USAGE OF MEMAKER EDA DELIVERABLES

Memaker Overview
Rev.: 1.0
Issue Date: December 2009

FARADAY

REVISION HISTORY

Date	Rev.	From	To
Dec. 2009	1.0	-	Original

Faraday Technology Corporation
No. 5, Li-Hsin Road III, Hsinchu Science Park, Hsinchu City, Taiwan 300, R.O.C.

Faraday's home page can be found at:
http://www.faraday-tech.com

TABLE OF CONTENTS

1.1 Memory EDA Deliverable Items

Deliverable Item	Description
Synthesis model	Synopsys synthesis timing model
Simulation models (Front-end)	● Verilog ● VHDL
ATPG models	● Synopsys TetraMAX model ● Mentor FastScan model
MBIST model	Mentor architecture MBIST model
MDT model	Novas Verdi memory content table
P & R model	Cadence LEF P & R model
Netlist	LVS netlist
Physical Layout	GDSII layout model

1.2 Using Verilog Simulation Model

After generating the Verilog model, users can refer to the following steps for simulating a design model.

Step 1: Check the syntax of the Verilog model:

ncverilog <name.v>

where ***<name>.v*** is the name of the Verilog model.

Step 2: Run the simulator:

ncverilog <test-bench>.v

The simulation output is written to the ncverilog.log file, which is saved in the current working directory.

1.3 Using VHDL Simulation Model

After generating the VHDL model, users can refer to the following steps for simulating a design model.

Step 1: Create the working library:

It is necessary to compile any VHDL model through ModelSim.

vlib *work*

Step 2: Compile the VHDL design:

vcom <name>.vhd

where ***<name>.vhd*** is the name of the VHDL model.

Step 3: Instantiate the VHDL model within a test bench:

Users can refer to <name>_pkgs by using the following clause:

Use work.<name>_pkgs.all;

Users are then ready to compile the test bench:

vcom test-bench.vhd

Step 4: Run the simulator:

Vsim

ModelSim will trigger a pop-up GUI window. Users can refer to the ModelSim user manual for more details. However, the retain time is not supported in the VHDL VITAL. When generating a SDF file, users may ignore the information of the retain time, or use SDF 2.1 instead.

1.4 Using Synopsys Synthesis Model

After generating the Synopsys model, users can refer to the following steps for simulating a design model.

Step 1: Initiate the Synopsys Design Compiler environment:

dc_shell

Step 2: Use the Synopsys commands to include the Synopsys model:

read_lib <name>_<case>.lib

write_lib <name>_<case>

link_library= <name>_<case>.db

target_library= <name>_<case>.db

read –f verilog design.v

write_timing –context verilog –f sdf –v2.1 –o output

where ***<name>_<case>*** is the name of the Synopsys library, ***<name>_<case>.lib*** is the name of the Synopsys model, ***design.v*** is the name of the top-level netlist, and ***output*** is the name of the output file. Faraday provides 3-corner models to the customers for the design synthesis. Please setup the 3-corner models as: <WC>: Worst case; <TC>: Typical case; <BC>: Best case

Step 3: Invoke the Synopsys PrimeTime environment when using PrimeTime to generate a SDF:

pt_shell

Step 4: Execute the following steps in the Synopsys commands to generate a SDF:

read_db <name>_<case>.db

set link_path <name>_<case>.db

read_verilog design.v

write_sdf –version 3.0 –o output

1.5 Using LEF P & R Model

After generating the LEF model, users can refer to the following steps for inputting the LEF model.

Step 1: Invoke the Cadence Silicon Ensemble environment

Step 2: Type the following command in the command window

INPUT LEF FILENAME “<path>/<name>.lef”

where ***<path>*** is the path to store the LEF model and ***<name>.lef*** is the name of the LEF model.

This creates the block and pin information in GDSII for P & R.

1.6 Using LVS Netlist, GDSII Layout, and H-Cell List

After generating the LVS netlist, GDSII layout, and H-Cell list, users are recommended to use these deliverables for verifying the conjunction. Users may use a verification tool, such as Cadence Dracul, or Mentor Graphic Calibre, to compare GDSII with the LVS netlist. However, the LVS netlist should be combined to the chip-level netlist when the chip is fully assembled. This precautionary action ensures that the Place & Route (P & R) tools will not cause circuit short or open circuit in a user chip. The list file of H-Cell is a cell-correspondence file that is used for the hierarchical LVS comparison of the Mentor Calibre. It contains one cell per line as listed below:

layout-name source-name

where *layout-name* and *source-name* are separated by one or more spaces or tab characters.

Users can refer to the following command to include the H-Cell list while performing the hierarchical LVS verification.

calibre –lvs –hcell ***filename***

where ***filename*** is generated along with the GDSII selection from Memaker.

Please use this hierarchical correspondence information to perform the hierarchical LVS verification according to the requirement of the tool.

1.7 Using TetraMAX Model and FastScan Model

Users can refer to the steps below for generating the ATPG model of Synopsys TetraMAX.

Step 1: Invoke the Synopsys TetraMAX GUI environment command.

> tmax

Step 2: Click **Netlist** to read the TetraMAX model.

If users want to use more detailed TetraMAX model, please refer to “TetraMax ATPG User Guide” by Synopsys.

Users can refer to the steps below for generating the ATPG model of Mentor FastScan.

Step 1: Create the do file, users can refer to the following example to create a do file.

Do file creation example:

analyze cont sig –auto

report clock

set atpg limits -pattern_count 123456789

set sys mode atpg

set simulation mode comb -d 4

create pat –auto

save patterns pat_verify.v -verilog –rep

save patterns pat_verify –rep

write netlist lib_top.v -verilog –rep

report loops

exit -d

Step 2: Invoke the Mentor FastScan environment command

fastscan all -model -lib <fastscan model>.fastscan -nogui -dofile dofile

If users want to use the more detailed FastScan model, please refer to “Design-for-Test Common Resources Manual” by Mentor.

1.8 Using MBIST Model

Users can refer to the steps below for generating the MBIST model of Mentor MBISTArchitect.

Step 1: To create a do file, users can refer the example listed below:

Do file creation example:

load library <MBIST>.mbist

add memory models try -collar mbistG1_try

set memory clock –test

set controller delay 2

setup pipeline register output 2

report pipeline registers

set controller debug –on

setup diagnostic clock -slow_tester_clk

set pin name diag_clk test_clk

set controller hold –on

setup mbist algorithms march2_cb march1_cb checkerboard

delete diagnostic monitor –all

add diagnostic monitor failmap addr_reg tstate rw_state

set design name controller -module mbistG1 -instance mbistG1

set design name collar -instance MEM

setup file naming -test_bench mbistG1_tb.v

run

rep algo step

save bist -verilog –replace

exit -discard

Step 2: Invoke the Mentor MBISTArchitect environment command.

Mbistarchitect –nogui –do dofile

If users want to use the more detailed MBIST model, please refer to “MBISTArchitect Process Guide” by Mentor.

Note: If users want to use the full-speed MBIST model, please add the following command lines to the Do file.

set memory clock -test invert
set controller delay 2
setup pipeline registers compare_result on

1.9 Using MDT Model

Users can refer to the steps below for generating the MDT model of Verdi.

Step 1: Invoke the Verdi GUI environment command
> verdi

Step 2: Click **Exploration -> Memory Definition Table** -> **Load** to read the MDT model.

If users want to use the more detailed MDT model, please refer "Novas Command Reference Manual" by SpringSoft