Studying the influence of chip temperatures on timing integrity

using improved power modeling

András Timár¹*, Márta Rencz¹

¹Dept. of Electron Devices, Budapest University of Technology and Economics, Budapest, Hungary, H-1117, {timar, rencz}@eet.bme.hu

* corresponding author: András Timár

Address:

Budapest University of Technology and Economics Department of Electron Devices Magyar Tudósok körútja 2, Q Building, Floor 3 Budapest, Hungary, H-1117

Office : (+36-1) 463-2797 Fax : (+36-1) 463-2973 Email : timar@eet.bme.hu

Date of Receiving: to be completed by the Editor

Date of Acceptance: to be completed by the Editor

Studying the influence of chip temperatures on timing integrity using improved power modeling

András Timár, Márta Rencz

Abstract — Thermal side-effects can detrimentally influence operation of integrated circuits. The increase of temperature changes the devices' characteristics and may result in timing integrity issues. In extreme cases the increased delays can foil correct operation of the circuit. This paper presents a methodology to address timing integrity errors caused by thermal effects. The methodology presented shows how the thermal distribution map on the IC surface can be used to calculate device delay changes during logic simulation. A software tool called **CellTherm** developed in the Department of Electron Devices, BME, Hungary is also briefly presented in this paper. With the help of the software, logic simulations of digital integrated circuits can be back-annotated with temperature-dependent delays during the running simulation. Also various aspects of power characterization methods are also presented which were used throughout the experiments.

Keywords — electro-thermal simulation, temperature distribution, timing integrity, delay backannotation

1 INTRODUCTION

In this paper an improved version of CellTherm[1] is presented as well as the characterization of cells in the library for power and temperature-dependent delays. Simulations were run in order to determine how the temperature fluctuations affect the cell delays in a certain standard cell library. The simulated standard cell library was a TSMC 0.35um technology design kit. With the chosen cell library simulations confirmed that with increasing device temperature, cell delays are also increasing. Logic simulations confirmed that the increased delays can mistune setup and hold timing so the circuit may fail to operate correctly. Using the simulation data, exact temperature-delay functions of the standard cells have been acquired.

This paper also presents several power modeling algorithms used in CellTherm. The simulator engine uses power characteristics of standard cells composing the design in question in order to calculate self heating of the circuit. These power characteristics have been simulated using different algorithms that approximate real power consumption. This paper introduces a methodology to characterize standard cells for power with minimum effort. The method proposed is an alternative to characterize standard cell libraries that do not include power/timing characterization databases, such as a **Liberty** database. The application of the measured energy data is easier and faster than that of the Liberty database.

2 RELATED WORK

2.1 Electro-thermal simulators and methodologies

The authors of [2] propose a methodology that constitutes the basic idea of this paper. It deals with temperature map generation of ICs from digital simulations. The method presented in this paper improves that methodology by allowing in-situ logic and thermal co-simulation of an IC design. The

work presented in [2] lacks the possibility of back-annotating temperature-dependent delays into the running simulation. Dealing with temperature-dependent delays become more and more important as feature size is shrinking and power density is growing. Failure to take the temperature-dependent delays into account can cause timing (e.g. setup- and hold time) violations that can foil correct logic operation. The methodology presented in this paper also improves the work of Torki et al by coupling the logic and thermal simulator engines with a custom controller and visualization application that can evaluate logical and thermal calculations and prepare delay back-annotation on-the-fly, in the middle of the simulation.

Present electro-thermal simulators approach the electrothermal simulation problem either by FEM simulation, relaxation method or simultaneous iteration. A method of electro-thermal simulation using simulator coupling is presented in [3]. The technique is based on the coupling of a FEM program with a circuit simulator. A pre-RTL temperature-aware design methodology is presented in [4], where a fast, yet accurate architectural thermal model that is able to explore large regions of the design space is proposed. [5] attempts to show that there is a significant peak temperature reduction potential in managing lateral heat spreading through floorplanning. As a demonstration, it uses a wire delay model and floorplanning algorithm based on simulated annealing to present a profile-driven, thermal-aware floorplanning scheme that significantly reduces peak temperature with minimal performance impact that is quite competitive with Dynamic Thermal Management (DTM). The floorplanning tool HotFloorplan is part of the HotSpot software [6] that is developed at DCS, University of Virginia. In addition, simulating the self-heating of the circuit in the early phase of the design before manufacture would make cooling issues less problematic. Self-heating simulations may also eliminate the need for design back-annotation after manufacture. An example of temperatureaware ASIC design flow can be found in [7].

2.2 Power characterization methods

Industry's leading and widely used Open Source Liberty format[8] developed by Synopsys, Inc. is a characterization database that is built to help circuit designers, EDA vendors, fabs, etc. to deal with power, timing, leakage issues. This open database consists of one or more data files that contain measurement data regarding power, noise and timing characteristics of every standard cell in a digital library. The library uses a novel modeling methodology called \emph{Composite Current Source} (CCS) modeling[9]. The database file contains look-up tables for timing, power and noise data and simulator tools can interpolate into these tables to acquire the needed values.

In order to create a Liberty database for a standard cell library the creator of the design kit has to characterize the library with tools like PrimeTime, Library Compiler, Liberty NCX, NanoChar, etc.¹ from Synopsys, Inc. This paper presents a power characterization methodology for designers who do not have access to such tools. The presented methodology requires only a standard SPICE-like simulator engine. In our work we used ELDO from Mentor Graphics.

Authors of [10] present a simple characterization method based on the linear delay model. The paper proposes power and delay characterization methods for standard library cells. Power characterization is done by exciting every cell in the library with every possible input combination. In our work we developed this approach further.

Kabbani introduces a simple and yet accurate closed-form expression to estimate the switching power dissipation of static CMOS gates in [11]. The paper focuses on switching power dissipation and creates a model for dynamic power consumption caused by gate output transitions and fan-out capacitance charging and discharging. This approach does not deal with internal energy dissipated when only gate inputs change but outputs remain static.

¹ Registered trademarks of Synopsys, Inc.

[12] presents a tool, **AutoLibGen**, that uses Composite Current Source (CCS) based technique for characterising a standard cell library comprising of basic combinational circuits like NAND, NOR, AND, OR, NOT and BUFFER cells for timing, power and noise.

3 HOT-SPOT DETECTION AND DESIGN FLOW

3.1 Hot-spot detection and temperature map determination

By calculating the temperature distribution of the circuit, local and global hot-spots can be detected. Hot-spot detection is crucial in today's ultra large scale integrated circuits as functional operation may completely fail due to unanticipated heating of devices at certain coordinates. With our presented methodology and application bundle, circuit designers can create temperature-aware IC designs where thermal issues emerge during the design phase rather than after manufacture.

The simulation stimuli resembles the real-world operation of the circuit thus the **fully packaged** design can be evaluated for defects or misfunctionality far before manufacture. The evaluation is feasible by using a thermal simulator that is capable of simulating the design together with an arbitrary compact thermal model of the package and its surroundings.

3.2 Design Flow

The presented methodology is based on standard tools such as Verilog hardware description language and its Programming Language Interface (PLI), thus it can be integrated into any logical simulation flow easily. The mayority of today's logic simulators (both commercial and open-source licensed) support this language and interface.

The **CellTherm** simulation software acts as a glue logic between the standard logic simulator and the thermal simulator. The **Therman** solver engine [13] was used as the thermal simulator that was also developed in the Department. The Therman solver engine can be augmented with the compact thermal model of the IC package where the compact model is generated from thermal transient testing measurements.

The glue logic interface application controlls communication between the logical and the thermal simulator engines and displays the results in real-time. Real-time means that the simulation results get displayed right after each simulation timestep, thus providing the user immediate graphical feedback of the result. In Fig. 1 the design flow of the logi-thermal simulation is shown.

The logi-thermal design flow starts with two concurrent processes. On one hand, the IC designer team describes the logic behaviour of the application in a high level language like SystemC, SystemVerilog, SystemVHDL, etc. Then the process continues with an iterative simulation cycle where the majority of the logic and timing errors can be detected and the design can be corrected. After successful logic and timing simulation, synthesis takes place, where logical RTL gates get mapped to the physical gates of the chosen process design kit (PDK). This synthesized description can then be simulated for correct timing and the floorplan can be created.

Simultaneously, the cell power characterization has to be done. Often the process design kit comes with Liberty or TLF files which contain data about the cells' timing, delay and power characteristics. Unfortunately, sometimes this is not the case. The PDK may be missing these files, or just on type of them, or the necessary power characteristics for the cells. Therefore a rather simple method has been developed to obtain the power characteristics of the cells. The characterization process can be done independently of the high level design and synthesis. The power characterization methodology is described in detail in Section 8.

The synthesized HDL description is stimulated with a logic simulator with test vectors and switching activity from the results can be determined for each cell. From the Place&Routed floorplan, the switching activity and the power characteristics of the cells the CellTherm application can calculate the temperature distribution in the design. This is done concurrently with the logic simulation thus immediate graphical results show the evolved hot-spots.

Fig. 2. shows a screenshot of the CellTherm application after the first few nanoseconds of the simulation time where steady-state temperature distribution has not yet been achieved.

Temperature-dependent delays could also be calculated and back-annotated to the simulation via Standard Delay Format (SDF) files. Delay back-annotation can take place before each timestep of the simulator.

4 PRELIMINARY TIMINGS

In the early phase of the design, when the behavioural description gets synthesized, pre-layout timing data can be approximated and the synthesizer software (e.g. LeonardoSpectrum from Mentor Graphics) can extract preliminary delay values from the design. The synthesizer software outputs the predicted post-synthesis pre-layout delay data into a **Standard Delay Format** (SDF) file, which can later be included in a logic simulation. The logic simulator can take these delay data as startup timing values and it is able to check against basic timing integrity issues. The SDF file contains not just delays of the individual cells of the design but setup and hold timing checks also. This way the simulator can send off an alert if timing requirements are not met. The preliminary delay and timing values are predicted values and do not take physical layout into account. Later on, after floorplanning and Place&Route, real timing data can be extracted from the design in SDF format. Fig. 3 shows a portion of a generated SDF file with the predicted delay and timing values.

5 LOGIC SIMULATION WITH SDF DATA

The annotation of delay values takes place via the *\$sdf_annotate()* Verilog construct in the simulator. A test case has been developed as a proof-of-concept to demonstrate setup and hold timing

failure caused by applied SDF delay values. A basic flip-flop chain has been built in Verilog with the SDF delay data acquired from the synhtesis and stimuli were intentionally corrupted in order to observe timing errors. The **setup** and **hold** timing constraints extracted from the SDF delay data are T_{setup} =470ps and T_{hold} =60ps, respectively. In Fig. 4 the simulator sends off an error message that the setup timing requirements are not met.

6 ANALOG SIMULATION RESULTS

Analog simulations were run with Mentor Graphics's ELDO simulator to examine cell delays in function of temperature. Each cell of the standard cell library have been simulated in a test environment where only the cell in question was present and the ambient temperature has been swept in a wide range. The ambient temperature was swept from -40°C to +90°C. The temperature-dependence of delays could unequivocally be demonstrated. Fig. 5. shows the temperature-delay function of a D flip-flop circuit. The delay between DATA input's rising edge and the CLK rising edge has been set so a setup timing failure could be detected above a certain temperature.

In Fig. 5 the curve runs from -40° C to $+40^{\circ}$ C. Over T>+40°C delay between CLK and the Q output cannot be interpreted because the output did not change. This is clearly a setup timing error. The intrinsic delay of the cell increased so much with the temperature that the setup time requirement could not be satisfied, the input transition could not propagate to the output.

Fig. 6 shows the transient simulation results where the ambient temperature of the cell has been swept. With the temperature rising, the delay between CLK and Q rises up to a certain point, where setup timing requirements are no longer met. Over T>+40°C the input signal change cannot propagate to the output thus leaving the output at a low logic level.

7 COMBINING TEMPERATURE MAP AND DELAY FUNCTIONS

With the proposed method demonstrated in Section 6 temperature-delay functions of the standard cells have been acquired. Using the temperature map resulting from the **CellTherm** application actual temperatures of the cells can be determined. Interpolating the temperature values into the temperature-delay functions the delays belonging to a certain temperature can be derived. The CellTherm application is able to back-annotate these temperature-dependent delay values into the running logic simulation via SDF structures. After every simulation time-step, the simulator continues calculation with the altered delay values. If delays change too much with the temperature and setup or hold timing requirements cannot be satisfied any more, the simulator sends off an error message and stops simulation.

8 POWER CHARACTERIZATION

8.1 Characterization background

Several power characterization methods were tested in order to approximate real power consumption of the standard cells as much as possible. This paper shows an alternative way of library power characterization for custom standard cell libraries that do not come with a Liberty database (see Subsection 2.2) or any other power/timing/noise characterization data. The presented method is used in the CellTherm software on a TSMC 0.35µm standard cell library where such a database was not available.

When characterizing cells for power the most important design variables are **output load capacitance** and **input transition time** (rise- and fall-time). Characterizing the cell library in a general manner means that these two variables have to be given in a semi-empirical way or a look-up table of different values has to be incorporated in a database (e.g. Synopsys OpenLiberty format) which than can be interpolated.

In CMOS circuits the majority of power consumption comes from dynamic switching activity. Static power consumption is fewer by almost three orders of magnitude². Altough in today's low-power, deep submicron (\leq 45nm) technologies leakage power tends to be an important issue, it is still fewer than power consumed during dynamic switching. Logic transitions can be held responsible for most of the power consumption that causes device heating. This means that logic transitions and their belonging power consumptions have to be unequivocally found.

When using a logic simulator to simulate the design, the **number** of signal transitions on each net and port can be extracted. If we could —with a good approximation— find a typical power consumption per logic transition per cell, we could simplify total power consumption calculation to a simple multiplication: $E_{real} \approx E_{calc} \cdot \#$ transitions.

The power characterization of the cells was done with an analog simulator (ELDO) using the following algorithm. A custom ELDO netlist was generated according to the simulated cell library and all of the cells of the library were simulated with every possible input combination. During the initial analog simulation with ELDO, dissipated energies during every possible switching event in a cell were collected and stored for every cell type. The energies dissipated during a logic transition were calculated using (1):

$$\varepsilon = \int_{T_1}^{T_2} P(t) dt \tag{1}$$

where P(t) is the total power consumption function of the cell after every possible logic transition. Fig. 7 shows a typical input voltage-power consumption characteristic.

Initially it was assumed that for every cell in a given design, power consumption is larger when the

² Observed from power simulations on a 0.35µm TSMC technology

cells' outputs change logic value than when only **internal** switching takes place (only inputs changed, no outputs). Therefore the CellTherm engine distinguished two energy values per cell: one for "output changing" and one for "internal switching". The engine registered both the logic transitions where output had changed as well as logic transitions where only input had changed. E_{max} denotes the largest energy value where output logic transition occured. E_{min} stands for the **maximal** energy consumption measured when only internal switching took place. This approach clearly overestimates real power consumption but simulations showed that this behaviour does not affect hot-spot detection considerably. The position of the detected hot-spots remains the same.

Table I shows calculated and real energy dissipation of a 4-bit counter circuit's cells. E_{calc} is calculated with (2):

$$E_{calc} = E_{max} \cdot \# \text{outputs_changed} + E_{min} \cdot \# \text{internal_changes}$$
(2)

 E_{real} is extracted from the analog ELDO simulator as the true energy consumption over the whole simulation time.

$$E_{real} = \int_{0}^{T_{end}} P_{cell}(t) dt$$
(3)

The difference between calculated and true simulated energy dissipations in percentage is given by (4):

Difference in percent =
$$\frac{E_{calc} - E_{real}}{E_{real}} \cdot 100\%$$
 (4)

8.2 Average of all measured energies per cell

In Table II the numerical average of all the measured energies –whether output changed or not– was calculated. The calculated energy values underestimated the real energy in almost all the cases. The

four exceptions are the DFFR flip-flops that drive the output of the counter. These flip-flops have a clock input that changes in every period and thus the flip-flops swayed throughout the whole simulation time. The other cells in the design had been idle for shorter or longer periods. The everchanging DFFR cells' average energy multiplied by the number of changes (that is, every clock cycle) quite apparently resulted in exactly the energy dissipation measured throughout the total simulation time.

8.3 Average of E_{max} and E_{min} per cell

In Table III power estimation was approximated by the average of the measured E_{max} and E_{min} values. Two outstanding difference values can be observed in the table at instances **ix140** and **ix90**. In the case of **ix140**, the **NAND03** gate changed internal state much more than its output. Throughout the simulation it changed its output four times but changed internal state twelve times. Because the internal switching dissipates less energy than output transitions and we multiply the number of transitions with a mean energy value between E_{max} and E_{min} , the result will be clearly an overestimate.

The **ix90** instance is a multiplexer circuit and simulations showed that this type of cell dissipates more power during internal switching activity than when driving its fan-out. Read more on this in subsection 8.5.

8.4 E_{min} per cell

In Table IV the transition numbers were multiplied by E_{min} values. Power estimation with these values showed the largest spread around the real values.

8.5 Monte Carlo method

The assumption that a cell dissipates more when its output changes (because it has to drive its fanout) and it dissipates less when only internal switching takes place, proved to be false. Simulations showed that for certain cells this assumption is simply not true. In our test case, multiplexer cells were the best instances to prove this theory. It clearly turned out that the multiplexer cells dissipate more power when changing logic states internally than when changing their outputs and charging/discharging their fan-outs. The same phenomenon was observed by authors of [10].

As a result of this experience a new, generic method was tried out to characterize cells for power. We excited every cell of the given design independently of each other but inside the same netlist. Only one type of switching event was registered: when logic state changed on either output or input. "Input-only" and "Input-output" transitions were interlaced.

The cell instances in the design were excited one-by-one leaving all other cell instances intact and floating. The exciting signal was a linear feedback shift register (LFSR) voltage source for every input of the cell being examined. With this pseudo-random stimuli a Monte Carlo simulation algorithm has been developed to extract cell power characteristics.

The simulated power dissipation has been averaged and divided by one period of the input stimuli. This value is then multiplied by the number of occured logic transitions on either outputs or inputs. Logic transitions taking place at the same time (in same time period window) are considered one transition. The resulting figures are shown in Table V.

8.6 Power characterization comparison

In Fig. 8 the comparison of the evaluated methods can be seen. From the numbers and the chart the Monte Carlo simulation can be said the most predictable solution. Altough it systematically underestimates the real energy values, it approaches them best with a minimal spread. From the hot-spot detection's point of view, this underestimation can be considered as a slight negative offset in temperature values, but it is done for all the cells thus the overall position of the hot-spots would not change.

9 CONCLUSION

A new simulation software tool, CellTherm, has been developed in the Dept. of Electron Devices, BME, Hungary which is able to provide cell-resolution thermal distribution map of the surface of a fully packaged integrated circuit. It uses stimuli of standard logic simulators (e.g. ModelSim) and the cells' power characteristics. The simulator is able to take the effect of the whole package into account by coupling compact models of the packaging. With the resulting time-dependent surface temperature map, the time-dependent average temperature of any standard cell block in the IC can be determined. By interpolating the cell's local temperature at a given time into the previously simulated temperature-delay functions, actual cell delays can be acquired.

With the help of the cells' temperature-delay characteristics, a continuous picture of the actual delays of any cell during the logic transient simulation can be monitored. The developed CellTherm simulator program is able to co-operate with a logic simulator and can immediately annotate the modified delays back to the running logic simulation. This way, besides getting a clear picture of the IC's surface temperature distribution in any moment, the timing violations caused by overheating can immediately delve out; the simulator signals when timing criteria are not met.

This paper has also introduced five power characterization methods that were also compared. A feasible method to calculate powers using realistic and actual fan-out load capacitances has been proposed. The introduced methodology differs from previous works in that the cells of the library are characterized inside the given circuit, not independently thus more accurate results can be achieved for power and timing measurements. The output load capacitances and input transition times do not have to be interpolated from a look-up table because they are given inherently by the netlist of the circuit. With the proposed method the hot-spot detection could be faster due to the elimination of the interpolation and parsing of the Liberty database file. The characterization needs only a SPICE-

compatible analogue simulator.

Our method can also be used when the technology design kit does not contain any power/timing charaterization libraries. Using the power data acquired with our proposed method the hot-spots on the IC surface as well as the temperature dependence of the delays could be approximated with good accuracy.

10ACKNOWLEDGEMENT

This work was partly supported by the IP 248603 THERMINATOR FW7 project of the European Union and by the Hungarian Government through TÁMOP-4.2.1/B-09/1/KMR- 2010-0002.

REFERENCES

- A. Timár, G. Bognár, A. Poppe, and M. Rencz, "Electro-thermal cosimulation of ICs with runtime back-annotation capability," in Proceedings of the 16th Therminic Workshop, Barcelona, Spain, 6-8 October 2010.
- [2] K. Torki and F. Ciontu, "Ic thermal map from digital and thermal simulations," in Proceedings of the 8th Therminic Workshop, Madrid, October 2002, pp. 303–308.
- [3] S. Wünsche, C. Clauß, P. Schwarz, and F. Winkler, "Electro-thermal circuit simulation using simulator coupling," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 5, no. 3, September 1997.
- [4] W. Huang, K. Sankaranarayanan, K. Skadron, R. J. Ribando, and M. R. Stan, "Accurate pre-RTL temperature-aware design using a parameterized, geometric thermal model," IEEE Transactions on Computers, vol. 57, no. 8, August 2008.
- [5] K. Sankaranarayanan, S. Velusamy, M. Stan, and K. Skadron, "A case for thermal-aware floorplanning at the microarchitectural level," Journal of Instruction-Level Parallelism, vol. 8, no. 1-16, 2005.
- [6] W. Huang, S. Ghosh, S. Velusamy, K. Sankaranarayanan, K. Skadron, and M. R. Stan,
 "Hotspot: A compact thermal modeling methodology for early-stage vlsi design," IEEE
 Transactions on Very Large Scale Integration (VLSI) Systems, vol. 14, no. 5, May 2005
- [7] W. Huang, M. R. Stan, K. Skadron, K. Sankaranarayanan, S. Ghosh, and S. Velusamy, "Compact thermal modeling for temperature-aware design," 41st Design Automation Conference (DAC), San Diego, CA, June 2004.

- [8] (2011, 27th June, 11:19) Open source liberty webpage. [Online]. Available: <u>http://www.opensourceliberty.org</u>
- [9] G. Mekhtarian, "Composite Current Source (CCS) Modeling Technology Backgrounder," Synopsys, Inc., November 2005.
- [10] J. B. Sulistyo and D. S. Ha, "A new characterization method for delay and power dissipation of standard library cells," VLSI design, vol. 15, no. 3, pp. 667–678, 2002.
- [11] A. Kabbani, "Logical effort based dynamic power estimation and optimization of static CMOS circuits," Integration, the VLSI Journal, vol. 43, no. 3, pp. 279–288, June 2010.
- [12] I. K. Rachit and M. S. Bhat, "AutoLibGen: An open source tool for standard cell library characterization at 65nm technology," in International Conference on Electronic Design, Penang, Malaysia, 1-3 December 2008.
- [13] V. Székely, A. Poppe, M. Rencz, M. Rosental, and T. Teszére, "Therman: a thermal simulation tool for IC chips, microstructures and PW boards," Microelectronics Reliability, vol. 40, no. 3, pp. 517–524, 2000.

FIGURES AND TABLES



Figure 1. Logi-thermal design flow



Figure 2. CellTherm application showing actual temperature map of simulated IC



Figure 3. Pre-layout SDF file



```
VSIM 9> run
```

Figure 4. Setup timing error in ModelSim



Figure 5. Delay vs. timing function of a D flip-flop cell



Figure 6. Output transient results when sweeping ambient temperature



Figure 7. Calculation of dissipated energy per logic transition



Figure 8. Graphical comparison of power estimation methods

Power characterization with $E_{\rm MIN}$ and $E_{\rm MAX}$ values				
Cell type	Instance name	E _{calc} [pJ]	E _{real} [pJ]	Diff. [%]
XOR2	ix80	21.145	13.129	61.06
NAND03	ix140	7.552	3.511	115.10
XNOR2	ix31	2.907	2.227	30.51
XNOR2	ix25	6.247	4.581	36.37
MUX21_NI	ix110	2.954	2.298	28.58
MUX21_NI	ix100	5.680	4.458	27.41
NAND02	ix131	7.274	5.151	41.21
OAI21	ix129	3.973	3.598	10.40
MUX21	ix90	8.994	4.198	114.26
DFFR	reg_output_3	90.143	67.063	34.42
DFFR	reg_output_2	95.023	71.919	32.12
DFFR	reg output 1	97.337	80.437	21.01

TABLE I

Cell type	Instance name	E _{calc} [pJ]	E _{real} [pJ]	Diff. [%]
XOR2	ix80	6.377	13.129	-51.43
NAND03	ix140	1.605	3.511	-54.29
XNOR2	ix31	0.255	2.227	-88.57
XNOR2	ix25	1.047	4.581	-77.14
MUX21_NI	ix110	0.328	2.298	-85.72
MUX21_NI	ix100	1.146	4.458	-74.29
NAND02	ix131	2.355	5.151	-54.29
OAI21	ix129	1.645	3.598	-54.29
MUX21	ix90	2.039	4.198	-51.43
DFFR	reg_output_3	67.062	67.063	0.00
DFFR	reg_output_2	71.919	71.919	0.00
DFFR	reg_output_1	80.436	80.437	0.00
DFFR	reg_output_0	101.472	101.473	0.00

TABLE II

POWER CHARACTERIZATION WITH AVERAGE OF ALL MEASURED ENERGIES

TABLE III

Power characterization with average of E_{max} and E_{min}

Cell type	Instance name	E _{calc} [pJ]	E _{real} [pJ]	Diff. [%]
XOR2	ix80	10.593	13.129	-19.31
NAND03	ix140	8.315	3.511	136.83
XNOR2	ix31	2.321	2.227	4.20
XNOR2	ix25	5.107	4.581	11.48
MUX21_NI	ix110	2.917	2.298	26.94
MUX21_NI	ix100	5.613	4.458	25.90
NAND02	ix131	7.274	5.151	41.21
OAI21	ix129	3.973	3.598	10.40
MUX21	ix90	8.909	4.198	112.23
DFFR	reg_output_3	101.276	67.063	51.02
DFFR	reg_output_2	105.505	71.919	46.70
DFFR	reg_output_1	103.943	80.437	29.22
DFFR	reg_output_0	109.271	101.473	7.68

TABLE IV			
POWER CHARACTERIZATION WITH E _{MIN}			
Cell type	Instance name E _{calc} [pJ]	E _{real} [pJ]	Diff. [%]

XOR2	ix80	0.041	13.129	-99.69
NAND03	ix140	6.789	3.511	93.37
XNOR2	ix31	1.149	2.227	-48.43
XNOR2	ix25	2.826	4.581	-38.31
MUX21_NI	ix110	3.105	2.298	35.12
MUX21_NI	ix100	6.218	4.458	39.47
NAND02	ix131	5.235	5.151	1.63
OAI21	ix129	0.828	3.598	-76.98
MUX21	ix90	10.354	4.198	146.65
DFFR	reg_output_3	88.707	67.063	32.27
DFFR	reg_output_2	91.917	71.919	27.81
DFFR	reg_output_1	91.775	80.437	14.10
DFFR	reg_output_0	91.744	101.473	-9.59

 TABLE V

 POwer characterization with pseudo-random input stimuli

Cell type	Instance name	E _{calc} [pJ]	E _{real} [pJ]	Diff. [%]
XOR2	ix80	24.407	30.596	-20.23
NAND03	ix140	8.732	9.578	-8.83
XNOR2	ix31	19.145	24.000	-20.23
XNOR2	ix25	18.344	22.996	-20.23
MUX21_NI	ix110	17.246	18.917	-8.83
MUX21_NI	ix100	18.055	19.804	-8.83
NAND02	ix131	9.326	11.691	-20.23
OAI21	ix129	9.834	10.787	-8.83
MUX21	ix90	13.006	14.266	-8.83
DFFR	reg_output_3	54.049	59.285	-8.83
DFFR	reg_output_2	48.108	60.306	-20.23
DFFR	reg_output_1	54.551	59.835	-8.83
DFFR	reg_output_0	55.580	60.965	-8.83

BIOGRAPHIES

András Timár is an assistant lecturer in the Department of Electron Devices, BME, Budapest, Hungary. His main fields of interest are programming, hardware description languages, computer graphics and thermal problems of microelectronics. He received the Ms.C. degree in electronic engineering in 2006 in BME. He finished his PhD studies in 2009 and since then working in the Department of Electron Devices as an assistan lecturer. He is responsible for the Mentor Graphics computer laboratory in the Department.

Márta Rencz received the electrical engineering degree in 1973 and the Ph.D. degree in 1980, both from the Technical University of Budapest, Budapest, Hungary.

Her first research area was the simulation of semiconductor devices. Later, she participated in the development of several CAD programs in microelectronics. Her latest research interests include the thermal investigation of ICs and MEMS, thermal sensors, thermal testing, thermal simulation, and electrothermal simulation. She is cofounder and CEO of MicRed. She has published her theoretical and practical results in more than 300 technical papers.

Dr. Rencz is a member of IMAPS. She is an organizing committee and program committee member of several international conferences and workshops. For her research results in thermal modeling, she has received the Harvey Rosten Award of Excellence from the electronics thermal management community.