NAME

tabs - technology abstraction tool

SYNOPSIS

tabs [options]

OPTIONS

-h, --help

Display a help message.

-v, --verbose

Turn on verbosity.

-V, --verboser

Turn also on *space3d* verbosity.

-q, --quiet

Don't produce informational messages. This option is cancelled by the **-v** option.

-s, --sfile FILE.s

Use the indicated technology file. By default, the default technology file of the current project is used. Note that the rules generated by *tabs* will be placed in this technology file.

-m, --maskdata FILE

Use the indicated maskdata file. By default, the default maskdata file of the current project is used.

-i, --ignore-preexisting

Ignore any preexisting capacitance rules present in the technology file.

-p, --precision low|medium|high

Use the given level of precision. Higher precision means longer computation time, in general. The default value "low" should suffice in most cases, otherwise "medium" should be fine. Note that the *precision* is not used for the *vertical* capacitances part.

-E VALUE

Sets the error-factor (default 0.1|0.05|0.05). This factor is used as stop-criterium for the values of the *edge-surface* and *lateral* capacitances.

-L VALUE

Sets the number of conducting layers over which capacitive coupling is considered significant (default 2). See below for details.

-M VALUE

Sets the **cap3d.be_mode** parameter (default 0c). (The *vertical* capacitances part uses always the default.)

-R, --edge-ratio VALUE

Sets the l/w ratio for *edge-surface* and *lateral* capacitance generation (default 10). A higher value gives possible better results, but cost more compute time.

-Z, --z-window VALUE

Sets the window in the z-direction for which capacitive coupling is significant (value in micron). This option is an alternative to the **-L** option.

--fake Fake run. It looks like a normal run without using the cache, but the extractor is not realy run. All output is generated, the surface cap values are calculated, but the rest of the values are zero.

--fix-vdim

Fix the vdimension conditions with the conductor conditions. Conductor conditions are more complex.

--no-cache

Normally, *tabs* caches its calculations in the *~/.cacd/cache/tabs4* directory. You can use this flag to disable the cache. The cache directory or the cache-files in the cache directory can be removed without consequences. But always *space3d* must be run.

--test-cache

Test against the cache to be sure that the results are the same as before. It gives also a test summary in file *./tabs_summary.txt*. And writes also results differences found to this file. You have no speed advantage of the cache, because *space3d* runs must be done.

--no-color

This option disables colored output.

--omit-vias

Don't add via conditions to vertical capacitances.

--skip-ecaps

With this option you can skip the generation part of *edge-surface* capacitances.

--skip-lcaps

With this option you can skip the generation part of *lateral* capacitances.

--add-lcaps2

To generate other *lateral* capacitances (part2).

--skip-vcaps

With this option you can skip the generation part of *vertical* capacitances. This are the *area* (surface) and *edge-edge* capacitances.

--skip-vdim NAME,...

To skip one or more vdimensions. Possible not for all vdimensions there must be generated capacitances. For example for active area's.

DESCRIPTION

The *tabs* tool allows the user to automatically produce capacitance rules for a given technology. See *tecc*(1ICD) or *space3d*(1ICD) for information on technology files. Capacitance rules enable a more efficient way of circuit extraction than full 3D extraction, at the cost of some accuracy.

When you wish to generate capacitance rules using *tabs*, first change your working directory to an appropriate project directory. *Tabs* will, by default, use the **maskdata** and **space.def.s** files of the process related to the current project. You can change this behavior by specifying the **--maskdata** and/or **--sfile** flags. (In case you specify both flags, you will not need to be in a project directory).

When *tabs* is run, it writes capacitance rules to the same technology file (extension **.s**) as used for the input. Generated rules are written to the file with a special tag, so that they can be automatically replaced when *tabs* is invoked at a later time. Hence, any pre-existing generated rules don't have to be removed from the file manually, before running *tabs.*

However, capacitance rules which were put in the technology file manually (i.e., without a special tag), should be removed, or *tabs* will complain. 'Experts' can use the **--ignore-preexisting** flag to ignore any capacitance rules which were entered manually. In that case, the manually entered capacitance rules will remain in the file, and *tabs* will add its own. It should be clear that this is not the preferred way to work, as manually entered and generated capacitance rules may conflict with each other (and this will generally result in too large capacitance values generated by space).

A minimal technology file suitable for input to *tabs* is shown below.

```
unit vdimension 1e-6

conductors :
  # name    : condition : mask : resistivity : type
```

```
    cond_mf : cmf      : cmf  : 0.045      : m
    cond_ms : cms      : cms  : 0.030      : m

vdimensions :
    v_cmf : cmf : cmf : 1.70 0.70
    v_cms : cms : cms : 2.80 0.70

dielectrics :
   # Dielectric consists of 5 micron thick SiO2
   # (epsilon = 3.9) on a conducting plane.
   SiO2   3.9   0.0
   air    1.0   5.0
```

Tabs essentially uses the **conductors** section, the **vdimensions** section, and the **dielectrics** section of the technology file. Please see *tecc*(1ICD) for more information on these sections.

Tabs computes rules for (vertical) area capacitances, (vertical and lateral) edge-edge capacitances, ground-capacitances, and (vertical) edge-surface capacitances. See the Space User's Manual for a description of these concepts. See also the sections below for more details on how the computations are performed.

The **-L** option should be used in all but the simplest cases. It allows you to specify a window in the z-direction in which capacitive coupling is considered significant. Normally, *tabs* takes into account the influences between all possible layers. This is, however, not very useful in realistic cases. For example, when your process contains several levels of metal, then the bottom-most layer of metal will, most likely, not have any significant capacitive coupling to the top-most layer. To reduce computation time, you should therefore use this option. The value specified after **-L** is the number of conductor layers which should be taken into account. In many cases, it is reasonable to use the value **1**, and in some cases you might need to use the value **2**. Higher values may not be useful in most situations.

In order for the **-L** option to work, the entries in the **vdimensions** section should be sorted according to increasing z-coordinate. (*Tabs* extracts the order of masks from the **mask** column, i.e., the third column, in the **vdimensions** section).

When the **-L** option is in effect, only the specified number of layers are taken into account when determining parasitics, as described above. If **n** is the conducting layer for which we determine parasitic coupling at a certain moment (the "victim" layer), and **k** is the number of layers specified for the **-L** option, then coupling between the victim layer **n** and layers **n-k** through **n+k** (excluding **n**) is determined by *tabs*. The metal of layers **n-k-1** and **n+k+1** is considered by *tabs* to be omnipresent, which means those layers form a shield. For (vertical) area capacitances, this means that when both the victim layer and a shielding layer are present, then a capacitance from the victim layer to the shielding layer will be generated (as normally), but when no metal is present in the shielding layer, then a capacitance from the victim layer to ground is generated. This capacitance to ground will improve the approximation of the total capacitance coupled to the victim layer.

As an alternative to the **-L** option, you may specify the window of significance in microns, using the **--z-window** option (or **-Z** for short). In this case, a reasonable value to use is **2**∗**(h+s)** where **h** is the maximum height of a conductor, and **s** is the maximum vertical spacing between two successive conductor layers. You can see what influences have been considered by *tabs* by inspecting the technology file after *tabs* has completed its computation.

Internally, *tabs* uses the 3d mode of the space extractor to compute the values for the capacitance rules, using various layout patterns. When you use the **--verbose** option, you will be able to see the output of the extraction process.

COMPUTATION OF AREA CAPACITANCES

Area capacitances are the capacitances associated with the overlap area of two (vertically) overlapping conductors. The capacitance value computed by *tabs* is the capacitance of an equivalent parallel plate capacitor (i.e., a parallel plate capacitor with the same area). The overlap area considered by *tabs* is the area in which there are no other intervening conductors. Since the actual overlap area of two conductors is known only when extracting, the area capacitances are normalized and have the unit **[F/m^2]**.

As described above, the **-L** or **--z-window** options can be used to limit the window in which overlap is considered by *tabs*, and therefore some more technicalities are involved. We will briefly discuss them now. Let us consider a "victim" conductor. It is assumed by *tabs* that two conducting ground planes (shielding planes) are always present above and below this conductor. If **n** is the number of the victim layer, and **k** is the number specified after the **-L** option, then the shielding planes are considered to be in layer **n-k-1** and **n+k+1**. (If the **--z-window** option is used instead, then the shielding layers are determined from their vertical distance from the victim layer).

The use of shielding planes is justified by the assumption that the conductor geometry in **k** layers is sufficiently dense to "catch" practically all field lines emerging from the conductor in the victim layer. For a higher value of **k**, this assumption becomes "more true".

The use of shielding layers has the following effect on the rules generated by *tabs* for (vertical) area capacitances. When both the victim layer and a shielding layer are present, then a capacitance from the victim layer to the shielding layer will be generated (as normally), but when no metal is present in the shielding layer, then a capacitance from the victim layer to ground is generated. This capacitance to ground will improve the approximation of the total capacitance coupled to the victim layer, and this will improve the simulated delay of signals propagating through metal in the victim layer.

COMPUTATION OF VERTICAL EDGE-EDGE CAPACITANCES

Vertical edge-edge capacitances are defined as the capacitances between the edges (side-walls) of two conductors on different layers, where the edges of the two conductors overlap exactly, when projected along the z-axis (in this definition, we ignore the slope which a side-wall may have). The unit of these capacitances is **[F/m]**.

Vertical edge-edge capacitances are calculated by computing the (3d) capacitance **c1** between two squared conductors of size **l*l** and calculating **c2** for two conductors of size **(2*l)*(2*l)**. The edge-edge component is then computed as **cee=(4*c1-c2)/(8*l)**.

If the **-L** or **--z-window** flag is specified, then vertical edge-edge capacitances are only computed for conductor planes which are located within the specified distance from each other.

COMPUTATION OF EDGE-SURFACE CAPACITANCES

Edge surface capacitances are defined as the capacitances between a side-wall of one conductor, and the top (bottom) of a conductor on a lower (higher) plane. The unit of these capacitances is **[F/m]**.

Space allows us to specify different values for the edge-surface capacitance, depending on the lateral distance from the side-wall in question to a nearby conductor. See the Space User's Manual for more information.

Computation by *tabs* is done by generating three rectangular conductors in the victim layer, oriented horizontally, each with a width of **w**, and a height of **l**, and at a horizontal distance of **s** from each other. A large conductor at another layer is then placed below (or above) this geometrical constellation. Now, the capacitance **c1** is calculated between the central conductor in the victim layer, and the large conductor at the other layer. The capacitance **c2** is calculated for the same constellation, but stretched by a factor **2** in the horizontal direction (increasing only the width of the conductors, not their spacing). The edge-surface capacitance is then calculated by the formula **cex = (2*c1-c2)/(2*l)**.

Different values of **cex** are calculated for increasing values of **s**, starting with **.5** times the value of the thickness of the victim layer, and increasing by a factor of **2** each time, until some large value of **s**.

When the **-L** or **--z-window** option is specified, then (shielding) ground planes are used when the distance between both conductor planes becomes too large; this works similarly w.r.t. the case of area-capacitances.

COMPUTATION OF LATERAL CAPACITANCES

Lateral capacitances are defined as the capacitances between conductors in the same layer. Lateral capacitances are only computed by *space* when there are no other intervening conductors. Lateral capacitances have unit **[F/m]**.

Tabs computes several values for the lateral capacitance, each value for a different lateral spacing **s** between the conductors. The first value of **s** used is **.5** times the value of the thickness of the victim layer, and **s** is increased by a factor **2** each time, until some large value of **s**.

EXAMPLE

The following example shows how *tabs* can be invoked.

```
% cd my-project-dir
% tabs -L 1 -s tech.s
```

PRECISION TUNING

Tabs runs the *space3d* extractor with a number of *cap3d* parameters. Which parameter values are chosen, is depended of the chosen *precision*. The following set parameter values is always used:

```
cap3d.be_mode = 0c
cap3d.max_be_area = computed-area / complexity
cap3d.be_window = infinity
```

Note that for the generation of *vertical* capacitances a fixed complexity is used (not depended of the chosen *precision*). To use **medium** *precision*, invoke *tabs* as follows:

```
% tabs -p medium
```

The *precision* calculates the **max_be_area** parameter used. How smaller the **max_be_area**, how more boundary elements are used for the layout area. For example, when one conductor plate area is divided in 16 elements, then the complete conductor has at least 2*16 + 8 elements. Thus, two conductors above each other have at least 80 elements. To calculate, a Schur matrix inversion with a dimension of 80 must be done. Note that, how bigger the dimension, how more matrix inversion computation time is needed. By the default **low** *precision* as less as possible boundary elements are used for the used models. That does not mean, that the results are per definition less accurate.

Note that the chosen *precision* and its parameters (**be-mode error-factor**) are written to
the GENERATED technology file parts.

BUGS AND CAVEATS

The progress indicator is too pessimistic (but it is only meant to give an indication anyway). Note that via's are not taken into account. When option **--omit-vias** is used, via's must be added by hand to the conditions of area capacitances. Note that you can change @gnd into @sub, if you want to connect to the substrate node. Note that the ground or substrate plane normally contains n-well area's. Thus, you can add the area to n-well capacitance rules, by deriving the values from the existing area to ground capacitances.

Report bugs to <bugs@space.tudelft.nl>

AUTHOR(S)

Kees-Jan van der Kolk, Wang Qiang, Simon de Graaf.

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.

space3d(1ICD), tabs-verify, tecc(1ICD).