A Tutorial on Path Delay Faults and Small Delay Defects Testing Using Synopsys TetraMax

Himanshu Thapliyal
Department of Electrical and Computer Engineering
Duke University, NC 27708
March 15, 2012

1. PREPARATION

In this tutorial, we will use the ISCAS s5378 benchmark. The scan inserted netlist and all the needed scripts are provided in the directory **/opt/digital/share/ece269_lib/**

COMMANDS:

(i) For running PrimeTime script for path delay fault :

pt_shell -f ./scripts/PT_scriptpd.tcl

(ii) For running PrimeTime script for small delay defects :
pt_shell -f ./scripts/PT_scriptsd.tcl

(iii) For running TetraMax script for path delay fault ATPG :
tmax -shell ./scripts/path_delay_atpg.tcl

(iv) For running TetraMax script for small delay defects ATPG :
tmax -shell ./scripts/sddatpg.tcl

2. PATH DELAY FAULT

The path delay fault model is used to detect whether the critical paths are too slow because of manufacturing defects or variations [1]. The path delay test flow using Synopsys tools is shown in Figure 1:

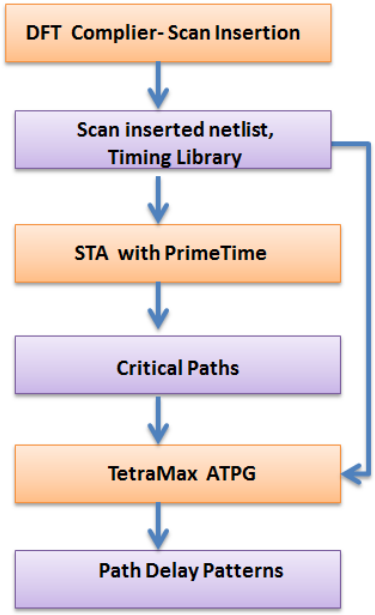

Figure 1: Path delay test flow using Synopsys tools

2.1 PRIMETIME SCRIPT FOR REPORTING THE WORST CRITICAL PATHS:

In order to do ATPG and fault simulation using TetraMAX for scan-based path delay fault testing, we need a list of critical paths reported by PrimeTime. The PrimeTime script for reporting the worst critical paths are shown in Table 1. The PrimeTime script calls the script pt2tmax.tcl. The script pt2tmax.tcl has the command write_delay_paths which writes the critical paths in the format that can be read by TetraMAX. The example script shown in Table 1 writes 200 worst critical paths. The example script writes the critical paths inside the directory 'pathdelay'.

Clock Perioid: In the PrimeTime script, the clock period is set using the command:
set CLK_PERIOD 0.15

In order to obtain the critical paths using the write_delay_paths command, the design need to be run at higher clock frequency so that there will be negative slacks in the path.

Table 1: PrimeTime Script for Reporting the Worst Critical Paths

```
set library_name NangateOpenCellLibrary
set link_library [list * ./nangate_scan.db]


read_verilog  s5378_bench.v

#ungroup -flatten -all
# Define top level in the hierarchy
current_design "s5378_bench"

link_design

set_max_area 0
#Clock setup commands
set CLK_PERIOD 0.15
set CLK "blif_clk_net"
create_clock -period $CLK_PERIOD [get_ports $CLK]
set_clock_transition -rise 0.05 [get_clocks $CLK]
set_clock_transition -fall 0.03 [get_clocks $CLK]
set_clock_latency -rise 0.01 [get_clocks $CLK]
set_clock_latency -fall 0.03 [get_clocks $CLK]
set_ideal_network [get_ports blif_clk_net]
#set_propagated_clock [all_clocks]

report_timing
source pt2tmax.tcl
write_delay_paths -max_paths 200  -nworst 1  -delay_type max ./pathdelay/timing200withoutmob.rpt

quit
```

2.2 TETRAMAX SCRIPT FOR PATH DELAY FAULT ATPG:

In order to perform test pattern generation for path delay faults, we need library file in the Verilog format. The basic DFT Compiler-to-TetraMAX design flow can be summarized in the following steps [2]:

1. Read in the Verilog netlist (*s5378_bench.v is the Verilog netlist inside the* 'Tut_Benchmark' directory *)*
2. Read in the library models (nangate_scan.v is the library available *inside the* 'Tut_Benchmark' directory
3. Build the ATPG design model
4. Read in the STIL test protocol file, automatically generated by DFT Compiler (*des2d1.spf is the STIL test protocol for the current benchmark)*
5. Set desired ATPG options:
   set_atpg -full_seq_merge medium
6. Add path delay faults by issuing these commands:
   set_faults -model path_delay
   set_faults -report collapsed
7. Read in delay paths:
   add_delay_path ./pathdelay/timing200withoutmob.rpt
   add_faults –all
   This adds worst 200 critical paths reported by PrimeTime.
8. Run ATPG:
   run_atpg –auto
9. Write out path delay test patterns:
   write_patterns ./pathdelay/patterns200.stil -internal -format STIL -unified_stil_flow -replace

The complete TetraMax script for path delay fault ATPG is shown in Table 2.

Table 2: TetraMax Script for Path Delay Fault ATPG

```
set top_module s5378_bench
set scan_lib ./nangate_scan.v
set stil_file  ./des2d1.spf

read_netlist -delete
# read in scan cell library
read_netlist $scan_lib -library
# read in user's synthesized verilog code
#read_netlist $synthesized_files

read_netlist  s5378_bench.v

run_build_model $top_module
# ignoring warnings like N20 or B10
# Set STIL file from DFT Compiler
set_drc $stil_file
# run check to see if synthesized code violates any testing rules
add_clocks 0 blif_clk_net
add_pi_constraints 0 test_se
run_drc

set_atpg -full_seq_merge medium
set_faults -model path_delay
set_faults -report collapsed
set_rule P7 war
add_delay_path ./pathdelay/timing200withoutmob.rpt
add_faults -all
#set_atpg -nofull_seq_atpg
run_atpg -auto
write_patterns ./pathdelay/patterns200.stil -internal -format STIL -unified_stil_flow -replace

exit
```

3. SMALL DELAY DEFECTS TESTING

Small Delay Defects (SDD) contribute to delays that are much smaller than the clock cycle [2]. The SDD test flow using Synopsys tools is shown in Figure 2.

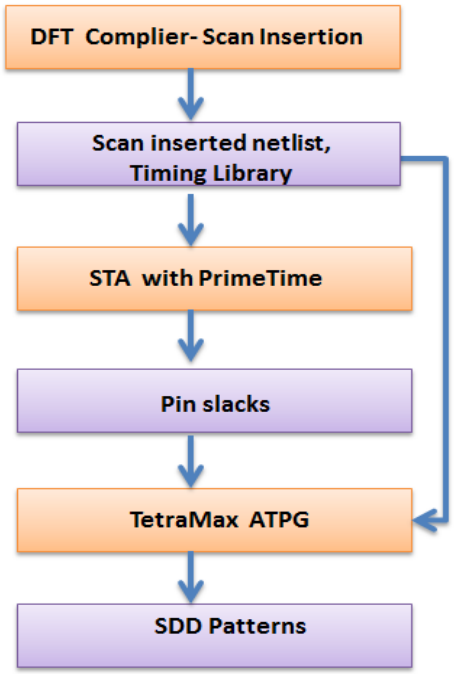

Figure 2: SDD test flow using Synopsys tools

3.1 PRIMETIME SCRIPT FOR EXTRACTING SLACK DATA

To test for small delay defects, PrimeTime extracts slack data from the netlist. PrimeTime command report_global_slack extracts slack data for all pins. Table 3 shows the PrimeTime script for extracting slack data for all pins. There are three new commands in the PrimeTime script which are as follows:

```
set timing_save_pin_arrival_and_slack TRUE
update_timing
report_global_slack -max -nosplit > ./SDD/slack.dat
```

The example script writes the slack information of the pins in the file slack.dat and stores it inside the SDD directory. In the example PrimeTime script, the clock period is setup as 1.1 ns.

Table 3. PrimeTime Script for extracting the slack data for all pins

```
set library_name NangateOpenCellLibrary
set link_library [list * ./nangate_scan.db]


read_verilog  s5378_bench.v

#ungroup -flatten -all
# Define top level in the hierarchy
current_design "s5378_bench"

link_design

# SET CONSTRAINTS

set_max_area 0
#Clock setup commands
set CLK_PERIOD 1.1
#set DFF_CKQ 0.02
#set SETUP_TIME 0.01
set CLK "blif_clk_net"
create_clock -period $CLK_PERIOD [get_ports $CLK]
set_clock_transition -rise 0.05 [get_clocks $CLK]
set_clock_transition -fall 0.03 [get_clocks $CLK]
set_clock_latency -rise 0.01 [get_clocks $CLK]
set_clock_latency -fall 0.03 [get_clocks $CLK]



set_ideal_network [get_ports blif_clk_net]
#set_propagated_clock [all_clocks]
#set_clock_uncertainty 0.2 [all_clocks]

#set_operating_conditions -min WORST -max WORST

report_timing


set timing_save_pin_arrival_and_slack TRUE
update_timing
report_global_slack -max -nosplit > ./SDD/slack.dat


quit
```

3.2 TETRAMAX SCRIPT FOR SMALL DELAY DEFECT ATPG:

The basic DFT Compiler-to-TetraMAX design flow can be summarized in the following steps:

1. Read in the Verilog netlist (*s5378_bench.v is the Verilog netlist inside the* 'Tut_Benchmark' directory *)*
2. Read in the library models (nangate_scan.v is the library available *inside the* 'Tut_Benchmark' directory
3. Build the ATPG design model
4. Read in the STIL test protocol file, automatically generated by DFT Compiler (*des2d1.spf is the STIL test protocol for the current benchmark)*
5. *Read Slack File (If there is no slack file then the regular transition delay ATPG is performed).*
   read_timing ./SDD/slack.dat
6. Transition-Delay Fault ATPG Timing Modes
   set_delay -launch_cycle last_shift
   (this sets ***Launch-On Shift (LOS) other modes can be setup similarly***
   set_faults -model transition
   add_faults -all
   set_faults -report collapsed
   set_faults -summary verbose
7. Delay effectiveness
   set_delay -max_tmgn 0.11
   Delay effectiveness is a coverage metric used to quantify only faults that are detected along paths having slacks less than or equal to max_tmgn [3]. You can vary max_tmgn on the basis of your clock period, if the clock period is 1.1 ns, the 10% slack would be 0.11 ns.
8. Run ATPG:
   run_atpg basic_scan_only -ndetects 1
9. *Report Delay Effectiveness and SDQL value*
   report_faults -slack effectiveness
   report_faults -slack sdql
10. Write out SDD test patterns:
    write_patterns ./SDD/pattern_sdd_slack10%.stil -internal -format STIL -unified_stil_flow -replace

Table 4: TetraMax Script for SDD ATPG

```
set top_module s5378_bench

set scan_lib ./nangate_scan.v
set stil_file  ./des2d1.spf

read_netlist -delete
# read in scan cell library
read_netlist $scan_lib -library
# read in user's synthesized verilog code
#read_netlist $synthesized_files

read_netlist  s5378_bench.v


run_build_model $top_module
# ignoring warnings like N20 or B10
# Set STIL file from DFT Compiler
set_drc $stil_file
# run check to see if synthesized code violates any testing rules
add_clocks 0  blif_clk_net
add_pi_constraints 0 test_se
run_drc

read_timing ./SDD/slack.dat
set_delay -launch_cycle last_shift
set_faults -model transition
add_faults -all
set_faults -report collapsed
set_faults -summary verbose
set_delay -max_tmgn 0.25
run_atpg basic_scan_only -ndetects 1
report_faults -slack effectiveness
report_faults -slack sdql
write_patterns ./SDD/pattern_sdd_slack10%.stil -internal -format STIL -unified_stil_flow -replace
exit
```

References:

1. TetraMAX® ATPGUser Guide Version F-2011.09-SP3, January 2012
   http://www.synopsys.com/Tools/Implementation/RTLSynthesis/Pages/TetraMAXATPG.aspx
2. PrimeTime® Fundamentals User Guide Version F-2011.12, December 2011
   http://www.synopsys.com/Tools/Implementation/SignOff/Pages/PrimeTime.aspx
3. R. Mattiuzzo, D.Appell and C.Allsup, "Small Delay Defect Testing",
   Test & Measurement World, pp. 37-41, June 2009.