Chapter 3

Statistical Model for Logic Errors in CMOS Digital Circuits

3.1 Introduction

In the past, noise was not such a big issue in digital integrated circuits. However, the continuous progress in semiconductor technology put the noise issue among the major concerns of digital CMOS IC designers. In modern and future LSI design, feature sizes shrinking to nanometer scale, clock frequencies reaching the multi-GHz level, and supply voltages declining to the sub-voltage range [1]. Consequently, the effects of various noise sources, explained in chapter two, are becoming stronger than ever before. These noise sources can cause sever problems to the system performance and/or reliability ranging from decreasing the system throughput to causing glitches on wires that can result in function failure.

There are extensive researches on modeling/analysis of individual noise source mentioned previously [4-17]. However, combining different noise sources tend to aggravate one another's effects, which makes stand-alone noise analysis inaccurate. Therefore, a methodology that models the effects of compound noise sources is inevitable. At the same time, the methodology should consider the unpredictable nature of the compound noise sources. Chong Zhao et al in [18] have presented a methodology to analyze the effects of compound noise on the reliability of a digital design by calculating the softness (vulnerability to noise) of each node. They used simple analytical expression comprising the timing, electrical and logical masking, however, the model does not include the effect of

parameters variation and hence, more accurate and comprehensive methodology is needed.

In chapter 2, the comparison between the different low power design schemes has been done based on simulation. However, the simulation is impractical in case of large circuit; moreover, the simulation time is not linearly varied with the circuit size. And hence a fast, and at the same time, accurate methodology is needed to evaluate the noise-immunity of a given design.

In this part of the research, a methodology, which is comprehensively model the effects of compound noise sources on the reliability of a given CMOS digital circuit, is presented. In contrast with the previous works, this methodology has the following features:

(i) The effects of different noise sources are analyzed simultaneously.

(ii) At a given noise level, the logic error probability of each node in a design can be calculated, and hence a special design considerations can be given to the weak (noise-sensitive) nodes.

(iii) The methodology reports the error rate in terms of supply and transistor threshold voltage and hence it can be used efficiently during the design of low power CMOS digital circuits for low-power high reliability trade-off.

(iv) The methodology can be used to predict the noise-induced logic errors in a given design at realistic noise levels where the transistor-level simulation-based approach cannot complete within a reasonable time.

Through this study, we would like to propose a measure of reliability used for reliability-oriented logic synthesis/layout.

3.2 Noise Model

To accurately model the noise in a digital system, the model should include all kinds of noise sources. In this work, the noise model presented in chapter 2 [19] is recalled. The model is applied to the individual gate of a given system as shown in figure 3.1.

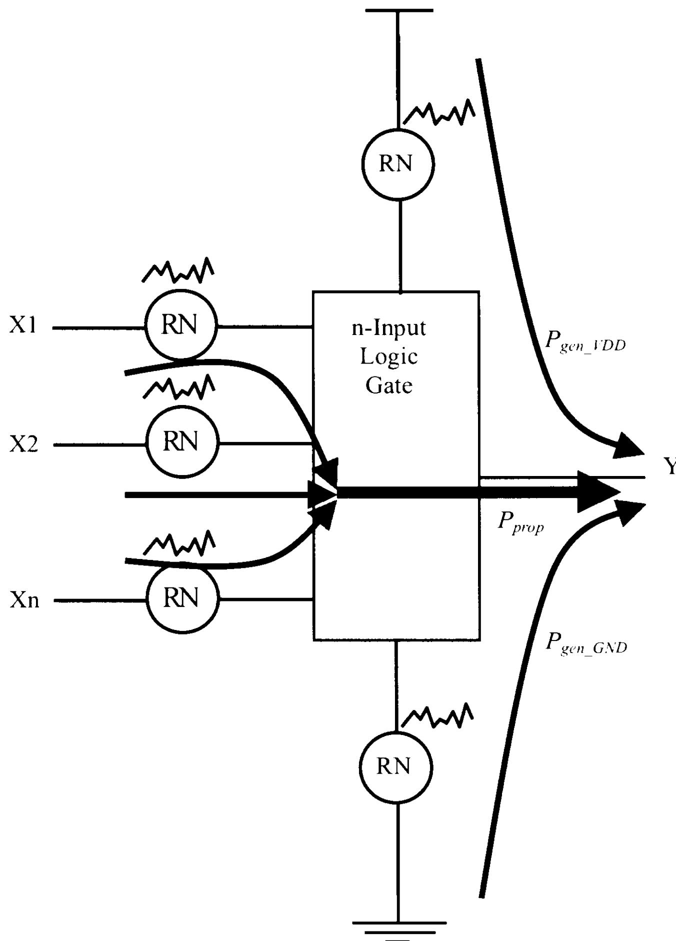


Figure 3.1 Illustrative diagram for the noise and error sources in a digital circuit.

3.3 Statistical Analysis of Logic Error

In this section, we aim to evaluate the logic error rate in a given CMOS digital circuit. As a first step, the noise-sensitivity of each node in a digital design is modeled. Using the model of this stage, the weak nodes in a system can be identified and hence special consideration can be given to these nodes during the design phase. The second stage of the model is to identify the overall noise immunity of a digital system under a given noise level. The second stage will be delayed to the future work.

3.4 Model Element

The model in this stage is composed of three factors, which are the electrical factor, timing factor and logic factor. In the following subsection, the evaluation of each factor and the effect of the combined factors are explained. The calculation of each factor is based on probabilistic assumptions of the noise sources and input patterns.

3.4.1 Electrical Factor

In a digital system, each gate is composed of specific number of transistors having threshold voltage V_{th} and supplied by a supply voltage V_{DD}. If the supply/ground line and/or the inputs are contaminated with noise, the error probability at the output will depend on V_{DD}, V_{th} and the noise level. The logic error probability is defined as the probable percentage fault at the gate output. Since that the logic error probability is strongly depend on the electrical parameters of gate, The logic error probability at a gate output (p_O) will be referred as the electrical factor and this term is used interchangeably with the logic error probability at a gate output. First, we analyze the noise-induced logic error probability in the basic building blocks of a digital circuit, which are inverter, NAND and NOR gates. The probable logic errors in the others gates can be calculated accordingly.

3.4.1.1 Logic Error probability in Inverter Gate

Based on the noise model discussed in the previous section, the inverter gate is affected by three noise sources as shown in figure 3.2. Consequently, the power/ground and input levels are fluctuated. If any (two or all) of them crosses a certain threshold level, the gate output will be likely in error. The error in the output comes either as propagated from the input or generated because of power supply and/or ground fluctuation. The input signal, power supply and ground are

assumed to be corrupted with noise having a Gaussian distribution function as shown in figure 3.3. Where the noise variances of input (0/1), V_{DD} and ground are σ^2_{I0}, σ^2_{I1}, σ^2_D and σ^2_G respectively, and V_{IL} is the maximum allowable input voltage to represent logic zero, and V_{IH} is the minimum input voltage that can be accepted as logic one. V_{IL} and V_{IH} are given by equations 3.1, 2 [20]. Referring to the noise model presented in chapter two, note that during the simulation or modeling of individual gates, both (Ncs+Ngs) and (Ncg+Ngg) are replaced by two noise sources having σ_D and σ_G respectively. Although V_{IL} and V_{IH} depend on the ratio of PMOS and NMOS sizes, we assumed symmetrical inverter for simplicity.

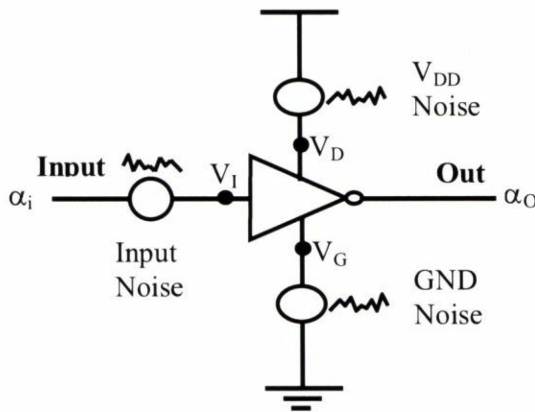


Figure 3.2 Noise in an inverter circuit.

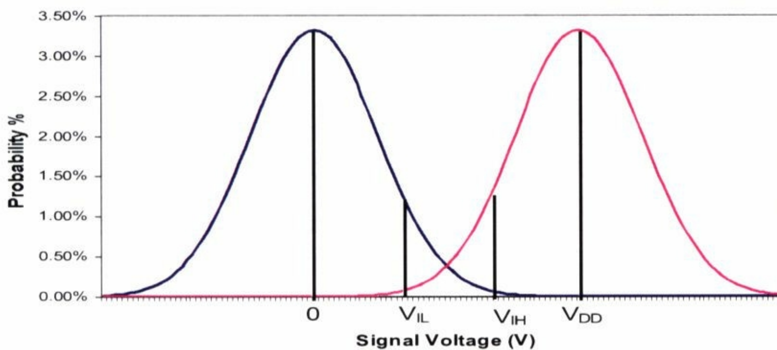


Figure 3.3 Probability distribution function of Input signal, Supply voltage and Ground voltage.

$$V_{IL} = \frac{1}{8}(3V_{DD} + 2V_{th}) \tag{3.1}$$

$$V_{IH} = \frac{1}{8}(5V_{DD} - 2V_{th}) \tag{3.2}$$

Consider the inverter gate as a transmission channel. The output will be likely in error in the following two cases:

1) Logic zero at the input is transmitted or generated as logic zero at the output. This will likely happen in the following circumstances:

i) If the difference between the input logic zero and the ground levels, (v_{i0g}=V_{I0}-V_G), exceeds V_{IL} while the ground voltage (V_G) is less than V_{IL} and the difference between the supply voltage and the input logic zero (v_{di0}= V_D-V_{I0}) is less than V_D-V_{IH}=(V_{IL}). In this case, the NMOS and PMOS will switch (by fault) as shown in figure 3.4-a. It is important to consider the value of V_G. That is, if V_G is higher than V_{IH}, the output will be logically correct though the input is not. To not to be pessimistic, we consider the values of V_G that are less than V_{IL} (we have ignored the cases of V_G in the range (V_{IL},V_{IH})). The condition, (v_{di0} is less than V_{IL}), is considered to limit the counted errors to the cases where the PMOS transistor is off.

ii) If the input logic zero level is correct (V_{I0}-V_G less than V_{IL} while the supply voltage (V_D) is less than V_{IH}. The value of V_D determines the output status. If V_D is less than V_{IH}, the output might be logically incorrect. The possible switching conditions and events are summarized in figure 3.4-c, wherein the solid lines show the considered conditions for the probable incorrect output zeros. The union of i and ii gives the probability of logic zero error, which can be formulated as follows:

$$\begin{aligned}
&prob\,(0_{inp} \rightarrow 0_{out}) = p_0 \\
&p_0 = p_{0\,prop} \cup p_{0\,gen} \\
&p_0 = p_{0\,prop} + p_{0\,gen} - p_{0\,prop}\, p_{0\,gen} \\
&p_0 \cong ABC + (1 - A)D
\end{aligned} \tag{3.3}$$

Where the subscripts *prop* and *gen* stand for propagation and generation respectively. *A* is the probability that v_{i0g} exceeds V_{IL}; *B* is the probability that V_G is less than V_{IL}, *C* is the probability that v_{di0} is less than V_{IL} and *D* is the probability that V_D is less than V_{IH}. *A, B, C* and *D* can be given by equations (3.4-3.7).

$$A = \frac{1}{\sqrt{2\pi(\sigma_{I0}^2 + \sigma_G^2)}} \int_{V_{IL}}^{\infty} e^{\frac{-(v_{i0g} - (\mu_{I0} - \mu_G))^2}{2(\sigma_{I0}^2 + \sigma_G^2)}}\, dv_{i0g} \tag{3.4}$$

$$B = \frac{1}{\sigma_G \sqrt{2\pi}} \int_{-\infty}^{V_{IL}} e^{\frac{-(v_g - \mu_G)^2}{2\sigma_G^2}}\, dv_g \tag{3.5}$$

$$C = \frac{1}{\sqrt{2\pi(\sigma_D^2 + \sigma_{I0}^2)}} \int_{-\infty}^{V_{IL}} e^{\frac{-(v_{di0} - (\mu_D - \mu_G))^2}{2(\sigma_D^2 + \sigma_{I0}^2)}}\, dv_{di0} \tag{3.6}$$

$$D = \frac{1}{\sigma_D \sqrt{2\pi}} \int_{-\infty}^{V_{IH}} e^{\frac{-(v_d - \mu_D)^2}{2\sigma_D^2}}\, dv_d \tag{3.7}$$

Where μ_{I0}, μ_G and μ_D are the average values of V_{I0}, V_G and V_D respectively.

2) Logic one at the input is transmitted or generated as logic one at the output. This will likely happen in the following circumstances:

i) If ($v_{di1}=V_D-V_{I1}$) is higher than V_{IL} while V_D is higher than V_{IH} and ($v_{i1g}=V_{I1}-V_G$) is less than V_{IL}. The NMOS and PMOS will switch (by fault) as shown in figure 3.4-b. In this case, the value of V_D determines the output

status. If V_D is higher than V_{IL}, the output might be incorrect, which means that the supply noise might correct the error caused by the input noise. We consider the values of V_D, which are higher than V_{IH} to not to be pessimistic. As in the case 1, the condition, (v_{i1g} is less than V_{IL}), is considered to limit the counted errors to the cases where the NMOS transistor is off.

ii) If the input logic one level is correct (v_{di1} less than V_{IL}) and V_G is higher than V_{IH}. If V_G is higher than V_{IH}, the output will be incorrect though the input is correct. In this case, figure 3.4-a summarizes the possible switching conditions and events, wherein the solid lines show the considered conditions for the probable incorrect output ones. The union of i and ii gives the probability of logic one error and it can be formulated as follows.

$$
\begin{aligned}
&prob\,(1_{inp} \rightarrow 1_{out}) = p_1 \\
&p_1 = p_{1prop} \cup p_{1gen} \\
&p_1 = p_{1prop} + p_{1gen} - p_{1prop}\, p_{1gen} \\
&p_1 \cong E(1-D)F + G(1-E)
\end{aligned}
\tag{3.8}
$$

Where E is the probability that v_{di1} is higher than V_{IL}, F is the probability that v_{i1g} is less than V_{IL} and G is the probability that V_G is higher than V_{IH}. E, F and G can be expressed by equation (3.9-3.11).

$$
E = \frac{1}{\sqrt{2\pi(\sigma_D^2 + \sigma_{I1}^2)}} \int_{V_{IL}}^{\infty} e^{\frac{-(v_{di1} - (\mu_D - \mu_{I1}))^2}{2(\sigma_D^2 + \sigma_{I1}^2)}} dv_{di1} \tag{3.9}
$$

$$
F = \frac{1}{\sqrt{2\pi(\sigma_{I0}^2 + \sigma_G^2)}} \int_{-\infty}^{V_{IL}} e^{\frac{-(v_{i1g} - (\mu_{I1} - \mu_G))^2}{2(\sigma_{I1}^2 + \sigma_G^2)}} dv_{i1g} \tag{3.10}
$$

$$
G = \frac{1}{\sigma_G\sqrt{2\pi}} \int_{V_{IH}}^{\infty} e^{\frac{-(v_g - \mu_G)^2}{2\sigma_G^2}} dv_g \tag{3.11}
$$

Where μ_{l1} is the average values of V_{l1}.

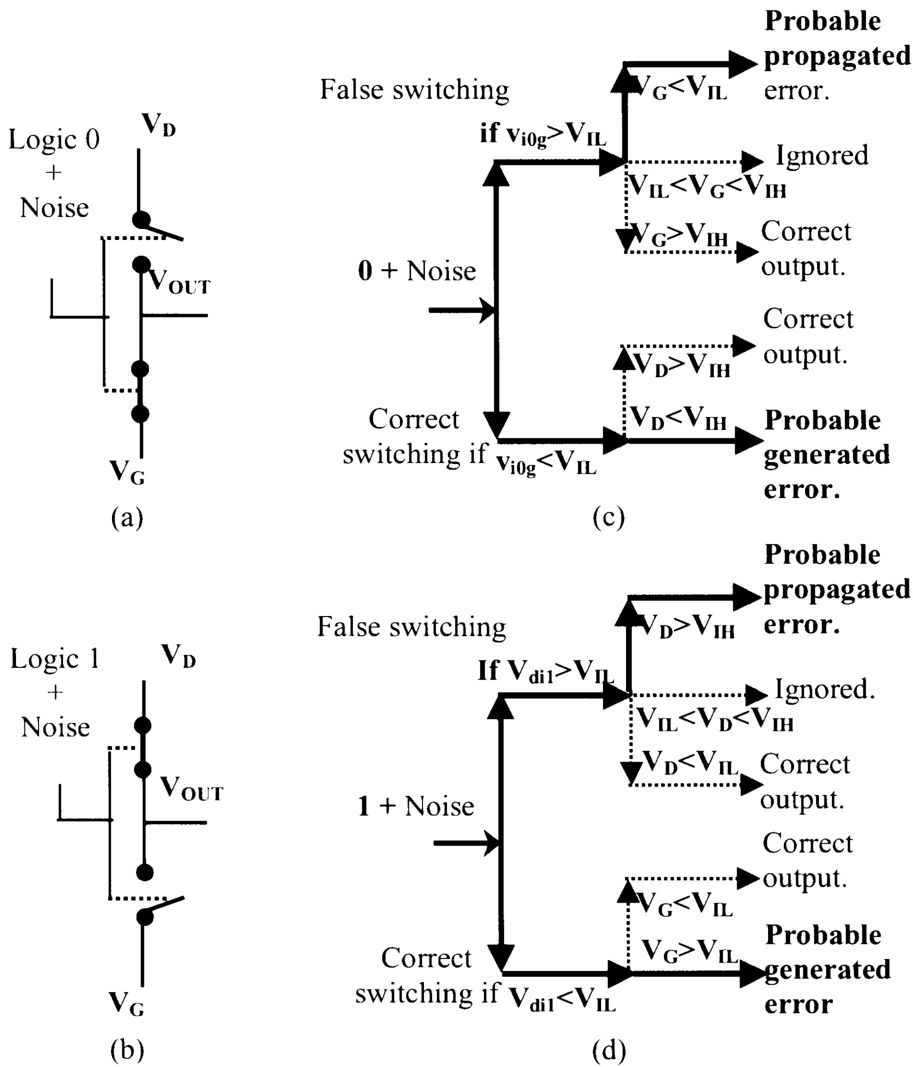


Figure 3.4 Erroneous switching and conditions in inverter circuit. a, b) False switching due to that the input is corrupted with noise. c, d) The possible conditions of the generated and propagated logic zero and one respectively (Solid lines). 'Ignored' means cases we omitted in our analytical modeling.

The total output error probability (probable logic error rate) p is calculated by equation (3.12).

$$\begin{aligned} p &= (1-\alpha)p_0 + \alpha p_1 \\ &\cong (1-\alpha)\big(ABC + (1-A)D\big) \\ &+ \alpha\big(E(1-D)F + G(1-E)\big) \end{aligned} \tag{3.12}$$

Where α is the probability that the input is logic one. α_o is equal to $(1-\alpha_i)$; where α_i, α_o are the static probabilities at the input and output respectively [2]. In the analysis, α_I is assumed to be 0.5.

As it has been mentioned in chapter two, the differences between the considered low power schemes are in V_{th} and V_{DD}. By using equation 3.12, we have calculated the probable logic error rate of three different inverter gates at different noise levels assuming that μ_{I0}, μ_G and μ_{I1}, μ_D are zero and V_{DD} respectively. The first gate has low V_{th} and is powered at low V_{DD}. It is referred to as (LV_{DD}-LV_{th}). The second has Low V_{th} and is powered at high V_{DD}. It is referred to as (HV_{DD}-LV_{th}). The third is powered at high V_{DD} and has high V_{th}. It is referred to as (HV_{DD}-HV_{th}). In this work, we assumed 0.18μm technology So that, we assigned 1.8V, 1.2V for HV_{DD} and LV_{DD} respectively; while LV_{th} and HV_{th} are assumed to be 0.15V and 0.4V respectively. The standard deviation of V_{DD}, GND and input noise sources (σ_D, σ_G and σ_I have been assigned similar values at each calculation point in all gates. The calculation results are shown in figure 3.5.

The results imply that, as the noise level increases, the curves saturate and all gates have almost equal error rate, that is, the noise level becomes larger than the threshold level of all gates. However, at low noise level, there is a clear difference between the error probability of the gates; more specifically, HV_{DD}-HV_{th} gate has higher noise immunity (lower logic error) than HV_{DD}-LV_{th} or LV_{DD}-LV_{th}. It can be inferred from these results that the system, which is composed (totally or partially)

of HV_{DD}-HV_{th} cells, would has lower logic error rate than that contains only HV_{DD}-LV_{th} or LV_{DD}-LV_{th} cells.

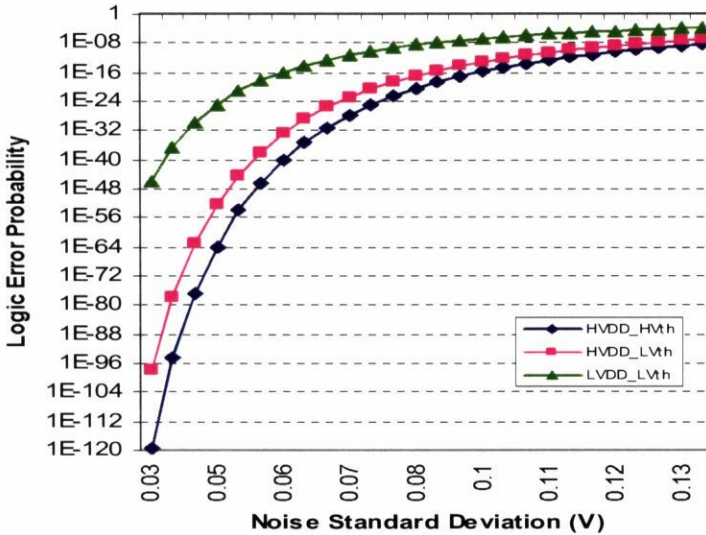
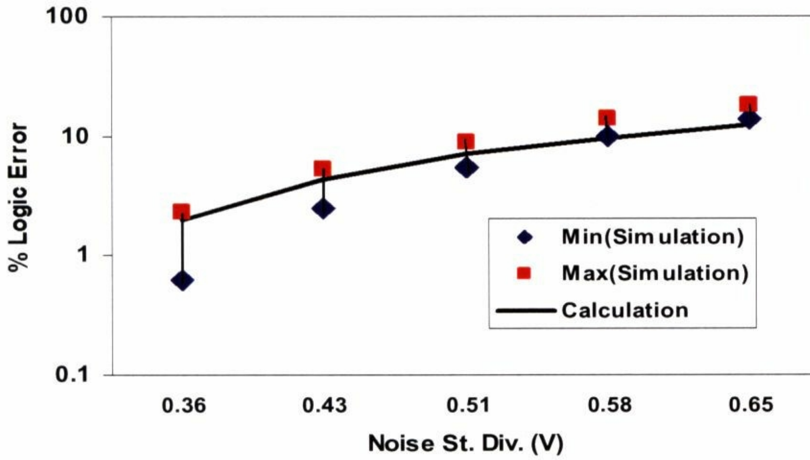


Figure 3.5 Logic error probabilities of three different gates. HV_{DD}=1.8V, LV_{DD}=1.2V, LV_{th}=0.15V and HV_{th}=0.4V. σ_D, σ_G and σ_I have been assigned similar values (shown on horizontal axis) at each calculation point in all gates.

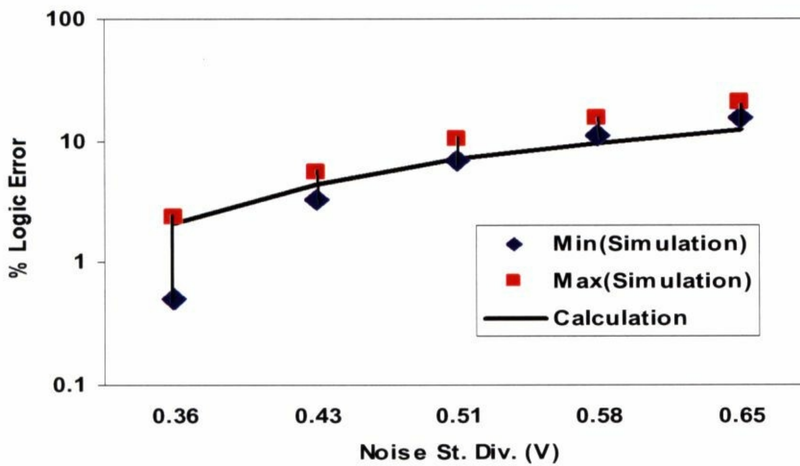
Before proceeding, to the next subsection, the accuracy of the logic error probability model of the inverter circuit is examined by comparing the model results with the simulation results of three inverter gates.

To examine the model, the simulation and calculation results of inverter have been compared as shown in figure 3.6. Note that it is difficult to find the logic error rate by simulation at practical noise values. So that, the comparison has been done at sever noise conditions. The confidence intervals of the simulation results have been calculated using the formula given in [21]. The confidence level is 0.99. The simulation and calculation have been carried out assuming similar values for σ_D, σ_G and σ_I at each calculation/simulation point. As shown in figure 3.6-a, 3.6-b, the calculation results of HV_{DD}-HV_{th} and HV_{DD}-LV_{th} fit well in the confidence

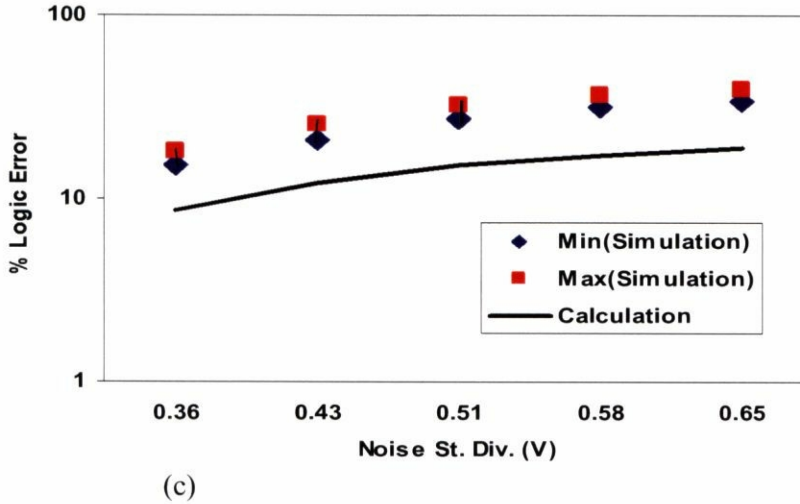
intervals (between the minimum and maximum values) of the simulation results. There is a relative systematic error (around -18%) between the simulation and calculation results of LV_{DD}-LV_{th} as shown in figure 3.6-c.



(a)



(b)



(c)

Figure 3.6 Simulated and calculated logic error of an inverter circuit. (a)HV_{DD}-HV_{th} Cell. (b)HV_{DD}-LV_{th} Cell. (c)LV_{DD}-LV_{th} Cell. σ_D, σ_G and σ_I have been assigned similar values (shown on horizontal axis) at each calculation/simulation point.

3.4.1.2 Logic Error in NAND Gate

We start by analyzing the logic error in two-input NAND gate and then we generalize the formulation to arbitrarily number of inputs NAND gate. Assume that there is a two-input NAND gate with the noise sources, which may affect its performance as shown in figure 3.7. p_1, p_2 and α_1, α_2 are the error and the static probabilities of the inputs respectively. α_o, p_o is the static and error probability of the output. The logic error at the gate's output has two sources. The first is the propagated from the inputs. The second is generated because of power/ground noise. We analyze the propagated error at first, and then we analyze the generated error.

The propagated error might happen in the following two cases:

(i) Only one input is incorrect.

(ii) All inputs are incorrect.

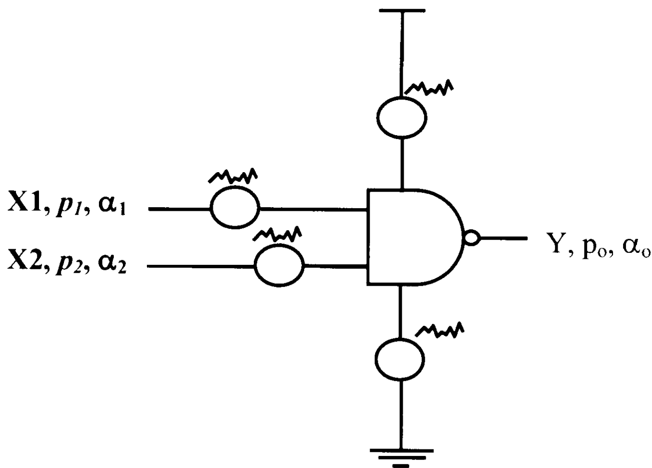


Figure 3.7 Noise in two-input NAND gate.

In the first case, the error will propagate if the correct input is logic one (If the correct input is logic zero, the output will be logic one regardless the value of the incorrect input) as shown in table 3.1.

X1	X2	Y
X	0	1
X	1	$\bar{X}$
0	X	1
1	X	$\bar{X}$

Table 3.1 Truth table of NAND gate, one input is in error.

The propagated error can be calculated using equation 3.13.

$$p_{pi} = \alpha_2(1 - p_2)p_1 + \alpha_1(1 - p_1)p_2 \qquad (3.13)$$

Where $(1\text{-}p_1)$ and $(1\text{-}p_2)$ are the probability that input1, input2 are correct respectively.

In case of all inputs are incorrect, by checking the truth table of NAND gate shown in table 3.2, the output will be likely error if all inputs are logic ones or

logic zeros. Therefore, the propagated error in such case can be calculated by equation 3.14.

$$p_{pii} = ((1-\alpha_1)(1-\alpha_2)+\alpha_1\alpha_2)p_1 p_2 \tag{3.14}$$

Correct Input and Output			False Input and Output		
X1	X2	Y	X1	X2	Y
0	0	1	1	1	0
0	1	1	1	0	1
1	0	1	0	1	1
1	1	0	0	0	1

Table 3.2 Truth table of two-input NAND gate, all inputs are in error.

Since the error probability is a small value, we can neglect the high order terms (p_1p_2, $p_1p_2p_3$, etc.) for simplicity. The total propagation error probability (p_{prop}) can be calculated by summing and simplifying equations (3.13, 3.14) as shown in equation (3.15)

$$p_{prop} \cong \alpha_2 p_1 + \alpha_1 p_2 \tag{3.15}$$

The second component of the output error probability is called generated error. This component is due to the noise acting on the power supply/ground lines. If all inputs are correct, the output might be error because of the supply level is less than V_{IH} or the ground level is higher than V_{IL}. By checking the gate truth table shown in table 3.3, the generated error can be calculated by using equation (3.16).

X1	X2	Correct Y	False Y
0	0	1	0
0	1	1	0
1	0	1	0
1	1	0	1

Table 3.3 Truth table of two-input NAND gate, effect of power/ground noise.

$$\begin{aligned} p_{gen} &= \alpha_1\alpha_2(1-p_1)(1-p_2)p_g \\ &\quad + (1-\alpha_1\alpha_2)(1-p_1)(1-p_2)p_d \\ p_{gen} &\cong \alpha_1\alpha_2 p_g + (1-\alpha_1\alpha_2)p_d \end{aligned} \tag{3.16}$$

Where p_g and p_d are the probability that V_G is higher than V_{IL} and the probability that V_D is less than V_{IH}. For simplicity, we considered that the term $(1-p_1)(1-p_2)$ is equal to unity. The error probability at the output (p_o) is the union of equations (3.15, 3.16); which can be calculated by equation (3.17).

$$\begin{aligned} p_o &= p_{prop} \cup p_{gen} \\ p_o &\cong \alpha_2 p_1 + \alpha_1 p_2 + \alpha_1\alpha_2 p_g + (1-\alpha_1\alpha_2)p_d \end{aligned} \tag{3.17}$$

The formula, shown in equation 3.17, depends on p_1, p_2, p_g and p_d. Note that p_{prop} is reduced to a formula where the output will be in error whenever one input is in error and the other is correct one. In this case the NAND circuit is collapsed to inverter gate as shown in figure 3.8, and hence, assuming symmetrical NAND gate, p_1, p_2, p_g and p_d can be calculated by using a method similar to that is used to calculate error the probability in inverter gate as shown by equation (3.18).

$$\begin{aligned} p_i &= p_{i1} + p_{i0} \\ p_i &= \alpha_i E_i(1-D)F_i + (1-\alpha_i)A_i B_i C_i \\ p_g &= G \\ p_d &= D \end{aligned} \tag{3.18}$$

Where the subscript i denotes the ith input and A, B, C, D, E, F and G are as defined above and p_{i1} is the probability that a logic one on input i is transferred as logic one and p_{i0} is the probability that a logic zero on input i is transferred as logic zero.

For n-input NAND gate, equations (3.15, 3.16) can generalized as shown by equations (3.19, 3.20) respectively.

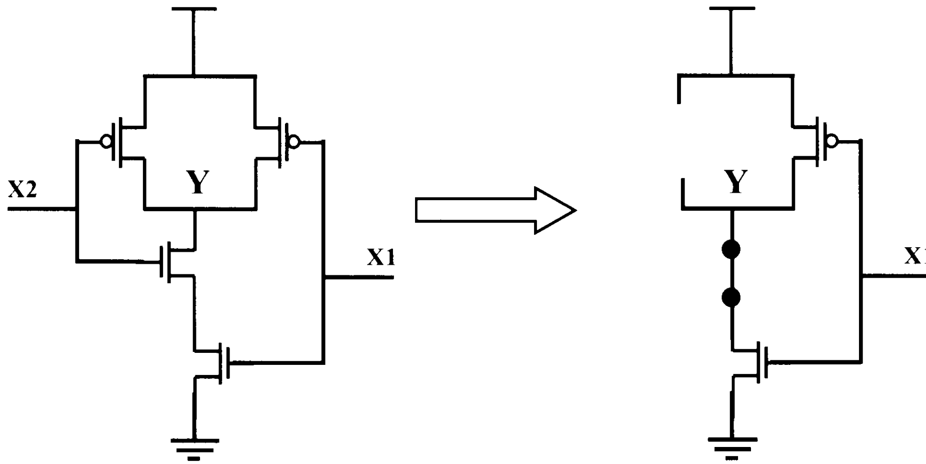


Figure 3.8 NAND gate is transferred to Inverter if one input is logic One.

$$p_{prop} \cong \sum_{i=1}^{n} p_i \prod_{j=1, j \neq i}^{n} \alpha_j \quad (3.19)$$

$$p_{gen} \cong p_g \prod_{i=1}^{n} \alpha_i + p_d (1 - \prod_{i=1}^{n} \alpha_i) \quad (3.20)$$

The static probability at the output can be calculated in terms of static probabilities of the inputs as shown in equation 3.21 [2].

$$\alpha_o = 1 - \prod_{i=1}^{n} \alpha_i \quad (3.21)$$

The accuracy of the formula is checked by comparing the calculation results with the results obtained form the simulation using HSPICE and the results are shown later in the next subsections.

3.4.1.3 Logic Error in NOR Gate

The noise-induced logic error analysis in NOR circuit is done in a similar way to that is shown in previous subsection. Assume that there is a two-input

NOR-gate with the noise sources as shown in figure 3.9. The difference between the error probability in NAND and NOR gate comes from the difference in functionality of both gates. The logic error at the gate's output has two components, which are the propagated error from the inputs and the generated error because of the noise acting on the power/ground lines.

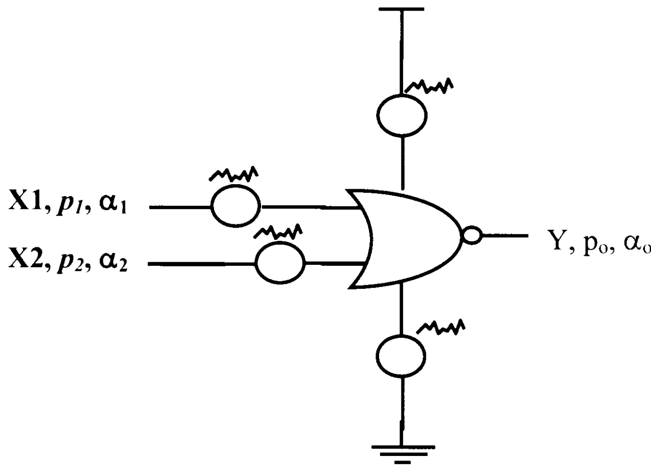


Figure 3.9 Noise in two-input NOR gate.

The analysis of the logic error in two-input NOR gate is done first, then the formula is generalized for n-inputs NOR gate. The error at the inputs might propagate to the output in the following two cases:

(i) Only one input is incorrect.

(ii) All inputs are incorrect.

In case (i) the error will propagate if the correct input is logic zero. Note that if the correct input is logic one, the output will be logic zero regardless the value of the incorrect input as shown in table 3.4.

The propagated error can be calculated using equation (3.22).

$$p_{pi} = (1-\alpha_2)(1-p_2)p_1 + (1-\alpha_1)(1-p_1)p_2 \tag{3.22}$$

Where $(1-p_1)$ and $(1-p_2)$ are the probability that the inputs of NOR gate are correct respectively.

X1	X2	Y
X	0	$\bar{X}$
X	1	0
0	X	$\bar{X}$
1	X	0

Table 3.4 Truth table of NOR gate, one input is in error.

If all inputs are incorrect, in such case and by checking the truth table of NOR gate, the output will be likely error if all inputs are logic one or logic zeros as shown in table 3.5. Therefore, the propagated error in such case can be calculated by equation 3.23.

$$p_{pii} = ((1-\alpha_1)(1-\alpha_2)+\alpha_1\alpha_2)p_1 p_2 \quad (3.23)$$

Correct Input and Output			False Input and Output		
X1	X2	Y	X1	X2	Y
0	0	1	1	1	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Table 3.5 Truth table of two-input NOR gate, all inputs are in error.

By neglecting the high order terms (p_1p_2, $p_1p_2p_3$, etc.) for simplicity, the total propagation error probability (p_{prop}) can be calculated by summing and simplifying equations (3.22, 3.23) as shown in equation (3.24)

$$p_{prop} \cong (1-\alpha_2)p_1 + (1-\alpha_1)p_2 \quad (3.24)$$

The noise sources acting on the power supply/ground lines cause the second component of the output error probability. If all inputs are correct, the output might be error because of the supply level is less than V_{IH} or the ground level is higher than V_{IL}. By checking the gate truth table shown in table 3.6, the output is always logic zero except the case that all inputs are logic zero and hence the generated error can be calculated by using equation (3.25).

$$p_{gen} \cong (1-\alpha_1)(1-\alpha_2)p_d + (1-(1-\alpha_1)(1-\alpha_2))p_g \tag{3.25}$$

Where p_g and p_d are is the probability that V_G is higher than V_{IL} and the probability that V_D is less than $V_{IH.}$ For simplicity, we considered that the term $(1-p_1)(1-p_2)$ is equal to unity.

X1	X2	Correct Y	False Y
0	0	1	0
0	1	0	1
1	0	0	1
1	1	0	1

Table 3.6 Truth table of two-input NOR gate, effect of power/ground noise.

The error probability at the output (p_o) is the union of equations (3.24, 3.25); which can be calculated by equation (3.26).

$$p_o = p_{prop} \cup p_{gen} \tag{3.26}$$

Assuming symmetrical NOR gate, p_1, p_2, p_g and p_d can be calculated by an equation similar to (3.18).

For n-inputs NOR gate, equations (3.24, 3.25) can generalized as shown by equations (3.27, 3.28) respectively.

$$p_{prop} \cong \sum_{i=1}^{n} p_i \prod_{j=1, j\neq i}^{n} (1-\alpha_j) \tag{3.27}$$

$$p_{gen} \cong p_D \prod_{i=1}^{n} (1-\alpha_i) + p_G (1 - \prod_{i=1}^{n} (1-\alpha_i)) \tag{3.28}$$

The static probability at the output can be calculated in terms of static probabilities of the inputs as shown in equation 3.29 [2].

$$\alpha_o = \prod_{i=1}^{n} (1-\alpha_i) \tag{3.29}$$

3.4.2 Timing Effect

The second factor determining the overall logic error probability in a system is the timing characteristics of the spurious pulse in addition to the clock period. The width and the generation timing of a spurious pulse determine weather (or not) it could be captured by the register element (FF) at the primary output. For a spurious pulse, to be captured, it should reach the input of the FF at T_{SU} (setup time) before the sampling edge of the of the control clock and sustain for T_H (hold time) after the sampling edge, which means that, the spurious pulse having width less than (T_{SU}+T_H) could not be captured by the memory element at the designated primary output. Moreover, the pulse should be generated at the node under test (T_D+T_{SU}) before the raising edge of the clock signal, where T_D is the delay time from the node under test (NUT) to the designated FF. For example, refereeing to figure 3.10, assume that the clock period is T and is started at reference time t=0, the delay time form the node under evaluation to the designated memory element is T_D; all the spurious pulses, which are generated after $t=T-T_D-T_{SU}$ will reach the FF after the raising edge of the clock signal, and hence they will not be captured by the FF.

To accurately model the timing effect, both the pulse width and generation time should be taken into account. Regarding the pulse width, we consider the width ranging from (T_{SU}+T_H) to T. the number of different pulses widths are theoretically infinite, however, to simplify the calculation, we assumed that the number of different pulses widths is limited to a specific number n.

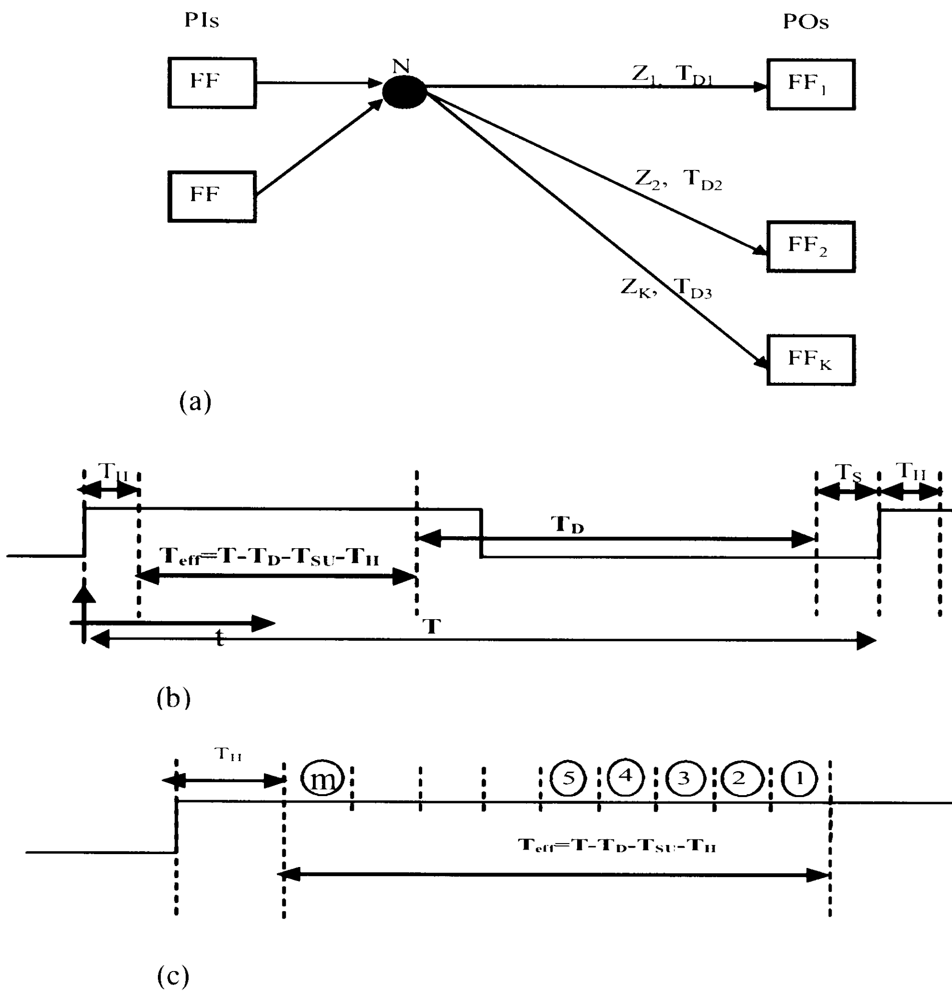


Figure 3.10 (a) Representation of a node (N) in a digital circuit. (b) Calculation of the spurious pulses effective timing window. (c) Dividing T_{eff} to a number of zones m.

In contrast with the other methodology concerning the modeling of time masking [11][18], In our methodology, both pulse-width and its generation time have been incorporated in the model of time masking τ_f. Assume that the noise pulse width having an arbitrarily distribution as shown in figure 3.11. This noise disturbs the node N, shown in figure 3.10-a. The probability of capturing a pulse (τ_f) is directly proportional to the ratio of its width (w_i) to the effective time interval T_{eff}, and hence, τ_f can be calculated as follows:

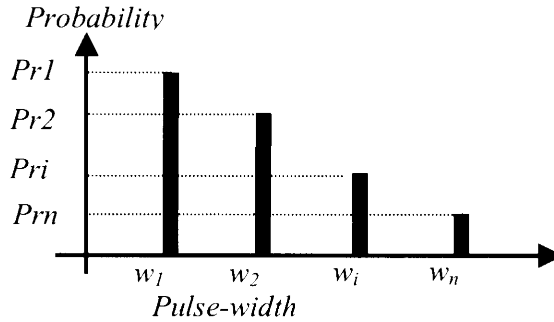


Figure 3.11 Arbitrarily noise-pulse width distribution.

$$\tau_{f1} = \frac{1}{T_{eff}} \int_{0}^{T_{eff}} \Pr(x) w(x) dx \cong \sum_{i=1}^{m} \Pr i . \frac{w_i}{T_{eff}} \quad (\mathrm{w_i} \le T_{eff})$$

$$\tau_{f2} = \int_{T_{eff}}^{T} \Pr(x) dx \cong \sum_{i=m+1}^{n} \Pr i \quad (\mathrm{w_i} > T_{eff}) \tag{3.30}$$

$$\tau_f = \tau_{f1} + \tau_{f2}$$

Equation 3.30 shows the general formula of τ_f, which can be adapted according to the noise distribution function as it is discussed below.

During the analysis and testing of our methodology, we have applied a noise pulses with widths having uniform distribution. We considered that the number of different pulse widths is n, and hence the probability of each width is 1/n, where n can be given by equation 3.31

$$n = \frac{T}{T_{\text{int}}} \tag{3.31}$$

Where T_{int} is stand for the interval difference between two successive pulses widths.

For a gate N in a path Z, we define an effective timing window T_{eff}, as shown in figure 3.10-b. Within T_{eff}, if a spurious pulse is generated, it will be likely

captured by a FF. We divide T_{eff} to a number of zones m, as shown in figure 3.10-c, where m can be given by equation 3.32.

$$m = \frac{T_{eff}}{T_{int}} \tag{3.32}$$

For better accuracy, T_{int} should be taken as small as possible. In our calculation, we have assigned the narrowest pulse width that can by captured by FF to T_{int}, which is equal to $(T_{SU}+T_H)$.

Applying the formulated τ_f model to the assumed noise distribution, τ_f can be calculated as follows:

$$\begin{aligned} \tau_f &= \frac{1}{n}\left[\frac{1}{m}+\frac{2}{m}+...+1+(\mathrm{n}-\mathrm{m})\right] \\ \tau_f &= \frac{1}{n}\left(n+\frac{1-m}{2}\right) \end{aligned} \tag{3.33}$$

Note that the minimum value for m is one.

3.4.3 Logic Masking

Through the path from the node under test to a designated primary output (PO), the noise pulses may cease to reach the destination. That is the pulse passes by logic gates whose output is determined by another signal lines. If those lines carry a specific pattern, the pulse could be blocked and hence it will not appear at the input of the PO. Hence, this effect introduces another factor in calculating the logic error probability, which is called logic masking. In this section, the calculation of logic un-masking is presented. The logic un-masking (L_f) is the probability that a specific path is transparent to a spurious pulse. To calculate L_f, the static probability (the probability of signal line to be logic high) at each gate output should be calculated first. Accurate estimation for the static probability requires exhaustive manipulation for the different input vectors, which is difficult. A good approximation is to assume that the static probability of the primary inputs

equal to 0.5. The estimation of other node static probability and the un-masking factor L_f are then calculated accordingly by using the breadth-first search (BFS) algorithm. By the aid of circuit shown in figure 3.12, the calculation of static probability (α_N) and L_f are explained.

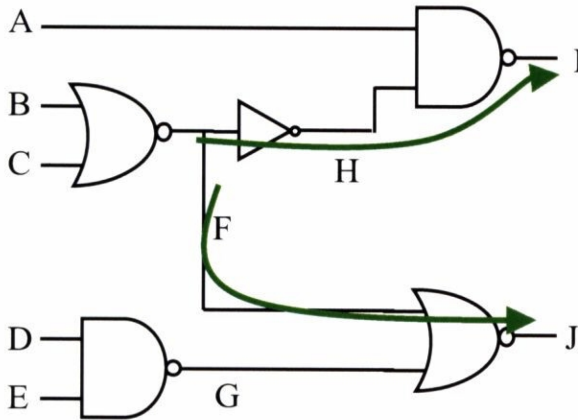


Figure 3.12 Schematic diagram of illustration example.

In the illustration example, A, B, C, D and E are the primary inputs (PIs) and I and J are the primary outputs (POs). Assuming that the static probabilities of PIs are given and each is equal to 0.5.

First; using equations 3.21, 3.29 and breadth-first search algorithm [23], the static probabilities of the other nodes can be calculated as follows:

$\alpha_F=1-\alpha_B\alpha_C=1-1/4=3/4$

$\alpha_G=(1-\alpha_D)\ (1-\alpha_E)=1/4$

$\alpha_H=1-\alpha_F=1/4$

$\alpha_I=(1-\alpha_A)\ (1-\alpha_H)=3/16$

$\alpha_J=1-\alpha_F\alpha_G=13/16$

Second; (suppose that node F is the node under test (NUT)) search for path form F to all POs.

Path1: FHI

Path2: FJ

Third; calculate L_F for each path.

Starting form the source node (F), the calculation of L_f is proceeded by calculating the subsequent node-error-transparency (ET) assuming ET(source node) is one. The procedure is carried out as follows:

For path1:

ET(H)=ET(F)=1; Inverter gate

ET(I)=ET(H).α_A=1.1/2=1/2; NAND gate

If I is a PO then L_{f1}=ET(I);

For path2:

ET(J)=ET(F).(1-α_G)=1.3/4=3/4; NOR gate

If J is a PO then L_{f2}=ET(J);

The above procedure is repeated until the last node in a design.

3.5 Model Formulation

The logic error probability (p_O) calculated by equations 3.12, 3.17, and 3.26 in conjunction with the timing factor (τ_f) given by equation 3.30 and logic masking L_{fZ} are combined to calculate the logic error probability p_{NZ} of a gate N on a path Z at the designated FF. Assuming linear proportionality between p_{NZ} and p_O, τ_f and L_{fZ}, p_{NZ} can be calculated by equation 3.34

$$p_{NZ} = W_{fN}.(p_{oN}\tau_{fZ}L_{fZ}) \tag{3.34}$$

Where W_{fN} is a weighting factor used to add more flexibility to the model. Despite that, we gave W_{fN} a unit value during the calculation; in future we can identify its values empirically base on on-chip noise measurement [22]. Assuming that the NUT N is existing in a digital circuit as shown in figure 3.10-a, the overall logic error probability at the primary outputs of a given circuit due to noise is the

summation of logic error probability at each primary output, which is connected to the node. If the number of paths from the output of gate N to the primary output is K, then the overall logic error probability p_N due to the noise at the gate can be given by equation 3.35.

$$p_N = W_{fN} p_{oN} \sum_{Z=1}^{K} \tau_{fZ} L_{fZ} \tag{3.35}$$

Where Z is the path index.

3.6 Results and Discussion

3.6.1 Model Testing

Before proceeding and using the model, it is better to check the model accuracy by comparing its results with a transistor-level simulation of the same gates at specific noise levels. We have simulated the individual logic gates with the possible noise sources, as shown in figures 3.2, 3.7, 3.9, at different noise levels by using HSPICE assuming 0.18μm technology and assuming normal distribution for the noise magnitude and uniform distribution for the noise pulse width. In the simulation file, each gate output is connected to an edge-triggered flip-Flop. We analyzed the HSPICE output files to calculate the logic error rate of each gate at different noise levels. The simulation takes long time, so that, we have simulated the gates with limited number of input sequences, then; the confidence intervals of the simulation results have been calculated using the formula given in [21]. The confidence level is taken to be 0.99. Since the maximum probable simulation results are the most important, we have added the half of confidence interval to the simulation results to get the maximum possible logic error rate at each noise level. Using our model, we calculated the probable logic error of each gate at noise levels similar to those were used in simulation. Because of lack of

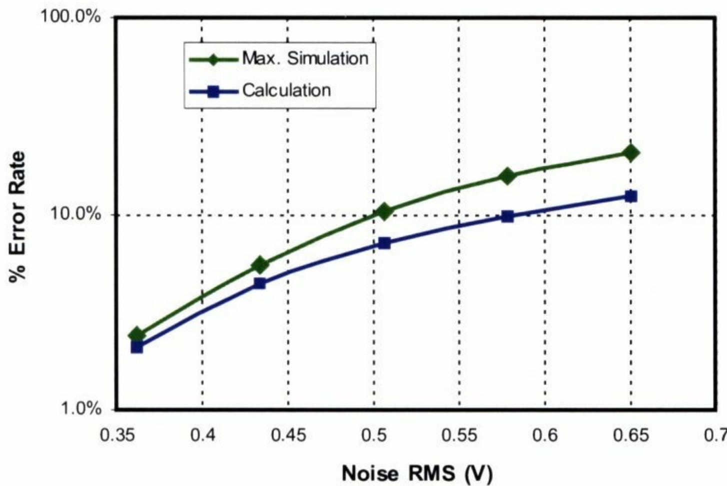
information about the relationship between σ_{I0}^2, σ_{I1}^2, σ_D^2 *and* σ_G^2, we have assumed that they are identical. The static probabilities of the inputs are assumed to be 0.5 for each. Figure 3.13-a, b and c show comparison between the simulation and calculation results. Note that it is very difficult to obtain a noise-induced logic error by simulation at low noise level, where it takes a very long simulation time. So that, to accelerate the logic error occurrence, we make the simulation at sever noise levels.

Figure 3.13 shows that the logic error probability, which is calculated using our model fits with that the maximum value of the logic error probability estimated form HSPICE simulation. In contrast with HSPICE, using our model, the designer has the capability to estimate the logic error probability in a gate at very low noise levels within part of a Second of the CPU time. Since the gates are directly attached to the FFs, neither the timing factor τ_f nor the logic un-masking L_f are taken into account during the calculation. In this case, it is assumed that all the generated/propagated error pulses are sampled by the FFs. When we apply the model to a gate within a circuit, τ_f and L_f are considered as it is shown in next section.

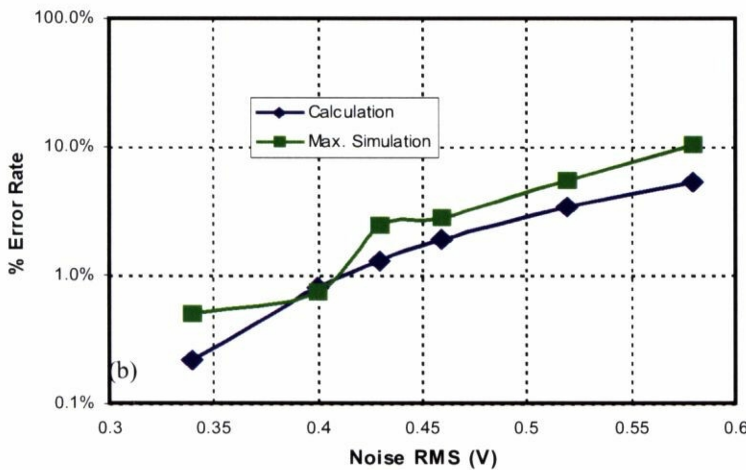
3.6.2 Accuracy and Efficiency

One of the objectives of the model is to locate the weak nodes (parts) against the noise during the design phase. To evaluate the accuracy and efficiency of the model for this purpose, we have applied the model to two circuits and compared the results with those obtained form HSPICE. The circuits are 4bit adder (ckt1) and random logic combinational circuit (ckt2). In the simulation, random noise sources are applied to each gate individually in the circuit, then, the circuit is simulated and the number of error pulses, which are sampled by the memory elements at the outputs is calculated. The logic error rate at the output is the ratio of the number of the error pulses to the number of input sequences.

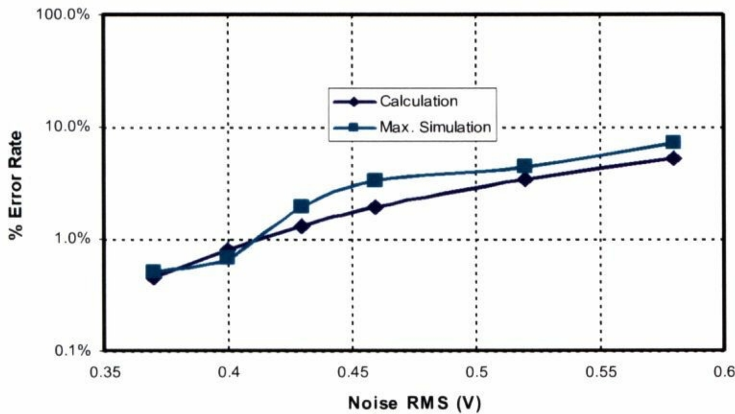
The noise sources are assumed to have random amplitude with normal distribution and uniform distribution for pulse width. Because of lack of information about the relationship between σ_{I0}^2, σ_{I1}^2, σ_D^2 *and* σ_G^2, we have assumed that they are identical. The static probabilities of the inputs are assumed to be 0.5 for each. Figure 3.14-a, b show the comparison between the calculation results and the transistor-level simulation ones obtained by HSPICE.



(a)



(b)



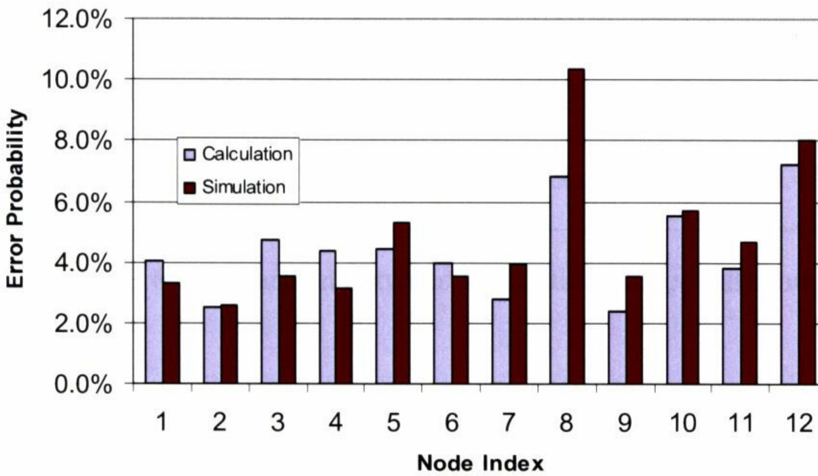
(c)

Figure 3.13 Comparison between the maximum simulation and model results.(a) Inverter gate (b) NAND2 gate (c) NOR2 gate.

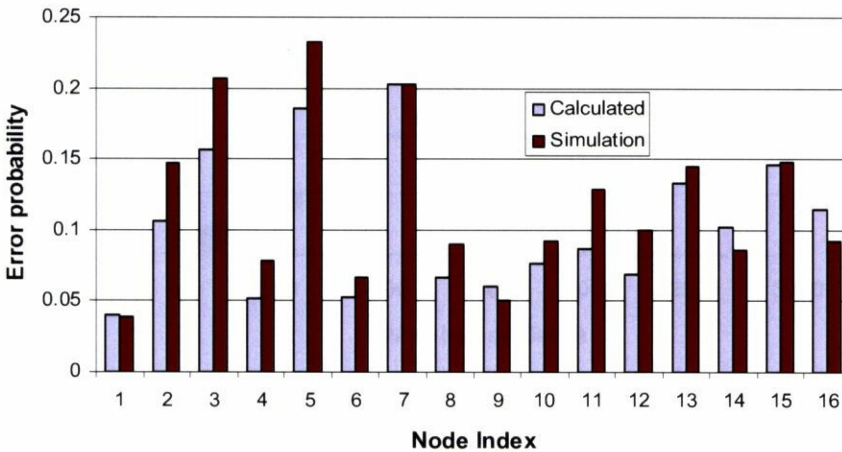
The model has been applied to evaluate the noise-induced logic error in the different nodes in ckt3 and ckt1 of chapter two. Here they are referred to as ckt2 and ckt1 respectively. The delay of the different gates is obtained by individual simulation for each gate and is feed for the program during the calculation. The logic error is also evaluated by using HSPICE. The simulation and calculation results are plotted in figure 3.14, which reveals that our methodology is accurate, that is the calculation results fit well with the simulation. The vertical axis of figure 3.14 indicates the node sensitively for the noise. For example, node 8 and 12 in ckt2 are the most sensitive nodes to the noise and hence, to increase the system overall immunity, it should be given special strengthen consideration during design phase. The same considerations should be given to nodes 3, 5 and 7 in ckt1.

To show how our methodology is fast compared with the transistor-level simulators, we have indicated the calculation time of our methodology and the simulation time using HSPICE in table 3.6. It is clear that our method speeds up the evaluation process by a factor of more than 1000 times higher than the

transistor-level simulator. Furthermore, as the circuit complexity increases, HSPICE simulation time increases drastically. However, our methodology shows calculation time linearity as the circuit's size increases. The methodology has been applied to some benchmark circuit to test its linearity; the calculation time versus the number of nodes is plotted in figure 3.15.



(a)



(b)

Figure 3.14 The simulation and calculated results of a) ckt2 and b) ckt1.

As we mentioned earlier, our methodology incorporate important design parameters like V_{DD}, V_{th} and hence the methodology is reliable not only to discover the weak area during the design phase but also can be used to optimize the design in terms of reliability, speed and power consumption.

	Number of nodes		Simulation Time	Calculation Time
ckt1	16(simu.)	27(calc.)	19530 sec	7.3 sec
ckt2	12(simu.)	15(calc.)	5045 sec	3.45 sec

Table 3.6 Comparison between simulation and calculation speed.

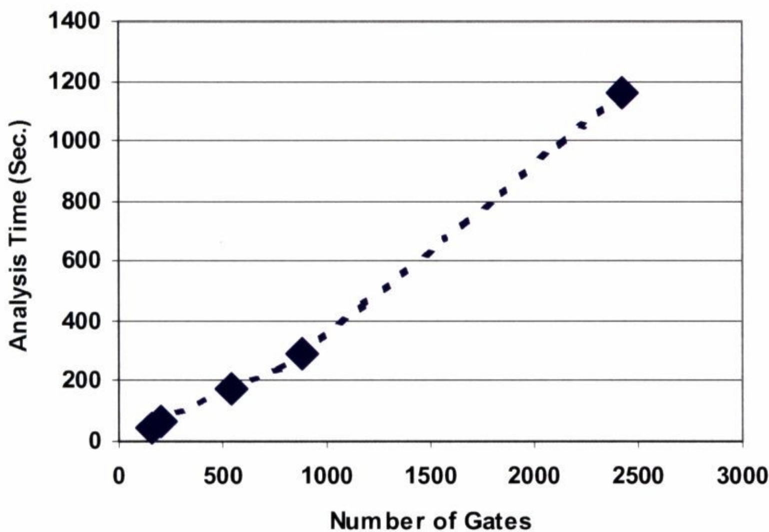


Figure 3.15 Analysis time of our methodology versus the number of gates in a circuit.

3.7 Chapter Conclusion

This chapter presents a methodology to evaluate the noise-induced logic error probability in a given CMOS digital design in terms of supply voltage, threshold voltage, noise level and circuit configuration. At first, we modeled the logic error probability in the different logic gates. Then, time masking has been modeled to include the effect of variation of the spurious pulse width and generation time. The model also considers the effect of logic masking. The model has been used to evaluate the logic error probability caused by the noise at the different nodes in two digital circuit examples. The model results have been compared with results obtained from HSPICE simulation. The comparison reveals that the model fit with the expected simulation results achieving speedup factor of more than 1000 over HSPICE. The calculation time of the methodology is proportional with the number of gates in a design, and hence, the method is suitable for investigating the big circuits. The model can be used to identify the weak parts against the noise in a given design during the design phase and hence it helps the designer in giving specific design considerations to strengthen the weak nodes.

Despite that the results show good accuracy for the model, it could be more accurate if the path sensitization is calculated taking into account the correlation between the different lines. The delay of different paths has been calculated by accumulating the delay of the gate in the path and a comprehensive logic analysis for the circuit under test is not done, and hence, false path detection cannot be done in this stage. Moreover, the study will be more attractive if it could be done using a real (measured) noise distribution. The model can be extended to estimate the overall reliability of a system and/or can be used to anticipate the design reliability after fabrication if the design is comprises an on-chip noise characterization circuit.

3.8 References

[1] http://public.itrs.net/

[2] J. Rabaey, A. Chandrakasan and B. Nikolic, Digital Integrated Circuits: A Design Perspective, Prentice Hall Electronics and VLSI Series, Pearson Education Inc., New Jersey, 2003.

[3] X. Aragonès, J. Gonzalez, F. Moll and A. Rubio, "Noise generation and coupling mechanisms in deep-submicron ICs", IEEE Design and Test of Computers, Sept.–Oct. 2002, pp. 27 - 35.

[4] P. Heydari and M. Pedram, "Ground bounce in digital VLSI circuits", IEEE Transaction on VLSI Systems, Vol. 11, No. 2, April 2003, pp. 180 – 193.

[5] Y. Jiang and K. Cheng, "Analysis of performance impact caused by power supply noise in deep submiron devices", Proceedings of IEEE Design Automation Conference, June 1999, pp. 760 - 765.

[6] P. B. Sabet and F. Ilponse, "A model for crosstalk noise evaluation in deep submicron process", Proceeding of IEEE ISQED, March 2001, pp. 139 - 144.

[7] J. Lou and W. Chen, "Crosstalk-aware placement", IEEE Transactions of Design & Test of Computers, Jan.- Feb. 2004, pp. 24 - 32.

[8] H. Li et al, "Comprehensive frequency-dependent substrate noise analysis using boundary element methods" Proceeding of ICCAD 2002, pp. 2 - 9.

[9] C. Hess et al, " Logic characterization vehicle to determine process variation impact on yield and performance of digital circuits" Proceeding of ICMTS 2002, pp. 189 - 196.

[10] Y. S. Dhillon, A.U. Diril and A. Chatterjee, "Soft-error tolerance analysis and optimization of nanometer circuits" Proceeding of DATE05, Vol. 1 pp. 288 - 293.

[11] Y. Deguchi, N. Ishiura and S. Yajima, "Probabilistic CTSS: analysis of timing error probability in asynchronous logic circuits", Proceedings of 28th ACM/IEEE , Design Automation Conference, 1991, pp. 650 - 655.

[12] M. A. Mendez, J. L. Gonzalez and A. Rubio, "An approach to the statistical characterization of the switching noise", Proceedings of DCIS2003, pp. 57 - 62.

[13] M. H. Chowdhury and Y. I. Ismail, "Possible noise failure mode in static and dynamic circuits", Proceedings of the 4th IEEE International Workshop on System-on-Chip for Real Time Applications, 2004. pp. 123 - 126.

[14] M. Omana, G. Papasso, D. Rossi and C. Metra, "A model for transient fault propagation in combinatorial logic", Proceedings of the 9th IEEE international On-Line test Symposium 2003, pp. 111 - 115.

[15] Y. S. Dhillon, A. U. Diril and A. Chatterjee, "Soft-error tolerance analysis and optimization of nanometer circuits", Proceedings of the Design, Automation and test in Europe Conference and Exhibition (DATE'05), 2005, Vol. 1, pp. 288 - 293.

[16] P. E. Dodd and L.W. Massengill, "Basic mechanisms and modeling of single-event upset in digital microelectronics" IEEE Transaction on Nuclear Science, Vol. 50 No. 3, June-2003, pp. 583 - 602.

[17] P. Hazucha, C. Svensson and S. A. Wender, " Cosmic-ray soft-error characterization of a standard 0.6μm CMOS process", IEEE Journal of Solid State Circuits, Vol. 35, No. 10 October-2000, pp. 1422 - 1429.

[18] C. Zhao, S. Dey and X. Bai, "Soft-spot analysis: Targeting compound noise effects in nanometer circuits" IEEE Design and Test of computers, July-Aug. 2005, pp. 362 - 375.

[19] M. Abbas M. Ikeda and K. Asada, "Noise effects on performance of low power design schemes in deep submicron regime", Proceedings of DFT 2004, pp. 87 - 95.

[20] S. Kang and Y. Leblebici, "CMOS Digital Integrated Circuits: Analysis and Design", McGraw-Hill, New York 1996.

[21] G. Hahn and S. Shapiro, "Statistical Models in Engineering", John Wiley & Sons Inc., USA, 1967

[22] M. Abbas, M. Ikeda and K. Asada, "On-chip detector for non-periodic high-swing noise detection" Proceedings of ISOCC2005, S. Korea, 2005.

[23] S. H. Gerze, "Algorithms for VLSI Design Automation", John Wiley & Sons Ltd. 1999.