Eindhoven University of Technology

MASTER

Standard cell library tuning for variability tolerant designs

Fabrie, S.A.J.

Award date:
2013

Link to publication

Disclaimer
This document contains a student thesis (bachelor's or master's), as authored by a student at Eindhoven University of Technology. Student theses are made available in the TU/e repository upon obtaining the required degree. The grade received is not published on the document as presented in the repository. The required complexity or quality of research of student theses may vary by program, and the required minimum study period may vary in duration.

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain

Standard Cell Library Tuning for Variability Tolerant Designs

Sebastien Fabrie
Dept. of Mathematics and Computer Science
Eindhoven University of Technology
Eindhoven, The Netherlands
sebasfabrie@gmail.com

Abstract— **In today's semiconductor industry we see a move towards smaller technology feature sizes. These smaller feature sizes pose a problem in terms of process controllability, e.g. mismatch between identical cells on a single die known as local variation. In this paper a library tuning method is proposed which makes a smart selection of cells in a standard cell library to reduce the design's sensitivity to local variability. This results in a robust IC design with an identifiable behavior towards local variations. Experimental results performed on a widely used microprocessor design synthesized for a high performance timing show that we can achieve a timing spread reduction of 37% at an area increase cost of 7%.**

Index Terms—**Standard cell library tuning, Local variation, Mismatch variation, Intra-die variation, Statistical library, Variability tolerant design, Gate delay variation.**

I. Introduction

In the semiconductor industry a trend has been going on for years to scale down transistor sizes in order to fit more transistors on the same die area, an evolution known as Moore's law [1]. The downscaling of devices makes the physical manufacturing process of chips more difficult to control and therefore the outcome is less predictable. Two effects which are introduced during the chip manufacturing process are global and local variations.

Global (or inter-die) variation is the phenomenon where two identical standard cells differ in characteristics from chip-to-chip, wafer-to-wafer or lot-to-lot. The chip-to-chip variation is largely due to the placement of the chip on a single wafer. A chip which was placed near the edge of the wafer may have a slightly different behavior than a chip in the center of a wafer [2]. For the wafer-to-wafer and lot-to-lot variation the same idea applies. The uncertainty in the manufacturing process of a wafer or a slight temperature difference between the production of a lot (a batch of wafers) results in global variations. Since the effect of the manufacturing process is known, it can be modeled and accounted for during the design of an integrated circuit (IC). The foundry which produces the physical chips provides transistor models for worst-case, typical, cross-corner, and best-case behavior. Commonly, to get the highest yield a designer will use the worst-case model and adjust the design to function properly under these conditions [3].

Local (or intra-die) variation is the variation between two identical standard cells on the same die. Compared to global variations, the cells on a die have a much lower distance to each other and are therefore much less susceptible to wafer surface in-homogeneity, temperature differences, etc. The sources for local variation can be found in e.g. the etching process, doping concentration and lithography. For instance, small differences in doping concentration make individual transistors of cells behave with slight variations in terms of timing, transitions but also, for example, power. Cell parameters such as orientation, channel length and oxide thickness make the cell more or less sensitive towards local variations.

Library tuning is a method by which a smart selection of standard cells from a logic library is made to create a subset of cells which have more desirable properties. There are examples for which library tuning is used to reduce soft errors [4], here different subsets are generated and used in synthesis to analyze their behavior to soft errors. Library tuning as used in [5] is meant to iteratively remove cells from the library to pursue a synthesis speed-up. In paper [6], library tuning is used to reduce power by only using a small subset of cells in the physical design.

To the extent of the authors' knowledge there is currently no publication on using library tuning to reduce the local variation of the data-path of a digital design. In this paper we introduce a library tuning method aimed at reducing the local variation of a design. Rather than creating a subset based on the exclusion of cells, a subset is created by constraining the cell's use to a range of slew and loading capacitance conditions where the cell exhibits the lowest timing-spread. This slew-load range constraining is passed over to the synthesis tool such that the choice of cells during synthesis is restricted to the allowed slew-load ranges. The result is a design that has diminished variability in all paths.

The remainder of this paper presents background theory, details of the proposed approach, and finally, experimental synthesis results using a small real-world microcontroller design (20K gates).

II. Library characterization

The behavior of a logic cell depends on its logic function, the input slew and the output load which the cell needs to drive. Depending on these conditions one can characterize a cell which results in look-up tables (LUTs) for e.g. the cell delay, cell transition and power dissipation.

By assigning different input slopes and output loads to a cell, a look-up table can be formed from which the synthesis tool can interpolate the required timing information. The LUT cannot of course contain every possible combination of input slew and output load and thus a selection is made. The selection for the input slew is based on the possible input slopes of a design, ranging from steep to shallow with an adequate number of slopes in-between. Cells with low drive strengths are not designed to drive a high output load since the output slope would be flat. Cells with high drive strength are used for bigger loads. Consequently, this also implies that the output load range for cells with different drive strengths is different. For complex cells the same procedure is used but more tables are generated depending on the timing arcs (possible paths between inputs and outputs). For example, a two input AND gate has two possible timing arcs and thus two cell rise, fall, and transition tables.

The above outlined characterization is done for all cells in a standard cell library and results in a library (.lib) file with timing information [7]. The library file also contains information about the power consumption of the cell for different transition stages but these are not discussed in this paper.

III. Local Variation Metric

Local variations are taken into account during the design of an integrated circuit by adding an uncertainty factor to the desired clock period. The synthesis process takes the clock uncertainty into account to ensure that even the worst-case dies will still meet their timing constraints. If one could reduce the impact of local variation, one could also reduce the clock uncertainty. A lower clock uncertainty means that the desired clock period can be decreased resulting in a faster design. A reduction of the local variation of a design leads to a more robust design.

To be able to tune the library we need a metric which reflects the cell's sensitivity to local variations. For our purposes we are mostly interested in the consequence of local variation on the setup timing behavior of a cell, but the methods which will be described can also be adjusted to measure the influence of local variation on other properties, such as transition power.

The local variation of the cell is modeled by a normal probability distribution. The distribution can be constructed by simulating the cell's timing behavior with modified transistor models that take into account the variability of the process. This is generally done through Monte Carlo simulations. The central limit theorem indicates that at least 30 independent random samples (*N = 30*) are needed to have a crude approximation of a normal distribution [8]. Monte Carlo sampling is used since the runtime and accuracy are dependent on the number of runs and not on the number of varying parameters.

Since we know that the approximated probability distribution is a normal distribution we can argue that that the dispersion of the distribution is a good metric. If a cell has a stretched distribution, or in other words a high dispersion, the timing behavior of a cell can vary a lot around the mean. The lower the dispersion, the closer the extreme values stay to the mean and hence the lower the cell proves to be sensitive for local variation. In industry, the coefficient of variation is used as a measure for gate delay variation [9, 10, 11]. The coefficient of variation, or variability, is the ratio between the standard deviation and the mean of a probability distribution (see eq. (1)). It is a normalized measure for the dispersion of the distribution or in other words, it shows how stretched a distribution is in relation to the mean.

$$Variability = \frac{Std.dev.}{Mean} = \frac{\sigma}{\mu} \qquad (1)$$

Working with the variability as a metric for the local variation of cell timing gives rise to possible selection problems. Consider the case depicted Figure 1. The left distribution has a mean of 0.5, standard deviation of 0.01 and a variability of 0.01 / 0.5 = 0.02. The right distribution has a mean of 5, standard deviation of 0.1 and a variability of 0.1 / 5 = 0.02. We see that both distributions have an identical variability but the left distribution is more desirable in terms of local variation because of the lower dispersion. This shows that the variability is not the desired selection metric for the local variation.

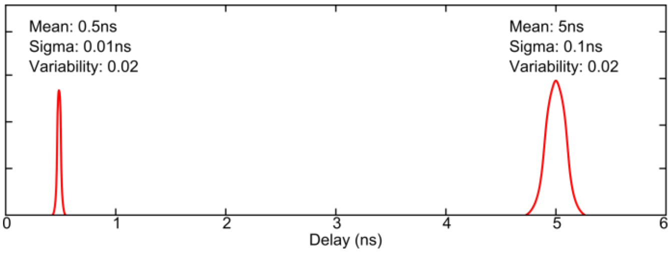


Fig. 1. Variability comparison where the left and right PDF have an identical variability of 0.02 but different std. deviations.

Another possible metric for the dispersion of a distribution is the standard deviation. The standard deviation shows the amount of dispersion with respect to the mean. A high standard deviation indicates that there is a significant spread in the samples from the distribution compared to the mean, in other words a high dispersion. If we look back at Fig. 1 we see that the left distribution has a smaller standard deviation (sigma) than the right one and a correct selection can be made based on this observation. The mean will be accounted for during the synthesis of a design where the synthesis tool tries to reduce the timing of a design by choosing low delay mean cells. In the remainder of this paper we will use the standard deviation as a metric for the local variation.

IV. Library cell local variation

Without loss of generality, assume that *N* distinct libraries are created from a Monte Carlo sampling that includes the effect of local variations. Let us now combine each of the libraries into a single statistical equivalent done by looking at individual entries of all the cell's tables. Each entry of a table denotes the same aspect of a cell (same slew and load conditions) and only differs across the libraries by the added effect of local variation. Figure 2 illustrates this process for an inverter cell. From this table we only consider the first entry which has a certain slew and load condition (marked entry in Fig. 2). The entry is extracted from the *N* libraries and the values are put into a temporary table with size *N*. From this table the mean and standard deviation are calculated and stored in the same position of the statistical library as where they were extracted from in the original table. When each entry and table is processed, the described approach results in a library file with identical tables as a nominal library but which contains local variation statistics instead.

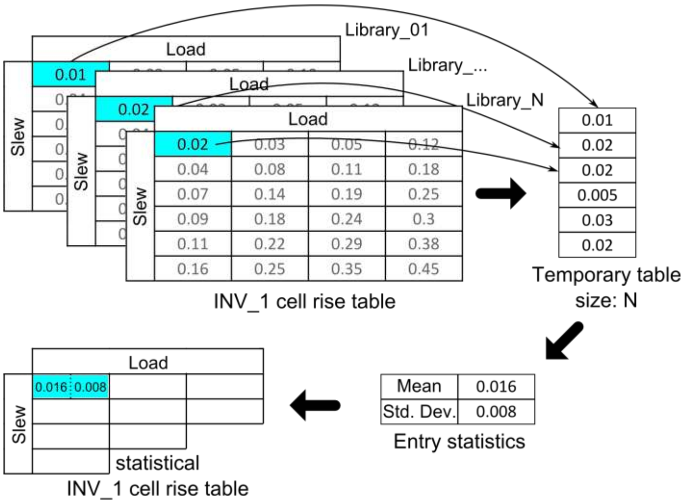


Fig. 2. Process of creating a statistical library. For a gate, a single LUT is considered. From the LUT a single entry is loaded into a temporary table across the available libraries. The last step is to extract the sigma and mean an from the temporary table and put it in the correct entry of the statistical library.

V. Measuring local variation of a design

The local variation of a design is a metric to compare multiple designs and to verify if tuning of the library is effective. To measure the local variation of a whole design one needs first to know the local variation of all paths which make up the design. In this section we will discuss how to calculate the local variation of a design from the cell up.

During the synthesis process the delay of a cell is extracted from the timing library based on cell characteristics, input slew and output load. The cell characteristics, such as logic function, determine which look-up table the synthesis process will use. The input slew and output load of the cell determine which values in the LUT will be used to interpolate the delay. The input slew and output load of a cell depend on the cell's preceding cells and its fanout, respectively. Because a look-up table does not contain all possible slew and load combinations, the exact sigma is calculated by using bi-linear interpolation.

A. Bilinear interpolation

Bilinear interpolation is a technique to calculate missing values between points in a two-dimensional grid [12]. Suppose that the sigma we are looking for is *X* and that we know that the corresponding load and slew are *L* and *S*, respectively. Assume that there are indices values for which $L_i < L \leq L_{i+1}$ for the load, $S_j < S \leq S_{j+1}$ for the slew and the look-up table has entries $Q_{11}(L_i, S_j)$; $Q_{12}(L_i, S_{j+1})$; $Q_{21}(L_{i+1}, S_j)$; $Q_{22}(L_{i+1}, S_{j+1})$. The ratio between the load *L* and the indices L_i and L_{i+1} is used to interpolate the intermediate value P_1 and P_2 as is shown in eqs. (2) and (3).The last step is to do a linear interpolation on P_1 and P_2 with the ratio between S_j and S_{j+1} to end up with the value for *X* as is shown in eq. (4). A graphical representation of the procedure is shown in Fig. 3. Depending on what information is present in the look-up table, *X* denotes the cell delay, transition power, mean or sigma.

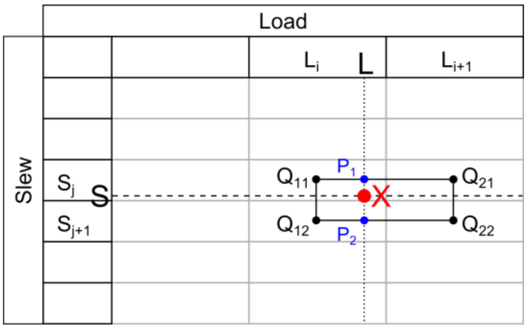


Fig. 3. Bilinear interpolation as seen from a LUT. Where L_i and L_{i+1} are load indices. S_j and S_{j+1} are slew indices. L and S are the load and slew values extracted from the synthesis. Q denotes a value in the LUT. X is the interpolated value.

$$P_1 = \frac{L_{i+1} - L}{L_{i+1} - L_i} Q_{11} + \frac{L - L_i}{L_{i+1} - L_i} Q_{21} \qquad (2)$$

$$P_2 = \frac{L_{i+1} - L}{L_{i+1} - L_i} Q_{12} + \frac{L - L_i}{L_{i+1} - L_i} Q_{22} \qquad (3)$$

$$X = \frac{S_{j+1} - S}{S_{j+1} - S_j} P_1 + \frac{S - S_j}{S_{j+1} - S_j} P_2 \qquad (4)$$

B. Convolution of probability distributions

A data-path is constructed out of a number of cells. Each cell has with a propagation delay mean and sigma. The path distribution timing parameters can be calculated by convoluting the timing distributions of the individual cells. We follow the calculation procedure described in [11, 13]. The average path delay is calculated by taking the summation of cell means for the cells comprising the path (eq. (5)).

$$\mu_{path} = \sum_{i=1}^{n} \mu_{cell_i} \qquad (5)$$

The standard deviation of a path is not as straightforward and requires the covariance of cells to be taken into account. Equation (6) shows the covariance matrix *C* where *n* is the number of cells in a path, c_{ij} denotes the covariance between two cells in the data-path and c_{ii} is the covariance of a cell with itself. The covariance between cells is determined by eq. (7) where ρ_{ij} is the correlation between two gates and $\sigma_{cell\,i}$ and $\sigma_{cell\,j}$ are the standard deviations of the cells under consideration.

$$C = \begin{bmatrix} c_{ii} & c_{i(j+1)} & \cdots & c_{in} \\ c_{(i+1)j} & c_{(i+1)(i+1)} & \cdots & c_{(i+1)n} \\ \cdots & \cdots & \cdots & \cdots \\ c_{nj} & c_{n(j+1)} & \cdots & c_{nn} \end{bmatrix} \qquad (6)$$

$$\begin{aligned} c_{ij} &= \sigma_{cell_i} * \sigma_{cell_j} * \rho_{ij} \\ c_{ii} &= (\sigma_{cell_i})^2 \end{aligned} \qquad (7)$$

The variance of a path can be formulated as eq. (8) and if we assume the correlation between cells to be identical $\rho_{ij} = \rho$, the equation can be rewritten to eq. (9). The assumption that $\rho_{ij} = \rho$, is made with the knowledge that cells in a path are not identical but their correlation is similar i.e. there are no outliers.

$$\sigma^2_{path} = \sum_{i=1}^{n} \sum_{j=1}^{n} \sigma_{cell_i} * \sigma_{cell_j} * \rho_{ij} \qquad (8)$$

$$\begin{aligned} \sigma^2_{path} &= \sum_{i=1}^{n} \sum_{j=1, j \neq i}^{n} \sigma_{cell_i} * \sigma_{cell_j} * \rho \\ &+ \sum_{i=1}^{n} (\sigma_{cell_i})^2 \end{aligned} \qquad (9)$$

Although local process variations are uncorrelated, e.g. threshold voltage mismatch of any two transistors, the propagation delay along cells exhibits a minor dependence on the cell's input and fanout. Since this dependence is very small, we assume that the correlation coefficient is $\rho = 0$. Equation (9) can be simplified to come to the formula to determine the standard deviation of a path as follows:

$$\sigma_{path} = \sqrt{\sum_{i=1}^{n} (\sigma_{cell_i})^2} \,. \qquad (10)$$

An identical approach is used to come to the distribution parameters of a total design as shown in eq. (11), where m is the number of paths.

$$\begin{aligned} \mu_{design} &= \sum_{j=1}^{m} \mu_{path_j} \\ \sigma_{design} &= \sqrt{\sum_{j=1}^{m} (\sigma_{path_j})^2} \end{aligned} \qquad (11)$$

VI. Library tuning

The library tuning approach taken in this paper is a two stage process. In the first stage, a threshold is extracted based on a tuning method. The second stage consists of applying the threshold to create a suitable subset of robust cells. The usual tuning of a library creates a subset of cells by excluding complete cells [4, 5, 6]. In this approach, instead of removing a cell completely, a restriction on the look-up table is imposed. The synthesis tool provides a way to restrict a look-up table on an individual cell's output pin. This means that for each pin of a standard cell a minimum and maximum slew and load value can be defined which effectively binds the synthesis tool to use only a section of the cell's look-up table. Hence, providing a fine grained tuning possibility.

A. Tuning methods

The tuning method proposed in this paper consists of two parts, one part denotes if the population of cells is considered on an individual bases or rather grouped per drive strength. The other part determines which threshold extraction is used (load slope bound, slew slope bound or a sigma ceiling).

In the first part we investigate constraining the load and slew values of the LUT by looking at an individual cell or a subset of drive strengths. Large drive strength cells can be created by either putting a number of transistors in parallel, basically creating a big transistor which can drive a bigger load, or, by using a different cell topology altogether. Note that cells which make use of larger transistors have a lower local mismatch variation [14]. Figure 4 shows a representation of the look-up tables for an inverter cell with different drive strengths. The horizontal axes show the load and slew values of the LUT and the vertical axis shows the delay's standard deviation (sigma). Note that the coloring is related to the height of the sigma where a dark color illustrates a low sigma. The figure shows that a low drive strength inverter (e.g. INV_1) has a smaller load range when compared to a larger drive strength inverter (e.g. INV_32). This is because a low drive strength inverter is not designed to drive a big load. Figure 4 also shows that the slew range for the different inverter cells is identical and furthermore, an inverter with higher drive strength i.e. large transistors (e.g. INV_32) has a lower overall sigma (the surface stays low). Note as well that the cell's gradient is lower than for a small drive strength inverter, i.e. small transistors (e.g. INV_1). These observations lead to an indication that the drive strength is a possible clustering parameter but needs further probing by looking at a cluster of cells with an equal drive strength as is shown in Fig. 5. Figure 5 displays the delay sigma for all cells with drive strength six. This specific drive strength was chosen to prevent cluttering of the image. For the same reason, only a single timing arc of each cell is shown. Observe also that not all cells seem to have an identical load range or slope (e.g. NR4_6, a four input NOR gate). Instead of looking at a subset of cells based on their drive strength, we can also look at the entire population of cells. In this case no assumption is made on cell parameters and only the actual standard deviation is taken into account to restrict the look-up table of a cell. This provides a contrasting comparison with the drive strength clustering approach.

For the second part of the tuning method, three different constraining parameters are considered, namely: load slope bound, slew slope bound and a sigma ceiling. In both slope bounding methods we look at the slope gradient to identify areas with a steep sigma increase. These areas are not preferred since a small increase in either load or slew would result in a large sigma increase (i.e. large gradient). Instead, a relatively flat surface is preferred. Next, the sigma ceiling restricts the use of sigma values above a certain threshold. This prevents the situation where a cell has a weak slope but high overall sigma and is therefore not restricted by the slope methods.

In total there are five tuning methods which are investigated, namely: Cell strength based slew slope bound, Cell strength based load slope bound, Cell based slew slope bound, Cell based load slope bound and Cell based sigma ceiling.

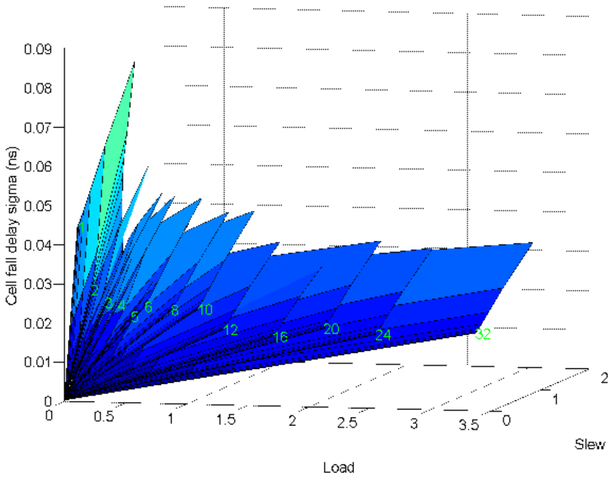


Fig. 4. Surface plot of the LUTs from several drive strengths of an inverter cell.

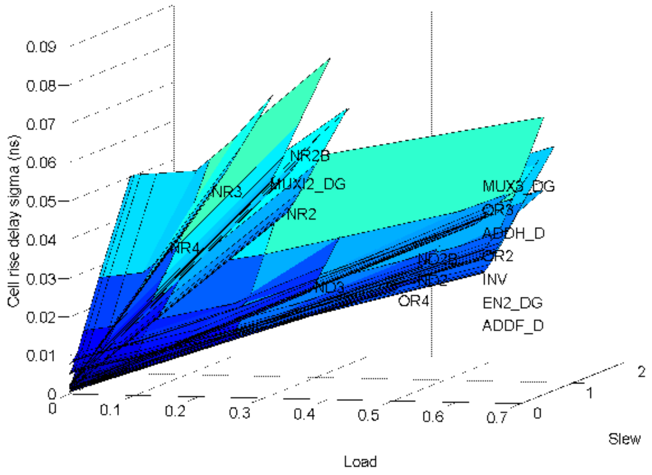


Fig. 5. Surface plot of the LUTs from several cells with drive strength 6.

B. Threshold extraction

Depending on the tuning method, a threshold is extracted from the statistical library. Note that this is only needed for the slew and load slope bound methods since the sigma ceiling is used as threshold on its own.

For the slope methods, a threshold is extracted by creating a maximum equivalent LUT for all the cells (and their related LUTs) in a cell cluster. The equivalent LUT contains the maximum sigma value for each individual table entry for the whole cluster. The equivalent LUT is then converted to a slope table for both the load and slew direction separately, by applying eqs. (12) and (13) where $1 < i \leq n$ and $1 < j \leq m$ are the indexes and $Q(i,j)$ is the table entry at index (i,j). Note that because the indexes start at greater than one, the first row or column of the slew and load slope tables is filled with zeros.

$$\alpha_{slew}(i,j) = \frac{Q(i,j) - Q(i-1,j)}{\Delta i} \qquad (12)$$

$$\alpha_{load}(i,j) = \frac{Q(i,j) - Q(i,j-1)}{\Delta j} \qquad (13)$$

Both slew and load slope tables are converted to binary slew and load tables, thresholded by an upper slope limit. This means that all table entries which are smaller than the slope threshold become a logic one and the remaining a logic zero. The contents of both binary load and slew tables are combined by taking the logic ‘and’ resulting in a single binary LUT with logic ones for all areas which are flat (i.e. have a slope lower than the slope threshold). In the flat region of the LUT we search for the largest rectangle starting as close as possible to the origin of the LUT, see Algorithm 1. The largest rectangle denotes the largest area for which the LUT is still flat. From this area a threshold is extracted by taking the sigma value corresponding to the rectangle coordinate furthest from the origin (see marked entry in Fig. 6). This sigma threshold is used in the next step to apply the actual slew and load restrictions on a cell’s LUT.

Algorithm 1. Largest Rectangle Extraction

```
[x_1, y_1, x_2, y_2] = getLargestRectangle(bin_LUT)
% Reset coordinates and best area
x_1 = 0;
y_1 = 0;
x_2 = 0;
y_2 = 0;
best_area = 0;
% Loop through all the possible table entries
for ll_x = 1:N
  for ll_y = 1:M
    for ur_x = ll_x:N
      for ur_y = ll_y:M
        % If the indices form a rectangle
        if (ur_x > (ll_x - 1)) && (ur_y > (ll_y - 1))
          % Store the area of the rectangle
          area = (ur_x -(ll_x - 1))*
                 (ur_y -(ll_y - 1));
        else
          % No rectangle, so no area
          area = 0;
        end
        % Check if the rectangle is bigger and has only logic ones
        if area > best_area && ...
           all_true( bin_LUT(ll_y:ur_y ll_x:ur_x) )
          % Store the new area and coordinates
          best_area = area;
          x_1 = ll_x;
          y_1 = ll_y;
          x_2 = ur_x;
          y_2 = ur_y;
        end
      end
    end
  end
end
```

	Load			
Slew	0	1	1	1
	1	1	1	1
	0	1	1	1
	1	0	0	0

Fig. 6. Result of the largest rectangle algorithm shown in a binary LUT. The marked entry denotes the entry for which the sigma is extracted.

C. Look-up table restriction

The synthesis tool only allows the confinement of a look-up table based on output pins. Thus, the worst case situation has to be taken into account. Since each output pin has a number of different LUTs, i.e. the cell rise and fall related timing arcs, the worst-case values for these tables are used. For every output pin of a cell, a maximum equivalent look-up table is created by taking the maximum value for each entry of the related tables. The equivalent table is then converted to a binary look-up table based on the extracted threshold. Values in the equivalent table which are smaller than the threshold will become a logic one whereas values greater than the threshold will translate to a logic zero. Applying the largest area rectangle algorithm (Algorithm 1) on the binary look-up table results in the coordinates of the rectangle which encapsulates the largest area for which the sigma values of the cell pin are still acceptable. The rectangle coordinates are directly related to the minimum and maximum load and slew values which are later used in the synthesis.

VII. Test design and experimental results

As a baseline set, a statistical library of 304 cells was generated to test the library tuning approach. A summary of the 304 cells can be found in Appendix VIII.A. The library is based on a CMOS 40nm technology. Further, all cells are characterized in the typical corner of the process, using a 1.1V supply voltage and temperature of 25°C (TT1P1V25C). The statistical library is created by combining 50 library files. A graphical representation of the statistical library is provided in Fig. 7 which shows the sigma delay surfaces of the look-up tables. The horizontal axes represents the load and slew indices and the vertical axis denotes the sigma delay in nano seconds (ns).

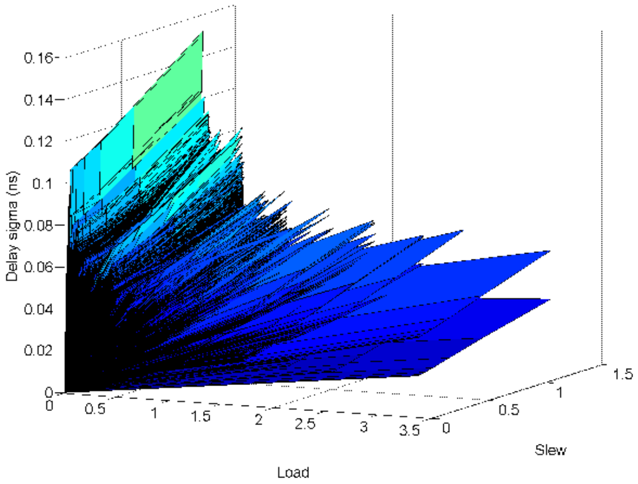


Fig. 7. All cell delay sigma look-up tables in the TT1P1V25C library, combined in a surface plot.

For evaluation purposes, a microcontroller design was used with a 32-bit CPU, AHB bus, 32KB SRAM, and a low gate count (20k gates). Three different timing constraint areas are considered. The first timing constraint is the minimum clock period achievable by synthesizing the microprocessor with the baseline set. The second one is a relaxed timing, and the last one is a low performance constraint, see Table 1. The minimum clock period is found by reducing the clock period until the synthesis fails to provide a design with positive slack. Figure 8 shows a plot with the clock period versus the total cell area of the design. The relaxed timing condition can be found on the point where the curve is linear which is at 10ns.

TABLE 1. CLOCK PERIODS FOR DIFFERENT CONSTRAINTS AND LIBRARIES

	Clock period
High performance	2.41 ns
Medium performance	4 ns
Low performance	10 ns
Close to maximum check	2.5 ns

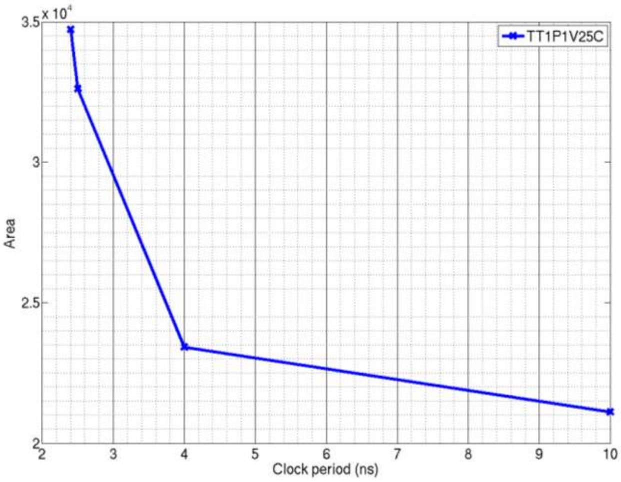


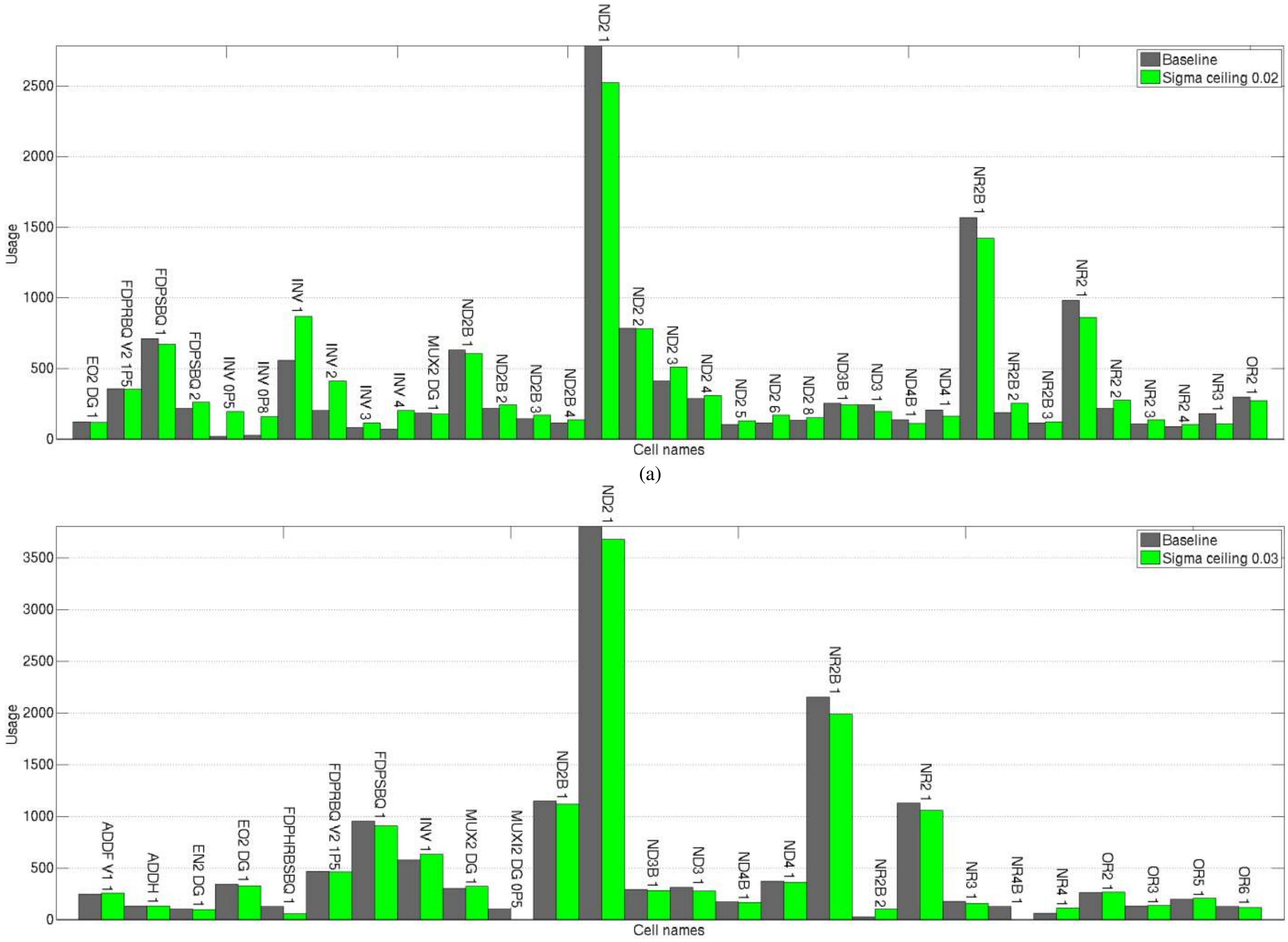
Fig. 8. Clock period versus area of the microprocessor after running a baseline synthesis with the TT1P1V25C library.

Table 2 shows the parameters used to tune the library. A slope constraint of 1 means that all the gradient values bigger than 1 will be excluded from synthesis. This removes those parts of the statistical library for which a load or slew increase results in a high sigma increase. The remaining slope constraint parameters are chosen based on the surface plot of the statistical library as shown in Fig. 7. A slope constraint of 0.05 will remove a majority of high gradient cells by restricting a cell's LUT in the load direction. The next slope constraints will restrict the use of larger parts of the look-up tables until the 0.01 constraint which, as Fig. 7 shows, leaves only shallow gradient values available for synthesis. The remaining sigma ceiling constraints will gradually remove larger parts of the LUTs without making the synthesis unfeasible.

TABLE 2. CONSTRAINT PARAMETERS USED DURING THRESHOLD EXTRACTION

	Constraint parameters	
	TT1P1V25C	***Default***
Load slope bounds	1, 0.05, 0.03, 0.01	1
Slew slope bounds	1, 0.05, 0.03, 0.01	0.06
Sigma ceiling	0.04, 0.03, 0.02, 0.01	100

During the cell selection stage, only one parameter is varied while the other two stay at the default value e.g. the slew and sigma are kept at 0.06 and 100, respectively, while the load bound is swept along 1, 0.05, etc.



(a)

(b)

Fig. 9. Cell use for a baseline synthesis and tuning method (marked in TABLE 3) at a clock period of (a) 2.41ns and (b) 10ns. Only cells which are used more than 100 times are listed.

Figure 9 shows a histogram with the cells that are used in the baseline synthesis for the high performance and relaxed timings. Only cells that were used more than 100 times are listed. Interesting to see is that in the synthesis of the processor, basic cells such as the NAND, NOR, INV and flip-flops are most often used. Also, the more time constraint a synthesis is, the more variety of simple cells it tends to use (Fig. 9a is more time constraint and uses a larger variety of cells compared to the relaxed timing in Fig. 9b), whereas a relaxed timing design uses the more dedicated cells such as adders.

Figure 10 shows the highest sigma reduction for an area increase less than 10% compared to the baseline, for different timing constraints. Each bar in the figure represents a tuning method. The constraining parameter which was used to achieve this reduction is shown in Table 3. Worth mentioning is that during synthesis a guard band of 300ps was used so the effective clock period becomes 2.11ns. We can see from the annotations in Fig. 10 that a relaxed timing results in a higher design sigma. The synthesis process is not restraint in the timing and can thus try to optimize the design in terms of area. A reduction in area can be achieved by using small cells and as few as possible cells. Both optimizations counteract the sigma reduction as will be further explained in this section. From Fig. 10 it is clear that the sigma ceiling method has a good sigma reduction of 37% with an area overhead of 7% for a high performance design and a reduction of 32% at the cost of 4% area overhead for a low performance design. Figure 10 furthermore shows that a tradeoff can be made in the sigma reduction versus area increase by selecting a different tuning method, i.e. the two strength based methods provide decent sigma reduction with less area overhead. Especially interesting are both cell strength methods with respect to the high performance design. Here they provide a sigma reduction of 31% while having a similar area compared to the baseline design. This sigma versus area tradeoff is not only visible between the different tuning methods but also for a single method by using a different bound, as is illustrated by Fig. 11.

For illustration purposes, the sigma ceiling method is discussed further since it clearly shows the effects of the library tuning approach.

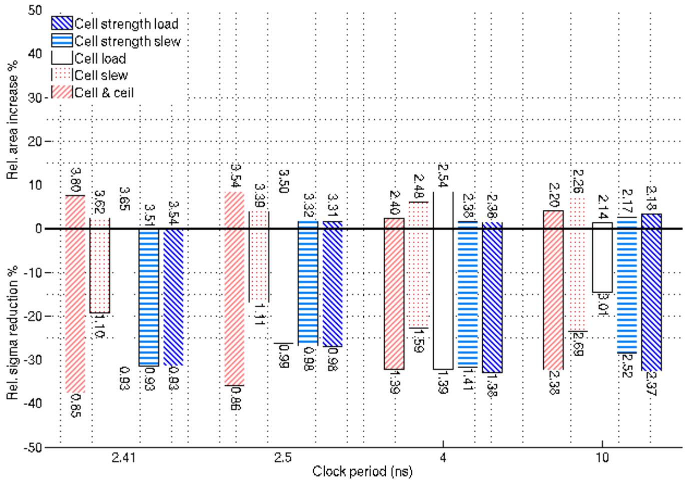


Fig. 10. Relative sigma decrease and area increase between baseline and tuning methods with highest sigma reduction at an area increase less than 10%, for different clock periods. The top part is the relative area increase with the real area value annotated (in 10^4 µm^2). The bottom part is the relative sigma decrease with the real sigma value annotated (in ns).

TABLE 3. CONSTRAINT PARAMETERS USED TO GET THE SIGMA DECREASE

	Clock period (ns)			
Tuning Method	***2.41***	***2.5***	***4***	***10***
Cell strength load	0.01	0.05	0.03	0.03
Cell strength slew	0.01	0.01	0.05	0.03
Cell load	0.01	0.01	0.03	1.00
Cell slew	0.05	0.01	0.03	0.01
Sigma ceiling	0.02	0.02	0.03	0.03

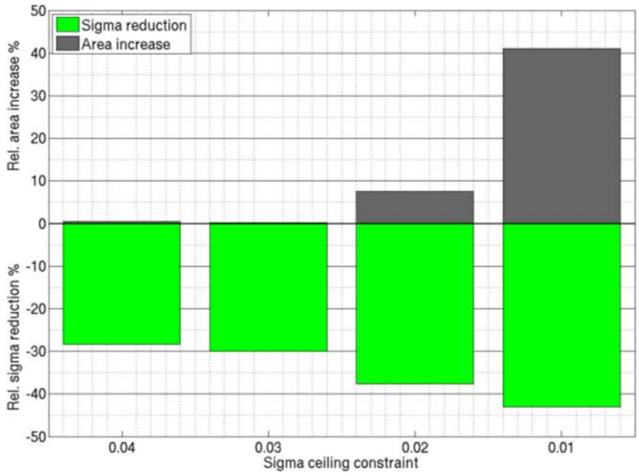


Fig. 11. Relative sigma decrease and area increase between baseline and the sigma ceiling procedure, for a clock speed of 2.41ns. The figure shows a clear tradeoff between sigma reduction and area increase.

A. Impact of library tuning on data-path depth

The area increase introduced by the library tuning can be explained when looking at the path depth. Figure 12 shows the path depths of the worst case paths connected to a unique endpoint for a clock period of 2.41ns. An overall increase in the path depth indicates that more cells are used for the restricted design. When a cell with a specific logic function is removed, the synthesis process can either use a combination of available cells to recreate the logic function, or use a higher drive strength of an identical function.

Inspecting the individually used cells for the sigma ceiling constraint in Fig. 9, illustrates an increase in the overall use of inverter cells. The most likely cause for the increase of inverter use is buffering. An inverter cell can be used by the synthesis tool as a buffer to restore signal integrity. Secondly, Fig. 9 confirms the increase in high drive strength cells. Looking at cell NR2B_1 (a 2-input NOR cell with drive strength 1), the cell is less used in the restricted design whereas the higher drive strengths of the same cell (NR2B_2, NR2B_3, etc) are more often included in the design.

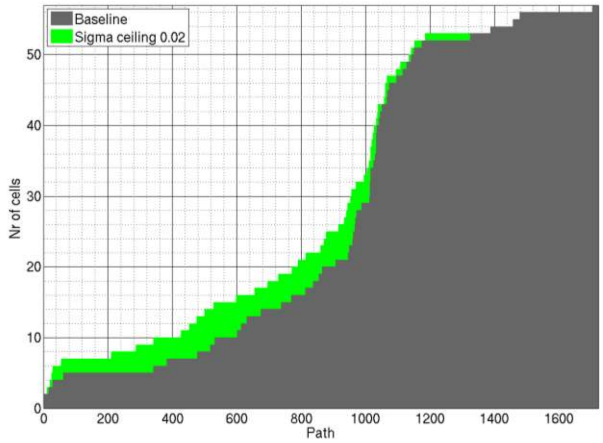


Fig. 12. Path depths for the worst case paths connected to a unique endpoint for a time constraint design (2.41ns). The depths of both the baseline and the sigma ceiling method are shown. Note that although not optimal in terms of area, the constrained sigma ceiling clearly illustrates the effect of library tuning.

B. Impact of data-path depth on local variation

Figure 13 shows the path timing spread plotted against the path depth for the baseline and the restricted sigma ceiling method at a clock period of 2.41ns. The figure illustrates that there is no direct relation between the path depth and the local variation of a path but instead, the local variation of a data-path is dictated by the used cells and their properties.

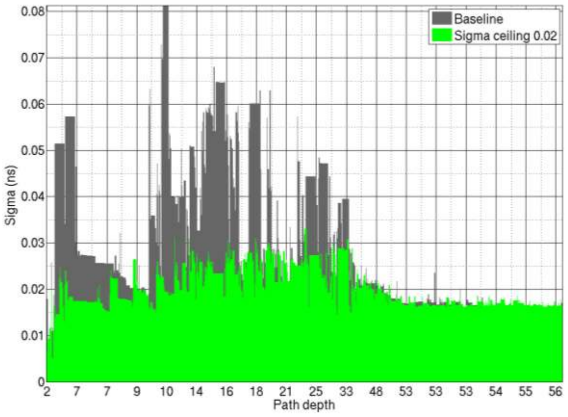


Fig. 13. Sigma versus path depth for the baseline and the sigma ceiling method are shown.

In Fig. 14a the mean and three sigma values of a path are shown for the baseline synthesis of the 2.41ns design. The paths are sorted according to their depth. The vertical axis shows the path delay. The graph shows that for the shortest paths (less than three cells) the relative sigma is higher than for the longest paths (more than 50 cells). However, some short (four - five cells) and medium sized paths (between seven and 40 cells) have a misbehaving relative sigma. This shows that in a digital design it is not sufficient to assume timing spread of a path to be only dependent on the path length. Furthermore, the setup time for final flip flops (a final flip-flop being the last element in the data-path which retains the signal for synchronization) is not added in Fig. 14a which is noticeable by mean values which fluctuate between 2.11ns and 2ns. Because of the design being at the high performance timing there are a number of medium depth paths (around 18 cells) which have a fairly high mean value (Fig. 14a).

In the ideal case, these paths will not cause a problem but with the added local variation (3σ) these paths will cause a timing failure since they get above the 2.11ns clock period. Looking at Fig. 14b, which shows the design after applying the sigma ceiling method, we see that the overall behavior is more homogenous due to a reduced mean and sigma. There are however still some paths which can cause the design to fail but this is largely due to an increase in the mean. The 3σ value for these paths is relatively low. Also the figure shows that the worst case values are lowered from 2.23ns to 2.19ns.

C. Validation on process corners

In these experiments only the typical corner is considered and the validity of the approach across other corners is verified by looking at the behavior of a set of extracted data-paths, simulating them for different corner conditions. The expected behavior of a circuit when moving to a different corner (fast, typical and slow) obeys an increase in path delay when moving towards a slow corner and vice versa moving towards a fast corner. If the local variability of the path scales in the same manner as the mean, the sigma of a design for other corners can be predicted and the library tuning can also be applied on those corners. Figure 15a, 15b and 15c show the results of running Monte Carlo on a short, medium and long path, respectively.

All paths are extracted from the baseline design at a clock period of 2.41ns where the short size path has a depth of three cells, the medium size path has 18 cells and the long size path has 57 cells. The relative mean and sigma shows that in all cases, moving towards a different corner scales the mean and sigma by the same factor when compared to the typical case. This indicates that the sigma of a design will scale by an identical factor and hence the library tuning method will also provide the scaled results.

The total variation of a path is made up out of the global and local variation. In this report the focus lies on local variation but the attribution of the global variation is also looked into. Figure 16a, 16b, and 16c gives the histogram plots for short, medium and long size paths, respectively. The paths are extracted from a baseline design with a clock period of 2.41ns. One histogram is a result of running 200 Monte Carlo (MC) simulations with global and local variation, the other one only includes local variation. The mean and sigma of the local MC are shown relative to the global and local MC. The figures clearly show that the impact of local variation is more pronounced in short paths and decays with the increase of path depth. The local variation contributes 65% of the total variation of a short path, 37% for a medium path and finally the contribution of local variation for a long path of 55 cells is 6%. In conclusion, about one third of the paths, connected to a unique endpoint, contribute to the total variability of the design by a predominant local variation.

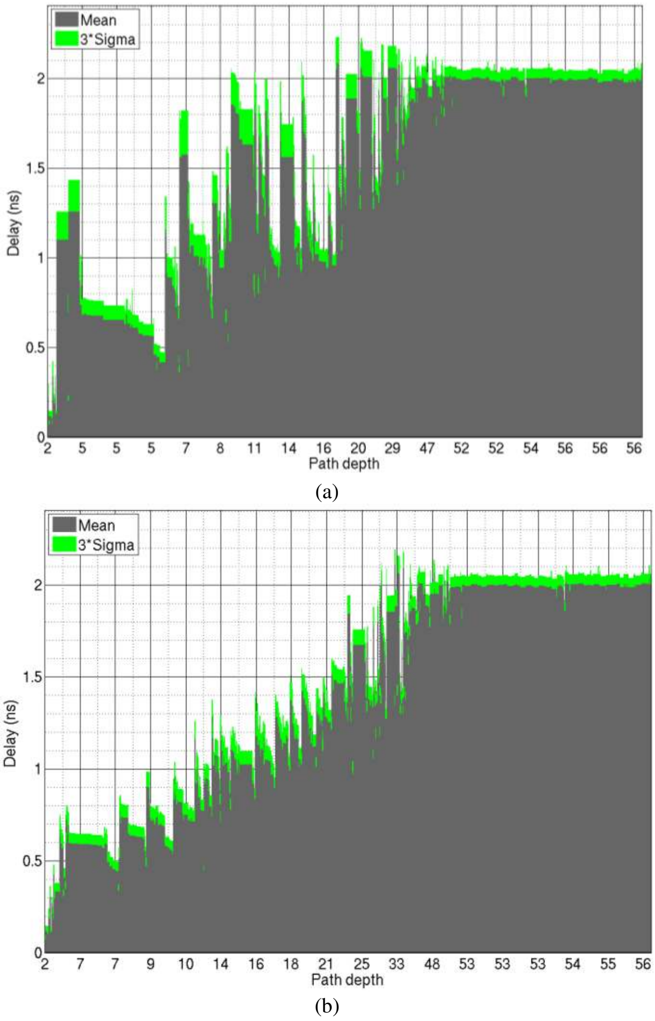


Fig. 14. Mean + 3 sigma path delay per path depth for (a) baseline synthesis and (b) sigma ceiling restriction with a clock period of 2.41ns.

Equation (10) shows that under the assumption that a path contains identical cells in terms of mean and standard deviation, the sigma of a deeper path is higher than the sigma of a short path. Looking at Fig. 16b and 16c (and Fig. 15b and 15c), however, shows this is not necessarily the case. Figure 16b which has a lower path depth, has a higher sigma (and mean for that matter) compared to the longer path of Fig. 16c. The reason for this lays in the behavior of the synthesis. Both the medium and long paths have the same timing constraint, however the long path (having more cells) reaches the clock constraint faster and thus the synthesis tool has to use larger drive strength cells in order to meet the timing constraint. On the other hand, a medium size path is not likely to overshoot the timing constraint and the synthesis tool will optimize the path in terms of area. This means that the path contains low drive strength cells. As we saw before, high drive strength cells tend to have a lower sigma value and hence a long path consisting of higher drive strength cells can have a lower sigma than a medium sized path.

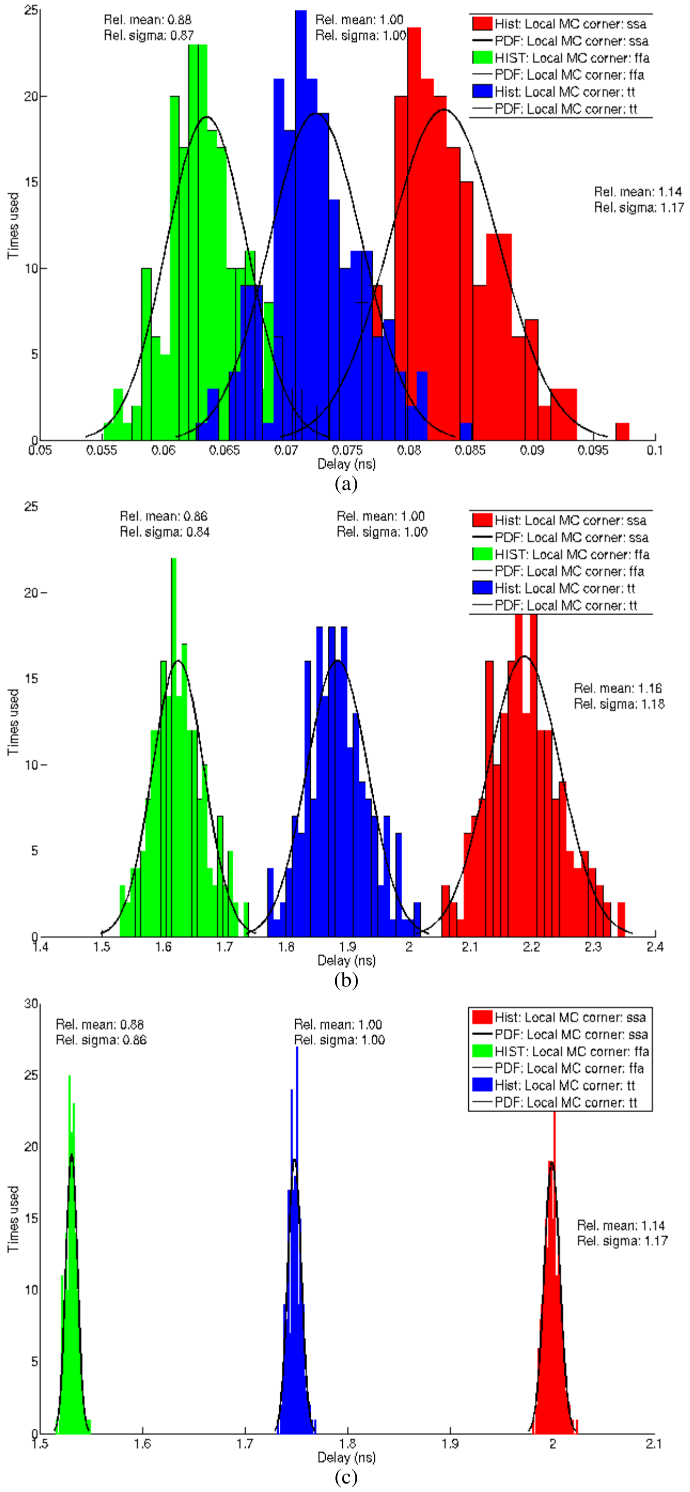
Comparing the sigma and mean of the medium and long path from the MC simulation and the one of the synthesized design and hence the statistical library, shows that the mean values are within the expected error of the library. The sigma values in the statistical library deviate to an upper-bound of two times bigger compared to the values from the simulations. This is due to the low number of samples and some inaccuracy in retrieving the sigma from a cell. Using more MC samples to create the statistical library would reduce this error but this is future work.

VIII. Conclusion

In this paper we have shown that by means of library tuning, the sensitivity of a design towards local variation can be reduced.

By using a library tuning method which does not remove from the library a complete cell but instead confines the use of the cell's look-up table, the tuning becomes finer grained. By utilizing one of the tuning methods in combination with a restriction parameter, the library tuning result can be directed towards a high sigma reduction of 37% at the cost of 7% area increase, depending on the clock speed or a smaller sigma reduction of 30% with almost no area increase (2%). Overall, the homogeneity of the design towards local variation is improved as is the case for the robustness due to a smaller sensitivity towards local variation of the design. The path depth is not directly correlated to the absolute sigma value of the path, but the contribution of local variation to the total process variation is larger for shorter paths and decreases the longer a path gets. Since around one thirds of paths to unique endpoints in the design are short paths, local variation does contribute to the total variation of the design. Because the local variability scales with the same factor as the mean across multiple PVT corners, the library tuning method can also be applied in combination with these PVT corners and the expected behavior scales with the aforementioned factor.

Further research is needed to improve the statistical library and to see if the local variation reduction which is shown in this work, also reduces the local variation of the design after place and route and clock tree synthesis (the next steps in IC design). The effectiveness of the method on the clock tree in particular needs further investigation.

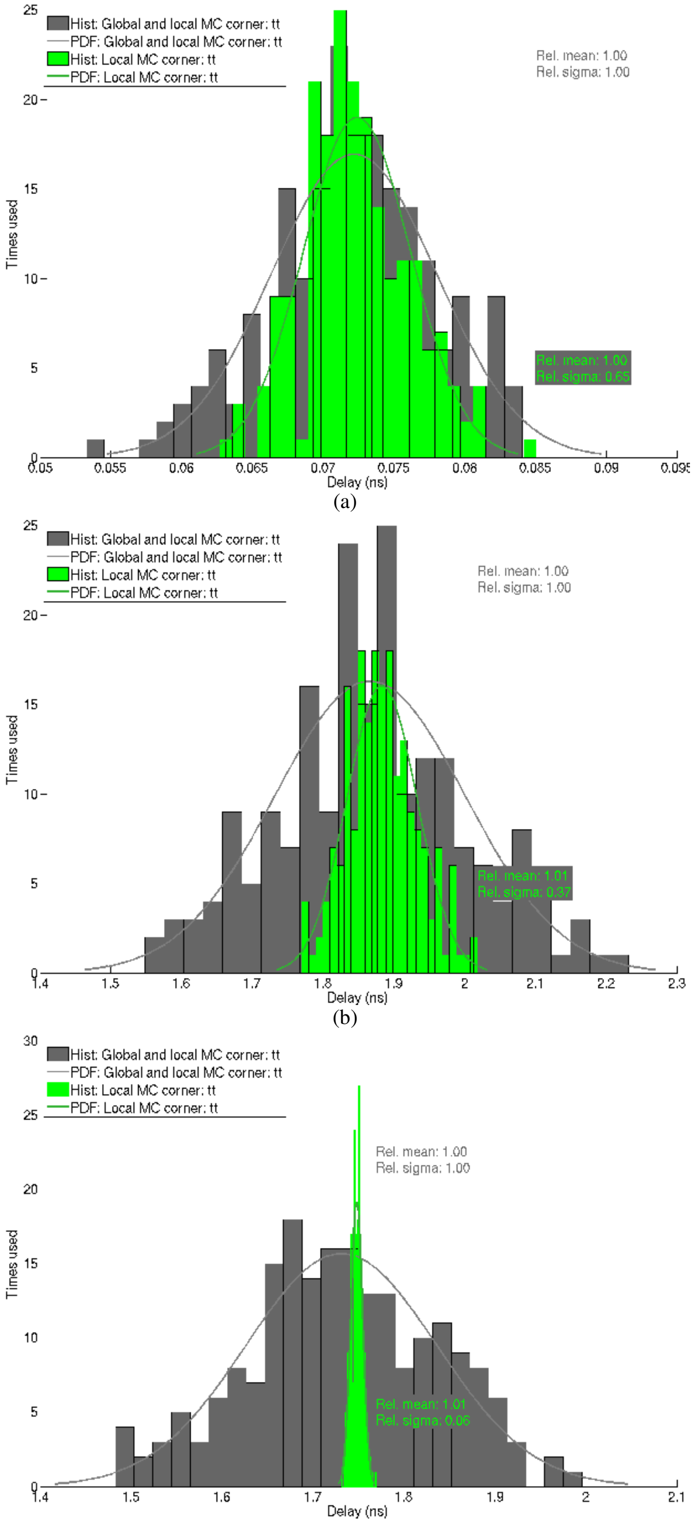


(a)

(b)

(c)

Fig. 15. Monte Carlo simulation (N=200) for three extracted paths from the design at a clock period of 2.41ns. The short path (a) has 3 cells, the medium path (b) has 18 cells and the long path (c) has 57 cells. The histograms are there for a fast, typical and slow corner, respectively. The sigma and mean relative to the typical corner is shown in text. Both the mean and sigma scale accordingly when moving to different corners.



(a)

(b)

(c)

Fig. 16. Monte Carlo simulation (N=200) for three extracted paths from the design at a clock period of 2.41ns. The short path (a) has 3 cells, the medium path (b) has 18 cells and the long path (c) has 57 cells. The histograms are for global and local variation and for local variation only to show the impact of both variation types. The sigma and mean relative to the typical corner is shown in text.

APPENDIX

A. Statistical library cells

Only a subset from the total standard cell library is used to create the statistical libraries. The cell name describes the cell and some of the characteristics. The naming convention of the cells is as follows: "Logic function[Nr input pins]_[Special ability_]Drive strength" where the parameters between [square brackets] are optional and a P between numbers denotes a decimal separator. In summary the statistical libraries have:

- 19 Inverter cells
- 36 Or
- 46 Nand
- 43 Nor
- 29 Xnor
- 34 Adders
- 27 Multiplexers
- 51 Flip-flops
- 12 Latches
- 7 Other cells

ACKNOWLEDGMENT

First, the author would like to thank his parents for their love and support during this project.

Secondly, the author would like to express his gratitude towards José Pineda de Gyvez and Maarten Vertregt for their guidance and knowledgeable input during the project. Furthermore, he would like to thank Juan Diego Echeverri for his comments and technical support.

Lastly, the author would like to thank the colleagues at NXP semiconductors for making this project a fruitful and enjoyable experience.

REFERENCES

[1] Semiconductor Industries Association, "International technology roadmap for semiconductors," 2012, http://www.itrs.net

[2] P. S. Zuchowski, P. A. Habitz, J. D. Hayes, and O. J. H., "Process and environmental variation impacts on ASIC timing," *IEEE/ACM international conference on computer aided design (ICCAD)*, Vermont, 2004, pp. 336-342.

[3] N. H. E. Weste and D. Harris, "CMOS VLSI design: A circuits and systems perspective," Addison-Wesley, 2010.

[4] D. C. Ness, C. J. Hescott, and D. J. Lilja, "Exploring subset of standard cell libraries to exploit natural fault masking capabilities for reliable logic," *Great lake symposium on VLSI*, Stresa-Lago Maggiore, Mar. 2007.

[5] A. Ricci, I. De Munari, and P. Ciampolini, "Performance-effective compaction of standard-cell libraries for digital design," *Euromicro conference on digital system design*, Patras, 2009, pp. 315-322.

[6] M. Rahman, R. Afonso, H. Tennakoon, and C. Sechen, "Power reduction via separate synthesis and physical libraries," *Design automation conference*, San Diego, 2011, pp. 627-632.

[7] Synopsys, "Open source Liberty library modeling format," 2013, http://www.opensourceliberty.org

[8] D. C. Montgomery and G. C. Runger, “Applied statistics and probabilty for engineers”, Fouth ed. John Wiley & Sons, Inc., 2007.

[9] R. Takahashi, et al., "Large within-die gate delay variations in sub-threshold logic circuits at low temperature," *IEEE transactions on circuits and systems*, vol. 59, no. 12, pp. 918-921, Dec. 2012.

[10] J. Hong, et al., "An LLC-OCV methodology for statistic timing analysis," *International symposium of VLSI design, automation and test 2007*, 2007, pp. 1-4.

[11] M. Eisele, J. Berthold, D. Schmitt-Landsiedel, and R. Mahnkopf, "The impact of intra-die device parameter variations on path delays and on the design for yield of low voltage digital circuits," *IEEE transactions on very large scale integration (VLSI) Systems*, vol. 5, no. 4, pp. 360-368, Dec. 1997.

[12] T. M. Lehmann, C. Gönner, and K. Spitzer, "Survey: Interpolation methods in medical image processing," *IEEE transactions on medical imaging*, vol. 18, no. 11, pp. 1049-1075, Nov. 1999.

[13] S. M. Ross, "Introduction to probability and statistics for engineers and scientists", Second ed. Elsevier academic press, 2004.

[14] M. J. M. Pelgrom, H. P. Tuinhout, and M. Vertregt, "Transistor matching in analog CMOS applications," *International electron devices meeting*, San Francisco, CA, USA, Dec 1998, pp. 915-918.