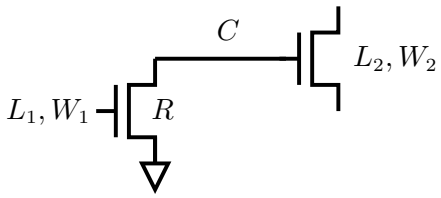
Handout: Transistor Sizing

We will approximate the time taken to switch a digital signal by a simple first order RC delay, where R is the resistance through which a capacitance C is switched. Consider driving the value of a signal to ground using a single n-type transistor as shown below.



The resistance of a transistor depends on the width and length of the device:

$$R \propto \frac{L_1}{W_1}$$

where L is the length of the transistor and W is the width of the transistor. The gate capacitance of a transistor depends on the width and length of the device:

$$C \propto L_2 \times W_2$$

which makes the RC delay for a transition on the wire:

$$RC \propto L_1 L_2 \frac{W_2}{W_1}$$

The capacitance of a transistor gate depends on the thickness of the gate oxide and its dielectric constant, and therefore does not depend on the type of transistor. However, the resistance does depend on the type of transistor due to the difference in electron/hole mobility. Therefore the constant of proportionality for the RC delay depends on the type of drive transistor; this (for us) translates to whether we are setting the signal to Vdd or GND, since we will only use p-type devices to set the output to Vdd and n-type devices to set the output to GND. From the equation, it is clear that increasing the length of either device only increases the signal delay and therefore we should make $L_1 = L_2 = L_{min}$, the minimum device length permissible by the technology. Therefore, we can write the RC delay expression as

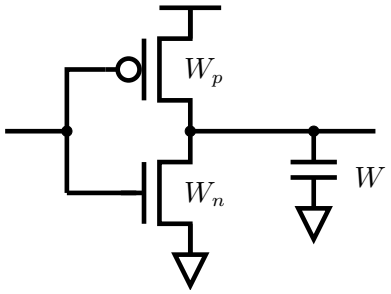
$$RC = \tau_n \frac{W_2}{W_1}$$

for an n-type drive transistor, where τ_n contains the constants of proportionality and the term L_{min}^2. Similarly, for a p-type transistor driving a load we can write

$$RC = \tau_p \frac{W_2}{W_1}$$

Since we are assuming minimum size devices, we will think of capacitances in units of transistor widths, with the actual capacitance being obtained by multiplying the width by L_{min} and the capacitance of a gate of size $1\lambda \times 1\lambda$.

Equal Transition Times. Given this simple model, we can determine how to size an inverter so as to make the delays for a $0 \to 1$ and a $1 \to 0$ transition equal. Consider the inverter shown below driving a load capacitance equal to W units of gate capacitance.



The delay for the $0 \to 1$ transition and $1 \to 0$ transition is given by

$$d(0 \to 1) = \tau_p \frac{W}{W_p}$$
$$d(1 \to 0) = \tau_n \frac{W}{W_n}$$

Equalizing the delays, we obtain the ratio of p-type to n-type transistor widths r^{max}:

$$r^{max} = \frac{W_p}{W_n} = \frac{\tau_p}{\tau_n}$$

Equalizing delays is *equivalent* to minimizing the maximum delay, under the constraint that $(W_p + W_n)$ is a constant.

Equal sum delay. For the next example, consider a transistor sizing problem where we have an inverter driving a load as earlier. Instead of equalizing the delays on the $0 \to 1$ and $1 \to 0$ transitions, consider optimizing the sum of the two delays. This may be important because, for instance, both delays appear as part of the timing budget for the cycle time of the system. The delay expression of interest is given by

$$d = \tau_n \frac{W}{W_n} + \tau_p \frac{W}{W_p}$$

The constraint that we introduce is that the total *input* switching capacitance is fixed; in other words, $W_p + W_n = C$ for some constant C. The delay can be re-written as

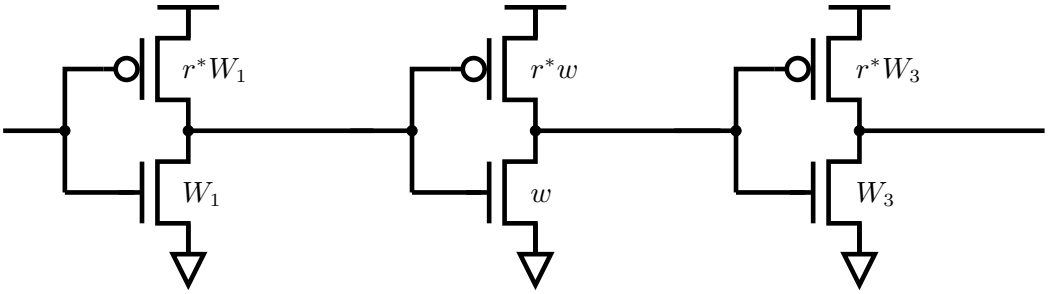
$$d = \tau_n \left(\frac{W}{W_n} \right) + \tau_p \left(\frac{W}{C - W_n} \right)$$

Differentiating this with respect to W_n and minimizing this expression, we obtain that the optimal ratio $r^+ = W_p/W_n$ that minimizes the sum of the two delays is given by:

$$r^+ = \sqrt{\frac{\tau_p}{\tau_n}} = \frac{W_p}{W_n}$$

The delay of the up-going transition will be r^+ times larger than the delay of the down-going transition. Therefore, minimizing the sum of the delays does not equalize the delays for the $0 \to 1$ and $1 \to 0$ transition for the inverter.

Three Inverters in Sequence. To begin tackling the problem of minimizing the delay through a number of gates in sequence, consider the simpler problem of optimizing the delay for three inverters in series. We assume that the first and third inverter have widths that are given, while we are allowed to adjust the widths of the transistors in the middle inverter to optimize the delay. We make the simplifying assumption that the ratio of p-type and n-type transistors is constrained so as to equalize the $0 \to 1$ and $1 \to 0$ transitions or to equalize the sum of the delays, depending on the circuit. We call the ratio selected r^*.



The delay as a function of the n-transistor width of the middle inverter is given by:

$$d(w) = \tau_n(1 + r^*)w/W_1 + \tau_n(1 + r^*)W_3/w$$

Note that while we have used $1 \to 0$ transitions for both the output of the first and second inverter, choosing $0 \to 1$ transitions for either of them will not change the result of the optimization since we have used the ratio r^* to constrain the relative delay of the $0 \to 1$ and $1 \to 0$ transitions. Minimizing this expression w.r.t. w, we obtain

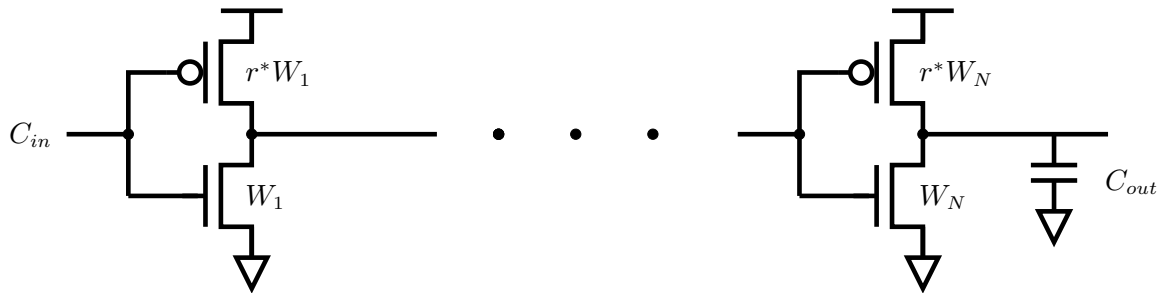
$$d'(w) = \tau_n(1 + r^*)/W_1 - \tau_n(1 + r^*)W_3/w^2 = 0$$
$$w = \sqrt{W_1 W_3}$$

Substituting this, we observe that the delay for the first transition is $\tau_n(1 + r^*)\sqrt{W_3/W_1}$, which is the *same* as the delay for the second transition! Also, we observe that we can write the widths as follows:

$$w = W_1 \times \sqrt{W_3/W_1}$$
$$W_3 = w \times \sqrt{W_3/W_1}$$

In other words, the transistor widths are in a geometric series.

Chain of Inverters. Using the results we have just derived, we can determine the optimal sizes for a chain of N inverters driving a load capacitance C_{out}, assuming that the first inverter in the chain has an input capacitance $C_{in} = (1 + r^*)W_1$.



Since the optimal delay will occur when each transition has equal delay (see above), and this happens when the inverter widths are in a geometric series, we know that:

$$W_N = s^{N-1}W_1$$

where s is the inverter width ratio. The delay for one transition is given by

$$d = \tau_n(1 + r^*)W_2/W_1 = \tau_n(1 + r^*)s$$

(Note: the inverter we choose to examine is not important, since all transitions will have equal delay as we have shown in the previous example.) The delay of the output inverter is given by

$$d_{out} = \tau_n \frac{C_{out}}{W_N} = \tau_n \frac{C_{out}}{s^{N-1}W_1}$$

At the delay optimal point, every transition has equal delay. Therefore, we know that $d = d_{out}$, and therefore:

$$\tau_n(1 + r^*)s = \tau_n \frac{C_{out}}{s^{N-1}W_1}$$

which translates to:

$$s^N = \frac{C_{out}}{(1 + r^*)W_1}$$

Since the input capacitance $C_{in} = (1 + r^*)W_1$, we can rewrite this as:

$$s^N = \frac{C_{out}}{C_{in}}$$
$$s = \sqrt[N]{\frac{C_{out}}{C_{in}}}$$

Therefore, the total delay through the N inverter stages is given by

$$d(N) = N\tau_n(1 + r^*)s = N\tau_n(1 + r^*)\sqrt[N]{\frac{C_{out}}{C_{in}}}$$

To determine the optimal number of stages, we minimize this expression with respect to N. Note that since $\tau_n(1 + r^*)$ is a constant, we can ignore this term for the minimization. Therefore, we obtain

$$d'(N)/(\tau_n(1 + r^*)) = \sqrt[N]{\frac{C_{out}}{C_{in}}} + N\sqrt[N]{\frac{C_{out}}{C_{in}}} \ln(C_{out}/C_{in}) \cdot \frac{-1}{N^2} = 0$$

which simplifies to

$$N = \ln(C_{out}/C_{in})$$

giving a scale factor s of

$$s = \left(\frac{C_{out}}{C_{in}}\right)^{1/ln(C_{out}/C_{in})} = e$$

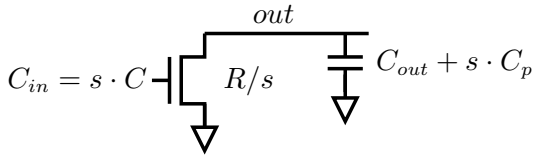
and a total delay of

$$d_{opt} = N\tau_n(1 + r^*)s = e\tau_n(1 + r^*)\ln(C_{out}/C_{in})$$

Logical Effort

The results presented above assumed that the gates being sized were inverters. We made the simplifying assumption that the ratio of p-type to n-type transistor widths was fixed, which allowed us to reason about the size of a gate in terms of a single parameter. This approach can be generalized to arbitrary gates; this generalized approach is called the method of logical effort.

Logical effort is a systematic way to size transistors in a circuit using the RC delay model. The model can be derived by considering a single n-fet driving a capacitive load.



We assume that the transistor W/L ratio can be scaled by a factor s, which effectively increases the gate capacitance of the transistor by a factor of s while reducing the "on-resistance" by a factor of s at the same time. The transistor is driving a load capacitance C_{out}, and a parasitic contribution from the source/drain capacitance of the transistor itself that is once again linear in the transistor size. The values C, C_p, and R correspond to the input capacitance, parasitic capacitance, and on resistance of a transistor of unit scale factor, i.e., where $s = 1$. The RC delay for *out*, denoted d_{out}, is given by

$$d_{out} = \frac{R}{s}\left(C_{out} + s \cdot C_p\right)$$

We can rewrite this as follows:

$$\begin{aligned} d_{out} &= \frac{R}{s}\left(C_{out} + s \cdot C_p\right) \\ &= \frac{RC_{out}}{s} + RC_p \end{aligned}$$

To try and separate technology-dependent optimizations from technology-independent ones, we normalize the delay by the term $R_{inv}C_{inv}$, where R_{inv} and C_{inv} are the R and C terms in the equation for an inverter with scale factor set to one (the "unit" inverter). This gives us:

$$\begin{aligned} d_{out} &= \frac{RC_{out}}{sR_{inv}C_{inv}} + \frac{RC_p}{R_{inv}C_{inv}} \\ &= \left(\frac{RC}{R_{inv}C_{inv}}\right) \cdot \left(\frac{C_{out}}{C_{in}}\right) + \frac{RC_p}{R_{inv}C_{inv}} \end{aligned}$$

(We used the fact that $s = C_{in}/C$.) Finally, we need to decide what transistor W/L ratio corresponds to a scale factor of one for every gate in our design. We pick the transistor width ratio that equalizes the drive strength of the gate to the unit inverter. (Note that this choice is arbitrary.) This means that the on resistance of a unit gate will match the on resistance of the unit inverter, since that is what limits the current drive. Therefore, $R = R_{inv}$, and we get:

$$\begin{aligned} d_{out} &= \left(\frac{RC}{R_{inv}C_{inv}}\right) \cdot \left(\frac{C_{out}}{C_{in}}\right) + \frac{RC_p}{R_{inv}C_{inv}} \\ &= \left(\frac{C}{C_{inv}}\right) \cdot \left(\frac{C_{out}}{C_{in}}\right) + \frac{C_p}{C_{inv}} \\ &= g \cdot h + p \end{aligned}$$

where $g = C/C_{inv}$ is called the logical effort of the gate, $h = C_{out}/C_{in}$ is called the electrical effort of the gate, and $p = C_p/C_{inv}$ is the parasitic delay. The product gh is called the effort. g is the

ratio of the input capacitance of the unit gate to the input capacitance of the unit inverter, h is the ratio of the output to input capacitance of the gate, and p is normalized parasitic capacitance of the gate.

The dynamic energy E can be measured in units of C_{inv}.

$$E_{out} = \frac{C_{out} + s \cdot C_p}{C_{inv}}$$

This can be re-written

$$\begin{aligned} E_{out} &= \frac{C_{out}}{C_{in}} \cdot \frac{sC}{C_{inv}} + sp \\ &= s(gh + p) \\ &= sd_{out} \end{aligned}$$

Delay Optimization: Three Gates. To optimize a chain of gates, we can write that the total delay is the sum of the delays of the gates along the path. In other words, we have

$$d = \sum_i g_i h_i + p_i$$

Consider three gates in series, where we are given the sizes of the first and third gates but are free to pick the scale factor that minimizes the total delay.

Since adjusting the scale factor for gate two affects both the output capacitance of gate one and the drive strength of gate two, we need to take both of these into account for delay minimization. Since parasitic terms do not contribute to the minimization problem, we can simply optimize the delay by examining:

$$D = g_1 h_1 + g_2 h_2$$

The logical efforts g_1 and g_2 are pure functions of the gate topology. The electrical efforts depend on the scale factor s, since h_1 is the ratio of the output capacitance of gate one (which is the input capacitance of gate two) to the input capacitance of gate one (fixed in this problem, denoted by C_{in}), and h_2 is the ratio of the output capacitance of gate two (fixed at C_{out}) to the input capacitance of gate two. Therefore, we obtain:

$$D = g_1(sC_2/C_{in}) + g_2(C_{out}/(sC_2))$$

where C_2 is the input capacitance of gate two with unit scale factor. Minimizing d, we get:

$$\frac{dD}{ds} = g_1 C_2/C_{in} - g_2 C_{out}/C_2 \frac{1}{s^2} = 0$$

which gives us $sC_2 = \sqrt{C_{in} C_{out} g_2/g_1}$ as the input capacitance of gate two. Substituting this into the delay equation, we observe that

$$\begin{aligned} D &= g_1 \frac{\sqrt{C_{in} C_{out} g_2/g_1}}{C_{in}} + g_2 \frac{C_{out}}{\sqrt{C_{in} C_{out} g_2/g_1}} \\ &= \sqrt{g_1 g_2 C_{out}/C_{in}} + \sqrt{g_1 g_2 C_{out}/C_{in}} \end{aligned}$$

That is, the optimal scale factor equalizes the effort per stage, i.e., $g_1 h_1 = g_2 h_2$.

Path Delay Optimization. In a delay optimization problem with N stages, we conclude that the effort per stage must be equal. Therefore, when we optimize a path, we can write

$$\begin{aligned} d &= \sum_{i=1}^{N} g_i h_i + p_i \\ &= Nf + P \end{aligned}$$

where $f = g_i h_i$ for any choice of $i = 1, 2, \ldots, N$ is the effort per stage, and $P = \sum_{i=1}^{N} p_i$ is the total parasitic delay along the path. The only constraint we have is that the input capacitance for gate one is fixed at C_{in}, and the final output capacitance for gate N is fixed and is given by C_{out}.

If the gates are in a linear chain, then the output capacitance of a gate is the input capacitance of the next gate, and we use C_i to denote the input capacitance of the ith gate (In terms of the scale factor for gate i, $C_i = s_i g_i C_{inv}$). Therefore, $h_i = C_{i+1}/C_i$. Since $f = g_i h_i = g_i C_{i+1}/C_i$, we conclude that $C_i = (f C_{i+1}/g_i)$. Therefore,

$$\begin{aligned} C_i &= \frac{f}{g_i} C_{i+1} \\ &= \left(\frac{f}{g_i}\right)\left(\frac{f}{g_{i+1}}\right) C_{i+2} \\ &= \frac{f^{N+1-i}}{\prod_{k=i}^{N} g_k} \cdot C_{out} \end{aligned}$$

Using the fact that $C_1 = C_{in}$, we obtain:

$$C_{in} = \frac{f^N}{\prod_{k=1}^{N} g_k} C_{out}$$

that gives us:

$$f^N = \left(\prod_{k=1}^{N} g_k\right)\left(\frac{C_{out}}{C_{in}}\right)$$

The quantity $\prod_{k=1}^{N} g_k$, denoted G, is called the path logical effort, and the quantity C_{out}/C_{in}, denoted H, the path electrical effort. The product $GH = F$ is called the path effort. The total delay along the path is given by $Nf + P = NF^{1/N} + P$.

Branching Paths. If the path is not linear, then the output capacitance of the ith gate is not the input capacitance of the $(i+1)$th gate, but instead is multiplied by some factor that is called the branching effort, b_i. (Note that this assumes that all branches are evenly scaled for the value b_i to be constant.) The term b_i corresponds to the ratio of capacitance that switches to the capacitance along the path being optimized. Denoting $B = \prod_i b_i$, the total effort F is given by GHB.