

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800

Open access books available

122,000

International authors and editors

135M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Multivariate Adaptive Regression Splines in Standard Cell Characterization for Nanometer Technology in Semiconductor

Taizhi Liu

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/intechopen.74854>

Abstract

Multivariate adaptive regression splines (MARSP) is a nonparametric regression method. It is an adaptive procedure which does not have any predetermined regression model. With that said, the model structure of MARSP is constructed dynamically and adaptively according to the information derived from the data. Because of its ability to capture essential nonlinearities and interactions, MARSP is considered as a great fit for high-dimension problems. This chapter gives an application of MARSP in semiconductor field, more specifically, in standard cell characterization. The objective of standard cell characterization is to create a set of high-quality models of a standard cell library that accurately and efficiently capture cell behaviors. In this chapter, the MARSP method is employed to characterize the gate delay as a function of many parameters including process-voltage-temperature parameters. Due to its ability of capturing essential nonlinearities and interactions, MARSP method helps to achieve significant accuracy improvement.

Keywords: multivariate adaptive regression splines (MARSP), semiconductor microelectronics, standard cell characterization, very large scale integration (VLSI), variation-aware gate-delay modeling, statistical timing analysis

1. Introduction

Multivariate adaptive regression splines (MARSP) was first proposed by Friedman [1] for solving regression-type problems. MARSP is widely used to predict the values of an outcome variable from a set of predictor variables. There are many methods for model fitting, and MARSP is one of them. Other modeling techniques include linear regression (e.g., general

linear model), nonlinear regression (e.g., generalized linear/nonlinear models), and regression trees (e.g., classification and regression trees), and so on. It is also worth noting that a neural network, which is very popular nowadays in an era of artificial intelligence and machine learning, is also a modeling technique.

MARSP is a nonparametric regression procedure that makes no assumptions about the underlying functional relationships between dependent and independent variables. The form of MARSP and its coefficients are entirely derived from the regression data. The modeling strategy is called “divide and conquer,” by which the input space is partitioned into a number of regions, with each region having its own regression equation. This makes MARSP particularly efficient for high-dimension problems, where other techniques most likely have accuracy issues.

As the name suggests, MARSP uses splines as its main component. Splines are piecewise curves from polynomial functions. When different splines are smoothly connected, it can result in a flexible model which can handle both linear and nonlinear situations. The connection points between different pieces are called knots, which connect the end of one region of data and the beginning of another.

The MARSP technique has been particularly popular in data mining because it does not require or assume any particular type or any class of relationship (e.g., logistic, linear, etc.) between the outcome variable of interest and the predictor variables. Instead, MARSP derives useful models (i.e., models that yield accurate predictions) even in situations where the relationship between the predictor variable and the predictor variables is difficult to approximate with parametric models. If you are interested in more information about MARSP and how it compares to other methods for nonlinear regression (or regression trees), please refer to Chapter 9 of [2].

2. Standard cell characterization in very large scale integration (VLSI) design

In semiconductor design, standard cell methodology is a method that is widely used for very large scale integration (VLSI) design, especially for digital logic circuits. It is a design abstraction, where the low-level circuit layout can be encapsulated into many abstract logic representations (e.g., NOR2, NAND2 cells). As a cell-based methodology, it can enable one designer to focus on the high-level aspect (logical function) of a design, while another designer can work on the implementation aspect (physical layout). As semiconductor fabrication technology progressed to sub-10 nm regime, standard cell methodology was the enabler to allow designers to scale application-specific integrated circuits (ASICs) from simple chips of several thousand cells, to complex chips with hundreds of millions of cells.

A standard cell provides a Boolean logic function (e.g., AND, OR) or a storage function (latch or flip-flop). A standard cell can be as simple as an inverter which consists of only two transistors. It can also be as complex as adders or multiplexers which have tens of transistors. As a standard cell is a logic gate, “cell” and “gate” are often interchangeable. Standard cell library is a collection of predefined cells which are usually fully customized to a specific technology and optimized

for best delay, power, area, and so on. The standard cell library provides a way for designers to place cells in rows, and it enables the automated layout generations for digital ASICs.

The objective of standard cell characterization is to create a set of high-quality models of a standard cell library that could accurately and efficiently model cell behavior. Cell behavior may come from a variety of parameters in different aspects like capacitance, power, timing, current, waveform, and so on. Among them, one of the most important models in standard cell characterization is timing delay models. In this chapter, for simplicity, we focus on the characterization of the pin-to-pin propagation delay of standard cells. Propagation delay is the time required for the input to be propagated to the output. In other words, it is defined as the time it takes for the effect change in input to be visible at the output. Propagation delay is important because it has a direct effect on the speed at which a digital device, such as a computer, can operate. This is true of memory chips as well as microprocessors. As mentioned earlier that a gate and a cell are often interchangeable in this chapter, cell delay and gate delay are interchangeable as well.

Cell propagation delays change with many factors, including the following:

1. The transition time of the input causing transition at the output.
2. The output load being felt by the logic cell/gate.
3. The process parameters (threshold voltages, channel lengths) of the transistors that the cell is consisted of.
4. The power supply voltage (VDD).
5. The temperature (Although temperature is not a factor with significant impact, it is still an impacting factor).

Among the different factors above, the process parameters are included because of the increasingly pronounced effect called process variations, which is introduced in details in Section 2.1.1. In the later sections, we use process-voltage-temperature (PVT) parameters to denote process parameters, VDD, and temperature.

2.1. Introduction (problem formation)

As mentioned above, one of the most important tasks in standard cell characterization is to find a model which can accurately capture the relationship between the cell propagation delay and the parameters that have impact on cell delay (as shown in the paragraph above). Here, the cell propagation delay is the response variable, and the impacting parameters (input transition time, output loads, VDD, and the process parameters) are the explanatory parameters.

We have not talked about the number of explanatory parameters yet. But as mentioned in Section 1, MARSP is suitable for the high-dimension problem while capturing essential nonlinearities and interactions. In the following subsections, we introduce the high-dimension parameter space when characterizing the delay models of standard cells, especially when the process variations and aging effect are included [3–7].

2.2. Parameter space

2.2.1. Process variations

When integrated circuits are fabricated, the parameters of individual transistors vary. The observed random distribution of identically drawn devices is caused by the fabrication process like impurity concentration densities, oxide thicknesses, and diffusion depths, and so on. These physical variations cause changes in the electrical characteristics of the transistors which eventually lead to the variability in the circuit performance. This is called process variation. Process variation is the naturally occurring variation in the attributes of transistors (length, widths, oxide thickness) during the chip fabrication. The scaling down of the VLSI process technologies has increased the process variations, especially in sub-45 nm era.

Process variations can be generally categorized into two classes: inter-die and intra-die variations. Inter-die variations occur from one die (chip) to another, meaning that the same transistor in the design can get different features (channel lengths, threshold voltages, etc.) among different dies (chips). Intra-die variations are variations in transistor features within a chip, meaning that transistors at different locations on the same die can get different features. Spatial correlations are often seen for intra-die variations, meaning adjacent transistors have a higher probability of having similar features than transistors that are far apart. In this work, we consider not only inter-die and intra-die variations, but also the intra-gate variations. Intra-gate variations are part of intra-die variations, in some sense. It is the variations within a gate (cell), meaning that the transistors within the same gate can have different features. While most of the literature works ignored the intra-gate variations, our work has included it. As VLSI technology continues to scale down to sub-10 nm process, intra-die variations (including intra-gate variations) are becoming more and more dominant.

The overall objective of standard cell characterization is to characterize a cell-delay model which is general and able to include inter-die, intra-die, and intra-gate variations with any kind of distribution and any correlation profile between different parameters. In this work, only process variations of standard cells are considered, meaning the variations in interconnect geometries are not considered.

2.2.2. Loading effect modeling (*pi*-model)

As technology scales down, the impact of interconnect on circuit timing cannot be neglected. In this work, we model interconnect as a resistive-capacitive (RC) network where all the capacitances are grounded.

A small patch of a gate-level circuit is illustrated in **Figure 1(a)**, where a driving Buffer gate has two loading gates, a NOR2 gate and an inverter gate. **Figure 1(b)** replaces the two loading gates with corresponding input capacitances. The input capacitances of loading cells, together with the interconnect network, form the load of the previous driving cell. With loading gates modeled as corresponding input capacitances, circuit timing can be analyzed in the way that each stage contains a standard cell and its connecting load as **Figure 1(b)** shows. If the readers are interested in the input-capacitance modeling of the standard cells, they can refer to [8, 9] for more details.

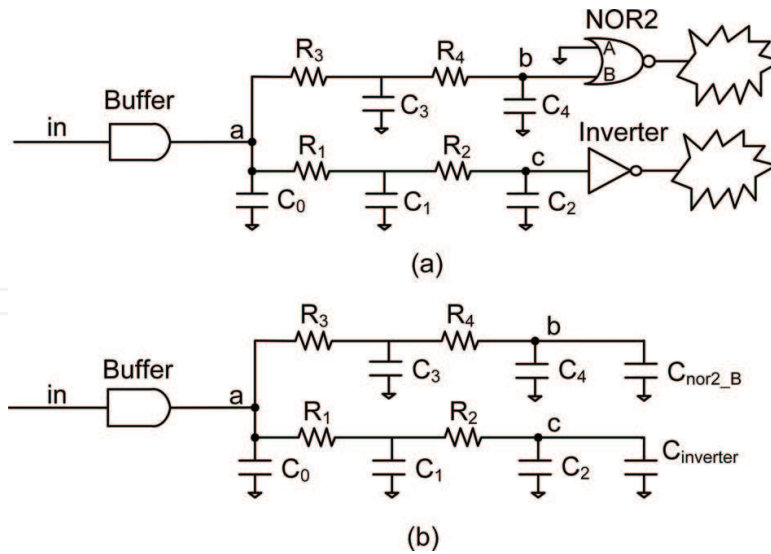


Figure 1. Input capacitances of the loading standard cells and the interconnect network formed the load of the previous driving gate. (a) A driving buffer and two loading cells (NOR2 and Inverter); (b) The driving buffer, with the input capacitances of the two loading cells incorporated into its load.

Reduced-order models are routinely used to replace the original large-order models. The Pi-model is the most popular reduced-order model to estimate the input admittance of RC interconnects. **Figure 2** gives the structure of the Pi-model, where $Y(s)$ denotes the input admittance of the original network and $Y'(s)$ denotes the input admittance of the Pi-model. The values of C_1 , R , and C_2 are obtained by equating the first, second, and third moments of the Pi-model to corresponding moments of the original network.

In Pi-model, we use three parameters to represent the loading effect of the whole RC interconnect. These three parameters C_1 , R , and C_2 , as well as the PVT parameters and input transition time, construct the parameter space for standard cell characterization which is introduced later in Section 3.

The shift in channel length (from the nominal value) is denoted as ΔL , and threshold voltage shift (from the nominal value) is denoted as ΔV_{th} . The supply voltage and temperature of a gate are denoted as ΔV_{DD} and ΔT , respectively, assuming that all the transistors within the same gate share the same voltage and temperature. The Pi-model which represents the load of a gate includes three parameters, namely R_{pi} , C_{pi1} , C_{pi2} . The input slew time (*Slope*) is also included for each timing arc. Note that in this work the effect of Multiple Input Switching (MIS) was not considered.

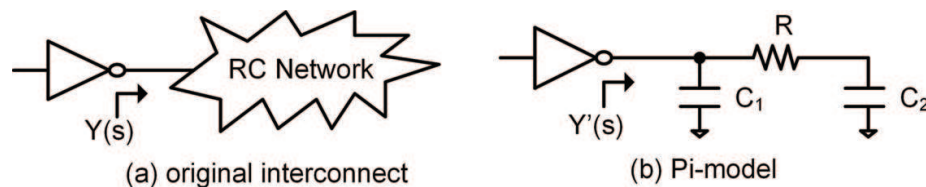


Figure 2. $Y'(s)$ in Pi-model as an approximation of original input admittance function $Y(s)$. (a) original interconnect (b) Pi-model.

For a cell which has N transistors, there are $2*N$ device parameters (i.e., $\Delta L, \Delta V_{th}$ for each transistor within the cell), and six global parameters ($\Delta VDD, \Delta T, R_{pi}, C_{pi1}, C_{pi2}, Slope$). This results in a total of $(2*N + 6)$ parameters for a cell with N transistors. In our experiments with a commercial standard cell library, the highest value of N is 32, making the highest $(2*N + 6)$ as 70, which results in a quite high-dimension parameter space for cell characterization.

At this point, we have not introduced the aging effect into the parameter space. If the characterized delay models need to be aging-aware, the aging parameters should be included in the parameter space. With aging parameters included, the dimension of the parameter space would be even higher. We discuss it in the following subsection.

2.2.3. High-dimensional parameter space in aging-aware standard cell characterization

For timing analysis, transistor aging is another source of variability besides PVT variations [10, 11]. Our work has considered the following wear-out mechanisms: bias temperature instability (BTI), hot carrier injection (HCI), and time-dependent dielectric breakdown (TDDB). The impact of BTI and HCI is similar as they both cause the threshold voltage of aged transistors to increase, which further decreases the driving strength and ultimately increases gate delay over time. TDDB degrades the drain current of the stressed devices which also results in increased gate delay. Overall, BTI, HCI, and TDDB ultimately cause the cell delay to increase over time. When the increased circuit delay exceeds the clock period, the degraded circuit will fail to work. Therefore, the aging effect needs to be taken into account in circuit timing simulations, especially for those high-reliability applications like aviation, space, automotive [12], medical [13–17], data center [18], and so on.

The variation of channel length and the variation of threshold voltage are denoted as ΔL and ΔV_{th} , respectively. For channel length, the variation (ΔL) comes from only process variation, while for transistor threshold voltage, the variation (ΔV_{th}) comes from both process variation and aging effect (BTI and HCI).

$$\Delta V_{th} = \Delta V_{th}_{process} + \Delta V_{th}_{BTI} + \Delta V_{th}_{HCI} \quad (1)$$

For the effect of TDDB, we need to include two additional parameters for each transistor within a gate, namely, R_{G2S} (gate-to-source resistance), and R_{G2D} (gate-to-drain resistance), because the gate-oxide breakdown paths can happen from gate-to-drain or from gate-to-source [19–22]. In summary, each transistor contributes four parameters ($\Delta V_{th}, \Delta L, R_{G2D}, R_{G2S}$) as input parameters. Thus, for a cell with N transistors, there are $4*N$ parameters for $\Delta L, \Delta V_{th}, R_{G2S}$ and R_{G2D} , plus six global parameters ($\Delta VDD, \Delta T, R_{pi}, C_{pi1}, C_{pi2}, Slope$), resulting in a total of $(4*N + 6)$ parameters for each cell.

As the value of N is as high as 32 in our experiments with a commercial library, the value of $(4*N + 6)$ can be as high as 134. Compared to 70, which is the value of $(2*N + 6)$ for cell characterization without aging effect, the dimension of parameter space in the aging-aware cell characterization has nearly doubled.

Var.	Corners	Var.	Corners
ΔL_p	[-30%, 30%]	ΔL_n	[-30%, 30%]
ΔV_{th_p}	[-30%, 30%]	ΔV_{th_n}	[-30%, 30%]
ΔV_{dd}	[-10%, 10%]	ΔT (°C)	[-50,50]
Slope	[10 ps, 3 ns]	C_{pi2} (pF)	[1100]
R_{pi} (Ohm)	[1, 1000]	C_{pi1} (pF)	[0.1,10]

Table 1. Variations and corners of considered parameters.

2.2.4. Training data

We have obtained our training data from simulation program with integrated circuit emphasis (SPICE) simulations. A mixture of central composite design and random samples are used for the design of experiments. **Table 1** shows the corners which are used for central composite design.

2.3. Why is multivariate adaptive regression splines (MARSP) better

Why is MARSP better than other methods in our application of standard cell characterization? Traditional methods like response surface methodology (RSM) use the same model to cover the entire parameter space. In our application where intra-gate variability is considered, the dimension of the parameter space is particularly high. When the number of input parameters is high, the parameter space is very high dimension. Using one single regression model to estimate gate delay (or slew time) over the whole parameter space is not sufficiently accurate, especially for a complex cell containing over 40 transistors. References [23, 24] proposed a clustering method which categorized transistors into switching/non-switching devices and on-transition/off-transition/non-transition devices. This method requires manual intervention to ‘filter out’ the negligible devices for each of the switching scenarios, which is quite cumbersome. Using MARSP, it can reduce the manual work and automatically capture the essential parameters in its intelligent process.

3. MARSP for standard cell characterization

This chapter employs MARSP to characterize a fitted function between response variables (gate delay or slew time) and the explanatory parameters (process-voltage-temperature parameters, aging parameters, and RC loads). MARSP uses piecewise polynomial segments to capture essential nonlinearities and interactions, and it is particularly suitable for high-dimension problems. This piecewise nature allows MARSP models to split the whole parameter space into multiple subspaces, and each subspace can have a unique regression model. By using hinge functions, MARSP then inherently integrates the regression models of all the subspaces into a single general form.

A hinge function has the form of $(x - t)_+$ or $(t - x)_+$ which are shown in **Figure 3**. They are defined as:

$$(x - t)_+ = \begin{cases} x - t, & \text{if } x > t, \\ 0, & \text{otherwise,} \end{cases} \tag{2}$$

$$(t - x)_+ = \begin{cases} t - x, & \text{if } x < t, \\ 0, & \text{otherwise,} \end{cases} \tag{3}$$

where t is a constant called the knot. MARSP forms a collection of hinge-function pairs for each explanatory parameter X_j with knots at $x_{j1}, x_{j2}, \dots, x_{jM}$ where M is the number of experiments.

MARSP models have the following form:

$$f(\vec{X}) = \beta_0 + \sum_{i=1}^T \beta_i h_i(\vec{X}) \tag{4}$$

where $h_i(\vec{X})$ is a basis function. There are two phases in the process of constructing a MARSP model: the forward stepwise addition and the backward stepwise deletion.

The first phase is the forward stepwise addition, where MARSP starts with a model consisting of an intercept term. It then repeatedly adds basis functions in pairs to the model step by step. At each step, MARSP finds the pair of basis functions which maximized the reduction in the residue sum-of-squares error. The two basis functions in the pair are identical except that the hinge functions used for each basis function are mirrored. The newly added basis function is constructed by a term that is already in the model (a constant 1 is also considered as an existing term) multiplied by a new hinge function. The process of forward addition phase continues until the residual error difference in two adjacent steps is smaller than a predefined threshold or until the number of terms in the model reaches the maximum.

The model from the forward addition phase usually overfits the data. If a model overfits, it means the model fits well to the training data that are used to build the model, but usually it does not fit to new test data. The second phase of MARSP, namely the backward stepwise deletion, is to build models that can generalize better to new data. Backward deletion phase

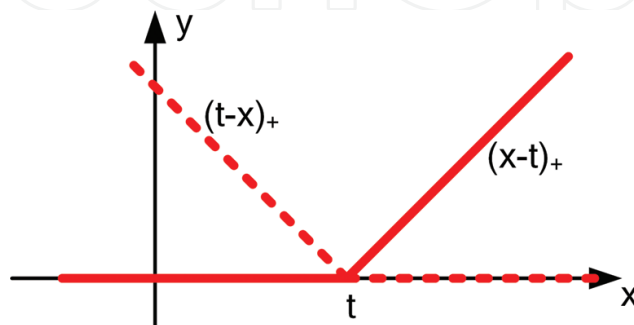


Figure 3. The solid line denotes the form of the hinge function $(x - t)_+$ while the dashed line denotes the hinge function $(t - x)_+$.

prunes the model obtained from the previous forward addition phase. In this phase, the technique called generalized cross validation (GCV) is used to trade off goodness-of-fit against model complexity. The stepwise backward deletion phase repeatedly deletes the least important term (according GCV) at each step until the model again has only the intercept term left. At the end of the backward deletion phase, from among the “best” models of each size at each step, the model with the lowest GCV value is selected, and it is outputted as the final model.

MARSP is a nonparametric regression method, so there are no predetermined forms of the model. Instead, the model is constructed adaptively according to the information extracted from the training data. It intelligently removes those negligible parameters that have limited impact on the to-be-modeled gate delays or output slew without manual intervention. Using MARSP for cell characterization can eliminate the need of clustering transistors into the categories of switching/non-switching devices and on-transition/off-transition/non-transition devices, as proposed in [23, 24].

The MARSP model is piecewise in nature, so MARSP can split the whole parameter space (which is high-dimension in our application) into multiple subspaces, with each subspace getting its own model. Then the regression models of all the subspaces are integrated into one general expression using piecewise hinge functions. In this way, MARSP can characterize standard cells only once over the whole PVT space, without the need of splitting parameter space and characterizing every subspace.

Figure 4 shows an example where the values of $h(X_1, X_2)$ change only when X_1 is large and X_2 is low. The nonlinear relationship can be easily handled by the following MARSP model:

$$h(X_1, X_2) = 15 + 0.015 * (20 - x)_+ * (y - 20)_+ \tag{5}$$

Figure 5 shows the quadratic regression model which has poor accuracy in this example.

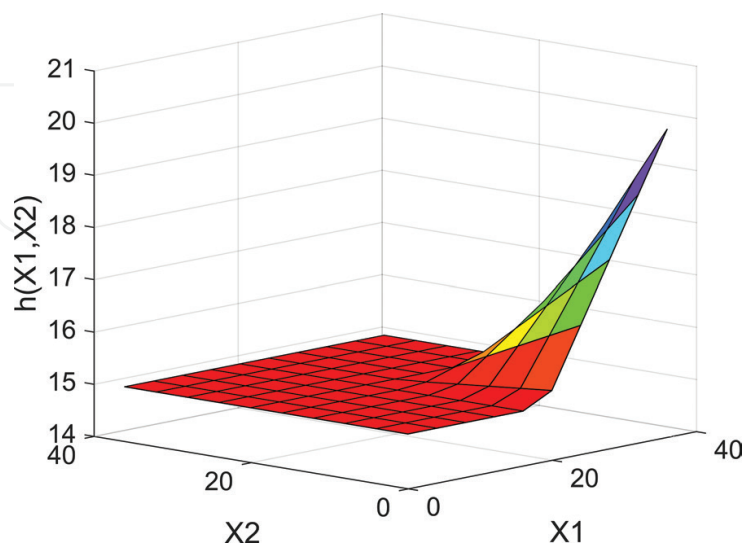


Figure 4. A nonlinear function $h(X_1, X_2)$ which changes only when X_1 is high and X_2 is low. The MARSP model in (4) perfectly matches the original relationship.

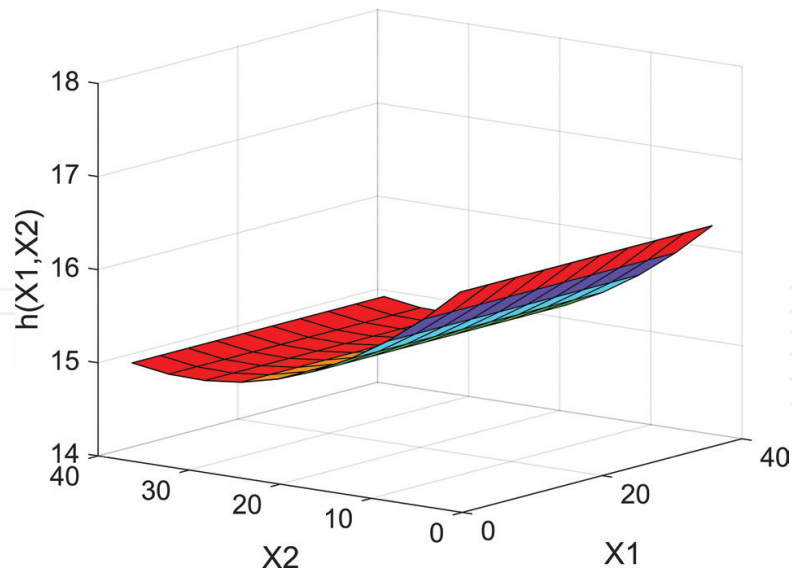


Figure 5. A quadratic model is regressed from $h(X_1, X_2)$ and it has poor accuracy compared to MARSP model.

4. Experimental results

The characterization variables are delay and transition time. Cell delay is the delay from the 50%-point at the cell input to the 50%-point at the cell output. Cell transition time is also called the output slew time, and it is the time between the 20%-point and the 80%-point at cell output (20–80% for rising transition and 80–20% for falling transition). The goal is to find a model that best fits the relationship between the gate delay (output slew) and the explanatory parameters.

In our work, MARSP is implemented using a Matlab toolbox called ARESLab [25]. Some key settings for ARESLab are as follows: the maximum degree of interactions between explanatory parameters is 3; the maximum number of basis functions is 30; the threshold for the stopping criteria is set to 10^{-4} . Please note that all the training data have been normalized.

A commercial library consisting of 247 standard cells was used, and every timing arc for every cell was characterized. The characterization results for some representative cells are shown in **Table 2**. The “ $4*N + 6$ ” column in the table means the number of parameters in the MARSP model, and the “Time(s)” column means characterization time. The “Error” column (“Mean” and “S.D.”) means the average value and standard deviation of the errors between MARSP and golden reference (SPICE), respectively.

The interconnect characterization is similar to the gate although there are only five considered parameters in our work. The details of reduced-order model of interconnect transfer function is not covered in this chapter (Please refer to [8, 9] for more details). The interconnect results are also shown in the last row of **Table 2**. Interconnect variability (spacing, width) is not included in our experiments. It is also worth noting that our methodology can support a higher-order $H'(s)$ -model which matches more moments of the original $H(s)$ at the expense of adding more parameters to the MARSP models.

Cell	(4*N + 6)	Time(s)	Error	
			Mean (%)	S.D. (%)
INVX1	14	861	0.41	2.54
NOR2X1	22	1281	0.23	2.67
CLKBUF3	134	6577	0.17	2.86
XOR2X1	54	3112	0.20	2.32
Interconnect	5	22	0.01	0.12

Table 2. MARSP characterization results on representative standard cells.

4.1. Validation using test paths

Our framework was implemented with C++ and Perl, and the experiments were run on a Linux platform with a 2.27 GHz CPU and 1GB memory without using multi-threading.

Our experiments are based on ISCAS85 benchmark circuits where temperature and supply voltage are considered as global parameters, meaning that all the transistors across the circuit have the same values of temperature and voltage. However, it is worth noting that our methodology can support a temperature profile from a thermal simulator and a voltage profile from an IR-drop simulator. For process variation, as mentioned earlier, we have considered inter-die, intra-die, and intra-gate variations. For channel lengths, we have considered inter-die and intra-die variation, and for threshold voltage, intra-gate variation is considered. This is because channel length is mostly impacted by lithography and etching which exhibit strong spatial correlations, while threshold voltage is strongly affected by random dopant fluctuations. Again, please note that our methodology can work with any inter- and intra-die variation model and with any distributions and any correlation profiles.

We have shown our MARSP models are perfectly accurate individually. Here we construct a framework to integrate our models and then verify its accuracy using test paths. We refer to our framework as GTSSSTA hereafter. Two thousand Monte Carlo samples were run for 10 randomly selected test paths from ISCAS85 benchmark. As shown in the framework above, path delay is calculated for each sample. This obtained delay value is compared to the delay value from hSpice [26], using Eq. (6).

$$Error_{each_sample} = \frac{DELAY_{GTSSSTA} - DELAY_{SPICE}}{DELAY_{SPICE}} \times 100\% \quad (6)$$

A quadratic delay model was also implemented and tested to give a comparison. The quadratic first generates a quadratic regression model as follows:

$$D = d_0 + \sum a_i X_i + \sum b_i X_i^2 + \sum_{i \neq k} b_{i,k} X_i X_k \quad (7)$$

D denotes gate delay, X_i denotes the explanatory parameters, d_0 denotes the constant term, and a_i and b_i denote coefficients of first-order and second-order terms, respectively.

Table 3 presents the results for our framework in comparison to hSpice using these 10 test paths. **Figure 6** gives the histogram comparison of one of the paths between hSpice and GTSSTA. Results in **Table 3** also show that quadratic model has limited accuracy for the 10 test paths.

4.2. Runtime analysis

Experimental results show our framework consumes only ~2% more runtime than quadratic delay model but achieves much better accuracy.

The quadratic delay model in Eq. (7) has a fixed number of operations, that is, 120 multiplications and 66 additions for a one-input gate and 224 multiplications and 120 additions for a two-input gate. The number of operations using MARSP models is not fixed, and it depends on which subspace the data sample falls into. Basically, calculating a MARSP model will have comparisons first and based on the comparison results, different equations (linear, quadratic etc.) are used for calculations. In average, the number of operations for the MARSP model is close to that of the quadratic delay model.

Path	Circuit name	Primary input to primary output	Num. of stages	Num. of samples	Running Time (s)			Path-delay error per sample (OurSSTA)		Path-delay error per sample (quadratic)	
					SPICE	Our	Quad.	Mean (%)	SD (%)	Mean (%)	SD (%)
1	c432	N102 to N421	60	2000	4211	198	192	3.21	1.21	-10.99	24.9
2	c499	N85 to N724	43	2000	2417	140	138	0.25	0.54	-13.95	26.6
3	c880	N1 to N878	57	2000	4148	189	185	-0.31	0.53	-14.10	25.4
4	c1355	G11 to G1352	42	2000	2412	143	144	3.54	1.49	-10.68	26.7
5	c1908	N19 to N2890	72	2000	5878	258	250	4.12	1.92	-10.76	26.9
6	c2670	N227 to N3851	54	2000	4001	184	180	2.90	1.16	-15.81	25.1
7	c3540	N33 to N5360	77	2000	6719	276	270	2.75	1.10	-5.94	27.5
8	c5315	N335 to N8128	71	2000	6454	255	251	2.53	1.55	-11.72	26.3
9	c6288	N290 to N6287	221	2000	19,898	704	698	4.29	1.50	-14.43	23.1
10	c7552	N18 to N11334	116	2000	9948	380	378	3.87	1.48	-10.23	25.8
avg.	—	—	81	2000	6608	273	268	2.72	1.25	-11.86	25.8

Table 3. Experimental results on 10 test paths for MARSP models and quadratic models (errors compared to golden SPICE results).

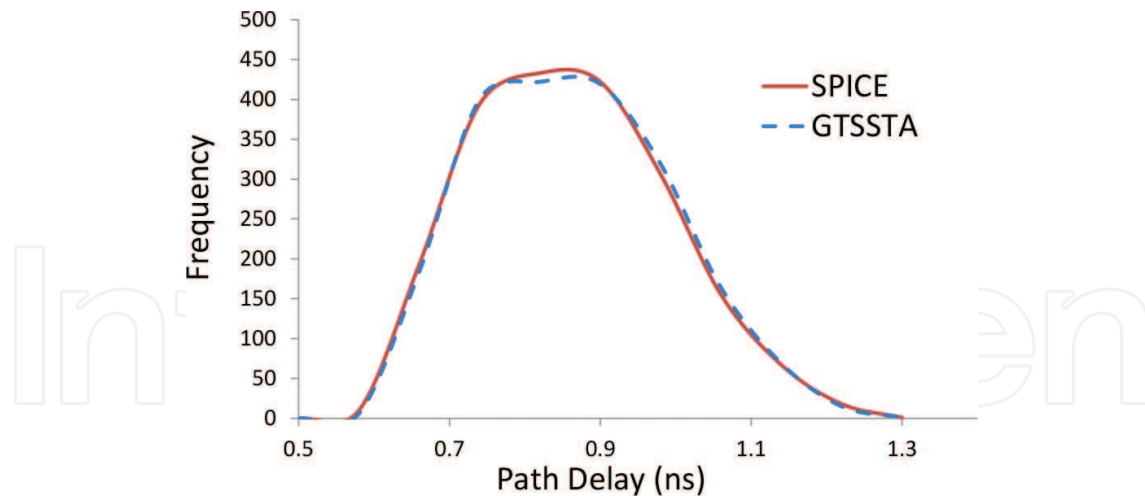


Figure 6. Monte Carlo histogram comparison between GTSSTA and SPICE for test path (N85 to N724) in circuit c499 (2000 samples are run).

5. Conclusion

This chapter talks about the technique called multivariate adaptive regression splines (MARSP). MARSP is a nonparametric regression without taking any pre-assumed form. Instead, it adaptively constructs the model according to the provided data. MARSP has been widely used in high-dimension problems and particularly popular in data mining.

This chapter also gives an application of MARSP in semiconductor field, more specifically, in standard cell characterization. The objective of standard cell characterization is to create a set of high-quality models of a standard cell library that accurately and efficiently model cell behavior. In this work, the MARSP method is employed to characterize the gate delay as a function of many parameters including process-voltage-temperature parameters. Due to its ability of capturing essential nonlinearities and interactions, MARSP method helps to achieve significant accuracy improvement.

Some future work that is worth investigating includes extending the aging-aware MARSP-based timing analyzer to 3D integrated circuits (IC) to study the reliability of 3D ICs which tend to have reliability challenges due to the stronger heat issues. 3D ICs requires more sophisticated thermal models [27–29] and more complicated power-grid analysis [30]. As mentioned earlier, the methodology in this chapter is general to support other thermal and IR-drop models

Author details

Taizhi Liu

Address all correspondence to: taizhiliu88@gatech.edu

Georgia Institute of Technology, Georgia, United States of America

References

- [1] Friedman JH. Multivariate adaptive regression splines. *Annals of Statistics*. 1991;**19**(1):1-67
- [2] Hastie T, Tibshirani R, Friedman J. *The Elements of Statistical Learning*. 2nd ed. Springer; 2009
- [3] Chen C-C, Cha S, Liu T, Milor L. System-level modeling of microprocessor reliability degradation due to BTI and HCI. In: *Proceedings International Reliability Physics Symposium (IRPS)*; 2014. pp. CA 8.1-CA 8.9. DOI: 10.1109/IRPS.2014.6861125
- [4] Chen C-C, Liu T, Milor L. System-level modeling of microprocessor reliability degradation due to bias temperature instability and hot carrier injection. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*. 2016;**24**(8):2712-2725
- [5] Cao Y et al. Cross-layer modeling and simulation of circuit reliability. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. 2014;**33**(1):8-23
- [6] Fang J, Sapatnekar SS. Incorporating hot carrier injection effects into timing analysis for large circuits. *IEEE Transactions on Very Large Scale Integration Systems (TVLSI)*. 2014;**22**(12):2738-2751
- [7] Liu T, Chen C-C, Kim W, Milor L. Comprehensive reliability and aging analysis on SRAMs within microprocessor systems. *Microelectronics Reliability*. 2015;**55**(9-10):1290-1296
- [8] Liu T, Chen C-C, Milor L. Accurate standard cell characterization and statistical timing analysis using multivariate adaptive regression splines. In: *Proceeding of International Symposium on Quality Electronic Design (ISQED)*; 2015. pp. 272-279
- [9] Liu T, Chen CC, Milor L. Comprehensive reliability-aware statistical timing analysis using a unified gate-delay model for microprocessors. *IEEE Transactions on Emerging Topics in Computing*. DOI: 10.1109/TETC.2016.2588724
- [10] Liu T, Chen C-C, Wu J, Milor L. SRAM stability analysis for different cache configurations due to Bias Temperature Instability and Hot Carrier Injection. In: *Proceedings of IEEE 34th International Conference on Computer Design (ICCD)*; 2016. pp. 225-232
- [11] Cha S, Chen C-C, Liu T, Milor L. Extraction of threshold voltage degradation modeling due to negative bias temperature instability in circuits with I/O measurements. In: *Proceedings of VLSI Test Symposium (VTS)*; 2014. pp. 1-6. DOI: 10.1109/VTS.2014.6818769
- [12] Jiang C, Habetler T. Static eccentricity fault detection of the BLDC motor inside the air handler unit (AHU). In: *2015 IEEE International Electric Machines & Drives Conference (IEMDC)*; 2015. pp. 1473-1476
- [13] Cai W, Li C, Luan S. SOI RF switch for wireless medical sensor network. *Advances in Engineering: An International Journal (ADEIJ)*. 2016;**1**(2):1-9
- [14] Cai W, Huang L, Wen WJ. 2.4 GHZ class AB power amplifier for wireless medical sensor network. *International Journal of Enhanced Research In Science Technology & Engineering (IJERSTE)*. 2016;**5**:94-98

- [15] Cai W, Xu J, Huang L. Low power SI class E power amplifier and RF switch for health-care. *An International Journal Informatics Engineering*. 2016;4(4):7-13
- [16] Cai W, Shi F. Design of low Power Medical Device. *International Journal of VLSI design & Communication Systems (VLSICS)*. 2017;8(2):1-7
- [17] Cai W, Shi F. Design of low power Si based DPA for medical device. *International Journal of Electronics Communication and Computer Engineering*. 2017;8(2):134-136
- [18] Zhang S et al. Syslog processing for switch failure diagnosis and prediction in datacenter networks. In: 2017 IEEE/ACM 25th international symposium on quality of service (IWQoS); 2017. DOI: 10.1109/IWQoS.2017.7969130
- [19] Chen C-C, Liu T, Cha S, Milor L. Processor-level reliability simulator for time-dependent gate dielectric breakdown. *Microprocessors and Microsystems*. 2015;39(8):950-960
- [20] Yang K, Liu T, Zhang R, Kim D-H, Milor L. Front-end of line and middle-of-line time-dependent dielectric breakdown reliability simulator for logic circuits. *Microelectronics Reliability*. 2017;76:81-86
- [21] Liu T, Chen C-C, Cha S, Milor L. System-level variation-aware aging simulator using a unified novel gate-delay model for bias temperature instability, hot carrier injection, and gate oxide breakdown. *Microelectronics Reliability*. 2015;55(9-10):1334-1340
- [22] Cha S, Kim D-H, Liu T, Milor L. The die-to-die calibrated combined model of negative bias temperature instability and gate oxide breakdown from device to system. *Microelectronics Reliability*. 2015;55(9-10):1404-1411
- [23] Zhang W, Singhee A, Xiong J, Habitz P, Joshi A, Visweswariah C, Sundquist J. A dynamic method for efficient random mismatch characterization of standard cells. In: *Proceedings ICCAD*; 2012. pp. 180-186
- [24] Sundareswaran S, Abraham J, Panda R, Ardelea A. Characterization of standard cells for intra-cell mismatch variations. *IEEE Transactions on Semiconductor Manufacturing*. 2009;22(1):40-49
- [25] Jekabsons G. ARESLab: Adaptive Regression Splines toolbox for Matlab/Octave [Internet]. [Updated: 2011]. Available from: <http://www.cs.rtu.lv/jekabsons/>
- [26] Synopsys. HSPICE User Guide [Internet]. Available from: <https://www.synopsys.com/verification/ams-verification/circuit-simulation/hspice.html> [Accessed: 2017]
- [27] Zhang Y, Sarvey TE, Bakir MS. Thermal challenges for heterogeneous 3D ICs and opportunities for air gap thermal isolation. In: *Proceedings IEEE 3D Systems Integration Conference (3DIC)*; 2014. DOI: 10.1109/3DIC.2014.7152174
- [28] Zhang Y, Sarvey TE, Bakir MS. Thermal evaluation of 2.5-D integration using bridge-chip technology: Challenges and opportunities. *IEEE Transactions on Components, Packaging and Manufacturing Technology*. 2017;7(7):1101-1110. DOI: 10.1109/TCPMT.2017.2710042

- [29] Zhang Y, Bakir MS. Integrated thermal and power delivery network co-simulation framework for single-die and multi-die assemblies. *IEEE Transactions on Components, Packaging and Manufacturing Technology*. 2017;7(3):434-443
- [30] Zhang F, Wu B, Zhai J, He B, Chen W. Finepar: Irregularity-aware fine-grained workload partitioning on integrated architectures. In: *IEEE/ACM Intl. Symp. on Code Generation and Optimization (CGO)*; 2017. pp. 27-38

IntechOpen

IntechOpen