Digital Power Estimation Flow Combining Academic and Industrial Tools

R. Mehrotra, T. English, K.L. Man, E. Popovici and M.P. Schellekens
Centre for Efficiency-Oriented Languages (CEOL) and
Department of Microelectronic Engineering,
University College Cork (UCC), Cork, Ireland
khushi.rush@gmail.com, tom.english@ue.ucc.ie, kalok2006@gmail.com,
e.popovici@ucc.ie and m.schellekens@cs.ucc.ie

Abstract

We analyse how a power estimation and optimisation design flow from the academic domain may be integrated into an industrial design flow and how this integration creates an open-source environment in which new techniques can be developed and compared. In this integrated design flow, new tools are introduced and several practical issues such as technology libraries and the use of simulation data to derive realistic power estimation results are addressed. Experimental results (enabling power saving of over 40%) demonstrate the applicability and effectiveness of the integration of the two design flows. The integrated design flow provides a solid platform for future research in logic synthesis, power estimation and power optimisation.

1 Introduction

Over the years, a number of academic EDA tools [9, 2, 13, 7, 12, 14, 8] have been proposed in the literature. These open source tools provide a programming environment and a solid platform for research in logic synthesis, power estimation and power optimisation as well as bringing new developers into such a platform. Although many techniques applied in the tools demonstrate robustness and optimality in EDA, the tools have not found application in industry and can only be considered theoretical frameworks. Practical reasons for this include:

- only a restricted subset of language constructs in Verilog and VHDL are supported for input and output;
- the tools do not support constructs from more modern HDLs such as Verilog-2001 and SystemVerilog;
- basic random-pattern simulation is supported, but complex testbenches with user-supplied test vectors and assertions are not;
- the academic tools only include generic technology libraries;
- data exchange between academic tools is complicated by occasional file format incompatibilities;

We have surveyed several academic tools used for logic synthesis, power estimation and power optimisation with the aim of building a better understanding of the theoretical foundations behind such tools as well as motivating research towards an integration of academic tools into an industrial design flow.

In this paper, we propose a design flow consisting of several well-known academic tools which can be integrated into an industrial design flow. This integration may be beneficial to researchers in logic synthesis, power estimation and power optimisation. The aims of the integration are as follows:

- to relate academic tools to an industrial design flow;
- to bridge gaps between the academic- and industrial design flows (new tools have been developed to achieve this);
- to make the academic tools easier to deploy and to use by connecting them together with a set of customisable Unix shell scripts;
- to create an open-source, academic tool flow within which new techniques can be developed and compared;
- to use data from the industrial design flow (such as event-driven simulation results) to support the development of logic synthesis, power estimation and power optimisation techniques in the academic domain.

To the best of our understanding, this is the first article which reports the integration of an academic design flow to an industrial design flow.

Structure The remainder of the paper is organised as follows. A standard industrial design flow and the proposed academic design flow are presented in Sections 2 and 3, respectively. Section 4 outlines an integrated design flow and the motivation for its development. Experimental results (enabling power saving of over 40%) are provided in Section 5 to demonstrate the applicability and effectiveness of such an integrated design flow. Finally, concluding remarks are made in Section 6, along with proposals for future work.

2 Industrial Design Flow

Our industrial design flow is presented in Figure 1.

In this flow, RTL simulation is performed using VCS [10], a commercial tool from EDA vendor Synopsys. VCS

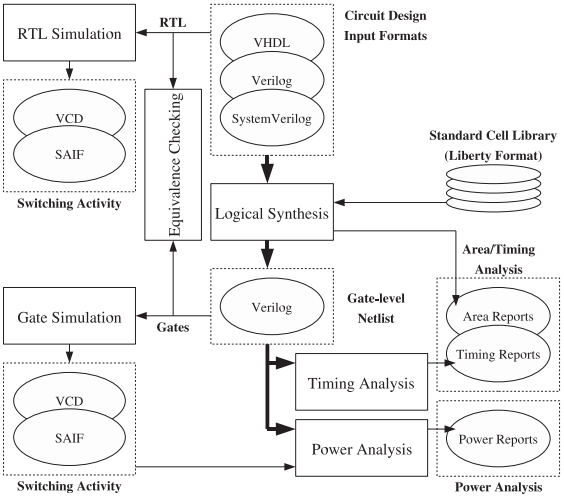

Figure 1. Industrial Design Flow

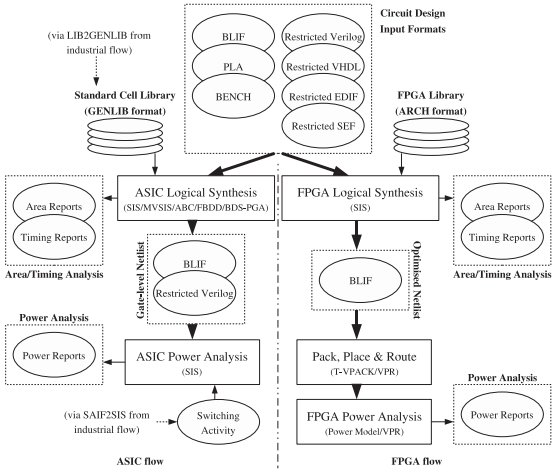

Figure 2. Academic Design Flow

(like some other commercial simulators) supports Verilog and VHDL as well as more recent IEEE-standard HDLs such as Verilog-2001 and SystemVerilog. The logic description is constrained and then synthesised to a gate-level netlist targeting our chosen standard cell library in Synopsys Design Compiler [10]. Formal verification is performed in Synopsys Formality [10], ensuring equivalence between RTL and the resulting gate-level netlist. Reports generated by the synthesis tool detail the silicon area consumed by the logic design. Timing reports are generated using Synopsys PrimeTime [10] determining whether the netlist meets constraints laid down prior to synthesis. PrimeTime also generates SDF [6] containing delay information for annotation onto the netlist during gate-level simulation.

The primary goal of constructing this flow was to perform power analysis on the pre-placement netlist using commercial tools. Switching probability information is recorded by VCS during simulation as SAIF [11]. Time-domain value-change data is recorded as VCD [5]. SAIF and VCD are used during power analysis to obtain realistic power figures for the simulated scenario. Two power analysis flows are used:

1. SAIF switching probabilities from RTL simulation are annotated onto the netlist in Synopsys Power Compiler [10]. The tool propagates switching activity through the netlist and reports average power for the simulated scenario. (SAIF from gate-level simulation can also be used, if available, for more accurate results at the expense of slower turnaround.)
2. The netlist is simulated in Synopsys PrimeTime PX [10] using value-change data from gate-level VCD. Using VCD allows both average- and instantaneous (peak) power reports to be generated, along with time-domain power waveforms from the simulation.

3 Academic Design Flow

A typical academic logic synthesis tool first constructs a two-level or multi-level Boolean network corresponding to the RTL description of a design. This network is then optimised using various technology-independent techniques. Finally, technology mapping transforms the technology-independent circuit into a network of gates in a given technology. Mapping is constrained by several factors including the available gates in the technology library, the drive sizes for each gate and the delay, power and area characteristics of each gate.

Binary Decision Diagrams (BDDs) [3] have become a very important representation for boolean functions in VLSI/CAD and they are widely used in many applications like logic synthesis and formal verification. *Berkeley Logic Interchange Format* (BLIF) [9] is the most popular format used in academic tools for describing logic-level hierarchical circuits in textual form. A circuit described in BLIF can be viewed as a directed graph of combinational logic nodes and sequential elements. Due to this, it is reasonably easy to transform a circuit described in BLIF into the corresponding BDD. This has already been automated in many academic EDA tools (e.g. SIS [9], VIS [13], MVSIS [2] and ABC [7]) and many successful optimisation techniques have also been developed in such academic EDA tools. Furthermore, several (bi-directional) translators between BLIF and HDL/netlists are publicly available.

3.1 Structural information

Figure 2 shows the structure of our proposed academic design flow. The flow is comprised of three types of tool– i) HDL translators, ii) logic synthesis and verification tools and iii) power analysis tools. The tools are:

- HDL translators EDIF2BLIF [4], BLIF2VHDL [1] and several built-in translators in VIS, FBDD and ABC;
- SIS - a classical interactive tool for logic synthesis and power estimation in academia;

- MVSIS - a multi-level, multi-valued logic synthesis tool;
- FBDD [14] - a BDD logic synthesis system based on folded logic transformation and two-variable sharing extraction [14];
- BDS-PGA [12] - a practical logic synthesis system based on a BDD decomposition technique [12];
- ABC - a logic synthesis and verification tool which performs scalable logic optimisation based on *And-Invert Graphs* (AIGs) [7];
- PowerModel [8] - a power analysis tool which estimates the internal-, switching- and leakage current power consumption of an island-style FPGA;
- VIS - a tool that integrates the verification, simulation and synthesis of hardware systems.

Presenting the theoretical backgrounds of these tools is beyond the scope of this paper. For a detailed account of them, we refer the reader to [9, 2, 7, 12, 14].

3.2 Details of the design flow

The input of the design flow can be a circuit description in BLIF, PLA, or BENCH formats. Restricted subsets of EDIF, Verilog, VHDL and Synopsys Equation Format are also accepted as input. The output of the design flow is an optimised circuit in BLIF or a restricted form of Verilog. The power analysis tools can then be used to calculate the output circuits' power dissipation. Note that this design flow also enables technology mapping which converts a circuit composed of simple gates into a circuit composed of lookup tables (LUTs) suitable for implementation on an FPGA. Moreover, various transformations of circuit descriptions (e.g. Verilog to BLIF) at different stages of the flow can be done by using HDL translators available within the design flow; the correctness of such transformations is ensured by the equivalence checker in VIS and ABC.

It is evident that the proposed academic design flow is the most complete academic design flow. However, there are several issues that can be improved, namely:

- The technology libraries used in all logic synthesis tools available within the design flow are outdated. Due to this, the final optimised circuit cannot be interfaced with commercial EDA tools;
- Some commercial EDA tool features are still missing in the proposed design flow. For example, switching activity from gate-level event-driven simulation cannot yet be annotated directly onto the circuit for power analysis.

4 Integrated Design Flow

We have integrated the two design flows presented in Sections 2 and 3 (with some enhancements), providing a solid platform for future research in low power design methodologies, a flexible programming environment, new capabilities and improved performance in some cases. Our integrated design flow is depicted in Figure 3.

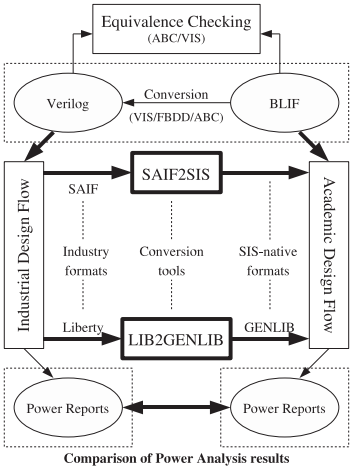

Figure 3. Integrated Design Flow

4.1 Tool flow

The integrated design flow is comprised of four different types of tool: i) unmodified existing academic tools (as presented already), ii) commercial tools (as presented already), iii) modified existing academic tools and new tools. Below gives a short description of the modified existing academic tool and new tools:

- The implementation of PowerModel was revised and improved to allow better conversion from a circuit described in BLIF into the corresponding circuit composed of LUTs.
- LIB2GENLIB is a new tool implemented in Java which converts a industrial technology (e.g. 65*nm* TSMC technology) to the corresponding *genlib* [9]—the technology library file format used by SIS, MVSIS, ABC and VIS. Note that the current release of LIB2GENLIB only supports the conversion of combinational cells.
- SAIF2SIS is a new tool implemented in C which converts relevant switching activity information from an SAIF file into the corresponding SIS switching activity file.

4.2 Attractive features of the integrated design flow

The integrated design flow consists of two sub-flows—the academic flow and the industrial flow (as shown in the Figure 3) and possesses the following attractive features:

- **Flexible programming environment** - the academic flow provides an open and flexible programming environment in which new techniques for logic optimisation, power estimation and power optimisation can be developed; and possibly compared with similar techniques presented in the industrial flow.
- **Up-to-date technology processes** - the tool LIB2GENLIB provides up-to-date commercial technology processes that can be used for mapping in the academic flow.

- **Comparison of techniques** - the academic flow provides a good platform that can be used for the comparison of the theoretical foundations behind different academic tools in such a flow (e.g. BDD decomposition technique in BDS-PGA against BDD folded logic transformation in FBDD).
- **Interoperability** - useful information (e.g. switching activity information from gate-level simulation) can be easily deduced from the industrial flow to the academic flow in order to obtain better power analysis results.
- **Comparative study** - the integrated design flow can serve as an optimal platform to evaluate the effectiveness and efficiency of techniques developed in the academic flow by means of comparative study amongst the academic flow and industrial flow.

5 Experimental Results

We ran experiments on several combinational benchmarks randomly selected from the MCNC using an AMD 2258 MHz CPU with 256MB RAM. Before running the experiments, we used our tools LIB2GENLIB to translate the 65*nm* CMOS TSMC standard cell library (tcbn65gplustc.lib) to the corresponding genlib for mapping. We used our tool SAIF2SIS to convert relevant switching activity information from SAIF files into the corresponding SIS switching activity files. (The SAIF files are obtained by running a gate-level simulation of the equivalent benchmarks modelled in Verilog using VCS.) We then ran the selected benchmarks on ABC for logic synthesis and used SIS to calculate their power dissipation with and without the switching activity files (SWF) provided by SAIF2SIS. The results shown in Table 1 indicate that annotating realistic switching activity has reduced estimated dynamic power by 40%.

circuit	without SWF [μW]	with SWF [μW]
cm138a	63.8	51.8
alu4	2519.3	1445.0
b1	58.3	31.9
c8	615.8	298.3
dalu	3880.7	1290.6
b9	417.6	113.3
count	654.5	115.4

Table 1: Power Dissipation

6 Conclusions and Future Work

An integrated design flow consisting of both academic and commercial tools has been presented, in which new techniques can be developed and compared. In addition, this integrated design flow provides a solid platform for future research in low power design methodologies, a flexible programming environment, new capabilities and improved performance in some cases (see the experimental results given in Section 5 for details). Future work will include:

1. *Scripting* - we plan to write some scripts which allow users to select and use some or all of the tools that are included in the academic flow of the integrated design flow.
2. *Extension of LIB2GENLIB* - we will extend the current release of LIB2GENLIB to translate sequential cells.
3. *Comparative study* - we plan to study the power estimation performance of the academic flow against that of the industrial flow in the integrated design flow by using the ISCAS and MCNC benchmark circuits (targeting the same technology in both flows).

Acknowledgement

We gratefully acknowledge the support of IDA Ireland and Synopsys in funding our research as indicated in this paper.

References

[1] BLIF2VHDL. Blif to vhdl translator. `http://tams-www.informatik.uni-hamburg.de/vhdl/tools/blif2vhdl/`.

[2] R. K. Brayton, M. Gao, J.-H. R. Jiang, Y. Jiang, Y. Li, A. Mishchenko, S. Sinha, and T. Villa. Optimization of multi-valued multi-level networks. In *Proceedings of the 32nd IEEE International Symposium on Multiple-Valued Logic (ISMVL 2002)*, pages 168 – 177, May 2002.

[3] R. Bryant. Graph-based algorithms for boolean function manipulation. *IEEE Transactions on Computers*, 35(8):677–691, 1986.

[4] EDIF2BLIF. An EDIF to BLIF conversion utility. `http://www.eecg.toronto.edu/~jayar/software/edif2blif/edif2blif.html`.

[5] IEEE. *IEEE standard hardware description language based on the Verilog®hardware description language (IEEE Std 1364-1995)*. IEEE, New York, 1995.

[6] IEEE. *Delay and power calculation standards - Part 3: Standard Delay Format (SDF) for the electronic design process (IEEE Std 1497-2004)*. IEEE, New York, 2004.

[7] A. Mishchenko, S. Chatterjee, and R. Brayton. Dag-aware aig rewriting a fresh look at combinational logic synthesis. In *Proceedings of the 43rd annual conference on Design automation*, pages 532 – 535, 2006.

[8] K. Poon. Power estimation for field programmable gate arrays. Master's thesis, University of British Columbia, Canada, 1999.

[9] E. Sentovich, K. Singh, L. Lavagno, C. Moon, R. Murgai, A. Saldanha, H. Savoj, P. Stephan, R. K. Brayton, and A. L. Sangiovanni-Vincentelli. SIS: A system for sequential circuit synthesis. Technical Report UCB/ERL M92/41, EECS Department, University of California, Berkeley, 1992.

[10] Synopsys. Synopsys design plateforms. `http://www.synopsys.com/products/products.html`.

[11] Synopsys SAIF. Switching Activity Interchange Format. `http://www.synopsys.com/partners/tapin/saif.html`.

[12] N. Vemuri, P. Kalla, K. O. TinMaung, and R. Tessier. BDD-based logic synthesis system for lut-based FPGAs. `http://www.eecg.utoronto.ca/~jzhu/fbdd.html`.

[13] VIS. VIS: Verification interacting with synthesis. `http://vlsi.colorado.edu/~vis/`.

[14] D. Wu and J. Zhu. FBDD: A folded logic synthesis system. In *Proceedings of the International Conference on ASIC (ASICON)*, October 2005.