ARRANGING TECHNOLOGY FILE
FOR
BEM/FEM SUPPORT

S. de Graaf

Circuits and Systems Group
Faculty of Electrical Engineering,
Mathematics and Computer Science
Delft University of Technology
The Netherlands

Report EWI-ENS 05-04
June 24, 2005

1. INTRODUCTION

This document describes the problems we had to change the technology file to add BEM/FEM wafer statements. And it also explains how *space3d* works with it.

I shall describes it for the "oscil" demo cell. This demo cell uses a cmos100 technology directory. We use in the project a lambda of 0.0125 micron.

The most important technology files are given in appendix A.
I have removed from the "space.def.s" technology file some parts. The "selfsubres" and "coupsubres" parts, which are used by simple subres extraction. The "second metal capacitances" part is left out, because "cms" is not used in our demo cell "oscil". Also the conductor for the second metal mask "cms" and the contact with "cms" is made a comment.

After a closer look to this technology file, i detected some strange specifications. Of coarse, because we don't have "cms", we can remove all references to "cms" from the capacitances. But why is "cms" specified in some of the conditions (see elements: ecap_cpg_sub, ecap_cpg_cwn, ecap_cmf_sub, ecap_cmf_cwn, ecap_cmf_caa, ecap_cmf_cpg)! The same question can i ask for "cmf", what is "cmf" doing in the conditions for elements: ecap_cpg_sub, ecap_cpg_cwn. I see also twice "!cca" in the condition for element acap_cmf_caa. I have also changed the conductivity value by sublayers from 6.7 into 10, because this is a more common used value. Also, for cap and substrate extraction "@gnd" must be changed into "@sub". Note that the "maxkeys" statement does not need to be specified (it is more or less obsolete).

In the condition list of the fets i miss the "cwn" mask. Use by the "nenh" the condition "!cwn" and by the "penh" the condition "cwn".

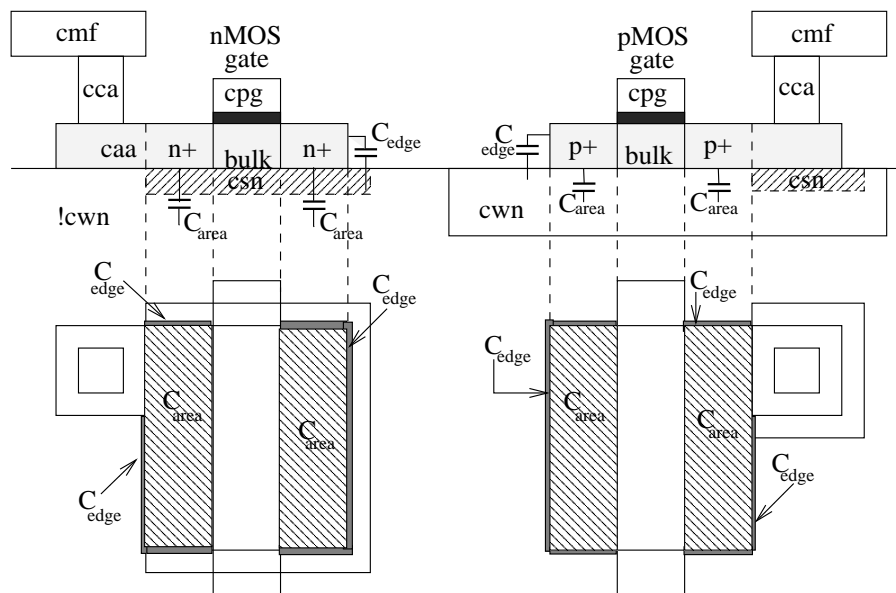See appendix B for an update of the "space.def.s" technology file.

2. TECHNOLOGY FILE DISCUSSION

In the introduction, i have already mentioned some things about the "space.def.s" technology file. When we look carefully, there are more things maybe, which are not ok.

First of all, the n-well conductor "cond_well" has an impossible low value.

Second, when we look to the contacts, we see by the metal to active area contact "cont_a" also the specification of "!cpg". This is of coarse correct, but needs maybe not to be specified. By the well contact "cont_w" is mask "caa" not specified (a don't care), but in the layout of sub-cell "inv" it is used. By the substrate contact "cont_b" is mask "caa" also not specified, but in the layout of sub-cell "inv" it is used. However, mask "caa" is not used by the substrate contact of the "sens" in cell "oscil". In the figures below i have drawn the "caa" mask below the contact area's. It makes it easier to define the substrate area in separate FEM wafer configurations. A disadvantage is maybe (see figure below), that the source/drain edge cap is not nicely completely around the source/drain area.

The junction capacitances seem good to be specified. By edge cap "ecap_na" surface condition "csn" is not specified (but a don't care), this is ok because "csn" does not always overlap (see figure). The surface condition "!cwn" could also be specified to be more accurate. But the edge condition "!-cwn" is enough. The pin "@sub" for the edge cap can be "!cwn (csn | !csn)". The edge condition "!-cpg" is important to skip the edge part where mask cpg arrives. By edge cap "ecap_pa" surface condition "cwn" is specified, this is needed to match the pin mask "cwn". The edge condition "!-csn" is important to skip the edge part where mask csn is used for a well contact (see figure). However, condition "-cwn" is not really needed. Note that we want to use a value of 600 (in place of 800) for the nwell sidewall.
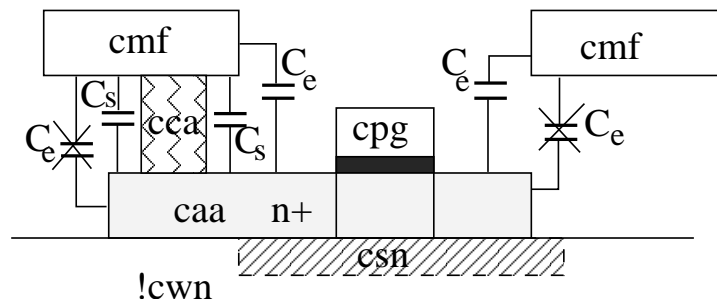
The polysilicon capacitances to different substrate configurations are ok. Of coarse there may not be "caa", else we think to find a transistor gate. However, this needs not always to be the case for all combinations, thus an "acap_cpg_caa" could also be specified. But it does not happen in our demo layout.

The metal capacitances to different substrate configurations are also all specified. These capacitances can only be there when mask "cpg" is missing. Also mask "caa" must be missing, else it is a "cmf" to "caa" capacitance. This gives a problem for the "sens" contact, where "caa" is missing, but "cca" is present. Thus, for the area cap "acap_cmf_sub", condition "!cca" needs also to be specified. And you can also add "!cca" to "acap_cmf_cwn", because the well contact does not force the use of "caa".
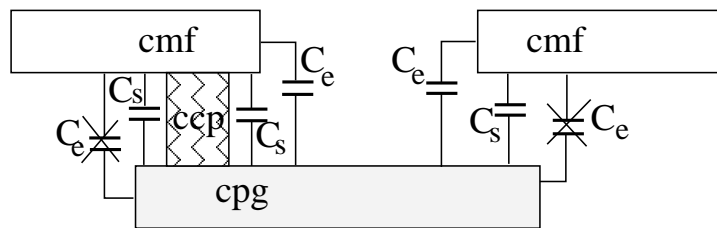
The metal capacitances to active area "caa" looks correct be specified. Now the contact mask "cca" is not forgotten. Note that edge cap "ecap_cmf_caa" does not match the following condition:

```
ecap_caa_cmf : !caa -caa cmf  !cpg  : -caa  cmf : 59
```


The same note can be given for the edge cap between "cmf" and "cpg" (see figure below). This mistake can easy be made. Thus, i must advice to add the following specification:

```
ecap_cpg_cmf : !cpg -cpg cmf        : -cpg  cmf : 59
```
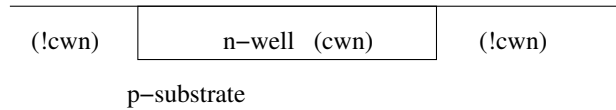

When we look to the vdimensions part, which is used by cap3D extraction, then it is also a little bit strange specified. The n-well bottom (cwn) lays higher than the active area bottom (caa). We see, that a different thickness is used for "cpg" gate and normal "cpg". Maybe, also different heights must be used for "cpg" and for "cmf" with or without "cpg".
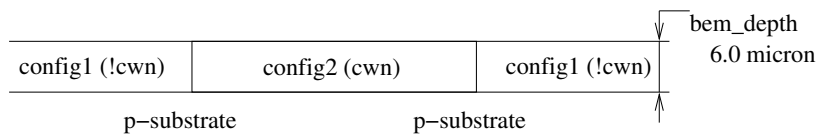
3. WAFER CONFIGURATIONS

When we look to the substrate, then we see that there is a positive substrate (p-substrate) and a negative well (n-well) part. The "cwn" mask gives this n-well part (see figure).
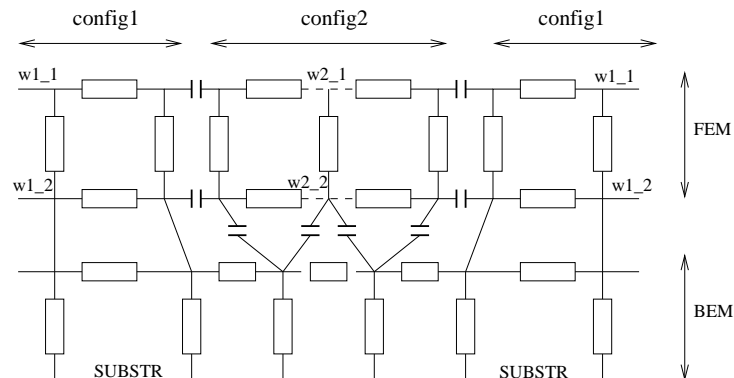

Thus, we can write two wafer configurations for it:

```
set bem_depth 6.0 # micron
wafer : !cwn : 1000 6.0 2 : restype=p subconn=on  # conductors: w1_1, w1_2
wafer :  cwn : 1000 6.0 2 : restype=n subconn=off # conductors: w2_1, w2_2
```


The two configurations are laying on top of the deep p-substrate, each configuration has a thickness of 6.0 micron. The top of the substrate is called the FEM-substrate area, which is calculated with the space3d resistance extraction method with Finite Element Mesh. The deep substrate (or bottom), starting on a bem_depth of 6.0 micron, is calculated with the accurate 3D substrate resistance extraction BEM (Boundary Element Mesh) method. This BEM method needs also a "sublayers" specification in the technology file, but note that the first top must now start at -6.0 micron. The conductances in config1 and config2 are not resistive coupled, because the areas are of different restype. And config2 is also not resistive coupled with the deep substrate. Between p- and n-types is only a capacitive coupling possible. We can translate the above figure in the following network:
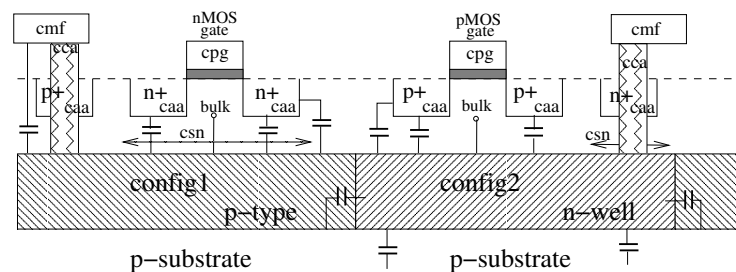

Note that the technology compiler *tecc* shall add a subcont-element for each wafer configuration (with different condition) to the technology file. Thus, there are always

substrate contact tiles, also for only capacitive coupled configurations. Also, when capacitances to substrate are not defined. The causing-conductor for config2 is in this case conductor w2_2 (for config1 is it w1_2). The technology compiler shall also add the needed extra conductor-elements and contact-elements to the technology file. The conductors w1_1, w1_2, etc. are also put with an internal new mask definition in the mask list. To use the BEM/FEM extraction method, you must use the *space3d* extraction option **-B**. However, it is independed of the choice of interconnect res options. When *space3d* detects the special conductors in the mask list, it sets bemfem extraction mode on. In this mode the substrate is always "distributed" (independent of the causing-conductor res-values). This means, that all tiles are giving separate substrate terminals.

Note that the deep substrate must be of certain restype ('p' or 'n'). The restype of the deep substrate is always equal to the restype of the first wafer configuration who is connected to it. The conductor, which lays on the bem_depth, is default connected with the substrate (with a zero contact value). When you don't want this, you must specify wafer option "subconn=off". When you don't specify "subconn=off" by wafer config2, you get a warning that config2 cannot be connected to the deep substrate, because it is of a different restype.
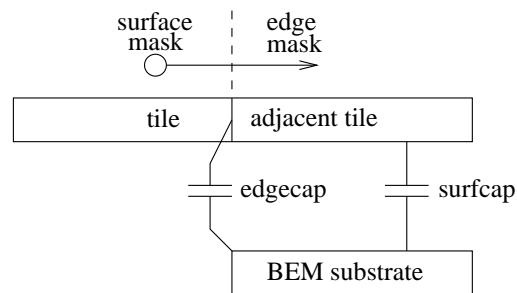
When the wafer configuration is not reaching the bem_depth, then nothing is connected to the substrate. In that case a warning message is given. Note that no wafer configuration may go deeper than the bem_depth. Thus, to make these tests possible, you must add a "set bem_depth 6.0" statement to the technology file.

The wafer configurations are an intermediate layer between the standard conductor pins and the deep substrate. Thus, all elements with @sub pins must be reconnected to the top conductor of config1 (w1_1). And all elements with cwn pins must be reconnected to the top conductor of config2 (w2_1). Note that in both above cases the other element pin is a standard conductor pin. In the special case (see junction cap nwell), whereby both pins are substrate pins, there means @sub deep substrate and is the cwn pin the bottom conductor of config2 (w2_2 in this case) for a surface element. For an edge element can each FEM conductor in the configuration side-wards be connected to FEM conductors of another configuration. But when you wants this, you needs to make a detailed connection specification with separate element values. If you do nothing, the edge cap is connected like the surface cap. See the following figure.

3.1 Special Note About Substrate Edge Capacitances

By normal substrate resistance extraction (no use of bemfem mode). The substrate terminals are separate area's, islands on the substrate surface. Thus, the *space3d* program tries always first to connect the substrate pin of an edge capacitance with the surface directly below the edge mask in the adjacent tile (see figure).

See also the source code of function updateResEdgeCap in "extract/enumpair.c". The surface side is normally tried first, only by substrate nodes the other side is first be chosen.

By bemfem mode, however, the total substrate area has substrate terminals. In this case, the *space3d* program tries first to connect the substrate pin of an edge capacitance with the surface tile not directly below the edge mask (see figure). Because the edge capacitances must not be connected on the wrong side of the substrate resistors (not be connected to the same node as the surface capacitances). Therefor, it uses the "pointStart" node point, and it connects the edgecap pin to the top/bottom fem conductor of this point.

See also the notes about edge caps in appendix D and E.

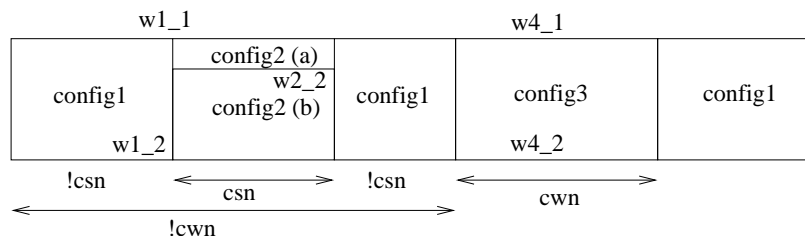### 3.2 Example With More Wafer Configurations

If you want, you can create more wafer configurations. As you see below, you can use more than one wafer statement in a wafer configuration (and you may even use different res-types). The wafers of one configuration must be put after each other, and the sum of the thickness of wafer2 and wafer3 must be equal to the *bem_depth* used. You don't need to, but you can use *new*-statements to specify the condition of each wafer configuration. Note that the "restype=p" and "subconn=on" options don't really need to be specified, because they are the defaults. For example:

```
new : !cwn !csn : config1
new : !cwn  csn : config2
new :  cwn      : config3

set bem_depth 6.0 # micron

wafer : config1 : 1000 6.0 2 : restype=p subconn=on  # w1_1, w1_2
wafer : config2 :   10 1.0 2 : restype=p             # w2_1, w2_2
wafer : config2 : 1000 5.0 2 : restype=p subconn=on  # w3_1, w3_2
wafer : config3 : 1000 6.0 2 : restype=n subconn=off # w4_1, w4_2
```
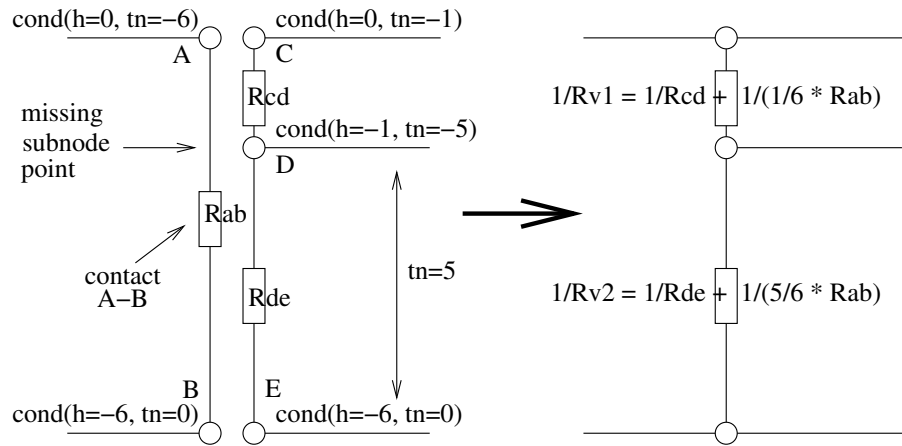

Note that each wafer conductor must have an unique conductor number to be referred. To make this possible, *tecc* has added *new* mask statements for this conductors. However, not all conductors need to have a *new* statement. For example, conductors "w1_1" and "w2_1" lay on the same height and are of the same *restype*. Thus, they are connected to each other. They have the same conductor number, but they don't need to have the same sheet resistivity. Thus, for both conductors, there is a different conductor element specified in the technology file. The same happens with the bottom conductors "w1_2" and "w3_2". Also the conductors "w2_2" and "w3_1" (between wafer2 and wafer3) can be taken together (and have the same conductor number). Because these conductors are parallel to each other, the new conductance value is the sum of each conductance value (and the sheet resistivity value is 1 / Gsum). And in this case only one conductor element needs to be specified in the technology file.

Thus, wafer config1 and config2 are connected with each other. But there is a problem, because config1 does not have a conductor on height -1.0 micron (see also the figure below). The *space3d* program shall fix this by adding an extra subnode at this height to the node point and splits the contact element between subnodes A and B in two parts.

This addition to the node point stack is easy possible, because each FEM conductor element contains also height and thickness information. See the figure below.

The conductor masks are arranged in the technology file in a certain order. First, the conductors of the standard masks. Second, the FEM conductors of the positive *restype*. Third, the FEM conductors of the negative *restype*. And at last, the @sub conductor. For this arrangement of conductor numbers, see the following technology mask list:

This conductor order is also used by the *space3d* program for the conductor list. Thus, the FEM conductors can easy be stripped off, when not used. Note that the positive FEM conductors are sorted at its height positions.

3.3 Procedure for Contact Elements

A wafer configuration can consist out of many layers. Between each layer is a contact element. Thus, many contact elements must be specified. However, the *tecc* program needs only to specify the complete contact element between the top and bottom of the wafer. The *space3d* program shall automatically cut this contact element into parts and connect them to each subnode in a node point conductor stack. For example:

```
wafer : config2 :   10 1.8 3
wafer : config2 : 1000 4.2 4
```

The first wafer has 3 layers and the second wafer has 4 layers. The last layer of the first wafer is merged together with the first layer of the second wafer. The following figure shows how the contact element splitting is done for the above specification.

3.4 Example with Hybrid Wafer Configuration

In the previous example, we saw that the n-well wafer configuration was modelled with 2 layers. When it is modelled with 1 layer, we get a special situation. In that case, the contact resistance element becomes 0 ohm. In this way, 2 layers can be modelled as 1 layer. See the figure below.

But as we want to use 1 layer for config3 (cwn), we can maybe also forget to specify the wafer configuration for cwn. We can maybe use the conductor cwn instead, in place of

specifying config3. In that case we have a hybrid situation. It is maybe not a good idea to support this, but it could easy be implemented. And it can be used as a testcase of the above 1 layer wafer situation.

To make the hybrid situation possible, there must be a substrate contact element under the cwn area. Thus, you must define a capacitance for cwn to substrate. Because *tecc* creates a substrate contact element for this capacitance. And *space3d* must not filter out substrate contact elements in bemfem mode.

3.5 Solution for N-Well Conductor

As noted before, the mask "cwn" is defined as conductor and used as conductor pin in the technology file. And the conductor is also specified as vdimension. As seen before, mask "cwn" defines also the negative FEM configuration, and conductor "cwn" is also a part of this FEM configuration. Because 3D capacitance extraction does not work with FEM conductors, but only with the standard conductors, the specification of conductor "cwn" is needed. When we want to use the same technology file for different extractions (for example **-3Cr** using the "cwn" conductor value, or **-3C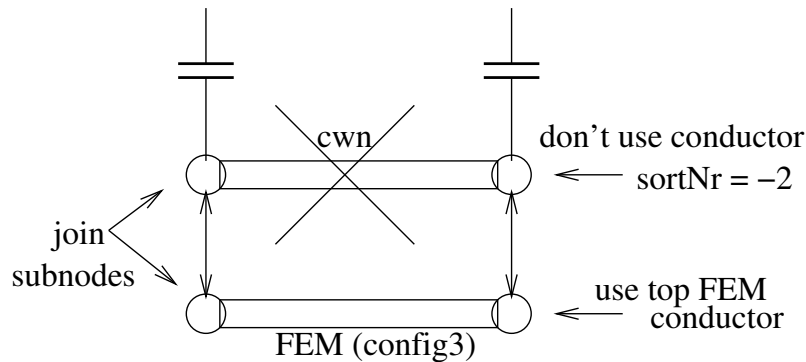rB** using the FEM wafer conductor values and not the "cwn" conductor value), then we must arrange this in the *space3d* program.

First, the *space3d* program must detect that a standard conductor is also used in a FEM configuration. When reading the technology file (see extract/gettech.cc) and there are FEM wafer configurations defined and bemfem_mode is used (because both interconnect and substrate resistance extraction is specified). When reading the conductor elements of the element table, first the conductor elements of the wafer configurations are read. The program saves the positions of the top wafer conductors. Thus, when a conductor element of a standard conductor is read, the color is tested with the top FEM conductors. When the color of the standard conductor is present in the bitmask of a FEM conductor, then the conductor element sortNr is set to -2 to flag this special situation. Note that, when this happens, also a warning message is given to the user.

Second, the *space3d* program must skip the standard conductor for interconnect res extraction (see extract/enumtile.c). Function resEnumTile puts the special sortNr -2 in array conSort before calling function triangular. There is nothing done with conductor numbers which have a negative value in the conSort array. These conductors are skipped, because now the conSort value is tested (see for example function doRectangle). Note that function parPlateCap is also using the special conSort -2 value for testing the pins. When one pin is a substrate pin and the other pin is a standard conductor pin with sortNr -2, then the standard pin is also a fem substrate pin. In that case, the substrate pin must not be changed in a top fem conductor, and the standard pin must be changed in a bottom fem conductor. Note that also for edge capacitances the sortNr must be tested (see extract/enumpair.c).

Third, after function triangular is called, the standard conductor with sortNr -2 is connected with the top fem conductor. This is done by joining the node points of the

standard conductor and the top fem conductor. See the figure below.

This is very useful for all connections which are made to the standard FEM conductor. Like drain/source contacts, other contacts and capacitance connections (in special 3D caps). However, for 3D caps, the connection to the FEM top conductor is not always the best choice. When a 3D cap is calculated between two central spiders of two FEM conductors (see figure below), the FEM conductors may have no thickness (as usual), but the real FEM conductor has always some thickness. This FEM is always represented by a number of conductors and has always a top and bottom conductor. Thus, one pin can better be assigned to the bottom FEM conductor.

The default 3D cap node assignment is not optimal (default the upper-right node is taken). Use the parameter "cap3d.cap_assign_type" to get a better cap node assignment. See report 04-07 "Space 3D Extraction Mesh Subnode Assignment Problem".
We have chosen to assign FEM and substrate terminal references to the closest top FEM subnode. See code in appendix G.

4. APPENDICES

APPENDIX A -- Most Important Technology Files

```
The project ".dmrc" file:
=========================

302
0.0125
8

The cmos100 "maskdata" file:
============================


#-----------------------------------------------------------------------------
#                      M A S K D A T A    I N F O
#
#      Layer fields (2):       field  1: layer name
#                              field  2: layer type
#                                        type = 0: normal layer
#                                        type = 1: interconnect layer (terminals etc.)
#                                        type = 2: symbolic layer
#
#      (Sea)Dali (2):          field  7: color number
#                                        0=black, 1=red, 2=green,
#                                        3=yellow, 4=blue, 5=violet, 6=aqua, 7=white
#                              field  8: fill style
#                                        0=hashed, 1=solid, 2=hollow
#                                        3,4,5 = 12,25,50% hash+outline
#                                        6,7,8 = idem, no outline
#
"scmos_n" "example double metal N-well cmos process"
#------------------------------------------------+--------------------
# Layer      | PG-Tape | CMask | Dali  | Plot   | Comment
#-----------+---------+-------+-------+-------+--------------------
# name       | job     | color | color | pen    |
# |    type | |  type | | fill| | fill| | fill|
# |    |    | | |    | | | | | | | | | | |
# 1    2    | 3  4    | 5  6  | 7  8  | 9  10 |
#-v----v----+-v--v----+-v--v--+-v--v--+-v--v--+--------------------
cpg    1      1  1      1  1    1  1    1  0    "polysilicon"
caa    1      2  1      2  1    2  1    2  0    "active area"
cmf    1      3  1      3  1    4  1    4  0    "metal"
cms    1      4  1      5  1    3  2    6  0    "metal2"
cca    0      5  1      8  1    0  1    8  0    "contact metal to diffusion"
ccp    0      7  1      8  1    0  1    8  0    "contact metal to poly"
cva    0      8  1      6  1    7  1    6  0    "contact metal to metal2"
cwn    0      9  1      4  1    5  0    5  0    "n-well"
csn    0     10  1      5  1    6  2    3  0    "n-channel implant"
cog    0     12  1      6  1    6  0    6  0    "contact to bondpads"
cx     2      0  1      0  0    7  0    8  0    "bounding box"
```

```
The cmos100 "space.def.s" technology file:
==========================================

# masks:
# cpg - polysilicon interconnect       ccp - contact metal to poly
# caa - active area                    cva - contact metal to metal2
# cmf - metal interconnect             cwn - n-well
# cms - metal2 interconnect            csn - n-channel implant
# cca - contact metal to diffusion     cog - contact to bondpads
#
# See also: maskdata

unit resistance     1     # ohm
unit c_resistance   1e-12 # ohm um^2
unit a_capacitance  1e-6  # aF/um^2
unit e_capacitance  1e-12 # aF/um
unit capacitance    1e-15 # fF
unit vdimension     1e-6  # um
unit shape          1e-6  # um

# maxkeys 13

colors :
    cpg   red
    caa   green
    cmf   blue
  # cms   gold    # not used in demo cell "oscil"
    cca   black
    ccp   black
  # cva   black   # not used in demo cell "oscil"
    cwn   glass
    csn   glass
  # cog   glass   # not used in demo cell "oscil"
    @sub  pink

conductors :
  # name    : condition     : mask : resistivity : type
    cond_mf : cmf           : cmf  : 0.045       : m    # first metal
  # cond_ms : cms           : cms  : 0.030       : m    # second metal
    cond_pg : cpg           : cpg  : 40          : m    # poly interconnect
    cond_pa : caa !cpg !csn : caa  : 70          : p    # p+ active area
    cond_na : caa !cpg  csn : caa  : 50          : n    # n+ active area
    cond_well : cwn         : cwn  : 0           : n    # n well

fets :
  # name : condition     : gate d/s : bulk
    nenh : cpg caa  csn : cpg  caa : @sub  # nenh MOS
    penh : cpg caa !csn : cpg  caa : cwn   # penh MOS

contacts :
  # name   : condition          : lay1 lay2 : resistivity
  # cont_s : cva cmf cms        : cmf  cms  :   1  # metal to metal2
    cont_p : ccp cmf cpg        : cmf  cpg  : 100  # metal to poly
    cont_a : cca cmf caa !cpg cwn !csn
           | cca cmf caa !cpg !cwn csn
                               : cmf  caa  : 100  # metal to active area
    cont_w : cca cmf cwn csn   : cmf  cwn  :  80  # metal to well
    cont_b : cca cmf !cwn !csn : cmf  @sub :  80  # metal to subs
```

```
junction capacitances ndif :
  # name    :  condition                : mask1 mask2 : capacitivity
     acap_na : caa         !cpg  csn !cwn  :  caa @gnd : 100  # n+ bottom
     ecap_na : !caa -caa !-cpg -csn !-cwn : -caa @gnd : 300  # n+ sidewall

junction capacitances nwell :
     acap_cw : cwn                   :  cwn @gnd : 100  # bottom
     ecap_cw : !cwn -cwn             : -cwn @gnd : 800  # sidewall

junction capacitances pdif :
     acap_pa : caa       !cpg  !csn cwn     :  caa cwn : 500 # p+ bottom
     ecap_pa : !caa -caa !-cpg !-csn cwn -cwn : -caa cwn : 600 # p+ sidewall

capacitances :
  # polysilicon capacitances
     acap_cpg_sub : cpg              !caa !cwn :  cpg @gnd : 49
     acap_cpg_cwn : cpg              !caa  cwn :  cpg cwn  : 49
     ecap_cpg_sub : !cpg -cpg !cmf !cms !caa !cwn : -cpg @gnd : 52
     ecap_cpg_cwn : !cpg -cpg !cmf !cms !caa  cwn : -cpg cwn  : 52

  # first metal capacitances
     acap_cmf_sub : cmf            !cpg !caa !cwn :  cmf @gnd : 25
     acap_cmf_cwn : cmf            !cpg !caa  cwn :  cmf cwn  : 25
     ecap_cmf_sub : !cmf -cmf !cms !cpg !caa !cwn : -cmf @gnd : 52
     ecap_cmf_cwn : !cmf -cmf !cms !cpg !caa  cwn : -cmf cwn  : 52

     acap_cmf_caa : cmf      caa !cpg !cca !cca :  cmf  caa : 49
     ecap_cmf_caa : !cmf -cmf caa !cms !cpg     : -cmf  caa : 59

     acap_cmf_cpg : cmf      cpg !ccp :  cmf  cpg : 49
     ecap_cmf_cpg : !cmf -cmf cpg !cms : -cmf  cpg : 59

vdimensions :
     v_caa_on_all : caa !cpg          : caa : 0.15 0.00
     v_cpg_of_caa : cpg !caa          : cpg : 0.40 0.10
     v_cpg_on_caa : cpg caa           : cpg : 0.35 0.15
     v_cmf        : cmf               : cmf : 1.00 0.35
  #  v_cms        : cms               : cms : 1.90 0.35
     v_well       : cwn               : cwn : 0.30 0

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

sublayers :
  # name       conductivity  top
     substrate 6.7           0.0
```

```
The cmos100 "space.def.p" parameter file:
==========================================

# space parameter file for scmos_n example process
#

elim_sub_con     on     # or parameter "elim_sub_term_node"
                          # Distributed contact nodes are default eliminated.
                      # To eliminate all other sub_term_node's also
                      # put this parameter "on". The result is a small R(C)
                      # network, but elimination is time consuming!
elim_sub_node    off   # currently only supported by makefem
node_pos_name    on     # ???

# the following is for non-bem/fem mode -rB (no wafers in technology file)
sub_term_distr_caa  on
sub_term_distr_cmf  on
sub_term_distr_cpg  on
sub_term_distr_cwn  on
#sep_sub_term on  # is default for distributed conductors
#cap_assign_type=2  # to get bottom and top FEM edge caps

low_sheet_res            0    # ohm   ???
min_res                  0    # ohm   ???
#max_par_res             20   # ratio ??
no_neg_res               on
min_art_degree           3
min_degree               4
min_coup_cap             0.05
lat_cap_window           6.0    # micron
max_obtuse             110.0    # degrees
equi_line_ratio          1.0

BEGIN disp   # Data for Xspace program
pair_boundary
draw_tile
END disp

BEGIN cap3d  # Data for 3D capacitance extraction
be_mode              0c
be_window            2.0
max_be_area          1.0
omit_gate_ds_cap        on
END cap3d

BEGIN sub3d  # Data for 3D BEM substrate resistance extraction
internal
be_shape             4
be_mode              0g     # ?
max_be_area          1      # ????
edge_be_ratio        0.01
edge_be_split        0.2
saw_dist             0
edge_dist            0
be_window            2.3    # ????
END sub3d
```

APPENDIX B -- Update for Technology File: space.def.s

```
junction capacitances ndif :
  # name    : condition                : mask1 mask2 : capacitivity
    acap_na : caa        !cpg  csn !cwn : caa @sub : 100  # n+ bottom
    ecap_na : !caa -caa !-cpg -csn !-cwn : -caa @sub : 300  # n+ sidewall

junction capacitances nwell :
    acap_cw : cwn                       : cwn @sub : 100  # bottom
    ecap_cw : !cwn -cwn                 : -cwn @sub : 800  # sidewall

junction capacitances pdif :
    acap_pa : caa        !cpg  !csn cwn      : caa cwn : 500 # p+ bottom
    ecap_pa : !caa -caa !-cpg !-csn cwn -cwn : -caa cwn : 600 # p+ sidewall

capacitances :
  # polysilicon capacitances
    acap_cpg_sub : cpg                 !caa !cwn :  cpg @sub : 49
    acap_cpg_cwn : cpg                 !caa  cwn :  cpg cwn  : 49
    ecap_cpg_sub : !cpg -cpg           !caa !cwn : -cpg @sub : 52
    ecap_cpg_cwn : !cpg -cpg           !caa  cwn : -cpg cwn  : 52

  # first metal capacitances
    acap_cmf_sub : cmf          !cpg !caa !cwn :  cmf @sub : 25
    acap_cmf_cwn : cmf          !cpg !caa  cwn :  cmf cwn  : 25
    ecap_cmf_sub : !cmf -cmf    !cpg !caa !cwn : -cmf @sub : 52
    ecap_cmf_cwn : !cmf -cmf    !cpg !caa  cwn : -cmf cwn  : 52

    acap_cmf_caa : cmf      caa  !cpg !cca     :  cmf  caa : 49
    ecap_cmf_caa : !cmf -cmf caa  !cpg         : -cmf  caa : 59

    acap_cmf_cpg : cmf      cpg       !ccp     :  cmf  cpg : 49
    ecap_cmf_cpg : !cmf -cmf cpg               : -cmf  cpg : 59

vdimensions :
    v_caa_on_all : caa !cpg        : caa : 0.15 0.00
    v_cpg_of_caa : cpg !caa        : cpg : 0.40 0.10
    v_cpg_on_caa : cpg caa         : cpg : 0.35 0.15
    v_cmf        : cmf             : cmf : 1.00 0.35
    v_well       : cwn             : cwn : 0.30 0

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

sublayers :
  # name       conductivity  top
    substrate  10            0.0   # !bemfem

  # substrate  10     -<bem_depth> # for bemfem wafer method
```
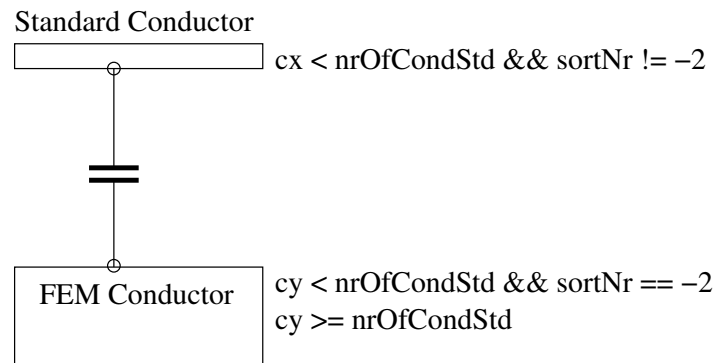
APPENDIX C -- Note to Test the Following Std/FEM Conductor Situation

Standard Conductor

cx < nrOfCondStd && sortNr != −2

FEM Conductor

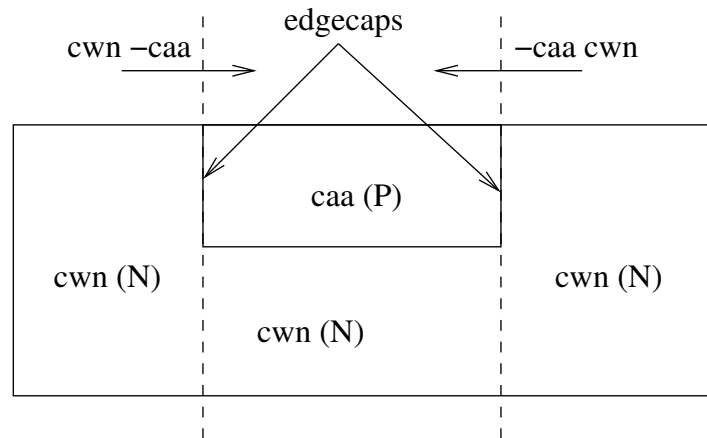cy < nrOfCondStd && sortNr == −2
cy >= nrOfCondStd

Note that in the figure above cx and cy may be swapped.

The following must be tested for the FEM conductor. The conductor must be the top FEM conductor. Thus, must be tested of the subnode in the nodepoint for the top FEM conductor exist.
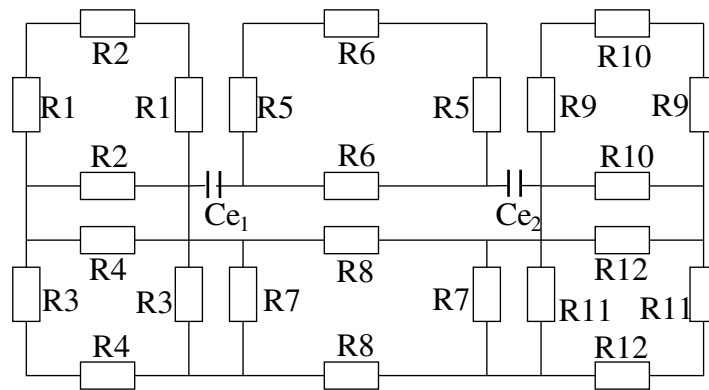
```
sn = nodepoint -> cons[cy]; /* != NULL */
if (sn -> cond -> type == 'p') top_nr = nrOfCondStd;
else /* type == 'n' */ top_nr = nrOfCondPos;
```

If cy ≥ nrOfCondStd, then 'cy' must be equal to 'top_nr'.
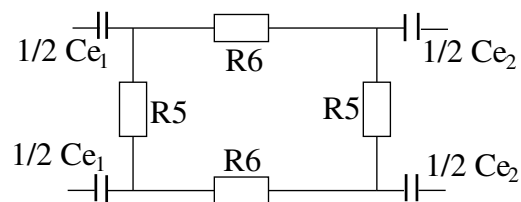If cy ≤ nrOfCondStd, then 'nodepoint -> cons[top_nr]' must contain a subnode.

APPENDIX D -- Note About FEM Edge cap_assign_type

edgecaps

cwn −caa →   ← −caa cwn

caa (P)

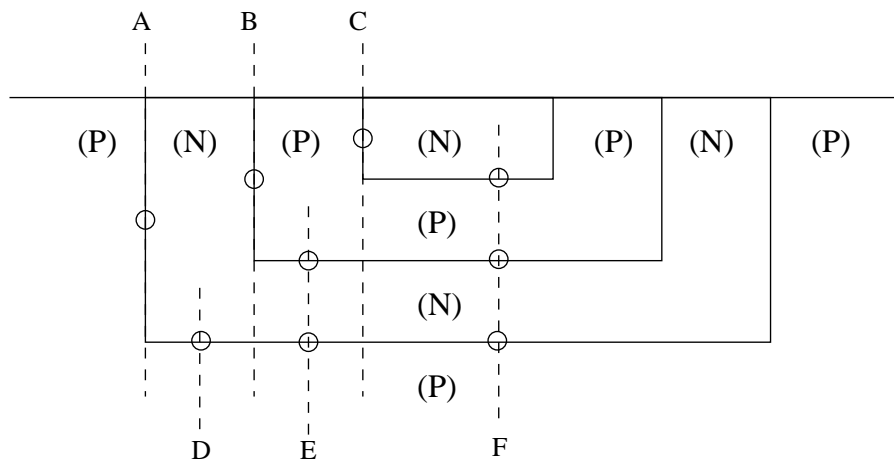cwn (N)   cwn (N)

cwn (N)

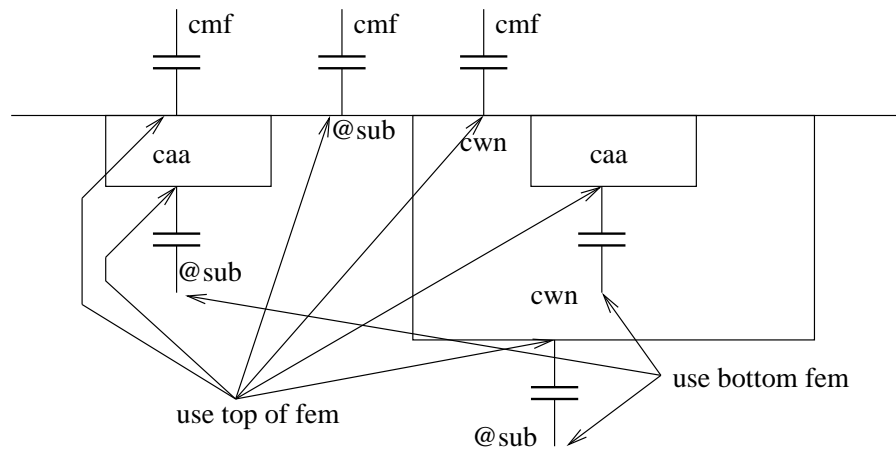cap_assign_type = 0 (1)

cap_assign_type = 2 (3)

The 2D cap_assign_type (default 0) is also used for FEM edge caps. The value 1 is used for normal edge/area 2D caps. The value 2 is now used for FEM edge caps (see above). Use the value 3, to get both the alternative normal 2D and FEM edge caps.
Note that only the top and bottom "caa" fem conductor gets an edge cap.

APPENDIX E -- Note About FEM Edge/Area Capacitances

In the figure above you see that there can be only 1 FEM edge cap on an edge position (see the positions A, B and C).

However, in a tile can be more than one FEM area cap (see positions D, E and F). For the area caps is the pin order of the defined area cap elements important. To match a N-fem to P-fem area cap correctly, the positive pin must be a N-type conductor and the negative pin must be a P-type conductor. By triple well technologies it is possible that there are more than one N/P (or P/N) area caps at the same time (see figure position F).
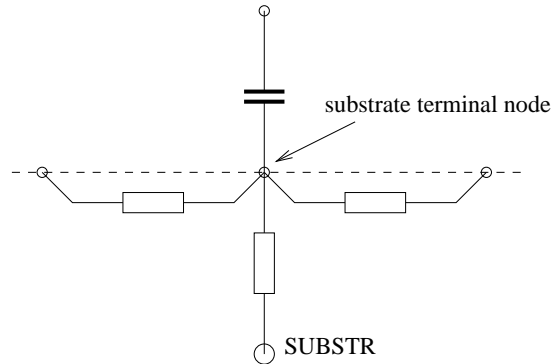
Thus, to match the correct N/P area cap, the order of the FEM conductors in the technology file is also important.

APPENDIX F -- Note About FEM Top or Bottom Conductor Choice

In the above figure a number of area caps are specified. You see area caps between the standard conductor "cmf" and the FEM area. The FEM area can be specified with a FEM conductor pin ("caa", "cwn" or "@sub"). In all this cases the top most FEM conductor of the conductor type of the FEM pin is taken and must exist.

In the other cases both pins are FEM conductors (for example: "caa @sub" and "caa cwn" and "cwn @sub). This are capacitances in the FEM area itself. For the first pin the bottom conductor of the FEM conductor of matching conductor type is taken and for the second pin the top of next FEM conductor is taken (at same height in the FEM area). The conductor type of the second pin can not be equal to the type of the first pin. Note that, when the second pin is "@sub", not always a FEM conductor needs to be there. In that case the BEM subnode needs to be taken.

APPENDIX G -- Note About FEM Top Assignment for 3D Capacitances

By extractGnd is the spider "subnode2" used to assign the bottom cap3D pin to the substrate terminal node. See code of function addCap in "spider/cap3d.c". For diffusion conductors is it only done when variable extractDiffusionCap3d is set (parameter "cap3d.omit_diff_cap" is "off"). For gates only when extractGateGndCap3d is set (parameter "cap3d.omit_gate_gnd_cap" is "off"). In all other cases it is always done.

We have decided to use the top FEM conductor for the substrate terminal node in bemfem_mode. And spiders how are using a FEM conductor are also assigned to the top FEM conductor. See the code of function spiderNew in "spider/sppair.c" below.

```
if (optRes) {
    ... // find closest nodepoint for coord x,y
    subnode = closest_np -> cons[conductor];
    if (bemfem_mode && subnode) {
        if (subnode -> cond -> sortNr == -2) { // fem conductor
            /* use top fem conductor */
#define FEM_TOP_NR(type) (type == 'p' ? nrOfCondStd : nrOfCondPos)
            conductor = FEM_TOP_NR (subnode -> cond -> type);
            subnode = closest_np -> cons[conductor];
        }
    }
    if (tile -> subcont) {
        if (bemfem_mode) { /* use top fem conductor */
            subnodeSC = closest_np -> cons[nrOfCondStd]; // try p-top
            if (!subnodeSC) subnodeSC = closest_np -> cons[nrOfCondPos]; // use n-top
        }
        else {
            subnodeSC = tile -> subcont -> subn;
        }
        ASSERT (subnodeSC);
    }
    ASSERT (subnode);
}
spider = newSpider (x, y, z, x, y, z, level, subnode, subnodeSC, conductor, isGate);
```

Note that subnGND ($\equiv$ NULL) is used, when "tile $\rightarrow$ subcont" does not exist.
The subnGND subnode is not equal to the subnSUB subnode.
Note that extract option **-c** connects all caps to subnGND and can better not be used.