

HOW THE NEW XSPACE PROGRAM WORKS

S. de Graaf

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

Report EWI-ENS 05-02
April 20, 2005

Copyright © 2004 by the author.

Last revision: May 10, 2005

1. INTRODUCTION

This document describes how the new *Xspace* program works in combination with the *space3d* program.

The program *Xspace* is now separated from the *space3d* program. Before the separation the programs were the same executable. The *space3d* program was a symbolic link to the *Xspace* program.

Now the programs are separated, but both programs need to share some sources of each other.

The *space3d* program uses the display drawing parameters to write to a display file. And the *space3d* program uses the drawing routines to write to a display file. But *space3d* does not need to be loaded with the X11 libraries anymore.

The *Xspace* program uses the same main routine for the command line options as *space3d*. And the *Xspace* program must init the design project, read the space technology file and parameter file. But *Xspace* does not need to be loaded with the space extractor module libraries. But it still uses some modules of the "ddm" and "auxil" libraries.

2. XSPACE MODULES

The following *zmake* target is used:

```
zmake_target Xspace {context} {

    term::invoke context sheet add compile:define:TOOLNAME \"Xspace\"
    term::invoke context sheet add compile:define:CONFIG_XSPACE 1
    term::invoke context sheet add compile:define:SCENE 1

    set sources {
        src/space/space/date.c
        src/space/space/main.c
        src/space/extract/gettech.cc
        src/space/scan/getparam.c
        src/space/scan/sp_main.c

        src/space/X11/interact.c
        src/space/X11/draw.c
        src/space/X11/exposel.c
        src/space/X11/menu.c
        src/space/X11/rgb.c
        src/space/X11/robot.c
        src/space/X11/x2d.c
        src/space/X11/x3d.c
    }

    set libraries {
        src::generic::libs::libstd::libstd
        src::libddm::libddm
        src::space::auxil::auxil
        src::space::xmenu::libmenu
        src::formats::libcirfmt
        src::formats::liblayfmt
    }
}
```

As can be seen, the compilation is started with define CONFIG_XSPACE. Only some source files of *space3d* are used by the *Xspace* program. Of coarse, the *sp_main* function in file "src/space/scan/sp_main.c". But, besides that, only the parameter file and technology file read modules.

3. SPACE3D MODULES

The following *zmake* target is used:

```
zmake_target space3d {context} {

    term::invoke context sheet add compile:define:TOOLNAME \"space3d\"
    term::invoke context sheet add compile:define:CONFIG_SPACE3D 1
    term::invoke context sheet add compile:define:SCENE 1

    set sources {
        src/space/space/date.c
        src/space/space/main.c
        src/space/X11/draw.c
        src/space/X11/interact.c
    }

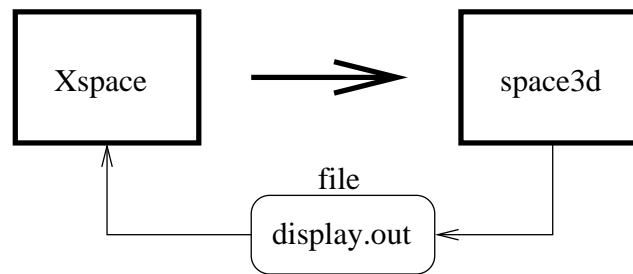
    set libraries {
        src::generic::libs::libstd::libstd
        src::libddm::libddm
        src::space::auxil::auxil
        src::space::scan::scan
        src::space::extract::extract
        src::space::lump::lump
        src::space::bipolar::bipolar
        src::space::substr::substr

        src::space::spider::spider
        src::space::green::green
        src::space::schur::schur
        src::formats::libcirfmt
        src::formats::liblayfmt
        src::space::scene::scene
    }
}
```

As can be seen, the compilation is started with define CONFIG_SPACE3D. Only two source files from "src/space/X11" are used. Source file "draw.c" for the drawing functions, which writes to the "display.out" file. And source file "interact.c" for the init of the used drawing parameters.

4. HOW XSPACE WORKS

The *Xspace* program runs the *space3d* program in the background for each extraction which must be done. The *space3d* program writes the wanted drawing elements (like edges and tiles) to the "display.out" file, and the *Xspace* program reads this "display.out" file and displays these drawing elements. Thus, the "display.out" file is the interface between this two programs.



Because there can be only one "display.out" file in a project directory, you (and other users) must not start two extractions with *Xspace* at the same time in the same project directory.

When you start *Xspace* with option **-X**, the program tries directly to extract and display the given cell name. This is only possible, if you don't forget to specify the cell name on the command line. When option **-X** is used, the program shall default also don't use menu's. Thus, it is not possible to choice a cell name out of the Database menu. In this situation, you can only give commands with the hot-keys. And for example redisplay an existing "display.out" file again.

To be compatible with old *space3d* invocations, the *Xspace* program starts *space3d* in a special way, thus it shall write a "display.out" file. To force this, the *space3d* program is started with options **-%%** and **-X**. Thus, when you specify (at least) twice the special **'%'** option and besides that the **-X** option, you get a "display.out" file.

Note, when you start *space3d* with option **-X** and not the special option more than once, the *space3d* program shall invoke *Xspace* with option **-X** in the background for each cell name you have specified.

Note, because of historic reasons, the *space3d* program uses the flat extraction mode, when option **-X** is specified. And it shall also use the internal extraction mode for the substrate prepass. To use an external prepass program, use parameter "sub3d.internal=off". Note that only in internal mode drawing elements of that pass can be written.

5. THE DISPLAY FILE FORMAT

The "display.out" file format is a new ASCII-text file format. To be flexible, it is not compatible with any other display file format. At this moment, the "display.out" file is not completely self supporting. Because *Xspace* must read the technology file for mask color information. Thus, it is possible, that this information is not suitable for an old "display.out" file. Below is given a short example of a "display.out" file.

```
#c inv
#s 4 4 0.03125
#z 3.5e-06
w 512 -32 832 688
#pe
e0 -2147483647 -2147483647 472 -2147483647
e1 472 -72 472 728
t0 0 -2147483647 -2147483647 2147483647 472 -2147483647 2147483647
t1 1000 472 -72 728 512 -72 728
#end
```

As can be seen, a number of lines start with a '#' character. These lines are not really comment lines, but informative lines. However, unrecognized '#' lines can be used as comment lines. The given '#' lines are important, because they are used to interpret the data correctly. Each extracted cell in the "display.out" file starts with three header lines, of which the first two are most important. The '#c' line specifies the cell name and the '#s' line gives the used scaling factors. These factors are respectively "in-scale", "out-scale" and "lambda". The '#z' line gives the used maximum z-top value, which is used for 3D drawings. The last line of a "display.out" file must be '#end', to be sure that it is finished. Each extraction pass starts with a '#p' line, to identify this pass. This makes it possible for *Xspace* to stop before a new pass is done. The 'w' statement for any pass is also important, because it gives the cell bounding box in internal *space3d* units. The *Xspace* display bounding box command uses this information to display data of the cell in its bounding box window.

Note that all given statement coordinates are in internal *space3d* units. The used color masks are in hexa-decimal notation.

A complete statement list of the "display.out" file is given below.

display file statement	comment
#end	end of display file
#c name	cellname
#s inScale outScale lambda	scale factors
#z zMax	in micron
#p0	prepass 0 (-z)
#p1	prepass 1
#p1b	prepass 1b
#p2	prepass 2 (-j,-k)
#pe	extract pass
w xl yl xr yr	cell bounding box
ec color xl yl xr yr	drawEdge
eN xl yl xr yr	drawPairBoundary
tN color xl bl tl xr br tr	drawTile
p x y	drawScanPosition
m conductor_nr x1 y1 z1 x2 y2 z2	drawSpiderEdge (mesh)
d xl yl xr yr	drawDelaunay
g xl yl z1 x2 y2 z2	drawGreen
r xl yl xr yr	drawResistor
ro xl yl xr yr	drawOutResistor
rs xl yl xr yr	drawSubResistor
ru xl yl xr yr	undrawResistor
c xl yl xr yr	drawCapacitor
co xl yl xr yr	drawOutCapacitor
cu xl yl xr yr	undrawCapacitor
f xl yl xr yr	draw FE mesh
l xl yl xr yr	draw equi line

Note, the digit 'N' behind the edge and tile statements mean (0, not a conductor), (1, is conductor) and (2, also substrate conductor). Note that a conductor_nr < 0 specifies contact mesh, which is displayed in white color.