INTRODUCING A NEW PROCESS

Arjan van Genderen

Circuits and Systems Group
Department of Electrical Engineering
Delft University of Technology
The Netherlands

Report ET-ENS 2003.01

Last revision: June 12, 2006

Contents

1 Introduction

This document explains how new processes (or technologies) can be added in the design system. Basically this document gives an overview of the process files that are required and how they are organized. For more details the document refers to other manuals and manual pages.

Note that to add process data to the system, you need information about the process like mask names, mask gds numbers, material resistances, material dimensions, transistor models etc., depending on which tools you are going to use. This information is typically provided by the organization/company via which you are going to have your circuit fabricated.

2 The Process Directory

The technology files for a process are stored in a separate (process) directory. This way, all design projects that use the process information can refer to the same directory.

2.1 Default Location of a Process Directory

The default location of a process directory is under `$ICDPATH/share/lib/process`, where `$ICDPATH` is the installation directory of the software. The standard software distribution contains several process directories that are all stored under this directory (e.g. c3tu, dimes03, fishbone, scmos_n, tsmc025). The file `$ICDPATH/share/lib/process/processlist` contains a list of these process directories and assigns an id (number) to each process.

```
% cat /usr/cacd/share/lib/process/processlist
# rcsid = "$Id: procdir.tex,v 1.3 2003/05/21 10:09:37 arjan Exp $"
# process list
# proc_id proc_name
1  nmos      # TUD demo nmos process
3  scmos_n   # scalable cmos process
6  gatearray # sea-of-gates gatearray process
18 c3tu      # 1.6 micron cmos process
23 dimes01   # first (bipolar) process of DIMES
40 octagon   # sea-of-gates octagon process (modified c3tu)
41 fishbone  # sea-of-gates fishbone process (1.6 micron)
44 scmos-orb-2 # Orbit 2.0 micron cmos process
45 ami-c5n   # AMI 0.5 micron cmos process
46 dimes03   # current bipolar process of DIMES, TU Delft
60 tsmc025   # MOSIS TSMC CMOS025 (0.25u) process
```

When creating a design project with `mkpr` you can select the process that you want to use by its number. When you want to add a process to the default location for process directories, you have to place the directory as a subdirectory under `$ICDPATH/share/lib/process` and you have to add a new line with the (unique) process id and process name to the file `$ICDPATH/share/lib/process/processlist`.

2.2 Other Locations of a Process Directory

A process directory may also reside at an arbitrary place on the file system. In that case, when you want to create a design project that refers to the process information, you have to use the option -p with `mkpr` to specify the path to the directory.

3 Process Files

This chapter lists the files that may be present in a process directory. It is described briefly which information is contained by the file, which tools are using it, and where more detailled information can be found.

Most of the process files can be created manually, using some text editor. As an alternative, a graphical technology interface tool called `spock` may be used to generate these files (see the manual page of `spock` for more details). Further, some process files are generated by some dedicated tool that is part of the software distribution.

3.1 bmlist.gds

Specifies the conversion of gds2 mask numbers to process mask names and vice versa

Used by: cgi, cig

More information: manual pages bmlist, cgi, cig

3.2 default_lambda

Specifies the default value for lambda for a design project for this process.

Used by: mkpr

More information: manual page mkpr

3.3 epslay.def

Describes how a postscript file is generated from a layout

Used by: getepslay

More information: manual page getepslay

3.4 maskdata

Specifies the masks that are used in the process and the properties of these masks as used in several tools.

Used by: all layout tools

More information: manual page maskdata

3.5 match_prim

Network primitive files

Used by: match

More information: manual page match.

3.6 slsmod

Device models for the switch-level simulator sls. The `slsmodgen` tool can be used to derive these models from spice models.

Used by: sls

More information: SLS Switch-Level Simulator User's Manual, the directory `$ICDPATH/share/lib/slsmodgen`, manual pages of sls and slsmod.

3.7 space.def.p

Parameter file for Space

Used by: space, space3d, Xspace

More information: All Space Manuals, manual page of space.

3.8 space.def.s

Element definition file for Space.

Used by: space, space3d, Xspace

More information: All Space Manuals, manual pages of space, subresgen and tecc.

3.9 space.def.t

Compiled element definition file for Space, as obtained after running the program tecc.

Used by: space, space3d, Xspace

More information: All Space Manuals, manual pages of space and tecc.

3.10 xspicerc and spice3f3.lib

Files that, among other things, contain the spice models

Used by: xspice

More information: Space User's Manual, manual page xspice.