RESEARCH ARTICLE OPEN ACCESS

Analysis of Nanometer CMOS ICS on Propagation Delay

K.Supriya[1], Dr.V.Thrimurthulu[2], S.Ali Asgar[3]
[1]II M.Tech VLSI SD Student, CR Engineering College, Tirupathi, Chittoor(Dist) A.P, India,
[2]Professor, Head of ECE Dept., CR Engineering College, Tirupathi, Chittoor (Dist) A.P, India,
[3]Assistant Professor of ECE Dept., CR Engineering College, Tirupathi,Chittoor(Dist) A.P, India,
[1]supri.kasala@gmail.com, [2]vtmurthy.v@gmail.com , [3]aliasgarsyed@gmail.com

Abstract—To analyze the dependence of complex gates delay with the sensitization vector and its variation that gets up to 40% in 65-nm CMOS technologies and include its effect in the path delay estimation that can be in the order of 16%. The gate delay is compute from a simple polynomial analytical description that requires a one time library parameter, making it highly scalable. An STA tool based on a single-pass true path computation is used to determine the critical path list. Since it does not rely on a two-step process, it can be programmed to find efficiently the N true paths from a circuit. Results from various benchmark circuits synthesized for three commercial technologies (130, 90, and 65 nm) provide better results in number of paths reported and delay estimation for these paths compared to a commercial tool. The impact of delay variation with the sensitization vector for paths with complex gates reveals as a significant mechanism that must be considered as it is comparable to the impact of parameter variations or interconnect-induced delay.

Key words—Delay model, timing analysis.

I. NTRODUCTION

TIMING analysis is a key step in the VLSI design flowwhose significance and complexity increases with technology scaling due to new physical phenomena appearing innanometer technologies [1], [2]. The yield of the manufacturing process can increase considerably using a highly accurate timing analysis tool capable of correctly finding true critical paths, and identifying those gates having higher sensibility to process variations and environmental conditions [3], [4].When a circuit design is synthesized using standard cells, computer-aided design (CAD) algorithms are designed to reduce circuit area, power consumption, and propagationdelays in addition to optimizing other parameters. To accomplish this goal, synthesis tools use library complex gates, i.e.,circuit structures that combine primitive logic functions, such as NOT , AND , OR , NAND , NOR , in a single CMOS structure that reduces the number of transistors required to perform a given logic function. Typically, complex gates comprisea combination of few primitive functions (as detailed inSection II) although more complex functions like full-adders or multiplexers are also common. the actual circuit structure being finally manufactured [7].

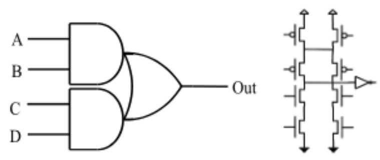


Fig. 1. Gate AO22.

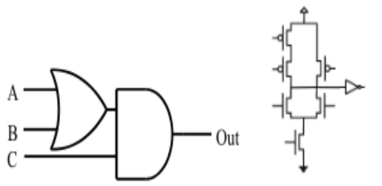


Fig. 2. Gate OA12.

TABLE I
AO22 PROPAGATION TABLE

	A	B	C	D	Out
Vector A1	T	1	0	0	T
Vector A2	T	1	1	0	T
Vector A3	T	1	0	1	T
Vector B1	1	T	0	0	T
Vector B2	1	T	1	0	T
Vector B3	1	T	0	1	T
Vector C1	0	0	T	1	T
Vector C2	1	0	T	1	T
Vector C3	0	1	T	1	T
Vector D1	0	0	1	T	T
Vector D2	1	0	1	T	T
Vector D3	0	1	1	T	T

TABLE II
OA12 PROPAGATION TABLE

	A	B	C	Out
Vector A1	T	0	1	T
Vector B1	0	T	1	T
Vector C1	1	0	T	T
Vector C2	0	1	T	T
Vector C3	1	1	T	T

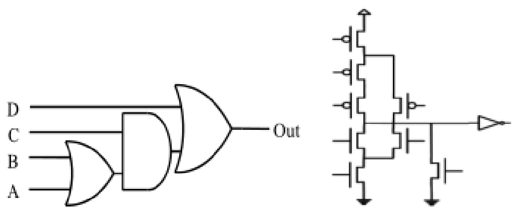


Fig. 3. Gate CB4I6.

II. COMPLEX GATES DELAY VARIATION

In general, all complex logic cells have more than one inputvector that sensitizes a transition propagation from each input toward the output. The sensitization vectors for each input areeasily computed from the gate logic function. In some cases,for some gate inputs, only one input vector allows propagatinga transition through such an input, but in most cases morethan one sensitization vector is found.In this paper, we onlyconsider the cases with steadyvalues in all inputs exceptA. Gate-Level AnalysisWithout loss of generality, we illustrate the delay dependence with the sensitization vector using four complex gates included in almost all standard cell libraries. The first gate isthe AO22 being a four input gate that implements the logic function in (1), whose

$$Out = A * B + C * D \tag{1}$$

$$Out = (A + B) * C \tag{2}$$

logic symbol and transistor topology are shown in Fig. 1.Table I shows the sensitization vectors for each gate input.

The logic value “T,” represents a transition either rising or falling. The second complex gate considered is the being a three input gate forwhich only one of its inputs has multiple input vectors to sensitize the gate. The gate logic function is given by (2),its symbol and transistor topology shown in Fig. 2, and the sensitization vectors in Table II.The other two gates are the CB4I6 and the AOI212 their logic functions are given in (3) and (4), respectively, their symbols and transistor topologies are shown in Figs. 3 and 4, whilethe sensitization vectors are shown in Tables III and IV,respectively.

As shown in Tables I–IV, the number of sensitization vectorsfor each input may vary significantly depending on the gateconsidered. For some cases, only one input vector sensitizesthe gate (e.g., inputs A and B of OA12), while in other casesthere is a considerable number of sensitization vectors (gateAOI212 has nine sensitization vectors for input E)

TABLE III
CB4I6 PROPAGATION TABLE

	A	B	C	D	Out
Vector A1	T	0	1	0	T
Vector B1	0	T	1	0	T
Vector C1	1	0	T	0	T
Vector C2	0	1	T	0	T
Vector C3	1	1	T	0	T
Vector D1	0	0	0	T	T
Vector D2	0	1	0	T	T
Vector D3	1	0	0	T	T
Vector D4	1	1	0	T	T
Vector D5	0	0	1	T	T

TABLE IV
AOI212 PROPAGATION TABLE

	A	B	C	D	E	Out
Vector A1	T	1	0	0	0	$\overline{T}$
Vector A2	T	1	0	1	0	$\overline{T}$
Vector A3	T	1	1	0	0	$\overline{T}$
Vector B1	1	T	0	0	0	$\overline{T}$
Vector B2	1	T	0	1	0	$\overline{T}$
Vector B3	1	T	1	0	0	$\overline{T}$
Vector C1	0	0	T	1	0	$\overline{T}$
Vector C2	0	1	T	1	0	$\overline{T}$
Vector C3	1	0	T	1	0	$\overline{T}$
Vector D1	0	0	1	T	0	$\overline{T}$
Vector D2	0	1	1	T	0	$\overline{T}$
Vector D3	1	0	1	T	0	$\overline{T}$
Vector E1	0	0	0	0	T	$\overline{T}$
Vector E2	0	0	0	1	T	$\overline{T}$
Vector E3	0	0	1	0	T	$\overline{T}$
Vector E4	0	1	0	0	T	$\overline{T}$
Vector E5	0	1	0	1	T	$\overline{T}$
Vector E6	0	1	1	0	T	$\overline{T}$
Vector E7	1	0	0	0	T	$\overline{T}$
Vector E8	1	0	0	1	T	$\overline{T}$
Vector E9	1	0	1	0	T	$\overline{T}$

$$Out = D + C * (B + A) \tag{3}$$

$$Out = E + D * C + B + A. \tag{4}$$

We carried out extensive electrical simulations tocompute the gate delays through each input for all thesensitization vectors for three commercial CMOS technologies(130, 90, and 65 nm) at nominal supply voltage and 25 °C.

Results in Table V show propagation delay variations withthe input sensitization vector that reaches up to 50% (49.85%)depending on the gate structure, input transition, and technology. The delay variation for the 65-nm technology may getto up to 43% (Vector E5 versus Vector E1 for gate AOI212propagating a falling input transition) suggesting that thisvariation may induce a large variance at the circuit level.

B. Circuit-Level Relevancy

As an initial experiment to analyze the impact of themultiple vector sensitization at the circuit level, we tookthe 1000 slowest paths of the ISCAS85 benchmark circuitsand computed how many of such paths contained multiplesensitization vectors. The benchmark circuits were synthesizedusing a commercial tool on a 65-nm CMOS commercialtechnology. Results are given in Table VI showing that, forthe large circuits (starting from the ISCAS c499) in almost all cases, the first 1000 slowest paths contain multiple-input gateshighlighting the relevancy that this phenomenon might have atthe circuit level.

III. TRANSISTOR LEVEL ANALYSIS

We investigated the root cause of the delay variations with the sensitization vector to get insight into this phenomenonthrough a transistor-level analysis. This analysis is carriedout on the two first gates considered since it was observedthat the delay variation root cause is common to all gates.The considered complex gates implement noninverting functions, and require an output inverter for a CMOS implementation. Such inverter does not influence the delay variationwith the sensitization vector and therefore it is not considered in the transistor-level analysis. Fig. 5 shows the transistor-level analysis for gate AO22 and represents the three inputvectors that propagate a falling transition through Input Results in Table V show that the transition in Fig. 5(a)corresponds to the fastest case, while Fig. 5(b) corresponds tothe slowest one. As shown in Fig. 5, the current charging theoutput node must pass always through transistor PA. In thefastest case, both parallel transistors PC and PD are ON ,allowing a higher current through PA, leading to a quickercharge of the output node. In the other two cases, only oneof the two top parallel transistors (either PC or PD) is ON.resulting in less current available to charge the output andhence resulting in a bigger delay. The relative delay differencebetween.Fig. 7 shows the transistor-level diagram for each sensitizationvectors that pass a rising transition at input C toward the gateoutput. Fig. 7(c) corresponds to the fastest transition. For thisinput vector, transistors NA and NB are both ON , increasingthe current available through NC with respect to the other twocases where only NA or NB is ON . The Vector C2 transition[Fig. 7(b)] shows a delay slightly larger than that for Vector C

BARCELÓ *et al.*: SENSITIZATION INPUT VECTOR IMPACT ON PROPAGATION DEL

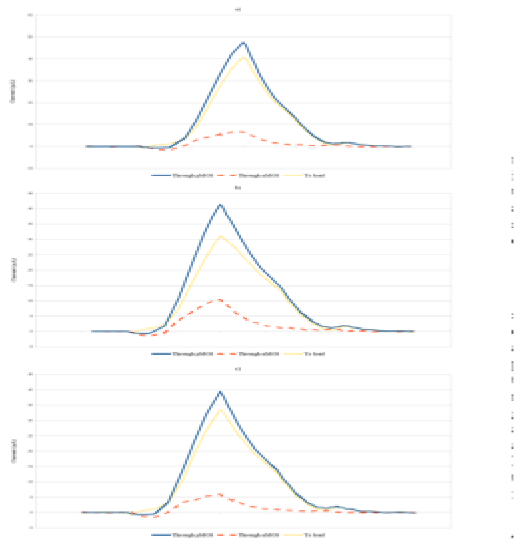


Fig. 6. Internal currents of AO22. (a) Vector A1. (b) Vector A2. (c) Vector A3.

TABLE VII
MEMORY USAGE BY THE MODEL

Max. Polynomial Order	LUT-Based Memory / Polynomial-Based Memory
1	28
2	15
3	7

.

IV. DELAY MODEL , HEURISTIC , AND TOOL

We developed a timing analysis tool that combines aspecific delay model and algorithm to find true paths in acombinational circuit. The delay model is analytical througha polynomial expression similar to SPDM . Such apolynomial model is used to estimate both the gate propagation delay and the output transition time, since the latter isrequired to compute the propagation delay of the followinggates within the path. The second component of the timinganalysis tool is the algorithm developed to find true paths ina combinational circuit. Such an algorithm is based on theRESIST algorithm and has been specifically developedto consider the dependence of the delay with the input vectorfor complex gates.

A. Delay Model

The delay model includes multiple variables, such as inputtransition time, output load, temperature, and supply voltage,and can be easily extended to accommodate additional variables. The analytical nature of the model provides some advantages over widely used lookup table (LUT)-based approaches.
Equation (5) shows the general form of the analytical modelused to compute propagation delay and output transition timeof each gate. C Load is the capacitive

load of the output node in the input transition time, T is the temperature, and V DDthe supply voltage. These parameters are independent for eachgate. In this way, the model can be applied to circuits havingvarious V DD regions and can be combined with other toolsthat compute the temperature and/or V DD values at differentcircuit regions.The model coefficients, represented by P ijkl in (5), areobtained from electrical simulations of the cell

$$f\left(C_{\text{Load}}, t_{\text{in}}, T, V_{\text{DD}}\right) = \sum_{i=0}^{m}\sum_{j=0}^{n}\sum_{k=0}^{o}\sum_{l=0}^{p} P_{\text{ijkl}} \cdot C_{\text{Load}}^{i} \cdot t_{\text{in}}^{j} \cdot T^{k} \cdot V_{\text{DD}}^{l} \,. \qquad (5)$$

The electrical simulations from which the model parameters are extracted, are carried automatically and systematically fora given technology library, and consist of a set of iterative simulations. Considered.These simulation data matrices are used to extract the parameters (P ijkl) of the polynomial model (5), using an iterativeprocess that repeats the same step as many times as variablesconsidered polynomialregression relative to one of the variables for each matrixobtained in the previous step is carried out. Expression (6)shows the general form of each step, where a polynomialregression of order m with respect to variable x 1 is appliedto an n-dimensional matrix f and the result is m +1 (n −1)

dimensional matrices f_i

$$\underbrace{f(x_1, x_2, \ldots, x_n)}_{\substack{\downarrow \\ 1n-\text{dimMatrix}}} \rightarrow \underbrace{\sum_{i=0}^{m} f_i(x_2, \ldots, x_n) \cdot x_1^{i}}_{\substack{\downarrow \\ m+1(n-1)\text{-dimMatrices}}} . \qquad (6)$$

$$\text{Relative Range}\,(X) = \frac{\max(X) - \min(X)}{|\overline{X}|} \equiv RR\,(X) \qquad (7)$$

$$RR\,(f(x)) < \text{Min Relative Range} \rightarrow f(x) = \overline{f(x)}. \qquad (8)$$

Step 1: Perform a polynomial regression for the first variable.
as shown in (9)

$$f(w, x, y, z) = \sum_{i=0}^{m} P_i(x, y, z) \cdot w^{i}. \qquad (9)$$

Step 2: Each P i obtained in the previous step is adjusted bya polynomial function for the first variable of which depends applying the same process used in Step 1.The result is a set of (m + 1) • (n + 1) parameters P ij thatdepend on the rem
Step 3: Polynomial regression of each parameter P ij withrespect to variable y, resulting in a set of (m + 1)•(n + 1)•
(o + 1) parameters as shown in (11)
Step 4: Finally, each parameter P ijk from Step 3 is adjustedby a polynomial function, resulting is the set of parametersP ijkl as shown in (12)

The orders (m, n, o, p) of each polynomial regressiondepend on the requirements imposed. These requirements areas follows:
1) maximum order of the polynomial regression;
2) minimum correlation coefficient;
3) maximum relative error;
4) minimum relative range.

The algorithm first computes the relative range of the datato be adjusted, if it is below the MRR, then such data areapproximated by its mean value, as If RR is larger than MRR, then the algorithm starts an iterative process of polynomial adjustment until the resultmeets the requirements.

Each process step is a polynomial regression of order n,starting with order 1 (i.e., a lineal regression), and thencomputes both the correlation coefficient and the maximumrelative error between the input data and the regression results.If these two values comply with the requirements imposed,then the process finishes; if not, the regression is repeatedincreasing the polynomial order by 1, and so on until the errorrequirements are fulfilled, or the allowed maximum order is reached.For each input matrix, the process generates m outputmatrices with a dimension lowered by 1 with respect indicated in (8). to the input one. The polynomial order m, is independent for eachinput matrix.
2) Then, the sensitization algorithm isapplied for the current node and current fan-out option.
3) If the sensitization algorithm returns true, then the sensitized gate output becomes the new current node. If thenode is not an output node, then the process continuesusing this node. If the node is an output, the path andits logic vectors are saved.
4) Once the path is saved, or the sensitization algorithmreturns false In this way, if a logic incompatibilityis found, the path tracing process stops, and all the pathssharing the current sub-path from the input to the currentnode are discarded. Now the algorithm jumps to the lastsaved point.

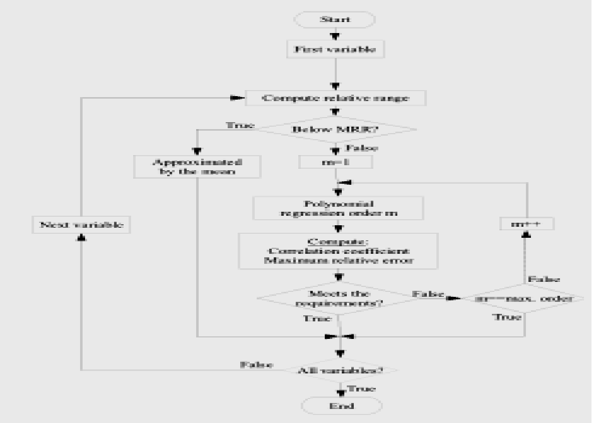


Fig. 8. Flowchart of the model parameter extraction algorithm.

5) If the stack is empty, all paths starting at this input havebeen explored, and the algorithm ends.
6) If the stack is not empty, the last state stored is restoredand the algorithm proceeds with the next stem.The sensitization algorithm is the innermost process, and isapplied to each gate that is crossed from an input node to anoutput node. The following paragraphs describe this algorithmillustrated in Fig 11.
1) The first step identifies which sensitization vectorsof the current gate are compatible with the currentlogic state.
2) If none of the options is compatible, then the processends and returns false.
3) Once the compatible vectors are identified, the logicvector representing the circuit state is cloned to have asmany vectors as the number of compatible sensitizationoptions.

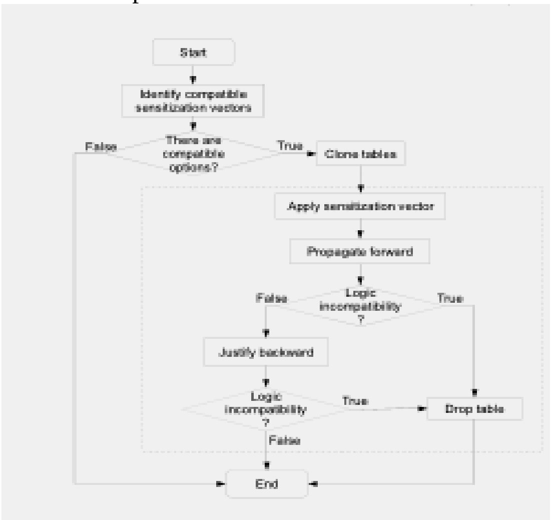


Fig. 11. Sensitization algorithm.

4) At this point process that can be run in parallel due to theindependent nature of each iteration in starts.
5) After cloning the logic state, the first step applies thesensitization vector to allow the gate to propagate atransition from an input to the output.
6) Next step is the forward propagation of assigned values.Each time that a logic value is assigned to a node, itis propagated through all the gates having such nodeas an input. This procedure, that does not imply anydecision, helps in early detection of logic inconsistenciesand improves the algorithm performance because it isless complex than a justification process.
7) If the logic propagation produces an incompatibility, thiscase is abandoned and its logic vector is dropped.

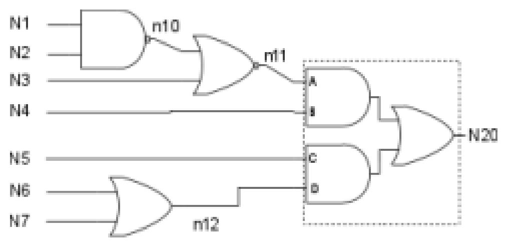


Fig. 12. Test circuit.

8) After the forward propagation, the next task is to justifythe values assigned to internal nodes, verifying if thesevalues can be assigned from the circuit inputs. Asmentioned before, this process is more complex, andtime consuming than the forward propagation becauseit implies taking decisions. Depending on the valueassigned, the type of gate and the logic values assignedpreviously to other nodes, each gate output can bejustified with more than one option.
9) Similarly to the forward propagation process, if it isnot possible to justify all nodes without logical incompatibilities this sensitization option is discarded. If thejustification is successful, then the sensitization processends and the resulting logical state is ready to sensitizethe next gate.

V. R ESULTS

We show the impact of the sensitization vector on the pathdelay estimation by reporting the model and tool results forvarious circuits and technologies. Without loss of generality,we consider a single supply voltage and a uniform 25 °Ctemperature, although these parameters can be modified asexplained before. As done in many works in this domain we compare the results obtained with a referencecommercial tool being Primetime from Synopsys .A. Test CircuitWe first report initial results on a simple circuit shownin Fig. 12 to illustrate how the developed algorithm works

TABLE VIII
DELAY VERSUS INPUT VECTOR FOR THE SIMPLE CIRCUIT IN FIG. 12

Input vector	Delay (ps)
N1=F, N2=1, N3=0, N4=1, N5=1, N6=0, N7=0	106.16
N1=F, N2=1, N3=0, N4=1, N5=0, N6=X, N7=X	97.73

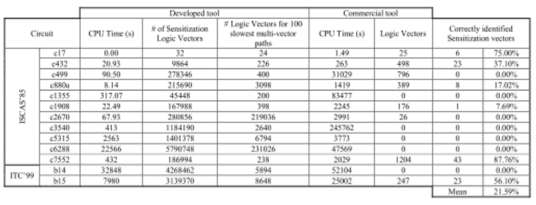
compared to a commercial tool. The easiest way to sensitize thecomplex gate leads to the smaller propagation delay for thispath, although it is also possible to sensitize the gate with aninput vector that exhibits a larger delay. The commercial toolcorrectly provides the critical path that propagates a fallingedge through nodes N1-n10-n11-N20, as expected. The inputvector used to sensitize the critical path is
N1 = F N2 = 1 N3 = 0 N4 = 1 N5 = 0 N6 = X N7 = X

and corresponds to the easiest option that assigns a logic0 to node N5 and therefore does not require assigning n12NOR justifying its value to an input node. Setting N5 to 0provides the shortest way to sensitize the AO22 gate, butignores another case having a larger propagation delay forsuch path. This can be obtained sensitizing gate AO22 with avector that results in a larger delay. This second vector requiresa more complex justification process to assign logic valuesuntil reaching an input node.

The tool developed provides two paths passing through thesame nodes and starting with a falling transition, each withdifferent input vector. One is the same vector provided by thecommercial tool, while the second one is N1 = F N2 = 1 N3 = 0 N4 = 1 N5 = 1 N6 = 0 N7 = 0.

Table VIII provides the delay obtained from electricalsimulations of the critical path for the two input vectors. It isshown that the additional path provided by the tool developedexhibits a delay increase of 8.6% with respect to the one givenby the commercial tool

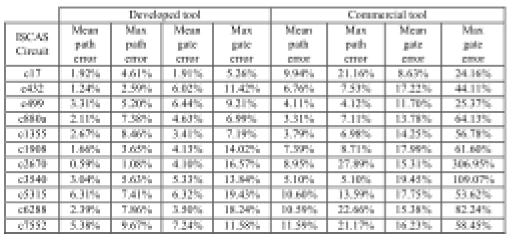
TABLE IX
TECHNOLOGY INDEPENDENT WORST SENSITIZATION VECTOR IDENTIFICATION RESULTS



Circuit		Developed tool: CPU Time (s)	# of Sensitization Logic Vectors	# Logic Vectors for 100 slowest multi-vector paths	Commercial tool: CPU Time (s)	Logic Vectors	Correctly identified Sensitization vectors	
ISCAS'85	c17	0.00	32	24	1.49	25	6	75.00%
	c432	20.93	9864	226	263	498	23	37.10%
	c499	90.50	278346	400	31029	796	0	0.00%
	c880a	8.14	215690	3098	1419	389	8	17.02%
	c1355	317.07	45448	200	83477	0	0	0.00%
	c1908	22.49	167988	398	2245	176	1	7.69%
	c2670	67.93	280856	219036	2991	26	0	0.00%
	c3540	413	1184190	2640	245762	0	0	0.00%
	c5315	2563	1401378	6794	3773	0	0	0.00%
	c6288	22566	5790748	231026	47569	0	0	0.00%
	c7552	432	186994	238	2029	1204	43	87.76%
ITC'99	b14	32848	4268462	5894	52104	0	0	0.00%
	b15	7980	3139370	8648	25002	247	23	56.10%
							Mean	21.59%

To generate the results, we first determined the paths havingmore than one sensitization vector. The third column shows the total number of sensitization logic vectors reported by the tool, and the fourthcolumn shows the number of input vectors reported by thedeveloped tool that sensitizes the functional paths considered,sequence of nodes and transitions on each node, but with

TABLE X
130-nm DELAY COMPARISON VERSUS ELECTRICAL SIMULATION



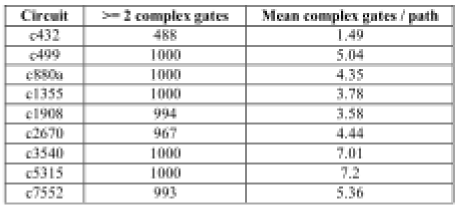
ISCAS Circuit	Developed tool: Mean path error	Max path error	Mean gate error	Max gate error	Commercial tool: Mean path error	Max path error	Mean gate error	Max gate error
c17	1.92%	4.61%	1.91%	5.26%	9.94%	21.16%	8.63%	24.16%
c432	1.24%	2.59%	6.02%	11.42%	6.76%	7.53%	17.22%	44.11%
c499	3.31%	5.20%	6.44%	9.21%	4.11%	4.12%	11.70%	25.37%
c880a	2.11%	7.58%	4.63%	6.98%	3.31%	7.11%	13.78%	64.13%
c1355	2.67%	8.46%	3.41%	7.19%	3.79%	6.98%	14.25%	56.78%
c1908	1.66%	3.65%	4.13%	14.02%	7.59%	8.71%	17.99%	61.60%
c2670	0.59%	1.08%	4.10%	16.57%	8.95%	27.89%	15.31%	306.95%
c3540	3.04%	5.63%	5.33%	13.84%	5.10%	5.10%	19.45%	109.07%
c5315	6.31%	7.41%	6.32%	19.45%	10.60%	13.59%	17.75%	53.62%
c6288	2.39%	7.86%	3.58%	18.24%	10.59%	22.66%	15.38%	82.24%
c7552	5.38%	9.67%	7.24%	11.58%	11.59%	21.17%	16.23%	58.45%

Different sensitization vector and propagation, and foreach functi onal path considered, the sensitization vectors arecompared.Finally, the last two columns of Table IX show the numberand percentage, respectively, of functional paths for which
the minimum effort algorithm provides the input vector thatproduces the worst delay for that functional path. Theseresults show the inefficiency of not considering the specificsensitization vector during the delay computation highlightingthe impact of the delay variation due to the sensitization vectorfor complex gates. In many cases, the commercial tool simplyfinds the case for which the complex gate input assignations are easier to justify instead of exploring all the possibilities.Results show that the delay modelused to estimate the gate propagation delay provides moreaccurate results than the commercial tool considered.In all the cases investigated the polynomial model providesbetter delay estimations than the LUT model used by thecommercial tool, even considering a first-order model..

VI. RELEVANCE AND COMPARISON TO OTHER E FFECTS

We compared the delay variation due to the sensitizationvector to the delay variations caused by other effects likeprocess parameter fluctuations or the interconnect system.Such analysis is a key to determine the relative significance ofthis phenomenon compared to other important delay variation sources. We carried this comparison for various combinationalISCAS circuits to estimate the relative impact at the circuitlevel. Table XIII shows the relative delay variations obtainedfor the c432 ISCAS circuit as an example. The first row shows
the delay variation due to the sensitization vector that gets upto 30%.To estimate the delay due to the interconnect system, wecompared the nominal delay of the ISCAS c432 using atiming simulator that neglected the impact of the interconnect

TABLE XIV
COMPLEX GATE PER PATH



Circuit	>= 2 complex gates	Mean complex gates / path
c432	488	1.49
c499	1000	5.04
c880a	1000	4.35
c1355	1000	3.78
c1908	994	3.58
c2670	967	4.44
c3540	1000	7.01
c5315	1000	7.2
c7552	993	5.36

oad to another simulation of the same circuit for whichinterconnect was estimated assuming a 10× area increase.Such analysis provides an estimation about the impact of theinterconnect delay for circuits having long wires. The secondrow in Table XIII shows the relative delay variation betweenboth circuit versions whose difference is mainly due to theinterconnect system. Such delay variation is 10% smaller thanthe delay variation due to the sensitization vector variation..

VII. C ONCLUSION

We have shown the importance of considering the inputvector used to sensitize a complex gate in the delay estimationreporting delay variations up to 43% for a 65-nm technologyat the gate level. A detailed transistor-level revealed that these variations are due to enabling or disabling parallel currentpaths as well as to parasitic contributions to/from internalcapacitances. Our experiments showed that this may have a significant impact at the circuit level resulting in delayvariations from one sensitization vector to another in the order of 16% for a 65-nm commercial CMOS technology.This methodallowed us to account for all sensitization vectors in eachcomplex gate and compute the gate delay accurately. Resultsfrom various benchmark circuits showed that the delay modelconsidered provides a quite good estimation, and demonstratedthe ability of the algorithm developed to find all input vectorsfor a given path, identifying correctly the worst input vector foreach path. Such a feature was not supported in the commercialtool that does not account for multiple sensitization vectors incomplex gates and assigned the vector whose justification issimpler. Results for all the circuits considered showed that thetool developed provides better results than the commercial toolas it reports more paths with a more accurate delay requiringless computation time.It was also shown that the impact ofsensitization vectoron the delay is comparable to the delay variation caused by other effects, such as parameter variations, interconnect delay,or temperature.

ACKNOWLEDEMENT

I express my sincere thanks to my guide Mr.S.ALI ASGAR and Project Coordinator Mr.S.ALI ASGAR, M.Tech, Assistant Professor of ECE Dept, and to my HEAD OF DEPARTMENT Dr. V. Thrimurthulu M.E., Ph.D., MIETE., MISTE. Professor & Head of ECE Dept. CREC, TIRUPATHI, for their valuable guidance and useful suggestions, which helped me in the project work

R EFERENCES

[1] I . Keller, K. H. Tam, and V. Kariat, "*Challenges in gate level modelingfor delay and SI at 65 nm and below*," in Proc. 45th ACM/IEEE DAC,Jun. 2008, pp. 468–473.
[2] S. R. Nassif, "*Modeling and forecasting of manufacturing variations*,"in Proc. ASP-DAC, 2001, pp. 145–149.
[3] D. Blaauw, K. Chopra, A. Srivastava, and L. Scheffer, "*Statisticaltiming analysis: From basic principles to state of the art*," IEEE Trans.Comput.-Aided Design Integr. Circuits Syst., vol. 27, no. 4, pp. 589–607, Apr. 2008.
[4] K. S. Kim, S. Mitra, and P. G. Ryan, "*Delay defect characteristics andtesting strategies*," IEEE Design Test Comput., vol. 20, no. 5, pp. 8–16,Sep./Oct. 2003.
[5] T. El Motassadeq, V. Sarathi, S. Thameem, and M. Nijam, "*SPICE versus STA tools: Challenges and tips for better correlation*," in Proc.IEEE SOCC, Sep. 2009, pp. 325–328.
[6] H. Yaun-Chung, C. Hsi-Chuan, S. Shangzhi, and D. H. C. Du, "*Timing analysis of combinational circuits containing complex gates*," inProc. Int. Conf. Comput. Design, VLSI Comput. Process., Oct. 1998, pp. 407–412.

AUTHORS:

K.Supriya received her B.Tech degree in Electronics & Communication Engineering from Priyadarshini Institute Of Technology, Tirupati (A.P), India, in the year 2012.Currently pursuing her M.Tech degree in VLSI System Design at Chadalawada Ramanamma Engineering College, Tirupati (A.P), India. Her area of research includes low power VLSI design.

Dr.V. Thrimurthulu M.E.,Ph.D., MIETE., MISTE Professor & Head of ECE Dept. He received his Graduation in Electronics & Communication Engineering AMIETE in 1994 from Institute of Electronics & Telecommunication Engineering, New Delhi, Post Graduation in Engineering M.E specialization in Microwaves and Radar Engineering in the year Feb, 2003, from University College of Engineering, Osmania University, Hyderabad, and his Doctorate in Philosophy Ph.D. from Central University, in the year 2012. He has done his research work on Ad-Hoc Networks.

S.Ali Asgar was born in Tirupati,Andhra Pradesh, India. He received B.Tech degree in electronics & Instrumentation engineering from Sree Vidyanikethan engineeringcollege.A.Rangampeta in the year 2007. He did his M. Tech. in Digital Signals in 2012 from SJCET,yemmiganur(A.P), India. He is currently working as Associate Professor in Chadalawada ramanamma Engineering College, Tirupati(A.P), and India.