

# **ADDING SUBSTRATE CAPACITANCES TO THE SPACE PROGRAM**

*S. de Graaf*

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

Report EWI-ENS 04-10  
October 5, 2004

Copyright © 2004 by the author.  
All rights reserved.

Last revision: August 30, 2005.

## 1. INTRODUCTION

To create substrate capacitances while extracting substrate resistors with option **-b** or **-B** with the *space* or *space3d* program, we added two parameters:

```
add_sub_caps      (default: "0")
```

and

```
sub_rc_const      (default: "0.0") (typical: "1e-10")
```

The substrate capacitance value is calculated with the following formula:

$$c_{\text{sub}} = \text{sub\_rc\_const} / r_{\text{sub}} = \text{sub\_rc\_const} * g_{\text{sub}}$$

Note that you can use the "add\_sub\_caps" parameter also for adding capacitances from a "subres" file (made with the *makefem* program) or from a "subcap" file (made with the *makesubcap* program). The "add\_sub\_caps" parameter is default "0" (off), thus no substrate capacitances are extracted. To use the fast rc-constant method, set the parameter "sub\_rc\_const" to a suitable value and set parameter "add\_sub\_caps" to "1". To use a more accurate 3D substrate capacitance method, set parameter "add\_sub\_caps" to "2". However, there must be a "subcaplayers" specification in the technology file. Use parameter "sub3d.makefem" to use the *makefem* program. That method is not using the "subcaplayers" specification.

You don't need to give a capacitance option anymore. Note that, when there are capacitance values in the "subres" file, these values are not always used (you need to specify "add\_sub\_caps=2").

To generate fast substrate capacitances, independent of capacitance values in the "subres" or "subcap" file, you must specify both parameters "add\_sub\_caps" and "sub\_rc\_const". Parameter "add\_sub\_caps" must be "1" and the "sub\_rc\_const" value must be greater than zero.

Note: You don't need to generate a new "subres" file to experiment with different "sub\_rc\_const" values. Use option **-%2** to run only the extract pass (see appendix B).

Appendix A gives a listing of all changes made to the source files of *space*.

I have chosen to set variables optCap and optCoupCap explicitly for the extract pass, when variable add\_sub\_caps is true (see scan/sp\_main.c).

I have also added variables extrEdgeCaps and extrSurfCaps, so that 2d capacitances are only extracted in the extract pass (see extract/enumpair.c and extract/enumtile.c), when option **-C** or **-c** is specified and only when there are such elements (see extract/gettech.cc).

Note that capacitance elements can only be added to nodes (function elemAdd), when variable optCoupCap is true in the extract pass (see also function allocNode in lump/node.c).

The substrate capacitances are added in function subContAssignRes, when variable readcap is less than 2 and variable addSubCap is true (see substr/subcont.cc).

After checkin of the sources, i found a problem with demo cell "oscil". This happens with "min\_coup\_cap" reduction (see appendix D). There exist unexpected capacitors to the GND node. This is repaired (see appendix C) by introducing a new sort number "sub\_caps\_entry" for the substrate caps. Thus, function reducGroupII can fold in this case the value of the small substrate couple cap to substrCap (see lump/reduc.c).

Note that now, the substrate caps are only used from a "subres" file, when parameter "add\_sub\_caps" is set to "2" (see substr/subcont.cc). This change was needed, else it is not possible to guess when the "sub\_caps\_entry" is used.

See appendix E for the last source files changes.

## 2. APPENDICES

### APPENDIX A -- Changes in source files

```

Index: scan/getparam.c
=====
2c2
< static char *rcsid = "$Id: getparam.c,v 4.40 2004/06/01 12:47:58 simon Exp $";
> static char *rcsid = "$Id: getparam.c,v 4.41 2004/10/05 13:18:02 simon Exp $";
55a56
> extern bool_t add_sub_caps;
65a67
> double sub_rc_const = 0.0;
154a157,158
>     add_sub_caps     = paramLookupB ("add_sub_caps",     "off");
>     sub_rc_const    = paramLookupD ("sub_rc_const",     "0.0");

Index: scan/sp_main.c
=====
2c2
< /* static char *rcsid = "$Id: sp_main.c,v 4.65 2004/10/04 14:34:19 keesjan Exp $"; */
> /* static char *rcsid = "$Id: sp_main.c,v 4.66 2004/10/05 13:18:03 simon Exp $"; */
82a83
>     optCoupCapSave = FALSE,
136a138
> bool_t add_sub_caps = 0;
140a143,144
> bool_t extrEdgeCaps = 0;
> bool_t extrSurfCaps = 0;
789a794
>     optCoupCapSave = optCoupCap;
906a912,916
>
>         extrEdgeCaps = optCapSave && hasEdgeCaps;
>         extrSurfCaps = optCapSave && hasSurfCaps;
>         if (add_sub_caps) {
>             optCap = optCoupCap = TRUE;
>         }
913a924,927
>         if (add_sub_caps) {
>             optCap = optCapSave;
>             optCoupCap = optCoupCapSave;
>         }
914a929
>             extrEdgeCaps = extrSurfCaps = 0;
941d955
<     bool_t optCoupCapSave = optCoupCap;

Index: scan/export.h
=====
1c1
< /* rcsid = "$Id: export.h,v 4.72 2004/10/04 14:34:18 keesjan Exp $" */
> /* rcsid = "$Id: export.h,v 4.73 2004/10/05 13:18:01 simon Exp $" */
62a63,64
> extern bool_t extrEdgeCaps;
> extern bool_t extrSurfCaps;

```

```

Index: extract/enumpair.c
=====
1c1
< static char *rcsid = "$Id: enumpair.c,v 4.102 2004/08/25 11:57:59 simon Exp $";
> static char *rcsid = "$Id: enumpair.c,v 4.103 2004/10/05 12:50:29 simon Exp $";
467c467
<     if (optCap && !COLOR_EQ_COLOR (tile -> color, newerTile -> color)) {
---
>     if (extrEdgeCaps && !COLOR_EQ_COLOR (tile -> color, newerTile -> color)) {
896c896
<     if (optCap) {
---
>     if (extrEdgeCaps) {

```

```

Index: extract/enumtile.c
=====
1c1
< static char *rcsid = "$Id: enumtile.c,v 4.71 2004/01/16 10:54:39 simon Exp $";
> static char *rcsid = "$Id: enumtile.c,v 4.72 2004/10/05 12:50:30 simon Exp $";
166c166
<     doSurfCap = optCap && !(optNoCore && HasCore (tile));
---
>     doSurfCap = extrSurfCaps && !(optNoCore && HasCore (tile));

```

```

Index: extract/export.h
=====
1c1
< /* rcsid = "$Id: export.h,v 4.65 2004/10/04 14:07:37 keesjan Exp $" */
> /* rcsid = "$Id: export.h,v 4.66 2004/10/05 13:25:32 simon Exp $" */
3c3
< * Copyright (C) 1988-2003 by
> * Copyright (C) 1988-2004 by
36a37,38
> extern int hasEdgeCaps;
> extern int hasSurfCaps;

```

```

Index: extract/gettech.cc
=====
1c1
< static char *rcsid = "$Id: gettech.cc,v 4.87 2004/10/05 07:58:32 keesjan Exp $";
> static char *rcsid = "$Id: gettech.cc,v 4.88 2004/10/05 13:25:33 simon Exp $";
68a69,70
> int hasEdgeCaps = 0;
> int hasSurfCaps = 0;
1362a1365,1371
>     switch (e1 -> type) {
>         case SURFCAPELEM:
>             hasSurfCaps = 1; break;
>         case EDGECAPELEM:
>         case LATCAPELEM:
>             hasEdgeCaps = 1;
>     }

```

```

Index: substr/subcont.cc
=====
1c1
< static char *rcsid = "$Id: subcont.cc,v 4.29 2004/06/21 09:03:08 simon Exp $";
> static char *rcsid = "$Id: subcont.cc,v 4.30 2004/10/05 13:36:11 simon Exp $";
41a42,43
> extern double sub_rc_const;
> extern bool_t add_sub_caps;
63a66
> static bool_t addSubCap;
141c144
<         if (readcap && optSubRes && optCoupCap) ++readcap;
---
>         if (readcap && optCoupCap) ++readcap;
142a146
>         addSubCap = (add_sub_caps && sub_rc_const > 0);
291a296,298
>         else if (addSubCap && val1) {
>             capAddSUB (sn, val1 * sub_rc_const);
>         }
331,339c338,351
<             conAddS (sn, nc -> subnTL, val1 / 4);
<             conAddS (sn, nc -> subnTR, val1 / 4);
<             conAddS (sn, nc -> subnBL, val1 / 4);
<             conAddS (sn, nc -> subnBR, val1 / 4);
<             if (readcap > 1 && val2) {
<                 capAddS (sn, nc -> subnTL, val2 / 4);
<                 capAddS (sn, nc -> subnTR, val2 / 4);
<                 capAddS (sn, nc -> subnBL, val2 / 4);
<                 capAddS (sn, nc -> subnBR, val2 / 4);
---
>             val1 /= 4;
>             conAddS (sn, nc -> subnTL, val1);
>             conAddS (sn, nc -> subnTR, val1);
>             conAddS (sn, nc -> subnBL, val1);
>             conAddS (sn, nc -> subnBR, val1);
>
>             if (readcap > 1) val2 /= 4;
>             else if (addSubCap) val2 = val1 * sub_rc_const;
>             else val2 = 0;
>             if (val2) {
>                 capAddS (sn, nc -> subnTL, val2);
>                 capAddS (sn, nc -> subnTR, val2);
>                 capAddS (sn, nc -> subnBL, val2);
>                 capAddS (sn, nc -> subnBR, val2);
350,352c362,366
<             if (readcap > 1 && val2) {
<                 capAddS (sn, nc_subn, val2);
<             }
---
>
>             if (readcap > 1) ;
>             else if (addSubCap) val2 = val1 * sub_rc_const;
>             else val2 = 0;
>             if (val2) capAddS (sn, nc_subn, val2);

```

**APPENDIX B -- Tests with cell: coilgen**

```

% cat q.s
conductors:
    metal5 : m5 : m5 : 63e-3 : m
    metal6 : m6 : m6 : 36e-3 : m
contacts:
    via5 : v5 m6 m5 : m6 m5 : 1e-12
capacitances:
    cap5sub : m5 !m4 !m3 !m2 !m1 : m5 @sub : 22.6e-6
    cap6sub : m6 !m5 !m4 !m3 !m2 !m1 : m6 @sub : 19.1e-6
sublayers:
    substrate      6      0.0

% cat q.p
BEGIN sub3d
max_be_area      10
edge_be_ratio    1
be_window        10
END sub3d

% space3d -B -Eq.t -Pq.p coilgen
% xls coilgen

network coilgen (terminal NL1, port1, port2, NR1, SL1, SE1, SW1, NW1, WU1, NW2,
                SR1, SW2, EL2, SE2, EU1, WL2, WU2, NE2, NL2, NR2, WL1, EU2,
                SL2, SR2, NE1, EL1)
{
    net {SW1, NE2};
    ...
    res 427.5816 (1, SUBSTR);
}

% space3d -B -Eq.t -Pq.p coilgen -Sadd_sub_caps -Ssub_rc_const=1e-10
% xls coilgen

network coilgen (terminal NL1, port1, port2, NR1, SL1, SE1, SW1, NW1, WU1, NW2,
                SR1, SW2, EL2, SE2, EU1, WL2, WU2, NE2, NL2, NR2, WL1, EU2,
                SL2, SR2, NE1, EL1)
{
    net {SW1, NE2};
    ...
    cap 233.8735f (1, SUBSTR);
    res 427.5816 (1, SUBSTR);
}

% space3d -%2B -Eq.t -Pq.p coilgen -Sadd_sub_caps -Ssub_rc_const=1e-12
% xls coilgen

network coilgen (terminal NL1, port1, port2, NR1, SL1, SE1, SW1, NW1, WU1, NW2,
                SR1, SW2, EL2, SE2, EU1, WL2, WU2, NE2, NL2, NR2, WL1, EU2,
                SL2, SR2, NE1, EL1)
{
    net {SW1, NE2};
    ...
    cap 2.338735f (1, SUBSTR);
    res 427.5816 (1, SUBSTR);
}

```

```
% space3d -B -Eq.t -Pq.p coilgen -Ssep_sub_term
% xsls coilgen

network coilgen (terminal NL1, port1, port2, NR1, SL1, SE1, SW1, NW1, WU1, NW2,
                 SR1, SW2, EL2, SE2, EU1, WL2, WU2, NE2, NL2, NR2, WL1, EU2,
                 SL2, SR2, NE1, EL1)
{
    net {SW1, NE2};
    ...
    res 42.81872k (1, 2);
    res 27.23131k (1, 3);
    res 581.5025k (1, 4);
    res 72.90266k (1, SUBSTR);
    res 2.19272k (2, 3);
    res 1.437982k (2, SUBSTR);
    res 18.47286k (3, 4);
    res 617.3594 (3, SUBSTR);
    res 73.96504k (4, SUBSTR);
}

% space3d -%2B -Eq.t -Pq.p coilgen -Ssep_sub_term -Sadd_sub_caps -Ssub_rc_const=1e-10
% xsls coilgen

network coilgen (terminal NL1, port1, port2, NR1, SL1, SE1, SW1, NW1, WU1, NW2,
                 SR1, SW2, EL2, SE2, EU1, WL2, WU2, NE2, NL2, NR2, WL1, EU2,
                 SL2, SR2, NE1, EL1)
{
    net {SW1, NE2};
    ...
    res 42.81872k (1, 2);
    res 27.23131k (1, 3);
    res 581.5025k (1, 4);
    cap 2.335427f (1, 2);
    cap 3.672244f (1, 3);
    cap 171.9683e-18 (1, 4);
    cap 1.371692f (1, SUBSTR);
    res 72.90266k (1, SUBSTR);
    res 2.19272k (2, 3);
    cap 45.60545f (2, 3);
    cap 69.54189f (2, SUBSTR);
    res 1.437982k (2, SUBSTR);
    res 18.47286k (3, 4);
    cap 5.413347f (3, 4);
    cap 161.9802f (3, SUBSTR);
    res 617.3594 (3, SUBSTR);
    cap 1.35199f (4, SUBSTR);
    res 73.96504k (4, SUBSTR);
}
```

**APPENDIX C -- Changes in source files 2**

Index: scan/sp\_main.c

```

=====
2c2
< /* static char *rcsid = "$Id: sp_main.c,v 4.66 2004/10/05 13:18:03 simon Exp $"; */
> /* static char *rcsid = "$Id: sp_main.c,v 4.67 2004/10/07 09:31:50 simon Exp $"; */
135a136
> int    sub_caps_entry = 0;
697a699,700
>     sub_caps_entry = (!optNoMenus || substrRes && add_sub_caps);
>
825a829
>     sub_caps_entry = 0;

```

Index: substr/subcont.cc

```

=====
1c1
< static char *rcsid = "$Id: subcont.cc,v 4.30 2004/10/05 13:36:11 simon Exp $";
> static char *rcsid = "$Id: subcont.cc,v 4.31 2004/10/07 09:31:52 simon Exp $";
144c144
<     if (readcap && optCoupCap) ++readcap;
---
>     if (readcap && add_sub_caps) ++readcap;

```

Index: lump/init.c

```

=====
1c1
< static char *rcsid = "$Id: init.c,v 4.62 2004/10/05 14:09:38 simon Exp $";
> static char *rcsid = "$Id: init.c,v 4.63 2004/10/07 09:31:47 simon Exp $";
34a35
> extern int sub_caps_entry;
327a329,330
>     if (sub_caps_entry) ++capSortTabSize;
>
390c393,405
<     ASSERT (k == capSortTabSize - 1);
---
>     if (sub_caps_entry) {
>         ++k;
>         capSortTab[k] = oldCapSortTab[0];
>         capPolarityTab[k] = 'x';
>         capSortEnable[k] = TRUE;
>         capOutFacEnable[k] = FALSE;
>         capOutFac[k] = 1.0;
>         capOutFacDimension[k] = 0;
>         capAreaPerimEnable[k] = FALSE;
>         sub_caps_entry = k;
>     }
>
>     ASSERT (k == capSortTabSize - 1);

```

```
Index: lump/lump.c
=====
1c1
< static char *rcsid = "$Id: lump.c,v 4.98 2004/08/24 16:02:46 simon Exp $";
> static char *rcsid = "$Id: lump.c,v 4.99 2004/10/07 09:31:48 simon Exp $";
34a35
> extern int sub_caps_entry;
103c104
<     elemAdd ('C', snA -> node, snB -> node, val, 0, NULL);
---
>     elemAdd ('C', snA -> node, snB -> node, val, sub_caps_entry, NULL);
111c112
<     n -> substrCap[0] += val;
---
>     n -> substrCap[sub_caps_entry] += val;
115c116
<     ASSERT (n -> substrCap[0] != 0.0);
---
>     ASSERT (n -> substrCap[sub_caps_entry] != 0.0);
```

```
Index: lump/reduc.c
=====
1c1
< static char *rcsid = "$Id: reduc.c,v 4.47 2004/08/24 16:02:48 simon Exp $";
> static char *rcsid = "$Id: reduc.c,v 4.48 2004/10/07 09:31:49 simon Exp $";
47a48
> extern int sub_caps_entry;
552,553c553,560
<         n -> gndCap[j] += cap -> val;
<         on -> gndCap[j] += cap -> val;
---
>         if (sub_caps_entry && j == sub_caps_entry) {
>             n -> substrCap[j] += cap -> val;
>             on -> substrCap[j] += cap -> val;
>         }
>         else {
>             n -> gndCap[j] += cap -> val;
>             on -> gndCap[j] += cap -> val;
>         }
```

**APPENDIX D -- Test results with demo cell: oscil**

```
% space3d -vF -P param.p -b oscil -Sadd_sub_caps -Ssub_rc_const=1e-10
```

```
network oscil (terminal in, out, vss, vdd, sens) /* WRONG RESULT */
```

```
{
  nenh w=4.4u l=800n (in, out, vss);
  ...
  res 513.9126k (8, 16);
  ...
  res 2.111757M (sens, 15);
  res 6.648565M (sens, vss);
  res 3.610183M (sens, 16);
  cap 90.0942e-18 (sens, GND);
  cap 1.142666f (sens, SUBSTR);
  res 87.51464k (sens, SUBSTR);
  res 387.7774k (15, vss);
  res 2.95793M (15, 16);
  cap 257.8799e-18 (15, vss);
  cap 81.16136e-18 (15, GND);
  cap 2.840396f (15, SUBSTR);
  res 35.20636k (15, SUBSTR);
  res 229.6561k (vss, 16);
  cap 435.4337e-18 (vss, 16);
  cap 15.04084e-18 (vss, GND);
  cap 7.310453f (vss, SUBSTR);
  res 13.67904k (vss, SUBSTR);
  cap 97.56765e-18 (16, GND);
  cap 2.666395f (16, SUBSTR);
  res 37.50382k (16, SUBSTR);
}
```

```
% space3d -vF -P param.p -b oscil -Sadd_sub_caps -Ssub_rc_const=1e-10
```

```
network oscil (terminal in, out, vss, vdd, sens) /* GOOD RESULT */
```

```
{
  nenh w=4.4u l=800n (in, out, vss);
  ...
  res 513.9126k (8, 16);
  ...
  res 2.111757M (sens, 15);
  res 6.648565M (sens, vss);
  res 3.610183M (sens, 16);
  cap 1.23276f (sens, SUBSTR);
  res 87.51464k (sens, SUBSTR);
  res 387.7774k (15, vss);
  res 2.95793M (15, 16);
  cap 257.8799e-18 (15, vss);
  cap 2.921557f (15, SUBSTR);
  res 35.20636k (15, SUBSTR);
  res 229.6561k (vss, 16);
  cap 435.4337e-18 (vss, 16);
  cap 7.325494f (vss, SUBSTR);
  res 13.67904k (vss, SUBSTR);
  cap 2.763963f (16, SUBSTR);
  res 37.50382k (16, SUBSTR);
}
```

**APPENDIX E -- Last changes in source files**

Index: scan/getparam.c

```

=====
2c2
< static char *rcsid = "$Id: getparam.c,v 4.44 2005/04/25 07:43:17 simon Exp $";
> static char *rcsid = "$Id: getparam.c,v 4.45 2005/08/23 06:43:54 simon Exp $";
52c52
< extern bool_t add_sub_caps;
---
> extern int add_sub_caps;
138c138
<     add_sub_caps     = paramLookupB ("add_sub_caps",     "off");
---
>     add_sub_caps     = paramLookupI ("add_sub_caps",     "0");

```

Index: scan/sp\_main.c

```

=====
2c2
< /* static char *rcsid = "$Id: sp_main.c,v 4.76 2005/06/10 19:15:28 simon Exp $"; */
> /* static char *rcsid = "$Id: sp_main.c,v 4.77 2005/08/23 06:43:57 simon Exp $"; */
132a131,132
> extern int subcap_cnt;
> extern double sub_rc_const;
135c135
< bool_t add_sub_caps = 0;
---
> int     add_sub_caps = 0;
688c688,700
<     sub_caps_entry = substrRes && add_sub_caps;
---
>     if (!substrRes) add_sub_caps = 0;
>     else if (add_sub_caps == 1 && sub_rc_const <= 0 || add_sub_caps > 1
&& subcap_cnt <= 0 && !optSubResMakefem) {
>         if (add_sub_caps == 1)
>             say ("warning: not adding substrate capacitances, parameter sub_rc_const must
be > 0");
>         else
>             say ("warning: not adding substrate capacitances, technology data must contain
subcaplayers");
>         add_sub_caps = 0;
>     }
>     else if (add_sub_caps > 1 && optSubRes && !optOnlyPrePassB1
&& (optSubResInternal || optSubResOldMesh)) {
>         say ("warning: not adding substrate capacitances, not supported in %s mode",
optSubResInternal ? "internal" : "old_mesh");
>         add_sub_caps = 0;
>     }
>     sub_caps_entry = add_sub_caps;
1213a1219,1222
>     if (add_sub_caps > 1) {
>         say ("warning: not adding substrate capacitances, subcap3d not supported for -b");
>         add_sub_caps = 0;
>     }
1263a1273,1278
>
>     if (!optSubResMakefem && add_sub_caps > 1) {
>         av[0] = "makesubcap";
>         if (optVerbose) verbose ("running: %s ...\n", av[0]);
>         run2 (av, "makesubcap: ");
>     }

```

```

Index: extract/gettech.cc
=====
1c1
< static char *rcsid = "$Id: gettech.cc,v 4.92 2005/04/19 07:11:19 simon Exp $";
> static char *rcsid = "$Id: gettech.cc,v 4.93 2005/08/23 07:07:27 simon Exp $";
158a158
> substrate_t * subcaps;
162a163
> int subcap_cnt;
358c359
<     if (majorPrNr > MajorProtNr || minorPrNr != 2) {
>     if (majorPrNr > MajorProtNr || minorPrNr < 2 || minorPrNr > 3) {
1461a1478,1497
>     if (minorPrNr > 2 && (subcap_cnt = getnum (fp_tech)) > 0) {
>         subcaps = NEW (substrate_t, subcap_cnt);
>         for (i = 0; i < subcap_cnt; i++) {
>             if (fscanf (fp_tech, "%s %le %le", subcaps[i].name,
>                 &subcaps[i].conduc, &subcaps[i].top) != 3) techReadError (16);
>             while ((c = getc (fp_tech)) != '\n' && c != EOF);
>         }
>     }

Index: substr/subcont.cc
=====
1c1
< static char *rcsid = "$Id: subcont.cc,v 4.32 2004/10/22 13:20:18 simon Exp $";
> static char *rcsid = "$Id: subcont.cc,v 4.33 2005/08/23 06:40:39 simon Exp $";
43c43
< extern bool_t add_sub_caps;
> extern int add_sub_caps;
57c57,60
< static int readcap, next_nr, next_grp;
---
> static DM_STREAM *dmsSubCap = NULL;
> static int readcap;
> static int next_nr, next_grp;
> static int Next_nr, Next_grp;
58a62
> static coord_t Next_xl, Next_yb;
67d69
< static bool_t addSubCap;
144,145c146,160
<         if (getc (dmsSubRes -> dmfp) != '\n') ++readcap;
<         if (readcap && add_sub_caps) ++readcap;
---
>         if (getc (dmsSubRes -> dmfp) != '\n') readcap = 1;
>     }
>
>     if (add_sub_caps > 1) {
>         if (!readcap) dmsSubCap = dmOpenStream (lkey, "subcap", "r");
>     }
>     else if (add_sub_caps) {
>         ASSERT (sub_rc_const > 0);
>     }
>
>     if (dmsSubCap) {
>         if (fscanf (dmsSubCap -> dmfp, "%*s %d %d %*s %ld %*s %ld",
>             &Next_nr, &Next_grp, &Next_xl, &Next_yb) < 1) Next_xl = Inf;
>         else ASSERT (Next_nr == next_nr && Next_yb == next_yb);
>         ASSERT (Next_xl == next_xl);
147d161
<         addSubCap = (add_sub_caps && sub_rc_const > 0);
203a218
>     dmCLOSE (dmsSubCap);

```

```

278a294
> if (dmsSubCap && fscanf (dmsSubCap -> dmfp, "%*s %le", &val2) != 1) ASSERT (0);
290c306
<   if (readcap > 1) {
----
>   if (add_sub_caps > 1) {
294c310
<       say ("found zero capacitance to substrate node in subres file");
----
>       say ("found zero capacitance to substrate node in %s file",
            dmsSubCap ? "subcap" : "subres");
297c313
<   else if (addSubCap && val1) {
----
>   else if (add_sub_caps && val1) {
308a325,329
>   if (dmsSubCap) {
>       if (fscanf (dmsSubCap -> dmfp, "%s %d %*s %le", buf, &j, &val2) != 3) ASSERT (0);
>       ASSERT (strsame (buf, "nc"));
>       ASSERT (i == j);
>   }
334c355
<   if (readcap > 1 && !val2 && !(sub_param_message & 8)) {
----
>   if (add_sub_caps > 1 && !val2 && !(sub_param_message & 8)) {
336c357
<       say ("found zero capacitance to neighbor node in subres file");
----
>       say ("found zero capacitance to neighbor node in %s file",
            dmsSubCap ? "subcap" : "subres");
345,346c366,367
<   if (readcap > 1) val2 /= 4;
<   else if (addSubCap) val2 = val1 * sub_rc_const;
----
>   if (add_sub_caps > 1) val2 /= 4;
>   else if (add_sub_caps) val2 = val1 * sub_rc_const;
364,365c385,386
<   if (readcap > 1) ;
<   else if (addSubCap) val2 = val1 * sub_rc_const;
----
>   if (add_sub_caps > 1) ;
>   else if (add_sub_caps) val2 = val1 * sub_rc_const;
384a406,414
>
>   if (dmsSubCap) {
>       if (fscanf (dmsSubCap -> dmfp, "%s %d", buf, &j) != 2) ASSERT (0);
>       ASSERT (strsame (buf, "nr_neigh"));
>       if (fscanf (dmsSubCap -> dmfp, "%*s %d %d %*s %ld %*s %ld",
>           &Next_nr, &Next_grp, &Next_xl, &Next_yb) < 1) Next_xl = Inf;
>       else ASSERT (Next_nr == next_nr && Next_yb == next_yb);
>       ASSERT (Next_xl == next_xl);
>   }
=====

```