3rd Laboratory Exercise

Theoretical & practical calculation of delays in CMOS VLSI inverters

1 Introduction

One of the most important steps in designing VLSI circuits is calculating the delay that each gate or stage introduces to its input signal. The delay is due to the parasitic capacitances between different levels, depends on the design technology and is what differentiates the operation of the transistor from the model of the ideal switch (with instantaneous transitions). The purpose of this laboratory exercise is the practical and theoretical calculation of the delay in CMOS inverter circuits
VLSI. Especially the theoretical calculation, which can be very well approximated by simple linear models, can greatly help the designer to choose the network topology and transistor sizes that minimize the delay, before starting the time-consuming process of layout design and practical calculation of circuit properties through simulation.

2 Definitions of arrears

Delay time in a VLSI circuit expresses the delay observed from the time a change occurs in the input until the corresponding change occurs in the output. Although there are different measurable quantities of delay, the most common is the propagation delay time (tpd), which is defined as the maximum time from the moment the input exceeds 50% of its value to the moment the output exceeds 50 % of its price. The propagation delay time is distinguished into a fall propagation delay time at the *tpdf* output and a rise propagation delay time at the *tpdr output.* In each circuit it is desirable that these two times coincide. For this reason, in the following we will deal with a single value of propagation delay d, which we consider corresponds to both practically equal values.

The normalized propagation delay *d* of a gate can be written as:

$$d = f + p \quad (1)$$

Where *p* is the *intrinsic parasitic delay* of the gate, when no load is attached, while *f* is the *load delay* (stage effort), which depends on the complexity of the gate *g* and the *fan* -out *h* which depends on the output capacitive load of the gate, according to the formula:

$$f = gh \quad (2)$$

The complexity is represented by the *logical load* (logical effort) g. An inverter is defined as having a logic load of 1. More complex gates have a larger logic load, which indicates that they require more time to drive a given output voltage. For example, the logic load *(g)* of a 2-input NAND gate is 4/3. The factor *h*
is called *the electrical load ,* and for a gate driving identical copies of it, it is defined as the number of copies. For a portal that does not drive identical copies of itself, o
electric charge *h* can be calculated as:

$$h = \frac{C_{out}}{C} \qquad (3)$$

Where *Cout* is the capacitance of the external load being driven and *Cin* is the input capacitance of the gate.

Figure 3.1 shows the normalized propagation delay as a function of *electrical effort* for an inverter and a 2-input NAND gate. The points of intersection with the *d*- axis denote the parasitic delay. The normalization is obtained by considering the parasitic delay of the inverter 1. The slope of the lines is the *logical load.* The inverter has a logic 1 load by default while the NAND gate has a logic 4/3 load.

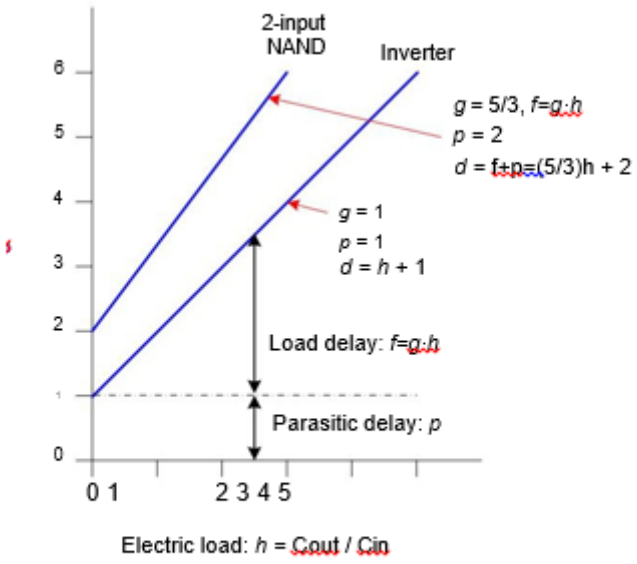


Figure 3.1: Normalized propagation delay

1 Delay calculations

Although transistors have complex voltage-current characteristics, they can be approximated fairly well as a switch in series with a resistor, where the active resistance is chosen to match the average current delivered by the transistor. Transistor gate nodes and drains exhibit capacitance. Such an approximate model, for both nmos and pmos transistors, is shown in figure 3.2.

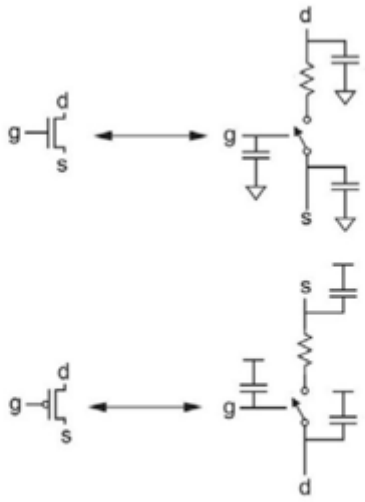


Figure 3.2: Approximate RC transistor model

The values of the resistances and capacitances of figure 3.2 depend on the dimensions of the transistors. Typically, all logic gates are minimum length (2m) elements, for minimum delay, area and power consumption. Consequently, resistances and capacitances ultimately depend on transistor widths, which the designer must choose appropriately.

A minimum width nmos transistor (2ÿ) is defined to have an active resistance R. A corresponding minimum width pmos transistor has a higher resistance which depends on the mobility of its carriers than the nmos transistor. Roughly this resistance can be considered *2R.* Larger width transistors have lower resistance. For example, a pmos transistor twice as wide as the minimum has an active resistance of R. In general, multiplying the width of a transistor by *n* means dividing its resistance by n. Transistors in series and parallel are combined like conventional resistors. When several transistors are connected in series, the resistance is the sum of the individual resistances. When connected in parallel, the resistance is the smaller the more they conduct together. The largest possible resistance occurs when only one transistor is conducting, so the total resistance is equal to the resistance of the conducting transistor.

When designing logic gates and VLSI integrated circuits we try to ensure that the active resistance is less than or equal to R, both in PUN and PDN, in order to minimize the delay (which in any *RC* circuit is proportional to both *R* and of C). For example, Figures 3.3 and 3.4 show the widths of the nmos and

pmos transistors so that both the inverter and the 3-input NAND gate have an active resistance R. Note especially the case of the NAND gate where in the Pull-Down mode the 3 transistors in in series they each have a resistance R/3 (so total R) while in Pull-Up, the largest resistance *R* is observed when only one transistor conducts. If two
(2) conduct there is resistance R/2 and if three (3) conduct the resistance is R/3.

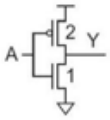


Figure 3.3: Inverter transistor back

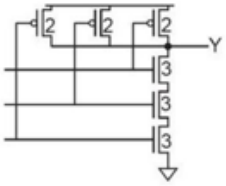


Figure 3.4: Backplane of a 3-input NAND gate transistor

Also, the capacitance of a transistor consists of the gate capacitance and the source/drain capacitance. Let us define the gate capacitance of a transistor of minimum length as Cg and the corresponding source/drain capacitances as Cdiff. In most implementation technologies these two capacitances are approximately equal so to keep the estimation simple we can denote *C*= Cg= Cdiff. For larger transistors, both nmos and pmos, the capacitance *C* is proportional to the transistor width. For example, Figure 3.5 shows the capacitances of a 3- input NAND gate.

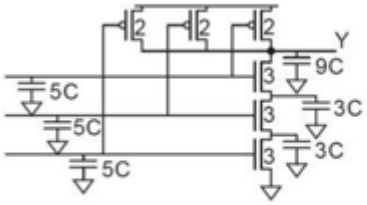


Figure 3.6: Simplified 3-input NAND gate capacitance circuit

Note that in series nmos transistors there is a capacitance for each common drain, which serves as a source on one transistor and a drain on the other. Also, any capacitors with shorted terminals can be ignored while the capacitors connected to the supply voltage Vdd can be considered to be connected to earth as the voltage level

(Vdd or *Gnd)* has no effect on the delay. Based on these assumptions and summing parallel capacitances we arrive at the equivalent Figure 3.6, in which each input sees a 5C capacitance and the output has a 9C capacitance.

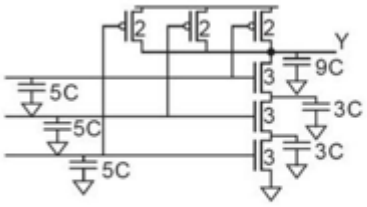


Figure 3.6: Simplified 3-input NAND gate capacitance circuit

Using these figures we can go back to formulas (1), (2) and (3), to calculate the propagation delay. The logic load of a compound gate can be estimated as the ratio of the input capacitance of the gate to the input capacitance of the minimum inverter. Figure
3.7 shows some typical examples, while a more detailed record is made in table 3.1.

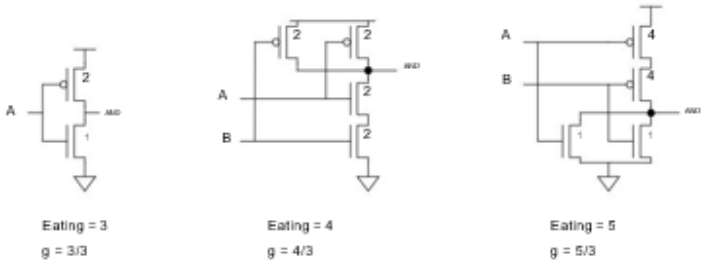


Figure 3.7: Logic load of inverter, NAND gate and 2-input NOR gate

Gate type	Number of inputs				
	1	2	3 4		n
inverter	1				
NAND		4/3	5/3	6/3	(n+2)/3
NOR		5/3	7/3	9/3	(2n+1)/3
multiplexer	2	2	2	2	2
CHORUS, XNOR		4, 4 6, 12, 6 8, 16, 16, 8			

Table 3.1: Logic load of logic gates

The parasitic delay of a gate can be approximated by the dissipation capacitance at the output node. For example, in Figure 3.7 the inverter has an output capacitance of *3C* and a resistance when conducting either Pull-Down or Pull-Up equal to R. Therefore, when operating with no external load it has a delay *of 3RC=ÿ.* The parameter ÿ can be calculated for each implementation technology (for MICROWIND cmos018 technology it has been calculated as ~12.4 ps) and the parasitic capacitances of all gates normalized based on this parameter

Looking at figure 3.7 we can say that the inverter has a normalized parasitic delay of 1 while the NAND and NOR gates with an output capacitance of 6 have a normalized parasitic delay of 2. Table 3.2 gives the normalized parasitic delay of basic digital gates.

Gate type	Number of inputs				
	1	2	3 4		n
inverter	1				
NAND		2	3	4	n
NOR		2	3	4	n
multiplexer	2	4	6	8	2n

Table 3.2: Parasitic delay of logic gates

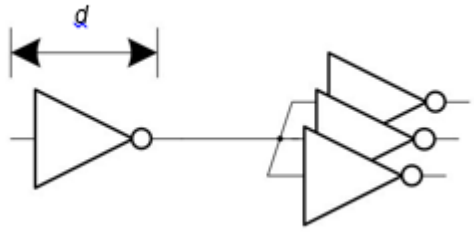
In summary, the approximate theoretical calculation of the propagation delay of a gate consists of the calculation of two normalized values, the parasitic delay and the product of the logical effort by the electrical effort. The resulting number is multiplied by the parameter ÿ to calculate the absolute value. For all calculations, the widths of the transistors and their connection are taken into account, chosen so as to result in a resistance *R* in both Pull-Down and Pull-Up. The lengths of the transistors are chosen minimally (2l). The resulting delays can be verified with the practically calculated values by simulation.

1 The requirements of the laboratory exercise

1. Design the layout with the *Euler Paths technique,* in *cmos65n* technology of a full adder (FA), with minimum channel length in the transistors (L=2ÿ) and width Wn=6ÿ for nmos and Wp=12ÿ for pmos. Then check their correct operation and calculate through the simulation the maximum delay introduced by the above circuit on both its outputs. We assume an output load of 2fF for both outputs.

2. In *cmos018* technology design a minimum inverter with dimensions: Wn=0.4ÿ, Ln=0.2ÿ and Wp=0.8ÿ, Lp=0.2ÿ. Calculate its no-load delay *(ÿ)* . Then drive a number of *n* identical inverters,
   with the above inverter, connected in parallel (ie the output of our inverter is connected to the common input node of the *n* inverters). The normalized approximate delay *d* of the minimum inverter is given by the table below:

Minimum inverter driving *n* minimum inverters	Calculation delay *d*
n=1	d=1+1*1=2
n=3	d=1+1*3=4
n=6	d=1+1*6=7



Verify the above formulas based on the delay ÿ of the minimum inverter
that you calculated before comparing with the values calculated through simulation
in MICROWIND.