

# Placement Method Targeting Predictability Robustness and Performance

Cristinel Ababei

Kia Bazargan

Electrical and Computer Engineering Department  
University of Minnesota, Minneapolis, MN 55455  
{ababei, kia}@ece.umn.edu

## ABSTRACT

We study the relationship between robustness, predictability and performance of VLSI circuits. It is shown that predictability and performance are conflicting objectives. Performance and robustness are statically conflicting objectives but they are statistically non-conflicting. We propose and develop means for changing a standard timing-driven partitioning-based placement algorithm in order to design more predictable and robust circuits without sacrificing much of performance.

## 1. INTRODUCTION

Ideally, we would like a design methodology to offer predictable and robust designs at the best performance. High robustness means that performance of the design is less influenced by noise factors and remains within acceptable limits, *i.e.*, the design is more tolerant to perturbations – such as process variations, temperature and supply voltage changes – and therefore more reliable.

We would also like our design methodology to be predictable. Accurate estimation techniques would allow correct decisions early in the design process, which would result in fast design convergence. Predictability is to be achieved in the face of design uncertainties, which are caused by either incomplete system specification or inherent difficulty of estimating performance metrics during the optimization process.

Several ways of defining predictability or uncertainty have been proposed. Uncertainty was considered as the unpredictable process variations, which can cause delays to change from their nominal values [10]. Srivastava and Sarrafzadeh define predictability as the quantified form of the accuracy of the cost function estimation [13]. In the context of floorplanning, Bazargan *et al.* describe uncertainty as multiple values for heights and widths of the same module, and the goal is to minimize a linear combination of the expected value and the standard deviation of the area of the floorplan [14]. Wang *et al.* consider uncertainty at the placement level as the incomplete information about modules in the netlist [23]. To make routing more predictable at the placement level, one can use techniques for increasing the flexibility of rectilinear Steiner trees [17]. Kahng *et al.* describe signals as having no uncertainty when they all arrive simultaneously, which means the output of the cell has little or no uncertainty [15]. However, when the uncertainty of the signals varies simultaneous arrival of all signals will actually cause greater average and standard deviation of the output cell distribution [9]. A design process is considered predictable in [16] when the analytical or statistical predictive tools are accurate and allow providing constraints for the following design steps.

In this paper we analyze the relationship between robustness, predictability and performance (optimality) and seek means for their control. We apply our methodology to timing-driven partitioning-based placement in order to design predictable and robust circuits. We regard the optimization process under uncertainty as the

iterative computation of a number of objective functions, which depend on variables whose values are known within a range of values (*i.e.*, as probability distributions or as intervals within which these variables lie). Predictable design means the ability to accurately compute the objective function (within the chosen modeling framework), and to find means of making current estimations closer to the real final values. We use the standard deviation (st. dev. - fraction of the mean delay value) as the measure of predictability of the overall circuit delay distribution at the primary outputs, as well as at the output of each cell inside the circuit. This means that the smaller the st. dev., the more predictable is the delay. The slope of the variation of the st. dev. of the overall circuit delay, when gate and wire delays change, characterizes the robustness of the circuit.

## 2. STATISTICAL TIMING ANALYSIS

We use statistical timing analysis (StTA) as a modeling framework for the purpose of characterizing circuits from the predictability and robustness perspectives. Uncertainties in gate and wire delays (such as fabrication variations, changes in supply voltage and temperature [2] [3]) are modeled in *statistical timing analysis* by modeling gate and wire delays as random variables (*i.e.*, probability distribution functions). We adopt the approach proposed by Berkelaar [4] [5] for its simplicity and because it represents the formulation which appears in other recent statistical timing analysis techniques [6] [7]. Delay distribution at primary outputs is obtained by computing the statistical latest arrival times. Statistical delays are forward-propagated from primary inputs (PIs) towards primary outputs (POs), using *statistical* addition and maximum operations. Interested readers can find details on the statistical addition and maximum operations in [5].

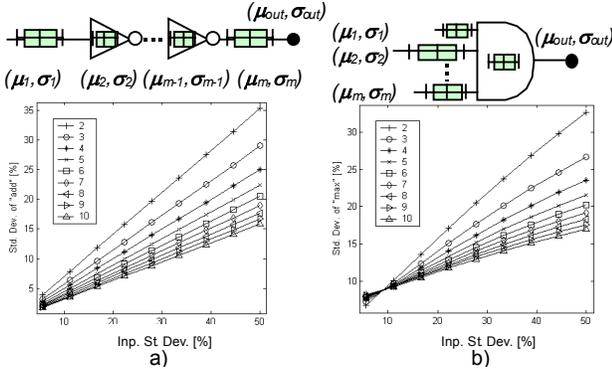
## 3. PREDICTABILITY ANALYSIS

To understand the mechanisms behind the interaction between the statistical addition and maximum operations and the predictability of a circuit we performed some studies. Initially, our focus is on two very simple toy-cases: (1) a chain (*e.g.*, path) of delay elements and (2) a generic gate with a given number of input pins with statistical arrival times. The output of the first case is calculated by successive *statistical add* operations, while the output of the second case is computed using the *statistical max* of the inputs, followed by the statistical addition of the gate delay. In each case, we want to find the variation of the st. dev. at the output  $\sigma_{out}$  when the st. dev. of the delay elements vary, for different values of the number of delay elements (or inputs)  $m$ . When the st. dev. of the delay elements vary within [5%, 50%] of their means, the output st. dev.  $\sigma_{out}$  varies as shown in Fig. 1.

**Observation 1:** We note that for large st. dev. of gate and wire delays (larger than about 9%), circuits that consist of gates with larger number of inputs show more robustness and better

predictability (Fig. 1.b). On the contrary, when gate st. dev. is small, fewer inputs translates into better robustness.

This observation has an implication for CAD developers: it can be used in a physical synthesis process to minimize circuit delay variations. Gates that are placed in regions with high temperature variations (e.g., close to a floating-point unit that is active only part of the time), and hence larger delay variations, can be mapped to large fanin gates in the library. Note that mapping gates to larger fanin gates could have a negative impact on the area or even delay of the circuit. Hence, the technology mapping process has to be done judiciously.

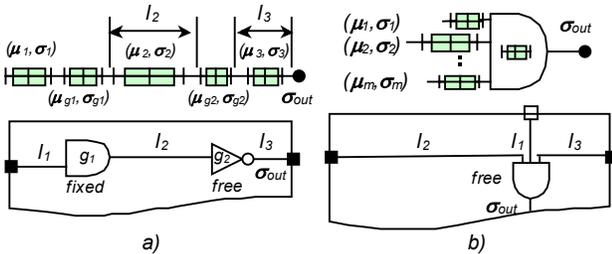


**Fig. 1 a) St. dev. at the output  $\sigma_{out}$  of a series of  $m$  delay elements b) St. dev. at the output of a gate with  $m$  inputs**

**Observation 2:** Better predictability and robustness is obtained for interconnects with more buffers (Fig. 1.a). Furthermore, smaller wire and inverter delay variations results in smaller output variation.

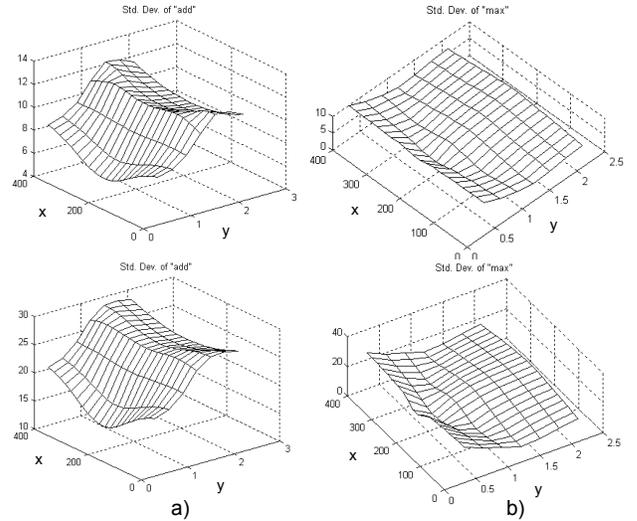
The observation is good news for interconnect optimization: more buffers not only helps in reducing the delay, it also helps reduce the variation at the output. Furthermore, this observation confirms the intuition that the uncertainty adds up: the more uncertainty individual elements in a path have, the more uncertain the output would be. It is important to note that observation 2 should not be generalized to any chain of gates with possible converging paths. As will be shown in Fig. 5.a, more elements on the chain does not necessarily result in smaller deviation at the output.

Next, we analyze the behavior of  $\sigma_{out}$  when two elements in the chain vary in opposite directions. This situation can happen when the length of a net increases while the length of a different net decreases but the sum of both remains the same (e.g., inverter free to move to left or right in Fig. 2.a). A similar situation can appear among the delay elements at the inputs of a gate when the decrease in a latest arrival time at one input can be at the expense of the increase of one at another input (e.g., gate free to move to left or right in Fig. 2.b).



**Fig. 2 Study cases**

The simulation result of the two cases is shown in Fig. 3. Plots in Fig. 3.a show that the st. dev. at the output of a chain is minimum when  $l_2=l_3$  and that it is better observable when the st. dev. of the elements in the chain increase (bottom). This is true as long as the contribution to  $\sigma_{out}$  of the two changing delays is significant (i.e., comparable to the contribution of the rest of delay elements - small values on y axis). When the contribution to  $\sigma_{out}$  of the two changing delays is very small (e.g., wire delay of wires of lengths  $l_2$  and  $l_3$  are much smaller than the wire delay of wire  $l_1$  and the gate delays), the plots become almost flat (for large  $y$  in Fig. 3) because the effect of length change of wires of length  $l_2$  and  $l_3$  would be absorbed by the computation of the overall  $\sigma_{out}$ . When the delay elements are latest arrival delays at the inputs of a gate (Fig. 2.b), and the contribution to  $\sigma_{out}$  of the two changing delays is significant, the minimum  $\sigma_{out}$  is achieved when  $l_2=l_3$  only for large st. dev. ( $>9\%$ ) of all delay elements. For small st. dev. of all delay elements, the minimum  $\sigma_{out}$  is achieved at the extremes when  $l_2=0, l_3=\max$  length or  $l_2=\max$  length,  $l_3=0$ .



**Fig. 3 Standard deviation when one delay element increases and another decreases for a) a series of delay elements and b) delay elements as LATs gate inputs. The x axis represents the varying length  $l_2$  and the y axis is the ratio between the sum of means of fixed elements over the sum of varying means. Default st. dev. of all delay elements is 10% (top) or 25% (bottom)**

**Observation 3:** Plots in Fig. 3 suggest the following intuition to be used in a placement algorithm. To minimize the standard deviation of the gate output delay, the LATs at its inputs should be equalized (Fig. 3.b). Furthermore, the delays of the wires on a critical path should be equalized so that the output deviation is minimized (Fig. 3.a). Observation 3 implies that the placement method should be aware of statistical slack distributions.

## 4. ROBUSTNESS ANALYSIS

To study the relationship between robustness and optimality, we adopt the methodology proposed by Lopez et al in the context of optimization of interconnection network throughput [8]. The idea of a robust design methodology is to study the effect of factors on the performance and the interactions between such factors. The key point of such an analysis is to correctly identify and classify variables that affect the design performance within the modeling

framework of the optimization problem. These variables are classified into two categories: *controlled factors* and *noise factors*. Controlled factors are design parameters that can be controlled directly or indirectly. Noise factors are random effects that cause performance variation. In the case of the timing driven placement, the response variable is the overall circuit delay, which should be minimized.

In what follows, we focus our attention on a generic gate (Fig. 1.b - top). For simplicity, we use the maximum difference between the latest arrival times at the inputs of the gate as a *control factor* (i.e., the length of the range  $[min, max]$  within which all latest arrival times lie, called the *input range*). We choose input range because of three reasons. First, it would allow pursuing the same objective as in the case of predictability as illustrated in Fig. 3.b. Secondly, this factor can be indirectly controlled during placement. The placement algorithm can control the lengths of all nets along paths and thus perform a delay budgeting, which affects the latest arrival times at any point inside the circuit. Finally, the selection of this control factor was suggested by the results obtained by Hashimoto et al [9] and Bai et al [10]. The *noise factor* is chosen as the standard deviation of gate and wire delays<sup>1</sup>.

The first part of the experiment consists of selecting three different values for the control factor, simulating the model of the gate, generating groups of output samples for each of the three values of the control factor, and analyzing them. We constructed a toy-case using a three-input gate<sup>2</sup> where the control factor (i.e., input-range) can have one of the values  $\{0, 0.5, 1\}$  for a gate with three inputs determined by the following three sets of LATs:  $\{10, 10, 10\}$ ,  $\{9.7, 9.7, 10.2\}$ ,  $\{9.5, 10, 10.5\}$ <sup>3</sup>. The value of “0” for the input-range represents the case when the latest arrival times have equal means, therefore the input-range is zero, and so on. After the gate model is simulated, groups of 10000 samples of the output delay are generated for each value of the control factor. These groups are then analyzed using ANalysis Of VAriance<sup>4</sup> (ANOVA) method using Matlab [11]. Fig. 4.a shows the significance of the control factor.

The plot in this figure shows that the smallest mean (i.e., 12.73 shown on top of Fig. 4.a) for the gate delay is obtained when the input-range is 0.5. In this case, the set of latest arrival times  $\{9.7, 9.7, 10.2\}$  is, from a statistical perspective, better than the set  $\{10, 10, 10\}$ . Note that static timing analysis would choose  $\{10, 10, 10\}$  as the best because it has a static delay of 10. Controlling the *input-range* as a control factor can be used in a placement algorithm to achieve a robust design that is less sensitive to delay variations.

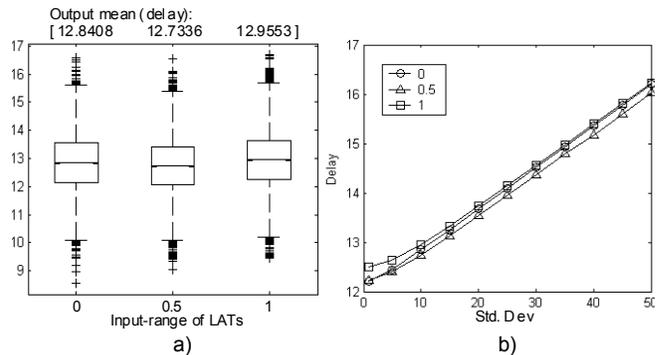
<sup>1</sup> Other possible noise factors, not considered in this experiment include: correlations between lines inside the circuit due to fanout re-convergence, approximation of the density function of all gate and wire delays with the normal distribution and, input patterns applied at the primary inputs of the circuit, which may not be known during the design.

<sup>2</sup> We performed similar analyses for gates with different number of inputs and similar results were obtained. We restrict our presentation to the three-input case for simplicity.

<sup>3</sup> The actual delay value, i.e., 10, is not relevant. Only the range matters.

<sup>4</sup> ANOVA is a well-known statistical method for studying the effect of control factors on the average value of the response variable, which in our case is the mean delay [8] [11].

The second part of the experiment studies the interaction between the control and the noise factors. Fig. 4.b shows the impact of the noise factor on the mean delay at the output of the gate for the three different values of the control factor. It can be seen that when the input range is 0.5, the slope of the output delay is smaller than the slope in the case when the input-range is 0. This means that the delay at the output increases at a higher rate when the LATs at the inputs are equal. Therefore, the gate is more robust to variations when the input-range is different from zero. In other words, if our modeling is based on static timing analysis, optimality (best cell delay is obtained when the input-range is zero) and robustness (cell is more robust for input-range 0.5 - Fig. 4.b) are conflicting objectives. On the other hand, if our modeling is based on statistical timing analysis, optimality (middle box in Fig. 4.a) and robustness (case 0.5 in Fig. 4.b) are non-competing objectives.



**Fig. 4 a) Average output delay of a three-input gate with input LAT range of 0, 0.5, and 1 b) Average cell delay for the combination of the control factor with the noise factor**

Results in the above analysis are in agreement with those obtained in [9] and exploited for circuit slack optimization in [10]. These results give us insight into the mechanisms, which determine a design to be more optimal or more robust and helps in identifying means for controlling them. It is often more costly to control causes of variations than to make a design process less sensitive to these variations.

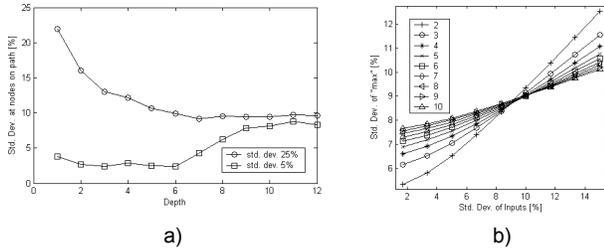
## 5. CASE STUDY: PARTITIONING-BASED PLACEMENT

We now describe how we can develop means for modifying a standard partitioning-based placement algorithm in order to achieve more predictable and robust circuits. Based on the analyses presented in the previous sections, we propose a new net-weight assignment scheme that we integrated into a timing-driven partitioning-based placement tool. The goal is to change the behavior of the placement such that the final placement solution is predictable and robust without sacrificing too much performance. Our placement tool is a modified version of Capo [12]. We developed our customized Capo placement algorithm by replacing the multi-level and flat partitioning algorithms of Capo with a timing-driven version of hMetis, a leading partitioning algorithm

[1]. Timing is minimized using timing criticalities (slack-based<sup>5</sup>) as net-weights inside the partitioning engine.

Two main observations are the basis for our motivation in the derivation of our new net-weight assignment scheme. The first observation is that the closer a wire is to the POs, cutting it is likely to have greater impact on the circuit delay, as it is more likely to lie on many different critical paths<sup>6</sup>. Second, st. dev. of the latest arrival times on timing paths decreases from PIs towards POs for large st. dev. for all wire and gate delays inside the circuit. However, the decreasing trend is not maintained for small st. dev. That is shown in Fig. 5.a, which depicts the st. dev. of the latest arrival times at intermediate nodes on the most critical path for typical st. dev. of 25% and 5% for two different placements of the circuit *too\_large*. It can be observed that as the statistical timing analysis advances towards POs with the computation of the latest arrival times, st. dev. tends to converge to values within a *convergence-region*. By comparing Fig. 5.a to Fig. 1.b, we can see that the behavior of a chain of inverters is quite different from a path with gates of different sizes and with converging fanins.

It can be seen in Fig. 5.a that for small st. dev. of the LATs at the inputs of a gate, the statistical maximum operation increases with depth. However, for large st. dev. values, the st. dev. of the intermediate nodes along the path actually decreases. The difference in behavior of small and large st. dev. values is similar to what we can observe in Fig. 5.b, which is the enlarged bottom-left corner of Fig. 1.b. For small (large) values of st. dev., gates with smaller (larger) number of inputs can better tolerate input variations. Gates at small (large) depths of the circuit show a similar statistical behavior as gates with smaller (larger) fanins.



**Fig. 5 a) St. dev. of LATs at nodes on critical path for *too\_large* for 25% and 5% st. dev. for wire and gate b) Enlarged bottom-left corner of Fig. 1.b**

In order to develop a methodology that minimizes output variations, we have to consider two factors: (1) how we can affect the deviation (the control factor discussed in Section 4), and (2) how effective our effort would be in minimizing the deviation at the output of a gate (Fig 5.a). We can control the deviation by affecting how close are the latest arrival times at the inputs of a given gate. Furthermore, the position of the gate on the path (close to PIs or POs) indicates how much influence we would have on optimizing the deviation. Therefore, for large (small) st. dev. values, we would like gates close to the PIs (POs) have their latest arrival times as

<sup>5</sup> Timing criticality is computed as inversely proportional to slack. The smallest slack determines the largest weight associated to the corresponding net, and so on.

<sup>6</sup> This observation was directly derived from our placement experiments, and not from the analyses in the previous sections.

close as possible in order for the maximum operation to provide the smallest st. dev. (see Fig. 3.b).

The delay of an input can be controlled by changing the bounding box of the net connected to it (in a partitioning-based placement algorithm, cutting a net at a higher level results in larger bounding box). Our new net-weight assignment scheme for large st. dev. is described by the following equation:

$$w_i = B_i \cdot (\alpha \cdot w_i^{slack} + \beta \cdot w_i^{lat}) \quad (1)$$

where,  $B_i$  is a biasing factor to emphasize weights of nets driven by nodes close to the PIs (nodes with small logic depths). The classic slack-based net-weight component is  $w_i^{slack}$  and  $w_i^{lat}$  is the “lat” net-weight component, which is introduced to achieve LATs equalization at the inputs of gates (nodes) with small logic depths. Parameters  $\alpha$  and  $\beta$  are used to put more weight on the “slack” or the “lat” weight components.

The biasing factor  $B$  will have values such that nets at large logic depths are cut more easily in order to capture the phenomenon of decrease of the standard deviation of the propagated LATs as described in the beginning of this section. At small logic depths (where the st. dev. of propagated LATs and the default st. dev. of gate and wire delays are large - Fig. 5.a) we put more emphasis on the “lat” net-weight component, which accounts for equalization of LATs in order to decrease the st. dev. as shown in Fig. 3 - bottom and to achieve robustness as described in Section 4).

## 6. SIMULATION EXPERIMENTS

We report simulation results obtained with the placement algorithm described in the previous section for a set of MCNC91, IWLS93 [21] and ITC99 [22] (last two in Table 1) circuits in two different scenarios. In the **first scenario** after the circuit is placed we set all gate and wire delays as samples of the corresponding distributions. This case mimics the manufacturing process when gate and wire delays can vary due to process variations [3]. In the **second scenario** we model the increase of gate and wire delays due to temperature increase. We consider a pattern where gate and wire delays increase by 25%<sup>7</sup> at the center of the chip, by 5% at the boundaries, and by 10% elsewhere. Although the temperature pattern on a chip can be different and the increase of delay larger [18] [19], we consider a rather simple pattern for simplicity. The simulation results (average of ten different runs) are presented in Table 1. After the placement is performed using the classic placement algorithm and our placement algorithm we compute the circuit delay denoted as *Delay* (static delay is computed using the Elmore delay for the lumped RC wire model and the half perimeter of the bounding box for the wire-length of a net) using the static timing analysis. Then, we model gate and wire delays as random variables and perform statistical timing analysis to obtain the *St. Dev.* (as percentage of the mean delay value) of the overall circuit delay.

After injecting simulated noise (based on scenarios 1 and 2) to gate and wire delays, static analysis is repeated and the change in delay is reported as *Delay-change* in Table 1. St. Dev.

<sup>7</sup> Interconnect Elmore delay increases approximately 5% for every 10°C increase and current designs can drive the operating temperature to higher than 100°C [20].

characterizes predictability while delay-change shows how robust the placement is. It can be seen that the standard deviation of the overall circuit delay is consistently smaller for placements obtained with our placement algorithm (the rather small differences are due to the convergence property described in Fig. 5.a), which means better predictability. The delay changes after noise injection are smaller for placements obtained with our placer, which means better robustness (34% and 8% on average in scenarios 1 and 2). A main benefit of better predictability and robustness and the same delay is the improved manufacturing yield. The run time is not reported because both placement algorithms have almost the same run-time. The run-time for the largest circuit is around 2 hours on a 1.5GHz CPU, 2GB memory running on Linux.

## 7. CONCLUSIONS

Analyses on the relation between predictability, robustness, and performance of VLSI circuits were presented. They served to finding novel means to change a standard timing-driven partitioning-based placement algorithm in order to design more predictable and robust circuits without sacrificing much in performance.

## REFERENCES

- [1] G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar, "Multilevel Hypergraph Partitioning: Application in VLSI domain", *Proc. ACM/IEEE DAC*, 1997.
- [2] S. R. Nassif, "Modeling and Forecasting of Manufacturing Variations", *Proc. ACM/IEEE ASP-DAC*, 2001.
- [3] D. Sylvester, "Measurement Techniques and Interconnect Estimation", *Proc. System-Level Interconnect Prediction Workshop (SLIP)*, 2000.
- [4] M. Berkelaar, "Statistical Delay Calculation, a Linear Time Method", *Proc. Int. Workshop on Timing Issues in the Specification and Synthesis of Digital Systems (TAU97)*, 1997.
- [5] E. T. A. F. Jacobs and M. R. C. M. Berkelaar, "Gate sizing Using a Statistical Delay Model", *Proc. ACM/IEEE DATE*, 2000.
- [6] M. Orshansky and K. Keutzer, "A general probabilistic framework for worst case timing analysis", *Proc. ACM/IEEE DAC*, 2002.
- [7] A. Agarwal, D. Blaauw, V. Zolotov, and S. Vrudhula, "Statistical Timing Analysis using Bounds", *Proc. ACM/IEEE DATE*, 2003.
- [8] P. Lopez, R. Alcover, J. Duato, and L. Zunica, "Optimizing Network Throughput: Optimal versus Robust Design", *Proceedings of the Seventh Euromicro Workshop on Parallel and Distributed Processing (PDP)*, 1999.
- [9] M. Hashimoto and H. Onodera, "Increase in Delay Uncertainty by Performance Optimization", *IEICE Trans. Fundamentals*, Dec. 2002.
- [10] X. Bai, C. Visweswariah, P. N. Strenski, and D. J. Hathaway, "Uncertainty-Aware Circuit Optimization", *Proc. ACM/IEEE DAC*, 2002.
- [11] <http://www.mathworks.com>
- [12] E. Caldwell, A. B. Kahng, and I. L. Markov, "Can Recursive Bisection Alone Produce Routable Placements?", *Proc. ACM/IEEE DAC*, 2000.
- [13] A. Srivastava and M. Sarrafzadeh, "Predictability, Analysis and Optimization", *Proc. ACM/IEEE ICCAD*, 2002.
- [14] K. Bazargan, S. Kim, and M. Sarrafzadeh, "Nostradamus: A Floorplanner of Uncertain Designs", *IEEE Trans. CAD*, Apr. 1999.
- [15] A. B. Kahng, R. Kastner, S. Mantik, M. Sarrafzadeh, and X. Yang, "Studies of Timing Structural Properties for Early Evaluation of Circuit Design", *Proc. The Tenth Workshop on Synthesis and System Integration of Mixed Technologies*, Oct. 2001.
- [16] E. Shragowitz, H. Youssef, and L. C. Bening, "Predictive Tools in VLSI System Design", *Proc. COMEURE*, 1988.
- [17] E. Bozorgzadeh, R. Kastner, and M. Sarrafzadeh, "Creating and Exploiting Flexibility in Steiner Trees", *Proc. ACM/IEEE DAC*, 2001.
- [18] H. Ajami, K. Banerjee, M. Pedram, and L. van Ginneken, "Analysis of Non-Uniform Temperature-Dependent Interconnect Performance in High Performance ICs", *Proc. ACM/IEEE DAC*, 2001.
- [19] Y. K. Cheng and S. M. Kang, "A Temperature-Aware Simulation Environment for Reliable ULSI Chip Design", *IEEE Trans. on CAD*, Oct. 2000.
- [20] C. H. Tsai, S. M. Kang, "Cell-Level Placement for Improving Substrate Thermal Distribution", *IEEE Trans. on CAD*, Feb. 2000.
- [21] <http://www.cbl.ncsu.edu>
- [22] <http://www.cad.polito.it/tools/9.html>
- [23] M. Wang, P. Banerjee, and M. Sarrafzadeh, "Potential NRG: Placement with Incomplete Data", *Proc. ACM/IEEE DAC*, 1998.

Circuit	No. of gates	PI/PO	Classic placement				Our placement			
			Delay	St. Dev [%]	Delay-change [%] Scenario 1	Delay-change [%] Scenario 2	Delay	St. Dev [%]	Delay-change [%] Scenario 1	Delay-change [%] Scenario 2
Cordic	881	23/2	7.54	7.45	8.48	12.57	7.7	7.36	6.79	11.94
Dalu	883	75/16	8.17	8.55	5.95	12.39	8.26	8.54	4.43	10.81
Misex3	1349	14/14	14.46	8.02	5.53	10.74	15.29	7.96	3.39	10.61
C5315	2062	178/106	11.2	8.9	9.04	12.86	12.06	8.18	2.99	11.86
C7552	2387	206/35	13.37	8.29	5.64	9.56	13.23	8.31	4.67	10.42
Des	3451	256/245	10.8	9.6	4.19	7.2	10.63	8.9	3.89	7.91
C6288	3598	32/32	37.02	8.71	3.82	11.44	37.4	8.66	2.18	11.8
Elliptic	4711	130/112	22.07	8.79	5.13	13.01	25.1	8.19	5.14	11.46
Pdc	4821	16/40	33.37	8.68	4.9	9.78	34.13	8.67	7.62	9.21
Ex1010	4898	10/10	28.57	8.45	3.79	9.8	28.42	8.45	2.97	10.79
too_large	6961	38/3	19.65	8.09	8.66	8.85	21.15	8.09	6.98	6.49
S35932	11304	35/32	21.7	10.81	10.71	9.51	24.08	10.56	6.02	7.63
S38584	12701	115/74	20.73	12.68	17.01	8.13	26.38	10.6	2.87	8.11
B21s	14606	32/22	147.07	8.35	2.53	11.28	155.07	8.35	1.45	10.71
B17s	36547	37/30	181.79	8.51	5	12.01	219.68	8.44	4.44	7.18
<b>Ratio of harmonic mean values</b>							<b>1.04</b>			
<b>Ratio of average values</b>							<b>0.96</b>	<b>0.66</b>	<b>0.92</b>	

**Table 1 Comparison of the proposed placement algorithm to the classic net-based timing-driven partitioning-based placement. Delay is the delay reported by a static timing analysis algorithm and St. Dev. is the standard deviation of the overall circuit delay after placement (the smaller it is the more predictable is the circuit). Delay-change is the change in static delay after the placement is changed in scenarios 1 or 2 (smaller means circuit more robust).**