Elektrotehniški vestnik XX(Y): 1–5, YEAR
Electrotechnical Review, Ljubljana, Slovenija

Transistor-level optimisation of digital cells

Janez Puhan, Iztok Fajfar, Tadej Tuma, Árpád Bűrmen
University of Ljubljana, Faculty of Electrical Engineering Ljubljana, Slovenia
E-mail: janez.puhan@fe.uni-lj.si

Abstract. Integrated-circuit manufacturers provide libraries of pre-designed digital cells for digital-circuit design. The transistor-level implementation of the cells is not considered during the design procedure. The cells, such as buffers, logic gates, adders, flip-flops, etc. are used without modifications wherever needed. For that reason the cells are not fine-tuned to their unique surrounding conditions in the circuit. This leaves space for cell optimisation to be used as a tool for achieving arbitrary customisation. This paper describes a case study of transistor-level digital-cell optimisation. Since the landscape of the cost function used in the process of optimisation proved to be too noisy, the optimisation runs were performed using a robust global-optimisation method. The results show that a substantial improvement of cell properties with respect to the manufacturer-supplied pre-designed cells can be obtained.

Key words: digital ASIC design, pre-designed cells, digital-circuit synthesis, transistor-level cell optimisation

1 Introduction

When designing a digital circuit, the designers work with pre-designed digital cells or blocks [1], such as buffers, logic gates, adders, latches, flip-flops, etc. The foundry usually provides a library of digital cells for every IC manufacturing process. It is customary that several versions of every cell are provided along with cell characteristics. Transistor-level simulations [2, 3, 4, 5] do not take place during the design phase. The circuit response is calculated using higher-level cell descriptions [6, 7].

In most cases, various versions of a cell share the same transistor topology. They differ in transistor sizing, usually only in transistor-channel widths. Different versions of a cell are a result of cell optimisation to different demands. We decided to test a foundry-provided cell library to verify if it is possible to improve the performance of foundry-provided cells. We also wanted to see if transistor-level cell optimisation makes sense, so that by automating it in future one could generate arbitrary cells customised to particular needs of a circuit.

The process of designing a digital Application-Specific Integrated Circuit (ASIC) [8, 9] starts with a Register Transfer-Level (RTL) implementation of the circuit specifications. The circuit described using a Hardware Description Language (HDL) is simulated and refined by the designer until the simulation yields satisfactory performance. The resulting HDL description of the circuit is then reduced into a gate-level netlist which in turn is optimised until timing constraints are satisfied. The reduction and gate-level optimisation are performed by a synthesis tool using a foundry-provided cell library. The final circuit is verified before and after the layout is generated. The synthesis has to be repeated if verification fails.

Our main idea depicted in Fig. 1 is to introduce transistor-level cell optimisation into the synthesis flow. Synthesis in step A produces a gate-level netlist of the circuit and a list of timing constraints for all the cells. Suitable cell implementations are selected from the foundry-provided library. What follows is the proposed optimisation step, which sizes the topologies of selected cells according to individual timing constraints. Some cells selected in step A barely fulfil the timing requirements while others have a broad safety margin. The former ones can be optimised for speed while the latter ones can be optimised for power consumption, without affecting the circuit performance. As a result of optimisation, a cus-

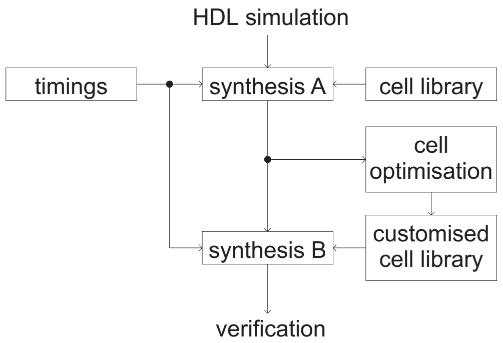

Figure 1. Introduction of transistor-level optimisation into the synthesis flow

Received 19 October 2010
Accepted

tomised cell library could be built. The synthesis in step B would swap the original cell library with a customised one. Successful step B would also validate the resulting design that uses customised cells.

2 Optimisation criterion (cost function)

The cost function (CF) is a measure of a candidate circuit quality. The better the candidate, the lower the CF value. By defining CF, the decision as to which circuit is better becomes clear [10].

The performance of a candidate circuit is evaluated by means of one or more transient analyses. Properties that are of interest are: chip area (defined by the transistor sizing), time domain behaviour (slopes, delays, etc.), and power consumption.

Each measured property x_i contributes a portion to the CF value. Until property goal g_i is not reached, its contribution $c_i(x_i)$ is proportionate to the violation of the goal. When the goal is reached or even exceeded, it becomes negative. The definition is given in Eq. (1).

$$c_i(x_i) = \begin{cases} \frac{t_i}{g_i}(x_i - g_i) & x_i \leq g_i \\ \frac{p_i}{g_i}(x_i - g_i) & x_i > g_i \end{cases} \tag{1}$$

Since all properties of interest $\mathbf{x} = [x_1, x_2, \ldots]^T$ have to be as low as possible, only one type of contribution function described in Eq. (1) is sufficient. The value of CF is a sum of contributions given in Eq. (2).

$$c(\mathbf{x}) = \sum c_i(x_i) \tag{2}$$

CF in Eq. (2) has a global minimum represented in Eq. (3) that corresponds to optimal circuit parameters $\mathbf{w}_{\mathrm{opt}}$. With different goals g_i, trade-offs (t_i) and penalty weights (p_i), optimisation can lead to several versions of the circuit with various tradeoffs between the circuit properties.

$$c(\mathbf{x}(\mathbf{w}_{\mathrm{opt}})) \leq c(\mathbf{x}(\mathbf{w})) \tag{3}$$

By definition, not achieving one of the goals can be compensated by exceeding others. To make sure that all the goals are achieved, relation $t_i \ll p_j$ has to hold for each pair $i \neq j$. Therefore, the CF contribution of a property that fails to satisfy its goal has to be significantly greater than the absolute values of contributions arising from properties exceeding their goals.

The vector of properties $\mathbf{x}$ cannot be determined for circuit candidates for which the transient analysis fails. When particular x_i is not known, its contribution $c_i(x_i)$ is set to some large value $c_{\max_i}$. Such a candidate produces a large CF contribution and represents a bad try. The same goes for candidates for which the transient analysis succeeds, but the circuit does not behave as expected and one or more properties cannot be determined (e.g. the output not having a rising edge where a rising edge is expected).

Auxiliary measurements additionally penalise circuit candidates that do not perform as expected. They check particular time points, where the state of the digital circuit is defined. Auxiliary measurements have large penalty weights that result in large CF contributions for candidate circuits that fail to satisfy the goal. On the other hand, they do not interfere (trade-off weights are zero) if the candidate behaves as expected. They enforce the correct response and therefore help the optimisation process.

3 Benchmark circuit

A pre-designed foundry-provided (350nm CMOS process) D flip-flop cell depicted in Fig. 2 was used as a benchmark. It is reasonable to expect that if transistor level optimisation significantly improves the flip-flop circuit, it should also improve arbitrary foundry-provided cells.

The benchmark cell is put into a test-bench circuit providing input signals, power supply voltage and output loads. Input slopes, supply voltage, and capacitances vary from corner to corner.

The input test signals describing the time-domain measurements are also shown in Fig. 2. Together with the chip area and power consumption they represent measurements x_i that contribute to the cost in Eq. (2).

There were twenty time-domain measurements made. Besides delays and slopes, the output setup time, input hold time, recovery time after reset, minimal clock period, and minimal reset impulse duration were taken into account. One delay and one slope are depicted in Fig. 2 for illustration.

The delay was defined as the time from the moment the input signal reached the 50% level to the moment the output signal reached the 50% level. The slope was defined as the time it took for the signal to go from 10% to 90% level.

4 Optimisation

Two optimisation runs were performed. In the first run the goal was to obtain a faster circuit without increasing the power consumption when compared to the foundry-provided cell (speed run). In the second run, the goal was to decrease the power consumption without degrading the timings (power run).

The original cell properties were used as goals g_i. High penalties p_i ensured that all the goals were reached in both runs. The two runs differed in trade-off weights t_i, which emphasized the speed in the first and the low power consumption in the second run.

The cell was optimised for various process and operating condition corners. Combinations of four process corners (worst power, worst speed, worst one, worst zero), two temperatures ($-25\,^{\circ}$C, $105\,^{\circ}$C), two power supply

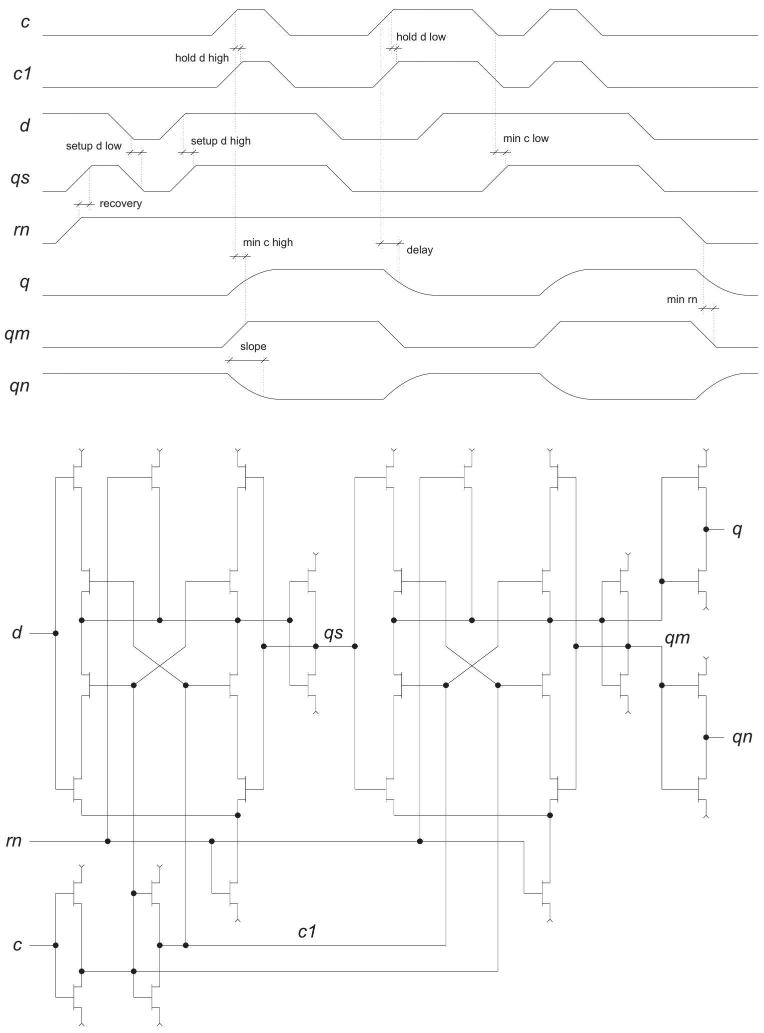

Figure 2. D flip-flop ($d = \mathrm{data}$, $c =$ clock, $rn = \overline{\mathrm{reset}}$, $q = \mathrm{out}$, $qn = \overline{\mathrm{out}}$)

voltages (3V, 3.6V), two output loads (10fF, 220fF), and two input-signal slopes (60ps, 4ns) were taken into account. In addition to these 64 extreme combinations, a typical corner was added.

Only four critical corners need to be evaluated to calculate the worst-case cell-property values. For this reason evaluating the properties over other 61 corners was needless. This reduced the number of corners to only four: wp/$-25\,^\circ$C/3.6V/220fF/4ns, ws/$105\,^\circ$C/3V/10fF/60ps, ws/$105\,^\circ$C/3V/220fF/60ps, and ws/$105\,^\circ$C/3V/220fF/4ns. The final results were verified across all 65 corners.

The independent optimisation variables were transistor channel widths and lengths. The lengths were matched and resulted in one optimisation variable. The total number of optimisation variables was 33. The explicit constraints were from 400nm to 2µm for the channel widths and from 350nm to 2µm for the channel lengths. Since the channel lengths tend to be as small as possible, the expected final channel length should be equal to the lower constraint. Contrary to our expectations, this does not seem to be true for both runs.

Parallel Simulated Annealing with the Differential-Evolution (PSADE) [11] optimisation method on eight AMD Athlon 3GHz processors was used. A random starting point was used and optimisation procedure was stopped after 150000 CF evaluations.

5 Results

In this section, the foundry-provided original cell properties are compared to the results obtained with both optimisation runs (TABLE 1). The speed column represents the results of the speed run. All obtained properties are at least as good as in the original cell. Timing improvements of up to 77% were achieved. Interestingly, a 5.8% decrease in the power consumption was observed, which was not expected in the speed run. As expected, the final channel length is the smallest possible (350nm). To find the minimum, 149689 CF evaluations were needed. The

Table 1. D flip-flop optimisation results

property		original	speed	power
area	$[\mathrm{pm}^2]$	59.1	59.1	48.5
c to $q\uparrow$ delay	[ns]	5.17	2.95	4.19
c to $q\downarrow$ delay	[ns]	6.71	2.78	5.97
c to $qn\uparrow$ delay	[ns]	4.80	2.67	4.76
c to $qn\downarrow$ delay	[ns]	7.27	2.84	6.54
rn to qn delay	[ns]	4.19	1.82	3.28
rn to q delay	[ns]	6.03	1.60	4.17
c to $q\uparrow$ slope	[ns]	6.49	4.45	6.47
c to $q\downarrow$ slope	[ns]	9.96	2.61	7.88
c to $qn\uparrow$ slope	[ns]	6.41	3.03	6.38
c to $qn\downarrow$ slope	[ns]	9.95	3.11	9.60
rn to qn slope	[ns]	6.41	3.03	6.37
rn to q slope	[ns]	9.95	2.32	7.77
$d\uparrow$ to c setup	[ns]	1.23	0.625	1.03
$d\downarrow$ to c setup	[ns]	0.984	0.983	0.982
$d\uparrow$ to c hold	[ns]	0.566	0.324	0.548
$d\downarrow$ to c hold	[ns]	0.560	0.325	0.559
rn to c recovery	[ns]	1.27	0.665	1.08
min. width of c_{high}	[ns]	1.71	1.54	1.70
min. width of c_{low}	[ns]	1.30	0.970	1.30
min. width of rn	[ns]	0.811	0.375	0.463
power	[pAs]	6.22	5.86	4.96

$\uparrow$ and $\downarrow$ mark rising and falling edge

first circuit candidate that was better than the foundry-provided circuit was found in the 3285^{th} CF evaluation.

The power column in TABLE 1 summarises the results for the power run. All the obtained properties are again at least as good as in the original cell. The power consumption was reduced by 20% while timings improved by up to 43%. Interestingly, the final channel length was not the smallest possible (390nm). 121406 CF evaluations were needed to find the minimum. The first circuit candidate that was better than the foundry-provided circuit was found in the 3398^{th} CF evaluation.

Input capacitances were not taken into account in the cost function. But since the input-transistor gates became smaller in both runs, the input capacitances also decreased.

Only few thousand evaluations were needed to find the circuit candidate that was better than the foundry-provided cell in both runs. This means that the majority of iterations were spent on fine tuning which is usually achieved with a fast local optimisation method, e.g. [12]. Unfortunately, all attempts to speed up optimisation with a local method failed. This can be attributed to the harsh CF landscape [13]. To illustrate this, three cost cross-sections of CF along three transistor-channel widths are depicted in Fig. 3.

The numerical noise is clearly seen in Fig. 3. It is the result of a non-infinitesimal time-step in transient analysis. By reducing the time-step, the noise becomes smaller. Regardless of how small the noise was, we did not succeed in fine tuning the circuit with local optimisation methods. Therefore, we were forced to rely entirely on robust global methods. To maintain a sufficient accuracy, the time-step still had to be kept fairly small. This led to long transient-analysis runtimes which resulted in long optimisation runs. One optimisation run took five days on eight parallel processors.

6 Conclusion

Pre-designed foundry-provided digital cells represent a pool of available cells that are not meant to be altered at the transistor level. A case study described in the paper shows that a significant improvement can be achieved by using transistor-level optimisation techniques. The obtained results demonstrate speed improvements of up to 77% without increasing the power consumption. On the other hand, a 20% reduction in the power consumption without reducing the speed can be achieved. Using custom cells optimised to specific demands leads to faster circuits with a smaller power consumption. To make the proposed procedure efficient, the transistor-level optimisation should be automatically incorporated into the synthesis tools. Because of the noisy cost function landscape, long optimisation runs remain the main obstacle. Since the cell-extraction and synthesis tools were not available to the authors, the final back annotation to the synthesis was not done.

7 References

[1] H. Kaeslin, *Digital integrated circuit design: from VLSI architectures to CMOS fabrication*, Cambridge University Press, 2008.

[2] *HSPICE® simulation and analysis user guide*, Synopsys®, 2005.

[3] K.S. Kunderth, *The designer's guide to SPICE and Spectre*, Kluwer Academic Publishers, 1995.

[4] *Virtuoso® Spectre® circuit simulator user guide*, Cadence Design Systems, Inc., 2008.

[5] T. Tuma, Á. Bűrmen, *Circuit simulation with SPICE Opus, theory and practice*, Birkhäuser, 2009.

[6] D.E. Thomas, P.R. Moorby, *The Verilog hardware description language*, 5th ed. Kluwer Academic Publishers, 2003.

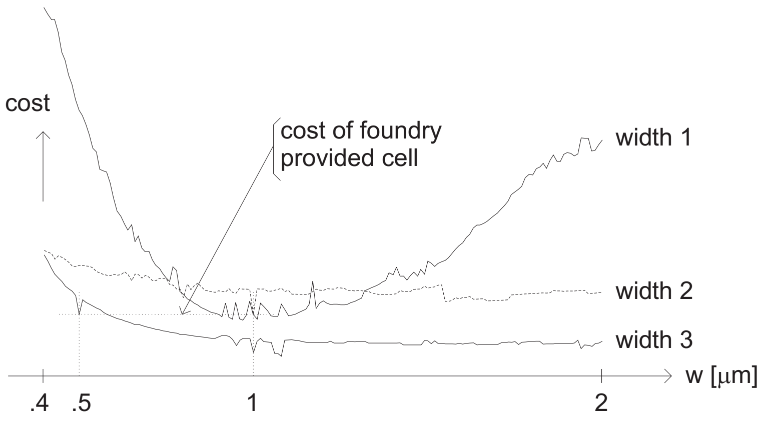

Figure 3. Cost function profiles for three transistor widths

[7] V.A. Pedroni, *Circuit design with VHDL*, Massachusetts Institute of Technology, 2004.

[8] P. Kurup, T. Abbasi, *Logic synthesis using Synopsys®*, 2nd ed. Kluwer Academic Publishers, 1997.

[9] H. Bhatnagar, *Advanced ASIC chip synthesis using Synopsys® Design Compiler™Physical Compiler™and PrimeTime®*, 2nd ed. Kluwer Academic Publishers, 2002.

[10] Á. Bűrmen et al., "Automated robust design and optimization of integrated circuits by means of penalty functions," *AEÜ, International Journal of Electronics and Communications*, vol. 57, no. 1, pp. 47–56, 2003.

[11] J. Olenšek, Á. Bűrmen, J. Puhan, T. Tuma, "DESA: a new hybrid global optimization method and its application to analog integrated circuit sizing," *Journal of Global Optimization*, vol. 44, no. 1, pp. 1–25, 2008.

[12] R. Hooke, T. Jeeves, "Direct search solutions of numerical and statistical problems," *Journal of the Association for Computing Machinery*, vol. 8, no. 2, pp. 212–229, 1961.

[13] Á. Bűrmen, I. Fajfar, T. Tuma, "Combined Simplex-Trust-Region optimization algorithm for automated IC design," *Proceedings of ECCTD07 European Conference on Circuit Theory and Design*, pp. 543–546, 2007.

ACKNOWLEDGEMENTS

The research has been supported by the Ministry of Higher Education, Science and Technology of the Republic of Slovenia under the program P2-0246 - Algorithms and optimisation methods in telecommunications.

Janez Puhan received his Ph.D. degree in Electrical Engineering from the University of Ljubljana in 2000. He is a teaching assistant at the Faculty of Electrical Engineering. His research interests include modelling, simulation and optimisation techniques in computer-aided integrated-circuit design.

Iztok Fajfar received his B.Sc., M.Sc., and Ph.D. degrees in electrical engineering from the Faculty of Electrical Engineering, University of Ljubljana, in 1991, 1994, and 1997, respectively. In 1991 he was researcher at the Joef Stefan Institute in Ljubljana. At the end of the same year he was granted a research position at the Faculty of Electrical Engineering, University of Ljubljana. Currently he holds the position of Associate Professor at the same faculty. He teaches several introductory and advanced courses in computer programming. He has also cooperated in several industrial software projects with Adacta and ICE Telecom. His research interests include design and optimisation of electronic circuits.

Tadej Tuma received his B.Sc, M.Sc. and Ph.D. degrees from the Faculty of Electrical Engineering, University of Ljubljana, in 1988, 1991, and 1995, respectively. He is a Full Professor at the same faculty where he teaches four undergraduate and three postgraduate courses. His research interests are mainly in the field of computer-aided circuit analysis and design.

Árpád Bűrmen received his Ph.D. degree in electrical engineering from the University of Ljubljana, Slovenia in 2003. Since 2005 he has been an assistant professor at the same university. His research interests include continuous and event-driven simulation of circuits and systems, optimization methods and their convergence theory and applications, and algorithms for parallel and distributed computation. He is one of the principal developers of the SPICE OPUS circuit simulator and has published over 20 papers in peer-reviewed journals.