SPACE INCORRECT VARIABLES
APPLICATION NOTE

S. de Graaf

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

Report EWI-ENS 03-03
January 31, 2003

Last revision: December 9, 2003.

1. INTRODUCTION

1.1 Problem Description

We must look out for incorrect set variables (parameters) in the *space* program. It leads to unpredicted *space* program behaviour. As an example i describe the following found mistake.

Function cap3dInitParam() is only called by extract() when variable optCap3D is true.

Function cap3dInitParam() sets the variable extractDiffusionCap3d. When extractDiffusionCap3d is unset, its value is false. This is also the default value, when parameter "cap3d.omit_diff_cap" is not used.

The variable is used by Cap3D extraction. However, function recognizeElements() is using this variable in all *space* program passes, when optCap3DSave is true.

Variable optCap3DSave is true in all passes, but variable optCap3D not. The variable optCap3DSave is created, to remember the initial optCap3D value. It is set in function optionImplications(). Function optionImplications() is called before extractTree() is done. This is also true for *Xspace*, the interactive version of the *space* program.

The values of variables optCap3D/optCap3DSave are set as follows:

Pass	Value optCap3D	Value optCap3DSave
initial	**True** (-3c | -3C)	True
prePass0	False	True
prePass1	False(-b)/**True**(-B)	True
prePass2	False	True
extrPass	**True**	True

Pass	Value optCap3D	Value optCap3DSave
initial	False	False
prePass0	False	False
prePass1	False(-b)/**True**(-B)	False
prePass2	False	False
extrPass	False	False

Thus, in prePass1 by option -b it happens that variable extractDiffusionCap3d is not set (because function cap3dInitParam() is not called) and in the extrPass is set (because parameter "cap3d.omit_diff_cap" is set to "off").

This gives a problem in prePass1, because the value of variable extractDiffusionCap3d can be the first time False and the second time True. See the following table:

Pass	Value optCap3D	Value extractDiffusionCap3d	Param Value
initial	True	False	"off"
prePass1	False	False	
extrPass	True	True	
prePass1	False	True	
extrPass	True	True	
initial	True	False	"on"
prePass1	False	True	
extrPass	True	False	
prePass1	False	False	
extrPass	True	False	

By *Xspace*, it is even possible, that the parameter is set to "on" again (by given the menu command of rereading the parameter file). Than arises the opposite situation (see also above).

Note that, when extractDiffusionCap3d is true, no 2D cap elements are used at all. Thus, in the first case, the prePass1 is using the 2D caps and the extrPass is not using the 2D caps. Thus there are substrate terminals for these 2D caps. Note that in the other case, there are also substrate terminals in the prePass1 by recent generated technology files, because there are also subcont elements generated in the technology file for all 2D caps connected to substrate. These subcont elements are not skipped!

Note that for option -B extractDiffusionCap3d is set to true in prePass1. Thus, no 2D cap elements are seen in prePass1 for option -B. Thus, by old technology files, also no substrate terminals for used 2D diffusion caps are made in prePass1.

2. CONCLUSION

Because variable extractDiffusionCap3d is used in combination with variable optCap3DSave, it is important to set extractDiffusionCap3d like optCap3DSave before all passes (before the call to extractTree).

To handle the generation of substrate terminals for 2D caps in prePass1 correctly, we must add 2D cap information to the present subcont elements.