

# **SCAN EDGE BUNDLE PROBLEM IN 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 04-03  
June 8, 2004

Copyright © 2004 by the author.

Last revision: June 10, 2004.

## 1. INTRODUCTION

By the substrate resistance extraction of the "infineon/coilgen" layout with the *space* program, i get the following problem:

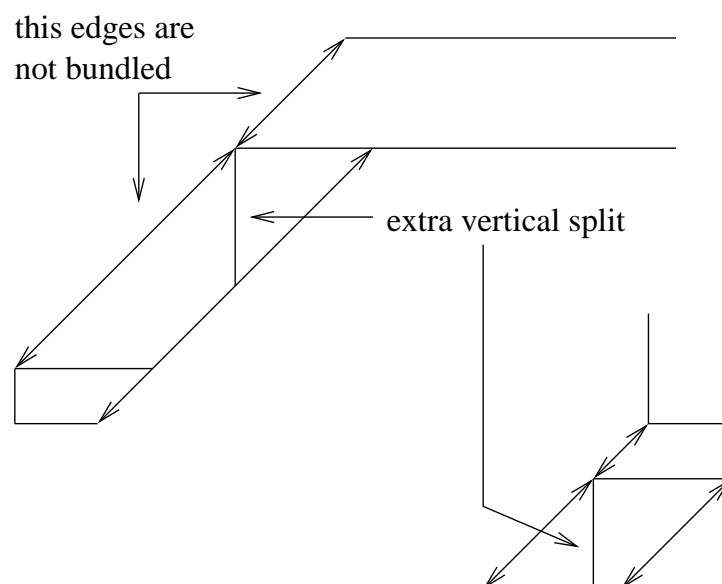
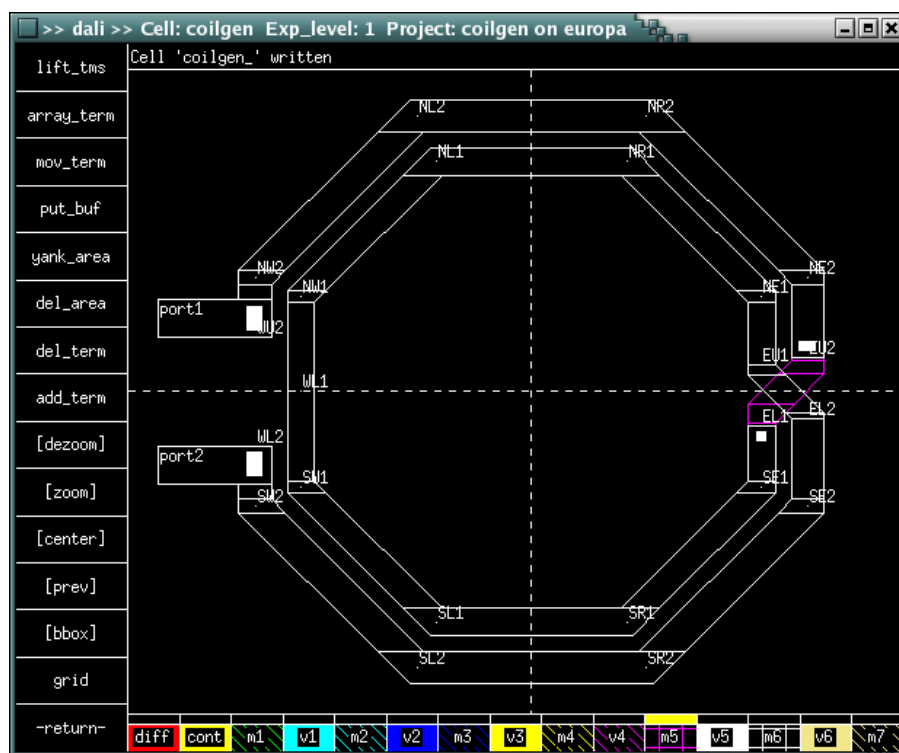
For testing, i changed something to the layout with the *dali* program, and after that the extraction results were completely different. What happens? The *dali* program uses a horizontal scan technique and saves the layout elements, like the internal vertical data structure used. This layout database storage is different than the initial storage used by the *cgi* program, that a gds2 file converts to the internal database format.

Thus, the *space* extraction result is depending on the method of internal layout storage used. This is, of course, not an allowed feature.

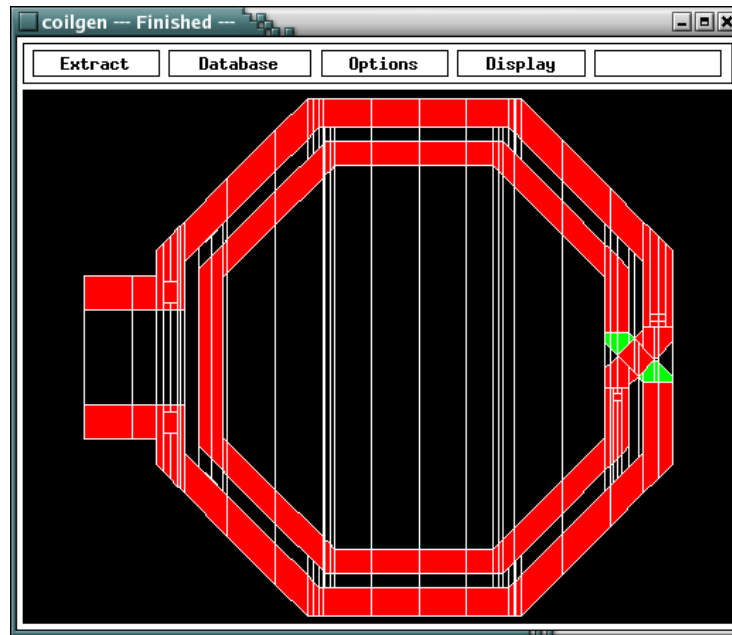
I thought that this could not happen, because of the layout expansion tools *makeboxl* and *makegln* used. I thought that *makegln* always produces the same general line segment data independent from the internal layout storage used. However, this seems not to be the case. Yes, the *makegln* program shall not always bundle touching line segments and does not make the line segments as long as possible. And it is not a good idea, to fix the problem in the *makegln* program. Because touching edges (line segments) can also happen by reading two edges of different masks.

Thus, to solve the *space* problem, we need to change the input scanning technique of the *space* program. Thus, the *space* program must also bundle touching edges. However, not for the special substrate prepass, where there is read another data format by the scanner and touching edges have a special meaning. Note, that i already had the idea before, that it was better to bundle also touching edges. But i could not make it hard, why it was a better choice. And after implementing the special substrate prepass, it seems not more to be a good idea.

### 1.1 Layout example as seen by dali

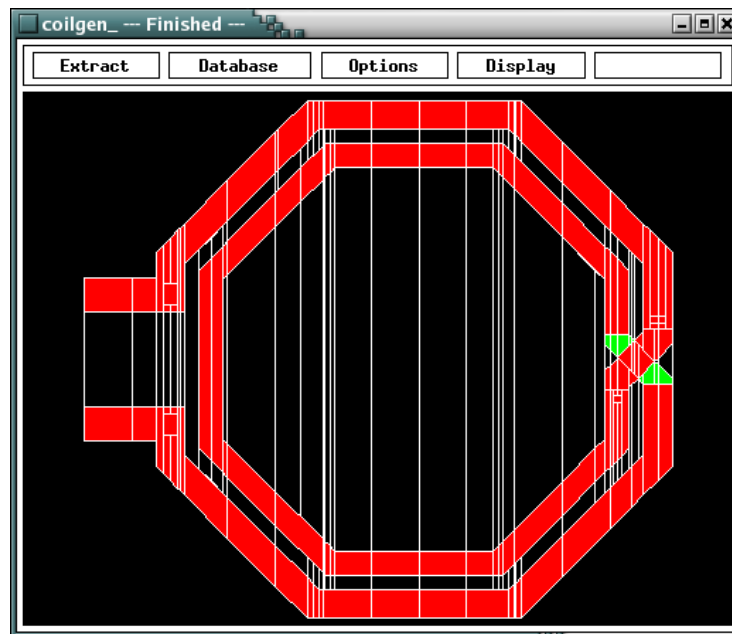


### 1.2 Layout extraction before using dali



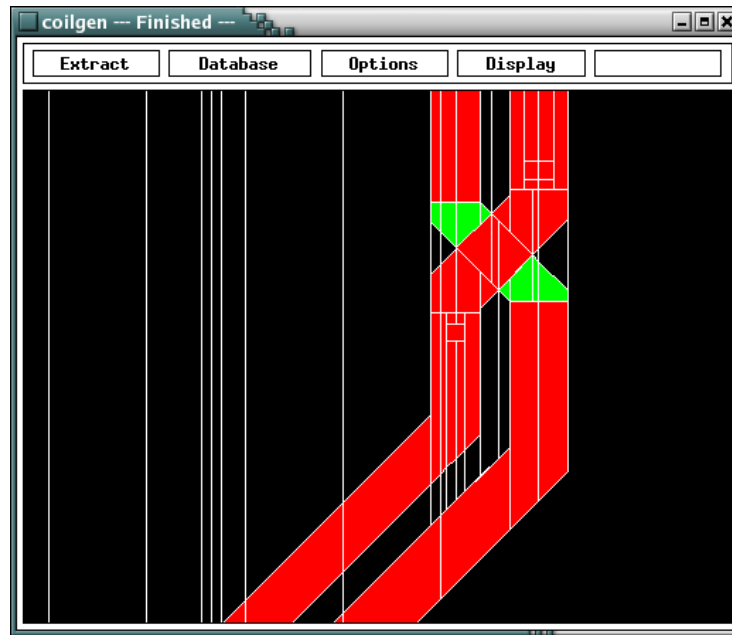
126 substrate contact tiles

### 1.3 Layout extraction after using dali

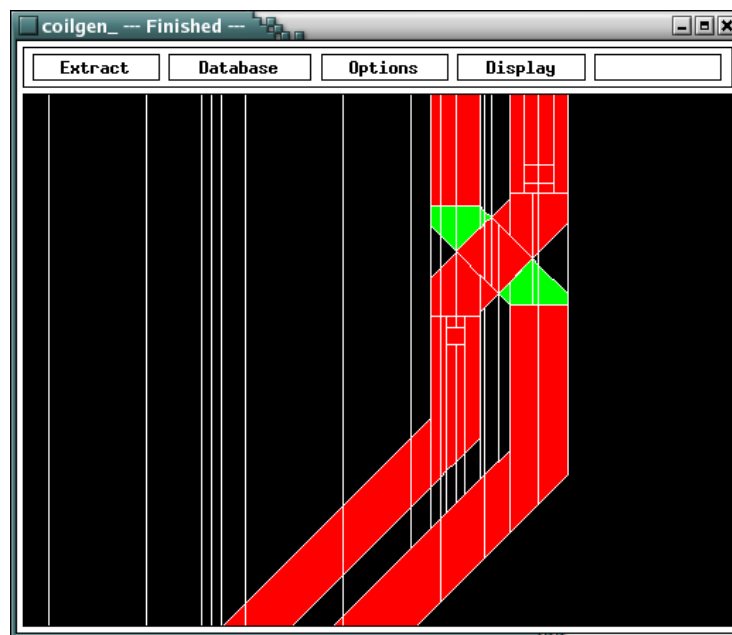


132 substrate contact tiles

#### 1.4 Layout extraction before using dali (detail)



#### 1.5 Layout extraction after using dali (detail)



## 2. APPENDICES

### APPENDIX A -- File scan/scan.c version 4.55

```

/* Comment added NvdM, 891213
 * the comparison with e1 -> xr is (probably!!!) necessary because
 * the ordering in the stateruler can become wrong
 * in cases where an edge under +45 straddles the scanline
 * in say, (x,y) and an edge under -45 ends in (x,y).
 * This can, however give other problems because horizontal
 * edges ending and starting at (x,y) will not be bundled.
 */
#define equalAtX(e1, e2) \
    (e2 -> xl == thisX && e2 -> yl == thisY && \
    compareSlope (e1, ==, e2) && e1 -> xr > thisX)

void scan ()
{
    ...

    while (thisX < INF) {
        nextX = INF;
        ...

        while (edge -> yr < INF || newEdge -> xl == thisX || termX == thisX) {

            thisY = Y (edge, thisX);

            ...
            if (edge -> xc == thisX && edge -> bundle) unbundle (edge);

            if (smallerAtX (edge, newEdge)) {
                ...
            }

            termSplit = 0;

            if (equalAtX (edge, newEdge)) {
                do {
                    if (optOnlyPrePassB1 && !(newEdge -> cc & 0xc00)) termSplit = 1;
                    bundle (newEdge, edge);
                    newEdge = fetchEdge ();
                } while (equalAtX (edge, newEdge));
                ...
            }
            ...
            ...
        }
        ...
        tileAdvanceScan (edge);
        ...
        thisX = nextX;
    }

    tileStopScan (head);
    ...
}

```

**APPENDIX B -- Diffs between file scan.c 4.55 and 4.56**

```

82c82
<      compareSlope (e1, ==, e2) && e1 -> xr > thisX)
---
>      compareSlope (e1, ==, e2) && (!optOnlyPrePassB1 || e1 -> xr > thisX))

```

**APPENDIX C -- Byte compare of the layout files**

file	coilgen	coilgen_ok	coilgen_bad
info	1002	1002	1002
annotations	6	6	6
box	48	264	264
nor	738	2462	2462
term	410	399	399
m5_gln	360	416	416
m6_gln	360	406	406
v5_gln	72	72	72
cont_bln	3569	3569	3737
cont_pos	45	45	45
subres	29337	29337	32159

Note that with *dali* the terminal "WU1" is deleted from cell "coilgen".  
The used *space* command and parameter file are:

```

% space3d -Br -P params coilgen
% cat params

sub3d.max_be_area      10
sub3d.edge_be_ratio    1
sub3d.be_window        10

sep_sub_term           on
sub_term_distr_m5      on
sub_term_distr_m6      on
elim_sub_term_node     off

low_sheet_res          0.0001
max_par_res            1
min_art_degree         3
min_degree             1
max_obtuse             90

```

**APPENDIX D -- Layout files compare: m5\_gln**

```
7c7
< -216000 -89480 89480 216000
---
> -216000 -113480 89480 192000
11c11
< -180000 -74560 74560 180000
---
> -180000 -94560 74560 160000
13a14,15
> -113480 -89480 192000 216000
> -94560 -74560 160000 180000
22c24
< 66280 160000 -160000 -66280
---
> 66280 151720 -160000 -74560
26c28
< 79520 192000 -192000 -79520
---
> 79520 182040 -192000 -89480
30c32,33
< 160000 192000 -9040 22960
---
> 151720 160000 -74560 -66280
> 160000 182080 -9040 13040
32a36,37
> 182040 192000 -89480 -79520
> 182080 192000 13040 22960
```

**APPENDIX E -- Layout files compare: m6\_gln**

```
7c7
< -216000 -89480 89480 216000
---
> -216000 -113480 89480 192000
11c11
< -180000 -74560 74560 180000
---
> -180000 -94560 74560 160000
13a14,15
> -113480 -89480 192000 216000
> -94560 -74560 160000 180000
22c24
< 66280 160000 -160000 -66280
---
> 66280 151720 -160000 -74560
26c28
< 79520 192000 -192000 -79520
---
> 79520 182040 -192000 -89480
29a32
> 151720 160000 -74560 -66280
32a36
> 182040 192000 -89480 -79520
```



**APPENDIX F -- SLS files compare**

```

5,7c5,7
< 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)
---
> network coilgen_ok (terminal EL1, NE1, SR2, SL2, EU2, WL1, NR2, NL2, NE2, WU2,
>                     WL2, EU1, SE2, EL2, SW2, SR1, NW2, NW1, SW1, SE1, SL1, NR1,
>                     port2, port1, NL1)
9d8
< net {WU1, WL1};
1208,1209c1207,1208
< res 8.86235m (NW1, WU1);
< res 8.86235m (WU1, SW1);
---
> res 8.86235m (NW1, WL1);
> res 8.86235m (WL1, SW1);

```

Note that *dali* has also reversed the order of the terminals!

**APPENDIX G -- SLS files compare**

```

5,7c5,7
< 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)
---
> network coilgen_bad (terminal EL1, NE1, SR2, SL2, EU2, WL1, NR2, NL2, NE2, WU2,
>                     WL2, EU1, SE2, EL2, SW2, SR1, NW2, NW1, SW1, SE1, SL1, NR1,
>                     port2, port1, NL1)
9d8
< net {WU1, WL1};
222,224c221,225
< res 32.03603k (28, 36);
< res 49.94359k (28, 30);
< res 21.2964k (28, SUBSTR);
---
> res 324.9656k (28, 37);
> res 2.95531M (28, 38);
> res 35.51792k (28, 36);
> res 50.45982k (28, 30);
> res 21.49582k (28, SUBSTR);

...
...

> res 155.8914k (127, 130);
> res 1.755393M (127, 128);
> res 15.44246k (127, 131);
> res 146.7762k (127, 129);
> res 36.94201k (127, 132);
> res 51.06337k (127, SUBSTR);
> res 412.3315k (128, 130);
> res 238.5612k (128, 129);
> res 1.063029M (128, 131);
> res 739.5815k (128, SUBSTR);
> res 936.5488k (129, 130);
> res 406.6347k (129, 131);
> res 794.9208k (129, SUBSTR);

```

---

```

>     res 17.70576k (130, 131);
>     res 276.6507k (130, 132);
>     res 63.39213k (130, SUBSTR);
>     res 201.2622k (131, 132);
>     res 76.90137k (131, SUBSTR);
>     res 25.52495k (132, SUBSTR);
1202c1319
<     res 259.3106m (SL2, SR2);
---
>     res 259.3105m (SL2, SR2);
1204,1224c1321,1341
<     res 256.9423m (SR2, SE2);
<     res 1.975785 (EL1, EU2);
<     res 65.20366m (EL1, SE1);
<     res 299.3041m (NW1, NL1);
<     res 8.86235m (NW1, WU1);
<     res 8.86235m (WU1, SW1);
<     res 299.3041m (SW1, SL1);
<     res 263.668m (SL1, SR1);
<     res 190.576m (SR1, SE1);
<     res 181.1799m (port1, NW2);
<     res 208.8902m (port1, WU2);
<     res 59.42574m (WU2, NW2);
<     res 206.8129m (NW2, NL2);
<     res 259.3741m (NL2, NR2);
<     res 196.5181m (NR2, NE2);
<     res 76.90015m (NE2, EU2);
<     res 263.7301m (NL1, NR1);
<     res 178.3569m (NR1, NE1);
<     res 93.04827m (EU1, EL2);
<     res 56.56543m (EU1, NE1);
<     res 99.7248m (SE2, EL2);
---
>     res 256.9156m (SR2, SE2);
>     res 1.928016 (EL1, EU2);
>     res 69.0132m (EL1, SE1);
>     res 299.6124m (NW1, NL1);
>     res 8.862246m (NW1, WL1);
>     res 8.862367m (WL1, SW1);
>     res 299.304m (SW1, SL1);
>     res 263.6346m (SL1, SR1);
>     res 198.0689m (SR1, SE1);
>     res 181.1801m (port1, NW2);
>     res 208.89m (port1, WU2);
>     res 59.42614m (WU2, NW2);
>     res 206.8358m (NW2, NL2);
>     res 259.3658m (NL2, NR2);
>     res 196.5183m (NR2, NE2);
>     res 76.89902m (EU2, NE2);
>     res 263.6521m (NL1, NR1);
>     res 178.3565m (NR1, NE1);
>     res 93.05616m (EU1, EL2);
>     res 56.56493m (EU1, NE1);
>     res 99.74066m (SE2, EL2);

```