



Goyatek Technology Inc.

Goyatek Toggle Rate Calculation User Guide

Alex Hsu
Version 1.0
Date 8/13/2002

CAD Section
Goyatek Technology Inc.

3F, No.25, R&D Road II, Science Based Industrial Park
Hsin-Chu, Taiwan, R.O.C.
alexh@goya.com.tw

@Copyright 2002 GOYATEK Technology Inc. Limited. All rights reserved.

GOYATEK reserves the right to change features or specifications at any time without notice.



Table of Contents

TABLE OF CONTENTS.....	1
ABSTRACT.....	2
GOYATEK TOGGLE RATE CALCULATION FLOW.....	4
GOYATEK TOGGLE RATE SOFTWARE.....	5
THE EXAMPLE OF TOGGLE RATE REPORT.....	8



Abstract

A toggle test is an application that monitors a set of data elements in a design to determine if these elements have made transitions through a predefined set of states during simulation. When a data element has made a transition through this predefined set of states, it has toggled. You can run the toggle test application during a regular logic simulation. Goyatek selects the Cadence NC-Verilog to be a golden sign-off simulator. The resulting information gives you an upper bound (the highest percentage number you can count on) on fault detection/fault coverage for a particular set of vectors. You can use the information on nets that do not toggle to determine where there is little activity and pinpoint which regions of your design are not being exercised by the stimulus.

The fault coverage information from a toggle test is not deterministic. However, the toggle test application can be used to establish the effectiveness of a set of vectors in less time than it takes for a full fault simulation.

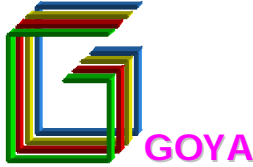
By using PLI (Programming Language Interface), we can access internal database in Verilog. For accessing toggle data, CAD section of Goyatek creates PLI routine to dynamically link in Cadence NC-Verilog and Verilog-XL simulation. Designers can use it easily in task style to report toggle rate.

PLI can only be used with ASIC cell-based designs where the lower level modules are identified as cells within the Verilog description. Its basic functionality is to monitor the toggle information at the inputs and outputs of each cell instance. A toggle is defined as an occurrence of 1 to 0 and 0 to 1 transitions, where a transition from an x to 1 is equivalent to a transition from 0 to 1 and a transition from x to 0 is equivalent to a transition from 1 to 0.

We can use toggle rate report to understand how to improve toggle nets count which make sure the toggle quality of design pattern is high enough and there are fewer redundant logic in design.

Besides, toggle rate can provide power consuming. We can use SAIF or DP format in SYNOPSIS PrimePower to calculate power dissipation. It takes internal power, switching power and leakage power at the same time. Beside this it calculates power dissipation much faster than SPICE or PowerMill.

PrimePower employs the following equation to calculate power dissipation that is based on gate level simulation.



$$P_{\text{total}} = P_{\text{internal}} + P_{\text{switching}} + P_{\text{leakage}}$$

P_{internal}:

Internal power includes short circuit power and internal cell charging and discharging power. It uses lookup-table in terms of input ramps and output loading.

$$P_{\text{Internal}} = \sum_{\forall \text{ cells}(i)} (E_i \times TR_i)$$

$$E_i = f(C_{\text{load}}, T_{\text{tran}})$$

P_{Internal} : Total internal power of the cell

E_i : Internal energy as a function of input transitions and output loads

TR_i : Toggle rate or number of toggles per unit time for a net or a cell input or output

C_{load} : Total output capacitive loads

T_{tran} : Input transition time

P_{switching}:

PrimePower uses the following equation to calculate switching power:

$$P_{\text{switching}} = \frac{V_{\text{dd}}^2}{2} \sum_{\forall \text{ nets}(i)} (C_{\text{load}(i)} \times TR_i)$$

TR_i : Toggle rate

$C_{\text{load}(i)}$: Total capacitive loads

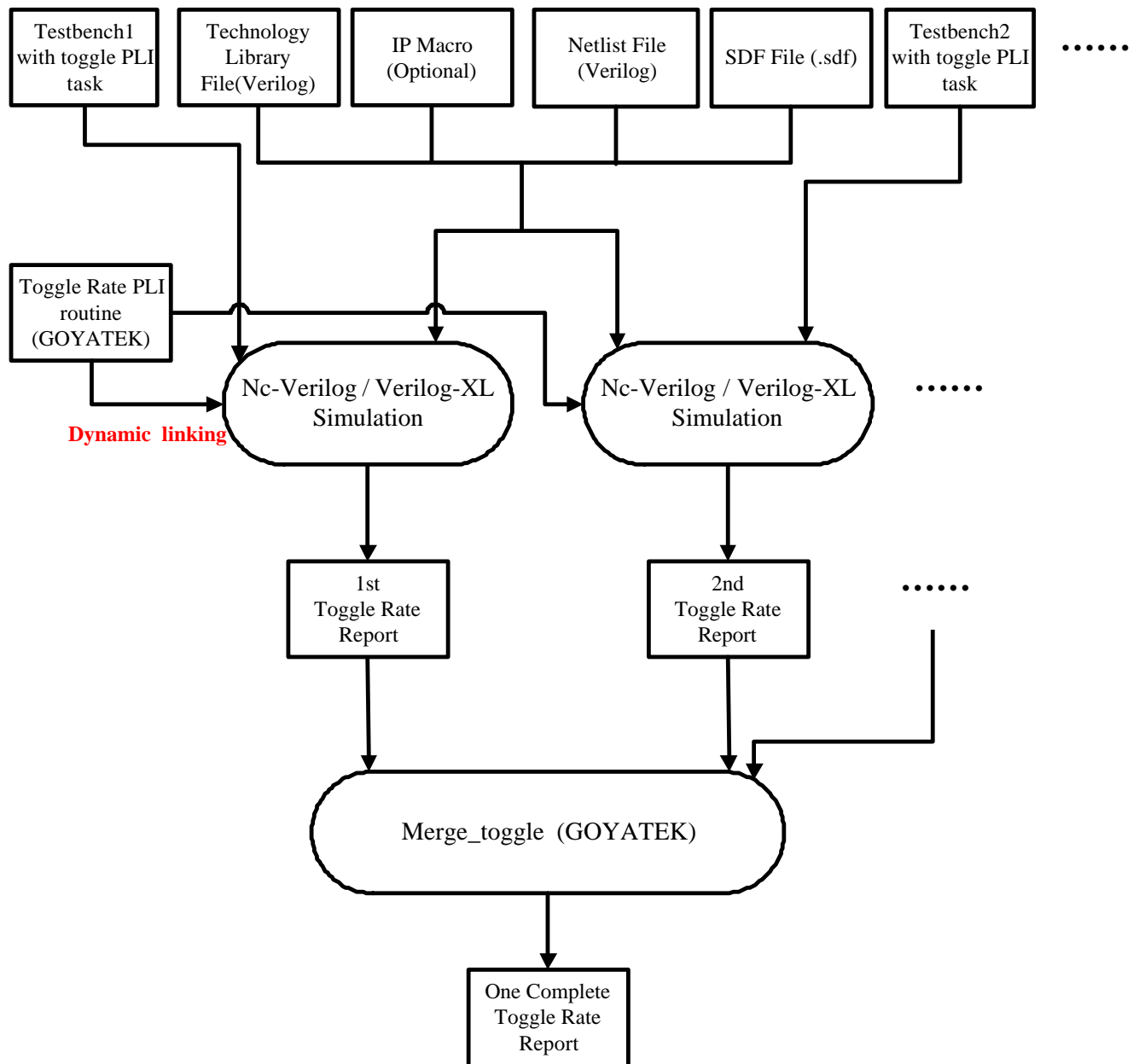
V_{dd} : Current Voltage

P_{leakage}:

Leakage power is much smaller than internal power and switching power. Therefore, we distribute a small constant value (about 1pw) to represent leakage power for cells in current version. In TSMC libraries, leakage power is about 10pW to 0.1pW per core cells.

$$P_{\text{leakage}} = \sum_{\forall \text{ cells}(i)} P_{\text{cellleakage}}$$

Goyatek Toggle Rate Calculation Flow





Goyatek Toggle Rate Software

We provide software for our customers to calculate their toggle rate of design. The first software is the PLI routine dynamic shared library for Cadence NC-Verilog or Verilog XL as following:

- **GOYA_tg_Linux.so**: Linux version
 - **GOYA_tg_Sun.so**: Solaris version
- The second software can merge every toggle report into one complete toggle rate report
- **merge_toggle_Linux**: Linux version
 - **merge_toggle_Sun**: Solaris version

Assumption

If the customer wants to use this software, they need the following database to complete it.

- The Verilog gate level netlist
- The Verilog testbench
- The Verilog standard cell simulation library
- If designers use timing back annotation in simulation, please attach SDF files.
- If there are memories or IP in design, please attach the memories and IP model.

The Goyatek provide PLI routine in GOYA_tg_Linux.so and GOYA_tg_Sun.so

- **\$toggle_count(instance1, instance2,)**
The \$toggle_count PLI routine uses to setup the scope for toggle rate calculation. The user can select the different module instance for its hierarchical design to calculate the different module instance toggle rate. The default usage is to setup the top instance name in \$toggle_count(top_instance_name).
For example, if the design top instance name is b_0, the user can specify as following:
\$toggle_count(b_0)
- **\$toggle_report("file_name", "file_format", time_unit, "strip_name")**
The \$toggle_report PLI routine uses to generate the toggle report that calculate by Goyatek



toggle rate PLI routine.

- **file_name**: the report file name
- **file_format**: we provide the Synopsys SAIF, Synopsys DP and Text format as following:
 1. toggle: provides toggle rate report.
 2. dp: provides the Synopsys PrimePower power calculation
 3. saif: provide saif (switch activity interchange format) report for PrimePower power calculation.

Before you read toggle.saif in PrimePower, it must change instance name to design top module name in saif file.

For example: if the top instance name is “b_0” and top module name is “top_module”, change “b_0” to “top_module” in saif file.

- **time_unit**: default is 1e-9
- **strip_name**: strip the hierarchical name to sub-module design name. The default is none.

For example: The testbench module name is stimulus and the top instance name is b_0.

If the strip_name is “stimulus.b_0”, the hierarchical name stimulus.b_0.n825 will change to n825

For saif format, please keep the module name you want to calculate.

For example: The testbench module name is stimulus and the top instance name is b_0.

Please set the strip_name to “**stimulus**” so that Synopsys PrimePower can read the SAIF correctly.

Usage example in design test bench

// start toggle count at instance stimulus.b_0 and specify the \$toggle_count PLI task before first simulation cycle

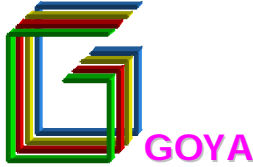
```
$toggle_count(stimulus.b_0);
```

// Please specify the toggle report PLI task at final simulation cycle

```
$toggle_report("toggle.rpt","toggle", 1e-9,"stimulus.b_0" );
```

```
$toggle_report("toggle.dp","dp", 1e-9, "stimulus.b_0");
```

```
$toggle_report("toggle.saif","saif", 1e-9, "stimulus");
```



Usage example in verilog simulation

```
ncverilog +loadpli1=GOYA_tg_Linux.so:bootstrap +access+rwc top.v test.v -v lib.v
```

If the machine operation system is the Solaris, please change the

GOYA_tg_Linux.so:bootstrap to GOYA_tg_Sun.so:bootstrap

Note: If you can't use the routine , please check if your tool environment is right, as follows:

```
Setenv LD_LIBRARY_PATH ./:$CDS_INST_DIR/tools/lib:$CDS_INST_DIR/inca/lib
```

The usage for merge toggle rate

The merge toggle rate software is to merge many testbenches toggle rate report to be a final toggle rate report. It can automatically to calculate the toggle and un-toggle netlist for each testbench. Then it will merge these toggle information to generate the final result.

```
merge_toggle_Linux <toggle_report1> <toggle_report2>.....<output_merge_report>
```

For example, if the design has three toggle report (1.rpt, 2.rpt and 3.rpt), so execute the merge toggle program as following:

```
merge_toggle_Linux 1.rpt 2.rpt 3.rpt out_all.rpt
```

If you want to merge toggle report in different module, please generate toggle report in top module or higher module (including module of other toggle report). Then assign this report to first argument.

Usage example for Synopsys PrimePower to calculate power consumption

```
target_library=~/.025/slow.db"
```

```
link_library="* ~/.025/slow.db"
```

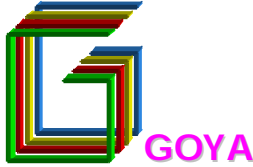
```
read -f verilog top_gate.v
```

```
current_design top
```

```
link
```

```
// Read SAIF or DP, please select either of two
```

```
read_saif -input toggle.saif -instance top
```



include toggle.dp

report_power

The example of toggle Rate Report

```
/*-----  
-- Copyright (c) 2002 by Goyatek Technology Company Ltd.  
-- All Rights Reserved.No part of this publication may be reproduced in  
-- whole or part by any means without the prior written consent.  
-----  
-- LICENSE: Goyatek Technology Company ("GOYATEK") grants you  
-- the non-exclusive right to use the enclosed software  
-- program. You will not use, copy, modify,  
-- display, rent, sell or transfer the Software or any portion  
-- thereof, except as provided in this Agreement.  
-----  
-- Routine: GoyaTek Toggle Count PLI  
-- Created By: Alex Hsu  
-- Created Date: 04/15/2002  
-- Version: 1.0  
-- Simulation time unit is : 10ps */  
/* Synthesis time unit set to: 1ns */  
  
-- Total Simulation Time: 100000  
  
-- TOGGLE COUNT NET INFORMATION  
  
-- NET NAME (0<->1 COUNT) T(0) T(1) T(X)  
  
b0/w_0/SEQ_OP_CMD[0] 1751 91135 8858 7  
ADDR_bist_8_ 35 51762 48212 25  
b0/c_0/s0/n368 10 18019 81970 11  
b0/c_0/s0/n281 16 17881 82109 11  
n825 4533 50552 49421 27  
n937 2 40960 59014 26  
b0/s_0/r349/N257 4 40960 59023 18  
b0/c_0/n487 3 535 99463 1  
n749 1750 9271 90712 17  
b0/s_0/n721 3 99424 565 11  
b0/s_0/n651 2 40960 59024 16
```

8



n876	141	50481	49492	26
n841	1133	50331	49642	26
b0/s_0/n542	8	59025	40959	15
b0/s_0/DONE_0	2	99970	20	11
b0/s_0/n605	793	50474	49514	13
n765	1750	90774	9210	16

 --The following are once-toggle nets

b0/c_0/State_1_	1	190	99804	6
n973	1	120	99880	0
b0/s_0/n719	1	82455	17533	11
b0/c_0/n509	1	99879	120	1
b0/s_0/n728	1	17534	82455	11
b0/c_0/n510	1	99484	510	6
b0/c_0/n519	1	510	99484	6
b0/s_0/CMD[3]	1	505	99484	11
MBS	1	120	99880	0
n972	1	99880	120	0
b0/c_0/s0/n400	1	99484	505	11
b0/c_0/s0/n442	1	99484	505	11
b0/c_0/n513	1	475	99513	12

 -- The following are stack-in-1 nets

b0/s_0/r349/N119	0	0	99987	13
b0/c_0/s0/n246	0	0	99989	11
b0/w_0/n380	0	0	99989	11
b0/s_0/n555	0	0	99954	46
b0/w_0/n383	0	0	99989	11
b0/s_0/n552	0	0	99954	46
b0/w_0/n386	0	0	99989	11
b0/c_0/n514	0	0	99994	6
b0/c_0/s0/n443	0	0	99988	12
b0/s_0/n549	0	0	99953	47
b0/s_0/n682	0	0	99988	12
b0/s_0/r349/N274	0	0	99988	12
b0/s_0/r349/N100	0	0	99988	12
b0/c_0/s0/n244	0	0	99989	11
b0/s_0/n731	0	0	99988	1

 -- The following are stack-in-0 nets

n992	0	99990	0	10
------	---	-------	---	----



Goyatek Technology Inc.

b0/w_0/SEQ_BG[1]	0	99989	0	11
b0/s_0/n712	0	99989	0	11
b0/c_0/n504	0	99994	0	6
b0/c_0/n507	0	99994	0	6
b0/OEB_0	0	99999	0	1
b0/c_0/s0/MSO_int	0	99973	0	27
b0/s_0/r349/N285	0	99988	0	12
b0/s_0/BG[1]	0	99989	0	11
b0/c_0/s0/n251	0	99989	0	11
b0/s_0/BG[0]	0	99990	0	10
b0/c_0/s0/n247	0	99989	0	11
b0/s_0/r349/N284	0	99988	0	12
b0/w_0/OEB	0	99999	0	12

 -- The following are stack-in-X nets

n921	0	0	0	100000
n822	0	0	0	100000
n889	0	0	0	100000
n953	0	0	0	100000
CSB	0	0	0	100000
n526	0	0	0	100000
n775	0	0	0	100000
n867	0	0	0	100000
n899	0	0	0	100000
n623	0	0	0	100000
n944	0	0	0	100000
ADDR0	0	0	0	100000
ADDR1	0	0	0	100000
ADDR2	0	0	0	100000
ADDR3	0	0	0	100000
ADDR4	0	0	0	100000
ADDR5	0	0	0	100000
ADDR6	0	0	0	100000
ADDR7	0	0	0	100000
ADDR8	0	0	0	100000
ADDR9	0	0	0	100000

 -- Summary

 -- TOTAL NETS : 1085

-- 0<->1 TOGGLE NETS : 921



-- ONCE-TOGGLE NETS : 13
-- STUCK-IN-1 NETS: 37
-- STUCK-IN-0 NETS: 47
-- STUCK-IN-X NETS: 67
-- 0<->1 TOGGLE RATE : 84.88%