NAME

tecc - technology compiler for space / space3d

SYNOPSIS

tecc [-cefFnostuUvV] [-m maskdatafile] [-p process] file

OPTIONS

The following options can be specified:

-c Write non coded. Default, the key lists are coded to get a smaller output file.

-e Print expanded condition list for each new mask and element.

-f Force rewrite of *unigreen* dielectric files (see option **-u**).

-F Force rewrite of *unigreen* substrate files (see option **-U**).

-n Do not compress table-format element-definition file. This option is useful during element-definition file development. It makes *tecc* run faster, and *space* somewhat slower.

-o Old mode, don't use the *unigreen* cache.

-m *maskdatafile*
Specifies the maskdatafile. Default, the maskdatafile is obtained from the process directory.

-p *process*
Specifies the process. The default process is determined by the current project directory or, if it is defined, the environment variable ICDPROCESS. *process* can be a process name, a process number or a process directory.

-s Silent mode.

-t Table-format output file goes to stdout.

-T *tclfile*
Generate *tclfile* for *tabs(1ICD)* in place of compiled technology files.

-u Enables the *unigreen* module for 3d capacitance extraction. Use this option when more than 3 dielectric interfaces are specified in the technology file (in the *dielectrics* section).

-U Enables the *unigreen* module for 3d substrate resistance extraction. Use this option when more than 2 substrate conductivity interfaces (or 3 when using metalization) are specified in the technology file (in the *sublayers* section).

-v Increases verbosity.

-V Special verbatim option to get a more readable output file.

DESCRIPTION

The program *tecc* acts as a pre-processor for technology descriptions for the circuit extraction program *space*. From a user-defined element-definition file, *tecc* produces a table-format element-definition file that can be used as input for *space*.

The user-defined input file should have the extension '.s', while the table-format output file will be called to the input file with '.t' substituted for '.s'.

THE ELEMENT DEFINITION FILE

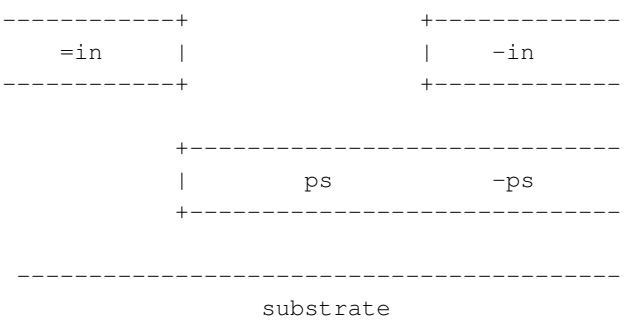
In the user-defined element-definition file the elements are listed that can be recognized from the layout description. Here, we only describe some basics of the element definition file. For a full description, see the different Space User Manuals.

For each of the elements, at least a name and a condition list has to be specified.

The name of an element is used to identify the element when error messages are generated or, for transistor elements, to identify the element in the circuit that is extracted. It is not allowed to use the same name in more than one element definition.

A condition list specifies how the presence of a particular element depends on the presence or absence of the different masks. The condition list is a boolean expression where the masks are used as the variables of the expression. In the expression, the AND operation is performed using simple concatenation, the OR operation is performed using the '|' character (AND has precedence over OR) and the INVERT operation is performed using the '!' character. Parentheses may be used to nest the expression.

References to masks in adjacent areas can be made by preceding the mask names with a '-' character, or a '=' character. The following (cross-section) gives an example.



When preceding the mask name with a '-' character, a reference is made to a mask that is in an adjacent area. When preceding the mask name with a '=' character, a reference is made to a mask that is in an area that is opposite to the area that contains the masks preceded with the '-' sign. The above can be used to define the elements that are present on the boundary between two regions of different mask combinations (e.g. edge capacitances) or elements that are present between the boundaries themselves (e.g. lateral capacitances between parallel wires).

In the following, the syntax and semantics of the element-definition file is described. The element-definition file may contain specifications for, among other things, the units, the key list, the resize list, the new list, the conductor elements, the transistor elements, the contact elements and the capacitance elements. Each of these specifications is optional but their order is fixed.

Unit specification

unit *variable value*

.

.

With one of the above commands, the unit of a variable in the conductor list, the contact list or the capacitance list is specified. The string *variable* can be either **resistance** to specify the unit for sheet resistance, **c_resistance** to specify the unit for contact resistance, **a_capacitance** to specify the unit for area capacitance, **e_capacitance** to specify the unit for edge capacitance (or - if the capacitance is specified by means of distance, capacitivity pairs - for lateral capacitance), **capacitance** to specify the unit for capacitance, **distance** to specify the unit for distance in distance, capacitivity pairs, or **resize** to specify the unit for resizing masks. The unit of each variable is specified by *value*, expressed in S.I. units.

The key list

keys: *mask1 mask2 ... maskN*

In the key list the masks are enumerated that should be used as key masks during element recognition. Based on the key masks *tecc* produces a hash table that allows *space* to find circuit elements without separately checking the element conditions that refer to the presence or absence of the key masks. Adjacent masks (that are specified by preceding them with a '-' or '=' character) might also be used as a key mask. By carefully choosing the key masks, the speed of element recognition will be optimal, while the hash table will be not too large (the size of the hash table will be equal to 2 ^ N, where N is the number of key masks).

When no key list is specified, but

maxkeys *N*

is specified instead, *tecc* will select the key masks itself. Up to a maximum of N, *tecc* will select the masks that are most frequently used for specifying the element conditions.

When neither a key list nor a maximum number of keys is specified, *tecc* will assume that maxkeys is equal to 12.

New mask specification

new: *condition-list : name*

.
.

This command allows to create a new mask from the combination of other masks. The new mask, given by *name*, is defined everywhere where the combination of masks satisfies the condition list of the specification. The characters '-' and '=' may not be used in the condition list for a new mask. Apart from the fact that the new command may be used to develop more compact element definition files, it can also be used to define one or more substrate layers.

Color definition list

colors :
mask color-name
.
.

The above color list is only used by the *Xspace* and *view3d* tools. The "color-name" must be in the X11 color database, or be a hexa-decimal RGB value with a leading '@'-sign. The color of the substrate can also be specified by using mask "@sub".

Resize mask specification

resize: *condition-list : mask : value*

.
.

This command allows to grow masks with a certain value (when a positive value is specified) or shrink masks with a certain value (when a negative value is specified). The "mask" that is specified must be in the condition list that is specified, or it must be a newly created mask name. In the last case, the new mask becomes a real mask. For one mask, more than one resize statement may be specified.

The wafer definitions

wafer: *condition-list : value1 value2 value3 [: option...]*

.
.

When using 3d substrate resistance extraction (by specifying the **-B** option to *space3d*), you may specify substrate wafer configurations for combined BEM/FEM extraction. For the FEM extraction must also one of the interconnect resistance extraction options be specified. The wafers modulate the top of the substrate. The 1-st value specifies the conductivity (in S/m), the 2-nd value specifies the thickness (in microns), and the 3-rd value specifies the number of layers (>= 1). The following options may be specified:
(a) **restype=n**, to specify a negative doped FEM type (default is 'p'); (b) **subconn=off**, to specify no FEM/BEM substrate connection (because of different type); (c) **viamask=***mask*, only on place of the via-mask is the layer stack.

The conductor list

conductors [*type*] :
name : condition-list : mask : sheet-resistivity [: carrier-type]
.

.

The conductor list contains the definitions for the conducting layers in the circuit. For each conductor specification, a specification of the actual conductor mask and a specification of the sheet-resistivity (in ohms) is required. For bipolar devices in particular, it is also necessary to specify the *carrier-type* of the conductor. The type can be 'n' for n-doped and 'p' for p-doped diffusion conductors and 'm' for metal otherwise (the default case).

Default, when extracting resistances, linear resistances will be extracted for a conductor. However, when a conductor type is specified with a conductor list, the extracted resistances for all conductors in that list will be of the specified type. An element definition file may contain more than one conductor list.

The filter list

filters :
name : masks1 : [masks2] : fmask : type
.
.

The filter list contains definitions for the possible to use filters during an extraction, which must be enabled with the *space* "enable_filter" parameter and a filter name list. (The wildcard character '∗' may be used to simplify the enabling.) The "name" identifies the filter. The groups "masks1" and "masks2" specify conductance masks (see the conductor list). The "fmask" specifies a physical mask which is used in the layout to enable the filtering (may not be a conductance mask). The "type" (a single character) specifies the filter type. There are currently two filter types possible:

Type 'c' for 2D/3D capacitance filters. The group "masks2" needs to be specified, because capacitances have 2 pins. In the group "masks2" may also be specified "@gnd" and "@sub". The capacitances between these pins are filtered out (if enabled).

Type 'r' for conductance filters. When group "masks2" is specified, then also contacts between different conductance masks can be filtered. (You may also use "@sub" in the the "masks2" group.) The conductances (and contacts) are filtered to zero ohm (if enabled).

The field-effect-transistor list

fets:
name : condition-list : mask-g mask-ds [/ mask-s] [(condition-list) [(cond-list-s)]] [: connect-b]
.
.

For a field-effect transistor (e.g. MOS or junction FET), the name and the condition list are followed by a specification of the gate mask *mask-g* and the drain/source mask *mask-ds*. Optionally, after a slash, an asymmetric source mask may be defined. The gate mask and the drain/source mask must be masks that are defined as a conductor in the conductor list. Optionally, in parentheses, a condition list for the drain/source region can be specified. For asymmetric source, in parentheses, another condition list for the source region can be specified. Further, optionally, at the end of the specification after a colon, a bulk connection *connect-b* can be specified for the transistor. This connection may consist of (1) a mask that is specified as a conductor in the conductor list, (2) the string "@sub" to denote the substrate area below the transistor gate, or (3) the notation "%(*condition-list*)" to denote a substrate area described by the condition list.

The bipolar transistor list

bjts :
name : condition-list : type : mask-e mask-b mask-c [: connect-s]
.
.

For a bipolar junction transistor, the name, the condition list and the transistor-type ("ver" for vertical or "lat" for lateral) are followed by a specification of the emitter mask *mask-e*, the base mask *mask-b* and the collector mask *mask-c*. These masks must be defined as a conductor in the conductor list. Optionally, after a colon, a substrate connection *connect-s* may be specified for the transistor. This connection may consist of (1) a mask that is specified as a conductor in the conductor list, or (2) the notation "%(*condition-list*)" to denote a substrate area described by the condition list.

The connect list

connects :
name : condition-list : mask1 mask2
.
.

The connect elements connect different semiconductor regions of the same carrier-type. They define the connectivity relation between the different conductors. *Mask1* and *mask2* are the conductor masks that are connected. Note: the connection of conductor layers via a contact or via should be specified in the contact list (see below).

The contact list

contacts [*type*] :
name : condition-list : mask1 mask2 : contact-resistance
.
.

The contact elements connect different conductors that are on top of each other. *Mask1* and *mask2* are the conductor masks that are connected by the contact. For *mask1* or *mask2* also the string "@sub" or the "%(*condition-list*)" notation may be used. The parameter at the end of the specification specifies the contact resistance in ohm for a (hypothetical) contact of 1 meter $*$ 1 meter. Default, when extracting resistances, linear resistances will be extracted for a contact. However, when a contact type is specified with the contact list, the extracted resistances for all contacts in that list will be of the specified type. An element definition file may contain more than one contact list.

The capacitance list

[**junction**] **capacitances** [*type*] :
name : condition-list : mask1 [mask2] : capacitivity
.
.

In the capacitance list, both ground and coupling capacitances can be defined. Ground capacitances are defined by using the string "@gnd", the string "@sub" or the notation "%(*condition-list*)" for either *mask1* or *mask2*. The difference between the use of the string "@gnd" and the two other cases is that (1) in the first case the ground capacitance will be connected to a node called "GND" while in the other two cases the ground capacitance will be connected to a node called "SUBSTR", and (2) - in case of substrate resistance extraction (see the Space Substrate Resistance Extraction User's Manual) - "@gnd" represents ideal ground while "@sub" and "%(*condition-list*)" represent substrate areas between substrate resistances are computed. Omitting *mask2* is equivalent to using the string "@gnd" for *mask2*.

Furthermore, capacitances are distinguished between surface capacitances, edge capacitances and lateral capacitances. To define edge capacitance, masks preceded with a '-' character are used in the condition list, and either *mask1* or *mask2* has to be preceded with a '-' character to denote an edge of an interconnection. To define lateral capacitances, masks preceded with a '-' character and mask preceded with a '=' character

are used in the condition list, and either *mask1* or *mask2* is preceded with a '-' character to denote one edge of an interconnection and the other mask is preceded with a '=' character to denote another (opposite) edge of an interconnection.

For surface capacitances, *capacitivity* is the capacitance per square meter. For edge capacitances, *capacitivity* is the capacitance per meter edge length. For lateral capacitances, the capacitance can be specified in two different ways. If only one value is specified, as in the above, *capacitivity* is the capacitance for a configuration where the spacing between two parallel wires is equal to length of the two wires. In that case, it is assumed that the lateral coupling capacitance is proportional to the distance between the two wires and inverse proportional to their spacing.

Lateral coupling capacitances can also be defined as follows.

name : condition-list : mask1 mask2 : distance1 capacitivity1
distance2 capacitivity2
.
.

In this case, the distance, capacitivity pairs specify the capacitance between two parallel wires of a length of 1 meter for different values of the distance between them. The lateral coupling capacitance for other configurations is then found from an interpolation between two distance, capacitivity pairs. For the interpolation, functions of type y = a/(x^p) and type y = a/x + b are used. If the actual distance is larger or smaller than any specified distance, an extrapolation is done using the above functions.

Default, when extracting capacitances, linear capacitances will be extracted. However, when a capacitance type is specified with the capacitance list, the extracted capacitances for all capacitance definitions in that list will be of the specified type. An element definition file may contain more than one capacitance list.

Normally, the positive node and the negative node of the extracted capacitance will be arbitrarily connected to the layers that are specified with *mask1* and *mask2*. However, if the keyword "junction" is used before the keyword "capacitance" of the capacitance list, for all elements in that list, *mask1* specifies the positive node of the element and *mask2* specifies the negative node of the element.

Default, junction capacitances will be extracted as linear capacitances. However, using the parameter **jun_caps** junction capacitances can be extracted in different ways. This is explained in more detail in the Space User's Manual.

The vertical dimension list

vdimensions:
name : condition-list : mask : value1 value2 [value3]
.
.
[**omit_cap3d** : *name1 name2* [: *capacitivity*]]
[**keep_cap2d** : *cap2d-name*]

When using 3d capacitance extraction (by specifying the **-3** option to *space3d*), it is necessary to specify the above list. The "name" specifies an unique name for the vdimension. The "condition-list" specifies the conductor mask surface condition. The "mask" field specifies the conductor mask. The 1-st value (> 0) specifies the distance between the substrate and the bottom of the conductor, and the 2-nd value (>= 0) specifies the thickness of the conductor. The 3-rd value (> 0) specifies the minimum spacing of the conductor for *tabs* usage (default: thickness / 2). Note that the unit for the values is default in meters. Another unit can be specified with the **vdimension** unit specification. See the cap3d manual when you want also use conductor (edge) shape definitions.

You can add the **omit_cap3d** option to specify that between 2 vdimensions no 3d surface capacitances must be extracted, but that the 2d capacitance method must be used. *Tecc* calculates a suitable 2d capacitivity value bases on the given dielectrics, but optional you may give a 2d surface capacitivity value. If you specify a zero value, then no surface capacitances are extracted.

You can add the **keep_cap2d** option to specify that some 2d capacitances must still be extracted during the 3d method.

The edge shape list

eshapes:
name : condition-list : mask : value1 value2
.
.

Edge extensions for the vdimension conductor masks can be specified with this list. Value1 specifies the extension of the bottom and value2 the extension of the top edge. Note that the unit for both values is default in meters. Another unit can be specified with the **shape** unit specification.

The cross-over shape list

cshapes:
name : condition-list : mask : val1 val2 val3 val4
.
.

Cross-over extensions for the vdimension conductor masks can be specified with this list. Value1 and value3 specify the extension of the bottom left and right side. Value2 and value4 specify the extension of the top left and right side. Note that the unit for the values is default in meters. Another unit can be specified with the **shape** unit.

The dielectric layers

dielectrics:
name permittivity bottom
.
.
[unigreen parameters]

When using 3d capacitance extraction (by specifying the **-3** option to *space3d*), it is not necessary to specify the *capacitance* section in the technology file. Instead, you should specify the permittivities of the dielectric layers above the substrate, in the *dielectrics* section. In the above template, *name* is the name of the dielectric layer (which you may choose freely, except that it should be unique), *permittivity* is the relative permittivity of the material, and *bottom* is the starting z-coordinate of the layer, **in microns**. The *bottom* field of the first layer should always be 0.

When using more than 3 dielectric interfaces, you should specify the **-u** option to *tecc*. This enables the "unigreen" method, which is a different kind of computation that allows for more dielectric interfaces. Be aware that the **-u** option will cause compilation of the technology file to take considerably more time.

For more information about 3d capacitance extraction and the additional unigreen parameters, please see the Space 3D Capacitance Extraction User's Manual.

The substrate layers

sublayers:
name conductivity top
.
.
[**neumann_simulation_ratio** : *value*]
[unigreen parameters]

When using 3d substrate resistance extraction (by specifying the **-B** option to *space3d*), you should specify the conductivity profile of the substrate. This is done by entering different values of the conductivity, for different layers. In the above template, *name* is the name of the layer (which may be chosen freely, except that it should be unique), *conductivity* is the conductivity of the layer (in S/m), and *top* is the z-coordinate

of the top of the layer. The first layer should have *top* equal to 0, and subsequent layers should have a negative value for *top*. The *top* value is specified **in microns**.

If the name of last layer is **metalization** , then a metal backplate is assumed, and the conductivity of the layer is ignored (since perfect conductivity will be assumed for the backplate). The conductivity value for the medium above the substrate is derived from the first conductivity value and is default divided by 100. Optional, you can specify the **neumann_simulation_ratio** to use another value.

When using more than 2 substrate interfaces (or 3 when using a metalization layer), then the **-U** option should be given to *tecc*. This option enables the "unigreen" method, which is a different kind of computation that allows for more substrate layers. Be aware that using this option will cause compilation of the technology file to take considerably more time.

For more information about substrate resistance extraction, please see the Space Substrate Resistance Extraction User's Manual. For the additional unigreen parameters, please see the Space 3D Capacitance Extraction User's Manual.

The substrate capacitance layers

subcaplayers:
name permittivity top
.
.

When using 3d substrate resistance extraction (by specifying the **-B** option to *space3d*), you can additional calculate 3d substrate capacitances. For this method you should specify the capacitance profile of the substrate. This is done by entering different values of the permittivity, for different layers. In the above template, *name* is the name of the dielectric layer (which may be chosen freely, except that it should be unique), *permittivity* is the relative permittivity of the substrate material, and *top* is the z-coordinate of the top of the layer. The first layer should have *top* equal to 0, and subsequent layers should have a negative value for *top*. The *top* value is specified **in microns**.

The self-substrate resistance list

selfsubres:
value1 value2 value3 value4
.
.

When using simple substrate resistance extraction (by specifying the **-b** option to *space*), you should specify the above list of typical interpolation values for different substrate contact dimensions. The 1-st value specifies the area (in square microns), the 2-nd value specifies the perimeter (in microns), the 3-rd value specifies the resistance (in ohms) to the substrate node, and the 4-th value specifies a ratio factor (for which the conductance must be decreased because of direct coupling). Tool *subresgen* may be used to generate automatically these values (also the coupsubres list).

The couple-substrate resistance list

coupsubres:
value1 value2 value3 value4 value5
.
.

When using simple substrate resistance extraction (by specifying the **-b** option to *space*), you should specify the above list of typical interpolation values for different substrate contact coupling situations. The 1-st and the 2-nd value specifies two substrate contact area's (in square microns), the 3-rd value specifies the minimum distance (in microns) between two contacts, the 4-th value specifies the direct coupling resistance (in ohms), and the last value specifies a ratio for which the conductance must be decreased.

Comments

Comments can be included in the element-definition file by preceding them with a '#' character. All text following the '#' character, until the end of the line, will be skipped as comment.

EXAMPLE

The following gives an example of a complete element-definition file for a cmos process.

```
# process : c5th
# author  : A.J. van Genderen, TU-Delft

unit resistance    1     # ohm
unit c_resistance  1e-12 # ohm per meter ^ 2
unit a_capacitance 1e-3  # farad per meter ^ 2
unit e_capacitance 1e-9  # farad per meter
unit capacitance   1e-18 # farad

conductors :
  # name   : condition      : mask : resistivity : type
    condM  : in             : in   : 0.040
    condP1 : ps sp          : ps   : 33
    condP2 : ps sn          : ps   : 30
    condP3 : ps !sn !sp     : ps   : 20
    condD1 : od !ps sp      : od   : 45 : p    # p+
    condD2 : od !ps sn      : od   : 30 : n    # n+
    condD3 : od !ps !sn !sp : od   : 38    # terminal

fets :
  # name : condition                     : gate d/s : bulk
    nenh : ps od sn !sp !nw !cop !con !cps : ps od
    penh : ps od !sn sp  nw !cop !con !cps : ps od

contacts :
  # name   : condition     : lay1 lay2 : resistivity
    contP  : cps in ps !od :  in   ps  :  90
    contD1 : con in !ps od :  in   od  : 270
    contD2 : cop in !ps od :  in   od  :  90

capacitances :
  # name   : condition     : lay1 lay2 : capacitivity

  # ground capacitances

    capM   : in !ps !od      : in  @gnd : 0.036
    capMe  : !in -in !ps !od : -in @gnd : 0.080

    capP   : ps !od          : ps  @gnd : 0.069
    capPe  : !ps -ps !od     : -ps @gnd : 0.051

    # diff - substr:
    capD1  : od !ps sn        : od  @gnd : 0.041
    capD1e : !od !-ps -od -sn : -od @gnd : 0.21

    # diff - well:
    capD2  : od !ps sp        : od  @gnd : 0.12
```

```
  capD2e : !od !-ps -od -sp : -od @gnd : 0.80

# coupling capacitances

  capMP  : in      ps : in ps  : 0.075
  capMPe : !in -in ps : -in ps : 0.090

  capMD1  : in      !ps od sn :  in od : 0.065
  capMD1e : !in -in !ps od sn : -in od : 0.090
  capMD2  : in      !ps od sp :  in od : 0.066
  capMD2e : !in -in !ps od sp : -in od : 0.090

# lateral coupling capacitances

  capMeMe : -in !in =in     : -in =in : 2.4
  capPePe : -ps !ps =ps     : -ps =ps : 1.8
  capMePe : -in !in !ps =ps : -in =ps : 1.2
```

AUTHOR

A.J. van Genderen, N. van der Meijs, S. de Graaf, K.J. van der Kolk, E. Matthijssen

FILES

`file.s`	input file
`file.t`	output file
`file.t.diel*`	unigreen output files
`file.t.subs*`	unigreen output files
`ICDPATH/share/lib/processcache`	unigreen system cache
`HOME/.cacd/cache/process`	unigreen user cache

UNIGREEN CACHE

The generated unigreen files are placed in a cache directory. Thus, they can be reused and don't need to be generated each time. When possible, they are placed in the system cache. When there is no write permission, they are placed in the user cache. If you don't have write permission, with the force option, you also place generated files in the user cache. The same files can be in the user and system cache. The *space3d* program shall look for the unigreen files on different places. First, in the technology directory. Second, in the user cache directory. And at last, in the system cache directory. When you don't want to use a cache, you can give option **-o** (old behaviour).

Note that the binary (blob) files don't have anymore the extension "#architecture". They have now an extension "#1" for little endian format (or "#0" for big endian).

SEE ALSO

N.P. van der Meijs, A.J. van Genderen, F. Beeftink and P.J.H. Elias, ''Space User's Manual,'' Delft University of Technology, Delft, The Netherlands.

N.P. van der Meijs and A.J. van Genderen, ''Space Tutorial,'' Delft University of Technology, Delft, The Netherlands.

A.J. van Genderen and N.P. van der Meijs, ''Space 3D Capacitance Extraction User's Manual,'' Delft University of Technology, Delft, The Netherlands.

A.J. van Genderen, N.P. van der Meijs and T. Smedes, ''Space Substrate Resistance Extraction User's Manual,'' Delft University of Technology, Delft, The Netherlands.

A.J. van Genderen and N.P. van der Meijs, ''Xspace User's Manual,'' Delft University of Technology, The Netherlands.

space(1ICD), space3d(1ICD), subresgen(1ICD), tabs(1ICD), view3d(1ICD), Xspace(1ICD).

DIAGNOSTICS

When no silent mode is specified, *tecc* will print information about the hash table that is being constructed for element recognition. This information can be used to tune the specification of the key masks.

Furthermore, *tecc* will check if legal mask names are used and if conductor masks are used with transistor, contact and capacitance definitions. During extraction itself it will be checked if appropriate conductor elements are present; for transistor elements and capacitance elements this is required, for contact elements this is not required.