Process Mask Data File : User's Manual

A.A.J. de Lange
S. de Graaf

Department of Electrical Engineering
Delft University of Technology
The Netherlands

Abstract

This manual describes the structure and formats of the maskdata technology file of the Nelsis IC Design System. With the help of this manual a maskdata file can be defined when a new process is introduced in the system.

Date: August, 1988.
Last revision: March, 2012.

1. Introduction

The mask data file **maskdata** contains the layer information of a process. A number of tools consult this file, e.g. the layout-editor *dali*(1ICD), the plot-preprocessor *preplot*(1ICD), the design rule checker *dimcheck*(1ICD), the layout expansion programs *makeboxl*(1ICD) and *makegln*(1ICD), and the layout to circuit extractor *space*(1ICD). The appropriate maskdata file for the process in use is automatically selected when a tool is invoked. For this to happen, the maskdata file must be present in the process directory "**PROCESSPATH**/*process_name*" † (e.g. *process_name* = nmos) and the name of the process must be present in the processlist file "**PROCESSPATH**/processlist". Normally, the user is not allowed to specify his/her own maskdata file. A new maskdata file (new process) must be added to the technology database by the manager of the design system (user name is *cacd*).

Note: When a new release of software and/or technology base is installed, the old technology base is destroyed !!! Therefore, before installing a new release, the design system manager must save the maskdata file (and other technology files) that are not part of the new release. After the installation procedure, the saved maskdata file(s) (and other technology files) can be put back into the technology database.

The **maskdata** file contains layer specifications for the given process, one line per layer. The process type and some comment on the process is given before the first layer-specification line in the file.
For every process mask (layer) a name, a type (interconnect, symbolic, other), some colors and fill-styles and a PG-tape generation type and sequence number are specified. When the type of layer *layer_name* is interconnect, automatically an accessory terminal layer†† with name **t_***layer_name* is defined.

Example : when a metal interconnect layer *nm* is defined the layer *t_nm* will automatically be defined as terminal layer for the interconnect layer *nm*.

Columns 5 through 10 specify colors and fill-styles on various types of graphics hardware. Devices that are supported in the **maskdata** file are plotters, XWindows devices and ColorMask terminals (obsolete). The most important entries are those for graphical programs that use the standard XWindows color table such as *dali*, and the plot programs such as *preplot*, which occupy respectively the 7/8 th column and the 9/10 th column in the **maskdata** file. The use of colors by the plotter also depends on the arrangement of pens in the plotter device. The appropriate ordering of pens must be

---

† PROCESSPATH is normally ICDPATH/share/lib/process

†† Terminals indicate and label the places where a cell may be interconnected with its surroundings. In all areas where connections are made between different cells, not only interconnect layers but also their accessory terminal layers must be present.

determined experimentally by the plotter manager.
Finally, the columns 5/6 are reserved for ColorMask devices and the columns 3/4 are used for PG-tape generation.

In the next section, the complete syntax description of the Mask-Data file is given. The last section gives two examples of Mask-Data files.

2. Syntax Description of the Mask-Data

The Syntax of the mask-data file is given in the table below. Here we use the Wirth [1] notation. The terminal symbols (tokens) are denoted by words between single quotes, and words in italic form, and non-terminal symbols by words in lower case. Alternatives are separated by a vertical bar, i.e. a|b means a or b. Repetition is denoted by curly brackets, i.e. {a} stands for empty|a|aa|aaa|... Optionality is expressed by square brackets, i.e. [ab] stands for empty|a|b. Parentheses serve for grouping, e.g. (a|b|c) means ac|bc. Literals are enclosed in single quote marks. Each production begins with a a non-terminal symbol followed by an equality-sign and a sequence of terminals and non-terminals and terminated by a period.

TABLE 2.1. Syntax Description of Mask Data File

mask_definition	=	process_type process_comment layer_specs.	
layer_specs	=	layer_spec {layer_spec}	
layer_spec	=	layer_name layer_type pg_job_nr pg_type color_1 fill_1 color_2 fill_2 color_3 fill_3 layer_comment.	
pg_job_nr	=	’0’ ... ’n’.	(0 = not processed)
pg_type	=	’0’ | ’1’.	(0 = negative, 1 = positive)
process_type	=	name.	(nmos, cmos, bipolar,...
process_comment	=	text.	
layer_comment	=	text.	
layer_name	=	name.	
layer_type	=	’0’ | ’1’ | ’2’.	(0 = other, 1 = interconnect, 2 = symbolic)
color_1	=	integer.	(ColorMask color table)
color_2	=	integer.	(dali color table)
color_3	=	integer.	(plotter pen table)
fill_1	=	fill.	(see color_1)
fill_2	=	fill.	(see color_2)
fill_3	=	fill.	(see color_3)
fill	=	’0’ | ’1’ | ’2’.	(0 = hollow polygons, 1 = solid, 2 = dashed)
integer	=	digit { digit }.	
digit	=	’0’ | ’1’ | ’2’ | ’3’ | ’4’ | ’5’ | ’6’ | ’7’ | ’8’ | ’9’.	
text	=	’"’ name { space name } ’"’.	
space	=	’ ’.	(blank-character)
name	=	’"’ letter { letter | digit } ’"’.	
letter	=	’a’ | ’b’ | ’c’ | ’d’ | ’e’ | ’f’ | ’g’ | ’h’ | ’i’ | ’j’ | ’k’ | ’l’ | ’m’ | ’n’ | ’o’ | ’p’ | ’q’ | ’r’ | ’s’ | ’t’ | ’u’ | ’v’ | ’w’ | ’x’ | ’y’ | ’z’ | ’A’ | ’B’ | ’C’ | ’D’ | ’E’ | ’F’ | ’G’ | ’H’ | ’I’ | ’J’ | ’K’ | ’L’ | ’M’ | ’N’ | ’O’ | ’P’ | ’Q’ | ’R’ | ’S’ | ’T’ | ’U’ | ’V’ | ’W’ | ’X’ | ’Y’ | ’Z’	

Extra comment starts on a line with ’#’ and ends at the end of the line.

3. Examples

In this section the example of a mask-data file for some frequently used processes : the TU Delft 4µ nmos process (table 3.1) and the Philips 2µ C5TH process (table 3.2). In table 3.1 we see that the third layer definition has name=’nm’, layer-type=1 (interconnect), **dali** color table entry=4 (i.e. color=blue, see column 7), and fill-style=1 (solid) (see column 8). The plotter has pen-number=4 (see column 9) and fill-style=0 (hollow) (see column 10) . The above items are printed in bold. The strings "standard nmos process", "polysilicon", "diffusion", "metal", etc., are comments.

The ColorMask device column entries and the PG-tape entries are of less importance, because all ColorMask devices are replaced by graphics terminals and workstations that use the **dali** color table. The PG-tape entries are scarcely used, because PG-tape generation is only carried out if the total design is finished and must be transported to the foundry. The *ftape* program uses the sequence number (job number) in which order the jobs must be processed (also glass/reticle number) by the foundry. Note: The pg-type entry is only used if the ftape program is used with the option -d (default pos/neg masks selection). First are all positive masks (jobs) generated and after that all negative masks (jobs) in sequence.

TABLE 3.1. Mask-Data file (PROCESSPATH/nmos/maskdata) of Delft 4µ nmos process

#										
"nmos"		"TUD standard single metal"								
np	1	5	1	1	1	1	1	1	0	"polysilicon"
nd	1	1	1	2	1	2	1	2	0	"diffusion"
nm	**1**	7	0	3	1	**4**	**1**	**4**	**0**	"metal"
ni	0	3	1	4	1	3	0	5	0	"implant"
nb	0	4	1	5	1	5	0	3	0	"buried contact"
nx	0	2	1	6	1	6	0	6	0	"undercrossing"
ng	0	8	0	7	1	7	1	7	0	"glass"
nc	0	6	1	7	1	0	1	8	0	"contact"
bb	2	0	1	0	0	7	0	8	0	"bounding box"

TABLE 3.2. Mask-Data file (PROCESSPATH/c5th/maskdata) of 2µ C5TH process

```
#
"cmos"    "TUD single metal derived from Philips C500"
ps      1     1    1    1    1     1    1    1    0    "polysilicon"
od      1     2    1    2    1     2    1    2    0    "active area"
in      1     3    1    3    1     4    1    4    0    "metal"
nw      0     4    1    4    1     5    0    5    0    "n-well"
sn      0     5    1    5    1     6    0    6    0    "shallow n implant"
sp      0     6    1    6    1     3    0    3    0    "shallow p implant"
con     0     7    1    7    1    15    1    8    0    "contact metal/n+area"
cop     0     8    1    7    1    15    1    8    0    "contact metal/p+area"
cps     0     9    1    7    1    15    1    8    0    "contact metal/poly"
cb      0    10    1    7    1    15    1    8    0    "contact to bondpads"
bb      2     0    1    0    0     7    0    8    0    "bounding box"
```

CONTENTS

LIST OF TABLES

1. N. Wirth,, ‘‘What can we do about the unnecessary diversity of notation for syntactic definitions ?,’’ Comm. ACM 20 (Nov. 1977).