Spansion Model Manual

User's Manual

Version 1.0

Table of Contents

1 Document History

Version/Date	Modification
V1.0: 01/23/04	**initial version**

2 Model Overview

The self-extracting archive contains model files for SPANSION Flash Memory (see Figure 1). These models support:

- High/Low or Top/Bottom boot option (model dependant, see data sheet)
- Verilog / Vhdl behavioral simulation
- Timing-accurate simulation (including support for interconnect path delays)
- Built-in Timing checks
- Pre-loading each instance of the model in the top-level netlist with
    - Protection Mode for all sectors (if applicable)
    - Contents of main memory array
    - Contents of Secure Silicon (if applicable)
    - Contents for Top/Bottom boot sectors (if applicable)

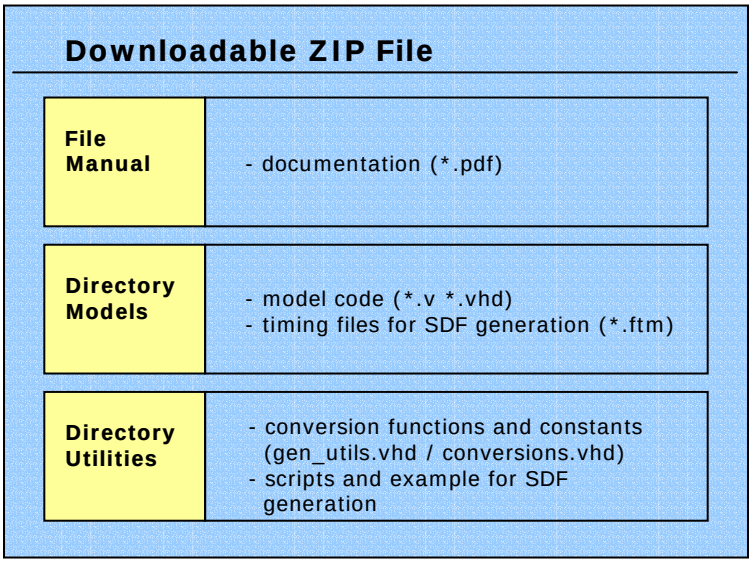

Figure 1 Contents of download file

3 VHDL/Verilog - Model

3.1 Required Files

3.1.1 Files for a Simple VHDL Behavioral Simulation

The model file `model.vhd` is located in the model directory. It relies on a set of functions defined in the files `gen_utils.vhd` and `conversions.vhd`. These 3 files comprise the minimum set of files for a behavioral VHDL simulation.

3.1.2 Files for a Simple Verilog Behavioral Simulation

The model file `model.v` is located in the model directory. It does not depend on other files and can be run as is in a behavioral Verilog simulation.

3.1.3 Pre-loading Flash Model in Behavioral Simulations

In order to reduce simulation time and simplify the simulation, each instance of the model can be pre-loaded, i.e. the simulation starts up as if the Flash memory had been programmed before and enters the simulation in a certain mode and contains certain data. The pre-loading feature is controlled by 2 attributes

- `UserPreload`
- `preload_file_name(s)`

These attributes should be assigned to each instantiation of the model in the top-level netlist.
If these attributes are not assigned, the model will default to not use pre-loading.

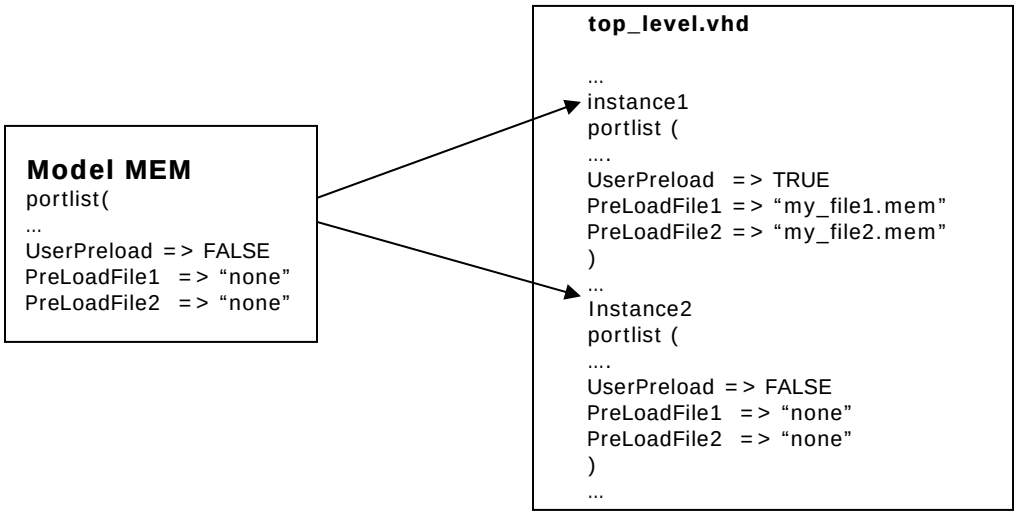

Figure 2 Instantiations of models with different pre-load configurations

3.1.4 Writing Pre-load Files

The test bench process loads default values into memory array, CFI field and protection bits. The Memory array is initialized to all 0xFFFF (default flash memory state), the CFI space is preloaded according to the data sheet. Sectors are unprotected by default. The values can be overwritten by defining pre-load files for the model instantiation in the test bench:

mem_file_name	=>	loads memory array with data
prot_file_name	=>	lists protected sectors
Secsi_file_name	=>	loads Secure Silicon Sector (if available)

**** NOTE: *Preload files should not have empty lines*
**** NOTE: *Provide leading zeros for addresses*

Example for memory/secsi pre-load file

```
// select Addr0: ADDR_0= AA, ADDR_1= 55, ADDR_2= 11
// select AddrA8: ADDR_8= 01, ADDR_9= 02, ADDR_10= 03
@00000
AA
55
11
@000A8
01
02
03
```

Example for protected sector pre-load file

```
// select sector 01, 19, EE: set 1 to protect
@01
1
@19
1
@EE
1
```

3.2 Compiling Model Files

3.2.1 Compilation of VHDL Model files

The following list shows which files need to be compiled to which library:

- conversions.vhd : compile to library FMF
- gen_utils.vhd: compile to library FMF
- model.vhd: compile to library work

The file `model.vhd` depends on `conversion.vhd` and `gen_utils.vhd` and therefore needs to be compiled only after these two files.

3.2.2 Compilation of Verilog Model Files

The file `model.v` can be compiled as is. No further libraries need to be provided.

3.3 Simulating Model Files

3.3.1 Basic and Timing Simulation

For a basic simulation no files beyond the files used for compilation are required.

For performing timing simulation, the timing information needs to be provided as an SDF file. This allows for a complete back-annotation including interconnect path delays (e.g. for high speed boards) on system level. This approach is identical to typical ASIC simulation flows.

3.3.2 Generating SDF files

The SDF information for all speed grades of a model is provided in the FTM file. The FTM files for the Verilog and VHDL version of the model are located in the model directory. From the FTM file, an SDF file can be generated by 2 different methods:

Method 1
Select the section with the desired speed grade (i.e. the OPN) in the FTM file. Copy and paste this section into the final SDF file for the overall simulation.

Method 2
For VHDL-only simulations a global SDF file can be generated automatically if an FTM files exist for each component of the overall simulation environment. The global SDF file can be created by executing the Perl script provided in the utilities directory. The script parses the top-level netlist for instances of components. Each instance has a timing attribute (TimingModel) with a value that is equivalent to its OPN in the data sheet (see Figure 3).

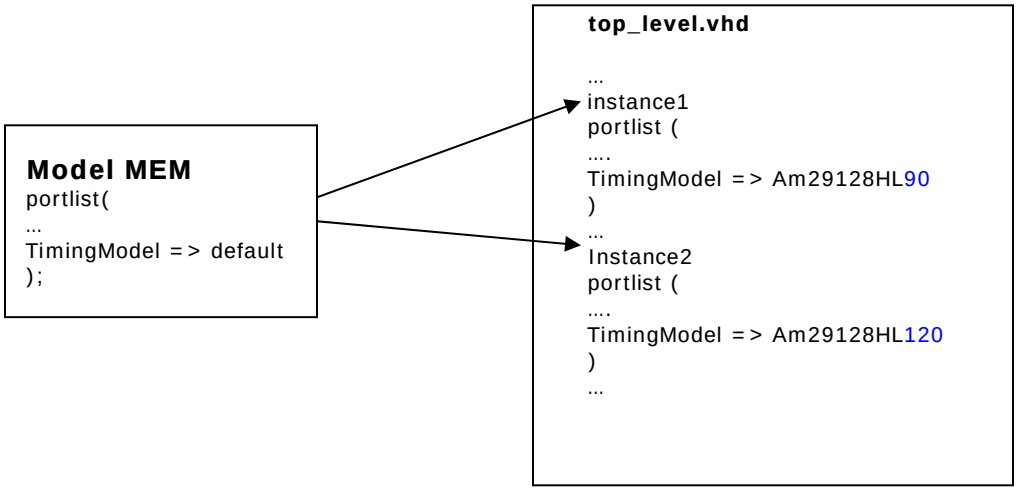

Figure 3 Instantiations of models with different speed grades

Based on these attributes the Perl script selects the appropriate timing information from the FTM files for each component and assembles the overall SDF file. The Perl script is executed by entering:

```
perl   mk_sdf_204.pl   top_level.vhd   // Unix environment
```

Note that this script is applicable to VHDL-only simulations (no mixed Verilog/VHDL modules) and requires the command script `mk_sdf.cmd` to reside in the same directory as the perl script. Both, `mk_sdf_204.pl` and `mk_sdf.cmd` as well as an example of a testbench, the corresponding timeing file and the resulting SDF file are provided in the directory `utilities/CreateSDF`.

Note: In command script `mk_sdf.cmd` the directory setting for the `timingfile_dir` has to be set to where `mk_sdf_204.pl` and `mk_sdf.cmd` are as well as testbench and timing file are residing e.g. to `/user/USERNAME/CreateSDF`.

4 Support

We appreciate your feedback or suggestions to ensure our models meet your needs. For contacting us please email to:

memorytools.help@spansion.com