Standard Cell Library Tuning for Variability Tolerant Designs

Sebastien Fabrie[1,2], Juan Diego Echeverri[2], Maarten Vertregt[2] and José Pineda de Gyvez[2,3]
[1]Dept. of computer science and engineering, [3]Dept. of electrical engineering, Eindhoven University of Technology
[2]Digital architectures, circuits, and signal processing, NXP Semiconductors
Eindhoven, The Netherlands
Sebastien.Fabrie@nxp.com

Abstract—**In today's semiconductor industry we see a move towards smaller technology feature sizes. These smaller feature sizes pose a problem due to 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 them intrinsically more susceptible for variations [2]. Environmental effects introduced during the physical manufacturing of the chips are divided into global and local variations.

Global (or inter-die) variation is a well know phenomenon [3] and can be accounted for during the design of an integrated circuit. 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. Cell parameters such as orientation, channel length and oxide thickness make the cell more or less sensitive towards local variations.

In this paper we introduce a library tuning method aimed at reducing the local variation of the data-path of a digital design. 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 [6], library tuning is used to reduce power by only using a small subset of cells in the physical design. [7, 8] both use library tuning to generate a library with optimal gate sizes. Furthermore, library tuning is used in [9] to optimize a design for low power, sub-threshold operation. In most of the aforementioned approaches, a subset is created based on the exclusion of cells. 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 our proposed method we create a subset by constraining the cell's use, individually, to a range of slew and loading capacitance conditions where the cell exhibits the lowest timing-spread. These slew-load range constraints are passed over to the synthesis tool such that the choice of cells is based on a fine grained selection based on the allowed slew-load range, instead of ignoring complete cells and hence over-restricting the synthesis. The result is a design that has diminished variability in all paths.

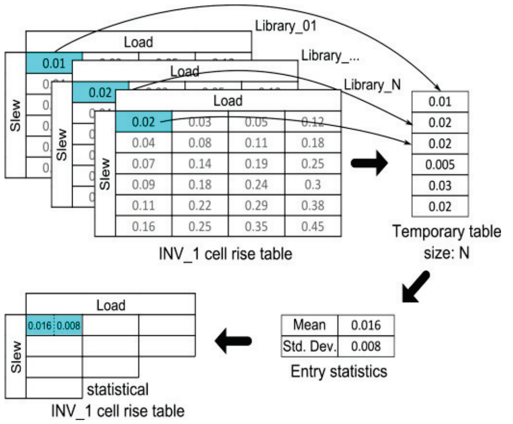
The remainder of this paper presents background theory, details of the proposed approach, and finally, experimental synthesis results using a small real-world microcontroller design (20K gates).

II. Library 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 1 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. 1).

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 originated from in the initial table. When each entry and table is processed, the described approach results in a library file [10] with identical table indices as a nominal library but which contains local variation statistics instead.

978-3-9815370-2-4/DATE14/©2014 EDAA



0.02	0.03	0.05	0.12
0.04	0.08	0.11	0.18
0.07	0.14	0.19	0.25
0.09	0.18	0.24	0.3
0.11	0.22	0.29	0.38
0.16	0.25	0.35	0.45

0.01
0.02
0.02
0.005
0.03
0.02

Mean	0.016
Std. Dev.	0.008

Fig. 1. 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.

III. Measuring local variation of a design

During the synthesis process, the cell's delay 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 bilinear interpolation. Bilinear interpolation is a technique to calculate missing values between points in a two-dimensional grid [11]. To extract the standard deviation and mean of a cell, bilinear interpolation is applied on the statistical library, with the same parameters as were used for the delay.

A data-path is constructed out of a number of cells. Each cell with a propagation delay mean and sigma. The path distribution timing parameters can be calculated by convolving the timing distributions of the individual cells. We follow the calculation procedure described in [12, 13]. The average path delay is calculated by taking the summation of cell means for the cells comprising the path (eq. (1)).

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

The standard deviation of a path is not as straightforward and requires the covariance of all cells in a path to be taken into account. By constructing a symmetrical covariance matrix [12], the variance of a path can be formulated as eq. (2) and if we assume the correlation between cells to be almost identical, e.g. $\rho_{ij} \approx \rho$, the equation can be rewritten as eq. (3). The assumption that $\rho_{ij} \approx \rho$, is made with the knowledge that cells in a path are not identical but that their correlation is similar i.e. there are no outliers.

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

$$\sigma^2_{path} = \sum_{i=1}^{n} \sum_{j=1, j \neq i}^{n} \sigma_{cell_i} * \sigma_{cell_j} * \rho + \sum_{i=1}^{n} (\sigma_{cell_i})^2 \qquad (3)$$

As local process variations are uncorrelated, the threshold voltage mismatch of any two transistors is uncorrelated also. Still, the propagation delay along cells exhibits a minor, correlating, dependence on the cell's input (slew) and fanout (load). Since this dependence is very small, we assume that the correlation coefficient is $\rho \approx 0$. Equation (3) can be simplified to come to the formula to determine the standard deviation of a path as follows:

$$\sigma_{path} = \sqrt{\sum_{i=1}^{n} (\sigma_{cell_i})^2} \; . \qquad (4)$$

An identical approach is used to come to the distribution parameters of a total design as shown in eq. (5), where *m* is the number of paths.

$$\begin{aligned} \mu_{design} &= \sum_{j=1}^{m} \mu_{path_j} \\ \sigma_{design} &= \sqrt{\sum_{j=1}^{m} (\sigma_{path_j})^2} \end{aligned} \qquad (5)$$

IV. 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 the timing robustness of the cells. The second stage consists of applying the threshold to create a suitable subset of robust cells. In our approach, instead of removing a cell completely [4, 5, 6, 7, 8, 9], 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 methods are deployed into two distinct clustering approaches of the cells. One such clustering is to group cells by driving strength, the other is simply to look at the entire population of cells. We investigate three constraining parameters, namely: load slope bound, slew slope bound and a timing spread (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. The sigma ceiling restricts the use of sigma values above a certain threshold which prevents the situation where a cell has a weak slope but high overall timing spread 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.

B. *Threshold extraction and Look-up Table restriction for synthesis purposes*

Depending on the tuning method, a unique threshold is extracted from the statistical library. This is primarily needed for the slew and load slope bound methods since the sigma ceiling is used as a threshold on its own. For the slope methods, a threshold is extracted by creating a maximum equivalent LUT for all cells (and their related LUTs). The equivalent LUT contains the maximum sigma value of each individual table entry for the whole cell cluster. The equivalent LUT is then converted to a "slope" table for both the load and slew directions separately by taking the derivative between any two consecutive entries in the table. Both slew and load slope tables are then converted to binary slew and load tables by thresholding through an upper slope limit. This means that all table entries which are smaller than the slope threshold become a logic one and the remaining ones 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 threshold slope). In the flat region of the LUT we search for the largest rectangle starting as close as possible to the origin of the LUT. The largest rectangle encapsulates the largest area for which the LUT is still flat. The rectangle coordinates are directly related to the minimum and maximum load and slew values which are used in the synthesis process. A sigma threshold is extracted from this area by taking the sigma value corresponding to the rectangle coordinate furthest from the origin. Figure 2 illustrates an example of sigma ceiling thresholding on a cell's LUT.

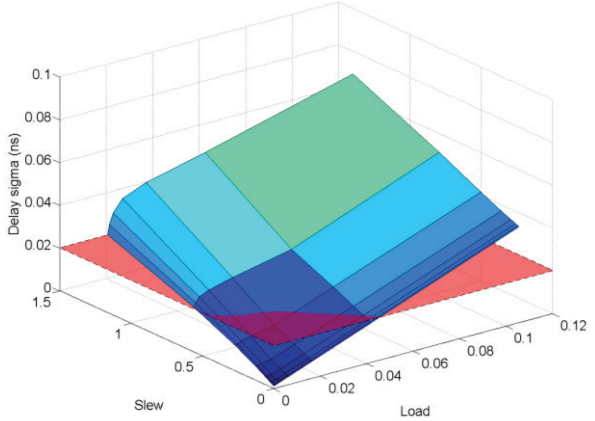


Fig. 2. Graphical representation of LUT. On vertical axis is the sigma delay, the x-y axis represent the slew and load. This figure illustrates the thresholding plane based on sigma ceiling that is cutting the LUT

The synthesis tool only allows the confinement of a LUT based on output pins. Each output pin has a number of different LUTs which correspond to the cell's rise and fall timing arcs. The extracted sigma value (as previously described) is then applied to threshold each of these LUTs.

V. Test Design and Experimental Results

As a baseline set, a statistical library of 304 cells was generated to test the library tuning approach. 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. 3 which shows the sigma delay surfaces of the look-up tables. The horizontal axes represent the load and slew indices and the vertical axis denotes the sigma delay in nano seconds. During the experiment only timing is taken into consideration. Power and clock tree aspects are left for future work.

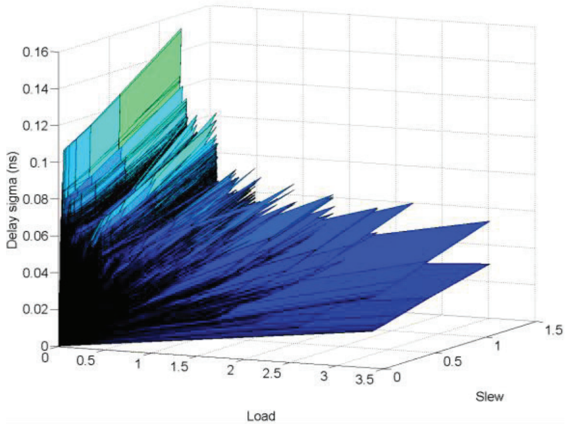


Fig. 3. All cell delay sigma look-up tables in the TT1P1V25C library, combined in a surface plot.

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, looking only at the setup time. 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.

Table 1. Clock periods for different constraints and libraries

	Clock period
High performance	2.41 ns (zero slack)
Medium performance	4 ns
Low performance	10 ns
Close to maximum check	2.5 ns

Table 2. Constraint parameters used during threshold extraction

	Constrain 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

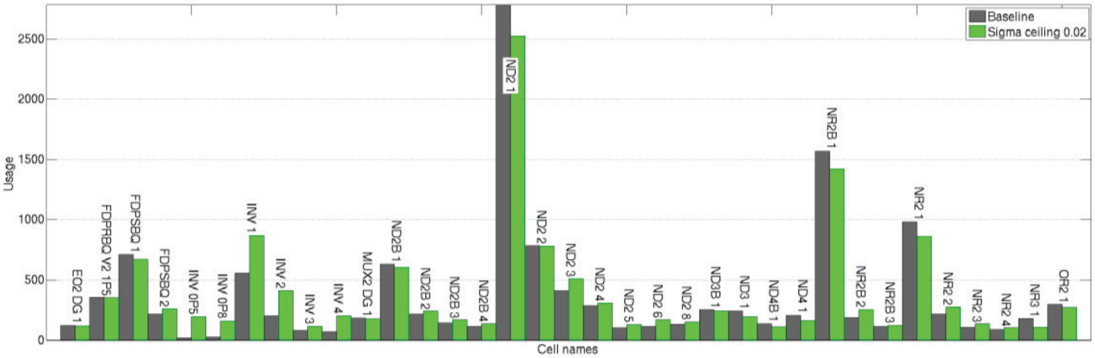


Fig. 4. Cell use for a baseline synthesis and tuning method at a clock period of 2.41ns. Only cells which are used more than 100 times are listed.

The minimum clock period is found by reducing the clock period until the synthesis fails to provide a design with positive slack. The relaxed timing condition is chosen based on the point where an increase in the clock period will not reduce the area noticeably. 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. 3. 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 leaving 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. 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.

Figure 4 shows a histogram with the cells that are used in the baseline synthesis for the high performance timing. Only cells that were used more than 100 times are listed. Interesting to see is that in the synthesis of the processor, basic cells (NAND, NOR, INV and flip-flops) are more often used.

Figure 5 shows the highest sigma reduction for an area increase of less than 10% compared to the baseline, for different timing constraints. Each bar in the figure represents a tuning method. We can see from the annotations 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. 5 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 5 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, reducing the local variation further comes at the cost of a high area increase. For illustration purposes, the sigma ceiling method is discussed further since it clearly shows the effects of the library tuning approach.

A. *Impact of library tuning on data-path depth*

The area increase introduced by the library tuning can be explained by inspecting the path depth. 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. 4, illustrates an increase in the overall use of inverter cells. An inverter cell can be used as a buffer. Secondly, Fig. 4 confirms the increase in high drive strength cells. Looking at cell NR2B 1 (2-input NOR cell with drive strength 1), we note that 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.

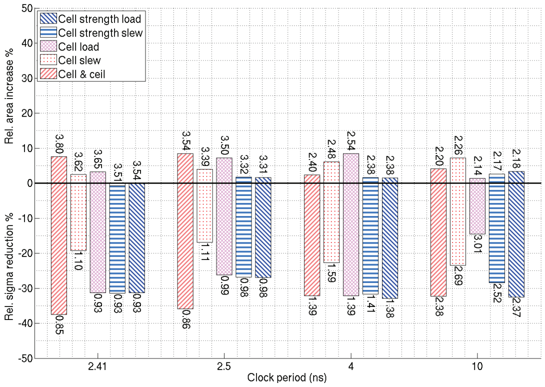


Fig. 5. Relative sigma decrease and area increase between baseline and tuning methods with highest sigma reduction at an area increase less than 10%, for different clock periods. The top part is the relative area increase with the real area value annotated (in 10^4 μm^2). The bottom part is the relative sigma decrease with the real sigma value annotated (in ns).

B. *Impact of data-path depth on local variation*

Figure 6 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 the sigma reduction is most noticeable in the short and medium size paths. The figure also shows 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.

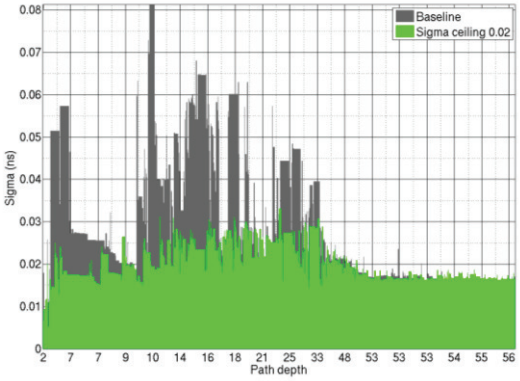
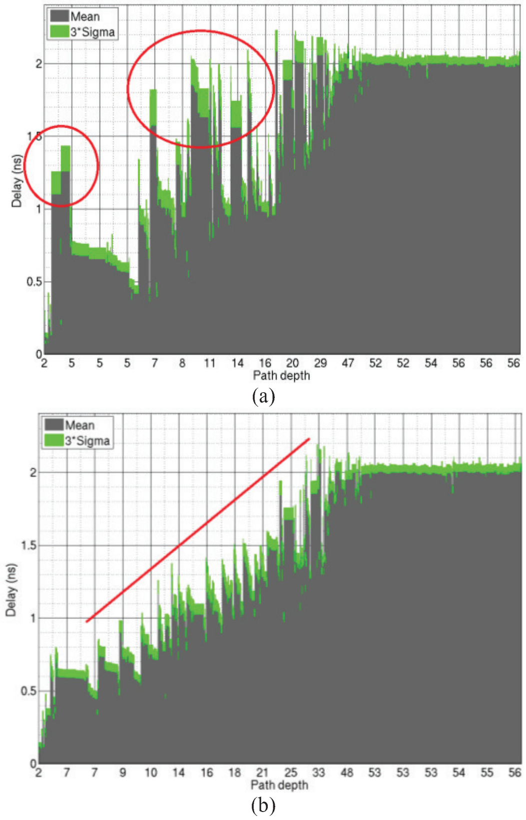


Fig. 6. Sigma versus path depth for the baseline and the sigma ceiling method are shown. The sigma is reduced especially in the low and medium sized paths.

In Fig. 7a the mean and three sigma values of every path are shown for the baseline synthesis of the 2.41ns design. Worth mentioning is that during synthesis a guard band of 300ps was used so the effective clock period becomes 2.11ns. The paths are sorted according to their depth. The vertical axis shows the path delay. 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. 7a 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. 7a). 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. 7b, 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 the previous 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 (fast, typical, and slow) as shown in Fig. 8. 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.



(a)

(b)

Fig. 7. Mean + 3 sigma path delay per path depth for (a) baseline synthesis and (b) sigma ceiling restriction with a clock period of 2.41ns. Note that in the baseline design (a), there are (short) paths with a high sigma. This behavior is no longer visible in the constraint design (b), there is more homogeneity.

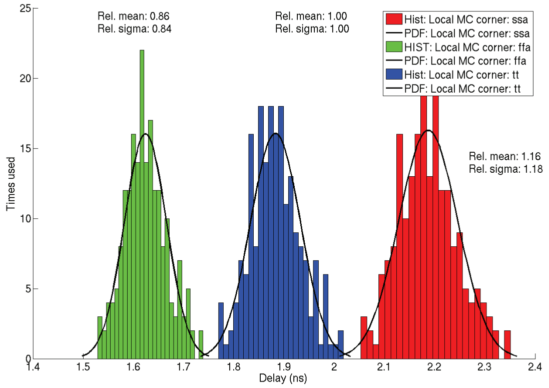


Fig. 8. Monte Carlo simulation (N=200) for medium sized path (18 cells), extracted from the design at a clock period of 2.41ns. The histograms are there for a fast, typical and slow corner, respectively. The sigma and mean relative to the typical corner is shown in text. Both the mean and sigma scale accordingly when moving to different corners.

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. Figure 9 gives the histogram plots for short and long size paths, respectively. The paths are extracted from the 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 includes local variation only. 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.

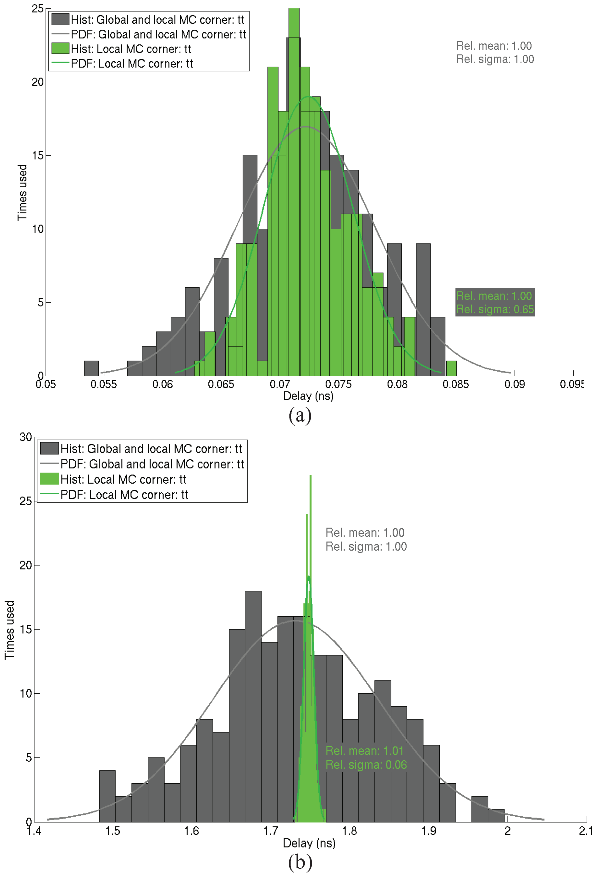


Fig. 9. Monte Carlo simulation (N=200) for two extracted paths from the design at a clock period of 2.41ns. The short path (a) has 3 cells and the long path (b) has 57 cells. The histograms are for global and local variation and for local variation only to show the impact of both variation types. The sigma and mean relative to the typical corner is shown in text.

VI. 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.

VII. REFERENCES

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

[2] 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.

[3] 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.

[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] F. Beeftink, P. Kudva, D. Kung, L. Stok, "Gate-size selection for standard cell libraries," *IEEE/ACM International conference on computer-aided design (ICCAD),* San Diego, 1998, pp, 545-550.

[8] V. Singhal, G. Girishankar, "Optimal gate size selection for standard cells in a library," *CAS workshop on design, applications, integration and software,* 2006, pp. 47-50.

[9] B. Liu, J.P. de Gyvez, M. Ashouei, "Library tuning for subthreshold operation," *Subthreshold microelectronics conference (SubVT),* 2012, pp. 1-3.

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

[11] 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.

[12] 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.

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