Gate Sizing for Optimal Path Delay

The proper specification of the optimal path delay problem involves the minimization of the path delay given both input and output loading constraints.

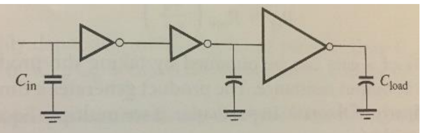


Figure 6.19
Input loading effects of large drivers.

In Figure 6.19, if we specify both C_{in} and C_{load} and ask for the optimal sizing to minimize the delay, the problem is properly specified, and we can focus on its solution.

$$path\_delay = \sum R_i C_i$$

where R_i is the driving resistance of each gate and C_i is the output loading capacitance on the gate. We select the transistor widths to minimize the delay.

Inverter Chain Delay Optimization: FO4 Delay

Changing the size of a gate affects both the *input **capacitance*** and the *output **resistance*** of a gate.

The input capacitance of a gate is given by

$$C_{in} = C_g(W_n + W_p)$$

For an inverter with $R_{out,NMOS} = R_{out,PMOS}$,

$$C_{in} = C_g(W_n + W_p) = C_g(W_n + 2W_n) = 3C_g W_n$$

The effective output resistance for the NMOS device is given by

$$R_{out,NMOS} = R_{eff} = R_{eqn}\left(\frac{L_n}{W_n}\right) = R_{out,PMOS}$$

We define

$$intrinsic\ time\ constant\ for\ an\ inverter\ \tau_{inv} = R_{eff}C_{in} = 3R_{eqn}C_g L_n$$

For an inverter,

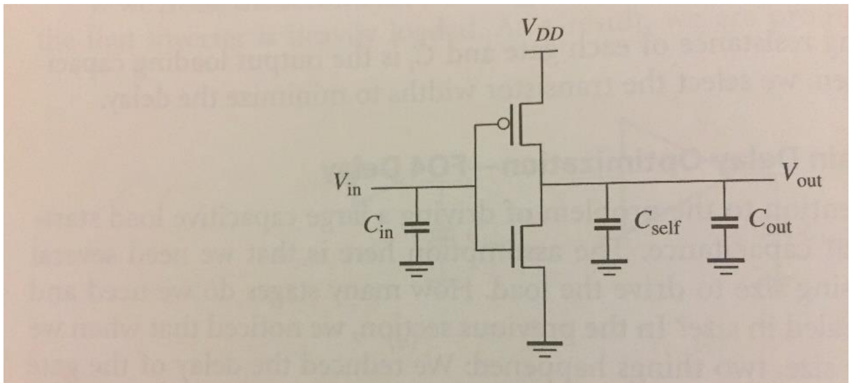


Figure 6.20
Delay for inverter driving a load.

$$t_{delay} = R_{eff}\left(C_{out} + C_{self}\right) = \tau_{inv}\left(\frac{C_{out}}{C_{in}} + \gamma_{inv}\right)$$

where $\gamma_{inv} = \frac{C_{self}}{C_{in}}$. Note that γ_{inv} is highly dependent on the layout of the gate, since C_{self} depends on the layout.

$$\text{fanout ratio (electrical effort)} \quad f \triangleq \frac{C_{out}}{C_{in}}$$

Conclusion: To obtain the minimum delay of the inverter chain, we can increase the size of each inverter by a fanout factor f. That is, each stage drives an inverter that is f times larger than itself. In this case, the delay through each gate must be the same since each one drives an inverter that is f times itself. *The overall delay is simply N times the delay of one inverter* because the optimal value of f would equalize the delays of all stages; the proper value of f would make the delays of all stages identical.

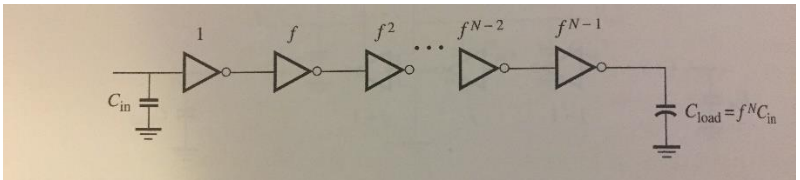


Figure 6.22
Series chain of NAND gates.

$$total_{delay} = \frac{\tau_{inv}(f + \gamma_{inv}) \ln\left(\frac{C_{load}}{C_{in}}\right)}{\ln f}$$

The optimal fanout ratio f is given by

$$N \ln f = \ln\left(\frac{C_{load}}{C_{in}}\right) \Rightarrow f = \left(\frac{C_{load}}{C_{in}}\right)^{\frac{1}{N}}$$

The optimal number of stages N is given by

$$N = \frac{\ln\left(\frac{C_{load}}{C_{in}}\right)}{\ln f}$$

The total delay of the inverter can be computed by

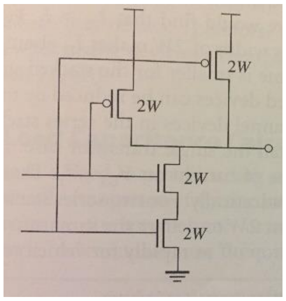
$$total\_delay = N\tau_{inv}(f + \gamma_{inv})$$

FO4 Rule: Each successive inverter is 4 times larger than the previous one.

Optimizing Paths with NANDs and NORs

Consider the following device sizes, we can compute the effective output resistance and the input capacitance., then determine the intrinsic time constant.

Case I: NAND 2



$$\frac{W_{P,effective}}{W_{N,effective}} = \frac{2}{1} \Longrightarrow R_{eff} = R_N = R_P$$

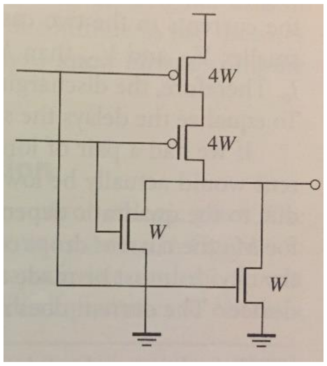
$$\therefore R_{eff} = R_{eqn}\left(\frac{L_n}{W_n}\right)$$

$$C_{in} = C_g\left(W_n + W_p\right) = C_g(2W + 2W) = 4WC_g$$

Note that C_{in} is the gate capacitance corresponding to *ONE input terminal*.

$$\tau_{nand} = R_{eff}C_{in} = 4R_{eqn}C_gL_n$$

Case II: NOR 2



$$\tau_{nor} = R_{eff}C_{in} = R_{eqn}\left(\frac{L_n}{W_n}\right)C_g(4W + W) = 5R_{eqn}C_gL_n$$

To optimize the delay of the series of mixed gates in a logic path, we must set the fanout portion of the delay to be equal for all gates. That is,

$$\tau_{gate_j} FO_j = \tau_{gate_{(j+1)}} FO_{(j+1)}$$

where $FO_j = \frac{C_{j+1}}{C_j}$ and $FO_{j+1} = \frac{C_{j+2}}{C_{j+1}}$.

Optimizing Paths with Logic Effort

Derivation of Logical Effort

The *Logical Effort* (LE) is the ratio of the intrinsic time constant for a gate to the intrinsic time constant of an inverter. That is,

$$\boxed{LE \triangleq \frac{\tau_{gate}}{\tau_{inv}} = \frac{\left(R_{eff} C_{load} C_{in}\right)_{gate}}{\left(R_{eff} C_{load} C_{in}\right)_{inv}}}$$

Is C_{load} in the numerator always the same as the one in the denominator?

For example, the logical effort of a NAND gate would be τ_{nand}/τ_{inv}; the logical effort for an inverter is $\frac{\tau_{inv}}{\tau_{inv}} = 1$.

Two approaches of quickly calculating the logical effort:

(i) Set the delays of the inverter and the gate to be the same; then, take the ratio of the input capacitances.

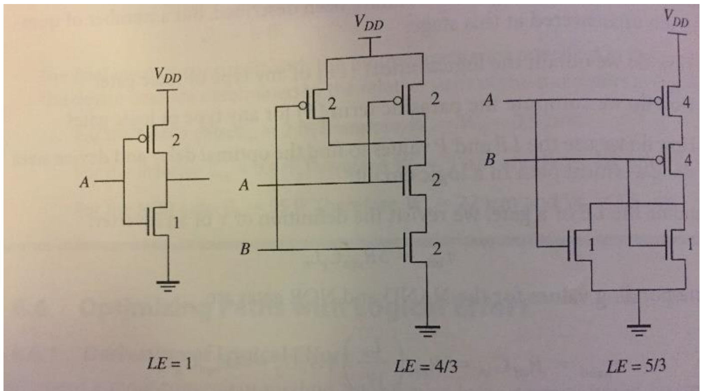


Figure 6.27
Logical effort for gates with equal delays.

In Figure 6.27, the NAND and NOR gates have already been sized to have the same delay as the inverter; this is achieved by making $\left(\frac{W_{p,eff}}{W_{n,eff}}\right)_{worst-case} = \frac{2}{1}$ for NAND and NOR gates. Therefore, we can simply use the input capacitance ratios.

Consider input A:

$$For\ the\ NAND\ gate: LE = \frac{(C_{in})_{nand}}{(C_{in})_{inv}} = \frac{2+2}{3} = \frac{4}{3}$$

$$For\ the\ NOR\ gate: LE = \frac{(C_{in})_{nor}}{(C_{in})_{inv}} = \frac{4+1}{3} = \frac{5}{3}$$

END

(ii) Set the input capacitances to be the same; then, take the delay ratio.

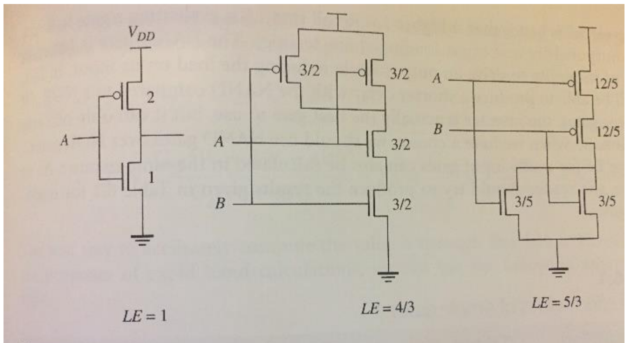


Figure 6.28
Logical effort using equal input capacitances.

The device sizes of Figure 6.27 have been uniformly scaled such that the input capacitances of all three gates are equal and are shown in Figure 6.28. This is possible because *the LE of a gate does not change if all devices are scaled uniformly*.

Check:

$$Inverter: C_{in} = C_g(2+1) = 3C_g$$

$$NAND: C_{in} = C_g\left(\frac{3}{2}+\frac{3}{2}\right) = 3C_g$$

$$NOR: C_{in} = C_g\left(\frac{12}{5}+\frac{3}{5}\right) = 3C_g$$

Since they all have the same input capacitance, we can take the ratio of the delays $\left(i.e., LE = \frac{(R_{eff}C_{out})_{gate}}{(R_{eff}C_{out})_{inv}}\right)$ to obtain the LE values. Assume $C_{out}(i.e., C_{load}\ for\ the\ gate)$ is the same for all of them.

$$LE_{nand} = \frac{\frac{1}{W_{n,eff,worst-case}}R_{eff}C_{out}}{R_{eff}C_{out}} = \frac{1}{W_{n,eff,worst-case}} = \frac{1}{\left(\frac{3}{2}\right)/2} = \frac{4}{3}$$

$$LE_{nor} = \frac{\frac{1}{W_{n,eff,worst-case}}R_{eff}C_{out}}{R_{eff}C_{out}} = \frac{1}{W_{n,eff,worst-case}} = \frac{1}{\left(\frac{3}{5}\right)} = \frac{5}{3}$$

This approach may be used where the input capacitances of two gates are known to be equal.

END

In summary,

$$\boxed{LE = \begin{cases} \frac{(C_{in})_{gate}}{(C_{in})_{inv}} & \text{gate is sized to have the same delay as the inverter} \\ \frac{1}{W_{n,eff,worst-case}} & \text{gate is scaled to have the same input capacitance as the inverter} \end{cases}}$$

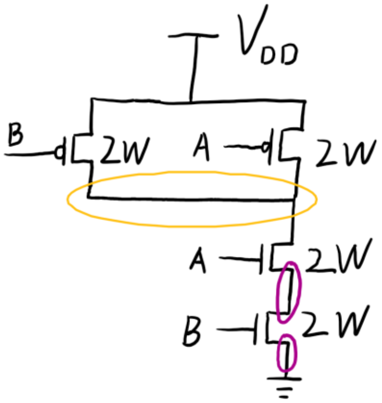
The gate with lower LE is better in terms of its ability to drive an output while reducing the load on its input. In fact, we will be able to produce a shorter delay with the lower-LE gates (e.g., NAND) compared to the higher-LE gates (e.g., NOR). So, when we have a choice, we should use NAND gates over NOR gates.

The parasitic term P is technology- and gate-dependent.

$$P = LE \times \gamma = LE \times \frac{C_{self}}{C_{in}} = LE \times \frac{C_{eff}(W_n + W_p)}{C_g(W_n + W_p)}$$

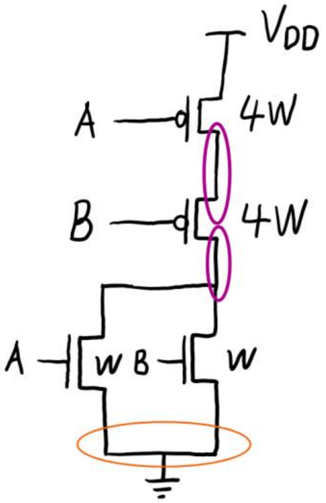
where $\frac{C_{eff}}{C_g} = \frac{1}{2}$.

Example: Calculate P_{nand} for the following circuit. Consider the shared source/drain regions in the layout.



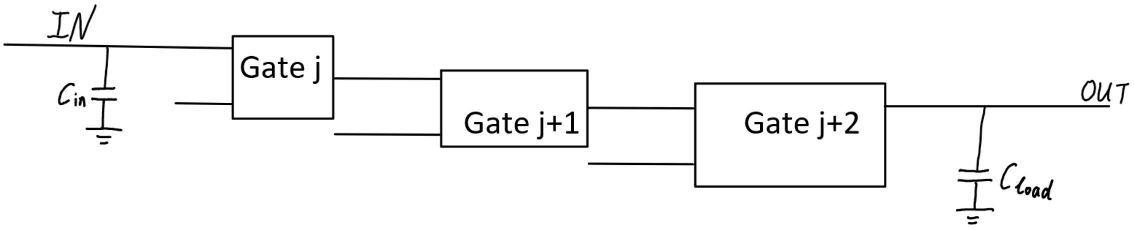
$$P_{nand} = LE_{nand} \times \gamma_{nand} = LE_{nand} \times \frac{C_{self}}{C_{in}} = \frac{4}{3} \times \frac{C_{eff}(2W + 2W + 2W)}{C_g(2W + 2W)} = \frac{4}{3} \times \frac{1}{2} \times \frac{3}{2} = 1$$

Example: Calculate P_{nor} for the following circuit. Consider the shared source/drain regions in the layout.



$$P_{nor} = LE_{nor} \times \gamma_{nor} = LE_{nor} \times \frac{C_{self}}{C_{in}} = \frac{5}{3} \times \frac{C_{eff}(W + 4W + 4W)}{C_g(W + 4W)} = \frac{5}{3} \times \frac{1}{2} \times \frac{9}{5} = \frac{3}{2}$$

Compute Optimal Gate Sizes along a Critical Path



Method 1: Without the Use of Logical Effort

1. Equalize the fanout portion of the delay.

$$\tau_{gate_j}\left(\frac{C_{j+1}}{C_{in}}\right)=\tau_{gate_{j+1}}\left(\frac{C_{j+2}}{C_{j+1}}\right)=\tau_{gate_{j+2}}\left(\frac{C_{load}}{C_{j+2}}\right)$$

2. Calculate the Fanout delay

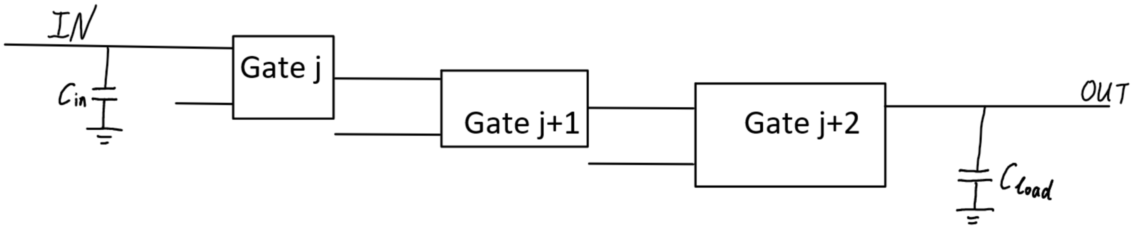
$$fanout\_delay=\sqrt[3]{\tau_{gate_j}\left(\frac{C_{j+1}}{C_{in}}\right)\times\tau_{gate_{j+1}}\left(\frac{C_{j+2}}{C_{j+1}}\right)\times\tau_{gate_{j+2}}\left(\frac{C_{load}}{C_{j+2}}\right)}=\sqrt[3]{\tau_{gate_j}\times\tau_{gate_{j+1}}\times\tau_{gate_{j+2}}\times\left(\frac{C_{load}}{C_{in}}\right)}$$

3. The input capacitance for each gate can be computed by setting the fanout delay to $fanout\_delay$.

$$\tau_{gate_{j+2}}\left(\frac{C_{load}}{C_{j+2}}\right)=fanout\_delay\Longrightarrow C_{j+2}=\frac{\tau_{gate_{j+2}}C_{load}}{fanout\_delay}$$
$$\tau_{gate_{j+1}}\left(\frac{C_{j+2}}{C_{j+1}}\right)=fanout\_delay\Longrightarrow C_{j+1}=\frac{\tau_{gate_{j+1}}C_{j+2}}{fanout\_delay}$$
$$\tau_{gate_j}\left(\frac{C_{j+1}}{C_{in}}\right)=fanout\_delay\Longrightarrow C_{in}=\frac{\tau_{gate_j}C_{j+1}}{fanout\_delay}$$

4. Solve for W_n and W_p by using $C_{in}=C_g(W_n+W_p)$.

Method 2: Use Logical Effort



We need to equalize the $LE\times FO$ components of the delay for all gates.

1. Compute the product of $LE\times FO$ for all the gates:

$$total\ path\ effort=LE_{gate_j}\times LE_{gate_{j+1}}\times LE_{gate_{j+2}}\times BE\times\left(\frac{C_{load}}{C_{in}}\right)$$

where $BE=1$ in this case.

2. Take the geometric mean of the result:

$$SE^{*} = Optimal\ Stage\ Effort = (total\ path\ effort)^{\frac{1}{N}} = \sqrt[3]{total\ path\ effort}$$

Stage Effort is the fanout portion of the delay; N represents the number of stages. The ***optimal*** value of the *total path delay* is known before the gate sizes have even been determined.

3. Compute the normalized total path delay:

$$D = N \times SE^{*} + \sum P \ = 3 \times (Stage\ Effort) + P_{gate_j} + P_{gate_{j+1}} + P_{gate_{j+2}}$$

That is

$$\boxed{D = N \times \sqrt[N]{LE_{gate_1} \times LE_{gate_2} \times LE_{gate_3} \times BE \times \left(\frac{C_{load}}{C_{in}}\right)} + \sum P}$$

4. The physical delay value is obtained by the following formula

$$\text{min\_path\_delay} = \tau_{inv} D$$

In short,

$$\boxed{\text{min\_path\_delay} = \tau_{inv}\left(3 \times \sqrt[3]{LE_{gate_1} \times LE_{gate_2} \times LE_{gate_3} \times BE \times \left(\frac{C_{load}}{C_{in}}\right)} + P_{gate_1} + P_{gate_2} + P_{gate_3}\right)}$$

By using the formula above, we can determine the minimum delay without sizing the gates. This is one of the key advantages of the LE approach. If the minimum possible delay is not within specifications, the logic can be modified, and the process repeated until a satisfactory solution is obtained. Once the target delay is achieved, gate sizing can be carried out.

5. Assuming that the delay is acceptable, we work backwards from the output to the input to compute the device sizes:

$$LE_{gate_{j+2}}\left(\frac{C_{j+3}}{C_{j+2}}\right) = SE^{*} \Longrightarrow C_{j+2} = LE_{gate_{j+2}}\left(\frac{C_{j+3}}{SE^{*}}\right)\ where\ C_{j+3} = C_{out}$$

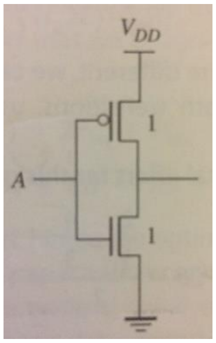
$$LE_{gate_{j+1}}\left(\frac{C_{j+2}}{C_{j+1}}\right) = SE^{*} \Longrightarrow C_{j+1} = LE_{gate_{j+1}}\left(\frac{C_{j+2}}{SE^{*}}\right)$$

$$LE_{gate_j}\left(\frac{C_{j+1}}{C_{in}}\right) = SE^{*} \Longrightarrow C_{in} = LE_j\left(\frac{C_{j+1}}{SE^{*}}\right)\ where\ C_{in} = C_j$$

LE for a Skewed Inverter

Problem:

What is the LE of this inverter?



Solution:

The delay and the input capacitance are not the same as the reference inverter. One of the two must be adjusted to obtain the LE.

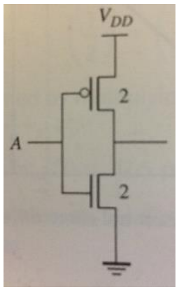
The rising and falling cases must be handled separately for this gate.

Method 1: Set the delays equal to that of the reference inverter and take the ratio of input capacitances.

(i) Falling case: fall delay is already the same as the regular inverter since it is 1.

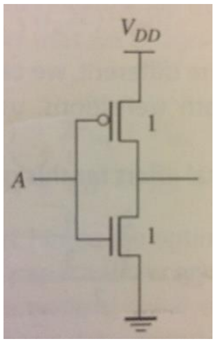
$$LF_F = \frac{C_{in}|_{gate}}{C_{in}|_{inv}} = \frac{1+1}{1+2} = \frac{2}{3}$$

Rising case: scale up all devices by $2\times$ to obtain the same rise delay:



$$LE_R = \frac{C_{in}|_{gate}}{C_{in}|_{inv}} = \frac{2+2}{1+2} = \frac{4}{3}$$

Method 2: Use the definition of logical effort:



$$LE = \frac{\left(R_{eff}C_{in}\right)_{gate}}{\left(R_{eff}C_{in}\right)_{inv}}$$

For the falling case:

$$LE_F = \frac{R_{eqn}\left(2C_gW\right)}{R_{eqn}\left(3C_gW\right)} = \frac{2}{3}$$

For the rising case:

$$LE_R = \frac{R_{eqp,gate}\left(2C_gW\right)}{R_{eqp,inv}\left(3C_gW\right)} = \frac{\left(2R_{eqn}\right)\left(2C_gW\right)}{R_{eqn}\left(3C_gW\right)} = \frac{4}{3}$$

since $R_{eqp,inv} = R_{eqn}\ and\ R_{eqp,gate} = 2R_{eqn}$.

Since the rising and falling LEs are different, we can optimize based on either one. But if we want to optimize both transitions, use the *average LE* to determine the transistor sizes.

Therefore, the average logical effort for this gate is

$$LE = \frac{LE_F + LE_R}{2} = \frac{\frac{2}{3}+\frac{4}{3}}{2} = 1$$

END

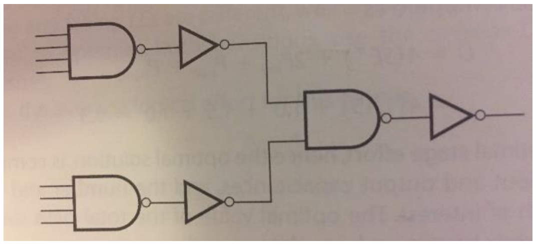
Designing an 8-Input AND Gate

Problem:

An 8-input AND gate is to be designed to drive a load of $200\ fF$ but is limited to an input capacitance of $20\ fF$. Since a single 8-input CMOS NAND gate is out of the question, choose two configurations that are more suitable and identify the solution with the fastest speed.

SOLUTION:

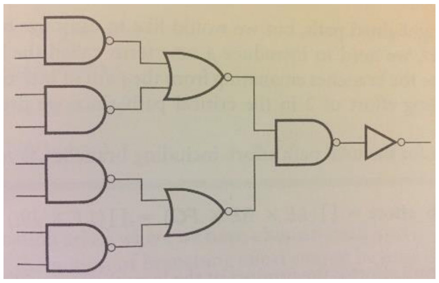
1. The first option is the NAND4-INV-NAND2-INV as shown below. Here the LE and parasitic terms are all known for these gates since they are standard gates.



$$D = N \times \sqrt[N]{LE_{NAND4} \times LE_{inv} \times LE_{NAND2} \times LE_{inv} \times \left(\frac{C_{load}}{C_{in}}\right)} + P_{NAND4} + P_{inv} + P_{NAND2} + P_{inv}$$

$$= 4 \times \sqrt[4]{2 \times 1 \times \frac{4}{3} \times 1 \times \left(\frac{200}{20}\right)} + 2 + \frac{1}{2} + 1 + \frac{1}{2} = 13$$

2. The second option is the NAND2-NOR2-NAND2-INV cascade.



$$D = N \times \sqrt[N]{LE_{NAND2} \times LE_{NOR2} \times LE_{NAND2} \times LE_{inv} \times \left(\frac{C_{load}}{C_{in}}\right)} + P_{NAND2} + P_{NOR2} + P_{NAND2} + P_{inv}$$

$$= 4 \times \sqrt[4]{\frac{4}{3} \times \frac{5}{3} \times \frac{4}{3} \times 1 \times \left(\frac{200}{20}\right)} + 1 + \frac{3}{2} + 1 + \frac{1}{2} = 13.3$$

Option 1 is better than option 2. We should always use delay calculation to determine if one configuration is superior to another.

END

Branching Effort and Sideloads

If a node has an additional load of fixed size, we treat it as a *sideload*.

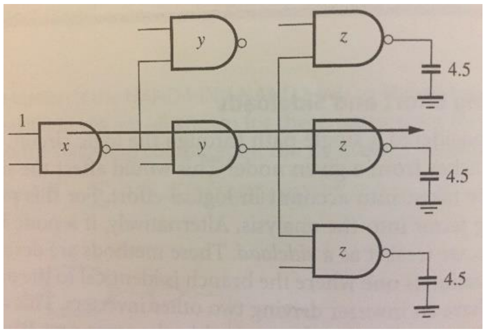
$$total\ path\ effort = \Pi(LE \times BE \times FO) = \frac{C_{load}}{C_{in}} \Pi(LE \times BE)$$

BE: branching efforts

The branching effort is often known in advance or can be estimated based on the circuit topology.

Branching Effort Problem:

Select gate sizes y and z to minimize delay in the highlighted path:



Solution:

$$BE = 2 \times 3 = 6$$

$$D = N \times \sqrt[N]{LE_{gate_1} \times LE_{gate_2} \times LE_{gate_3} \times BE \times \left(\frac{C_{load}}{C_{in}}\right)} + \sum P = 3 \times \sqrt[3]{LE^3_{NAND2} \times 6 \times \left(\frac{4.5}{1}\right)} + 3 \times 1$$

$$= 3 \times \sqrt[3]{LE^3_{NAND2} \times 6 \times 4.5} + 3 = 15$$

$$SE^* = \sqrt[3]{LE^3_{NAND2} \times 6 \times \left(\frac{4.5}{1}\right)} = 4$$

$$z = N_{branch} \times LE_{NAND2}\left(\frac{C_{out}}{SE^*}\right) = 1 \times \frac{4}{3}\left(\frac{4.5}{4}\right) = 1.5$$

$$y = N_{branch} \times LE_{NAND2}\left(\frac{C_{NAND2}}{SE^*}\right) = 3 \times \frac{4}{3}\left(\frac{4.5/3}{4}\right) = 1.5$$

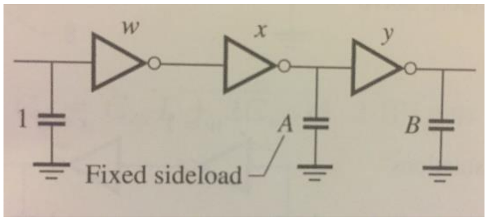
$$x = N_{branch} \times LE_{NAND2}\left(\frac{C_{NAND2}}{SE^*}\right) = 2 \times \frac{4}{3}\left(\frac{4.5/3}{4}\right) = 1$$

END

Computing Delay with Sideloads

Sideload: A known fixed load in a circuit along the path of interest.

Problem: Compute the gate sizes $w, x, and\ y$ for the following logic circuit to produce the minimum delay. Assume that $A = 8$ and $B = 64$. Here, A should be considered as a sideload.



Solution:

If we remove the sideload and use FO4 sizing rules, we would size the inverters as $w = 1, x = 4, and\ y = 16$. This satisfies the geometric relationship between consecutive stages and would produce the minimum delay.

If we now insert the sideload, we would derive the following delay equations

$$D = LE_{gate_{j-1}}\left(\frac{Size_{gate_j} + C_{sideload}}{Size_{gate_{j-1}}} + \gamma_{gate_{j-1}}\right) + \cdots$$

First two stages:

$$D_1 = LE_{inv}\left(\frac{x}{w} + \gamma_{inv}\right) + LE_{inv}\left(\frac{A+y}{x} + \gamma_{inv}\right)$$

$$\frac{\partial D_1}{\partial x} = \frac{1}{w} - \frac{A+y}{x^2} = 0 \Rightarrow \frac{x}{w} = \frac{A+y}{x}$$

Next two stages:

$$D_2 = LE_{inv}\left(\frac{A+y}{x} + \gamma_{inv}\right) + LE_{inv}\left(\frac{B}{y} + \gamma_{inv}\right)$$

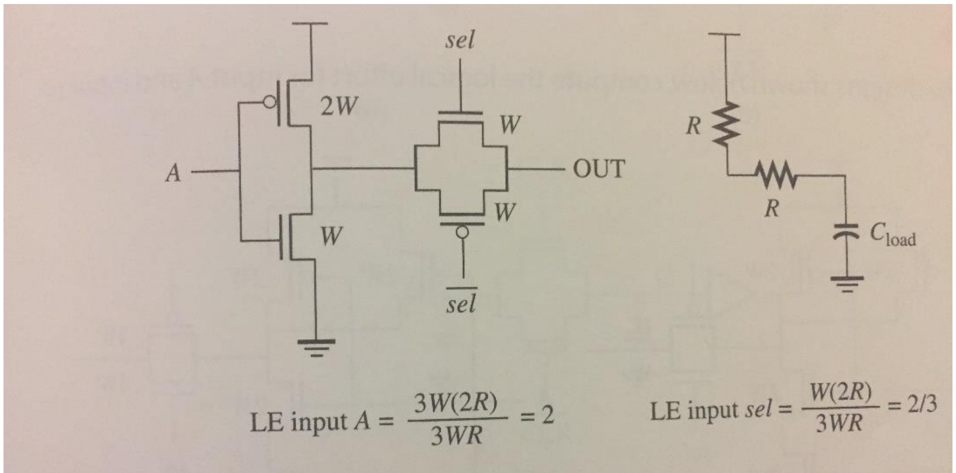
$$\frac{\partial D_2}{\partial x} = \frac{1}{x} - \frac{B}{y^2} = 0 \Rightarrow \frac{y}{x} = \frac{B}{y}$$

$$\left.\begin{array}{c}\dfrac{x}{w} = \dfrac{A+y}{x} \\ \dfrac{y}{x} = \dfrac{B}{y}\end{array}\right\} \Rightarrow \left\{\begin{array}{c}\dfrac{x}{1} = \dfrac{8+y}{x} \\ \dfrac{y}{x} = \dfrac{64}{y}\end{array}\right\} \Rightarrow \begin{cases} x = 5 \\ y = 18\end{cases}$$

END

How to use logical effort to estimate the delay of a single TG??????

Logical Effort with CMOS Transmission Gates (Page 332)



LE input $A = \frac{3W(2R)}{3WR} = 2$

LE input $sel = \frac{W(2R)}{3WR} = 2/3$

Figure 7.19
Logical effort computation for transmission gates.

We have three inputs: $A, sel, and\ \overline{sel}$. Since both sel and $\overline{sel}$ are related, we can compute the LE for the sel and A inputs.

For input A,

$$C_{in} = C_{gate_{inv}} = C_g(W + 2W) = 3WC_g$$

Given that the inverter and TG each have a resistance of R,

$$R_{eff,TG,input\ A} = 2R$$

$$LE\ input\ A \triangleq \frac{\tau_{gate}}{\tau_{inv}} = \frac{\left(R_{eff}C_{in}\right)_{gate}}{\left(R_{eff}C_{in}\right)_{inv}} = \frac{2R \times 3WC_g}{R \times 3WC_g} = 2$$

For input sel,

$$C_{in} = C_gW$$

The total path resistance (including the inverter, which inverter does this refer to?) is

$$R_{eff,TG,sel} = 2R$$

$$LE\ input\ sel \triangleq \frac{\tau_{gate}}{\tau_{inv}} = \frac{\left(R_{eff}C_{in}\right)_{gate}}{\left(R_{eff}C_{in}\right)_{inv}} = \frac{2R \times WC_g}{R \times 3WC_g} = \frac{2}{3}$$

Question: If I include the resistance of the inverter in the calculation of $R_{eff,TG,sel}$, then why do I not need to include the gate capacitance of the inverter in the calculation of C_{in}?

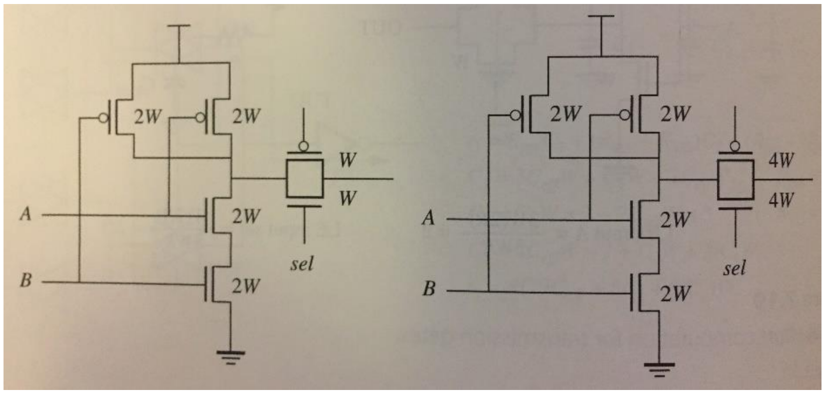
If we increase the size of the transmission gate, the LE of the input A decreases while the LE for the sel input increases. This makes sense intuitively since input A experiences a smaller output resistance without any change in its input capacitance, while input sel experiences an increase in its input capacitance but ***not*** a reduction in its output resistance.

END

Logic Effort for NAND Driving CMOS TG

Problem:

For the two designs shown below, compute the logical effort for input A and input sel.



Solution:

(i)

$$\frac{W_{eff,worst-case,p}}{W_{eff,worst-case,n}} = \frac{2}{1} \Longrightarrow R_n = R_p = R$$

$$LE\ input\ A \triangleq \frac{\tau_{gate}}{\tau_{inv}} = \frac{\left(R_{eff}C_{in}\right)_{gate}}{\left(R_{eff}C_{in}\right)_{inv}} = \frac{(R+R)(2W+2W)}{(R)(3W)} = \frac{8}{3}$$

$$LE\ input\ sel\ (including\ the\ inverter) \triangleq \frac{\tau_{gate}}{\tau_{inv}} = \frac{\left(R_{eff}C_{in}\right)_{gate}}{\left(R_{eff}C_{in}\right)_{inv}} = \frac{(R+R)(W)}{(R)(3W)} = \frac{2}{3}$$

(ii)

$$\frac{W_{eff,worst-case,p}}{W_{eff,worst-case,n}} = \frac{2}{1} \Longrightarrow R_n = R_p = R$$

$$LE\ input\ A \triangleq \frac{\tau_{gate}}{\tau_{inv}} = \frac{\left(R_{eff}C_{in}\right)_{gate}}{\left(R_{eff}C_{in}\right)_{inv}} = \frac{\left(R+\frac{R}{4}\right)(2W+2W)}{(R)(3W)} = \frac{5}{3}$$

$$LE\ input\ sel\ (including\ the\ inverter) \triangleq \frac{\tau_{gate}}{\tau_{inv}} = \frac{\left(R_{eff}C_{in}\right)_{gate}}{\left(R_{eff}C_{in}\right)_{inv}} = \frac{\left(R+\frac{R}{4}\right)(4W)}{(R)(3W)} = \frac{5}{3}$$

END