1. Logical Effort

1.1. Gate Delay Modeling

The logical effort method [1, 2] proposes a simple and easy-to-use delay model for a gate. As shown in Figure 1, the model is RC-based: resistance R_{gate} represents the pull-up and pull-down capabilities of gates, whereas the load at the output consists of the capacitance of the external load and the internal parasitic capacitance of the gate.

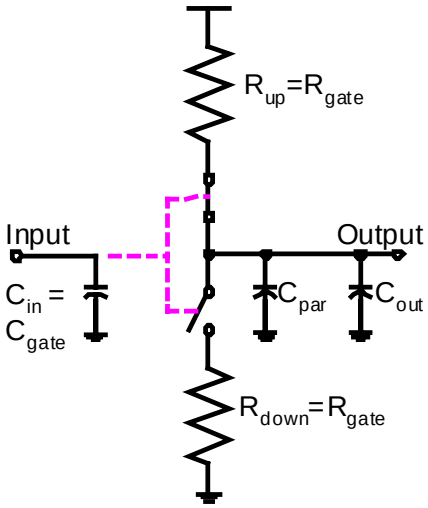

Figure 1. RC-based delay model of a gate

The delay of the gate is proportional to the RC constant

$$T_d = \kappa R_{gate}(C_{out} + C_{par}) = (\kappa R_{gate} C_{gate})\left(\frac{C_{out}}{C_{gate}}\right) + (\kappa R_{gate} C_{par})$$

For an inverter,

$$T_d = (\kappa R_{inv} C_{inv})\left(\frac{C_{out}}{C_{inv}}\right) + (\kappa R_{inv} C_{par})$$

The constant κ is constant and determined by process technology. Notice that components $R_{gate}C_{gate}$ and $R_{gate}C_{par}$ are approximately constant and depend on gate topology and process technology only. In other words, they characterize the delay behavior of the gate.

In addition, the relative ratios of $R_{gate}C_{gate}$ and $R_{gate}C_{par}$ are not significantly different for different technologies. Therefore, The delays of gates are most conveniently normalized to a unit delay of inverter, $\tau = \kappa R_{inv}C_{inv}$ to achieve the technology independent form.

$$T_{d,norm} = \left(\frac{\kappa R_{gate} C_{gate}}{\kappa R_{inv} C_{inv}}\right)\left(\frac{C_{out}}{C_{gate}}\right) + \left(\frac{\kappa R_{gate} C_{par}}{\kappa R_{inv} C_{inv}}\right)$$

or, in shorter notation,

$$T_{d,norm} = gh + p = f + p$$

where $g = \frac{\kappa R_{gate} C_{gate}}{\kappa R_{inv} C_{inv}}$: logical effort, characterizing the delay per unit of external load

$h = \frac{C_{out}}{C_{gate}}$: electrical effort, normalizing external load to gate input

$f = gh$: delay effort, equivalent to the fan-out delay

$p = \frac{\kappa R_{gate} C_{par}}{\kappa R_{inv} C_{inv}}$: parasitic delay, characterizing the inherent delay within the gate

The graphical representation of the delay is illustrated in Figure 2.

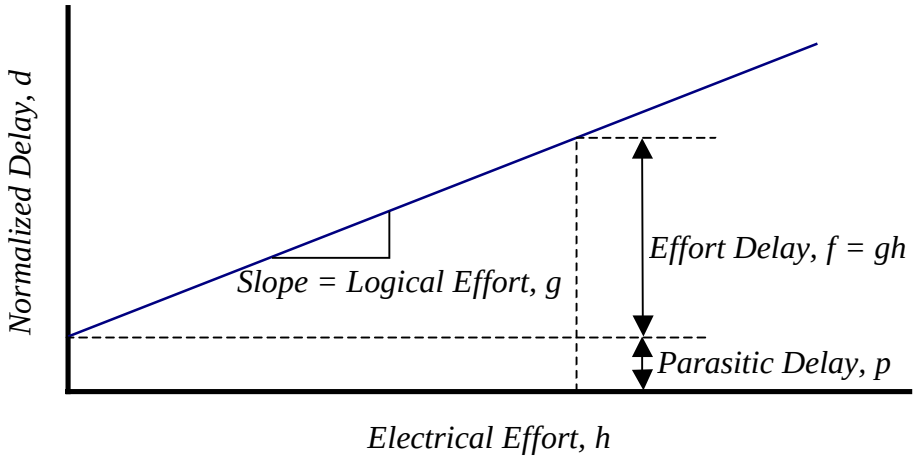

Figure 2. Delay Components of Logical Effort Model

1.2. Estimation of *g* and *p* Terms

As mentioned above, the logical effort g and the parasitic delay p are constant for a given technology. They depend on the topology of the gate.

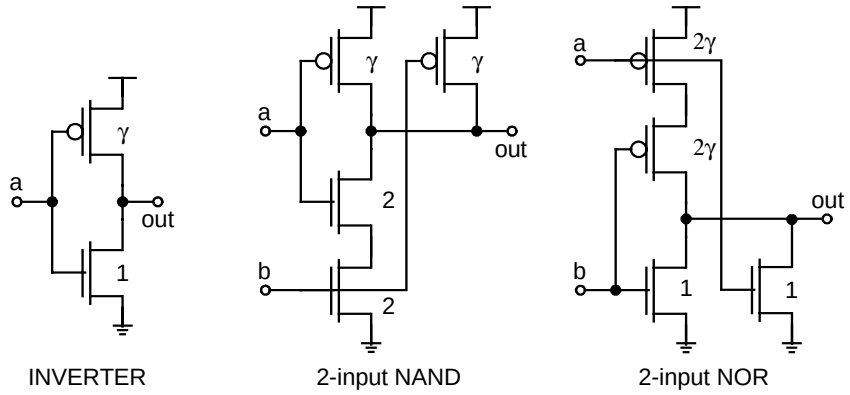

Figure 3. Basic CMOS gates

Figure 3 shows the schematics of some basic CMOS gates, where γ is the well-known effective PMOS-to-NMOS ratio. The logical effort *g* is estimated in two steps. First, the gates are sized such that they have the same driving capability as an inverter. Then, the

logical effort is the input size ratio of the gate relative to the inverter. The estimated *g* values of some CMOS gates are listed in Table 1.

Table 1. Estimated logical effort *g*

Gates	***Logical Effort, g***
INV	1
n-input NAND	$(\gamma + n)/(\gamma + 1)$
n-input NOR	$(n\gamma + 1)/(\gamma + 1)$
2-input XOR	4

Similarly, the estimated parasitic delay of a gate is taken relative to that of the inverter. Table 2 shows a very rough estimation.

Table 2. Estimated logical effort *g*

Gates	***Parasitic Delay, p***
INV	p_{inv}
n-input NAND	np_{inv}
n-input NOR	np_{inv}
2-input XOR	$4p_{inv}$

1.3. Application to Gate Sizing of Paths

The above delay model can be used to minimize delay along a path of gates. An n-gate path is shown in Figure 4. Assume fixed output load C_{Load} and fixed input capacitance C_{in}.

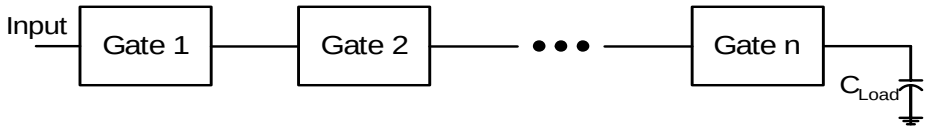

Figure 4. n-gate single path

The total delay in the path is

$$T_{d,total} = \sum_{i=1}^{n}(g_i h_i + p_i) = \sum_{i=1}^{n}\left[g_i\left(\frac{C_{i+1}}{C_i}\right) + p_i \right], with\, C_{n+1} = C_{Load}$$

The minimal delay is found when the capacitance C_i satisfy $\partial T_{d,total}/\partial C_i = 0$ for every i from 2 to n. Then, for i = 2→n

$$\frac{\partial T_{d,total}}{\partial C_i} = g_{i-1}\frac{1}{C_{i-1}} - g_i\frac{C_{i+1}}{C_i^{\,2}} = 0 \Leftrightarrow g_{i-1}\frac{C_i}{C_{i-1}} = g_i\frac{C_{i+1}}{C_i} \Leftrightarrow f_{i-1} = f_i$$

That is, $f_1 = f_2 = \ldots = f_n$. It is equivalent to sizing all gates to a fixed fan-out delay. Denote the path effort F as

$$F = \prod_{i=1}^{n} f_i = \left(\prod_{i=1}^{n} g_i\right)\left(\frac{C_{out}}{C_{in}}\right) = GH$$

For optimal delay,

$F = f_o^n$ or $f_o = F^{1/n} = (GH)^{1/n}$

The sizing of all gates in the path can be found using $f_0 = g_i C_{i+1}/C_i$. As C_{in} and C_{out} are known, the gate sizes can be started from either the input or the output.

In reality, a circuit typically consists of multiple paths that can merge at some internal nodes. The above optimal solution is also applied as long as off-path load is proportional to on-path load *regardless of input and output conditions*. In order to account for off-path load, the path effort F needs to be modified.

$$F = \prod_{i=1}^{n} f_i = \left(\prod_{i=1}^{n} g_i\right)\left(\prod_{i=1}^{n} \frac{C_{on,i} + C_{off,i}}{C_{on,i}}\right)\left(\frac{C_{out}}{C_{in}}\right) = GBH$$

Note that the total branching factor $B = \prod_{i=1}^{n} b_i = \prod_{i=1}^{n}\left(\frac{C_{on,i} + C_{off,i}}{C_{on,i}}\right)$ is added.

For optimal delay,

$F = f_o^n$ or $f_o = F^{1/n} = (GBH)^{1/n}$

Example 1. Given a 4-gate path as seen in Figure 5 with $C_{out} = 5.4$ and $C_{in} = C_1 = 1$. Assume $\gamma = 2$ and $p_{inv} = 1$. Using Tables 1 and 2, $(g, p)_{NAND2} = (4/3, 2)$ and $(g, p)_{NAND2} = (5/3, 2)$.

- Total logical effort, G = $(1)(4/3)(5/3)(4/3) = 80/27$
- Total electrical effort, F = $(5.4)/(1) = 5.4$
- Total branch factor, B = 1
- Total delay effort, F = GBH = $(80/27)(1)(5.4) = 16$
- Optimal effort delay, f_o = $F^{1/4} = 16^{1/4} = 2$
- Size of gate 4 (NAND2), C_4 = $g_4 C_{out} / f_o = (4/3)(5.4)/2 = 3.6$
- Size of gate 3 (NOR2), C_3 = $g_4 C_4 / f_o = (5/3)(3.6)/2 = 3.0$
- Size of gate 2 (NAND2), C_2 = $g_2 C_3 / f_o = (4/3)(3.0)/2 = 2.0$

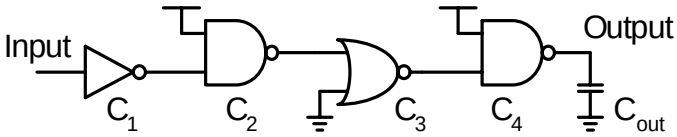

Figure 5. Example of a 4-gate path

1.4. Gate Characterization

The estimation of logical effort *g* and parasitic delay *p* are not accurate in real process technology due to several factors, such as different threshold voltages, saturation velocity, and different stack height. Therefore, these parameters need to be obtained via simulation.

Due to different gate driving and loading, two test setups are needed to characterize static gates and dynamic gates as shown in Figure 6. The importance is to load the gates to achieve approximately the same delay effort to all gates and to apply the real signal to the test gate.

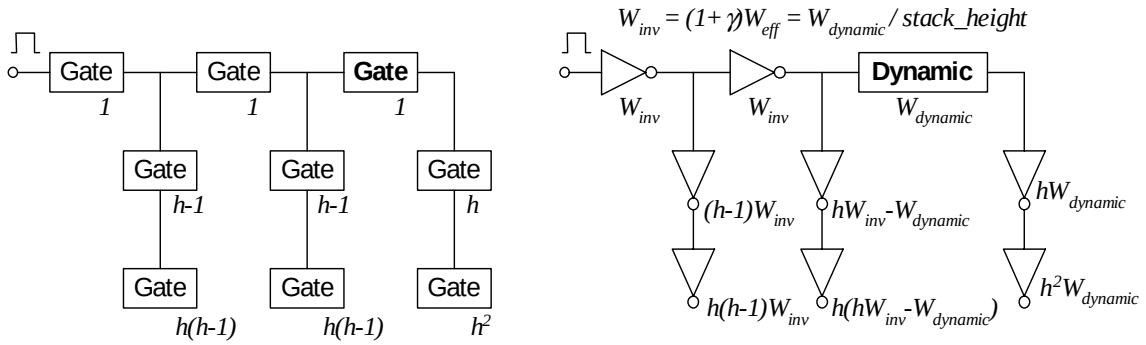

Figure 6. Test Setup for Gate Characterization

(Will need to add a table of simulated g and p for static and dynamic gates for a CMOS technology, from MOSIS for example. The 0.11um, 1.2V CMOS is used for now)

Table 3. *Characterization results of gates (*)*

	Static		***Dynamic***	
Gate	**g**	**p**	**g**	**p**
INV	1.00	1.13	0.40	1.02
NAND2	1.13	1.59	0.59	1.14
NAND3	1.26	2.49	0.77	1.24
NAND4			0.97	1.25
NOR2	1.54	2.89	0.40	1.15
NOR4			0.46	1.74
OAI	1.37	3.54	0.47	1.68
AOI	1.74	2.70	0.59	1.19

(*) τ = 9.00ps, effective PMOS/NMOS = 2.5, local wire capacitance is included

1.5. Drawbacks of Logical Effort

Logical effort method is easy to understand and apply in practice to have a quick estimation of delay and how to size the circuit. However, it is far from perfect due to its many drawbacks, as shown next.

1.5.1. Input Dependence

The most obvious drawback of logical effort model is the dependency of logical effort *g* and parasitic delay *p* on the gate input. Figure 7 illustrates that on the pull-down path of a 2-input NAND gate.

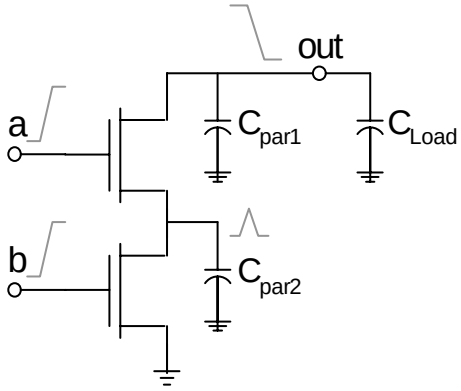

Figure 7. Pull-down path of 2-input CMOS NAND gate

First, the logical efforts *g* are not equal. The main reason lays on the R_{gate} value. By definition, R_{gate} is the average driving capability of the gate. Let us consider the 2-NMOS stack. While the bottom NMOS is always in triode region, the top NMOS first starts in saturation region before get into the triode region. Therefore, the R_{gate} value of the top NMOS will be larger than that of the bottom NMOS.

Second, the parasitic delays *p* are not equal either. In addition to the different values of R_{gate}, the parasitic capacitances C_{par} are also different. The top NMOS has to discharge less capacitance than the top one. Thus, it has less parasitic delay.

Finally, combining the effects on the logical effort and parasitic delay, it is impossible to achieve equal pull-up and pull-down paths for all inputs of the gate. Therefore, for a fixed load and input size, one input is faster than the other.

As these effects naturally exist in the gate, we cannot help but accepting them. They will cause problems in deciding which input delay will represent the delay in the gate. We suggest using the one with faster average delay. That input will be assigned to the critical path so that the delay is most aggressively targeted.

1.5.2. Optimal Delay for Multi-Path Circuits

For multi-path circuits such as adders, multipliers, etc., at each node along any path, there may exist off-path loading from other gates as shown on Figure 8. The off-path loads are represented by capacitance, C_{off}.

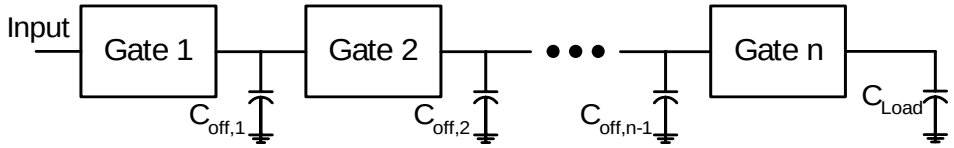

Figure 8. n-gate path with off-path loading

The total delay in the path is

$$T_{d,total} = \sum_{i=1}^{n} (g_i h_i + p_i) = \sum_{i=1}^{n} \left[g_i \left(\frac{C_{i+1} + C_{off,i+1}}{C_i} \right) + p_i \right], with\, C_{n+1} = C_{Load}$$

The optimal equations for minimal delay is found when the capacitance C_i satisfy $\partial T_{d,total}/\partial C_i = 0$ for every i from 2 to n. Then, for $i = 2 \rightarrow n$

$$\frac{\partial T_{d,total}}{\partial C_i} = g_{i-1}\frac{1+\frac{\partial C_{off,i}}{\partial C_i}}{C_{i-1}} - g_i\frac{C_{i+1}}{C_i^{\,2}} = 0$$

There are two cases to consider:

(a) $C_{off,i}$ is proportional to C_i for every i (i.e. $C_{off,i} = k_i C_i$ where k_i = constant). It happens when the circuit consists of gate loading only and gates in the same stages have the same equal stage effort delay. Then,

$$\frac{\partial T_{d,total}}{\partial C_i} = g_{i-1}\frac{1+k_i}{C_{i-1}} - g_i\frac{C_{i+1}+C_{off,i+1}}{C_i^{\,2}} = 0 \Leftrightarrow g_{i-1}\frac{(1+k_i)C_i}{C_{i-1}} = g_i\frac{C_{i+1}+C_{off,i+1}}{C_i}$$

or $\quad f_{i-1} = f_i$

(b) $C_{off,i}$ is *not* proportional to C_i for some i. It happens when the off-path load is constant or wire capacitance is included. Then,

$$C_{off,i} \neq \frac{\partial C_{off,i}}{\partial C_i}C_i$$

Therefore,

$$g_{i-1}\frac{\left(1+\frac{\partial C_{off,i}}{\partial C_i}\right)C_i}{C_{i-1}} = g_i\frac{C_{i+1}+C_{off,i+1}}{C_i} \Leftrightarrow f_{i-1} \neq f_i$$

1.5.3. Effects of Parasitic Delay Difference in Multi-Path Circuits

Let us consider a 2-path circuit as shown in Figure 9. The gate types in each path do not match. Relative to each path, the optimal delay occurs when stage effort is equal in each gate.

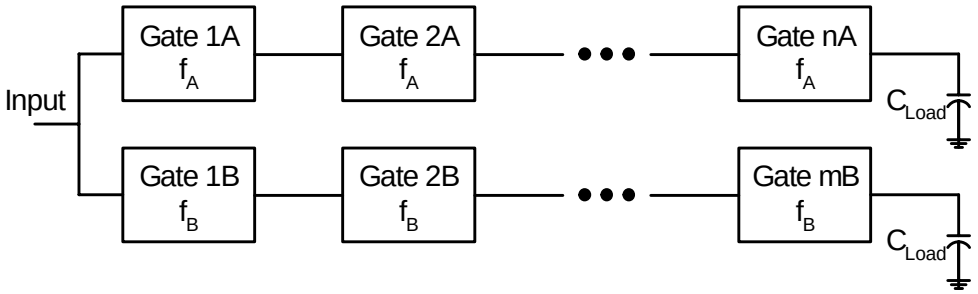

Figure 9. 2-path circuit

$$T_{d,totalA} = \sum_{i=1}^{n}\left(g_{iA}h_{iA} + p_{iA}\right) = \sum_{i=1}^{n}\left[g_{iA}\left(\frac{C_{iA+1}}{C_{iA}}\right) + p_{iA}\right] = nf_A + \sum_{i=1}^{n}p_{iA}$$

$$T_{d,totalB} = \sum_{i=1}^{m}\left(g_{iB}h_{iB} + p_{iB}\right) = \sum_{i=1}^{m}\left[g_{iB}\left(\frac{C_{iB+1}}{C_{iB}}\right) + p_{iB}\right] = mf_B + \sum_{i=1}^{m}p_{iB}$$

To guarantee the same delay in each path,

$$nf_A + \sum_{i=1}^{n}p_{iA} = mf_B + \sum_{i=1}^{m}p_{iB} \Leftrightarrow f_A = \frac{m}{n}f_B + \frac{1}{n}\left(\sum_{i=1}^{m}p_{iB} - \sum_{i=1}^{n}p_{iA}\right)$$

Note that even if the number of gates in the paths is equal, their stage efforts are not necessary equal because their total parasitic delays can be different.

1.5.4. Error due to Unmatched Signal Rates

The delay model of the logical effort method has a serious drawback: it does not include the effect of input transition. Obviously, it is not true in reality. Different input transitions will directly affect the pull-up and pull-down capability of gates and therefore the delay through the gate will be changed.

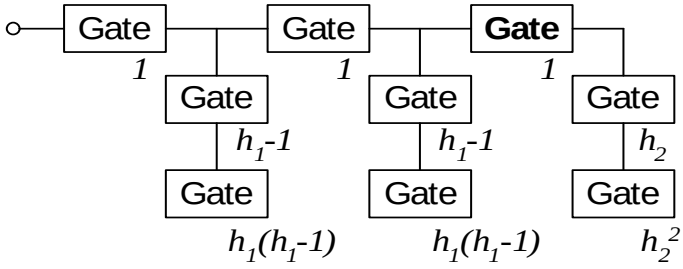

Figure 10. Modified Test Setup Circuit

The setup circuit for gate characterization can be modified to illustrate this effect, as shown on Figure 10. The simulation result for the inverter is shown on Table 3. Similar delay errors are seen for other gates.

Table 4. Inverter delay error due to signal transition

H Input	H Output	Td (tau)	Error (%)
2	4	4.46	13.9
3	4	4.75	8.3
4	4	5.18	0.0
6	4	5.59	-7.8
8	4	6.20	-19.6

1.5.5. Effects of Wire

Recent technologies have seen an increasing effect of wire on the delay. When the wire is short, it is sufficient to include its capacitance in the gate load. Figure 11 shows the RC model of the gate including the capacitance and resistance of the wire.

The estimated RC delay of the gate is

$$T_d = \kappa R_{gate}(C_{par} + 0.5C_w) + \kappa(R_{gate} + R_w)(0.5C_w + C_{out})$$

$$= \kappa R_{gate} C_{par} + \kappa R_{gate}\left(1 + \frac{R_w}{R_{gate}}\right)(C_w + C_{out})$$

When the wire is short, R_w is insignificant compared to R_{gate}. Then, the effect of wire on delay is reduced to its capacitance. However, when the wire is long (few tenths to several millimeters), R_w is comparable to R_{gate}. Its resistance must be included in delay calculation.

For 0.11μm, 1.2V CMOS technology, the wire resistance is about 0.4Ω/μm and the wire capacitance is 0.2fF/μm. For NMOS in triode region, the resistance is roughly

700Ω/μm. Therefore, as a rule of thumb, the wire length of 175μm or less is considered short for a 1-μm effective NMOS.

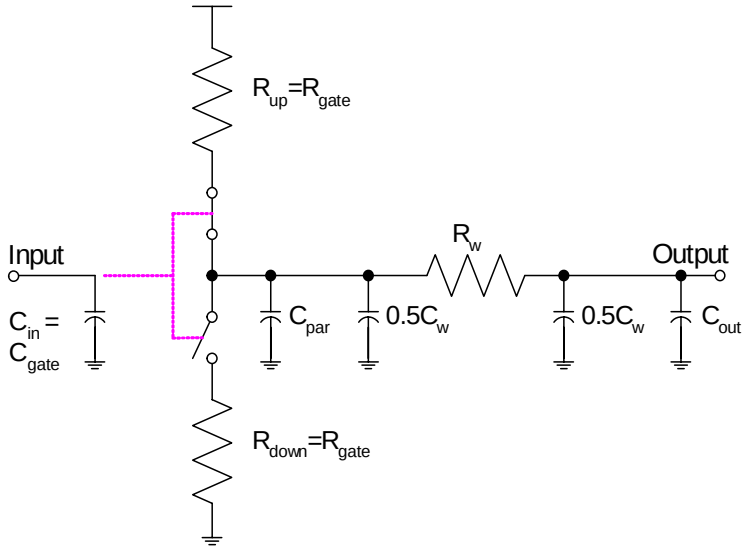

Figure 11. RC-based delay model of a gate with wire capacitance and resistance

Reference:

1. R. F. Sproull and and I. E. Sutherland, "Logical Effort: Designing for Speed on the Back of an Envelop," IEEE Advanced Research in VLSI, C. Sequin (editor), MIT Press, 1991.
2. D. Harris, R.F. Sproull, and I.E. Sutherland, "Logical Effort Designing Fast CMOS Circuits," Morgan Kaufmann Publishers, 1999.
3. T. Sakurai and A. R. Newton, "A Simple MOSFET Model for Circuit Analysis," IEEE Transaction on Electronic Devices, vol. 38, pp. 887-894, April 1991

1.5.4. Error due to Unmatched Signal Rates: *Further Elaboration*

There are two general approaches to correct the delay error caused by input transition.

i. Correct the delay model to include the input transition and modify the method accordingly to size gates;
ii. Use the current model to size gates and then apply delay correction to get more accurate result.

The first approach requires another delay term for input transition. In addition, the output transition needs to be modeled as well [3],[4]. The delay and output transition must be solved simultaneously to achieve the desired delay or energy objective. A method that sizes all gates iteratively is required. In term of big-O notation, its complexity is $O(N^P)$ where N is the total number of gates and P is a function of N. The result is an NP-hard problem. For complex circuits such as adders and multipliers of large bits, the running time will be high, especially when under energy constraint.

We prefer the second approach. First, given a delay target, the current delay model allows us to quickly size the circuit no matter of its complexity; in fact, the solution can be found in O(N) time. In addition, as shown in [5], the optimal energy solution for the given delay can be found in O(N+n) where n is the maximal number of stages in the circuits.

The problem is how to correct the delay to get more accurate value. As shown in the gate characterization, the delay terms are obtained when input and output transitions match. The delay changes when these transitions differ. As we wish not to introduce more delay terms, the delay of previous gate is used to correct the error. From simulation results, the corrected delay is modeled as

$$T_{d,i} = T_{d0,i}\left[\frac{T_{d0,i}}{T_{d0,i-1}}\right]^{\alpha}$$

where T_d = delay including variation in signal transitions
T_{d0} = delay when signal transitions are matched
α = constant, depending on gate

The term α is from –0.20 to –0.32 for static gates and from –0.15 to –0.25 for dynamic gates. Table 4B shows the inverter delay error with correction. As seen, the errors are reduced to less than 5%. However, some errors remain. The above model is only able to correct the first-order effect of signal transition; it cannot correct the error due to the secondary effects of signal transition over multiple stages.

Table 4B. Inverter delay error after correction

H Input	H Output	Td (τ)	Error (%)	Td_Corr (τ)	Error (%)
2	4	4.46	13.9	4.24	5.0
3	4	4.75	8.3	4.77	-0.3
4	4	5.18	0.0	5.18	0.0
6	4	5.59	-7.8	5.83	-4.3
8	4	6.20	-19.6	6.34	-2.2