Data Mining Software Tools and Methodologies for VLSI Design

QING ZHU
Electrical Engineering Department
International Technological University
355 W. San Fernando St., San Jose, CA 95113
USA

Abstract: - Modern VLSI design process employs numerous EDA tools to produce large-size report files. It is time consuming as well as the possibility to miss some violations from EDA report files. We developed two data mining software tools targeting for the timing, noise and power grid analysis report files in one high-performance SOC design project. The software extracts, sorts and displays the key information from EDA report files into HTML forms, which can be viewed by designers using the web browser. Organization of EDA report files, configure files, program execution flows and output HTML forms are described for the developed data mining software tools.

Key-Words: - VLSI design, SOC, Data mining, Timing, Noise, IR drop, Power, EM, HTML, EDA, Tapeout.

Received: April 5, 2021. Revised: January 3, 2022. Accepted: January 17, 2022. Published: March 1, 2022.

1 Introduction

The complexity of VLSI design keeps growing due to advanced process technology, high speed performance, mix-signal and low-power features [1]. EDA tools have been applied in modern VLSI design methodology, which improves the productivity and design quality. EDA tools output significant number of report files, containing thousands of lines in text files and requiring designers to review and digest in order to capture design issues. This paper presents two data mining software tools to help the productivity of reviewing EDA report files. These software tools were developed in one SOC (system-on-chip) design project based on report files from Cadence's static timing tool (Tempus) and power grid simulation tool (Voltus) [2], [3]. One publication related to our work was a web-based software tool, which was developed based on Synopsys Primetime report files [4]. MySQL and Apache servers were used to save and extract report files [5], [6]. Our work differs from this previous work by using text files only to save intermediate and output tables. It has the simplicity with no need to install Apache servers.

The paper is organized in 5 sections. Section 2 describes the data mining software tool for timing and noise report files. Section 3 describes the data mining software tool for power grid simulation report files. Section 4 shows performance results in the SOC project. Section 5 gives conclusions.

2 Data Mining Tool for Timing and Noise Report Files

EDA tools and generated report files from the design database were organized in a tree directory as shown in Figure 1. The root of the tree directory is the specific EDA analysis category (i.e. STA) followed by the list of macro blocks in the chip. Design revisions are listed after each macro block. At each design revision, a list of analysis scenarios is followed based on multiple analysis corners and logic modes of the design. Leaf nodes in the tree directory are the list of timing and noise analysis report files, which are described as follows.

- *clock_skew.rpt: the report file of clock groups and clock skew numbers in the design.*
- *<block>.<group>.setup.rpt: the report file of setup violations for each clock group in the design.*
- *<block>.<group>.hold.rpt: the report file of hold violations for each clock group in the design.*
- *design.sdc: the timing constraints file in the design.*
- *constraint_maxtran.rpt: the report file of transition time violations in the design.*
- *constraint_maxcap.rpt: the report file of maximum capacitance violations in the design.*

To run the data mining program, the user needs to prepare a configure file with the following key words.

- *Proj*: the project name.
- *Root*: the root directory to access timing and noise analysis result files.
- *Block*: the list of block names in the design.
- *Starev*: the list of design revisions.
- *Bins*: the list of windows to report timing violations.

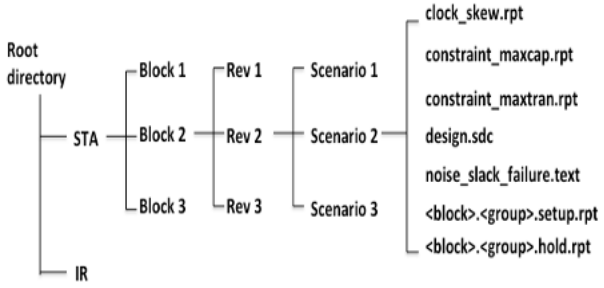


Fig. 1: Tree Directory of Timing/Noise Report Files

Then the user can submit the following command to run the program.

>> STA_report <configFile>

Figure 2 shows the high-level flow to execute the program, which extracts key statistics from timing and noise report files. The program outputs a set of HTML forms which contain key numbers and statistics in timing, clock and noise analysis report files. Numbers in HTML forms contain hyperlinks to original timing/noise report files. When the user clicks on the number in HTML forms, the corresponding original report file is popped up for reviewing. Using the data mining tool, it eliminates the manual step to open and review original report files. Productivity and accuracy are improved; it avoids possible missing of timing/noise violations in the design.

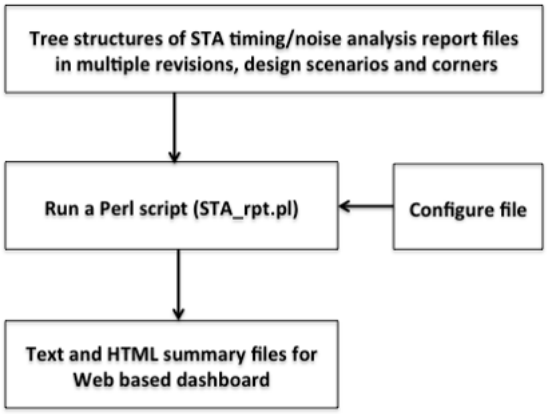


Fig. 2: High-Level Flow to Execute Data Mining Program.

Here are detailed steps to execute the program. The following detailed steps are shown in Figure 3.

(1) Process the configuration file.
   - Check the quality of parameters specified in the configure file.
   - Assign parameters to the program.
(2) Generate a set of text files, which contain summary tables with key information from timing/noise report files.
(3) Convert output text files to HTML forms, which contain hyperlinks to original timing/noise report files.

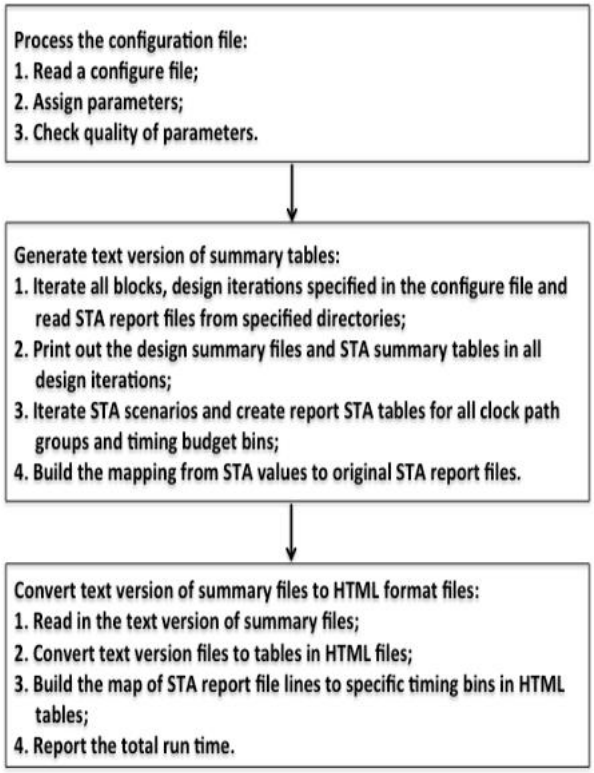


Fig. 3: Detailed Steps to Execute Data Mining Program.

The user can use the web browser to view the generated HTML forms from the data mining software. Here is the command to launch Mozilla's Firefox to view the top-level web page, which is generated from the data mining program.

>> firefox runDir/HTML/<project.design_run_summary.html>

The top-level web page will contain the list of design revisions and block names that were specified in the configure file. When the user clicks on a design revision name in the top-level web page, a second-level web page is popped up, which includes the following information.

- *Run directory*: static timing analysis run directory.
- *Design name*: top block name.

- *Run iteration*: design revision name.
- *A list of statistics tables*: timing and noise summary numbers and violations for analysis scenarios specified in the configure file.

The second-level HTML forms contain the following timing and noise violations for every analysis scenario.

- *Reg2Reg setup violations (WNS/FEP):* the worst timing slacks and number of violated paths for setup constraints between adjacent registers.
- *Reg2Reg hold violations (WNS/FEP):* the worst timing slacks and number of violated paths for hold constraints between adjacent registers.
- *IO2Reg setup violations (WNS/FEP):* the worst timing slacks and violated paths of setup constraints from IOs to adjacent registers.
- *IO2Reg hold violations (WNS/FEP):* the worst timing slacks and violation paths of hold constraints from IOs to adjacent registers.
- *Data transition violations (WNS/FEP):* the maximum transition violations of data paths.
- *Clock transition violations (WNS/FEP):* the maximum transition violations of clock paths.
- *Capacitance load violations (WNS/FEP):* the maximum capacitance load violations of data and clock paths.
- *Noise violations (WMV/FNN):* the noise threshold violations.

Each analysis scenario shown in the second-level HTML forms is hyper-linked to the third-level HTML forms with the following information.

- *Run directory*: static timing analysis run directory.
- *Design name*: top block name.
- *Run iteration*: design iteration name.
- *Scenario*: analysis scenario name.
- *Timing violations distribution*: the number of violated paths is listed based on the order of timing violation bins for every clock group.
- *Clock skew summary*: a list of clock nets in the design, clock skew numbers and setup/hold uncertainty constraints.

Figure 4 is the snapshot of third-level HTML forms, which contain timing or noise violations in the design. In the report files, WNS stands for the worst-case negative slacks and FEP stands for the number of timing paths. In addition, timing path groups are associated with a set of timing bins to sort timing violations in the increasing ranges of WNS violations in static timing paths. The user can give the specification of timing bins in the configure file in the following commands.

- *Setup bins (negative slacks)*: [0-100ps], [100-200ps], [200-400ps], [400-600ps], [600-1000ps], [>1000ps].
- *Hold bins (negative slacks)*: [0-50ps], [50-100ps], [100-200ps], [200-300ps], [300-400ps].
-

Fig. 4: Third-Level HTML Forms

We have developed a set of advanced features for the data mining tool during the design project of one high performance SOC (system-on-chip). The advanced features of the tool are summarized as follows.

(1) User can specify the order of scenario names in the configure file. The software will sort out timing analysis scenario names in output HTML forms. The default order of scenario names in output HTML forms are based on the alphabetical order.

(2) User can specify timing accuracy thresholds in the configure file. The software will remove insignificant timing violations less than specified thresholds. In this way the user can review significant timing violations, which are larger than thresholds, in output summary tables.

(3) User can waive specific timing/noise errors in the configure file. The software highlights waived errors in the yellow color in output summary tables. The waved timing/noise errors do not need to be fixed but still show in the summary tables as the reference.
(4) User can waive transition and capacitance load errors for specified instance names (i.e.: pads), which are specified in the configure file. Those errors will not be included in output summary tables.

User can specify the command of timing accuracy in the configure file. For example, only timing violation slacks less than -20ps will be shown in output HTML forms. This can be done in the following command which could be specified in the configure file.

staacc 20

The following command in the configure file specifies specific columns or rows of timing/noise errors to be marked in the yellow colour, which are waived in the design after evaluated by the team:

stawaive _tt_*_Ctype_* C1 C6 C8

Figure 5 shows the summary tables with three columns marked in yellow colours to indicate waived errors based on the above command.

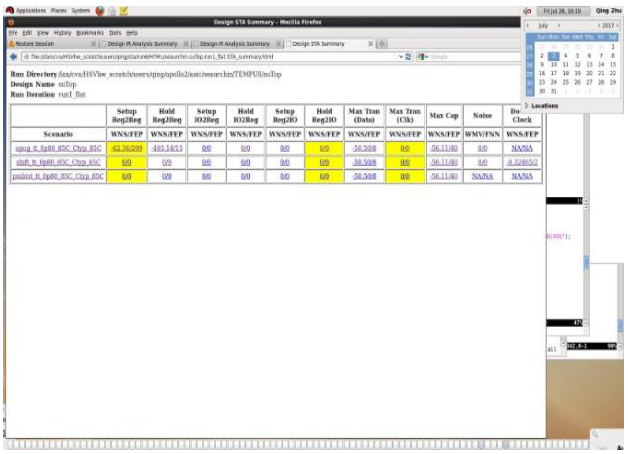


Fig. 5: Three Columns in Yellow Color to Indicate Waived Errors

Another command, defined in the configure file, allows the user to specify the waiving patterns for the maximum transition and capacitance load errors. Here is the command format:

waivetrancap tx_m_ln PAD_N

The details to implement the above command are explained as follows. The software copies the original "constraint_maxtran.rpt" file that contains the violations of maximum transition time or capacitance load constraints to a new file. Then the new file is modified to remove all of timing violations and instance names that match to specified string patterns. Figure 6(a) shows the original report file for transition time violations in one design case. Figure 6(b) shows the modified report file, which has removed all of instance names that contain the "PAD_N" string. The modified report file is then used to report transition time and capacitance load violations in output HTML forms.

Check type : max_transition

Pin Name	Required	Actual	Slack
I1/PAD_P	4.0000	54.5000	-50.5000
I2/PAD_P	4.0000	53.6000	-49.6000
I1/PAD_N	4.0000	52.1000	-48.1000
I2/PAD_N	4.0000	51.9000	-47.9000
I3/PAD_P	4.0000	50.7000	-46.7000
I5/PAD_P	4.0000	50.5000	-46.5000

(a) Original timing report file.

Check type : max_transition

Pin Name	Required	Actual	Slack
I1/PAD_P	4.0000	54.5000	-50.5000
I2/PAD_P	4.0000	53.6000	-49.6000
I3/PAD_P	4.0000	50.7000	-46.7000
I5/PAD_P	4.0000	50.5000	-46.5000

(b) Modified timing report file.

Fig. 6: Specific Instances Matching to Specified String Names Which Are Removed from Output HTML Forms.

The list of scenario names in HTML forms are in general based on the alphabetical order. For example, scenario name s_1 will be arranged before name s_2 or scenario name s_1 will be arranged before t_1 in output HTML forms. The user can specify the order of scenario names in output HTML forms based on the prefix string patterns of scenario names. Other scenario names that do not match to specified prefix string patterns are listed in the alphabetical order afterwards. The following command gives one example to specify prefix string patterns of scenario names in the configure file:

staorder func opc shift

The program will then assign {*func1, func2, opc1, opc2, shift1, shift2*} in the front of output HTML forms. The remaining scenario names are listed in the alphabetical order afterwards in the HTML forms.

3 Data Mining Tool for Power, IR and EM Report Files

Gate delays and circuit stability depend greatly on instantaneous voltages at the on-chip power grid [10], [11]. IR drop analysis calculates the currents and voltages flowing through metal lines and nodes at the power grid [7], [8], [9]. Electromigration (EM) is another concern, which is the gradual displacement of atoms in metal lines when a large amount currents through the power grid. When the process geometry is scaling, with more thin and long interconnects on the chip, combined with high currents, the power grid could have violations of EM rules in the deep submicron process [12]. We developed one data mining software to extract and display key information in HTML forms from the power analysis results including IR drops and EM violations using Cadence Voltus tool [5].

All the report files to the second data mining software are organized in a tree structure as shown in Figure 7. Here is a list of power, IR and EM report files from power grid analysis.

- *design.main.rpt* - Summary file including the power net names and analysis method (static/dynamic).
- *VDD_VSS_div.iv* - IR drop and voltage results on every instance in the layout.
- *VDD_VSS.gif* - IR drop distribution map in the layout.
- *power.rpt* - Report file of power consumption in each run. It can be based on static or dynamic simulation methods.
- *VDD.rj.avg.rpt, VSS.rj.avg.rpt* - Report files of VDD and VSS nets with current limit (EM) violations.
- *VDD.rj.gif, VDD.rj.gif* - EM violations and distribution maps.

Here is the content in the “VSS_VDD_div.iv” file:

```
VERSION "2.0"
CREATION "Wed Mar 29 17:02:40 2017"
CREATOR "kenth"
PROGRAM "VOLTUS"
DIVIDERCHAR "/"
DESIGN "suTop"
UNITS VOLTAGE VOLT 1
INSTANCE_COUNT 4418111
NOMINAL_VOLTAGE 0.88
POWER_NET VDD
GROUND_NET VSS
RP_VALUE IV/DIV
RP_FORMAT BRIEF
RP_INST_LIMIT 2000000000
RP_PIN_NAME FALSE
INSTANCESUPPLY 4418111 VDD 0.88
DIFFERENTIATE
BEGIN
-core__umux/n_185739_center_double_buf_2
0.82457 BUFH_X24N_A9PP96CTUL_C16
- core__umux/n_186074_center_double_buf_2
0.82476 BUFH_X24N_A9PP96CTUL_C16
- core__umux/n_186061_center_double_buf_2
0.82476 BUFH_X24N_A9PP96CTUL_C16
- core__umux/n_185733_center_double_buf_2
0.82477 BUFH_X24N_A9PP96CTUL_C16
- core__umux/n_186058_center_double_buf_2
0.82484 BUFH_X24N_A9PP96CTUL_C16
- core__umux/n_185739_center_double_buf_1
0.82489 BUFH_X24N_A9PP96CTUL_C16
- core__umux/n_186074_center_double_buf_1
0.82491 BUFH_X24N_A9PP96CTUL_C16
- core__umux/n_186073_center_double_buf_2
0.82492 BUFH_X24N_A9PP96CTUL_C16
- core__umux/n_186058_center_double_buf_1
0.82494 BUFH_X24N_A9PP96CTUL_C16
- core__umux/n_185742_center_double_buf_2
0.82501 BUFH_X24N_A9PP96CTUL_C16
```

Here is the content in the power analysis report file (power.rpt):

Total Power

```
----------------------------------------------------------------
Total Internal Power: 4732.05416431 38.2611%
Total Switching Power: 1484.25196958 12.0009%
Total Leakage Power: 6151.48985074 49.7380%
Total Power: 12367.79597228
----------------------------------------------------------------
Group Internal Switching Leakage Total Percentage
Power Power Power Power (%)
----------------------------------------------------------------
Sequential 2773 112.5 1216 4102 33.16
Macro 336.7 6.738 117.4 460.8 3.726
IO 13.91 2.676 1.819 18.41 0.1488
Combinational 1404 723.6 4658 6786 54.87
Clock (Combinational) 140.1 625.3 109.1 874.5 7.071
Clock (Sequential) 64.48 13.47 48.78 126.7 1.025
```

Here is the content in the EM report file (VDD.rj.avg.rpt):

```
# The current or EM limit violations
# EM Reporting Threshold: 0.9
# Total Violations: 276351
# Report format:
# I/Ilimit I(mA) Ilimit(mA) Layer Location ShortLength Width ViaArea Via Fix Width/
```

Current Benefit J(A/cm^2) Jmax(A/cm^2) ResistorID
{X1 Y1 X2 Y2} (um) (um) (nm^2) Cuts Via Direction Factor
4.45691 4.383146e-02 9.834496e-03 VIA2 {6672.21 1712.4 6672.21 1712.4} N/A N/A 2458.62 2 9 N/A 1.00 4.383146e-05 9.834496e-06 46005601
4.45691 4.383146e-02 9.834496e-03 VIA3 {6672.21 1712.4 6672.21 1712.4} N/A N/A 2458.62 2 9 N/A 1.00 4.383146e-05 9.834496e-06 42397244
4.26149 4.190961e-02 9.834496e-03 VIA3 {6672.21 2036.21 6672.21 2036.21} N/A N/A 2458.62 2 9 N/A 1.00 4.190961e-05 9.834496e-06 43036126
4.26149 4.190961e-02 9.834496e-03 VIA2 {6672.21 2036.21 6672.21 2036.21} N/A N/A 2458.62 2 9 N/A 1.00 4.190961e-05 9.834496e-06 46644483

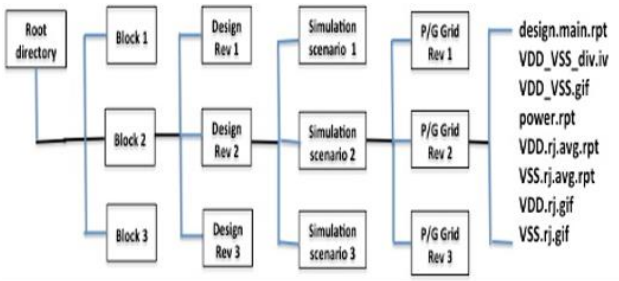


Fig. 7: Tree Structure of Input Power, IR and EM Report Files

The data mining program for power, IR and EM result files uses a configure file with the following keywords.

- *Proj*: the project name.
- *Root*: the root directory to access timing and noise analysis result files.
- *Block*: a list of block names that have IR and EM analysis results.
- *Irrev*: a list of run revisions in IR and EM analysis.

The data mining program is launched using the following command.

>> IR_report.pl <configFile.txt>

The program generates a set of HTML files in the run directory. All the HTML files are linked to one top-level file *<project>.IR_run_summary.html*. The designer can view this top-level file (<project>.IR_run_summary.html) using the web browser (i.e. Mozilla's Firefox) as follows.

>> firefox HTML/ <project.IR_run_summary.html>

Figure 8 shows high-level flow to execute the data mining program. Figure 9 lists detailed steps involved in program execution:

(1) Process the config file and check grammars, etc.
(2) Process IR, power analysis and EM report files and then extract the information to generate a set of summary tables in text files.
(3) Convert text files of summary tables to HTML forms.

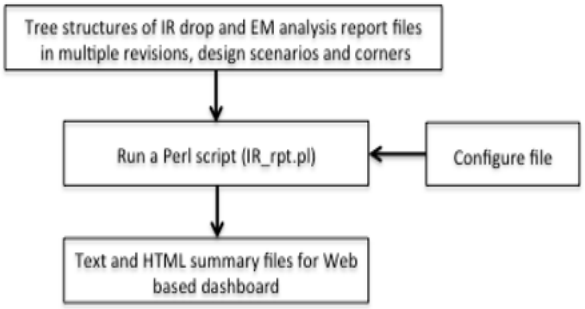


Fig. 8: High-level Flow to Execute Data Mining Program.

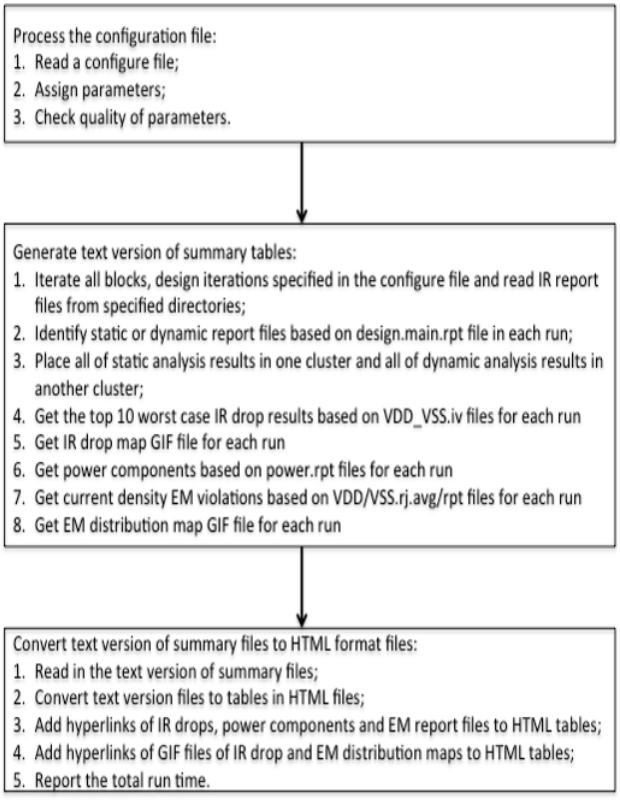


Fig. 9: Steps of Program Execution.

Top-level web page generated from the data mining program shows the following information:

- Run directory: run directory for IR/Power/EM analysis results.
- Design name: top block name.
- Run iteration: IR/Power/EM Analysis.

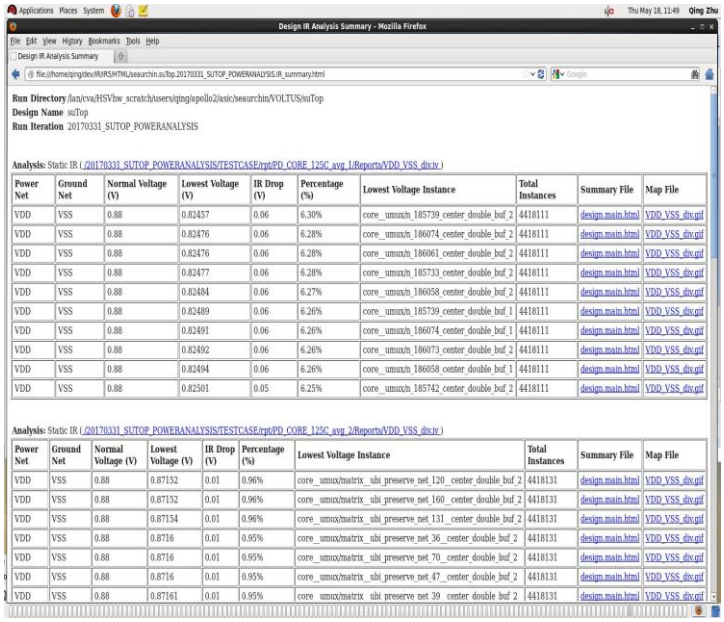
- Iteration_0: first design iteration name (i.e. 20170331_top_IR).
- Iteration_1: second design iteration name (i.e. 20170401_top_IR).

When the user clicks on one design revision in the top web page, it opens the next-level web pages with the following information:

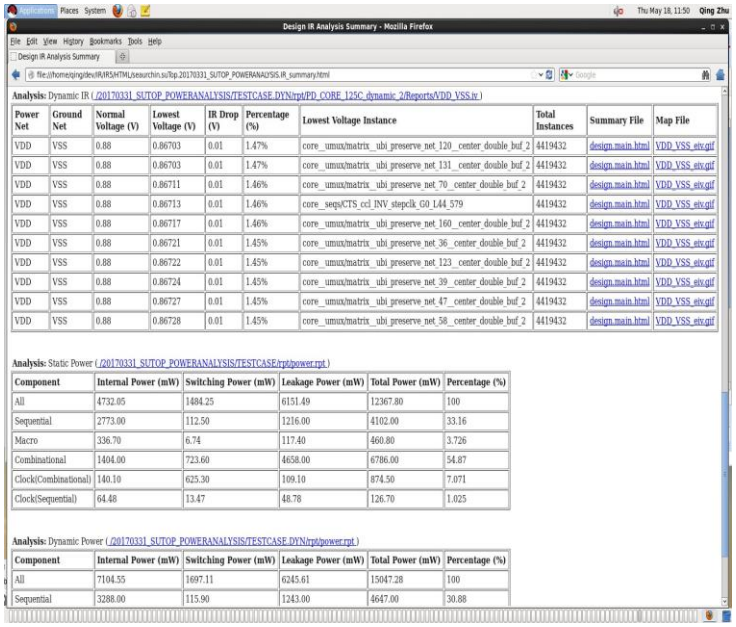
- Design name: top block name.
- Run iteration: design iteration name.
- Static IR tables: power nets, ground nets, normal voltage (V), the 10 lowest voltages (V), IR drop percentages (lowest voltages versus the normal voltage), instances/cells with the lowest voltages, total number of instances in design, IR run summary file, IR drop map file.
- Dynamic IR tables: power nets, ground nets, normal voltage (V), top 10 lowest voltages (V), IR drop percentages (lowest voltages versus the normal voltage), Instances/cells with the lowest voltages, total number of instances in design, IR run summary file, IR drop map file.
- Static power: the internal power, switching power, leakage power and total power for sequential circuits, macro blocks, combinational circuits, clock networks and total components in the block.
- Dynamic power: the internal power, switching power, leakage power and total power for sequential circuits, macro blocks, combinational circuits, clock networks and total components in the block.
- Static EM: net names, top 10 highest current/limit violations, average current, current limit, percentage, violated layer, EM map file.
- Dynamic EM: net names, top 10 highest current/limit violations, average current, current limit, percentage, violated layer, EM map file.

Figure 10 shows snapshots of summary tables in the second level web pages. Figure 10(a) is the snapshoot of static IR drop summary tables. Figure 10(b) is the snapshoot of dynamic IR drop summary tables and power dissipations of components. Figure 10(c) is the snapshoot of static EM summary tables. In Figure 10(a), each net name contains the hyperlink to original IR analysis report file and original IR drop map GIF file. In Figure 10(c), each net name contains the hyper link to original EM map GIF file.

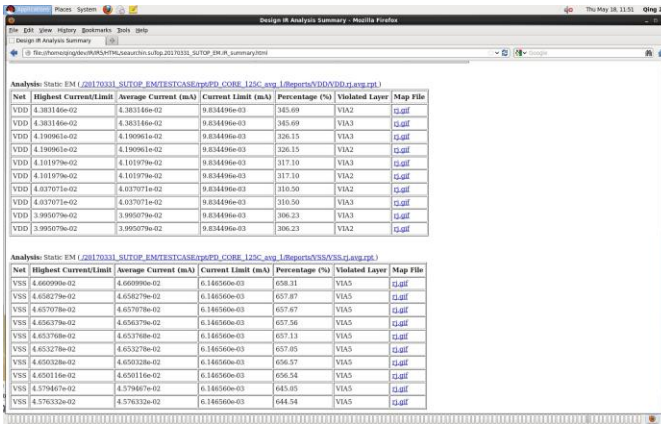
Figure 11 shows IR drop and EM map plots of the design. Figure 11(a) shows dynamic IR drop map of the design. Figure 11(b) shows static IR drop map of the design. Figure 11(c) shows static EM map of the design. In the above IR drop maps, dynamic IR drops are much severer with wide-spread red colors over the die. Power consumption numbers using dynamic analysis strategy are higher compared to results obtained by static IR drop analysis strategy. Dynamic IR drop analysis strategy is therefore necessary to estimate the worst-case voltage drops over the power grid on the die for high-performance chip design.



(a)

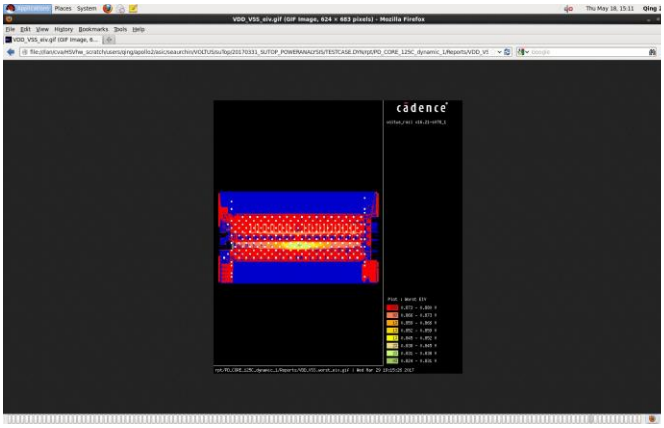


(b)

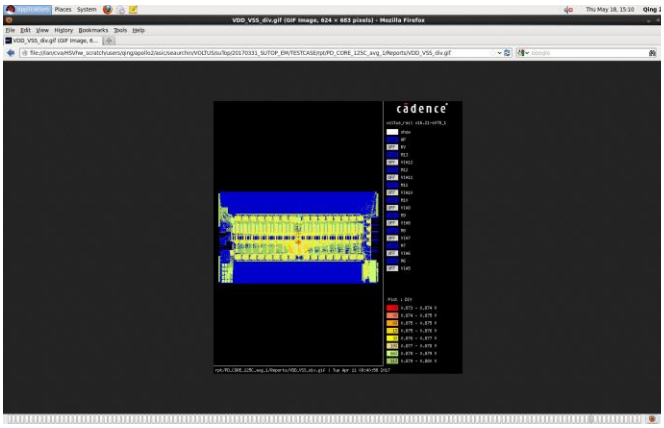


(c)

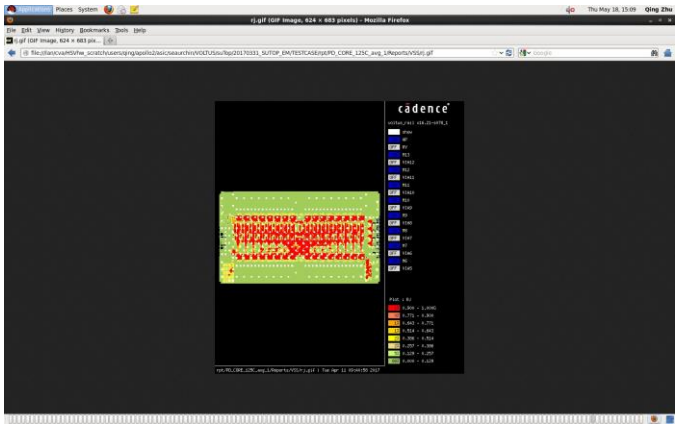
Fig. 10: HTML Forms in Second-Level Web Pages.



(a)



(b)



(c)

Fig. 11: Maps of IR Drops and EM Violations

4 Experimental Results

Data mining programs presented run on RedHat Linux 6.5OS and Intel Xeon® E5-2697 v4 @ 2.3GHZ multi-core CPUs environment. Table I shows the sizes of timing/noise report files in one high-performance SOC design project. Table II shows the measured run time and total output file sizes using the developed data mining tool based on timing/noise report files described in Section 2. Figure 12 shows graphical trends of the running time and output file sizes based on the results in Table II.

Table I. *Sizes of Timing/Noise Report Files in SOC Design Project.*

FILE NAME	NUMBER OF FILES	FILE SIZE (BYTES)
CLOCK_SKEW.RPT	1	4M
LAYFILE: LAYOUT DESIGN FILE	1	20K
CONSTRAINT_MAXTRAN.RPT	1	12M

Table II. *Run Time and Total File Sizes Using Timing and Noise Data Mining Program.*

# ANALYSIS SCENARIOS	RUN TIME (SEC)	TOTAL FILE SIZES (B)
3	113	6G
6	252	13G
12	503	25G
24	1074	49G
48	2004	98G

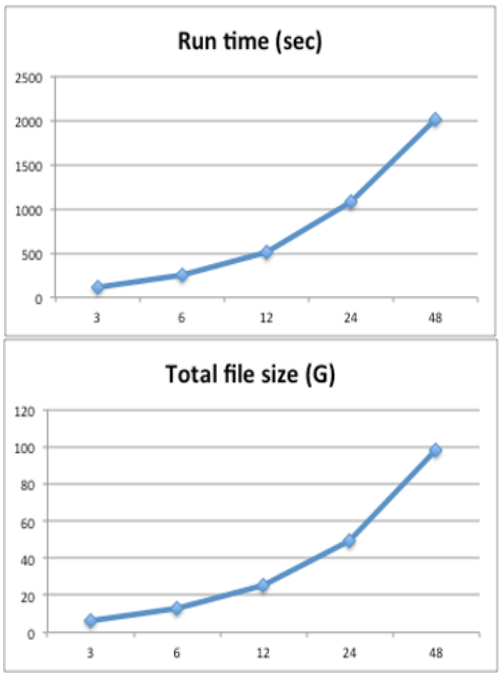


Fig. 12: *Run Time and Total File Size Using Timing and Noise Data Mining Program*

Table III shows the sizes of power, IR and EM report files from one SOC design project. Table IV shows the running time and output files of the data mining software based on the number of design revisions in the project. Figure 13 shows graphical trends of the running time and total sizes of output files based on the data in Table III and Table IV. When the sizes of report files increase, the running time of the data mining program is increased slowly as shown in Figure 13. Note that we used the technique of look-up tables extensively in data mining programs to speed up the running time for large-size report files.

Table III: *Number of Files and Disk Sizes in One SOC Project.*

Files	# of Files	Total Disk Size
VDD_VSS.iv	1	385M
Power.rpt	1	22K
VDD/VSS.rj.avg.rpt	2	122M

Table IV: *Run Time and Output File Sizes of Power, IR and EM Data Mining Program.*

# Design revisions	Run Time (Sec)	Output File Sizes (K)
2	2	108
4	2	196
8	3	336
16	6	640
32	11	1300

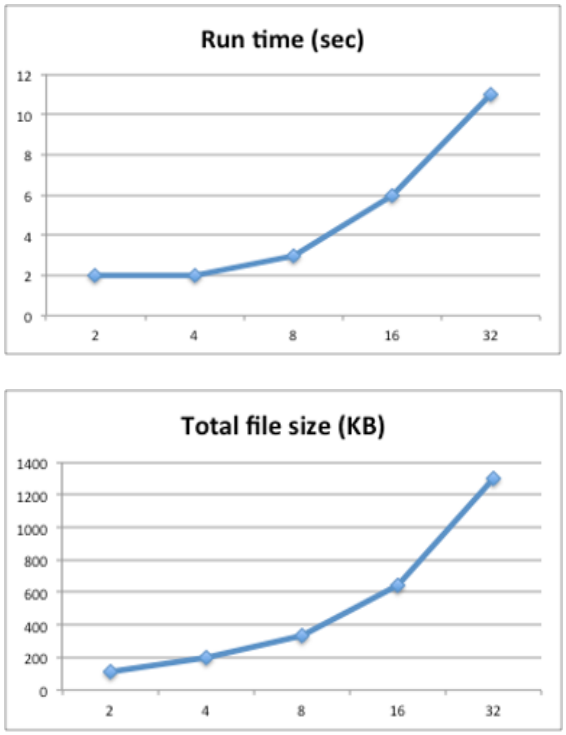


Fig. 13: Run Time and Total File Sizes Using Power, IR and EM Data Mining Program

5 Conclusion

This paper describes two data mining software tools that extract and display key design information from EDA report files in the VLSI design. The first data mining program extracts key design information and displays in HTML forms based on timing and noise analysis report files from Cadence statical timing analysis tool Tempus [2]. The second data mining program extracts key information and displays in HTML forms based on power, IR and EM analysis report files from Cadence power grid analysis tool Voltus [3]. A tree structure of EDA report files is used as the input to the software. A configure file is specified by the user, which specifies the following information for the timing/noise data mining software: (a) project name, (b) root directory of report files, (c) block name, (d) design revisions, and (d) timing bins. The data in output HTML forms are also linked to original EDA report files, so users could review EDA report files in details. Therefore, the developed data mining tools could eliminate the need for manual review of EDA report files especially in the complex design. We have applied the described data mining programs in one high-performance SOC design [13], [14]. The programs and methodologies can be applied to other VLSI design projects.

Acknowledgment: Yogesh Chandra and Syed Ahmed provided valuable comments on the development of software and methodologies described in this paper.

References:

[1] M. Lapedus, “10nm Versus 7nm”, https://www.semiengineering.com, 2016.
[2] C. Edwards, "Cadence tackles timing signoff with Tempus", https://www.techdesignforums.com, 2013.
[3] R. Goering, "Voltus-Fi custom power integrity solution: electromigration and IR drop at the transistor level", https://community.cadence.com, 2014.
[4] F. M. Jumaah, S. Baskara, R. M. Sidek, F. Z, Rokhani, "PrimeTime web-based report analyzer (PTWRA) tool", Proc. of 6th Asia Symposium on Quality Electronic Design (ASQED), 2015.
[5] “MySQL for absolute beginners”, http://www.elated.com, 2018.
[6] R. Nixon, "Learning PHP, MySQL & JavaScript: with jQuery, CSS & HTML5 (learning PHP, Mysql, Javascript, Css & Html5)", ISBN-10: 1491918667, www.amazon.com, 2015.
[7] “IR drop analysis - II”, https://vlsi-soc.blogspot.com, 2019.
[8] G. Shomron, "VCD-based RTL power estimation", https://gilshm.github.io/vcd-power, 2020.
[9] C. Liu, Y. Wu, Y. Huang, "Effect of IR-drop on path delay testing using statistical analysis", Proc. of Asian Test Symposium, 2007.
[10] D. Payne, "Full chip IR drop analysis using distributed multi-processing", https://www.semiwiki.com, 2013.
[11] G. Rangarajan, J. Deng, "Addressing signal electromigration (EM) in today's complex digital designs", https://www.edn.com, 2013.
[12] M. Santarini and R. GoeringH, "Web-based EDA tools find limited acceptance", http://www.eetimes.com, 2001.
[13] Q. Zhu, "Data mining tool of timing analysis data in complex VLSI design", International Conference on Data Mining and Knowledge Discovery (DMKD 2019), 2019.
[14] Q. Zhu, “Data mining and web-based reporting software for power, IR drop and EM analysis in VLSI design”, International Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions) (ICTUS), 2017.

Conflicts of Interest

The author(s) declare no potential conflicts of interest concerning the research, authorship, or publication of this article.

Contribution of individual authors to the creation of a scientific article (ghostwriting policy)

The author(s) contributed in the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of funding for research presented in a scientific article or scientific article itself

No funding was received for conducting this study.

Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)