SPACE 3D SUBSTRATE EXTRACTION BY USING MAKESUBCAP

S. de Graaf

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

Report EWI-ENS 05-05
August 11, 2005

Last revision: September 1, 2005.

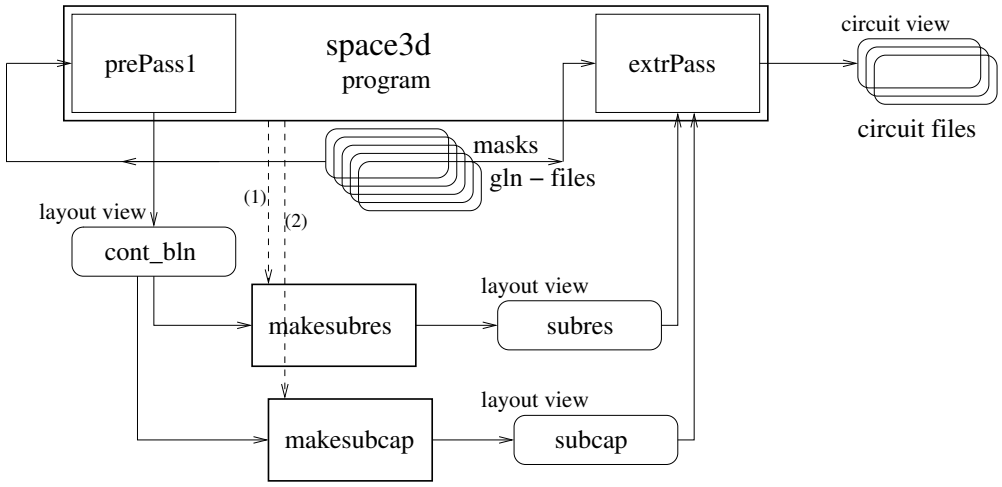
1. INTRODUCTION

Normally, the *space3d* layout to circuit extractor uses for accurate (3D) substrate resistance extraction (option **-B**) the *makesubres* program to calculate the substrate resistances. See also the "Space 3D Substrate Extraction Application Note" (report EWI-ENS 03-04 and 04-06).

As an addition to the 3D substrate res method, we can also calculate 3D substrate capacitances. I have modified the *makesubres* program sources to make this *makesubcap* program. You only need to compile the sources with the define MAKESUBCAP.

For the calculation of 3D capacitances, the *makesubcap* program uses the same input data as the *makesubres* program. Thus, in this implementation, the *makesubcap* program must also use the same "sub3d.be_window". Thus, the same initial substrate spider mesh is used. But possible is chosen for another mesh refinement and calculation method. But the result is mapped to the same number of substrate terminals. Note that also the sub3d Green's functions are used for the calculation. The result values are written in the cell layout view "subcap" file.

To use these results, the *space3d* extract pass must also read this "subcap" file and add these substrate capacitances into the extracted circuit network (like the substrate resistors). Set parameter "add_sub_caps" to "2" to let this happen. See the figure below.

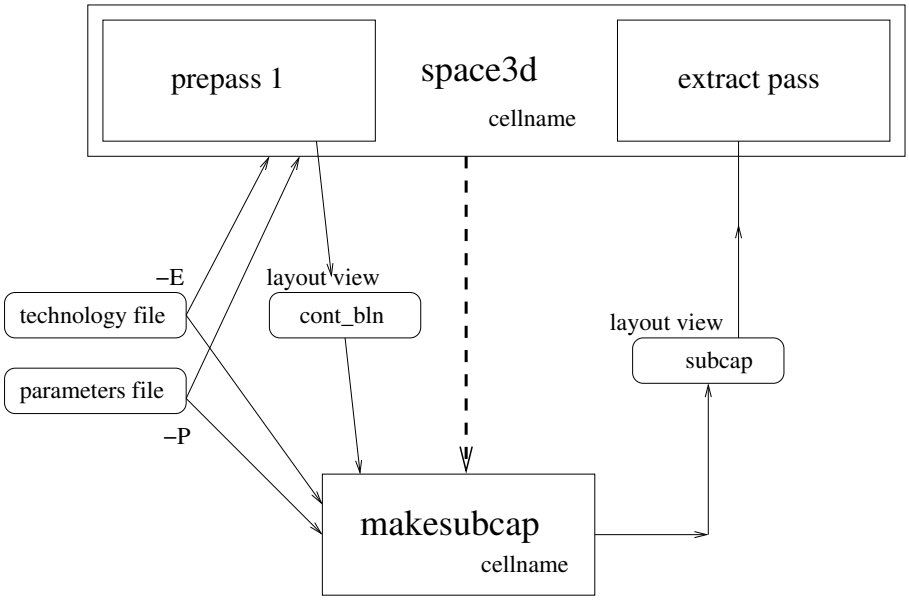


Note that the *makesubcap* program must use the same extraction environment as the *space3d* program. Thus the same *space* parameter file, technology file and cell name (and project path) must be used.

When the *makesubcap* program sources are made free in the public domain, the user can easy modify these sources to make his/her own program. For a more detailed explanation of the *makesubcap* user program, see the following section.

2. THE MAKESUBCAP PROGRAM

The *makesubcap* program reads the substrate contacts and calculates the substrate capacitance values. The calculated substrate capacitance values are written to stream "subcap". For an overview see the following flow chart:



Note that by the mesh refinement option **-z** also a prepass 0 is used.

In appendix A, a listing is given of a number of files. As example, the cell "sub3term" is used (see the "Space Substrate Res. Extraction User's Manual").

Appendix B gives a flow chart about how the *makesubcap* program can be tested. Note that the *space3d* "sub3d.internal" mode cannot be used to calculate 3D capacitances.

Note that the *makesubcap* program is default using the "sub3d.xxx" parameters. To use different parameters as the *makesubres* program, use a private "subcap3d.xxx" parameter. Only for parameter "sub3d.be_window" this is not allowed. For the Green module, who only is working with "sub3d.xxx" parameters, the "subcap3d.xxx" parameters must be converted to the "sub3d.xxx" parameters.

3. MAKESUBCAP OPTIONS

Command line:

```
makesubcap [-e def | -E file] [-p def | -P file] [-S param=value] [-Uhv] cell
```

Options explanation:

-e,-E	for specifying a technology file (see *space*)
-p,-P	for specifying a parameter file (see *space*)
-S	for specifying additional parameters (see *space*)
-U	for estimation, no green / schur calculation (see *space*)
-h	for getting help info
-v	for setting verbose mode (equal to **-i**)

4. MAKESUBCAP PARAMETERS

parameter	default	comment
debug.print_cap3d_init	off	prints debug information
debug.server_matrix	off	prints debug information
debug.print_matrix	off	prints debug information
debug.print_green	off	prints debug information
debug.print_spider	off	prints debug information
no_green	off	see option **-U**
no_schur	off	see option **-U**
param_verbose	off	prints read parameters
print_time	off	prints used time of program modules
sub3d.be_shape	4	only for pwc setable to 3
sub3d.be_window	--	must be specified
sub3d.edge_be_ratio	1	see documentation
sub3d.edge_be_split	0.5	see documentation
sub3d.edge_be_split_lw	4	see documentation
sub3d.max_be_area	-1	value > 0 needed
elim_sub_node	off	eliminate the substrate node "SUBSTR"
verbose	off	verbose mode, equal to **-v**

Note that "sub3d.be_shape" is always 3 for pwl.
The program does not support parameter "extract_window".

5. GREEN / SCHUR MODULE PARAMETERS

parameter	default	comment
collocation_mode	0	
check_collocation	0	
an_turnover	100	
accelerate_levin	off	
debug.print_green_init	off	
debug.print_green_terms	off	
debug.print_green_gterms	off	
use_adptv_intgrtn	off	
force_adptv_intgrtn	off	
force_mp_anaint	on	
force_mp_exint	off	
use_multipoles	on	
test_multipoles	off	
multipoles_oMaxMpOrder	oMaxMpOrder	
multipoles_cMaxMpOrder	cMaxMpOrder	
sub3d.use_mean_green_values	off	
sub3d.merge_images	on	
sub3d.mp_min_dist	2.0	
sub3d.mp_max_order	2	
sub3d.saw_dist	infinity	in microns
sub3d.edge_dist	0.0	in microns
sub3d.be_mode	0c	
sub3d.use_lowest_medium	on	
sub3d.use_old_images	off	
sub3d.neumann_simulation_ratio	100	
sub3d.point_green	infinity	
sub3d.collocation_green	0	
sub3d.asym_collocation_green	infinity	
sub3d.max_green_terms	500	
sub3d.y_stretch	1.0	
sub3d.green_eps	0.001	
sub3d.col_rel_eps	0.2	
lu	off	schur module
coefficient	off	schur module
print_coef	off	schur module, for debugging

Note that "sub3d.collocation_green" is "infinity" for FeModeGalerkin.

6. PREPASS PARAMETERS

For information about *space3d* prepass parameters, see the *makesubres* report 04-06.

7. THE COILGEN EXAMPLE

For information about this example, see the *makesubres* report 04-06.

8. MAKESUBCAP SOURCES

The *makesubcap* program is almost equal to the *makesubres* program.
For more information about these sources, see also the *makesubres* report 04-06.

Here, i shall only explain the changes made to the *makesubres* program sources to get the *makesubcap* program. See appendix C for the changes made to the source files.

The ZMAKE script has get a new zmake_target "makesubcap" to compile the sources with the *zmake* program. Note that the sources are compiled with the compile define MAKESUBCAP.

The gettech.c source file must now read technology files with minorPrNr 1 till 3. A technology file with minorPrNr == 3 can contain subcaplayers data. These data must be present, else the *makesubcap* program cannot work. The subcaplayers data is stored into the data struct of the sublayers data. Note that variable usedTechFile is set to an empty name string. This is needed, else the Green module tries to use the technology unigreen file.

In the sub3d.c source file is added function init_sub3d_params. This function calls init_param_sub3d for each "subcap3d.xxx" parameter which must set the "sub3d.xxx" parameter. Note that for "subcap3d.be_window" this is not done. And note that "sub3d.use_unigreen" is always set to "off". All calculated Green values are divided by Epsilon0 to get the correct capacitance values after Schur matrix inversion.

At last, the subcont.c source file is changed to write stream "subcap". Because capacitance values are written, the "g" before a value is changed into "c".

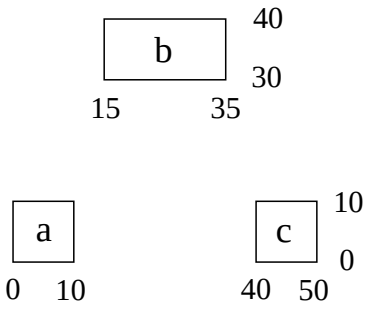
Of course, the *space3d* program sources are also changed. Now after running *makesubres* the *makesubcap* program must be running. And the *space3d* program must be prepared to read capacitance values from stream "subcap".

Note that the technology compiler *tecc* is also changed, to make it possible to specify the "subcaplayers" data.

9. APPENDICES

APPENDIX A -- List of files

layout cell "sub3term"
==========



cont_bln
==========

```
% dbcat -vs cont_bln sub3term
=> layout/sub3term/cont_bln
  xl:   xr:   yb:   yt:   ct:
    0    40     0     0     1 (BN 1)
    0    40    40    40  1025 (EN 1)
   60   140   120   120     1 (BN 1)
   60   140   160   160  1025 (EN 1)
  160   200     0     0     1 (BN 1)
  160   200    40    40  1025 (EN 1)
```

subcap
==========

```
c 1 1 xl 0 yb 0 c 1.167170e-16
nr_neigh 2
c 2 2 xl 60 yb 120 c 1.790923e-16
nc 1 c 1.303761e-17
nr_neigh 2
c 3 3 xl 160 yb 0 c 1.167170e-16
nc 2 c 1.303761e-17
nc 1 c 6.995180e-18
nr_neigh 2
```

NOTE: coordinates are in internal units (4 * database unit).

APPENDIX B -- Testing makesubcap

```
% cat elem.s
  ...
sublayers :
  # name       conductivity  top
    epi        6.7           0.0
    substrate  2.0e3        -7.0

subcaplayers :
  # name       conductivity  top
  c_epi        6.7           0.0
  c_substrate  2.0e3        -7.0

#EOF

% space3d -B -Eelem.t -Pparam.p sub3term -Sadd_sub_caps=1 -Ssub_rc_const=8.855e-12
% xsls sub3term > netlist1
% space3d -B -Eelem.t -Pparam.p sub3term -Sadd_sub_caps=2
% xsls sub3term > netlist2
% diff netlist1 netlist2
2c2
< /* Date: 31-Aug-05 15:56:09 GMT */
---
> /* Date: 31-Aug-05 15:56:11 GMT */

% cat netlist2
/* Generated by: xsls 2.37 30-Aug-2005 */
/* Date: 31-Aug-05 15:56:11 GMT */
/* Path: /u/52/52/work/simon/CACD/demo/demo/sub3term/projectname */

network sub3term (terminal c, b, a)
{
    res 1.265872M (c, a);
    res 679.1891k (c, b);
    cap 6.99518e-18 (c, a);
    cap 13.03761e-18 (c, b);
    cap 116.717e-18 (c, SUBSTR);
    res 75.86724k (c, SUBSTR);
    res 679.1891k (a, b);
    cap 13.03761e-18 (a, b);
    cap 116.717e-18 (a, SUBSTR);
    res 75.86724k (a, SUBSTR);
    cap 179.0923e-18 (b, SUBSTR);
    res 49.44378k (b, SUBSTR);
}
```

APPENDIX C -- Changes to the Source Files

```
Index: ZMAKE     (revision 1.2, date: 2005/08/22)
===================================================================
29a30,54
>
> zmake_target makesubcap {context} {
>
>     term::invoke context sheet remove compile:define:LICENSE
>
>     set sheet [term::new sheet]
>     term::invoke sheet add compile:define:MAKEDELA 1
>     term::invoke sheet add compile:define:MAKESUBCAP 1
>
>     set sources {
>         edge.c enumtile.c face.c gettech.c input.c
>         lump.c main.c mesh.c pqueue.c refine.c scan.c slant.c
>         spedge.c spider.c sppair.c sptile.c
>         strip.c sub3d.c subcont.c tile.c update.c
>     }
>
>     set libraries {
>         src::libddm::libddm
>         src::space::auxil::auxil
>         src::space::green::green
>         src::space::schur::schur
>     }
>
>     return [zmake_construct_program "makesubcap" $context $sources $libraries $sheet]
> }

Index: gettech.c (revision 1.4, date: 2005/08/22)
===================================================================
73c73
<     if (majorPrNr > MajorProtNr) {
---
>     if (majorPrNr > MajorProtNr || minorPrNr < 1 || minorPrNr > 3) {
142,156c142,144
<     /* Read dielectric information */
<     if ((diel_cnt = getnum (fp_tech)) > 0) {
<         for (i = 0; i < diel_cnt; i++) {
<             while ((c = getc (fp_tech)) != '\n') if (c == EOF) techReadError (13);
<         }
---
>     /* Skip dielectric information */
>     for (i = getnum (fp_tech); i > 0; --i) {
>         while ((c = getc (fp_tech)) != '\n') if (c == EOF) techReadError (13);
158a147
> #ifndef MAKESUBCAP
167a157,174
>     else say ("error: no sublayers found in technology data!"), die ();
> #else
>     /* Skip substrate res information */
>     for (i = getnum (fp_tech); i > 0; --i) {
>         while ((c = getc (fp_tech)) != '\n') if (c == EOF) techReadError (14);
>     }
>
>     /* Read substrate cap information */
>     if (minorPrNr > 2 && (substr_cnt = getnum (fp_tech)) > 0) {
>         substrs = NEW (substrate_t, substr_cnt);
>         for (i = 0; i < substr_cnt; i++) {
>             if (fscanf (fp_tech, "%s %le %le", substrs[i].name,
>                 &substrs[i].conduc, &substrs[i].top) != 3) techReadError (15);
```

```
>             while ((c = getc (fp_tech)) != '\n') if (c == EOF) techReadError (15);
>         }
>     }
>     else say ("error: no subcaplayers found in technology data!"), die ();
> #endif
192a200,202
> #ifdef MAKESUBCAP
>         usedTechFile = "";
> #else
193a204
> #endif

Index: sub3d.c   (revision 1.5, date: 2005/08/22)
===================================================================
111a112,152
> #ifdef MAKESUBCAP
> void init_param_sub3d (char *k, char *v)
> {
>     if (v || (v = paramLookupS (mprintf ("subcap3d.%s", k), v)))
>         paramSetOption2 (mprintf ("sub3d.%s", k), v);
> }
>
> void init_sub3d_params ()
> {
>     /* MAKESUBRES */
>  // init_param_sub3d ("be_window", NULL);
>     init_param_sub3d ("max_be_area", NULL);
>     init_param_sub3d ("edge_be_ratio", NULL);
>     init_param_sub3d ("edge_be_split", NULL);
>     init_param_sub3d ("edge_be_split_lw", NULL);
>     init_param_sub3d ("be_shape", NULL);
>
>     /* GREEN module */
>     init_param_sub3d ("use_unigreen", "off");
>  // init_param_sub3d ("use_unigreen_interpolation", NULL);
>  // init_param_sub3d ("test_unigreen", NULL);
>     init_param_sub3d ("use_mean_green_values", NULL);
>     init_param_sub3d ("merge_images", NULL);
>     init_param_sub3d ("mp_min_dist", NULL);
>     init_param_sub3d ("mp_max_order", NULL);
>     init_param_sub3d ("saw_dist", NULL);
>     init_param_sub3d ("edge_dist", NULL);
>     init_param_sub3d ("be_mode", NULL);
>     init_param_sub3d ("use_lowest_medium", NULL);
>     init_param_sub3d ("use_old_images", NULL);
>     init_param_sub3d ("neumann_simulation_ratio", NULL);
>     init_param_sub3d ("point_green", NULL);
>     init_param_sub3d ("collocation_green", NULL);
>     init_param_sub3d ("asym_collocation_green", NULL);
>     init_param_sub3d ("max_green_terms", NULL);
>     init_param_sub3d ("y_stretch", NULL);
>     init_param_sub3d ("green_eps", NULL);
>     init_param_sub3d ("col_rel_eps", NULL);
> }
> #endif
>
117a159,162
> #ifdef MAKESUBCAP
>     init_sub3d_params ();
> #endif
>
520a566,570
>
> #ifdef MAKESUBCAP
```

```
> #define Epsilon0 8.855e-12 /* Farad/meter */
>     val /= Epsilon0;
> #endif

Index: subcont.c (revision 1.2, date: 2005/08/22)
===================================================================
38a39,41
> #ifdef MAKESUBCAP
>     dmsSubres = dmOpenStream (lkey, "subcap", "w");
> #else
39a43
> #endif
172a177,180
> #ifdef MAKESUBCAP
>             fprintf (fp_subres, "c %d %d xl %ld yb %ld c %le\n",
>                 inSubTerm, sc -> group, sc -> xl, sc -> yb, n -> substrCon);
> #else
174a183
> #endif
180a190,192
> #ifdef MAKESUBCAP
>                     fprintf (fp_subres, "nc %d c %le\n", on -> sc_nr, con -> val);
> #else
181a194
> #endif
```