

Extraction of a Resistor

S. de Graaf

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

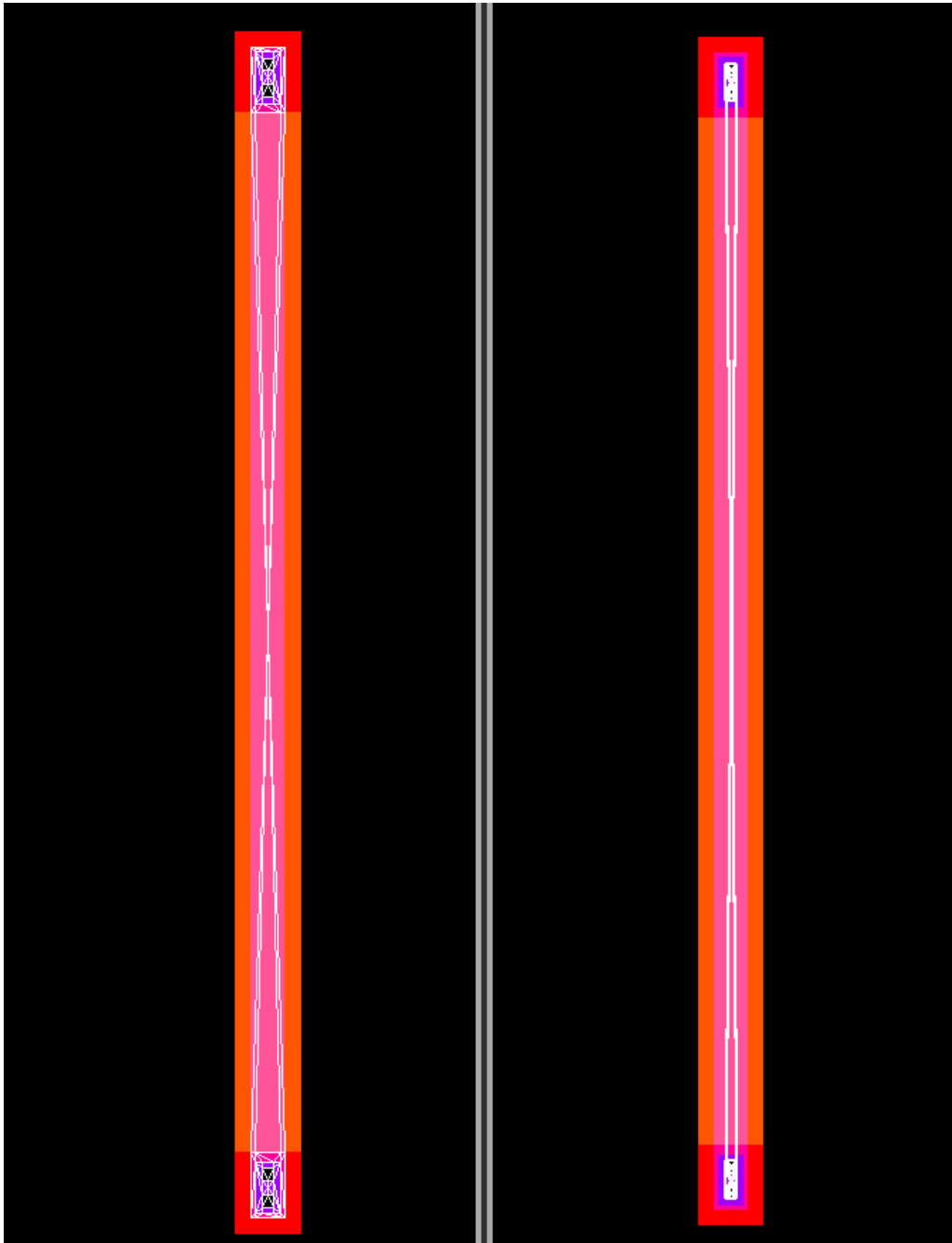
Report EWI-ENS 12-01
April 25, 2012

Copyright © 2012 by the author.

Last revision: May 7, 2012.

1. INTRODUCTION

In the figure below (on the left side) you see the finite element mesh of an extracted resistor. On the right side you see the resistor output mesh of the same extracted resistor. Note that each mesh line represents a resistor.

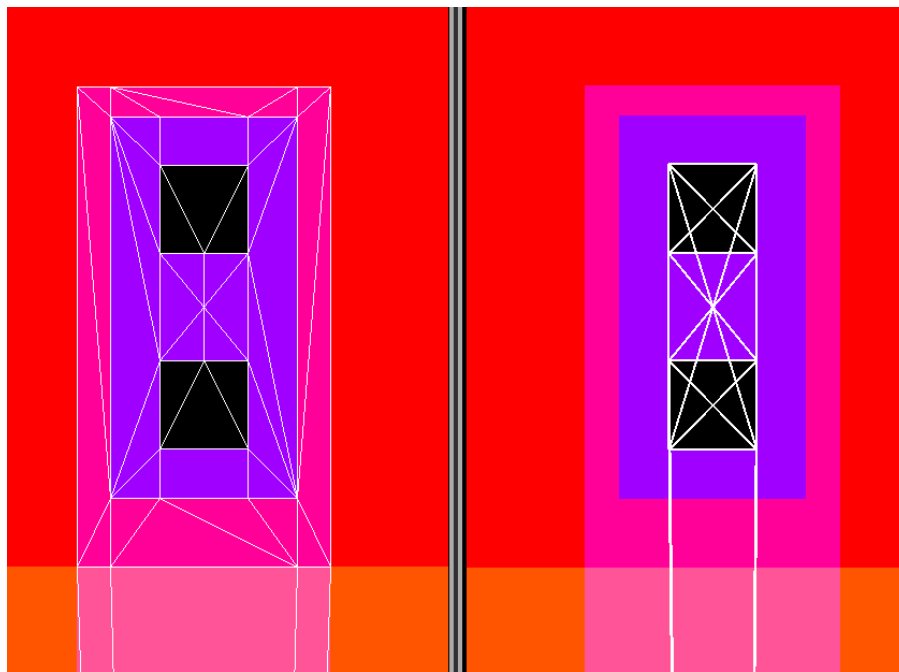


The first question, why is there a cross in the finite element mesh? Answer: There is an equi-potential line node because *space* parameter **equi_line_ratio** is set.

Note that the following technology file definitions are used and all resistances are extracted using *space* parameter **low_sheet_res=0**.

```
conductors:
  cond_M1   : M1           : M1       : 0.08   # ohm
conductors RH5:
  cond_RH5  : RH           : RH       : 5000   # ohm
contacts:
  cont_RH5  : CONT M1 RH   : M1 RH    : 48.6e-12 # ohm m^2
```

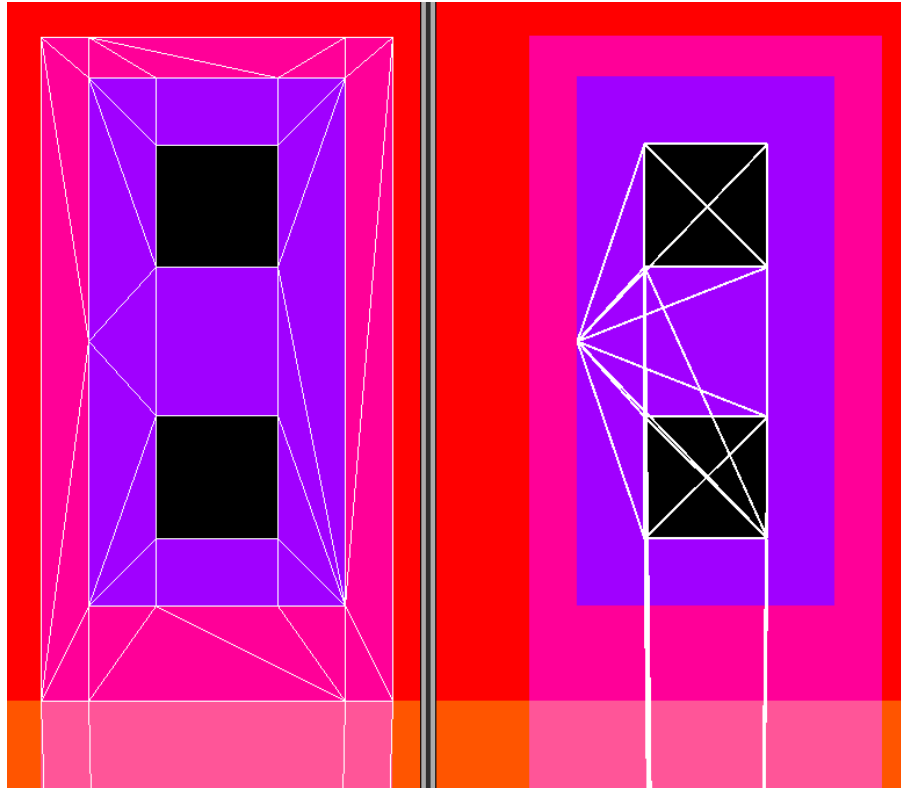
We shall zoom-in on the top of the figures and answer more questions.



The second question, what does the colors mean? Answer: The dark blue area is metal one layer (M1) and the purple area is the high-res layer (RH). The black holes are the contacts (using layer CONT) between the M1 and the RH layer. The orange part (inside P) is the N layer, the red part is the P layer.

The third question, why is there a cross in the mesh between the contacts? Answer: There is a terminal box defined in M1 with the size of the dark blue area. By default the center x,y position is used for this terminal and a terminal is always laying on a vertical tile edge position. Therefor the tile between the two contact areas is split in two parts. If you don't want to use the center x positions, but the left x position, use *space* parameter **term_use_center=off**. Or use point terminals in place of box terminals.

Using `term_use_center=off` gives the following resistance mesh:



A more important question, why are there so many output resistors? Answer: Indeed the network is not good reduced and many nodes are left in the network. This happens because many nodes have two different types of resistors attached. These nodes can't be eliminated.

There are different tricks possible to reduce the network. A very simple solution, make the contact of the same resistor type:

```
conductors:
  cond_M1 : M1      : M1      : 0.08 # ohm
conductors RH5:
  cond_RH5 : RH      : RH      : 5000 # ohm
contacts RH5:
  cont_RH5 : CONT M1 RH : M1 RH : 48.6e-12 # ohm m^2
```

This gives the following *spice* circuit:

```
* circuit test_RH5 M P
r1 P M_2 76.77365k RH5 $ x1=0.425 y1=39.6 x2=0.65 y2=19.5
r2 M M_2 76.77365k RH5 $ x1=0.175 y1=-1.325 x2=0.65 y2=19.5
* end test_RH5
```

In the *spice* circuit you see that the equi-potential line node is not eliminated. This is unexpected and comes maybe by the order of evaluation of the nodes by the articulation reduction heuristic. It evaluates each node only ones.

If you put off parameter **equi_line_area**, you get the following circuit:

```
* circuit test_RH5 M P
r1 P M 153.5473k RH5 $ x1=0.425 y1=39.6 x2=0.175 y2=-1.325
* end test_RH5
```

You can also put off parameter **equi_line_ratio** by setting it to a zero or negative value. In that case no equi-potential line nodes are created / used.

Note that the above netlisting with the program *xspice* can only be done, if a RH5 model line is added to the "xspicerc" file, i.e.:

```
include_library spice3f3.lib
...
model RH5 RH5 r ( )
```

To get a ".model" statement also in the netlisting, you must add a correct "model" into the **include_library** file. For example:

```
model RH5 r ( ...
               ... )
```

When you want to know what happens during the extraction, you can put on the *space* parameters **debug.ready_group** and **debug.ready_group2** (give these parameters a value of 1). You get the following ready group debug output:

```
ready_grp: nodes=85 (delayed=66 term=2 keep=16 area=1) adjgrps=0
              total(node=86 cap=0 res=210)
ready_grp: n_cnt=19 eliminateGroup: delay_cnt=66 degree=3
ready_grp: n_cnt=19 reducArtDeg: total(cap=0 res=144)
ready_grp: n_cnt=19 reducMinRes: total(cap=0 res=144)
ready_grp: n_cnt=17 reducSepRes: total(cap=0 res=128)
ready_grp: n_cnt=3 reducNegRes: total(cap=0 res=2)
ready_grp: n_cnt=3 reducParRes: total(cap=0 res=2)
ready_grp: n_cnt=3 do outGroup: total(cap=0 res=2)
```

From the above lines you can read that the ready group has initial 85 nodes, where of 66 nodes are delayed and can be eliminated. The other 19 nodes can be classified as 2 terminal nodes, 1 line area node and 16 nodes with the keep flag set. The keep flag is set because these nodes have different resistor types connected and can not be eliminated (each contact hole has 4 of these nodes).

Further more you see that the reducMinRes heuristic can eliminate 2 nodes, because these nodes have a small resistor < min_res (= 100 ohm) connected. Note that these are the two terminal nodes and a neighbor node inherits the terminal name. Note that nodes with keep flag are not evaluated by the reducMinRes heuristic.

Further more you see that the `reducSepRes` heuristic eliminates 14 nodes, because these nodes have a small resistor $< \text{min_sep_res}$ ($= 10 \text{ ohm}$) connected. Note that this heuristic can also eliminate nodes with different types of resistors connected (with a set keep flag). This is possible, because this heuristic does a node join instead of a gaussian elimination.

The user of the *space* extractor may have the following general question: how do i know which parameter values are used? The answer is, that you can ask the extractor to output these parameter settings. To do this set parameter **param_verbose** to "on" and you get the following output:

```
parameter(b) param_verbose = '' (set to 'on')
parameter(b) verbose = 'off' (default)
parameter(b) param_unused = 'off' (default)
parameter(b) simple_sub_extraction = 'off' (default)
parameter(i) progress_timer = '0' (default = '0')
parameter(b) component_coordinates = 'on' (set)
parameter(b) backannotation = 'off' (default)
parameter(b) heuristics = 'on' (default)
parameter(b) network_reduction = 'on' (default)
parameter(b) print_time = 'off' (default)
parameter(b) flat_extraction = 'off' (default)
parameter(b) expand_connectivity = 'off' (default)
parameter(b) no_parasitics = 'off' (default)
parameter(r) low_contact_res = '1e-13' (default = '1e-13')
parameter(r) max_res = '1e50' (default = '1e+50')
parameter(r) low_sheet_res = '0' (set to '0')
parameter(b) add_ds_terms = 'off' (default)
parameter(b) omit_ds_caps = 'off' (default)
parameter(b) omit_incomplete_tors = 'off' (default)
parameter(b) capitalize = 'off' (default)
parameter(r) equi_line_ratio = '1.0' (set to '1')
parameter(b) equi_line_area = 'on' (default)
parameter(i) min_art_degree = '3' (set to '3')
parameter(i) min_degree = '4' (set to '4')
parameter(b) no_neg_res = 'on' (set)
parameter(r) min_res = '100' (set to '100')
parameter(b) lowest_min_res = 'off' (default)
parameter(r) min_sep_res = '10' (set to '10')
parameter(r) max_par_res = '25' (set to '25')
parameter(b) delete_dangling = 'off' (default)
parameter(r) lat_base_width = '3' (default = '3')
parameter(b) merge_par_bjts = 'off' (default)
parameter(b) term_use_center = 'off' (set)
parameter(i) max_message_cnt = '-1' (default = '-1')
parameter(b) debug.gettech = 'off' (default)
parameter(s) keep_nodes = '' (default)
...
parameter(s) pos_name_prefix = '' (default)
parameter(b) duplicate_list = 'off' (default)
```