Optimising digital circuit cells

Janez Puhan, Dušan Raič, Tadej Tuma, Sašo Tomažič and Árpád Bűrmen

Faculty of Electrical Engineering, University of Ljubljana

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

Abstract

Pre-designed cells, such as buffers, adders and flip-flops are provided by foundries and used in digital circuit design. Actual cell implementation at transistor level is not considered during the synthesis of a digital circuit. The paper describes four cases of transistor-level cell optimisation that can be employed to reach arbitrary customisation. Due to the landscape of the cost functions a global optimisation method was used. The results show that up to 80% improvement of the properties of pre-designed cells can be obtained.

Optimizacija gradnikov digitalnih vezij

Ključne besede: načrtovanje digitalnih integriranih vezij, splošni osnovni gradniki, sinteza digitalnih vezij, optimizacija na tranzistorskem nivoju

Povzetek

Načrtovalec sestavi digitalno integrirano vezje iz osnovnih gradnikov kot so medpomnilniki, seštevalniki, flip-flopi ipd. Knjižnico z naborom osnovnih gradnikov zagotovi izdelovalec integriranih vezij. Izvedba posameznega gradnika na tranzistorskem nivoju med postopkom načrtovanja ni več pomembna. Članek se ukvarja z možnostjo prilagoditve posameznega gradnika točno na zahteve, v katerih deluje. Opisani so štije primeri optimizacije osnovnih gradnikov na tranzistorskem nivoju. Pri tem je bila zaradi narave kriterijske funkcije uporabljena globalna optimizacijska metoda. Tako je možno doseči najboljšo prilagoditev gradnika na specifične zahteve. Rezultati kažejo do 80% izboljšanje glede na lastnosti splošnega gradnika, ki ga zagotovi izdelovalec integriranih vezij.

Introduction

Digital circuits are not designed at transistor level any more. The designers work with predesigned digital cells or blocks [1] such as buffers, logic gates, adders, latches and flip-flops etc. Those are then grouped into higher-level building elements like registers, decoders, comparators, counters, etc. The foundries usually provide a whole library of digital cells for their various IC manufacturing processes. It is customary that there are several versions of every cell like low and high voltage version, low power version, high speed version, and of course various combinations of those such as low power low voltage version etc. Every cell has a detailed description. Their characteristics like setup, delay, hold, minimum impulse width, recovery, etc. times are given for different output loads and input signal slopes. Power consumption, input capacitances, area, etc. are also given. Transistor-level simulations do not take place during the circuit design. Analog integrated circuit simulators [2, 3, 4, 5] are replaced with higher-level simulations [6, 7], where the circuit response is calculated according to cell descriptions. Since the simulation is no longer performed on transistor level, it is significantly faster. On the other hand the designers still like to do their final check with a classical integrated circuit simulator to verify the actual behaviour of the circuit before production takes place.

Various versions of a cell share the same transistor topology in most cases. They differ only in transistor sizing, usually only in transistor channel widths. Different versions of the same cell are actually a result of cell optimisation to different demands. We decided to test a foundry-provided digital transistor-level cell library to verify if it is possible to achieve better performance. We also wanted to see if cell optimisation makes sense, so that by automating it in the future one could generate an arbitrary cell version customised to particular needs of the circuit. Since we are searching for the most appropriate transistor channel widths, cell optimisation is performed on transistor level with an analog integrated circuit simulator [2, 3, 4, 5].

The left side of figure 1 shows digital Application Specific Integrated Circuit (ASIC) design flow [8, 9]. Design starts after architectural and electrical specifications of the circuit are set. Register Transfer Level (RTL) coding is used to implement the specifications. A Hardware Description Language (HDL) such as VHDL (Very High Speed Integrated Circuit *V*HSIC HDL) and Verilog (*Veri*fying *Log*ic) is used. The circuit described in HDL is simulated. It is of critical importance that appropriate input stimuli testing the circuit are provided for the simulation. The reduction of a HDL-described circuit into a gate-level netlist is called synthesis. Synthesis also performs gate-level optimisation with regard to timing constraints defining signal-clock relationships. The cell library contains foundry-provided descriptions of cells for synthesis. A cell description is extracted from transistor-level simulations. Verification validates the RTL code against the gate-level netlist. Static Timing Analysis (STA) double checks the timing constraints fed to the synthesis. Placement, Clock Tree (CT) insertion, and routing take place during the layout phase. Post layout verification and an additional STA are performed to check the result of the layout phase.

– figure 1 –

Our main idea (depicted on the right side of figure 1) is to introduce transistor-level cell optimisation into synthesis. Synthesis in step A produces a gate-level netlist of the circuit and a list of timing constraints for all gates. Cell implementations are selected from the foundry-provided library according to these constraints. What follows is our proposed optimisation step which sizes the topologies of selected foundry-provided cells according to the constraints obtained in step A of the synthesis. Some cells selected in step A barely fulfil the timing requirements while others have a broad safety margin. Former ones can be optimized for speed while latter ones can be optimized for power consumption, without affecting circuit's performances. Cell descriptions are then extracted from transistor-level simulations of optimized cells resulting in a customised cell library. This library is then used as input to synthesis in step B which selects appropriate cells from the customised library for the gate-level topology obtained in synthesis step A. A successful synthesis in step B (back annotation) validates the design that uses customised cells instead of foundry-provided ones.

Cost definition

The criterion is a mathematical function, which evaluates a circuit candidate or a particular set of transistor channel widths $\mathbf{w}$. It is calculated from the circuit's response at given transistor widths. The better the candidate, the lower the criterion value. By establishing a mathematical criterion or cost function, one can always decide, which circuit is better [10].

Usually only the transient response matters for a digital circuit. It reveals the circuit's dynamics, general time domain behaviour and power consumption. Only one transient analysis per circuit candidate is needed. Since the goal for digital circuits is always the same (as fast as possible for as little power as possible), there are only few properties of interest:

- chip area,
- various time measurements such as slopes, delays, minimal impulse widths, setup, hold and recovery times, etc., and
- power consumption, which can be expressed as time integral of power supply current.

The first property is defined by the channel widths and the others can be obtained from the transient response.

A goal value g_i for every property has to be chosen. Each measured property contributes a portion to the cost function value. Until the goal is not reached the contribution is proportional to the goal violation. When the goal is reached or even exceeded it becomes negative. For this purpose we define the contribution $c_i(x_i)$ (1) of a particular measurement $x_i(\mathbf{w})$. It is a broken linear function depicted in figure 2.

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

– figure 2 –

Since all measurements listed above have to be as low as possible, only one type of contribution function $c_i(x_i)$ (1) is sufficient. The final cost function is a sum of n contributions (2).

$$c(\mathbf{x}) = \sum_{i=1}^{n} c_i(x_i) \tag{2}$$

With appropriate settings of goals g_i, trade-off (t_i), and penalty weights (p_i) optimisation for an arbitrary version of the circuit (high speed or low power version etc.) can be achieved. Individual measurements can be made more or less important by adjusting t_i and p_i. The optimal circuit parameters $\mathbf{w}_{\mathrm{opt}}$ are those where the cost function (2) has its global minimum (3).

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

Also if $t_i \ll p_j$ holds for every pair $i \neq j$, the cost function will guarantee that the first objective of the optimisation is to achieve all the goals. If only one of the goals is not achieved, its contribution will be very high compared to other contributions. Therefore the optimisation process tends to fulfil all the goals. Not achieving one of the goals cannot be compensated by exceeding others.

Circuit candidates, that do not converge, need special treatment regarding the cost function evaluation. Such pathological candidates normally appear during the optimisation process and in general cannot be avoided. When the simulation fails, transient response is not available and the measurements $\mathbf{x}$ cannot be determined. When a particular measurement x_i is not known, its contribution $c_i(x_i)$ will be set to some large value c_{max_i}. A pathological candidate produces a huge cost value and represents a bad try. The same goes for semi-pathological candidates, for which the transient response is available, but the circuit does not behave as expected. In such cases one or more required measurements still cannot be determined. For instance a slope cannot be measured if there is no edge in the response.

To speed up the optimisation process semi-pathological circuits are additionally penalised by auxiliary measurements. Auxiliary measurements enforce the correct transient response. With digital circuits this is again a fairly simple task. Assuming proper behaviour, the state of the circuit at particular time points is known in advance. For instance, if some node voltage at some time point should be high but is not, an auxiliary measurement will considerably increase the cost value. On the other hand an auxiliary measurement will not interfere with the cost value when the selected node voltage fulfils the expectations. To penalise semi-pathological circuits auxiliary measurements have large penalty weights. Their trade-off weights are set to zero to eliminate them from the cost function when the circuit behaves as expected.

Benchmark circuits

Four pre-designed foundry-provided digital cells were used as benchmarks. The four cells are half adder (fig. 4), full adder (fig. 5), and D flip-flop with and without scan inputs (figs. 6 and 7). In our opinion they represent a fair sample of the cell library. Although there are many different cells, the transistor configurations remain the same. The most characteristic transistor arrangements are included in the selected cells. The foundry-provided transistor models for the digital cell library are proprietary and cannot be revealed.

A benchmark cell is put into a test bench circuit providing input signals, power supply voltage and output loads. Figure 3 shows a test bench circuit ready for simulation. Values of input signals' slopes, power supply voltage, and output capacitances vary according to the operating corner conditions described later.

– figure 3 –

For every cell the transistor topology and input test signals are given. Time-domain measurements are described. Together with chip area and power consumption they represent the measurements x_i, that contribute to the cost function (2). The chip area is the sum of all transistor areas, and the power consumption is the time integral of the power supply current.

– figure 4 –

There were 24 time-domain measurements for the half adder case. 12 input to output delays and 12 output signal slopes were taken into account. For instance a delay from falling b to rising s edge is depicted in figure 4. The figure also depicts a rising edge slope of co at rising a.

– figure 5 –

There were 24 time-domain measurements for the full adder case. 12 input to output delays and 12 output signal slopes were taken into account. A delay from rising ci to rising s edge is depicted in figure 5. A falling edge slope of co at rising b is also shown.

– figure 6 –

There were 20 measurements for the D flip-flop case without scan inputs. Beside 6 input to output delays and 6 output signal slopes, output setup, input hold, recovery after reset, minimal clock, and minimal reset impulse duration were taken into account. Time-domain measurements excluding the delays and slopes are depicted in figure 6. A delay from rising c to falling q edge and a falling edge slope of qn at rising c are shown for illustration.

– figure 7 –

There were 26 measurements for the D flip-flop case with scan inputs. Beside 6 input to output delays and 6 output signal slopes, output setup, input hold, recovery after reset, minimal clock, and minimal reset impulse duration were taken into account. Time measurements, excluding the delays and slopes, are depicted in figure 7. A delay from rising c to falling q edge and a falling edge slope of qn at rising c are shown for illustration.

The delay was defined as the time between the points where the input voltage reaches its 50% level until the output voltage reaches its 50% level. The slope was defined as the time between the 10% and 90% signal level.

Optimisation

Two optimisation runs were performed for every cell. In the first run the goal was to obtain a fast circuit, but at the same time the power consumption should not exceed that of the foundry-provided cell (optimisation for speed). In the second run the goal was to decrease the power consumption while keeping the timings at least as good as those exhibited by the foundry-provided cell (optimisation of power consumption).

The two optimisation runs differed in trade-off (t_i) and penalty (p_i) weights in (1). Actually only the power consumption trade-off weight was varied making the measurement more or less important in comparison to area and time measurements. The original cell properties were used as goals g_i. The penalty weights p_i ($\gg t_j$) were identical for all runs.

There were 65 process and operating condition corners taken into account. They consisted of:

- four process corners (worst power (wp), worst speed (ws), worst one (wo), and worst zero (wz)),
- two temperatures ($-25\,^\circ$C and $105\,^\circ$C),
- two power supply voltages (2V and 3.3V for adders and 3V and 3.6V for flip-flops),
- 10fF and 220fF output loads (and an additional 230fF output load for adders) and

- 60ps and 4ns input signal slopes (and additional 90ps and 6ns slopes for adders).

The typical corner was added to the 64 extreme combinations, resulting in 65 corners.

Out of 65 corners only a few are important. For instance power consumption is always the highest in wp/−25 °C/3.3V(or3.6V)/220fF/4ns corner. For this reason evaluating power consumption in other corners is needless. Number of corners was therefore significantly reduced since only the worst corner measurement value is considered in the cost function. Performing analyses and evaluating measurements was unnecessary in most corners. So, only three[1] corners for the adders and four[2] corners for the flip-flops were taken into account during the optimisation. The final results were verified across all 65 corners.

Transistor channel widths were independent optimisation variables with explicit constraints from 400nm to 2um (3um in half adder case). All transistors shared the same channel length, which resulted in one additional optimisation variable with explicit constraints from 350nm to 2um. Therefore the number of optimisation variables was equal to the number of transistors plus one, which means 15 for half adder, 29 for full adder, 33 for D flip-flop without scan inputs, and 41 for D flip-flop with scan inputs. The channel length variable was added for generality because we expected that it tends to be as small as possible. The optimisation process did not confirm that, since the resulting length was not equal to the lower constraint (350nm) in all runs.

Due to many optimisation variables and a harsh cost function landscape, which will be explained later, a robust global optimisation method was used. We decided for Parallel Simulated Annealing with Differential Evolution (PSADE) [11], since it is able to run on several processors in parallel. We used eight AMD Athlon 3GHz processors. The method was started from a random initial point and was stopped after 150000 evaluations.

[1] wp/−25 °C/3.3V/220fF/4ns, ws/105 °C/2V/230fF/90ps, ws/105 °C/2V/230fF/6ns

[2] wp/−25 °C/3.6V/220fF/4ns, ws/105 °C/3V/10fF/60ps, ws/105 °C/3V/220fF/60ps, ws/105 °C/3V/220fF/6ns

Results

The results of all eight optimisation runs are listed in tables 1 to 4. The foundry-provided original cell properties are compared to the results from the speed and power optimisation runs.

– table 1 –

Table 1 summarises results for the half adder benchmark circuit. The results show that both optimisation runs found a solution with properties (area, timings and power consumption) at least as good as in the original foundry-provided cell. The speed optimisation run resulted in improvements up to 80% without an increase in power consumption. Vice-versa in power optimisation an improvement of 19% without an increase in timings was obtained. Symbols ↑=↓ in figure 4 depict transistor channel width changes after both optimisation runs with respect to the foundry-provided values. The left symbol corresponds to speed optimisation and the right symbol corresponds to power optimisation. Speed optimisation resulted in final channel length of 350nm (as expected). Interestingly the final channel length was to 450nm in the power optimisation run. Input capacitances were not taken into account in the cost function. But since input transistor gates became smaller the input capacitances also decreased. To find a minimum 143347/125519 evaluations were needed in the speed/power optimisation run. The first candidate circuit, that was better than the foundry-provided original, was found after 1661/1412 evaluations.

– table 2 –

The results for the full adder circuit are in table 2. Again both optimisation runs found a solution at least as good as that provided by the foundry. The speed optimisation resulted in timing improvements up to 77% while keeping the same power consumption. Power optimisation resulted in an improvement up to 22%. Transistor channel width changes during both optimisations are depicted in figure 5. The resulting channel length was 350nm/450nm. 139817/91550 candidate circuits were evaluated for the speed/power optimisation. The first circuit candidate, better than the foundry-provided circuit, appeared in the 3556th/2286th evaluation. Since input transistor gates became smaller the input capacitances decreased.

– table 3 –

– table 4 –

Similar results were obtained for the D flip-flop circuits (tables 3 and 4). Speed optimisation resulted in speed improvement up to 77% while keeping the same power consumption for both cases. Power optimisation resulted in 20% improvement for both cases. Transistor channel width changes during both optimisations are depicted in figures 6 and 7. The final channel length was 350nm/390nm after 149689/121406 evaluations (speed/power optimisation) of the D flip-flop without scan inputs. For the D flip-flop with scan inputs the final channel length was 350nm/420nm after 145247/147554 evaluations. The first circuit that performed better than the foundry-provided cell was found in the 3285th/3398th and 4104th/3505th evaluation, respectively. Since input transistor gates are again smaller than in the original cell the input capacitances were decreased by the optimisation.

For all of the test circuits only a few thousand evaluations were needed for finding the first circuit that performed better than the foundry-provided cell. Most of the 150000 available iterations were spent to fine tune the circuit. At first glance this should be an easy task for a fast local optimisation method [12]. For that reason we used PSADE to provide a useful initial point for the local method. We further speculated that the speed optimised cell should be a suitable initial guess for the power optimisation and vice-versa. Unfortunately all our attempts to accelerate the optimisation by using a local method failed. The answer lies in the harsh cost function landscape [13]. Three cost function profiles for the D flip-flip with scan inputs are depicted in figure 8. Each profile represents a cross section of the cost function along one transistor channel width. All other widths had foundry-provided values.

– figure 8 –

From profiles it can be seen that the main cause for the failure of the local method is numerical noise. The noise is a result of limited numerical accuracy and non-infinitesimal time-step in transient analysis. By reducing the time-step numerical noise becomes smaller. Even with a fairly small time-step the cost function landscape still caused problems for local methods. Because local optimisation methods failed on our circuits we were forced to rely entirely on a robust global method. Due to the small time-step the number of calculated points in the transient analysis becomes very high. This leads to prolonged simulations and long optimisation runs. Despite using several processors in parallel one optimisation run took one day for the smallest half adder circuit up to a week for the D flip-flop with scan inputs.

Summary

Pre-designed foundry-provided digital cells are designed to be general. They are not meant to be altered at transistor level and represent a pool of cells available to the synthesis. However they can be significantly improved by changing transistor channel widths and lengths. Using transistor-level optimisation techniques we managed to get up to 80% faster cells at the same power consumption and saved up to 20% of power at the same cell speeds. Therefore instead of using only foundry-provided cells each cell in a larger digital or mixed circuit could be independently optimised to satisfy the specific demands for the cell. A circuit as a whole would become faster with smaller power consumption. By automating the transistor-level optimisation procedure during the synthesis the entire process of digital circuit design would became more efficient. The main obstacle at the moment is the time needed for the optimisation. Noisy cost functions are the main reason why fast local methods cannot be used to speed up the optimisation. Also detailed property extraction of the optimised cells and back annotation of the synthesis was not done since the extraction and synthesis tools were not available to the authors.

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, fifth edition. *Kluwer Academic Publishers*, 2003

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

[8] P. Kurup, T. Abbasi. Logic Synthesis Using Synopsys®, second edition. *Kluwer Academic Publishers*, 1997

[9] H. Bhatnagar. Advanced ASIC Chip Synthesis Using Synopsys®Design Compiler$^{\text{TM}}$Physical Compiler$^{\text{TM}}$and PrimeTime®, second edition. *Kluwer Academic Publishers*, 2002

[10] Á. Bűrmen, D. Strle, F. Bratkovič, J. Puhan, I. Fajfar, T. Tuma. *Automated Robust Design and Optimization of Integrated Circuits by Means of Penalty Functions*, AEÜ, International Journal of Electronics and Communications, Volume 57, No. 1, pages: 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, Volume 44, No. 1, pages: 1-25, 2008

[12] R. Hooke, T. Jeeves. *Direct Search Solutions of Numerical and Statistical Problems*, Journal of the Association for Computing Machinery, Volume 8, No. 2, pages: 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, pages: 543-546, 2007

Asst. Prof. Dr. Janez Puhan, univ.dipl.ing.el.
Faculty of Electrical Engineering, University of Ljubljana
Tržaška cesta 25, 1000 Ljubljana
tel.: (01) 4768 322, fax: (01) 4264 630
e-mail: janez.puhan@fe.uni-lj.si

Assoc. Prof. Dr. Dušan Raič, univ.dipl.ing.el.
Faculty of Electrical Engineering, University of Ljubljana
Tržaška cesta 25, 1000 Ljubljana
tel.: (01) 4768 324, fax: (01) 4264 630
e-mail: dusan.raic@fe.uni-lj.si

Prof. Dr. Tadej Tuma, univ.dipl.ing.el.
Faculty of Electrical Engineering, University of Ljubljana
Tržaška cesta 25, 1000 Ljubljana
tel.: (01) 4768 329, fax: (01) 4264 630
e-mail: tadej.tuma@fe.uni-lj.si

Prof. Dr. Sašo Tomažič, univ.dipl.ing.el.
Faculty of Electrical Engineering, University of Ljubljana
Tržaška cesta 25, 1000 Ljubljana
tel.: (01) 4768 432, fax: (01) 4264 630
e-mail: saso.tomazic@fe.uni-lj.si

Asst. Prof. Dr. Árpád Bűrmen, univ.dipl.ing.el.
Faculty of Electrical Engineering, University of Ljubljana
Tržaška cesta 25, 1000 Ljubljana
tel.: (01) 4768 322, fax: (01) 4264 630
e-mail: arpad.buermen@fe.uni-lj.si

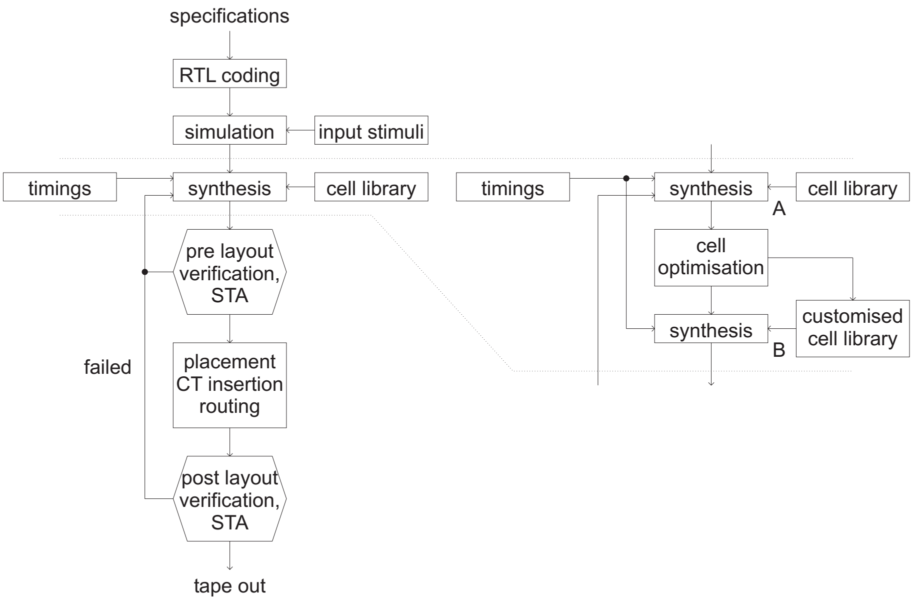

Figure 1: Digital ASIC design flow

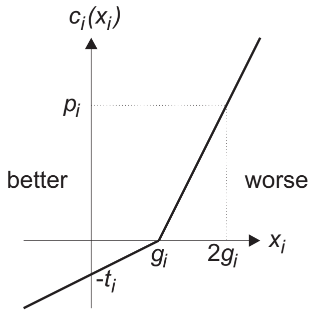

Figure 2: Measurement contribution function

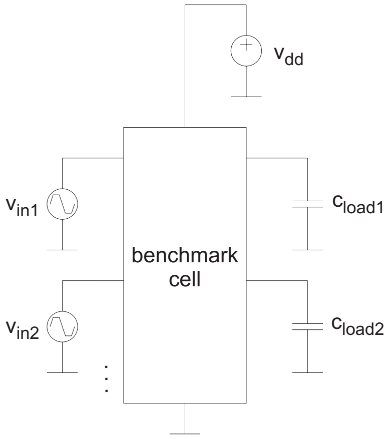

Figure 3: Test bench circuit

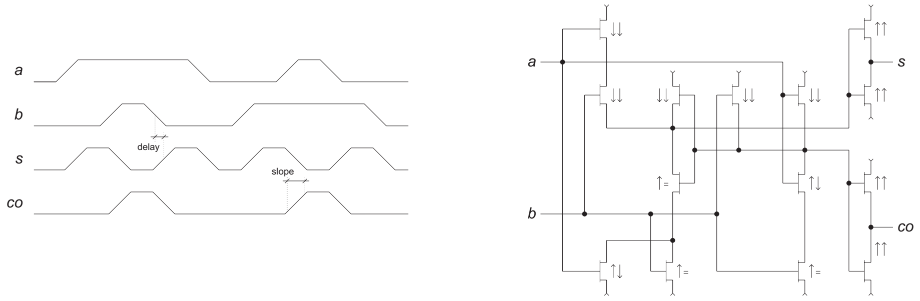

Figure 4: Half adder

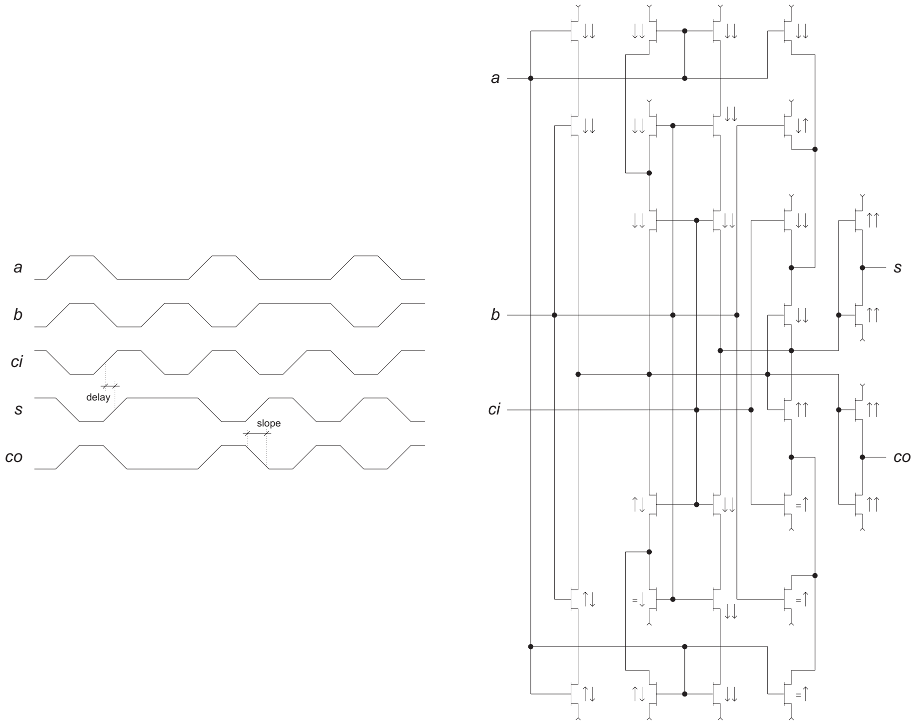

Figure 5: Full adder

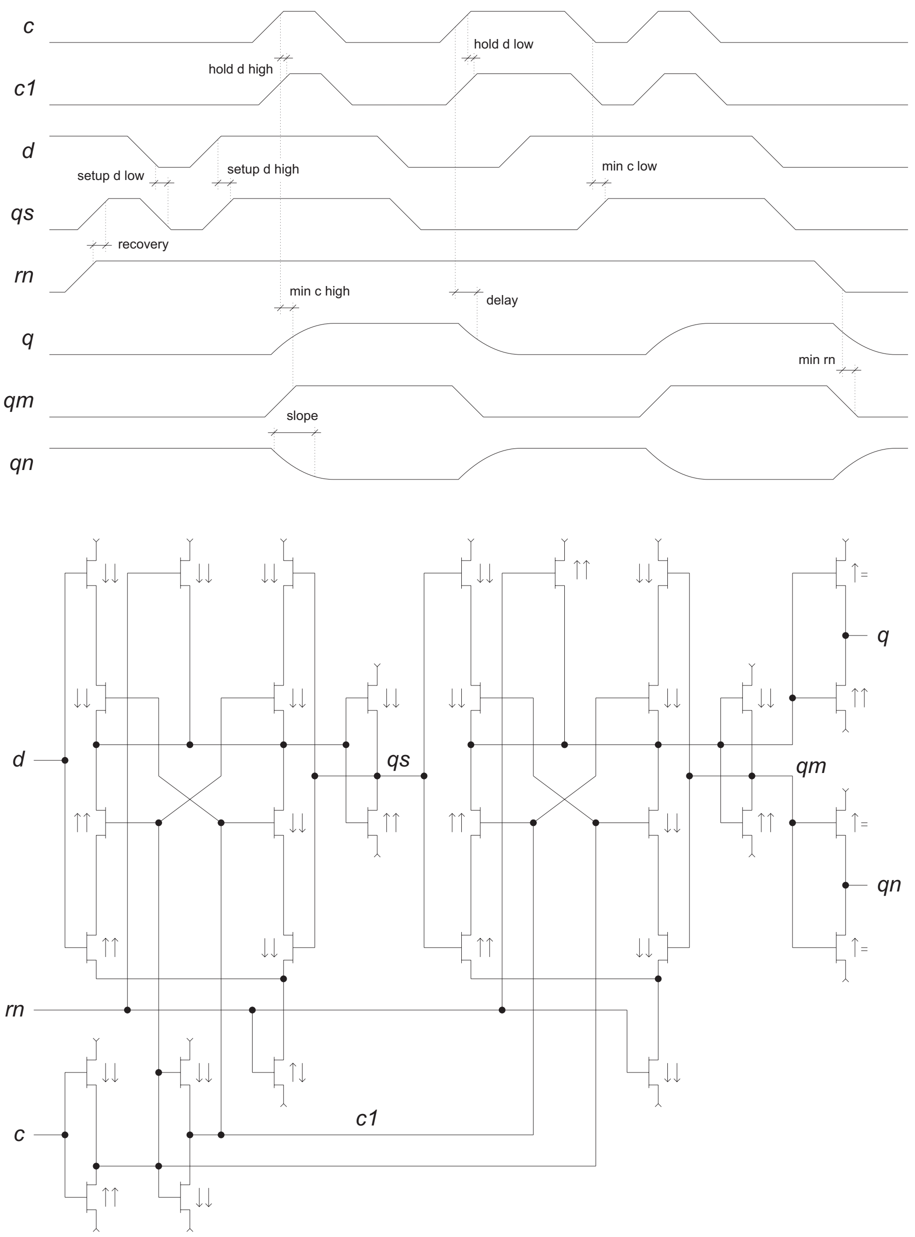

Figure 6: D flip-flop without scan inputs

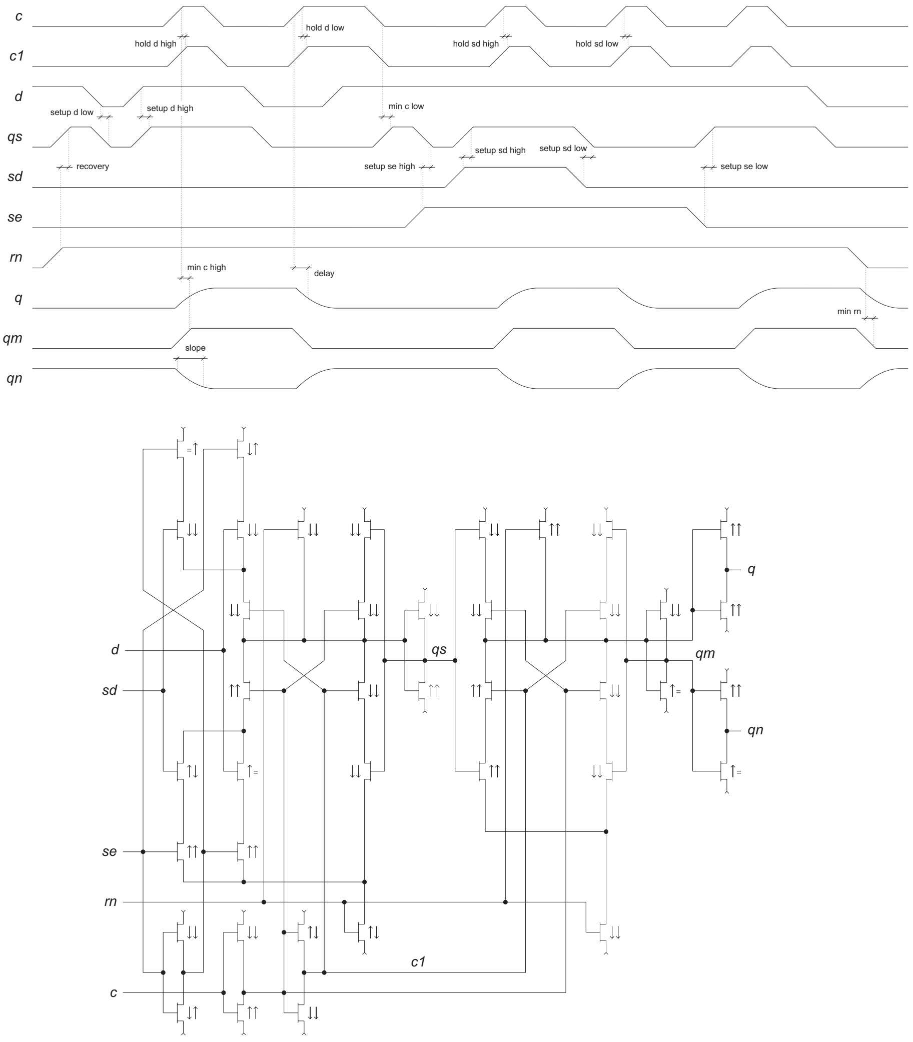

Figure 7: D flip-flop with scan inputs

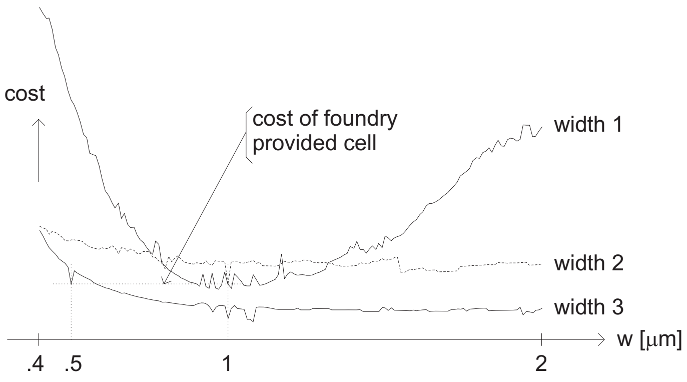

Figure 8: Cost function profiles for three transistor widths

	foundry	high speed	low power
area [pm^2]	47.6	47.6	33.5
delays[a] [ns]	7.28/10.4/7.11/10.6	3.63/4.86/3.22/5.28	7.25/9.85/6.92/10.3
delays[b] [ns]	8.78/8.31/10.9/9.46	3.95/5.65/4.57/6.62	8.13/8.21/8.61/9.41
delays[c] [ns]	7.56/9.44/10.7/9.36	3.78/6.18/4.15/6.65	6.54/9.42/8.28/9.30
slopes[a] [ns]	10.1/15.4/10.1/15.5	4.27/3.61/4.26/3.62	8.23/13.0/8.24/13.0
slopes[b] [ns]	10.1/10.1/15.4/15.4	3.85/3.81/3.14/3.57	8.31/8.17/7.28/7.66
slopes[c] [ns]	10.1/10.1/15.4/15.4	3.85/3.83/3.13/3.57	8.18/8.31/7.28/7.66
power [pAs]	5.66	5.63	4.58

[a] a to $co\uparrow$ / a to $co\downarrow$ / b to $co\uparrow$ / b to $co\downarrow$

[b] $a\uparrow$ to $s\uparrow$ / $a\downarrow$ to $s\uparrow$ / $a\uparrow$ to $s\downarrow$ / $a\downarrow$ to $s\downarrow$

[c] $b\uparrow$ to $s\uparrow$ / $b\downarrow$ to $s\uparrow$ / $b\uparrow$ to $s\downarrow$ / $b\downarrow$ to $s\downarrow$

Table 1: Results for the half adder

	foundry	high speed	low power
area [pm^2]	46.9	46.9	43.6
delays[a] [ns]	8.08/10.4/8.79/12.2	4.72/4.88/6.33/6.55	8.06/10.2/8.63/12.2
delays[b] [ns]	8.48/10.7/8.85/12.6	4.64/5.05/6.51/6.36	8.44/10.5/8.84/11.4
delays[c] [ns]	8.72/10.9/8.68/12.8	4.95/5.51/6.83/6.52	8.70/10.5/8.65/12.2
slopes[a] [ns]	10.2/15.5/10.2/15.5	4.87/3.71/4.81/3.51	8.81/10.7/7.00/10.6
slopes[b] [ns]	10.2/15.5/10.1/15.4	4.78/3.57/4.79/3.48	8.76/10.6/6.96/10.5
slopes[c] [ns]	10.2/15.5/10.1/15.5	4.80/3.60/4.83/3.62	8.77/10.5/6.95/10.6
power [pAs]	7.46	7.23	5.83

[a] a to $co\uparrow$ / a to $co\downarrow$ / a to $s\uparrow$ / a to $s\downarrow$

[b] b to $co\uparrow$ / b to $co\downarrow$ / b to $s\uparrow$ / b to $s\downarrow$

[c] ci to $co\uparrow$ / ci to $co\downarrow$ / ci to $s\uparrow$ / ci to $s\downarrow$

Table 2: Results for the full adder

	foundry	high speed	low power
area [pm^2]	59.1	59.1	48.5
delays^a [ns]	5.17/6.71/4.80/7.27	2.95/2.78/2.67/2.84	4.19/5.97/4.76/6.54
delays^b [ns]	4.19/6.03	1.82/1.60	3.28/4.17
slopes^a [ns]	6.49/9.96/6.41/9.95	4.45/2.61/3.03/3.11	6.47/7.88/6.38/9.60
slopes^b [ns]	6.41/9.95	3.03/2.32	6.37/7.77
setup^c [ns]	1.23/0.984	0.625/0.983	1.03/0.982
hold^c [ns]	0.566/0.560	0.324/0.325	0.548/0.559
recovery^d [ns]	1.27	0.665	1.08
width^e [ns]	1.71/1.30/0.811	1.54/0.970/0.375	1.70/1.30/0.463
power [pAs]	6.22	5.86	4.96

[a] c to $q\uparrow$ / c to $q\downarrow$ / c to $qn\uparrow$ / c to $qn\downarrow$

[b] rn to qn / rn to q

[c] $d\uparrow$ to c / $d\downarrow$ to c

[d] rn to c

[e] minimum impulse width of c_{high} / c_{low} / rn

Table 3: Results for the D flip-flop without scan inputs

	foundry	high speed	low power
area [pm^2]	73.3	73.3	63.5
delays[a] [ns]	5.17/6.71/4.80/7.27	2.97/2.73/2.69/2.91	4.31/6.25/4.79/6.57
delays[b] [ns]	4.19/6.03	1.82/1.55	3.24/4.41
slopes[a] [ns]	6.49/9.96/6.41/9.95	4.56/2.57/3.03/3.33	6.46/8.18/5.79/9.65
slopes[b] [ns]	6.41/9.95	3.03/2.28	5.79/8.05
setup[c] [ns]	2.01/1.44/1.80/1.46	1.01/1.27/1.27/1.20	1.71/1.44/1.80/1.45
hold[c] [ns]	0.563/0.561/0.566/0.561	0.260/0.263/0.258/0.264	0.553/0.560/0.555/0.560
recovery[d] [ns]	2.13	1.11	1.84
width[e] [ns]	1.70/1.95/8.15	1.49/1.23/3.91	1.70/1.93/0.621
power [pAs]	6.22	5.86	4.96

[a] c to q↑ / c to q↓ / c to qn↑ / c to qn↓

[b] rn to qn / rn to q

[c] d↑ to c / d↓ to c / sd↑ to c / sd↓ to c

[d] rn to c

[e] minimum impulse width of c_{high} / c_{low} / rn

Table 4: Results for the D flip-flop with scan inputs