High Speed CMOS VLSI Design

Lecture 1: Gate Delay Models

1.0 Designing for Speed on the Back of an Envelope

Custom IC design is all about speed. For a small amount of money, one can synthesize a functional description of a chip and toss it into a field-programmable gate array running at 50 MHz, or even code the problem in software and run it on a PC. Thus, the main value of custom logic chips is to claim performance that automated CAD tools can't deliver.

Unfortunately, interesting designs often involve ten million or more transistors. For a team with finite resources to complete the design in a reasonable time, only a small amount of time may be spent on each transistor. Since processor performance doubles every 18-24 months, slipping the schedule by a year on a custom design may be no more competitive than synthesizing a gate array on time. Thus, a good designer attempts to spend as little time as necessary to meet specifications. For the most critical portions of the chip, detailed simulation and extensive optimization can be justified. Non-critical blocks may just be synthesized, though as cycle times decrease, the number of non-critical blocks dwindle. The large fraction of remaining moderately critical paths must be designed using fast, approximate techniques that are "good enough."

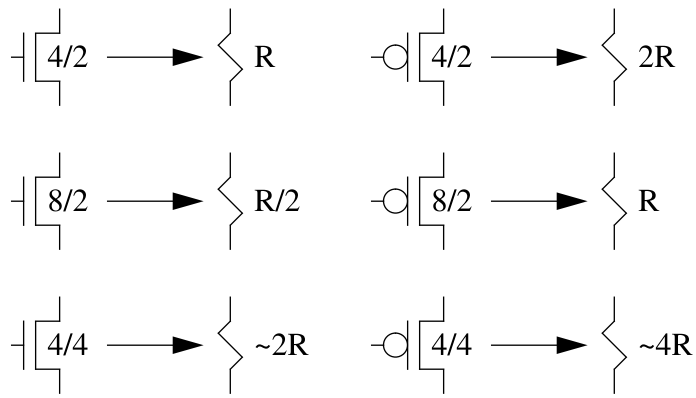
The emphasis of this course is on designing high speed custom CMOS chips with as little effort as possible. We'll develop intuitive models which quickly produce circuits that are close to optimal. Moreover, simulating is a difficult art and usually the model given to the simulator is wrong. If the designer doesn't know approximately what the result of the simulator should be, there is very little chance of getting a correct answer. By quickly being able to roughly predict the performance of a circuit or how the performance will change when a parameter is twiddled, the designer has a better hope of getting meaningful results from more detailed simulation.

2.0 RC Transistor Model

Traditional classes teach a quadratic model of the MOS transistor, involving saturation, linear, and cutoff regimes of operation and messy formulae for current-voltage relationships. More advanced classes add second-order corrections which are very important for short-channel transistors. One such model, agreeing to within 3% of SPICE on inverter delay, takes three pages to express [1] and is far too difficult to use without computer simulation anyway. These models have value in understanding the precise shape of switching waveforms, in analyzing certain analogish circuit structures, and in simulation, but are completely unmanageable for delay estimation in digital circuits.

Remarkably, we can predict delay fairly well by just modeling a transistor with an resistance from source to drain when ON and a few capacitances. Figure 1 shows the resistances of several geometries of NMOS and PMOS transistors. Dimensions are labeled as WIDTH/LENGTH.

FIGURE 1. Resistance of ON transistors



We define a unit sized NMOS transistor, i.e. a 4/2 λ transistor with the minimum channel length and contacted diffusion width, to have resistance R. Resistance scales inversely with width, so an 8/2 NMOS transistor has resistance R/2. To first order, resistance scales proportionally with channel length, so a 4/4 NMOS transistor has resistance 2R. However, the effective channel length L_{eff} is less than the drawn channel length by some amount of lateral diffusion and depletion width. Therefore, doubling drawn channel length more than doubles the effective channel length and thus more than doubles resistance. Fortunately, digital circuits seldom use longer than minimum channel length devices so accurately modeling the resistance is seldom an issue. A PMOS transistor has lower carrier mobility than an NMOS transistor and hence a higher resistance. The mobility ratio is typically 2-3. For hand estimation lacking specific process information, it is convenient to assume that a unit PMOS transistor has resistance 2R.

The primary capacitance is from gate to source. Parasitic diffusion capacitances on the source and drain are also large enough to be important. For a unit device, define the gate-source capacitance to be C, and the diffusion capacitance to be C_{diff}. C is typically slightly under 2 fF/μm of gate width and doesn't change much between process generations because linearly shrinking channel length and oxide thickness cancel each other out. C_{diff} depends on doping levels but is usually comparable to C for contacted diffusion regions and about half C for small shared diffusions. Routing signals in diffusion leads to huge diffusion parasitics and should be avoided for critical gates.

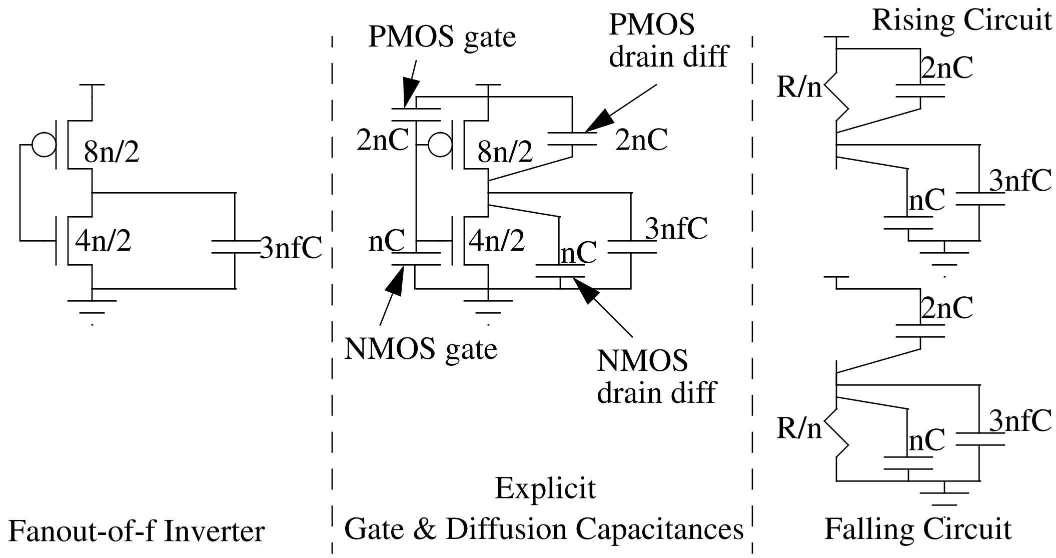
The RC product is defined to be τ, known as the intrinsic delay of an NMOS transistor. This product is the delay of an inverter driving its own gate. R, C, and t are tabulated below for four generations of HP processes fabricated through MOSIS. Some of the processes are simulated over various voltages. Notice that reducing the supply in a given process increases delay while saving power. The capacitance also appears slightly higher at higher voltage because faster switching times increase the Miller multiplication of gate-drain parasitic capacitances.

TABLE 1. Four generations of transistor parameters

Channel Length (μm)	**Voltage**	**C (fF/μm)**	**R (kΩ*μm)**	τ (ps)
1.2	5	1.79	14.5	26
0.8	5	1.91	11.0	21
0.8	3.3	1.83	15.8	29
0.6	3.3	1.64	7.9	13
0.6	2.5	1.58	10.8	17
0.35	2.5	1.69	7.1	12

Given the resistance and capacitances, we can redraw an equivalent circuit for a gate and use it to estimate propagation delay. The propagation delay is defined to be the time from when the input reaches VDD/2 until the output crosses VDD/2. For example, consider a fanout-of-f inverter, shown in Figure 2. The left side of the figure shows the inverter schematic. A gate with a fanout of f drives a load equal to f times the input capacitance. The inverter is sized n times unit size, so the width of the NMOS transistor is 4n. The PMOS transistor is 8n wide, to provide equal rise and fall resistances. Thus, the total input capacitance of the inverter is nC + 2nC = 3nC. For a fanout of f, the load capacitance is 3nfC. The middle portion of the figure shows the schematic annotated with capacitances. The NMOS and PMOS transistors are n and 2n times unit size and thus have gate-source capacitances of nC and 2nC, respectively. We can estimate the source/drain diffusion capacitances to be equal to the gate capacitance. Notice that the diffusion capacitances on the sources never change voltage and therefore never impact delay, so they are left out of the figure. Finally, the right side of the figure shows simplified equivalent circuits for rising and falling transitions. The resistances of either transistor while on are R/n. The capacitance on the output node, consisting of the load and the two drain diffusion capacitances, must be charged or discharged through the resistor. For an inverter, the gate capacitance does not affect the delay because it is not charged or discharged during an output transition. Of course, it does contribute loading to the previous stage.

FIGURE 2. Fanout-of-f inverter and equivalent circuit

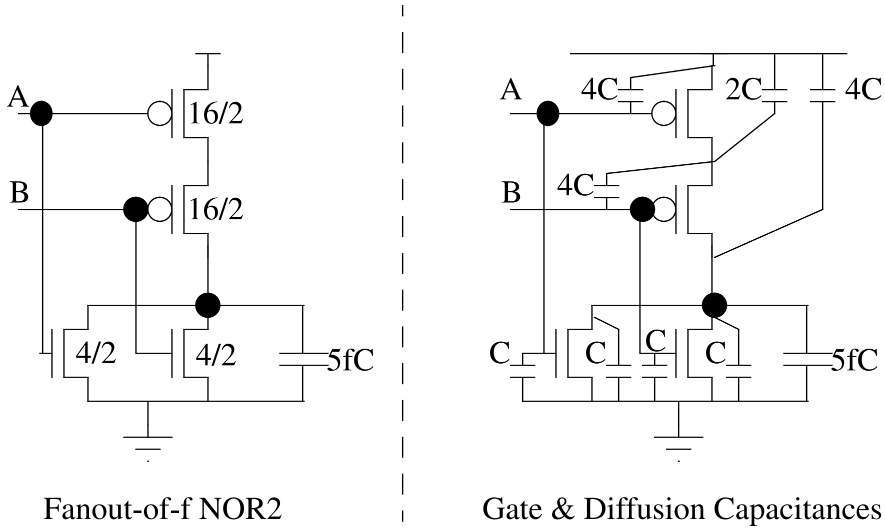


The delay is computed by the Elmore delay model, which approximates delay as the sum over all circuit nodes of the resistance between that node and VDD/GND times the total capacitance on the node. For an inverter, the output is the only node. Rising and falling delays are identical, with a resistance of R/n multiplied by the total capacitance of nC + 2nC + 3nfC for a product of 3(f+1) τ.

Notice that the fact that the inverter is n times unit size does not show up in the delay. Delay is only a function of fanout and intrinsic parasitics, not of absolute size. Therefore it is generally unwise to increase the size of all gates in a path because it only increases area and power dissipation without improving delay. Two exceptions to this guideline involve wiring capacitance and folded transistors. If wiring capacitance dominates gate capacitance, as it frequently does in random logic blocks automatically placed and routed, using larger than minimum devices reduces the effective gate resistance while only slightly increasing total load capacitance. Larger than minimum transistors may also be folded to reduce diffusion capacitance and hence reduce the intrinsic delay somewhat.

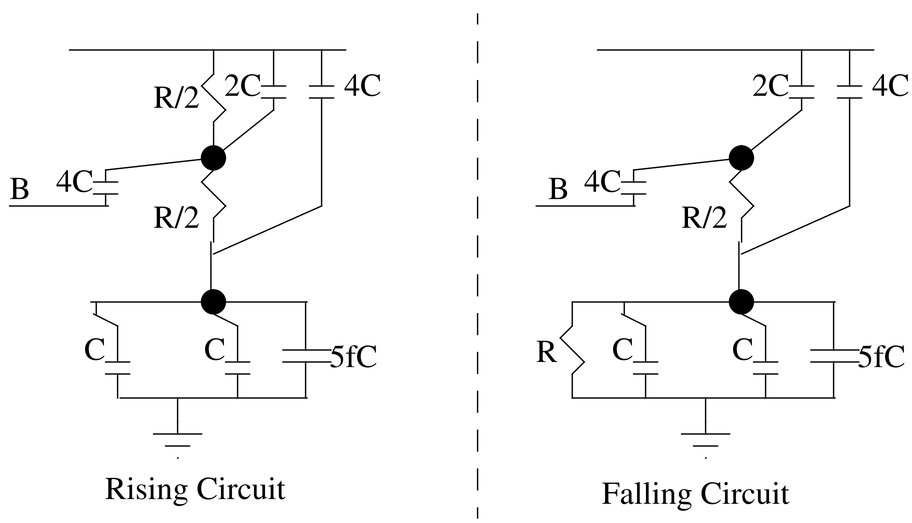
As a second example, consider a 2-input NOR gate with a fanout of f shown in Figure 3. Since the delay only depends on fanout, not on absolute channel width, use unit sized NMOS transistors. To achieve equal rise and fall resistances, the PMOS transistors must be sized 4 times unit size because each is half as strong as an NMOS and there are two in series. The capacitances are annotated on the right side of the figure. Notice how the gate-source capacitance of the bottom PMOS device connects to an internal node. We assume each NMOS transistor has a contacted drain diffusion, though good layout might share the contact between the two diffusions. The drain of the lower PMOS device must be contacted, but the middle node in the PMOS stack does not need a contact and therefore contributes only half as much diffusion capacitance. Since the total input capacitance is 5C on either input, the output load is 5fC for a fanout of f.

FIGURE 3. Fanout-of-f NOR gate.



The rise and fall paths shown in Figure 4 are not identical, so their delays must be computed separately. First consider a rising output. Input B is stable while the output switches, but the source of the transistor may switch from 0 to 1. Therefore, the gate-source capacitance on that transistor must also be charged and contributes to the delay. Now we can apply the Elmore delay model to a circuit with two nodes. The internal node has a resistance R/2 to VDD and a capacitance of 2C + 4C from the diffusion and gate-source capacitance, respectively. The output node has a resistance R/2 + R/2 and a capacitance 5fC + C + C + 4C from the output load, the two NMOS diffusion capacitances, and the PMOS drain diffusion. Therefore, the total rise delay is R/2 [2C+4C] + (R/2 + R/2)[5fC + C + C + 4C] = (9+5f) τ.

FIGURE 4. NOR2 equivalent circuits



Next, consider the falling output. The output node discharges through one or both NMOS transistors; the slower case is when only one NMOS transistor is on. Moreover, if input B is 0 and input A rises, the internal node in the PMOS stack will also discharge. How do we model this internal node discharge? The Elmore delay model does not apply well because the internal node is not on a path between the output and a rail. We can either ignore the capacitance on the internal node or conservatively lump it onto the output node. Ignoring the capacitance simplifies hand calculation and is not too bad because the main effect of such capacitance is to create a longer tail on the switching waveform, rather than to greatly effect propagation delay. If we ignore the internal node capacitance, the Elmore delay is $R\,[5fC + C + C + 4C] = (6 + 5f)\,\tau$. Notice how the falling delay is faster than the rising delay, even though the pullup and pulldown transistors were sized for equal resistance. This is because the pulldown path has less internal parasitic capacitance to drive.

Also note that for all the gates we have considered, the delay is in the form $a + bf$, where a represents an intrinsic delay of the gate charging its own diffusion capacitance and b represents the additional delay per fanout. More complex gates increase both the intrinsic delay and the delay per fanout.

To conclude, we can estimate the delay of a gate by modeling it with an RC circuit, then using the Elmore delay formula. With practice, such estimation becomes very fast and easy, far faster than firing up a circuit simulator. Since the purpose of the model is fast, approximate answers, keep the model simple. For instance, internal wiring capacitances are required for an accurate delay estimate, but are difficult to compute without detailed layout information. Similarly, details about shared diffusion contacts are also important but require layout information. An easy compromise is to assume all diffusions are contacted, and that no wire is present. This overestimates diffusion capacitance and underestimates wire capacitance, resulting in an answer that is hopefully not too bad and avoiding any layout concerns. It also gives an analytic expression for delay so it is easy to understand how changing sizes or topology will affect overall delay. We will use this model extensively in the class to understand sizing of transistors.

3.0 RC Model Limitations

Transistors do not have linear I-V characteristics, so modeling them as resistors is necessarily inaccurate. Nevertheless, the models agree remarkably well with actual circuits in many cases. It is important to understand the limitations of the RC model to know when the results are trustworthy.

The resistances in the RC model are found by curve-fitting to the delay of an inverter driving a particular fanout (generally 4) and with a particular input slope. Therefore, the model is most accurate for circuits that resemble such an inverter. Three sources of inaccuracy are different topologies, different slopes, and velocity saturation.

Circuits that don’t look like inverters may not match the model as well. For example, transmission gate circuits and pseudo-NMOS circuits do not switch through simple stacks of NMOS or PMOS gates connected to a rail.

Input slope makes an important difference in propagation delay. If the input instantly rises from 0 to 1, the gate will be much faster than if the input slews over a nanosecond. The resistance is normally characterized for an input driven by a fanout-of-4 inverter. Therefore, different input slopes produce inaccuracy in the delay estimate. Static timing analyzers often use an RC delay model with additional terms to reflect slope dependence, but these additional terms defeat the point of having a simple model for hand analysis. Fortunately, circuits optimized for speed generally have gate delays comparable to a fanout-of-4 inverter, as we will see later. Therefore, in well-designed paths the slopes are close to the slope to which the model is calibrated and the model works well.

Velocity saturation is another important contributor to model inaccuracy. Two unit sized NMOS transistors in series would be expected to have resistance 2R. However, each sees a smaller drain-source voltage, so each has slower-moving carriers which are not as velocity saturated. Therefore, the currents are better than expected and the actual resistance is lower than predicted, sometimes as much as by 30% depending on the amount of velocity saturation. In summary, series stacks of NMOS transistors run faster than the model would predict. PMOS transistors are not as affected because they are not very velocity saturated.

4.0 Tapered Buffer Chains

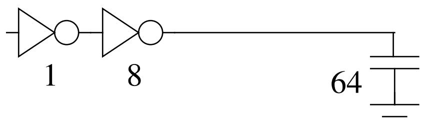
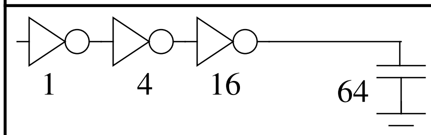
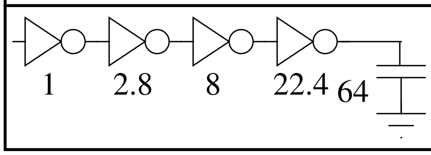
Let us apply the RC delay models to the problem of quickly driving a large capacitive load with a chain of inverters or buffers. The analysis will provide insight on the optimal fanout of each stage in the chain. Later, we can generalize this result to understand the optimal sizing of an arbitrary network of gates.

We begin with a block specification of the problem. A block spec includes the function of the block, the load capacitance on each output, the maximum capacitance that may be presented on each input, and the maximum allowable delay through the block. It is important for the spec to include all four parts. Frequently specs are given with just the first two pieces of information. If no maximum input capacitance is given, the circuit designer may use huge gates on the input of the path, decreasing path delay but passing the problem off to another unfortunate designer who now must drive the huge input. Moreover, the huge gates in the block may occupy vast amounts of silicon real estate and dissipate massive amounts of power. Similarly, the maximum delay part of the spec is sometimes not given. This is reasonable during feasibility studies, which may explore how fast it is possible to implement a particular function. However, it is a waste during actual design because the block may get overdesigned, taking more area, power, and designer time than necessary to meet a loose maximum delay.

In our tapered buffer chain problem, the block specification is to buffer a weak input to drive a large load. The function of the block is either a buffer or an inverter; for this analysis we'll be indifferent as to the polarity of the output. In some cases this is realistic because the next block may be easy to design with either polarity input or may require both true and complementary inputs anyway. In other cases, maintaining the polarity of the signal may be important. The output load is C_L and the maximum input load is C_{in}. The delay requirement is to minimize delay.

Consider a variety of ways in Figure 5 to drive a load C_L of 64 units given an input capacitance C_{in} of 1 unit (where units are arbitrary, since only fanouts and not absolute sizes impact inverter delay). Use our earlier result that the delay of an inverter is 3(f+1) τ. We see that using more stages reduces the fanout per stage and the delay of each stage. However, using too many stages increases overall delay. The optimum is 3 stages for this example, though 2 stages is not bad and 4 stages is very close to optimal, so little is lost by using an even number of stages to preserve signal polarity. Using fewer stages is better to reduce area and power consumption, both of which are dominated by the large final stage.

FIGURE 5. Driving a fanout of 64

Circuit	# of stages	f	delay/stage	total delay
	1	64	195	195
	2	8	27	54
	3	4	15	45
	4	2.8	11.4	45.9

We can solve in general for the optimal fanout f of each stage. The number of inverters in the chain is $\log_f C_L/C_{in} = \ln(C_L/C_{in}) / \ln f$. The delay of each inverter was earlier found to be 3(f+1) τ assuming diffusion parasitics equal to gate capacitance. To be more general, we can let the delay be 3(f+α) τ where α is a parameter reflecting the ratio of parasitics to gate capacitance. Folding the transistors halves the diffusion capacitance, reducing α toward 0.5. Wiring between the inverters increases the parasitic capacitance and hence α. An ideal inverter chain with no parasitics would have $\alpha = 0$.

The total delay of the stages is therefore:

$$t = 3\ln\frac{C_L}{C_{in}}\frac{(f+\alpha)}{\ln f} \qquad \text{(EQ 1)}$$

We want to find the value of f which minimizes delay. We can do this by taking the derivative of delay with respect to f and setting it equal to 0:

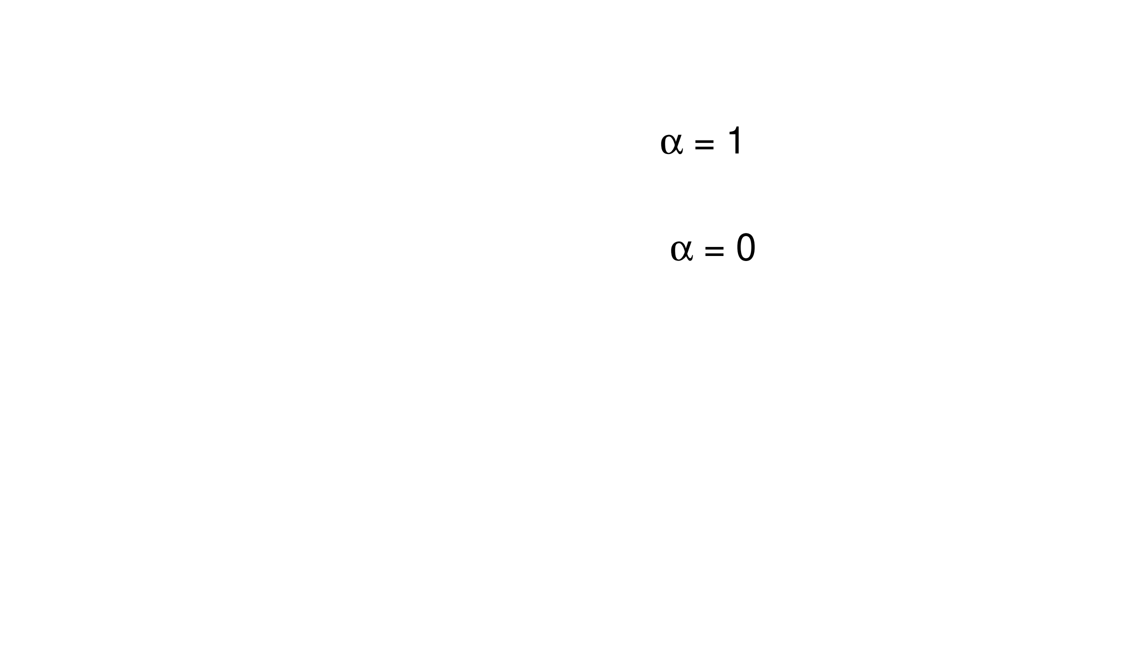
$$\frac{dt}{df} = 3\ln\frac{C_L}{C_{in}}\left[\frac{\ln f - \frac{(f+\alpha)}{f}}{(\ln f)^2}\right] = 0 \qquad \text{(EQ 2)}$$

This can only be true when the numerator is 0:

$$\ln f = \frac{(f+\alpha)}{f} \qquad \text{(EQ 3)}$$

Unfortunately, there is no closed-form solution to this equation. If $\alpha = 0$, the equation simplifies to ln f = 1, or f = e, an optimal fanout of about 2.7. This is the common textbook result ignoring parasitics. For larger values of α, the optimal fanout increases. This makes sense because if each stage has a large intrinsic delay, it is faster to use fewer stages and reduce the total intrinsic contribution to the delay at the expense of higher fanout-dependent delay. Figure 6 below plots relative delay vs. fanout for a = 0, 0.5, and 1.

FIGURE 6. Delay vs. fanout[1]



Notice how the minimum delay point increases with α. For real circuits with parasitics, a fanout of about 4 per stage is good, giving roughly minimum delay and also reducing the number of stages required. Also notice how flat the curves are. For fanouts anywhere in the range of 2-6, the delay is within a few percent of optimal. Therefore, using a fanout of precisely 4 is not very important; depending on the needs of the circuit a higher or lower fanout can be used.

1. This figure is from Stanford's EE271 lecture notes, (c) 1996 Mark Horowitz. Reprinted without permission.

This flat optimum is characteristic of many optimization problems in circuit design. This is good because it allows the designer to be sloppy: if the circuit is sized anywhere near optimum, the delay is extremely close to optimal. Therefore, even if the RC delay models aren't especially accurate, the sizes they dictate using will be close enough to right that the delay is nearly optimal. Also, the designer can round sizes to convenient integers for back-of-the envelope calculation rather than tuning circuits to several significant figures.

5.0 The Fanout-of-4 Inverter

We have seen that the optimal design for buffering heavy loads is a chain of inverters with fanouts of about 4. The fanout-of-4 inverter delay is very useful for thinking about circuits. It has a delay of $(12+3\alpha)\ \tau$, which is about 15 τ and only weakly dependent on the process-dependent value of α. We can express circuit delays in process-independent terms of fanout-of-4 inverters, henceforth abbreviated FO4 delays. We can also quickly estimate the delay of certain functions based on the number of FO4 inverters required to drive the fanout.

Processes continue to advance, so reporting that an adder operates in 0.92 ns says little about the actual merits of the adder unless a good deal of information is available about the process and about the operating conditions under which the adder was simulated. On the other hand, reporting that the adder takes 7 FO4 delays provides a great deal of information. As long as the relative delay of the gates in the adder tracks well with the delay of a FO4 inverter across process and environmental variations, we can expect that the same adder will take about 7 FO4 delays in any other process and environment. Simulation shows that across four generations of processes, delay of various circuits measured in terms of fanout-of-4 inverters remains within 15% of constant, frequently closer [2], so the metric works well. Publishing results expressed in terms of FO4 delays, or in terms of nanoseconds in conjunction with the delay of a FO4 inverter in the same environment, will make circuits much more useful to people working in with different processes.

FO4 delays are also useful for estimating many circuit delays. For example, consider control logic driving a select signal across a 64 bit datapath. If both the gate producing the control signal and the receivers in the datapath are unit sized, three fanout-of-4 delays are required for a buffer chain to power up the control signal to drive the datapath, since $\log_4 64 = 3$. This gives intuition why time must be budgeted when control signals interact with datapaths.

6.0 Conclusions

We have seen that to reach market on time, circuit designers need simple delay estimation techniques that are more efficient than detailed simulation. We have explored the RC delay model which, with practice, allows rapid delay estimation and which provides an analytic expression of delay. Using the analytic expressions, we have shown that the optimal fanout per stage of a buffer chain is about 4, and that the optimum is very broad so fanouts between 2 and 6 work well.

7.0 References

[1] P. Cocchini, G. Piccinini, and M. Zamboni, “A Comprehensive Submicrometer MOST Delay Model and Its Application to CMOS Buffers,” JSSC 8/97, p. 1254-1262.

[2] D. Harris, R. Ho, G. Wei, and M. Horowitz, “The Fanout-of-4 Inverter Delay Metric,” unpublished manuscript.