Chapter 4: Delay of CMOS logic gates

The overall aim of this chapter is to derive a simple method for calculating propagation delay of CMOS logic gates; a simple model similar to the two-port model for calculating the inverter *RC* delay. In this lecture, all delay calculations will be made relative to the τ=5 ps *RC*-constant of the ideal inverter in terms of the gate parasitic and fanout delays, *p* and *f*, respectively.

$$\text{gate delay} = \tau \times (p + f). \tag{4.1}$$

During the lecture, we will learn how to calculate the parasitic delay of logic gates, and how to calculate the fanout delay. We will also learn how to calculate the fanout delay that is a bit more complicated for logic gates than for inverters due to their more complex topology. To handle this, a new concept will be introduced here, namely the logical effort, *g*, and the fanout will be shown to be given by the product of the gate electrical effort and the gate logical effort,

$$\text{fanout delay } f = g \times h. \tag{4.2}$$

As before, the electrical effort is given by the number of loading identical gates, or equivalently, by the capacitance ratio C_L/C_{IN}, where C_L is the external load capacitance and C_{IN} is the input capacitance of the gate. A NOR gate loaded by three identical NOR gates have an electrical effort of three, and so does a NAND gate loaded by three identical NAND gates.

The first step in the process of estimating the gate delay is to use the same type of two-port model for describing the electrical behavior of the logic gate as we did for the inverter. We will start by determining the three model parameters, C_{IN}, C_D and R_{ref}, for the most common logic gates from the MOSFET schematic of the gate. As for the inverter, we will assume that the two *RC* products, i.e. $R_{ref}C_{IN}$ and $R_{ref}C_D$, are constants when gates are scaled. If we increase the driving capability of a gate by making the MOSFETs wider, the capacitances will of course increase but the effective resistance will decrease accordingly. The logic gate two-port model is shown in Fig. 4.2 together with the two-port model of an ideal inverter (no parasitics) that will serve as our reference.

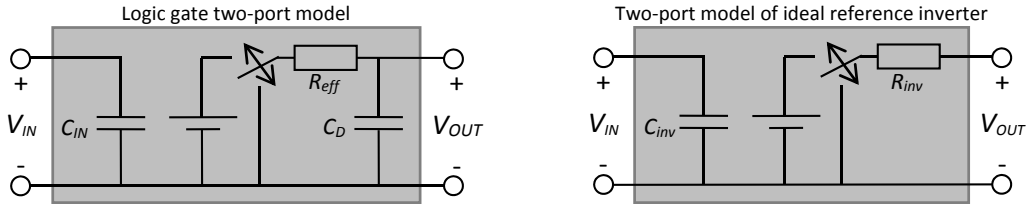

Fig. 4.1. Two-port model of a logic gate together that of the reference inverter without parasitics.

The next step is to use this figure to define the parasitic delay and the logical effort of the logic gate,

$$\text{logical effort } g = \frac{R_{eff}C_{IN}}{R_{inv}C_{inv}},\ \text{parasitic delay } p = \frac{R_{eff}C_D}{R_{inv}C_{inv}}. \tag{4.3}$$

In this process, we will find that the RC product of a logic gate is larger than that of an inverter. This is due to the increased complexity of the gate topology, and the larger number of MOSFETs needed to implement the logic function. We will soon illustrate this with some gate examples.

However, let us first relate the logical effort to the delay calculation by a simple example of a logical gate driving another logical gate, and we want to estimate the input to input propagation delay. This is

illustrated in Fig. 4.2, where we have replaced the two logic gates by their two-port models. The result of this operation is an RC circuit from which we can estimate the propagation delay.

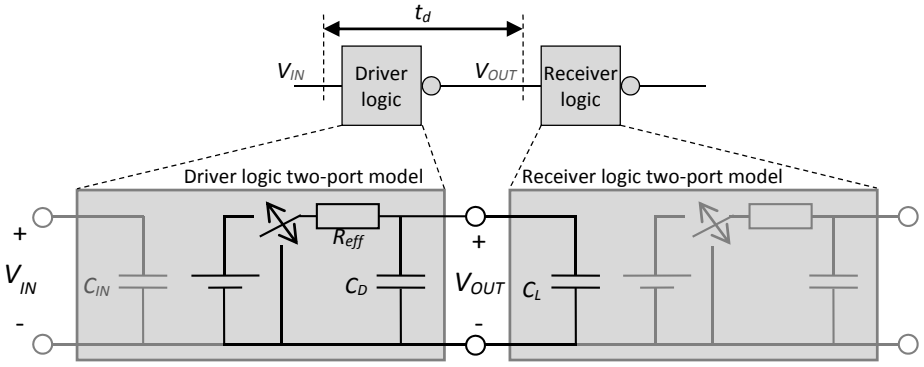

Fig. 4.2. Replacing the driver and receiver logic gates by their two-pole electrical RC models.

As for the inverter in the previous chapter, the propagation delay can be written

$$t_{pd} = 0.7R_{eff}\left(C_D + C_G\right). \tag{4.4}$$

Now, let us normalize the delay with respect to the delay of the ideal reference inverter loaded by another identical reference inverter, i.e. divide the gate delay by $R_{inv}C_{inv}$,

$$t_{pd} = 0.7R_{eff}\left(C_D + C_G\right) = 0.7R_{inv}C_{inv}\left(\frac{R_{eff}C_D}{R_{inv}C_{inv}} + \frac{R_{eff}C_L}{R_{inv}C_{inv}}\right) = \tau \times \left(p + f\right), \tag{4.5}$$

where τ=5 ps for the 65 nm process. In this equation, we immediately identify the parasitic delay p that we already defined in (4.3), and by rewriting the fanout,

$$f = \frac{R_{eff}C_L}{R_{inv}C_{inv}} = \frac{R_{eff}C_{IN}}{R_{inv}C_{inv}} \times \frac{C_L}{C_{IN}} = gh, \tag{4.6}$$

we can also identify the logical effort previously defined in (4.3) and the electrical effort $h = C_L/C_{IN}$. As before, the electrical effort is simply the effective number of loading gates. The logical effort, on the other hand, is a measure of how much larger the fanout will be when a logical gate is driving a certain load compared to when an inverter is driving the same load.

It may take some time to comprehend the logical effort concept, but once this is done, and we have collected a table of logical efforts and parasitic delays for some of the most common logical gates, we have at hand a very convenient model for estimating propagation delays. The most convenient property of the relative delay model,

$$d = p + f = p + gh, \tag{4.7}$$

is that it can be considered technology independent, because both p and g turn out to be more or less technology independent. What changes between technologies is the time constant, τ, the scaling of which can be more or less predicted for future technologies. This means that delay estimations are portable, and that they can be easily transferred from one technology node to a more advanced technology node. Another advantage of the relative delay model is that it reduces delay calculations to simple manipulations of integer and rational numbers. We do not need to calculate the effective resistances and capacitances for every size of a logical gate, it is enough to know how much larger the RC products are compared to the RC product of a reference inverter.

Calculating logical effort and parasitic delay

Now, let us start getting used to the new delay model by calculating the parasitic delay, *p,* and the logic effort, *g,* for some of the most common logic gates. Our first example will be the tri-state inverter shown together with the reference inverter in Fig. 4.3. The inverter MOSFETs have been sized for equal effective pull-up and pull-down resistances, i.e. the p-channel MOSFET is twice as wide as the n-channel MOSFET. The same is true for the tri-state inverter in Fig. 4.3b that has been designed with the same size MOSFETs as the inverter. This means that the tri-state inverter has the same input capacitance and the same parasitic capacitance as the reference inverter, 3C. However, since the tri-state inverter pull-up and pull-down networks have two MOSFETs in series, its effective resistance is twice that of the inverter. Hence, the tri-state inverter RC products are twice those of the inverter. Hence, the logical effort of the tri-state inverter is g=2, and its parasitic delay is p=2p_{inv}. Usually, we assume p_{inv}=1, and if so, p=2. In Fig. 3c, the tri-state inverter has been resized with wider MOSFETs to double its driving capability. Now, the tri-state inverter has the same effective resistance as the reference inverter. However, at the same time its input and parasitic capacitances have doubled, so the logical effort and the parasitic delay are still twice those of the reference inverter.

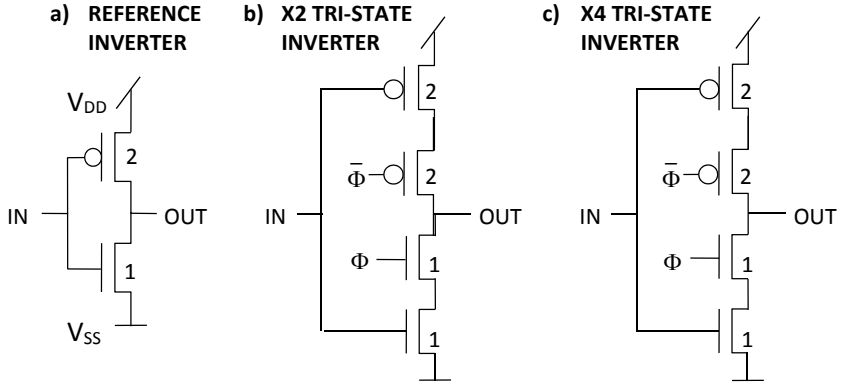

Fig. 4.3. MOSFET sizing for logical effort calculations of some logical gates.

The two-port models for the X2 and X4 tri-state inverters are shown in Fig. 4.4.

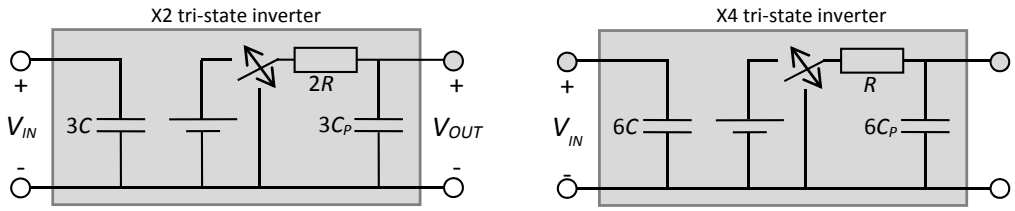

Fig. 4.4. Two-port model of the two tri-state inverters.

Now, we are ready to look at some more examples, as the logic gates shown in Fig. 4.5. Here, the AOI22 gate is maybe the simplest, since it reminds of the tri-state inverter from Fig. 4.3c. Its logical effort is 2, but the parasitic delay is now 4 due to all the parasitic drain capacitances connected to the output (12C_P). The NAND2 and NOR2 gates have different number of series MOSFETs in their pull-up and pull-down paths, but still all MOSFETs have been sized for the same effective resistance, R, in all paths. For the NAND2 gate the input capacitance is 4C for all inputs, while for the NOR2 gate it is 5C. This means that the logical efforts are 4/3 and 5/3, respectively. Both gates have the same parasitic capacitance, 6CP, rendering the parasitic delay a value twice that of the reference inverter, p=2p_{inv}.

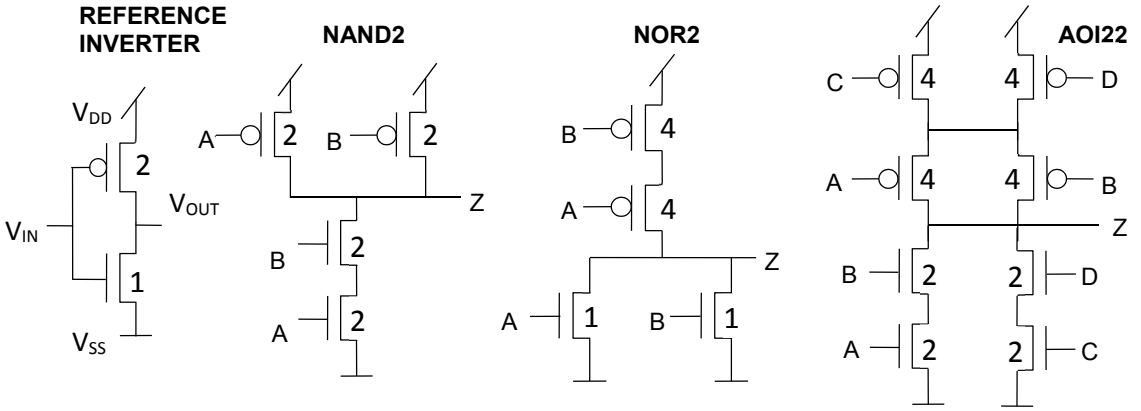

Fig. 4.5. MOSFET sizing for logical effort calculations of some logical gates.

The logical efforts and parasitic delays of these gates are summarized in this table. Also, the parasitic delay and the logical effort of the two worst-case inputs of an AOI21 gates are given for you to check. By the way, what would the logical effort of the third input be?

	parasitic delay	logical effort
NAND2	2	4/3
NOR2	2	5/3
AOI22	4	2
AOI21	7/3	2

Applying the delay model

Having gone through how to calculate the parasitic delay and the logical effort of some basic logic gates, we are now ready to apply the delay model to calculate, for instance, the NAND2 and NOR2 FO3 delays. These situations are illustrated in Fig. 4.6. The two FO3 delays are given by the following relationships:

$$\begin{cases} \text{NAND2 FO3 delay} = \tau \times (p + gh) = \left(2 + \frac{4}{3} \times 3\right)\tau = 6 \times 5 \text{ ps} = 30 \text{ ps} \\ \text{NOR2 FO3 delay} = \tau \times (p + gh) = \left(2 + \frac{5}{3} \times 3\right)\tau = 7 \times 5 \text{ ps} = 35 \text{ ps} \end{cases}, \tag{4.8}$$

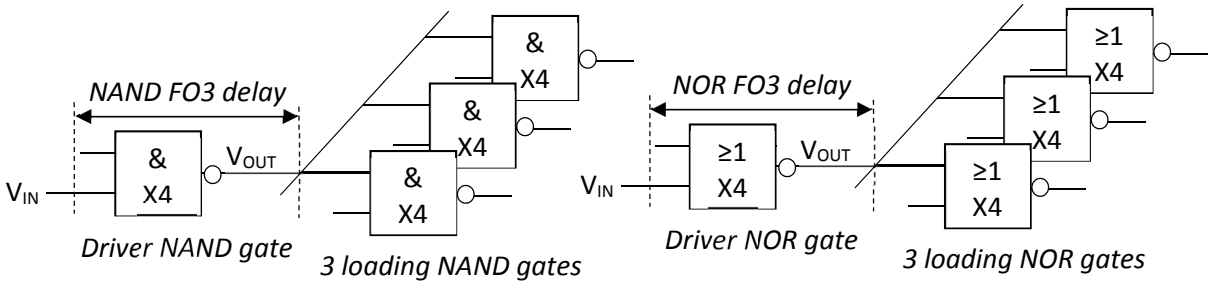

Fig. 4.6. Illustrations of the NAND2 and NOR2 fanout-of-3 (FO3) situations.

Two other FO3 situations are illustrated in Fig. 4.7. Here, the FO3 delays are the same as before, but now the electrical effort is given by

$$h = \frac{C_L}{C_G} = \frac{x_{load}}{x_{driver}} = \frac{12}{4} = 3, \tag{4.9}$$

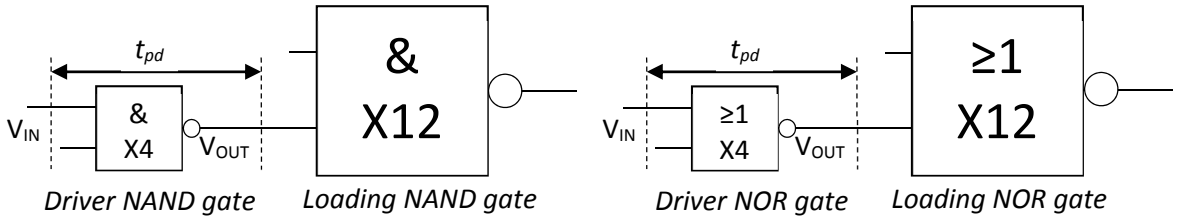

Fig. 4.7. Different illustration of the fanout-of-3 (FO3) situations.

where x_{driver} and x_{load} are measures of the driving capabilities, or sizes, of the driver gate and the loading gate, respectively. Another situation is shown in Fig. 4.8 where an X8 NAND gate is loaded by three other logic gates of different sizes. In this situation the propagation delay is given by

$$d = p_{NAND} + g_{NAND} \times \frac{9g_{NOR} + 8g_{AOI} + 12g_{NAND}}{9g_{NAND}} = 2 + \frac{4}{3} \times \frac{9 \times \frac{5}{3} + 8 \times 2 + 12 \times \frac{4}{3}}{9 \times \frac{4}{3}} \approx 7, \tag{4.10}$$

where the general expression for the electrical effort is given by

$$h = \frac{C_L}{C_G} = \frac{\sum_1^n g_i x_i}{(gx)_{input}}, \tag{4.11}$$

where g_i and x_i are the logical effort and driving capability, respectively, of the loading gate, i, and where $(gx)_{input}$ is the input capacitance of the driver gate relative to a unit size reference inverter. Now, let us conclude this section by two more examples.

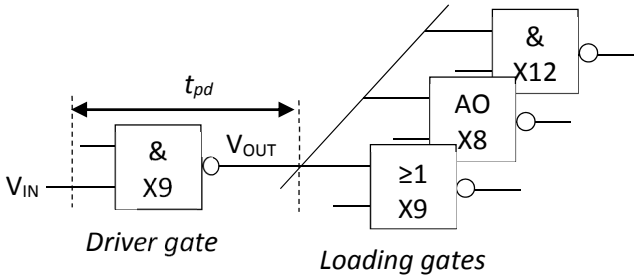

Fig. 4.8. Calculating the propagation delay of a general load situation.

Example 4.3: Calculate the path delay shown in Fig. 4.9.
Solution: The relative path delay is obtained by adding the three relative stage delays,

$$\text{relative path delay } d = d_1 + d_2 + d_3 = \sum_{k=1}^{3} d_k = \sum_{k=1}^{3} (p_k + f_k) = \sum_{k=1}^{3} p_k + \sum_{k=1}^{3} g_k h_k.$$

By inserting values for logical efforts, g_k, from the previous table, and electrical efforts, h_k, from the figure, we obtain

$$\text{relative path delay } d = \left[\underbrace{\overbrace{2}^{p_1} + \overbrace{\frac{4}{3}}^{g_1} \cdot 3 \cdot \overbrace{\frac{8\cdot 5/3}{8\cdot 4/3}}^{h_1}}_{d_1} + \underbrace{2 + \frac{5}{3} \cdot \frac{12\cdot 4/3}{8\cdot 5/3}}_{d_2} + \underbrace{2 + \frac{4}{3} \cdot \frac{48}{12\cdot 4/3}}_{d_3} \right] = 25 \Rightarrow t_{pd} = 125 \text{ ps}. \blacksquare$$

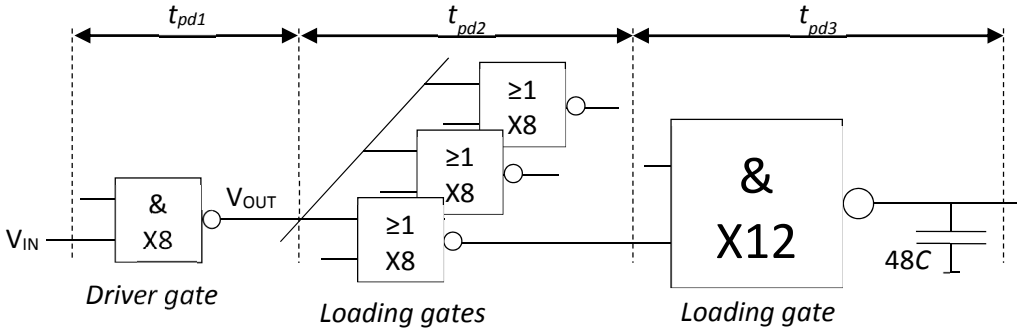

Fig. 4.9. Calculating the path delay.

Example 4.4: Non-inverting logical gates are built by adding an inverter at the output of the corresponding inverting gate. An example of an AND2 gate consisting of an X4 NAND2 gate and and an X8 inverter is shown in Fig. 4.10. Calculate the parasitic delay and the logical effort of such a 2-input AND gate.

Solution: As before, the relative AND2 delay can be found by adding the two stage delays,

$$d = d_{NAND} + d_{inv} = \underbrace{2 + \frac{4}{3}\cdot\frac{8}{4}}_{d_{NAND}} + \underbrace{1 + \frac{C_L}{8C}}_{d_{inv}} \approx 6 + 0.5\frac{C_L}{4C}. \qquad (4.12)$$

From expression we can draw the conclusion that the parasitic delay of the AND2 gate is almost 6, and that its logical effort is 0.5. This is because effective resistance of the X8 output driver of the AND gate is only half of what is expected from a logical gate with a 4X input capacitance. ■

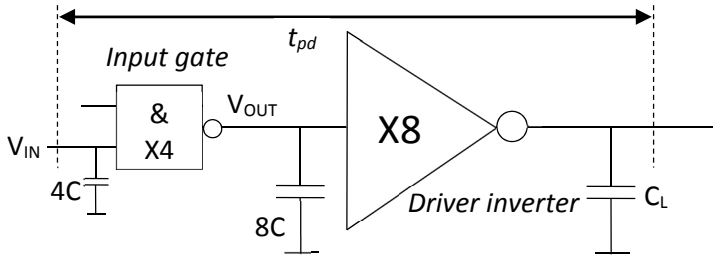

Fig. 4.10. Delay calculation for a non-inverting NAND gate.

Because of their simplicity, relative delay calculations are very convenient for getting quick back-of-the-envelope estimates before turning to the EDA tools. Standard cell library databooks usually list the intrinsic delays, the effective resistances, and the input capacitances for the different driving capabilities that a library cell is available. See table below for a 2-input AND gate example. These values are usually specified for a certain input rise/fall time, in this case for an edge rate of 20 ps.

Description	Intrinsic delay [ps]					KLOAD=0.7Reff [kΩ] or [ps/fF]				
	X4	X8	X16	X24	X32	X4	X8	X16	X24	X32
A to Z ↓	40	36	35	32	33	2.35	1.20	0.60	0.40	0.30
A to Z ↑	45	40	35	35	35	3.25	1.65	0.80	0.55	0.40
B to Z ↓	40	35	32	32	30	2.35	1.20	0.60	0.40	0.30
B to Z ↑	44	40	35	35	35	3.25	1.65	0.80	0.55	0.40
Input cap [fF]						1.2	1.5	2.2	3.6	4.5

In a more accurate propagation delay model suitable for EDA tools, but not so user-friendly for back-of-the-envelope pre-simulation estimates, the propagation delay also depends on the input edge rate (i.e. rise/fall time). In a slope-based linear delay model, there are three model parameters, the intrinsic (or parasitic) delay, t_{int}, the effective resistance, $K_{LOAD} = 0.7R_{eff}$, and the input edge rate, K_{edge}, from which the propagation delay can be found for any combination of load capacitances, C_L, and input edge rates,

$$t_{pd} = t_{int} + K_{LOAD} \times C_L + K_{edge} \times edge\_rate, \tag{4.13}$$

where, for simplicity, the input edge rate usually is assumed to be proportional to the propagation delay of the previous stage.

Unfortunately, the real world is not always linear, particularly not if wide ranges of capacitive loads and input edge rates are to be covered. This can be verified by simulating the output response for a large number of input edge rates and output load capacitances. Data obtained from such simulations under typical as well as under worst case conditions, are then stored in look-up tables (LUTs) such as the one shown below. The propagation delay for any edge rates and load capacitances between the given corner-points can be obtained by interpolation. A typical relationship between the approximate RC model used in previous sessions, and a more complete nonlinear model is shown for the inverter in Fig. 4.11. In this case the linear approximation has been chosen to agree with the more detailed slope-based model for short edge rates and small load capacitances, and for large edge rates and large load capacitances. This method has been selected for reasons that short delays are usually associated with small edge rates, while long delays are mainly associated with large load capacitances and large edge rates.

Description		input transition time [ps]			
		5	120	240	480
output load capacitance [fF]	1.5	10	50	70	120
	9	30	90	125	190
	18	50	120	165	240
	35	90	160	225	320
	70	170	240	320	450

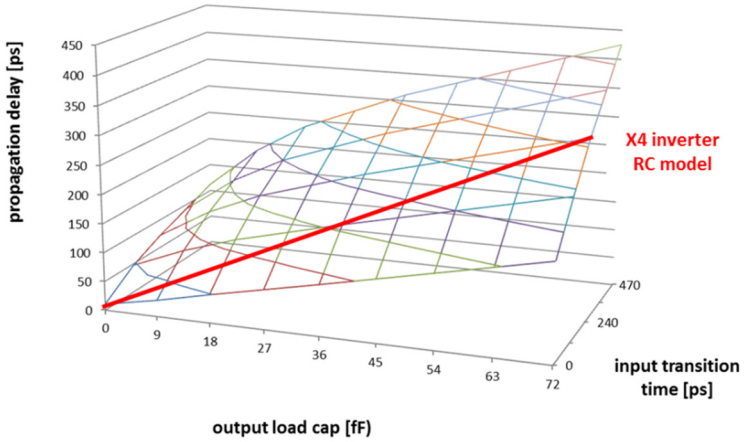

Fig. 4.11. Comparison between the RC delay model and the slope-dependent nonlinear delay model.

Chapter summary: to follow