As Is I [Book Title], Edited by [Editor's Name].
ISBN 0-471-XXXXX-X

Chapter 3

CMOS Speed: Method of "Logical Effort"

Oklobdzija, Yano

with

Contributions by Zeydel and Dao

Introduction

Digital circuits design is an art. A good digital circuit designer considers many choices and many ways to solve a particular problem. However, often none of the choices is obvious, neither they do not guarantee the best solution. As a matter of fact, the best solution may not even be known and designer can only hope to come close to what he/she considers to be the best. As in the art creation process, there are many attempts and trials guided by designer's intuition and often inspiration, before the final design stage.

Until recently, the main objective in designing a digital circuit was the maximal speed, which was determined by the critical path of the design. Other factors, such as power, were secondary or of negligible importance. The Computer Aided Design (CAD) tools, that help the design process, were developed mainly around the critical path optimization, which was the most critical and the most difficult task. However, even with that attention given the speed of the digital circuit could not have been determined until the final design stage was reached. Only at that point, simulation results were to produce the timing information which was to be judged acceptable or unacceptable. The apparent difficulty of such design process is that the result will not be known until the design is nearly completed, so that making changes at this point would be costly in terms of design time. This would usually limit the number of attempts toward achieving the best design, or even make a design a "one-shot" approach, often the case when working under the strict deadlines.

On the other hand, development of hardware oriented computer algorithms requires some input from the technology in terms of its most important characteristics: speed and power. To be able to judge the effectiveness of an algorithm in terms of computational speed, some way of estimating the digital logic speed was necessary. Historically, various estimates for the speed of the digital logic were used, but mostly the speed of the algorithm was judged by the number of logic stages (logic levels) needed for its realization in hardware. In the early days such estimates were adequate for that purpose because the logic was built from discrete parts, mostly dominated by the pin I/O speed, thus a number of discrete logic gates in the critical path did reflect the maximal achievable speed of the implementation. However, with the transition to CMOS and large scale of integration, such measures started to exhibit their shortcomings. The speed was not only dependent on the number of the logic elements in the critical path, but also on various other factors, most importantly, fan-out and fan-in of the circuit, representing the output load and the number of transistors in the serial path, respectively. The other factors such as the wire delay and wire loading were impacting delay to the smaller degree and were often neglected.

In 1991 Sproull and Sutherland published the paper in which they established a method for simple delay estimation and transistor sizing on the signal path in a digital circuit. They called it the method of “logical effort”. The method brings the following important contributions to the art of circuit design:

(a) Provides a quick way of estimating delay of a path.

(b) Provides a simple way of estimating optimal transistor sizes with the objective of minimizing delay on the path.

(c) Gives the estimate for the optimal number of stages in the path with respect to minimizing delay.

(d) Provides a technology independent measure for delay, thus enabling to compare the effectiveness of designs implemented across different technologies.

Analysis of an invertor delay

Every technology specifies its minimal size which represents the limit of their lithography and ability to fabricate. Thus, a particular technology a number of minimal sizes related to the fabrication of the transistor, among them the channel length L, as well as some minimal width W for n and p transistors. Let us assume that we have an inverter built from the minimal size transistors, but with the width W_p of the p transistor adjusted in size so that the pull-up and pull-down currents of those two transistors are the same, establishing equal transition times: t_r and t_f. Consider now making both transistors proportionally larger for a factor α, in effect creating and invertor that is α times larger than the one we can make without going beyond the minimal allowed size of the transistors. This situation is depicted in Fig.1.

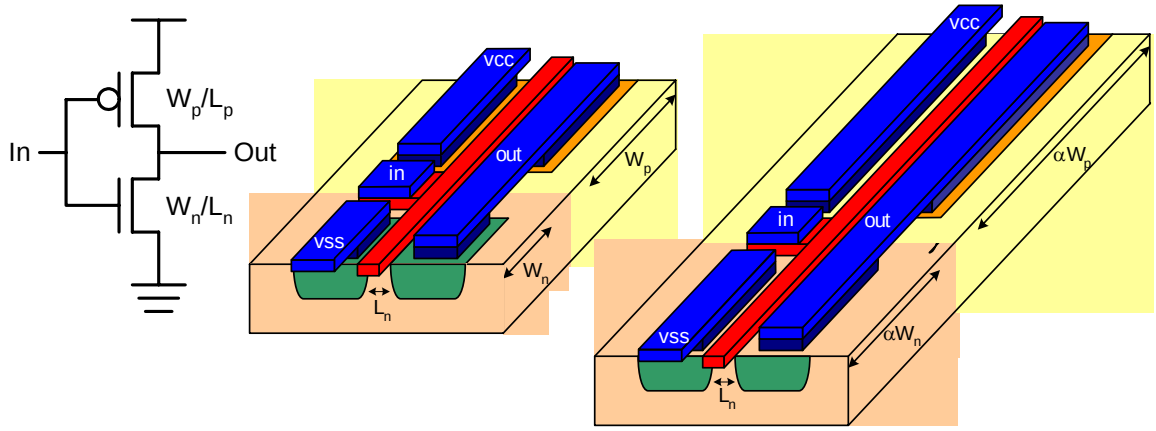


Fig. 1. Basic CMOS inverter (a) inverter template, minimal size (b) inverter sized to scale

In this section we want to show the physical meaning of the Logical Effort. First we will show that the Logical Effort is derived from a basic *RC* equation.

$$t_d = \kappa R\left(C_{out} + C_p\right) \tag{1}$$

Where, C_{out} represents all of the capacitive load at the output that is external to the gate (no diffusion cap) and R represents the channel resistance of the transistor. While, C_p represents the parasitic capacitance seen by the inverter. The constant κ is dependent on the technology parameters and will be derived later in this text. The model is useful for describing a single inverter, however it is difficult to apply the model directly to a path or even a single gate of a different size. To further the analysis we should examine how those parameters change with changing of the inverter size.

Size scaling

Using the template shown in Fig. 1, we examine how does increasing the size from the minimal invertor affects the electrical parameters such as pull-up and pull-down resistance, input and parasitic capacitance. We consider changing the size proportionally by a factor α.

Input Capacitance

Input capacitance of a transistor is dominated by its gate to channel capacitance C_g shown in Fig. 2. This capacitance can be expressed as:

$$Cg = \varepsilon_{ox}\frac{S}{t_{ox}} = c_{ox}WL_{eff} = c_{ox}WL(1-2x_d/L) \tag{2}$$

We define oxide surface capacitance $c_{ox} = \frac{\varepsilon_{ox}}{t_{ox}}$ expressed as the capacitance of the gate per surface area of the gate expressed in F/μ^2. Due to the under-etching the effective length of the channel is different from the so called, drawn length L_{eff}. As seen from the Fig.2: $L_{eff} = L - 2x_d$ where, x_d represent the distance of under-etching.

We define the constant k_1 which is a parameter of the given technology, so that we can express the input capacitance C_g as:

$$Cg = k_1 WL \quad \text{where } k_1 = c_{ox}(1 - 2x_d / L) \tag{3}$$

The input capacitance of the inverter is then:

$$C_{in} = k_1(W_p L_p + W_n L_n) = \alpha C_t \tag{4}$$

where C_t is the input capacitance of the template inverter. The template inverter could be representing a minimal size inverter that can be realized in a given technology, or any other size chosen for the template.

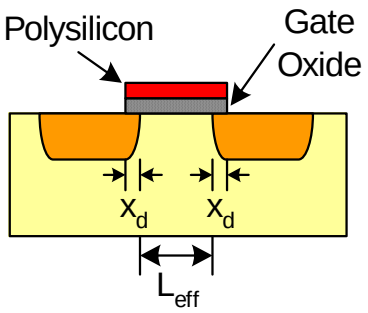
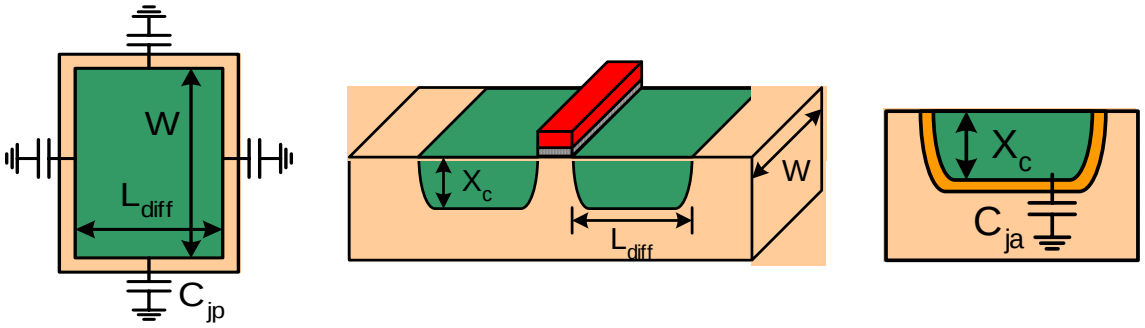


Fig. 2. Gate capacitance C_g

Parasitic Capacitance

In addition to the input capacitance seen by the stage driving the input, there is capacitance present at the output even when the output is not connected to any load. This capacitance comes from the intrinsic capacitance of the inverter and it consists of the parasitic capacitance components associates with the *n* and *p* transistors. This capacitance is illustrated in Fig. 3. and described further in this section.



(a) (b) (c)

Fig. 3. Components of the parasitic capacitance associated with the transistor

There are two basic components of the parasitic capacitance:

c_{ja} junction capacitance expressed in F per area in μ^2 (Fig.3.c.)

c_{jp} periphery capacitance per μ of the peripheral length (Fig.3.a)

$$C_p = c_{ja} \cdot W \cdot L_{diff} + c_{jp} \cdot (2 \cdot W + 2 \cdot L_{diff}) \quad (5)$$

Where W is the width of diffusion and L_{diff} is the effective length of the diffusion region. Thus the parasitic capacitance C_p is the function of the channel width of the transistor and it grows proportionally with the scaling factor α:

$$C_p = \alpha C_{pt} \quad (6)$$

Channel Resistance

The channel resistance of a transistor can be expresses as: $R = \rho\ (L_{eff} / t\ W)$ as shown in Fig. 4.

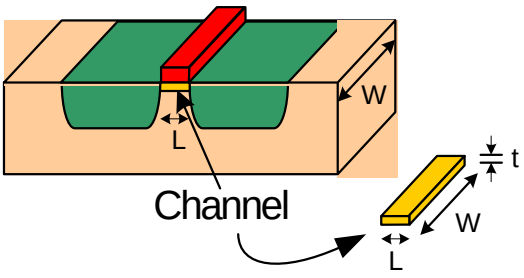


Fig. 4. Transistor channel resistance when conducting

Where: t is the thickness (depth) of the channel, L_{eff} is the effective length of the channel and W is the width of the channel.

For the specific thickness (depth) of the channel, which is determined by the technology, we can define the resistance per square $r=\rho/t$, also called *sheet resistance* because the resistance of the channel is dependent only of its geometry:

$$R = r\frac{L_{eff}}{W} \quad \text{where } \textit{sheet resistance} \text{ is } \quad r = \frac{\rho}{t} \tag{7}$$

From the transistor physics the sheet resistance is given by:

$$r = \frac{1}{\mu_n C_{ox}(V_{GS} - V_T)} \tag{8}$$

The expression for the channel resistance is therefore given by:

$$R_{channel} = \frac{1}{\mu_n C_{ox}(V_{GS} - V_T)\dfrac{W}{L}} \tag{9}$$

Since the effective channel length L_{eff} is fixed (by technology) the equation describing the channel resistance changes with the scaling factor α as:

$$R = r\frac{L_{eff}}{W} = r\frac{L_{eff}}{\alpha W_t} = \frac{R_t}{\alpha} \tag{10}$$

Transition delay

In this section we will derive an expression for the transition delay, assuming that the *p* and *n* transistors are sized such that the pull-up and pull-down resistance is the same, resulting in an equal rise t_r and fall t_f transition times. Assume that the output capacitance *C* is fully charged and that the discharge will take place through the *n* transistor which is fully ON, Fig. 5.

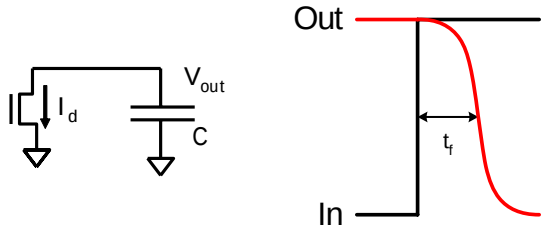


Fig. 5. Transition delay

When discharging the capacitor *C*, the transistor starts in the saturation region when the transistor can be modeled as a constant current source, and transits into a linear region. The expression for the transistor current in the saturation and linear region is given by equations (9) and (10) respectively:

$$I_{d(sat)} = \mu_n \cdot C_{ox} \cdot \frac{W}{L} \frac{(V_{GS} - V_t)^2}{2} \tag{11}$$

$$I_{d(lin)} = \mu_n \cdot C_{ox} \cdot \frac{W}{L} \left((V_{GS} - V_t)^2 \cdot V_D - \frac{V_D{}^2}{2} \right) \tag{12}$$

The expression for the fall time tf can be obtained by solving a simple first order differential equation given by:

$$-I_d = C \frac{dV_{out}}{dt} \tag{13}$$

Substituting (11) and (12) into (13) leads to the following integral:

$$t_f = -\int_{Vdd}^{V_{dd}-V_t} \frac{C}{I_{d(sat)}} dV - \int_{V_{dd}-V_t}^{V_{dd}/2} \frac{C}{I_{d(lin)}} dV \tag{14}$$

leading to:

$$t_f = -\int_{Vdd}^{V_{dd}-V_t} \frac{C}{I_{d(sat)}} dV - \int_{V_{dd}-V_t}^{V_{dd}/2} \frac{C}{I_{d(lin)}} dV - \mu_n \cdot C_{ox} \cdot \frac{W}{L} \int_{Vdd-Vt}^{Vdd/2} \left((V_{DD} - V_t)^2 \cdot V_{OUT} - \frac{V_{OUT}{}^2}{2} \right) dv \tag{15}$$

and providing the final solution for t_f:

$$t_f = \frac{C}{\mu_n \cdot C_{ox} \cdot \frac{W}{L} (V_{DD} - V_t)} \left(\frac{2 \cdot V_t}{V_{DD} - V_t} + \ln \cdot \frac{4 \cdot (V_{DD} - V_t) - V_{DD}}{V_{DD}} \right) \tag{16}$$

The channel resistance, when the transistor is on and V_{GS}=V_{DD} is given by:

$$R_{channel} = \frac{1}{\mu_n C_{ox} (V_{DD} - V_T) \frac{W}{L}} \tag{17}$$

Substituting (17) into (16) we obtain:

$$t_f = RC \left(\frac{2 \cdot V_t}{V_{DD} - V_T} + \ln(3 - 4 \frac{V_T}{V_{DD}}) \right) = \kappa RC \tag{18}$$

where:

$$\kappa = \left(\frac{2 \cdot V_T}{V_{DD} - V_T} + \ln(3 - 4\frac{V_T}{V_{DD}} \right) \tag{19}$$

Logical Effort of an Inverter

Starting with the delay equation (1) and substituting (6) and (10) into (1) we obtain:

$$t_d = \kappa \frac{R_t}{\alpha} \left(C_{out} + \alpha C_{pt} \right) \tag{20}$$

where C_t , R_t , and C_{pt} are representing values of the template size of the gate.

Performing the following modification removes the scaling factor α and introduces C_{in} which is the value that can be easily determined:

$$t_d = \kappa \frac{R_t}{\alpha} \alpha C_t \frac{C_{out}}{\alpha C_t} + \kappa R_t C_{pt} \tag{21}$$

Substituting $C_{in} = \alpha C_t$ we obtain:

$$t_d = \kappa R_t C_t \frac{C_{out}}{C_{in}} + \kappa R_t C_{pt} \tag{22}$$

This delay model now contains elements that can be obtained from a single simulation, and applied across a wide range of C_{out} and C_{in}. This is shown in Fig. 6.

$$t_d = s_0 \frac{C_{out}}{C_{in}} + s_1 \tag{23}$$

where: $$s_0 = \kappa R_t C_t \text{ and } s_1 = \kappa R_t C_{pt} \tag{24}$$

From (22) we obtain:

$$t_d = \kappa \frac{R_{inv} C_{inv}}{R_{inv} C_{inv}} \left(R_t C_t \frac{C_{out}}{C_{in}} + R_t C_{pt} \right) \tag{25}$$

$$t_d = \kappa R_{inv} C_{inv} \left(\frac{R_t C_t}{R_{inv} C_{inv}} \frac{C_{out}}{C_{in}} + \frac{R_t C_{pt}}{R_{inv} C_{inv}} \right) \tag{26}$$

The equation (26) defines the following constants:

$$\tau = \kappa R_{inv} C_{inv} = \kappa R_t C_t = s_0 \tag{27}$$

$$g = \frac{R_t C_t}{R_{inv} C_{inv}} = 1 \tag{28}$$

$$p = \frac{R_t C_{pt}}{R_{inv} C_{inv}} = \frac{R_t C_{pt}}{R_t C_t} = \frac{C_{pt}}{C_t} = \frac{s_1}{s_0} \tag{29}$$

The variable g is defined as the *logical effort* of the invertor which is, by definition equal to one.

The variable p is the contribution of the parasitic capacitance to the inverter delay and its value is close to one, which is often used. Simulation results would produce an exact value (29).

The variable h is defined as:

$$h = \frac{C_{out}}{C_{in}} \tag{30}$$

This is resulting in the following delay equation:

$$t_d = \tau(gh + p) \tag{31}$$

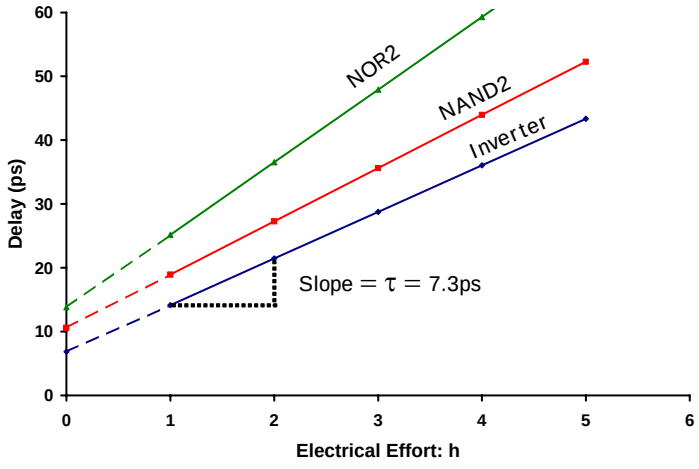


Fig.6. Simulation results determining the values for g, p and τ

Gate Delay Modeling

The logical effort method [1, 2] proposes a simple and easy-to-use delay model for the logic gate. As shown in Fig. 7, the model is *RC*-based: resistance R_{gate} represents the pull-up and pull-down capabilities of a gate, whereas the load at the output consists of the capacitance of the external load and the internal parasitic capacitance of the gate.

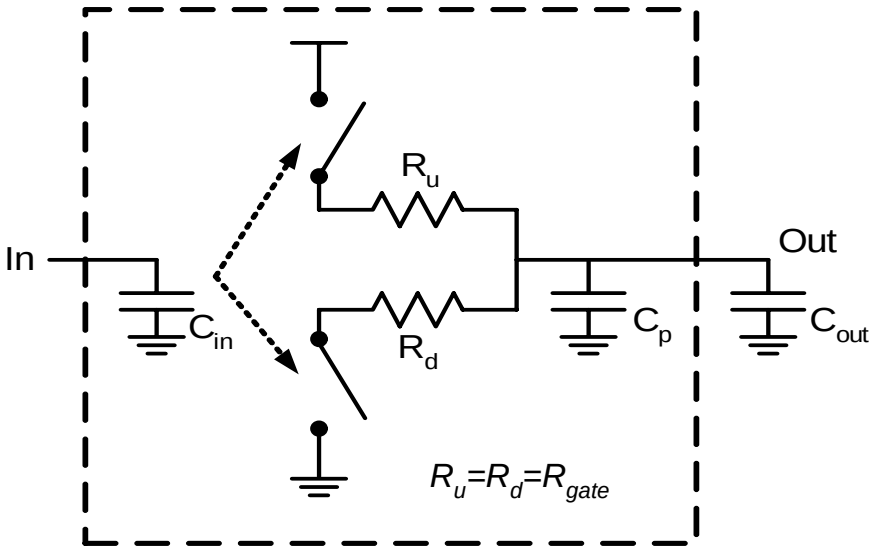


Fig. 7. RC delay model of a gate

The delay of the gate is proportional to the *RC* constant, equations (1, 22) where R_{inv} is replaced by R_{gate} under the assumption that the pull-up and pull-down resistances of both: the *p* and the *n* transistor switching networks are the same: $R_{gate}= R_{up} = R_{down}$. This assumes that the gate is design for equal rise and fall times of the signal at the output. The reader should take a note that this may not be universally true, given that, the ratio of pull-up R_{up} and pull-down R_{down} resistances is sometimes adjusted for the reasons of achieving speed.

$$t_d = \kappa R_{gate}(C_{out} + C_p) = \left(\kappa R_{gate} C_{in}\right)\left(\frac{C_{out}}{C_{in}}\right) + \left(\kappa R_{gate} C_p\right) \tag{32}$$

The constant κ is determined from the parameters of a particular process technology as shown in (19). Notice that similarly to inverter case the components $R_{gate}C_{in}$ and $R_{gate}C_p$ remain approximately constant as the size of the gate changes proportionally for a factor α. Their value is determined by the gate topology and process technology only. In other words, they characterize the delay behavior of the gate. We can find them by simulation (constants: s_0 and s_1) for a particular gate, Fig. 6.

In addition, the relative ratios of $R_{gate}C_{gate}$ and $R_{gate}C_p$ are not significantly different for different technologies. Therefore, the delay of a gate could be conveniently

normalized to a unit delay $\tau = \kappa R_{inv} C_{inv}$ to achieve a technology independent expression for delay:

$$t_{d,norm} = \left(\frac{\kappa R_{gate} C_{gate}}{\kappa R_{inv} C_{inv}} \right) \left(\frac{C_{out}}{C_{gate}} \right) + \left(\frac{\kappa R_{gate} C_{par}}{\kappa R_{inv} C_{inv}} \right) \tag{33}$$

or, in shorter notation,

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

where: $g = \dfrac{R_{gate} C_{gate}}{R_{inv} C_{inv}}$ is defined as the *logical effort* of the gate, representing the sensitivity of the gate to the external load. In Fig. 7. this is the slope of the delay versus electrical effort..

$h = \dfrac{C_{out}}{C_{gate}}$ is the *electrical effort*, representing external load capacitance normalized to the gate input capacitance.

$f = gh$ is defined as *stage effort*, and it is equivalent to the gate delay for a given load normalized to the invertor delay.

$p = \dfrac{R_{gate} C_{par}}{R_{inv} C_{inv}}$ is a parasitic delay of the gate normalized to the invertor delay.

The graphical representation of the delay components is illustrated in Figure 7.

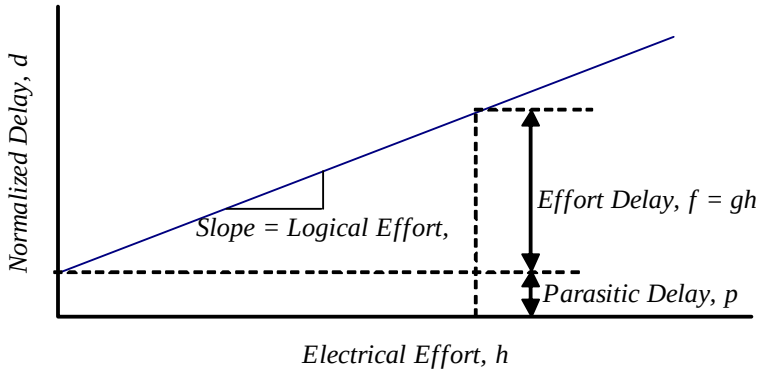


Fig. 8. Delay Components of Logical Effort Model

Estimation of *g* and *p* Terms

As mentioned before, the logical effort *g* and the parasitic delay *p* are constant for a gate implemented in a given technology. Their value is determined by the topology of the gate. In this case we will derive the expression for the logical effort g allowing that the *p/n* transistor size ratio changes, thus, without assuming equal transition times: t_r and t_f.

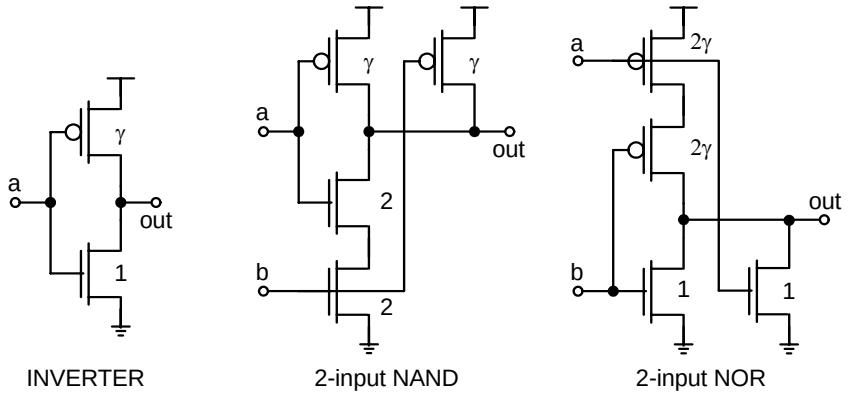


Fig. 9. Basic CMOS gates

Fig. 9. shows the schematics of basic CMOS gates, where γ is the 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 capacitance (size) ratio of the gate relative to the inverter. The estimated *g* values of representative CMOS gates are listed in Table 1.

Table 1. Estimated logical effort *g*

Gates	***Logical Effort, g***	***For*** γ=2
INV	1	1
n-input NAND	(γ + n)/(γ + 1)	(2+n)/(2+1)
n-input NOR	(nγ + 1)/(γ + 1)	(2n+1)/(2+1)
2-input XOR	4	4

Similarly, the estimated parasitic delay of an equal driving capability gate is taken relative to that of the inverter. Table 2 shows a 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	$4p_{inv}$

References

[1] R. F. Sproull, I. E. Sutherland,"*Logical Effort: Designing for Speed on the Back of an Envelope*",IEEE Advanced Research in VLSI, C. Sequin, ed., 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.
[4] B. S. Amrutur, M. A. Horowitz,"*Fast Low-Power Decoders for RAMs*",IEEE Journal of Solid-State Circuits, Vol. 36, No. 10, October 2001.