

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 04-04
June 21, 2004

Copyright © 2004 by the author.

Last revision: June 22, 2004.

1. INTRODUCTION

By the substrate resistance extraction with the *makefem* program (using FEMLAB), there are also substrate capacitances (for certain frequency) calculated and written to the “image part” of output file "femsub.out" (see also report 04-01 "Space 3D Substrate Extraction by Using FEMLAB").

The *makefem* program shall add these capacitance values between the substrate terminal nodes to the "subres" file. Thus, the *space3d* program must be modified to read this new "subres" file format. And, if requested, *space3d* must output these substrate capacitances as well to the circuit "net" and "mc" streams.

Now, the substrate capacitances are outputted in the extract pass, when option **-C** is used.

Appendix A gives a listing of the modifications made to the *space3d* program.

The sources are checked in at June 21, 2004.

Note that the *makefem* program can only be used with this new version. Older versions of *space* cannot read the new "subres" file format.

1.1 A Short Explanation of the Source Changes

Like the way substrate conductance values are read in the extract pass by function `subContAssignRes` (in file "subcont.cc"), we can also read the substrate capacitance values.

By first read of a "subres" file record (by function `initSubstr`), a check is done for a capacitance value after the conductance value. When found, the "readcap" flag is set to 1. And when the capacitance values must be used, the flag is set to 2.

I added two functions, `capAddSUB` and `capAddS`. For the capacitance to SUBSTR, there was already a `substrCap` array in the node structure. However, this array is only allocated (see `createNode`) when `optCoupCap` is set.

Note that the substrate capacitances must also be distributed by distributed contacts. See functions `subnodeSubcontReconnect` and `subnodeSubcontEmpty` in file "lump.c".

Note that only sort number 0 is used for substrate capacitances (like the conductances).

1.2 Test Results

I have tested the implementation with cell "coilgen" (see appendix B, C and D). Note that i used the test bench of *makefem* to create output file "femsub.out" (no real FEMLAB run was done). Note that program *sub2out* has created the “image part” out of the values of the “real part”.

2. APPENDICES

APPENDIX A -- Overview of source file changes

```

Index: lump/export.h
=====
diff -r4.67 export.h
3c3
< * Copyright (C) 1988-2003 by
---
> * Copyright (C) 1988-2004 by
30c30,32
< void conAddS P_((subnode_t *subnA, subnode_t *subnB, double val, int sort));
---
> void capAddS P_((subnode_t *subnA, subnode_t *subnB, double val));
> void capAddSUB P_((subnode_t *sn, double val));
> void conAddS P_((subnode_t *subnA, subnode_t *subnB, double val));

Index: lump/lump.c
=====
diff -r4.96 lump.c
79,81c79
< void conAddS (subnA, subnB, val, sort) subnode_t *subnA, *subnB;
< double val;
< int sort;
---
> void conAddS (subnA, subnB, val) subnode_t *subnA, *subnB; double val;
85c83
<     elemAdd ('S', subnA -> node, subnB -> node, val, sort, NULL);
---
>     elemAdd ('S', subnA -> node, subnB -> node, val, 0, NULL);
101a100,116
>
> void capAddS (snA, snB, val) subnode_t *snA, *snB; double val;
> {
>     elemAdd ('C', snA -> node, snB -> node, val, 0, NULL);
>     inCap++;
> }
>
> void capAddSUB (sn, val) subnode_t *sn; double val;
> {
>     node_t *n = sn -> node;
>
>     n -> substrCap[0] += val;
>
>     /* The substrCap[0] value may not become zero because otherwise
>        it will be eliminated (too soon). */
>     ASSERT (n -> substrCap[0] != 0.0);
> }
////////// subnodeSubcontReconnect //////////
1367a1383,1393
>
>     if (nS -> substrCap) {
>         if (nS -> substrCap[0])
>             nD -> substrCap[0] += nS -> substrCap[0] * frac;
>
>         for (i = 0; i < capSortTabSize; i++) {
>             for (el = nS -> cap[i]; el; el = NEXT (el, nS)) {
>                 elemAdd ('C', nD, OTHER (el, nS), el -> val * frac, i, NULL);
>             }
>         }
>     }

```

```

////////// subnodeSubcontEmpty //////////
1392a1419,1428
>
>     if (node -> substrCap) {
>         node -> substrCap[0] = 0.0;
>         for (i = 0; i < capSortTabSize; i++) {
>             for (el = node -> cap[i]; el; el = el_next) {
>                 el_next = NEXT (el, node);
>                 elemDel (el, i);
>             }
>         }
>     }

Index: spider/cap3d.c
=====
diff -r4.46 cap3d.c
919c919
<         if (subn1 != subn2) conAddS (subn1, subn2, -val, 0);
---
>         if (subn1 != subn2) conAddS (subn1, subn2, -val);

Index: substr/subcont.cc
=====
diff -r4.28 subcont.cc
42a43
> extern bool_t optCoupCap;
54c55
< static int next_nr, next_grp;
---
> static int readcap, next_nr, next_grp;
55a57
> static double val1;
////////// initSubstr //////////
122a125,126
>     int i;
>
131,132c135,142
<     if (fscanf (dmsSubRes -> dmfp, "%*s %d %d %*s %ld %*s %ld",
<             &next_nr, &next_grp, &next_xl, &next_yb) < 1) next_xl = INF;
---
>     readcap = 0;
>     if ((i = fscanf (dmsSubRes -> dmfp, "%*s %d %d %*s %ld %*s %ld %*s %le",
>             &next_nr, &next_grp, &next_xl, &next_yb, &val1)) < 1) next_xl = INF;
>     else {
>         ASSERT (i == 5);
>         if (getc (dmsSubRes -> dmfp) != '\n') ++readcap;
>         if (readcap && optSubRes && optCoupCap) ++readcap;
>     }
////////// subContAssignRes //////////
253c263,264
<     int i, cnt;
---
>     double val2;
>     int i, j, cnt;
255d265
<     double val;
262d271
<     if (fscanf (dmfp, "%*s %le", &val) != 1) ASSERT (0);
264c273,275
<     if (optEstimate3D) val = 1e99;
---
>     if (readcap && fscanf (dmfp, "%*s %le", &val2) != 1) ASSERT (0);
>
>     if (optEstimate3D) val1 = 1e99;

```

```

266c277
<      if (val == 0.0) {
---
>      if (vall == 0.0) {
273c284,291
<          conAddSUB (sn, val);
---
>          conAddSUB (sn, vall);
>      if (readcap > 1) {
>          if (val2) capAddSUB (sn, val2);
>          else if (!(sub_param_message & 4)) {
>              sub_param_message |= 4;
>              say ("found zero capacitance to substrate node in subres file");
>          }
>      }
281c299,300
<      if (fscanf (dmfp, "%d %s %le", &i, &val) != 2) ASSERT (0);
---
>      if (fscanf (dmfp, "%d %s %le", &i, &vall) != 2) ASSERT (0);
>      if (readcap && fscanf (dmfp, "%s %le", &val2) != 1) ASSERT (0);
300c319
<      if (val == 0.0) {
---
>      if (vall == 0.0) {
306a326,329
>          if (readcap > 1 && !val2 && !(sub_param_message & 8)) {
>              sub_param_message |= 8;
>              say ("found zero capacitance to neighbor node in subres file");
>          }
308,311c331,340
<          conAddS (sn, nc -> subnTL, val / 4, 0);
<          conAddS (sn, nc -> subnTR, val / 4, 0);
<          conAddS (sn, nc -> subnBL, val / 4, 0);
<          conAddS (sn, nc -> subnBR, val / 4, 0);
---
>          conAddS (sn, nc -> subnTL, vall / 4);
>          conAddS (sn, nc -> subnTR, vall / 4);
>          conAddS (sn, nc -> subnBL, vall / 4);
>          conAddS (sn, nc -> subnBR, vall / 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);
>          }
320c349,352
<          conAddS (sn, nc_subn, val, 0);
---
>          conAddS (sn, nc_subn, vall);
>          if (readcap > 1 && val2) {
>              capAddS (sn, nc_subn, val2);
>          }
335,336c367,369
<      if (fscanf (dmfp, "%s %d %d %s %ld %s %ld",
<          &next_nr, &next_grp, &next_xl, &next_yb) < 1) next_xl = INF;
---
>      if ((i = fscanf (dmfp, "%s %d %d %s %ld %s %ld %s %le",
>          &next_nr, &next_grp, &next_xl, &next_yb, &vall)) < 1) next_xl = INF;
>      else ASSERT (i == 5);

```

APPENDIX B -- Network using space3d -BC with makesubres

```

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 23.25625f (SW1, 1);
    cap 55.76297f (SW1, 2);
    cap 383.7266e-18 (SW1, 3);
    cap 353.5486e-18 (SW1, 4);
    res 42.81872k (1, 3);
    res 2.19272k (1, 2);
    res 1.437982k (1, SUBSTR);
    res 27.23131k (2, 3);
    res 18.47286k (2, 4);
    res 617.3594 (2, SUBSTR);
    res 581.5025k (3, 4);
    res 72.90266k (3, SUBSTR);
    res 73.96504k (4, SUBSTR);
}

```

APPENDIX C -- Network using space3d -BC with makefem

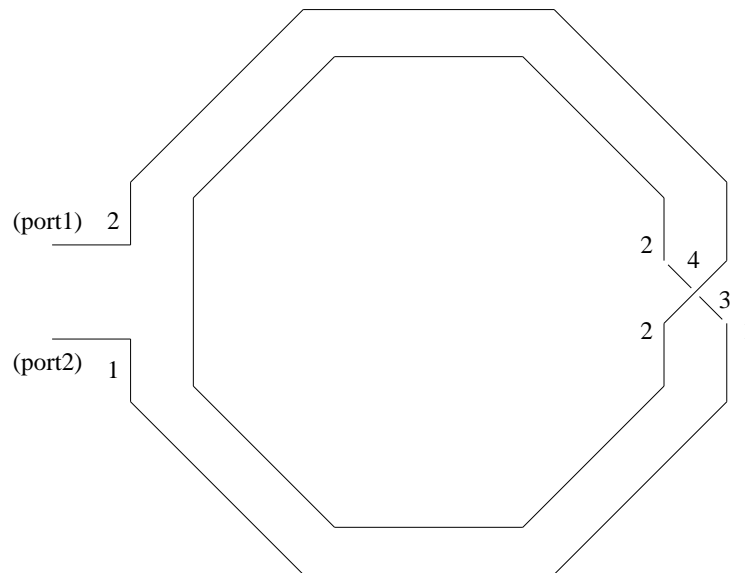
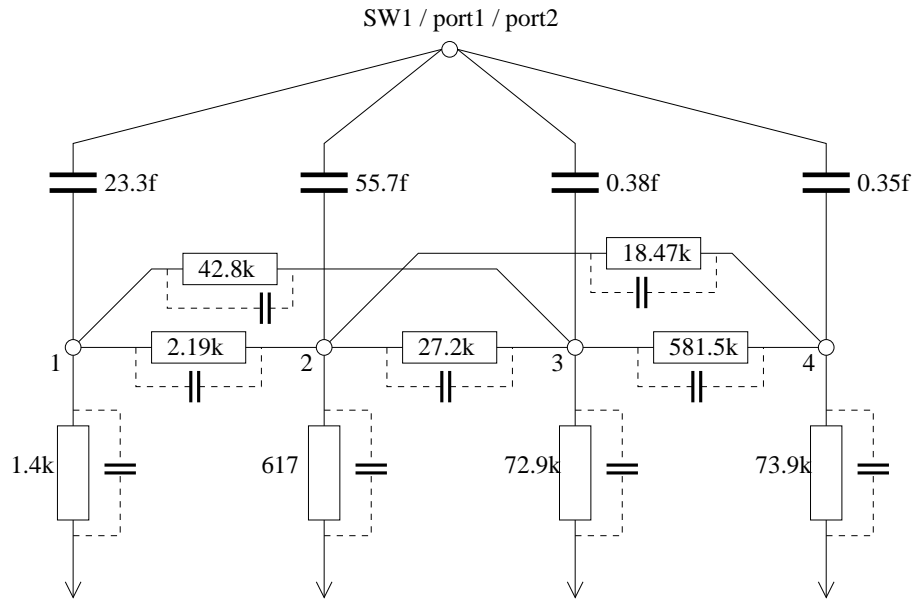
```

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 23.25625f (SW1, 1);
    cap 55.76297f (SW1, 2);
    cap 383.7266e-18 (SW1, 3);
    cap 353.5486e-18 (SW1, 4);
    res 42.81872k (1, 3);
    res 2.19272k (1, 2);
    cap 3.716948u (1, 3);
    cap 72.58333u (1, 2);
    cap 110.6794u (1, SUBSTR);
    res 1.437982k (1, SUBSTR);
    res 27.23131k (2, 3);
    res 18.47286k (2, 4);
    cap 5.844558u (2, 3);
    cap 8.615609u (2, 4);
    cap 257.7995u (2, SUBSTR);
    res 617.3594 (2, SUBSTR);
    res 581.5025k (3, 4);
    cap 273.6961n (3, 4);
    cap 2.183116u (3, SUBSTR);
    res 72.90266k (3, SUBSTR);
    cap 2.151759u (4, SUBSTR);
    res 73.96504k (4, SUBSTR);
}

```

NOTE: This network contains substrate capacitances.
The values are not real, but test values.

APPENDIX D -- Cell coilgen network and layout sketch



The interconnect layers "m5" and "m6" are connected with via's to each other. Thus, all terminals are in the same net. There is no interconnect resistance calculated. The capacitors to "1" and "2" are for "m5" interconnect ("m6" is also above "m5"). The capacitors to "3" and "4" are for "m6" interconnect (because there is only "m6"). Net "1" is not connected by a substrate resistor to net "4", because the be_window is too small.