S=C+D*(S-E)"
C1	F	S	F	0	0.00E+00		LOWER BOUNDARY : BORON : VALUE OF "C" IN "A*F+B*S=C+D*(S-E)"
C2	F	S	F	0	0.00E+00		LOWER BOUNDARY : PHOSPHOR: VALUE OF "C" IN "A*F+B*S=C+D*(S-E)"
C3	F	S	F	0	0.00E+00		LOWER BOUNDARY : ANTIMONY: VALUE OF "C" IN "A*F+B*S=C+D*(S-E)"
C4	F	S	F	0	0.00E+00		LOWER BOUNDARY : ARSENIC : VALUE OF "C" IN "A*F+B*S=C+D*(S-E)"
D1	F	S	F	0	0.00E+00		LOWER BOUNDARY : BORON : VALUE OF "D" IN "A*F+B*S=C+D*(S-E)"
D2	F	S	F	0	0.00E+00		LOWER BOUNDARY : PHOSPHOR: VALUE OF "D" IN "A*F+B*S=C+D*(S-E)"
D3	F	S	F	0	0.00E+00		LOWER BOUNDARY : ANTIMONY: VALUE OF "D" IN "A*F+B*S=C+D*(S-E)"
D4	F	S	F	0	0.00E+00		LOWER BOUNDARY : ARSENIC : VALUE OF "D" IN "A*F+B*S=C+D*(S-E)"
E1	F	S	F	0	0.00E+00		LOWER BOUNDARY : BORON : VALUE OF "E" IN "A*F+B*S=C+D*(S-E)"
E2	F	S	F	0	0.00E+00		LOWER BOUNDARY : PHOSPHOR: VALUE OF "E" IN "A*F+B*S=C+D*(S-E)"
E3	F	S	F	0	0.00E+00		LOWER BOUNDARY : ANTIMONY: VALUE OF "E" IN "A*F+B*S=C+D*(S-E)"
E4	F	S	F	0	0.00E+00		LOWER BOUNDARY : ARSENIC : VALUE OF "E" IN "A*F+B*S=C+D*(S-E)"
V1	F	S	F	0	0.10E+01		UPPER BOUNDARY : BORON : VALUE OF "V" IN "V*F+W*S=X+Y*(S-Z)"
V2	F	S	F	0	0.10E+01		UPPER BOUNDARY : PHOSPHOR: VALUE OF "V" IN "V*F+W*S=X+Y*(S-Z)"
V3	F	S	F	0	0.10E+01		UPPER BOUNDARY : ANTIMONY: VALUE OF "V" IN "V*F+W*S=X+Y*(S-Z)"
V4	F	S	F	0	0.10E+01		UPPER BOUNDARY : ARSENIC : VALUE OF "V" IN "V*F+W*S=X+Y*(S-Z)"
W1	F	S	F	0	0.00E+00		UPPER BOUNDARY : BORON : VALUE OF "W" IN "V*F+W*S=X+Y*(S-Z)"
W2	F	S	F	0	0.00E+00		UPPER BOUNDARY : PHOSPHOR: VALUE OF "W" IN "V*F+W*S=X+Y*(S-Z)"
W3	F	S	F	0	0.00E+00		UPPER BOUNDARY : ANTIMONY: VALUE OF "W" IN "V*F+W*S=X+Y*(S-Z)"
W4	F	S	F	0	0.00E+00		UPPER BOUNDARY : ARSENIC : VALUE OF "W" IN "V*F+W*S=X+Y*(S-Z)"
X1	F	S	F	0	0.00E+00		UPPER BOUNDARY : BORON : VALUE OF "X" IN "V*F+W*S=X+Y*(S-Z)"
X2	F	S	F	0	0.00E+00		UPPER BOUNDARY : PHOSPHOR: VALUE OF "X" IN "V*F+W*S=X+Y*(S-Z)"
X3	F	S	F	0	0.00E+00		UPPER BOUNDARY : ANTIMONY: VALUE OF "X" IN "V*F+W*S=X+Y*(S-Z)"
X4	F	S	F	0	0.00E+00		UPPER BOUNDARY : ARSENIC : VALUE OF "X" IN "V*F+W*S=X+Y*(S-Z)"
Y1	F	S	F	0	0.00E+00		UPPER BOUNDARY : BORON : VALUE OF "Y" IN "V*F+W*S=X+Y*(S-Z)"
Y2	F	S	F	0	0.00E+00		UPPER BOUNDARY : PHOSPHOR: VALUE OF "Y" IN "V*F+W*S=X+Y*(S-Z)"
Y3	F	S	F	0	0.00E+00		UPPER BOUNDARY : ANTIMONY: VALUE OF "Y" IN "V*F+W*S=X+Y*(S-Z)"
Y4	F	S	F	0	0.00E+00		UPPER BOUNDARY : ARSENIC : VALUE OF "Y" IN "V*F+W*S=X+Y*(S-Z)"
Z1	F	S	F	0	0.00E+00		UPPER BOUNDARY : BORON : VALUE OF "Z" IN "V*F+W*S=X+Y*(S-Z)"
Z2	F	S	F	0	0.00E+00		UPPER BOUNDARY : PHOSPHOR: VALUE OF "Z" IN "V*F+W*S=X+Y*(S-Z)"
Z3	F	S	F	0	0.00E+00		UPPER BOUNDARY : ANTIMONY: VALUE OF "Z" IN "V*F+W*S=X+Y*(S-Z)"
Z4	F	S	F	0	0.00E+00		UPPER BOUNDARY : ARSENIC : VALUE OF "Z" IN "V*F+W*S=X+Y*(S-Z)"

KEYS OF THE "SOLVE " COMMAND IN ALPHABETICAL ORDER.

KEY-NAME	OBL	TYPE	LV	IV	RV	CV	MEANING
COMMENT	F	S	F	0	0.00E+00		DEFINES A COMMENT WRITTEN TO THE OUTPUT FILE
OP	F	S	F	0	0.00E+00		CALL "DEVIPR" TO GET THE INITIAL SOLUTION (PSI-N-P) FOR A DIODE
E1	F	I	F	0	0.00E+00		COPY STORED VARIABLE ON POSITION S1 TO POSITION E1: S(S1)-E(E1)
E2	F	I	F	0	0.00E+00		COPY STORED VARIABLE ON POSITION S2 TO POSITION E2: S(S2)-E(E2)
E3	F	I	F	0	0.00E+00		COPY STORED VARIABLE ON POSITION S3 TO POSITION E3: S(S3)-E(E3)
E4	F	I	F	0	0.00E+00		COPY STORED VARIABLE ON POSITION S4 TO POSITION E4: S(S4)-E(E4)
E5	F	I	F	0	0.00E+00		COPY STORED VARIABLE ON POSITION S5 TO POSITION E5: S(S5)-E(E5)
FBEGIN	F	S	F	1	0.00E+00		POSITION OF THE FIRST FLUX TO BE PRINTED
FEND	F	S	F	0	0.00E+00		POSITION OF THE LAST FLUX TO BE PRINTED
GR	F	S	F	0	0.00E+00	MODIFY	SPECIFIES POSSIBLE GRID MODIFICATIONS: RIGID, ENLARGE, MODIFY
KDEUFLHART	F	S	F	10	0.00E+00		MAXIMUM NUMBER OF DEUFELHART DAMPS FOR A NEWTON ITERATION STEP
KGRID	F	S	F	4	0.00E+00		MAXIMUM NUMBER OF GRID UPDATES DURING SOLVING THE PDE-SYSTEM
KNEWTON	F	S	F	20	0.00E+00		MAXIMUM NUMBER OF NEWTON ITERATIONS FOR SOLVING A NONLINEAR SYS
KRANGERJCT	F	S	F	5	0.00E+00		MAXIMUM NUMBER OF RANGE REJECTS (QU:LB-UB) FOR THE SOLUTION

KEYS OF THE "SOLVE " COMMAND IN ALPHABETICAL ORDER (CONTINUED).

KEY-NAME	OBL	TYPE	LV	IV	RV	CV	MEANING
MG	F	S	F	0	0.00E+00		SPECIFIES THE MINIMUM GRID WIDTH FOR ADDITIONAL GRID REFINEMENT
MNUMBER	T	I	F	0	0.00E+00		DEFINES THE NAMES OF THE DEF*** SUBROUTINES WHICH ARE CALLED
MODELNAME	T	C	F	0	0.00E+00		DEFINES THE NAME OF THE PHYSICAL MODEL (MUST FIT WITH "DEF*MO")
NA	T	I	F	0	0.00E+00		DEFINES THE NUMBER OF INDEPENDENT VARIABLES IN THE PDE-SYSTEM
NG	F	S	F	11	0.00E+00		SPECIFIES THE NUMBER OF EQUIDISTANT POINTS IN THE INITIAL GRID
NX	F	S	F	201	0.00E+00		SPECIFIES THE MAXIMUM NUMBER OF GRID POINTS DURING THIS COMMAND
OFILE	F	S	F	0	0.00E+00		UNIT NUMBER FOR THE OUTPUT FILE DURING SOLVE/TRANSIENT COMMANDS
P0	F	S	F	0	0.00E+00		CALL THE USER PROVIDED SUBROUTINE "PROF0"
P1	F	S	F	0	0.00E+00		CALL THE USER PROVIDED SUBROUTINE "PROF1"
P2	F	S	F	0	0.00E+00		CALL THE USER PROVIDED SUBROUTINE "PROF2"
P3	F	S	F	0	0.00E+00		CALL THE USER PROVIDED SUBROUTINE "PROF3"
P4	F	S	F	0	0.00E+00		CALL THE USER PROVIDED SUBROUTINE "PROF4"
P5	F	S	F	0	0.00E+00		CALL THE USER PROVIDED SUBROUTINE "PROF5"
P6	F	S	F	0	0.00E+00		CALL THE USER PROVIDED SUBROUTINE "PROF6"
P7	F	S	F	0	0.00E+00		CALL THE USER PROVIDED SUBROUTINE "PROF7"
P8	F	S	F	0	0.00E+00		CALL THE USER PROVIDED SUBROUTINE "PROF8"
P9	F	S	F	0	0.00E+00		CALL THE USER PROVIDED SUBROUTINE "PROF9"
PP	F	S	F	0	0.00E+00		CALL "PROCPR" TO GET THE INITIAL SOLUTION (PSI) FOR PROCESS SIM
S1	F	I	F	0	0.00E+00		COPY STORED VARIABLE ON POSITION S1 TO POSITION E1: S(S1)=E(E1)
S2	F	I	F	0	0.00E+00		COPY STORED VARIABLE ON POSITION S2 TO POSITION E2: S(S2)=E(E2)
S3	F	I	F	0	0.00E+00		COPY STORED VARIABLE ON POSITION S3 TO POSITION E3: S(S3)=E(E3)
S4	F	I	F	0	0.00E+00		COPY STORED VARIABLE ON POSITION S4 TO POSITION E4: S(S4)=E(E4)
S5	F	I	F	0	0.00E+00		COPY STORED VARIABLE ON POSITION S5 TO POSITION E5: S(S5)=E(E5)
ST	F	S	F	0	0.00E+00		SIMULATION-START-TIME GIVEN TO THE USER SUBROUTINES BY PTX(11)
TE	T	RC	F	0	0.00E+00		SPECIFIES THE VALUE OF THE TEMPERATURE GIVEN TO FUNCTION TEMPER
U0	F	R	F	0	0.00E+00		THE 1-ST INTERVAL BOUNDARY FOR A USER REFINED GRID
U1	F	R	F	0	0.00E+00		THE 2-ND INTERVAL BOUNDARY FOR A USER REFINED GRID
U2	F	R	F	0	0.00E+00		THE 3-RD INTERVAL BOUNDARY FOR A USER REFINED GRID
U3	F	R	F	0	0.00E+00		THE 4-TH INTERVAL BOUNDARY FOR A USER REFINED GRID
U4	F	R	F	0	0.00E+00		THE 5-TH INTERVAL BOUNDARY FOR A USER REFINED GRID
U5	F	R	F	0	0.00E+00		THE 6-TH INTERVAL BOUNDARY FOR A USER REFINED GRID
U6	F	R	F	0	0.00E+00		THE 7-TH INTERVAL BOUNDARY FOR A USER REFINED GRID
U7	F	R	F	0	0.00E+00		THE 8-TH INTERVAL BOUNDARY FOR A USER REFINED GRID
U8	F	R	F	0	0.00E+00		THE 9-TH INTERVAL BOUNDARY FOR A USER REFINED GRID
U9	F	R	F	0	0.00E+00		THE 10-TH INTERVAL BOUNDARY FOR A USER REFINED GRID
VBEGIN	F	S	F	1	0.00E+00		POSITION OF THE FIRST VARIABLE TO BE PRINTED
VEND	F	S	F	0	0.00E+00		POSITION OF THE LAST VARIABLE TO BE PRINTED
X0	F	SC	F	0	0.00E+00		SETS THE VALUE OF THE PARAMETER PTX(10) IN THE USER SUBROUTINES
X1	F	SC	F	0	0.00E+00		SETS THE VALUE OF THE PARAMETER PTX(1) IN THE USER SUBROUTINES
X2	F	SC	F	0	0.00E+00		SETS THE VALUE OF THE PARAMETER PTX(2) IN THE USER SUBROUTINES
X3	F	SC	F	0	0.00E+00		SETS THE VALUE OF THE PARAMETER PTX(3) IN THE USER SUBROUTINES
X4	F	SC	F	0	0.00E+00		SETS THE VALUE OF THE PARAMETER PTX(4) IN THE USER SUBROUTINES
X5	F	SC	F	0	0.00E+00		SETS THE VALUE OF THE PARAMETER PTX(5) IN THE USER SUBROUTINES
X6	F	SC	F	0	0.00E+00		SETS THE VALUE OF THE PARAMETER PTX(6) IN THE USER SUBROUTINES
X7	F	SC	F	0	0.00E+00		SETS THE VALUE OF THE PARAMETER PTX(7) IN THE USER SUBROUTINES
X8	F	SC	F	0	0.00E+00		SETS THE VALUE OF THE PARAMETER PTX(8) IN THE USER SUBROUTINES
X9	F	SC	F	0	0.00E+00		SETS THE VALUE OF THE PARAMETER PTX(9) IN THE USER SUBROUTINES
XCRITICAL	F	S	F	0	0.10E+02		CRITICAL PERCENTAGE OF INTERPOLATED POINTS FOR SOLVING AGAIN
ZINCL2NORM	F	S	F	0	0.10E+16		ZERO FOR THE L2-NORM OF THE SCALED INC. "ZERO" = ZINCL2*EPSILON
ZINCLINORM	F	S	F	0	0.10E+16		ZERO FOR THE LI-NORM OF THE SCALED INC. "ZERO" = ZINCLI*EPSILON
ZRHSL2NORM	F	S	F	0	0.13E+03		ZERO FOR THE L2-NORM OF THE SCALED RHS. "ZERO" = ZRHSL2*EPSILON
ZRHSLINORM	F	S	F	0	0.13E+03		ZERO FOR THE LI-NORM OF THE SCALED RHS. "ZERO" = ZRSHLI*EPSILON

KEYS OF THE "TRANSIENT " COMMAND IN ALPHABETICAL ORDER

KEY-NAME	OBL	TYPE	LV	IV	RV	CV	MEANING
COMMENT	F	S	F	0	0.00E+00		DEFINES A COMMENT WRITTEN TO THE OUTPUT FILE
DP	F	S	F	0	0.00E+00		CNL "DEVIPR" TO GET THE INITIAL SOLUTION (PSI-N-P) FOR A DIODE
E1	F	I	F	0	0.00E+00		COPY STOREO VARIABLE ON POSITION S1 TO POSITION E1: S(S1)=E(E1)
E2	F	I	F	0	0.00E+00		COPY STORED VARIABLE ON POSITION S2 TO POSITION E2: S(S2)=E(E2)
E3	F	I	F	0	0.00E+00		COPY STCRED VARIABLE OII POSITION S3 TO POSITION E3: S(S3)=E(E3)
E4	F	I	F	0	0.00E+00		COPY STORED VARIABLE ON POSITION S4 TO POSITION E4: S(S4)=E(E4)
E5	F	I	F	0	0.00E+00		COPY STCRED VARIABLE ON POSITION S5 TO POSITION E5: S(S5)=E(E5)
ETIME	T	R	F	0	0.00E+00		END TIME FOR THE TRANSIENT INTEGRATION OF THE SYSTEM OF POE
FBEGIN	F	S	F	1	0.00E+00		POSITION OF THE FIRST FLUX TO BE PRINTED
FENO	F	S	F	0	0.00E+00		POSITION OF THE LAST FLUX TO BE PRINTED
GR	F	S	F	0	0.00E+00	MODIFY	SPECIFIES POSSIBLE GRID MODIFICATIONS: RIGID, ENLARGE, MODIFY
KDEUFLHART	F	S	F	10	0.00E+00		MAXIMUM NUMBER OF DEUFELHART DAMPS FOR A NEWTON ITERATION STEP
KGRID	F	S	F	4	0.00E+00		MAXIMUM NUMBER OF GRID UPDATES DURING SOLVING THE PDE-SYSTEM
KNEHTON	F	S	F	20	0.00E+00		MAXIMUM NUMBER OF NEWTON ITERATIONS FOR SOLVING A NONLINEPR SYS
KRANGERJCT	F	S	F	5	0.00E+00		MAXIMUM NUMBER OF RANGE REJECTS (QU:LB-UB) FOR THE SOLUTION
MG	F	S	F	0	0.00E+00		SPECIFIES THE MINIMUM GRID HIDTH FOR ADDITIONAL GRID REFINEMENT
HNUMBER	T	I	F	0	0.00E+00		OEFINES THE NAHES OF THE DEF*** SUBROUTINES HHICH ARE CALLED
MODELNAME	T		C F	0	0.00E+00		DEFINES THE NAHE OF THE PHYSICAL MODEL (MUST FIT HITH "DEFMO")
NA	T	I	F	0	0.00E+00		DEFINES THE NUHBER OF INOEPENDENT VARIABLES IN THE PDE-SYSTEM
NG	F	S	F	11	0.00E+00		SPECIFIES THE NUMBER OF EQUIDISTANT POINTS IN THE INITIAL GRID
NORDER	F	S	F	6	0.00E+00		MAXIMUM ORDER OF (BDF) FOR THE TRANSIENT INTEGRATION OF THE PDE
NX	F	S	F	201	0.00E+00		SPECIFIES THE MAXIMUM NUMBER OF GRID POINTS DURING THIS COMMAND
OFILE	F	S	F	0	0.00E+00		UNIT NUMBER FOR THE OUTPUT FILE DURING SOLVE/TRANSIENT COMMANDS
PO	F	S	F	0	0.00E+00		CALL THE USER PROVIDED SUBROUTINE "PROFO"
P1	F	S	F	0	0.00E+00		CALL THE USER PROVIDED SUBROUTINE "PROFI"
P2	F	S	F	0	0.00E+00		CALL THE USER PROVIDED SUBROUTINE "PROF2"
P3	F	S	F	0	0.00E+00		CALL THE USER PROVIDED SUBROUTINE "PROF3"
P4	F	S	F	0	0.00E+00		CALL THE USER PROVIOED SUBRDUTINE "PROF4"
P5	F	S	F	0	0.00E+00		CALL M E USER PROVIDED SUBROUTINE "PROFS"
P6	F	S	F	0	0.00E+00		CALL THE USER PROVIOEO SUBROUTINE "PROF6"
P7	F	S	F	0	0.00E+00		CALL THE USER PROVIDED SUBROUTINE "PROF7"
P8	F	S	F	0	0.00E+00		CALL THE USER PROYIOED SUBROUTINE "PROF8"
P9	F	S	F	0	0.00E+00		CALL THE USER PROVIOEO SUBROUTINE "PROF9"
S1	F	I	F	0	0.00E+00		COPY STORED VARIABLE ON POSITION S1 TO POSITION E1: S(S1)=E(E1)
S2	F	I	F	0	0.00E+00		COPY STORED VARIABLE ON POSITION S2 TO POSITION E2: S(S2)=E(E2)
S3	F	I	F	0	0.00E+00		COPY STORED VARIABLE ON POSITION S3 TO POSITION E3: S(S3)=E(E3)
S4	F	I	F	0	0.00E+00		COPY STOREO VARIABLE ON POSITION S4 TO POSITION E4: S(S4)=E(E4)
S5	F	I	F	0	0.00E+00		COPY STORED VARIABLE ON POSITION S5 TO POSITION E5: S(S5)=E(E5)
SPATIAL	F	S	T	0	0.00E+00		T/F THE SYSTEM OF PDE HAS/HAS NOT SPATIAL OPERATORS
ST	F	S	F	0	0.00E+00		SIHULATION-START-TIHE GIVEN TO THE USER SUBROUTINES BY PTX(11)
TE	T	RC	F	0	0.00E+00		SPECIFIES THE VALUE OF THE TEMPERATURE GIVEN TO FUNCTION TEMPER
TINCREASE	F	S	F	0	0.33E+01		MAXIMUM TIME STEP INCREMENT FACTOR FOR SUCCEEOING TIME STEPS
TLARGE	F	S	F	0	0.36E+04		LARGEST POSSIBLE TIME STEP FOR THE TRANSIENT INTEGRATION
TN	F	S	F	10	0.00E+00		MINIMUM NUMBER OF ELAPSED TIME STEPS BETWEEN GRID UPDATES
TREDUCE	F	S	F	0	0.50E+00		REDUCTION FACTCR FOR TIME STEPS AFTER AN ERRONEOUS INTEGRATION
TSMALL	F	S	F	0	0.10E-11		SMALLEST POSSIBLE T I M STEP FOR THE TRANSIENT INTEGRATION
TTYPICAL	F	S	R F	0	0.00E+00		ASSUMED INITIAL TIME STEP FOR THE TRANSIENT INTEGRATION

KEYS OF THE "TRANSIENT " COMMAND IN ALPHABETICAL ORDER (CONTINUED)

KEY-NAME	OBL	TYPE	LV	IV	RV	CV	MEANING
U0	F	R	F	0	0.00E+00		THE 1-ST INTERVAL BOUNDARY FOR A USER REFINED GRID
U1	F	R	F	0	0.00E+00		THE 2-ND IIITERVAL BOUNDARY FOR A USER REFINED GRID
U2	F	R	F	0	0.00E+00		THE 3-RO INTERVAL BOUNDARY FOR A USER REFINED GRIO
U3	F	R	F	0	0.00E+00		THE 4-TH IIITERVAL BOUNDARY FOR A USER REFINED GRIO
U4	F	R	F	0	0.00E+00		THE 5-TH INTERVAL BOUNDARY FOR A USER REFINED GRID
U5	F	R	F	0	0.00E+00		THE 6-TH INTERVAL BOUNDARY FOR A USER REFINED GRIO
U6	F	R	F	0	0.00E+00		THE 7-TH IIITERVAL BOUNDARY FOR A USER REFINED GRID
U7	F	R	F	0	0.00E+00		THE 8-TH INTERVAL BOUNDARY FOR A USER REFINED GRIO
U8	F	R	F	0	0.00E+00		THE 9-TH INTERVN BOUNDARY FOR A USER REFINED GRID
U9	F	R	F	0	0.00E+00		THE 10-TH INTERVN BOUNDARY FOR A USER REFINED GRIO
VBEGIN	F	S	F	1	0.00E+00		POSITION OF THE FIRST VARIABLE TO BE PRINTEO
VENO	F	S	F	0	0.00E+00		POSITION OF THE LAST VARIABLE TO BE PRINTEO
X0	F	SC	F	0	0.00E+00		SETS THE VALUE OF THE PARAMETER PTX(10) IN THE USER SUORWTINES
X1	F	SC	F	0	0.00E+00		SETS THE VALUE OF THE PARAMETER PTX(1) IN THE USER SUBROUTINES
X2	F	SC	F	0	0.00E+00		SETS THE VALUE OF THE P W E T E R PTX(2) IN THE USER SUORWTINES
W	F	SC	F	0	0.00E+00		SETS THE VALUE OF THE P W E T E R PTX(3) IN THE USER SUBROUTINES
X4	F	SC	F	0	0.00E+00		SETS THE VALUE OF THE P W E T E R PTX(4) IN THE USER SUBROUTINES
W	F	SC	F	0	0.00E+00		SETS THE VALUE OF THE PARAMETER PTX(5) IN THE USER SUSROUTINES
X6	F	SC	F	0	0.00E+00		SETS THE VALUE OF THE PARAMETER PTX(6) IN THE USER SUBROUTINES
X7	F	SC	F	0	0.00E+00		SETS THE VINUE OF THE PARAMETER PTX(7) IN THE USER SUBROUTINES
X8	F	SC	F	0	0.00E+00		SETS THE VALUE OF THE P M T E R PTX(8) IN THE USER SUBROUTINES
X9	F	SC	F	0	0.00E+00		SETS THE VALUE OF THE PARAMETER PTX(9) IN THE USER SUBROUTINES
XCRITICAL	F	S	F	0	0.10E+02		CRITICAL PERCENTAGE M INTERPOLATED POINTS FOR SOLVING AGAIN
XN	F	S	F	10	0.00E+00		MINIMUM NUMBER OF ADDITIONAL POINTS F W A GRID UPDATE
ZINCL2NORM	F	S	F	0	0.10E+16		ZERO FOR THE L2-NORM OF THE SCALED INC. 'ZERO" - ZINCL2*EPSILON
ZINCLINORM	F	S	F	0	0.10E+16		ZERO FOR THE LI-NORH OF THE SCALED INC. "ZERO' - ZINCLI'EPSILON
ZRHSL2NORM	F	S	F	0	0.13E+03		ZERO FOR THE L2-NORM OF THE SCALED RHS. 'ZERO" - ZRHSL2*EPSILON
ZRHSLINORM	F	S	F	0	0.13E+03		ZERO FOR THE LI-NORH OF THE SCALED RHS.'ZERO" - ZRSHLI*EPSILON

KEYS OF THE'VARIABLE * COMMAND IN ALPHABETICAL ORDER.

KEY-NAME	OBL	TYPE	LV	IV	RV	CV	MEANING
CIMMEDIATE	F	S	T	0	0.00E+00		T: ONLY VALUES BETWEEN (LB,UB) ARE GIVEN TO THE USER INTERFACES
DIFFERENCE	F	S	F	0	0.47E+20		SPECIFIES THE MAXIMUM DIFFERENCE BETHEEN VALUES AT ADJACENT PTS
ELEMENT	T	C	F	0	0.00E+00		NAMES A VARIABLE BY A PREDEFINED NAME OR BY AN AVAILAELE NUMBER
LBOUNDARY	F	S	F	0	-0.47E+20		SPECIFIES THE LOWER BOUNDARY FOR THE VALUES OF THE VARIABLE
LINEARVARI	F	S	F	0	0.00E+00		T/F: ERROR CONTROL IN SPACE AND TIME IS W N E ABSOLUTE/RELATIVE
POSITION	T	I	F	0	0.00E+00		SPECIFIES THE WSITION OF THE VPRIABLE DURING THE SIMULATION
RATIO	F	S	F	0	0.47E+20		SPECIFIES THE MAXIMUM RATIO BETHEEN VALUES AT ADJACENT POINTS
SACCURACY	F	S	F	0	0.10E-01		SPECIFIES THE ACCURACY F W THE MODIFICTIONS OF THE SPATIAL GRIO
TACCURACY	F	S	F	0	0.10E-01		SPECIFIES THE ACCURACY FOR THE TRANSIENT INTEGRATION OF THE PDE
TH	F	S	F	0	0.47E+20		SPECIFIES THE MAXIMUM ERROR FOR THE TRANSIENT INTEGRATION
UBOUNDARY	F	S	F	0	0.47E+20		SPECIFIES THE UPPER BOUNDARY FOR THE VALUES OF THE VARIABLE
VALUE	F	S	F	0	0.10E+01		SPECIFIES THE CONSTANT VALUE TO WHICH THIS VARIABLE IS SET
XA	F	S	F	0	0.10E-01		SPECIFIES THE ABSOLUTE DEVIATION OF THE PROFILES IN X-DIRECTION

4. Syntax of the Command Language

The INPUT-DECK consists of lines which can have a length of up to 80 characters per line. The lines can be

* command lines
* continuation lines
* comment

A "command" consists of the

* command name
* "key = value" sequences
* delimiters

A "command line" contains the command name and may contain an arbitrary number of "key = value" sequences.

A "continuation line" has a '+' in the first column and may contain an arbitrary number of "key = value" sequences.

A "comment" begins with a '*' and may contain any characters. A "comment" can be a whole line or a part of the line.

We denote

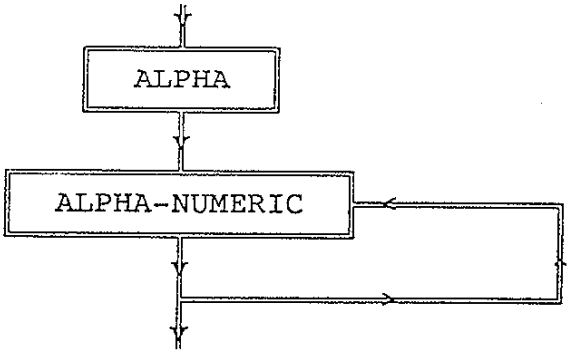
ALPHA = [A,B,C,D,E,F,G,H,I,J,K,L,M,
N,O,P,Q,R,S,T,U,V,W,X,Y,Z]
only upper case letters are pennitted

ALPHA-NUMERIC = [A,B,C,D,E,F,G,H,I,J,K,L,M,
N,O,P,Q,R,S,T,U,V,W,X,Y,Z,
0,1,2,3,4,5,6,7,8,9]
only upper case letters are permitted

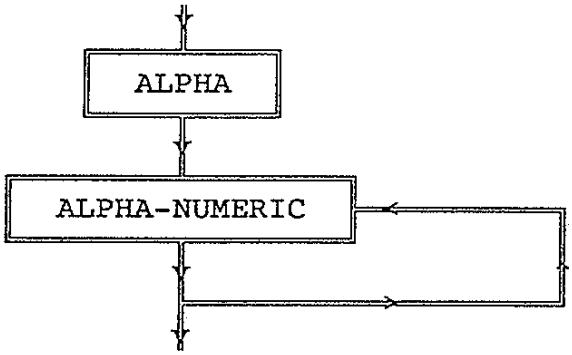
DELIMITER = ['BLANK', 'COMMA', 'SEMICOLON']

SYMBOL = any ASCII character except
['BLANK' 'COMMA' 'SEMICOLON' '*']

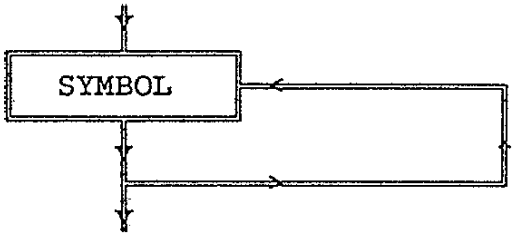
A "COMMAND" may be written as:



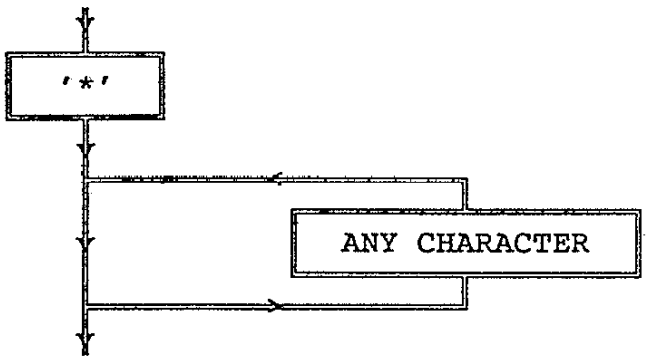
A "KEY" may be written as:



A "VALUE" may be written as:

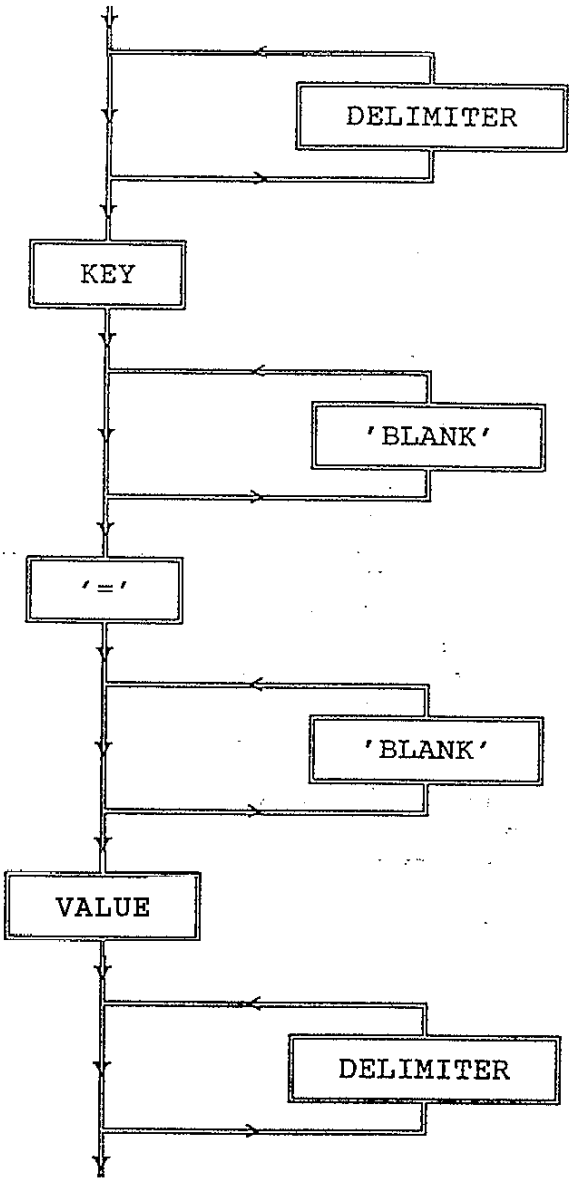


A "COMMENT" may be written as:



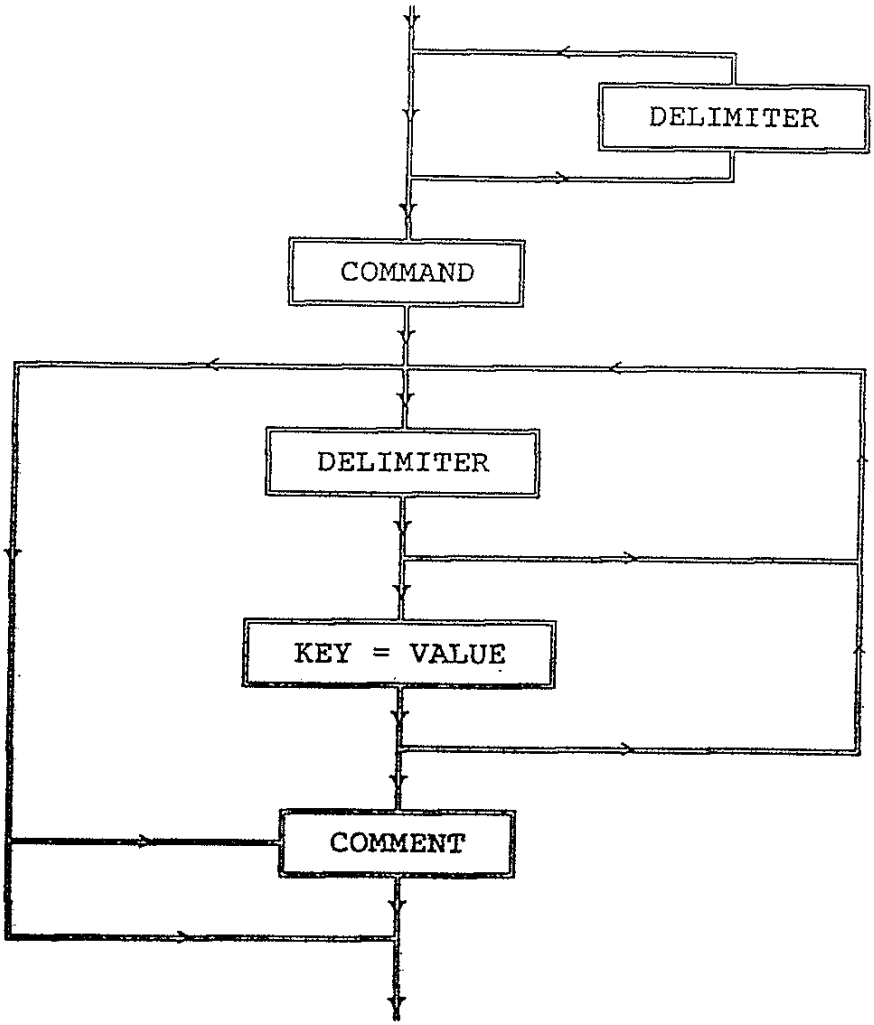
Syntax of the Command Language

A "KEY = VALUE" may be written as:

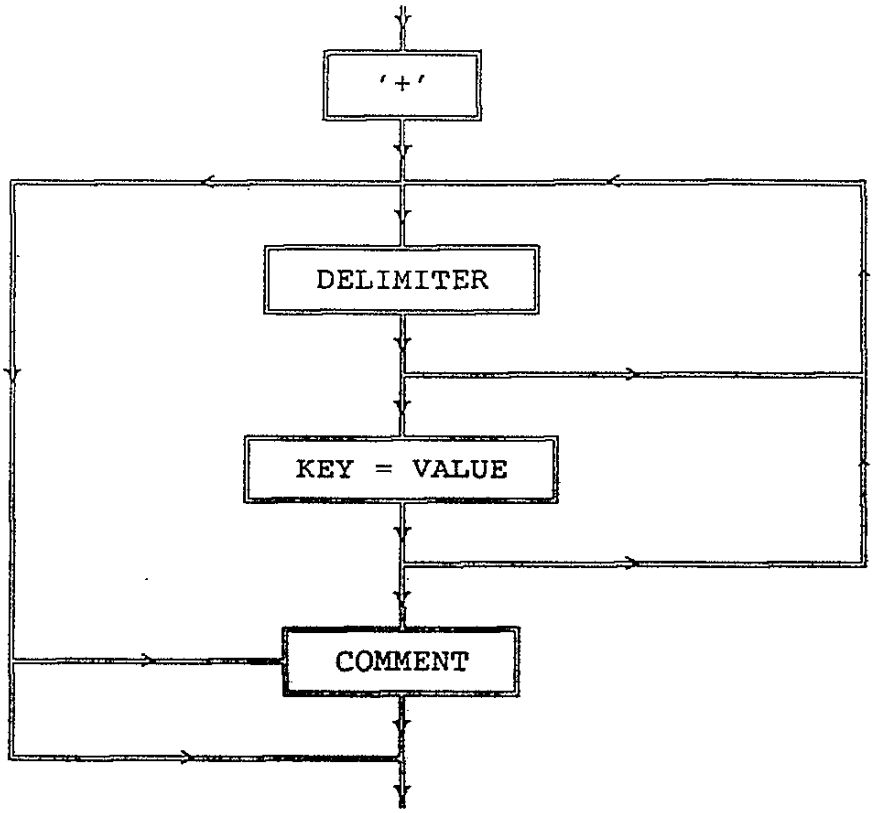


A "KEY = VALUE" sequence must be specified within one line in the INPUT-DECK.

A "COMMAND LINE" may be written as:



A "CONTINUATION LINE" may be written as:



5. UTILITY SUBROUTINES

Utility subroutines are **FORTRAN-77** subroutines or functions. The routines can be called from every **user** provided subroutine. Utility subroutines are used to compute

* the function arsinh(x) = ln(x+sqrt(x^2+1))
* the intrinsic number n.
* the temperature with respect to specifications in the INPUT-DECK (TE-key)
* the concentration of electrons and **holes** and their derivatives with respect to a singly charged donor concentration (N_D^+).
* the field enhancement factors for a specified **system** of diffusivities.
* the default model for device simulation of a diode (subroutines **DEVI****).
* the default model for process simulation (subroutines **PROC**** in connection with the SETPROCESS **command**)

5.1 The Function ARSINH(X)

The function is computed in the following way:

$x > +\sigma$: $\quad \text{arsinh}(x) = +\ln(+x+\sqrt{x^2+1})$

$-\sigma < x < +\sigma$: $\quad \text{arsinh}(x) = x + a\,x^3 + b\,x^5 + \ldots$

$x < -\sigma$: $\quad \text{arsinh}(x) = -\ln(-x+\sqrt{x^2+1})$

Example: Y = **ARSINH(X)**

5.2 Computation of the Intrinsic Number

The subroutine CINTRC computes the intrinsic number in silicon in units of $[\mu m^{-3}]$ as a function of the temperature in degree Celsius.

Example: CI = **CINTRC(TC)**

CI...n_i...**intrinsic** number in $[\mu m^{-3}]$
TC...**temperature** in degree Celsius (-273 < TC < 1412)

5.3 Computation of the Temperature

The function TEMPER computes the temperature in degree Celsius as a function of the simulation time (**ST-key** and ET-key) and the depth. The function considers specifications in the INPUT-DECK (TE-key) and **calls** the **user** provided subroutines TEMP* (*=0...9) if necessary.

Example: TC = **TEMPER(TIME,DEPTH)**

TC...temperature in degree Celsius
TIME...**t...simulation** time (ST-key, ET-key)
DEPTH...**x...depth** at which the function **is** called.

5.4 Computation of the Carrier Concentrations

The subroutine CARRIE computes the concentration of the electrons and holes as a function of the intrinsic number, the concentration of donors and acceptors and the charge state of donors and acceptors. Boltzmann statistics and the quasi-neutral approximation are used for the computation. Furthermore the derivatives with respect to a singly charged donor (N_D^+) is computed

Example:

```
DIMENSION S(NA), CHRG(NA)
...
CALL CARRIE (CN,CP,DCN,DCP,CI,NA,S,CHRG)
```

CN...n...concentration of electrons
CP...p...concentration of holes

$$DCN = \frac{\delta CN}{\delta S} = \frac{Sn}{\delta N_D^+} \qquad (N = N_D + N_D^{++} + \ldots - N_A^- - N_A^{--} - \ldots)$$

$$DCP = \frac{SCP}{\delta S} = \frac{\delta p}{\delta N_D^+} \qquad (N = N_D^+ + N_D^{++} + \ldots - N_A^- - N_A^{--} - \ldots)$$

CI...n_i...intrinsic number
NA...number of donors and acceptors
S...concentration of donors and acceptors
CHRG...charge state of donors (usually +1) and acceptors (usually -1)

5.5 Computation of the Field Enhanced Diffusion

The subroutine FLDENH modifies the matrix D_{ij}:=D(I,J) in the subroutine DEF*FL to account for the field enhancement. Boltzmann statistics and the quasi-neutral approximation are assumed to be valid.

Example:

```
LOGICAL DIA
DIMENSION S(NA),CHRG(NA),D(NA,NA),DD(NA,NA,NA)
DIMENSION DMX(NA,NA),DAUX(NA,NA),DDAUX(NA,NA,NA)
...
CALL FLDENH (DIA,CI,NA,S,CHRG,D,DD,DMX,DAUX,DDAUX)
```

DIA...is .TRUE. if only the diagonal elements of D_{ij} are non zero
...is .FALSE. in the general case. (This increases the CPU-time)
CI...n....intrinsic number
NA...nhmber of donors and acceptors
S...concentration of the donors and acceptors
CHRG...charge state of the donors and acceptors
D(I,J)...D_{ij}....matrix of the diffusivities...(see DEF*FL in chapter 6)
DD(I,J,K)...parital derivative of the diffusivity D(I,J) with respect to the solution S(K)...(see DEF*FL in chapter 6)
DMX(I,J)...maximum summand during the computation of D(I,J)...(see DEF*FL)
DAUX...auxiliary array of the length NA*NA
DDAUX...auxiliar· array of the length NA*NA*NA

Before the subroutine FLDENH is called, D, DD and DMX contain the diffusivities without any field enhancement. After the subroutine is called, D, DD and DMX contain the diffusivities including the field enhancement factors and their derivatives. The original matrices D and DD are stored in DAUX and DDAUX after the subroutine has been called.

5.6 Subroutines DEVI**

This set of subroutines specifies the physical model for the device simulation of a diode. The first variable must be POTENTIAL, the second one must be ELECTRON, the third one must be HOLE, the fourth one must be DONOR, PHOSPHORUS, ARSENIC or ANTIMONY and the fifth one must be ACCEPTOR or BORON.

At the boundaries the values of the electrons and holes are set to equilibrium values. The external voltage is set to the value of the 'PTX(1)' specified by the X1-key. The values of PTX(2..5) influence the physical model used for the device simulation.

X1-key = PTX(1): external voltage

X2-key = PTX(2) > 1: no Auger recombination taken into account

X3-key = PTX(3) > 1: no impact ionization is taken into account

X4-key = PTX(4) > 1: no dependence of the mobilities on the donor and acceptor concentration

X5-key = PTX(5) > 1: no velocity recombination is taken into account

All quantities are computed in [µm], [sec], [A], [^{o}C]. The subroutine is activated by setting:

MN=-2 and MO=DEVICE

5.7 Subroutines PROC**

This set of subroutines specifies the default physical model for the process simulations. Up to 5 variables can be specified in arbitrary ordering. The variables can be any combination of BORON, ARSENIC, ASCLUSTER, PHOSPHORUS and ANTIMONY. If ASCLUSTER is set ARSENIC must be set too. The models for the simulation considers field enhancement and dynamic clustering.

The boundary conditions must be set by the SETPROCESS command before execution of a TRANSIENT command. All physical parameters are computed in [µm], [sec], [^{o}C] and [A]. The subroutines are activated by setting:

MN=-1 and MO=PROCESS

6. USER PROVIDED SUBROUTINES

User provided subroutines are FORTRAN-77 subroutines or functions. They must be compiled and linked to the main program during PASS2. The routines are called during the execution of a PROFILE, a SOLVE and a TRANSIENT command.

User provided subroutines are used to

* specify a system of partial differential equations (DEF*BC, DEF*CO, DEF*FL, DEF*MO, DEF*TT)
* modify the spatial distribution of a variable (PROF*)
* specify a space and time dependent temperature (TEMP*)
* specify a time dependent Parameter (PARA*)

(*=0...9). If user provided subroutines are activated in the INPUT-DECK they must be linked to the main program of "pass2".

6.1 User Provided Subroutines Describing the Partial Differential Equation and Boundary Conditions

Five subroutines are necessary to completely describe the system of partial differential equations (Eq.1 and Eq.2) and its boundary conditions (Eq.7). These subroutines are FORTRAN-77 subroutines called DEF*MO, DEF*TT, DEF*CO, DEF*FL and DEF*BC. They are activated from the INPUT-DECK by setting MN=* (*=0...9).

The dimension 'ne' of the solution vector in the program can be larger than the number of coupled differential equations 'na'. In this case the variables with the indices 'na+1' to 'ne' are not modified during the SOLVE or TRANSIENT command. They may be used as space and/or time dependent information for the differential equations.

An example is a device simulation: 'na'=3 and 'ne'=5.

$S_1 = \phi$ ..... electrostatic potential
$S_2 = n$ ..... electron concentration
$S_3 = p$ ..... hole concentration
$S_4 = N_D$ .... concentration of donors
$S_5 = N_A$ .... concentration of acceptors

6.1.1 Subroutine DEF*MO

DEF*MO is called once in the beginning of a SOLVE or a TRANSIENT command. It is called before DEF*TT, DEF*BC, DEF*CO and DEF*FL are called. The subroutine should be used to check if the number, kind and ordering of the variables in the INPUT-DECK coincides with the number, kind and ordering in the user provides subroutines. Furthermore the model name specified by the MO-key in the INPUT-DECK can be checked. DEF*MO must be a FORTRAN-77 subroutine with the following parameter list:

```
      SUBROUTINE DEF*MO (NA,NE,KSYS,CMODEL)
      CHARACTER*63 CMODEL
      DIMENSION KSYS(NE)
      ...
      ...
      ...
      END
```

NA...'na'...is the number of variables in the system of partial differential equations
NE...'ne'...is the dimension of the solution vector
KSYS(IE)...specifies the meaning of the IE^{th} variable S_{ie} (see element predefinitions in subroutine 'WWWWEL' and VARIABLE command)
CMODEL...contains the model name specified by the MO-key in the INPUT-DECK.

NA, NE, KSYS and CMODEL must not be modified.

6.1.2 Subroutine DEF*TT

DEF*TT is called once in the beginning of a SOLVE command and once at every time step during the TRANSIENT command. It is called after DEF*MO and before DEF*CO, DEF*FL and DEF*BC are called. The subroutine should be used to specify all only time dependent parameters of the models. DEF*TT must be a FORTRAN-77 subroutine with the following parameter list:

```
      SUBROUTINE DEF*TT (NA,NE,NEMX,NX,PTX,X,S)
      DIMENSION PTX(12),S(NEMX,NX),X(NX)


      ...
      END
```

NA...'na'...is the number of variables in the system of partial differential equations
NE...'ne'...is the number of all specified variables.
NEMX...is the dimension of the solution vector
NX...is the current number of grid points
PTX(1)...contains the value specified by the X1-key
PTX(2)...contains the value specified by the X2-key
...
PTX(10)...contains the value specified by the XO-key
PTX(11)...t...contains the current simulation time (ST-key, ET-key)
PTX(12)...contains the last time step width used in a TRANSIENT command
X(IX)...contains the current spatial grid used for the simulation
S(IE,IX)...S_{ie}(x=X(IX),t=PTX(11))...contains the value of the IEth variable at the depth X(IX) at the time PTX(11).

NA, NE, NEMX, NX, PTX, S and X must not be modified.

6.1.3 Subroutine DEF*CO

DEF*CO is called at every grid point at each time step. DEF*CO is called after DEF*MO and DEF*TT have been called. It is used to specify the functions T_{ij} and G_i in equation (1). Furthermore the partial derivatives of these functions with respect to S_k and F_l must be computed. DEF*CO must be a FORTRAN-77 subroutine with the following Parameter list:

```
      SUBROUTINE DEF*CO (NA,NE,PTX,S,F,
     +                   T,TMX,G,DGDS,DGDF,GMX)
      DIMENSION PTX(15),S(NE),F(NA),T(NA,NA),TMX(NA,NA)
      DIMENSION G(NA),DGDS(NA,NA),DGDF(NA,NA),GMX(NA)
      ...
      ...
      ...
      END
```

NA...'na'...is the number of variables in the system of partial differential equations
NE...'ne'...is the dimension of the solution vector
PTX(1)...contains the value specified by the X1-key
PTX(2)...contains the value specified by the X2-key
...
PTX(10)...contains the value specified by the XO-key
PTX(11)...t...contains the current simulation time (ST-key and ET-key)
PTX(12)...contains the last time step width used in a TRANSIENT command
PTX(13)...x_{ix-1}...contains the depth of the nearest lower grid point (at the lower boundary PTX(13)=PTX(14))
PTX(14)...x_{ix}...contains the depth at which the subroutine is called
PTX(15)...x_{ix+1}...contains the depth of the nearest upper grid point (at the upper boundary PTX(15)=PTX(14))
S(IE)...$S_{ie}(x=PTX(14),t=PTX(11))$...contains the value of the IE^{th} variable at the time PTX(11) and the depth PTX(14).
F(IA)...$F_{ia}(x=PTX(14),t=PTX(11))$...contain the value of the flux of the IA^{th} variable at the time PTX(11) and the depth PTX(14).
T(I,J)...T_{ij}
TMX(I,J)...maximum absolute summand during the computation of T(I,J)
G(I)...G_i

GMX(I)...maximum absolute summand during the computation of G(I)

$$\text{DGDS(I,J)} = \frac{\delta G(I)}{\delta S(J)} = \frac{\delta G_i}{\delta S_j}$$

$$\text{DGDF(I,J)} = \frac{\delta G(I)}{\delta F(J)} = \frac{\delta G_i}{\delta F_j}$$

ZOMBIE sets all elements of T, TMX, G, GMX, DGDS and DGDF to zero. Therefore only non zero elements must be specified.

NA, NE, PTX, S and F must not be modified.

T, TMX, G, DGDS, DGDF and GMX must be specified if they contain non zero elements.

6.1.4 Subroutine DEF*FL

DEF*FL is called at every midpoint in the spatial grid at each time step. DEF*FL is called after DEF*MO and DEF*TT have been called. It is used to specify the function $U_{..}$ and D_{ij} in equation (2). Furthermore the partial derivatives with respect to S_k must be computed. DEF*FL must be a FORTRAN-77 subroutine with the following parameter list:

```
SUBROUTINE DEF*FL (NA,NE,FS,PTX,S,F,U,D,DU,DD,UMX,DMX)
LOGICAL FS
DIMENSION PTX(15),S(NE),F(NA)
DIMENSION U(NA,NA),D(NA,NA),DU(NA,NA,NA)
DIMENSION DD(NA,NA,NA),DMX(NA,NA),UMX(NA,NA)

...
...
END
```

NA...'na'...is the current number of independent variables
NE...'ne'...is the dimension of the solution vector
FS....FALSE. means that the vector F contains zero elements
  ....TRUE. means that the vector F contains the flux F
PTX(1)...contains the value specified by the X1-key
PTX(2)...contains the value specified by the X2-key
...
PTX(10)...contains the value specified by the XO-key
PTX(11)...t...contains the current simulation time (ST-key and ET-key)
PTX(12)...contains the time step width of the last iteration of a TRANSIENT command
PTX(13)...x_{ix-1}...contains the depth of the nearest lower grid point
PTX(14)...x_{ix}...contains the actual depth at which the subroutine is called (this is a position between two existing grid points)
PTX(15)...x_{ix+1}...contains the depth of the nearest upper grid point
S(IE)...S_{ie}(x=PTX(14),t=PTX(11))...contains the value of the IE^{th} variable at the time PTX(11) and a depth PTX(14)
F(IA)...contains zero elements if FS=.FALSE.
    ...contains the F_{ia}(x=PTX(14),t=PTX(11))
U(I,J)...$U_{..}$
D(I,J)...D_{ij}
UMX(I,J)...maximum absolute summand during computation of U(I,J)

DMX(I,J)...maximum absolute summand during computation of D(I,J)

$$DD(I,J,K) = \frac{\delta D(I,J)}{\delta S(K)} = \frac{\delta D_{ij}}{\delta S_k}$$

$$DU(I,J,K) = \frac{\delta U(I,J)}{\delta S(K)} = \frac{\delta U_{ij}}{\delta S_k}$$

ZOMBIE sets all elements of U, D, UMX, DMX, DU and DD to zero. Therefore only non zero elements have to be specified. If U and D depend on the flux F, the computation of the correct Jacobian is not possible.

NA, NE, FS, PTX, S, F must not be modified.

U, D, DU, DD, UMX and DMX must be specified if they contain non zero elements.

6.1.5 Subroutine DEF*BC

DEF*BC is called at each time step at the lower and at the upper boundary of the simulation domain. DEF*BC is called after DEF*MO and DEF*TT have been called. It is used to specify the functions A_{ij}, B_{ij} and C. in equation (7). Furthermore the partial derivatives of C. with respect to S_k and F_l must be specified. DEF*BC must be a FORTRAN-77 subroutine with the following parameter list:

```
      SUBROUTINE DEF*BC (KB,NA,NE,PTX,S,F,
     +                   A,B,C,DCDS,DCDF,AMX,BMX,CMX)
      DIMENSION PTX(5),S(NE),F(NA)
      DIMENSION A(NA,NA),B(NA,NA),C(NA)
      DIMENSION DCDS(NA,NA),DCDF(NA,NA)
      DIMENSION AMX(NA,NA),BMX(NA,NA),CMX(NA)
      ...
      ...
      ...
      END
```

KB...+1: DEF*BC is called at the lower boundary
...+2: DEF*BC is called at the upper boundary
NA...'na'...is the number of variables in the system of partial differential equations
NE...'ne'...is the dimension of the solution vector
PTX(1)...contains the value specified by the X1-key
PTX(2)...contains the value specified by the X2-key
...
PTX(10)...contains the value specified by the XO-key
PTX(11)...t...contains the current simulation time (ST-key and ET-key)
PTX(12)...containts the time step width used in a TRANSIENT command
PTX(13)...x_{ix-1}...contains the depth of the nearest lower grid point (at a lower boundary PTX(13)=PTX(14))
PTX(14)...x_{ix}...contains the actual depth at which the subroutine is called
PTX(15)...x_{ix+1}...contains the depth of the nearest upper grid point (at an upper boundary PTX(15)=PTX(14))
S(IE)...S_{ie}(x=PTX(14),t=PTX(11))...contains the value of the IE^{th} variable at the position PTX(14) and at the time PTX(11).

F(IA)...F_{ia}(x=PTX(14),t=PTX(11))...contains the value of the flux of the IAth variable at the position PTX(14) and the time PTX(11)

A(I,J)...A_{ij}...coefficients of the fluxes F_{ia} in Equation (7)

B(I,J)...B_{ij}...coefficients of the variables S_{ie} in Equation (7)

C(I)...C_i

AMX(I,J)...maximum absolute summand during the computation of A(I,J)

BMX(I,J)...maximum absolute summand during the computation of B(I,J)

CMX(I)...maximum absolute summand during the computation of C(I)

$$\text{DCDS(I,J)} = \frac{\delta C(I)}{\delta S(J)} = \frac{\delta C_i}{\delta S_j}$$

$$\text{DCDF(I,J)} = \frac{\delta C(I)}{\delta F(J)} = \frac{\delta C_i}{\delta F_j}$$

ZOMBIE sets all elements of A, B, C, DCDS, DCDF, AMX, BMX and CMX to zero. Therefore only non zero elements must be specified.

KB, NA, NE, PTX, S and F must not be modified.

A, B, C, DCDS, DCDF, AMX, BMX, CMX must be specified if they contain non zero elements.

6.2 User Provided Subroutines Modifying the Profiles

Up to ten subroutines can be called during the execution of a PROFILE, a SOLVE or a TRANSIENT command to modify the distribution of the variables. (You should not modify the first NA variables during a SOLVE or a TRANSIENT command but you can modify the NA+1st to the NEth variable).

The subroutines must be FORTRAN-77 subroutines with the names PROF* (*=0...9) and with the following parameter list:

```
      SUBROUTINE PROF* (NE,KSYS,PTX,S)
      DIMENSION KSYS(NE),PTX(15),S(NE)

      ...
      ...
      END
```

NE...'ne'...is the dimension of the solution vector
KSYS(IE)...contains a positive integer number which specifies the IEth variable S_{ie} (see element predefinitions (subroutine WWWWEL) and VARIABLE command)
PTX(1)...contains the value specified by the X1-key
PTX(2)...contains the value specified by the X2-key
...
PTX(10)...contains the value specified by the XO-key
PTX(11)...t...contains the current simulation time (ST-key, ET-key)
PTX(12)...contains the last time step width during a TRANSIENT command
PTX(13)...x_{ix-1} ..contains the nearest lower grid point (at the lower boundary PTX(13)=PTX(14))
PTX(14)...x_{ix}...contains the depth of the grid point at which the function is called
PTX(15)...x_{ix+1}...contains the nearest upper grid point (at the upper boundary PTX(15)=PTX(14))
S(IE)...S_{ie}(x=PTX(14),t=PTX(11))...contains the value of the IEth variable at the position PTX(14) and the time PTX(11).

NA, NE, KSYS, PTX must not be modified.

S can be modified if required

6.3 User Provided Subroutines to Specify the Temperature

Constant temperatures can be specified by the TE-key in the INPUT-DECK. Time and/or depth dependent temperatures can be specified by up to ten functions. called TEMP* (*=0...9). These functions will be called by the system provided function TEMPER if they are specified in the INPUT-DECK by setting TE=TEMP* (*=0...9). The functions must be FORTRAN-77 functions with the following parameter list:

```
      FUNCTION TEMP* (TIME,DEPTH)
      ...

      ...
      TEMP* = ..
      END
```

TIME...t...current time during the simulation (ST-key, ET-key)
DEPTH...x...current depth at which the function is called
TEMP*...(*=0...9)...temperature in degree Celsius

TIME and DEPTH must not be changed

6.4 User Provided Specification of the Parameter

Constant parameters can be specified by the X*-key from the INPUT-DECK. Time dependent values for the parameter can be specified by the functions PARA* (*=0...9) which are activated from the INPUT-DECK by setting the X*-key (**X*=PARA%, %=0..9.** '*' and '%' are not correlated). The function must be a FORTRAN-77 function with the name PARA* and the following parameter list:

```
      FUNCTION PARA* (TIME)

      ...
      ...
      PARA* = ...
      END
```

TIME...t...current simulation time when the function **is** called (ST-key, ET-key)
PARA*...value of the parameter **(X%-key, PTX(%),** % and * are not correlated)

TIME must not be changed

7. O U T P U T F I L E S

File Unit 1:

Contains an unchanged copy of the user specified INPUT-DECK.

File Unit 2:

Contains the checked INPUT-DECK. If errors occur in the INPUT-DECK the location and the kind of error is indicated. Each of the commands get an increasing command number.

File Unit 3:

This file should be assigned to the SYSTEM-OUTPUT unit. The program writes short messages to this unit to informs the user about the execution of an interactive (batch) job.
A messages consists of

"used CPU time" "Command" "Parameters"

Following commands contain parameters:

PROFILE: "PRGR" nxl, nx2, nx3, nx4

nx1...number of spatial grid points in the new grid
nx2...number of interpolated grid points in the new grid
nx3...number of grid points which are common to the old and to the new grid
nx4...number of grid points in the old grid

IMPLANT: "IMGR" nxl, nx2, nx3, nx4

nxl...number of spatial grid points in the new grid after the ion implantation
nx2...number of interpolated grid points in the new grid
nx3...number of grid points which are common to the old and to the new grid
nx4...number of grid points in the old grid

SOLVE: "SOGR" nxl, nx2, nx3, nx4

nxl...number of spatial grid points in the new grid after the solution of the differential equation
nx2...number of interpolated grid points in the new grid
nx3...number of grid points which are common to the old and to the new grid
nx4...number of grid points in the old grid

SOLVE: "SOSO" ng, iter, nx, L2-Norm

ng...number of grid updates during the execution
iter...number of iterations during solving the nonlinear equation system
nx...number of grid points the the spatial grid
L2-Norm...L2-Norm of the Right-Hand-Side at the end of the Newton iteration

TRANSIENT: "TRGR" nxl, nx2, nx3, nx4

nxl...number of spatial grid points in the new grid after a completely new grid has been set up.
nx2...number of interpolated grid points in the new grid
nx3...number of grid points which are common to the old an the new grid
nx4...number of grid points in the old grid

TRANSIENT: "TRSO" nt, no, time, dtime, nx, iter, L2-Norm

nt...number of the transient iteration since begin of the command
no...order used for the integration of the transient Operator
time...simulation time used for the models (ST- and ET-key)
dtime...time step width used for the integration
nx...number of grid points used for the integration
iter...number of Newton iterations to solve the time step
L2-Norm...L2-Norm of the Right-Hand-Side.

In additon warnings and fatal error messages are sent to this file unit.

File Unit 4:

This is the standard output and result file.

File Unit 5: unused

File Unit 6: unused

File Unit 7:

Contains information about all grid modifications during the execution of the job. Most of the Parameters are in close connection to the PROFILE command.

The numbers in the columns have following meaning:

01....number of the command
02....number of specified variables
03....number of elapsed time steps
04....simulation time (ST- and ET-key)
05....kind of grid modification (0=rigid, 1=enlarge, 2=modify)
06....T/F: sufficient/insufficient grid points to satisfy the system defined grid refinements
07....T/F: sufficient/insufficient grid points to satisfy the user defined grid refinements
08....T/F: sufficient/insufficient grid points to satisfy the specified maximum deviation (SA-key) of the distribution of a variable from an optimal polynomal of second order through four adjacent grid points.
09....T/F: sufficient/insufficient grid points to satisfy the maximum difference criterium
10....T/F: sufficient/insufficient grid points to satisfy the maximum ratio criterium
11....number of grid updates necessary for (06)
12....number of grid updates necessary for (07)
13....number of grid updates necessary for (08)
14....number of grid updates necessary for (09)
15....number of grid updates necessary for (10)
16....number of grid points necessary for (06)
17....number of grid points necessary for (07)
18....number of grid points necessary for (08)
19....number of grid points necessary for (09)
20....number of grid points necessary for (10)
21....number of grid points in the new grid
22....number of interpolated grid points in the new grid
23....number of grid points which are identical in the old and the new grid
24....number of grid points in the old grid
25....T/F: subroutine PROF0 will/will not be called
26....T/F: subroutine PROF1 will/will not be called
27....T/F: subroutine PROF2 will/will not be called
28....T/F: subroutine PROF3 will/will not be called
29....T/F: subroutine PROF4 will/will not be called
30....T/F: subroutine PROF5 will/will not be called
31....T/F: subroutine PROF6 will/will not be called
32....T/F: subroutine PROF7 will/will not be called
33....T/F: subroutine PROF8 will/will not be called
34....T/F: subroutine PROF9 will/will not be called
35....T/F: subroutine DEVIPR will/will not be called
36....T/F: subroutine PROCPR will/will not be called
37....T/F: an ion implantation will/will not be executed
38....39: the values of the S1- and the E1-key
40....41: the values of the S2- and the E2-key
42....43: the values of the S3- and the E3-key
44....45: the values of the S4- and the E4-key
46....47: the values of the S5- and the E5-key

File Unit 8:

This file contains information about the Newton iterations during SOLVE and TRANSIENT commands.

01....number of the executed command
02....number of the elapsed time steps since begin of the command
03....number of partial differential equations to be solved
04....number of specified variables
05....number of the model (MN-key)
06....Simulation time (ST- and ET-key)
07....counts the number of grid refinements
08....counts the number of range rejects (LB-, UB- and CI-key in the VARIABLE command)
09....counts the number of Newton iterations
10....counts the number of Deuflhardt dampings
11....L2-Norm of the Right Hand Side
12....L-Infinte-Norm of the Right Hand Side
13....L2-Norm of the last increment of a Newton iteration
14....L-Infinie-Norm of the last increment of a Newton iteration
15....number of computations of a Jacobian
16....number of computations of the Right Hand Side

File Unit 9:

This file unit contains information about the transient integration

01....number of the executed command
02....number of elapsed time step since execution of the command
03....order of the polynomal used for the transient integration
04....counts the rejected Newton iterations during the TRANSIENT command due to too large time steps which caused a divergence
05....counts the rejected Newton iterations due to too large integration errors
06....simulation time (ST- and ET-key)
07....time step width of the last transient iteration
08....T/F: the last time step has been/has not been accepted due to spatial grid modifications
09....zero means: Newton converged properly
10....T/F: system required grid refinement fulfilled/not fulfilled
11....T/F: user required grid refinement fulfilled/not fulfilled
12....T/F: spatial discretisation error minimized/not

minimized (SA-key)
13....T/F: maximum difference criterium fulfilled/not fulfilled
14....T/F: maximum ratio criterium fulfilled/not fulfilled
15....L2-Norm of the Right Hand Side
16....L-Infinte-Norm of the Right Hand Side
17....L2-Norm of the last increment of a Newton iteration
18....L-Infinie-Norm of the last increment of a Newton iteration
19....model number specified by the MN-key
20....T/F: spatial operators active/not active (SPATIAL-key)

File Unit 10:

Binary communication file between "pass1" and "pass2" of ZOMBIE.

File Units 11...20:

Data output files for PRINT, SAVE and GET commands. Do not mix up PRINT and SAVE and GET command file units. These file units depends upon local installation. (look at variables IIIIII(21) and IIIIII(22) in subroutine ZZZZOC)

8. RELATED LITERATURE

Jüngling W., "Entwicklung und Auswertung verbesserter Modelle für die Prozeß- und Bauelementesimulation", Dissertation am Institut für Allgemeine Elektrotechnik und Elektronik, Technische Universität Wien, 1986.

Pichler P., "Numerische Simulation kritischer Prozeßschritte in der Halbleitertechnik", Dissertation am Institut für Allgemeine Elektrotechnik und Elektronik, Technische Universität Wien, 1985.

Budil M., Jüngling W., Guerrero E., Selberherr S., Pötzl H., "Modeling of Point Defect Kinetics During Thermal Oxidation", Proceedings SIMULATION OF SEMICONDUCTOR DEVICES AND PROCESSES Conf., (21.-23.July 1986, Swansea), pp.384-397.

Jüngling W., Hobler G., Selberherr S., Pötzl H., "Adaptive Grids in Space and Time for Process and Device Simulators", Proceedings NUMERICAL GRID GENERATION IN COMPUTATIONAL FLUID DYNAMICS Conf., (14.-17.July 1986, Landshut), pp.729-739.

Jüngling W., Pichler P., Selberherr S., Pötzl H., "Automation in Process- and Device-Simulation", Proceedings AUTOMATION 86 Conf., (10.-12.March 1986, Houston), pp.530-534.

Jüngling W., Pichler P., Selberherr S., Pötzl H., "ZOMBIE - A Coupled Process-Device Simulator", Abstracts AMSE Conf. MODELLING AND SIMULATION, (25.-28.November 1985, Monastir), Vol.1, p.137-138 and Proc. Vol.2A, pp.137-146.

Pichler P. ,Jüngling W., Selberherr S., Guerrero E., Pötzl H., "Simulation of Critical IC Fabrication Steps", IEEE Trans. Electron Devices, Vol.ED-32, No.10, pp.1940-1953, 1985. IEEE Trans. Computer-Aided Design, Vol.CAD-4, No.4, pp.384-397, 1985.

Pichler P., Jüngling W., Selberherr S., Guerrero E., Pötzl H., "Process and Device Simulation with One and the Same Program", Abstracts NASECODE IV Conf., (19.-21.June 1985, Dublin) and Proc. pp.477-482.

Jüngling W., Pichler P., Selberherr S., Guerrero E., Pötzl H., "Spatial and Transient Grids for Process and Device Simulators", Abstracts NASECODE IV Conf., (19.-21.June 1985, Dublin) and Proc. pp.320-325.

Jüngling W., Pichler P., Selberherr S., Guerrero E., Pötzl H., "Simulation of Critical IC Fabrication Processes Using Advanced Physical and Numerical Methods", IEEE Trans. Electron Devices, Vol.ED-32, No.2, pp.156-167, 1985. IEEE J. Solid-State Circuits, Vol.SC-20, No.1, pp.76-87, 1985.

Jüngling W., Pichler P., Selberherr S., Guerrero E., Pötzl H., "Comparison of Advanced Models for Coupled Diffusion", Proceedings ICCAD Conf., (12.-15.November 1984, Santa Clara), pp.167-169.

S.Selberherr, "Analysis and Simulation of Semiconductor Devices", Springer-Verlag Wien-NewYork, ISBN 3-211-81800-6, 1984.