Selective
Resistance
Extraction

S. de Graaf

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

1. INTRODUCTION

The *space* layout to circuit extractor extracts normally resistances for all high resistive conductors in the layout.

Parameter "low_sheet_res", default 1 ohm, controls which conductors are low resistive.

When we look into the technology file of the "scmos_n" example process, we can find the following resistivities for the conductors:

```
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
```

For example, when we do a flat resistance extraction of the "switchbox4" demo layout, we only extract resistances for the (high res) masks "cpg" and "caa". Note that "caa" represents the active (drain / source) areas of the "nenh" and "penh" fets. And that "cpg" represents the poly gate and interconnect areas.

Note that, besides the above ones, there are also resistances extracted for the contacts (vias). The following contacts are defined in "scmos_n" technology file:

```
contacts :
 # name    : condition            : lay1 lay2 : resistivity (ohm um^2)
   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
```

Note that parameter "low_contact_res", default 0.1 ohm um^2, controls the usage. Thus, the above contact resistivities are all extracted. Note that for the above contacts "unit c_resistance 1e-12" is specified in the technology file.

The following command can be used to run the extraction:

```
% space -Fr switchbox4
```

The following command can be used to view the resulting circuit in SLS format:

```
% xsls switchbox4
```

Note that normally the resulting resistance network is reduced. For example, when parameter "min_res" is set to 100 ohm (default 0), then the resulting resistors smaller than 100 ohm are removed!

2. SELECTIVE EXTRACT

When you don't want to extract all high resistive conductors in the layout, you can select some of these high res conductors with option **-k**. Which conductors you want to select must be specified in the "sel_con" file. For example:

```
=sel_0_0
=out_0
 23 9 cpg
 13 18 cmf
```

You see in the above file two possibilities to specify a conductor. One, you can specify a terminal or label of the top layout cell. This specification must begin with a '=' character on the line. Or, second, you can specify coordinates and a conductor mask name on the line. In the above specification, name "sel_0_0" and "out_0" are two terminals in the layout of "switchbox4". Note that point (23,9) on mask "cpg" is connected with terminal "sel_0_0". Thus, this specification was not needed. Also the specification of point (13,18) on mask "cmf" was not needed. Because in the normal case, this conductor is already low sheet res. Note that the file "sel_con" needs to be placed in the project directory or in the "scmos_n" process directory.

The following command can be used to run the selective extraction:

```
% space -Fk switchbox4
```

Note that you don't find resistors for "out_0" in the resulting network. This happens, because all resistors are less than 100 ohm and are reduced. Specify another value for parameter "min_res" to see these resistors. For example, give the following command:

```
% space -Fk -Smin_res=10 switchbox4
```

An other way around:
When you don't want to extract all high resistive conductors in the layout, you can select some of the high res conductors with option **-j**, which you want to exclude from the resistance extraction.

Give the following command to exclude the specified "sel_con" conductors:

```
% space -Fj -Smin_res=10 switchbox4
```

Note that a conductor path with contacts via substrate are not selected as one conductor path. Note that, when substrate resistance extraction is chosen, there is no possibility to selective extract these substrate resistors.

Note that you may use the '*' character as wildcard at the begin or end of terninal and/or label names to specify a set of conductors in the "sel_con" file.

3. SELECTIVE RES EXAMPLES

```
% space -Fj -v -Smin_res=10 switchbox4
Version 5.4.6, compiled on Tue, 31 Dec 2013 12:10:08
See http://www.space.tudelft.nl
parameter file: $ICDPATH/share/lib/process/scmos_n/space.def.p
technology file: $ICDPATH/share/lib/process/scmos_n/space.def.t
prepassing switchbox4 for selective extraction
2 (out of 154) interconnects selected for resistance extraction
extracting switchbox4
space: warning: no node join of subnodes with different conductor-type
    for conductor mask 'caa' at position (19, 28).

extraction statistics for layout switchbox4:
    capacitances       : 0
    resistances        : 509
    nodes              : 504
    mos transistors    : 144
    bipolar vertical   : 0
    bipolar lateral    : 0
    substrate nodes    : 4

overall resource utilization:
    memory allocation  : 1.556 Mbyte
    user time          :          0.0
    system time        :          0.0
    real time          :          0.0  75.0%

space: --- Finished ---
```

Compare this result with a full res extraction:

```
% space -Fr -v -Smin_res=10 switchbox4
Version 5.4.6, compiled on Tue, 31 Dec 2013 12:10:08
    ...
extraction statistics for layout switchbox4:
    capacitances       : 0
    resistances        : 524
    nodes              : 517
    mos transistors    : 144
    bipolar vertical   : 0
    bipolar lateral    : 0
    substrate nodes    : 4

overall resource utilization:
    memory allocation  : 1.152 Mbyte
    user time          :          0.0
    system time        :          0.0
    real time          :          0.0 100%

space: --- Finished ---
```

```
% space -Fk -v -Smin_res=10 switchbox4
Version 5.4.6, compiled on Tue, 31 Dec 2013 12:10:08
See http://www.space.tudelft.nl
parameter file: $ICDPATH/share/lib/process/scmos_n/space.def.p
technology file: $ICDPATH/share/lib/process/scmos_n/space.def.t
prepassing switchbox4 for selective extraction
2 (out of 154) interconnects selected for resistance extraction
extracting switchbox4
space: warning: no node join of subnodes with different conductor-type
    for conductor mask 'caa' at position (19, 28).

extraction statistics for layout switchbox4:
    capacitances       : 0
    resistances        : 15
    nodes              : 99
    mos transistors    : 144
    bipolar vertical   : 0
    bipolar lateral    : 0
    substrate nodes    : 0

overall resource utilization:
    memory allocation  : 0.173 Mbyte
    user time          :          0.0
    system time        :          0.0
    real time          :          0.0 100%

space: --- Finished ---
```

You can also request debug output:

```
% space -Fk -Smin_res=10 -Sdebug.selective_res switchbox4
selected high-res tiles are:
 tile_nr  con    xl     bl
     603    2     22     8
     607    2     22    37
     643    0     23     9
    1110    2     41    46
    1155    2     43     8
    1156    2     43    33
    2042    2     70    46
    2087    2     72     8
    2088    2     72    33
    4734    3    154    41
    4769    0    155    42
    5061    3    165    41
    5403    3    175    16
    5460    3    177    41
    5535    0    179    28
 netname 'sel_0_0' used 1 time
 netname 'out_0' used 1 time
```

The conductor numbers (con) can be found in the 3rd column of the header of the "space.def.t" file. Thus: 2=cpg, 0=cmf and 3=caa.

4. IMPLEMENTATION DETAILS

The selective resistance extraction method uses a prepass "pp2EnumPair" to select the node groups. Because we don't know, which groups gonna be selected, we start saving important vertical edges of high res conductors of all the groups. All these edges are placed into the "selResTiles" list by function "newTileRef". Each node group gets only one "contRef_t" data struct.

Note that a high res contact (via) between two conductors must also try to place two "contRef_t" structs, one for each conductor group. Because the conductors can be low res. This must be done in "pp2EnumPair", because in the extract pass the contact conductor groups are connected when the tile becomes ready in function "resEnumTile". But group select and "prick" setting is done in function "resEnumPair".

On the end a group can have more than one data struct, because groups are joined together. The "selResTiles" list is always sorted on tile number (we don't need to sort). Each new tile gets a unique tile number in function "enumpair". Both the prepass and the extract pass must have exact the same number of tiles. All tiles are cleared by function "clearTile". When the last subnode of a node group becomes ready, then function "subnodeDel2" is called. The node group becomes ready (because each group has only one node) and we look to the group "prick" flag. When the group is selected (by mask coordinate or a name), the group "prick" flag was set. In that case we don't delete the "contRef_t" data structs from the "selResTiles" list. When a group is not selected, we delete all the group "contRef_t" data structs from the list. Thus, on the end of prepass2, we have only left over a list of selected data structs. In the extract pass we use this list of selected data structs again to select the node groups and to set of these groups the "prick" flag again. How to use the selected groups, depends on the given option **-k** or **-j**.

When in function "resEnumPair" for a new tile the tile number matches with the data struct(s) in the "selResTiles" list, then these data struct(s) are removed from the "selResTiles" list. We remember each found conductor struct in the subnode list of the new tile, by setting the "highres" flag of the conductor subnode. Note that the structs are not sorted on conductor number. We call function "connectPoints" and this function shall set the group "prick" flag. Note that the subnode "highres" flag is also used to flag real high res conductors. Thus, it must be reset in "connectPoints", when the conductor is not high res. The special "highres" value 2 is used to indicate, that it was high res, but that the conductor was put off (deselected or because of a res filter). Note that only "highres" value 1 conductors are really high res. Thus, we need this "highres" flag in each used subnode of a tile. We can't use a tile known bitmask to do this, because there can be too many conductors. However, the tile structs become bigger in size, because the subnodes are all part of the tile. And when by cap3d mode a bandwidth is used, the tiles live longer than normal. Note that the live time ends for a tile, when function "clearTile" is called.