4. Inverter speed and propagation delay

In this chapter we are going to look at the switching speed of the CMOS implementation of the inverter and develop some models for the propagation delay. As shown in the introduction to chapter 3 a delay model can be added to the Boolean description of the inverter so that the Boolean output is available only after a certain propagation delay. See Fig. 4.1.

![Fig. 4.1. An inverter with a delay model.](image)

4.1 Definition of propagation delay

In any electronic implementation of an inverter there is a delay between the switching of the input and the switching of the output. The rise and fall delays, $t_{pdr}$ and $t_{pdf}$, respectively, are defined in Fig. 4.2.

![Fig. 4.2. Propagation delay and rise/fall times [From Weste & Harris].](image)
The propagation delay is usually defined at the 50% level, but sometimes the propagation delay can be defined at other voltage levels. Also defined in this figure is the rise and fall times, $t_r$ and $t_f$, respectively. The rise and fall times are usually measured between the 10% and 90% levels, or between the 20% and 80% levels as in the figure.

A simplified illustration of the propagation delay with input and output voltages approximated as ramps is shown in Fig. 4.3. The propagation delay of the inverter under different conditions can be thoroughly analyzed using circuit simulations. However, before we do that we should develop simple models suitable for hand calculations using paper and pencil to develop our basic understanding of the switching behavior. Therefore, in this section we will discuss analytical models that can be used for predicting the propagation delay through hand calculations or timing estimation tools. Also, without approximate models for back-of-the-envelope calculations we are unlikely to catch inevitable bugs in our simulation model. Through simple analytical models the good engineer develops their physical intuition to rapidly predict the circuit behavior [Harris].

![Fig. 4.3. Propagation delay with input and output voltages approximated as ramps.](image)

### 4.1.1 The transient response

The non-zero propagation delay is due to the capacitive load at the output node and the limited current driving capability of the logic gate. To calculate the propagation delay we must solve a differential equation describing the output voltage as a function of time. The solution of the differential equation is called the transient response, and the delay can be found as the time when the output reaches $V_{DD}/2$. 

63
The differential equation describing the current $I$ for charging a capacitor with a capacitance $C$ when the voltage $V$ is changing is given by

$$I = C \frac{dV}{dt}.$$  \hspace{1cm} (4.1)

To get any further we must have a picture of the CMOS inverter and its electrical model during the part of the charging and discharging of the load capacitor that is relevant for the propagation delay estimation. This is illustrated by Fig. 4.4 where the two MOSFETs are represented by current sources. For simplicity we will limit ourselves to calculating the step responses for $v_{IN}=V_{DD}$ and $V_{SS}$, respectively. The saturation currents $I_{DSAT,P}$ and $I_{DSAT,N}$ are then the maximum saturation currents that the MOSFETs can deliver at the given supply voltage $V_{DD}$ of the implementation technology.

Fig. 4.4. Charging and discharging a load capacitor through MOSFET constant current sources.

Solving the differential equation in (4.1) for a constant-current source is of course very simple since the voltage will change linearly with time. The propagation delay – that is the time it takes to remove the charge $C_L V_{DD}/2$ by means of a constant-current source $I_{DSAT}$ - is then given by

$$t_{pd} = \frac{C_L V_{DD}}{2 I_{DSAT}}.$$  \hspace{1cm} (4.2)

Hence, we have developed a simple step response delay model. However, the input voltage is most often not a step function but rather a voltage with a certain rise time. During this input voltage rise time, a current less than the full $I_{DSAT}$ is flowing. Therefore, the propagation delay also depends on the...
input rise time which must be considered in a detailed delay model. Usually, the input and output voltages are approximated by voltage ramps as already discussed in Fig. 4.3.

### 4.1.2 The RC model and its effective resistance

However, for obtaining a simple first order model for hand calculations we will assume that the input rise/fall time is equal to the output fall/rise time. Circuit simulations have shown that in this case the propagation delay increases by approximately 40% [Hedenstierna & Jeppson]. Being conservative engineers eager to increase model safety margins, we therefore simply add another 40% to our propagation delay model obtaining the following modified delay model

\[
  t_{pd} = 0.7 \frac{C_{VDD}}{I_{DSAT}}.
\]  

(4.3)

If we define an effective rise/fall resistance \( R_{eff} = V_{DD}/I_{DSAT} \) we can write our propagation delay model as

\[
  t_{pd} = 0.7 R_{eff} C_L.
\]  

(4.4)

The delay model is now the same as the delay model for a capacitor being charged or discharged through a resistor with resistance \( R_{eff} \). This is excellent, since now we can associate our inverter with an RC circuit where the load capacitance \( C_L \) is switched from rail-to-rail through a resistor representing the charging/discharging MOSFET. The delay model is shown in Fig. 4.5 where the load capacitor is being charged to \( V_{DD} \) through an effective resistance \( R_{eff,P} \) representing the p-channel device, and discharged to \( V_{SS} \) through an effective resistance \( R_{eff,N} \) representing the n-channel device. If the two MOSFETs have the same effective resistance, the RC model circuit can be further simplified as shown in the figure. In this case, the rise and fall delays are equal and we do not have to make separate rise and fall calculations which is a nice simplification.

The definition of the effective resistance is shown in Fig. 4.6. Instead of using the MOSFET current model marked \( I_{DSAT} \) during the discharge from \( V_{DD} \) to \( V_{SS} \) we assume a more conservative resistive current model represented by the dashed line.
Once we have established the knowledge of representing the inverter output by an RC circuit and a propagation delay of $0.7RC$ we can make one more simplification. As stated by Weste and Harris in their textbook [Integrated Circuit Design, 4th edition, p107], the 0.7 prefactor is cumbersome. Since the effective resistance $R$ is an empirical parameter anyway, one might as well incorporate the factor of $\ln 2$ to define a new effective resistance $R' = R \ln 2$. For the sake of convenience, we usually drop the prime symbols and just write

$$t_{pd} = RC. \quad (4.5)$$

where the effective resistance is chosen to give the correct delay.
4.1.3 Model validity and switching trajectories

Weste & Harris have a somewhat different way of calculating the effective resistance than I have shown here, but the end result is about the same. If one considers the switching event in more detail [as they do in their section 3.3.7] one finds the following switching trajectories as illustrated in Fig. 4.7. The switching trajectory for a step input is illustrated in the top figure, while the switching trajectory for a ramp input is illustrated in the bottom figure.

In the case of an input step voltage, the discharging current increases instantaneously from zero to \( I_{DSAT} \) while the output voltage remains at \( V_{DD} \).

![Fig. 4.7. MOSFET switching trajectories for the step and ramp responses.](image-url)
As the discharge of the capacitor proceeds and the output voltage decreases, the switching trajectory follows the blue line until the capacitor is fully discharged. In the case of an input ramp voltage, the discharging current increases slowly from zero to $I_{DSAT}$ while $v_{OUT}$ is decreasing. For slow enough input ramps the MOSFET might even have left the saturation region before the input ramp reaches $V_{DD}$. The switching trajectory for the effective RC circuit is of course along the dashed line as before.

In conclusion, we should of course be aware of that the RC-model is just that – a model. As illustrated in Fig. 4.8, it does not in any way represent the correct charge/discharge behavior, but it results in the same approximate propagation delay. Therefore, it is quite a useful model in its simplicity, particularly for first order estimates using paper and pencil.

![Fig. 4.8. Comparison between the output ramp-response and the RC model.](image)

4.2 The complete inverter two-port RC model

To round off this discussion on the RC model and the propagation delay let us illustrate the two-port model of the CMOS inverter that we have developed. The previous discussion mainly concerned the driving properties of the inverter. However, we must also consider the capacitive properties of the inverter by adding the input and output capacitances to the two-port model as shown in Fig. 4.9. The inverter input capacitance is of course the
sum of the two intrinsic MOSFET gate capacitances, while the output capacitance is the sum of the parasitic MOSFET drain capacitances.

\[ t_{pd,\text{pair}} = t_{ib} + t_{df} = (R_{\text{eff},p} + R_{\text{eff},n})(C_D + C_G), \quad (4.6) \]

where \( C_G \) is the input capacitance of the inverter and \( C_D \) is the parasitic output capacitance. Using the input gate capacitance as references, the RC delay can be written

\[ t_{pd,\text{pair}} = t_{ib} + t_{df} = C_G \left( R_{\text{eff},n} + R_{\text{eff},p} \right) \left( \frac{C_D}{C_G} + 1 \right). \quad (4.7) \]
Usually $p = C_D / C_O$ is considered a constant and most often this constant is equal to one, i.e. $p=1$. The second term, being equal to one, indicates that each inverter is loaded by one other identical inverter.

The fanout-of-4 (FO4) delay illustrated by the example in Fig. 4.11 can be calculated similarly. We obtain the following average rise/fall delay

$$t_{pd,ave} = \frac{t_d + t_f}{2} = C_G \left( \frac{R_{eff,p} + R_{eff,N}}{2} \right) (p + 4),$$

(4.8)

In general terms, for any fanout of $f$, the average delay is given by:

$$t_{pd,ave} = \frac{t_d + t_f}{2} = C_G \left( \frac{R_{eff,p} + R_{eff,N}}{2} \right) (p + f).$$

(4.9)

In the topmost example of Fig. 4.11, the FO4 is due to branching while in the bottommost figure it is due to sizing. Of course both delay expressions are conveniently simplified if the inverter is electrically symmetrical, i.e. if $R_{eff,p} = R_{eff,N}$

$$t_{pd,ave} = \begin{cases} R_{eff,C_G} (p + 4) & \text{FO4 delay} \\ R_{eff,C_G} (p + f) & \text{general delay} \end{cases}$$

(4.10)

Fig. 4.11. The fanout-of-four (FO4) delay.
4.3 Sizing the p-channel device with respect to the n-channel device.

For hand calculations, the most convenient inverter model assumes that the effective resistances of the p-channel and n-channel devices are equal. This assumption yields equal rise and fall delays. In a real design this might not be the case, and this could then be considered in a more detailed second-order calculation or be left to circuit simulations. Therefore, for first-order hand calculations we will always assume equal nMOS and pMOS driving capabilities.

However, the question we will address now is whether exists an optimum aspect ratio between the p-channel and n-channel devices that yield a minimum delay. The way to size a MOSFET is of course to modify its channel width, \( W \). If the width of a MOSFET is increased, its resistance will decrease and the driving capability increase correspondingly. On the other hand, the MOSFET capacitances increase when the device is made wider, since

\[
R_{\text{eff}} \sim \frac{L}{W} \quad \text{and} \quad C_G \sim WL. \tag{4.11}
\]

For this reason, \( R_{\text{eff}} \) is usually given as ohms/unit width \([\Omega \mu\text{m}]\) while \( C_G \) is given as farads per unit width \([\text{F}/\mu\text{m}]\). The unit width is often one micron of transistor width, but maybe it will change to nanometer or tens of nanometer for nanoscale MOSFET devices. If we now consider individual n-channel and p-channel MOSFET devices of the same unit width, they usually have the same gate capacitance but different effective resistances. This is because n-channel devices have a higher driving capability due to the higher electron mobility. However, we can compensate for this by making the p-channel device \( x \) times wider. Then we obtain the following RC models

\[
\begin{cases} 
    R_{\text{eff}} & \text{for the n-channel device} \\
    \frac{\mu R_{\text{eff}}}{x} \times C_G & \text{for the p-channel device}
\end{cases} \tag{4.12}
\]

where \( \mu \) models the higher driving capability of the n-channel device due to the higher electron mobility. Using this notation, the inverter pair delay now becomes
This expression for the pair delay has a minimum for $x = \sqrt{\mu}$, a fact that is easily proved by taking the derivative and setting it equal to zero. Therefore, the p-channel device should be widened by a factor of $\sqrt{\mu}$ to minimize the average delay.

We can also see that $x = 1$ and $x = \mu$ result in the same pair delay. These two extremes represent the geometrically symmetrical inverter where the two MOSFETs are of equal size, but have different driving capabilities, and the electrically symmetrical inverter where the two MOSFETs have the same driving capability but are of different size. These two cases are illustrated by the inverter layouts in Fig. 4.12.

Between these two $x$-values there is obviously a $\mu$-value that yields the minimum delay.

Fig. 4.12. Sizing the p-channel MOSFET with respect to the n-channel device.

For simplicity, we are going to choose $x = \mu$ in all our hand calculations so that we only have to calculate one value representing both the rise and the fall delay. However, from this reasoning we can also understand why p-channel devices sometimes are not fully scaled to $x = \mu$. Often we can see designs
where \( x = 2 \) even if \( \mu \) is equal to 2.3 or 2.5. It is also worth noting that the minimum delay for \( x = \sqrt{\mu} \) is only about 3 per cent less than the average delay obtained when \( x = 1 \) or \( x = \mu \) so to some extent this minimum delay exercise is only of academic interest while it still provides some useful insight into transistor sizing.

### 4.4 Linear delay model (LDM)

For simplicity, we will from now on only consider electrically symmetrical inverters where the two MOSFETs have the same driving capabilities. The equal rise and fall delays can then be written

\[
d_t = RC \left( p + f \right), \tag{4.14}
\]

where \( C \) is the inverter input capacitance representing the loading properties of the inverter and \( R \) is the effective resistance representing the internal source resistance of the inverter seen as a driver voltage or current source. Often it is convenient to consider separately the intrinsic delay \( RC \), neglecting the parasitic output capacitance, and the relative delay

\[
d = p + f. \tag{4.15}
\]

Here \( p \) is the relative parasitic delay and \( f \) is the relative fanout delay (or effort delay) of the inverter. The advantage of the normalized delay is that it is technology independent. Empirically, \( p \) has found to be roughly independent of the technology node\(^4\), a fact that makes normalized delay optimizations extremely valuable. Considering the width and length dependencies of \( R \) and \( C \) in (4.11) the intrinsic \( RC \) product can easily be shown to scale as

\[
RC \sim \frac{L^2}{V_{DD}}. \tag{4.16}
\]

Some typical values for the FO4 delay for different technology nodes are shown in Fig. 4.13.

\(^4\)Typically, \( p \) used to be somewhat larger than one (\( p=1.5 \)) but for technology nodes with shallow trench isolation (STI) \( p \) is now less than one (\( p=0.5 \)). Generically, \( p=1 \) is used in first order calculations.
4.4.1 Limitations to the linear delay model

The largest source of error in the linear delay model is the input slope effect. Fig. 4.14 shows an FO4 inverter driven by ramps with different edge rates (slopes). Here, it is shown that the propagation delay increases for longer input edges. Hedenstierna and Jeppson showed that the delay increased linearly with the input edge rate. This linear relationship is shown in Fig. 4.14b. Accounting for input slopes is important for accurate timing analysis, but is generally more complex than necessary for hand calculations. [Weste & Harris].

Remember that when we derived our linear delay model we calculated the step input delay and then we added another 40% to account for a typical situation when the input and output edge rates are about equal. According to this reasoning the propagation delay could be written

\[ t_{pd} = 0.5RC + 0.2T_{input\_edge}. \]  \hspace{1cm} (4.17)

Assuming a typical output edge rate of twice the step delay, \( T_{output\_edge}=RC \), we arrived at the familiar delay model for equal input and output edges

\[ t_{pd} = 0.7RC. \]  \hspace{1cm} (4.18)
Consequently, a simple approach to extend the delay model is by adding a term reflecting the input slope as in (4.17). Considering that the load capacitance consists of the parasitic capacitance $C_D$ intrinsic to the inverter and an external load capacitance $C_L$, the delay model can now be written

$$t_{pd} = \frac{RC_{\text{intrinsic}} + \text{resistance} \cdot C_L + \text{slope} \cdot T_{\text{input \_ edge}}}{\text{resistance}}.$$  \hspace{1cm} (4.19)

where $RC_{\text{intrinsic}}$ is the intrinsic $RC$ delay due to the parasitic drain capacitances, $\text{resistance}$ is an effective resistance seen by the external load capacitance and $\text{slope}$ is a factor modeling the influence of the input edge rate. This delay model is illustrated in Fig. 4.15 where the propagation delay is plotted vs. fanout and input edge rate.

If we assume that the input edge rate is proportional to the delay of the previous stage, the delay model can be written as in Weste & Harris

$$\text{delay} = \text{intrinsic} + \text{resistance} \cdot C_L + \text{slope} \cdot \text{previous \_ delay}.$$  \hspace{1cm} (4.20)

In essence, what this model says is that three parameters, intrinsic [ns], resistance [ns/pF], and slope, are needed to characterize the inverter delay for appropriate timing analysis. An example of such a linear delay model where the propagation delay depends linearly on both the fanout and the input edge is shown in Fig. 4.15.
Fig. 4.15. The linear delay model vs. fanout and input edge rate.

The simplified 0.7RC model falls more or less diagonally across this delay plane. For any given fanout there is certain input edge that is equal to the output edge, and for any given input edge there is a certain fanout that produces an output edge equal to the input edge. For these fanout and input edges the two models produce the same delay. This is shown in Fig. 4.16.

Fig. 4.16. The linear delay model compared to the 0.7RC model.
### 4.4.2 Nonlinear delay models (NLDM)

Linear delay models are not accurate enough to handle the wide range of input slopes and loads found in synthesized circuits, so they have largely been superseded by nonlinear delay models. A nonlinear delay model (NLDM) looks up the delay from a table based on the load capacitance and the input slope. Separate look-up tables (LUTs) are used to look up rise and fall delays of an inverter or logic gate in general. Fig. 4.17 shows an example of a nonlinear delay model for the fall delay of an inverter. The timing analyzer uses interpolation when a specific load capacitance or input slope is not found in the table making the model a piecewise linear model. [This paragraph is copied from Weste & Harris, p 132.]

Fig. 4.18 illustrates the historical trends in microprocessor cycle time based on chips reported at the International Solid-State Circuits Conference. Early processors operated at close to 100 FO4 delays per cycle. The Alpha line of microprocessors from Digital Equipment shocked the staid world of circuit design in the early nineties by proving that clock cycles below 20 FO4 delays were possible.

![NLDM Graph](image)

*Fig. 4.17. The nonlinear delay model (NDLM) is a piecewise linear model where the delay is stored in a look-up table. Data from Table 3.7 in W & H.*
By the late 1990s, Intel and AMD marketed processors primarily on frequency. The Pentium II and II reached about 20-24 FO4 delays/cycle. The Pentium 4 drove cycle times down to about 10 FO4 at the expense of a very long pipeline and enormous power consumption. Microarchitects predicted that performance would be maximized at a cycle time of only 8 FO4 delays/cycle.

The short cycle time came at the expense of vast numbers (20-30) of pipeline stages and enormous power consumption (nearly 100 W). As a consequence, power became as important as performance specifications. The number of gates per cycle rebounded to a more power-efficient point. It was observed that 19-24 FO4 delays per cycle provides a better trade-off between performance and power.

Application-specific integrated circuits have generally operated at much lower frequencies (200-400 MHz) so that they can be designed more easily. Typical ASIC cycle times are 40-100 FO4 delays per cycle, although performance-critical designs sometimes are as fast as 25 FO4s.

### 4.4.3 Usefulness of relative delay model

The rest of this chapter will be spent on demonstrating the usefulness of the relative, technology independent, delay,

\[
d = p + f.
\]
The relative delay model is very useful in optimization problems because the same calculations are valid for most, if not all, technology nodes. In this section we will consider how to minimize the delay when the load capacitance is much larger than the input capacitance of the driver inverter. The problem is shown in Fig. 4.19 where an inverter with parameters $R$, $C$, and $p$ is shown loaded by a load capacitance $x$ times larger than the inverter input capacitance. These parameters indicate that the inverter has an input capacitance $C$, a parasitic output capacitance $pC$, and a driving capability given by the internal voltage source resistance $R$.

![Fig. 4.19. Inverter driving a large capacitor $C_L$ with and without buffer.](image)

The relative delay of this configuration is given by

$$d = p + x. \quad (4.22)$$

Now, if $x >> 1$ there will be a long delay and an output edge much longer than the input edge. This problem of unbalanced input/output edge rates can be solved by inserting a buffer between the inverter and the capacitive load. A buffer is an inverter with a larger driving capability. An example of such a buffer with twice the original driving capability is shown in Fig. 4.20. However, as a consequence the input capacitance of the buffer is scaled by the same sizing factor.

Assuming that the buffer has a driving capability $f$ times larger than the inverter, the relative delay is easily found by adding the inverter and buffer stage delays,

$$d = p + f + p + \frac{x}{f}. \quad (4.23)$$
This delay expression shows how the capacitive load has been distributed between the inverter and the buffer. Furthermore, it can easily be shown that this delay has a minimum when both inverters carry the same capacitive load relative to their driving capability. This is another way of saying that they both should have the same fanout. This is the case when

$$f = \frac{x}{f}, \text{ i.e. for } f = \sqrt{x}.$$  \hfill (4.24)

Inserting a buffer will decrease the delay as soon as $x$ is larger than a certain value given by

$$2(p + \sqrt{x}) < p + x,$$ \hfill (4.25)

which is true for

$$x > p + 2 + \sqrt{(p + 2)^2 - 1}, \text{ i.e. for } x > 5.8 \text{ if } p=1.$$ \hfill (4.26)

Similarly, inserting two inverters (that is a non-inverting buffer) is faster than inserting only one inverter already for such low values as $x>22$. The insertion of a non-inverting buffer with two inverters is shown in Fig. 4.21. Since the RC product of an inverter is independent of its size, the relative delay can be found by adding the three relative inverter delays

---

\textit{Fig. 4.20. Inverter sized to double driving capability.}
Again, it is relatively easy to show that minimum delay is obtained for equal tapering factors, i.e. for $f_1 = f_2 = f_3 = \sqrt[3]{x}$. This result is obtained by simply taking the derivatives of the delay with respect to the two independent tapering factors $f_1$ and $f_2$. As shown in Fig. 4.22, the non-inverting two-stage buffer solution yields the minimum delay in the load range $22 < x < 82$. 

**Fig. 4.22.** Relative delay for one, two, and three buffer inverters vs. the fanout.
4.4.4 The optimum tapering factor

In the previous section we showed that propagation delay can be minimized by inserting buffers when driving large capacitors. We showed that for each large load capacitor there is a certain number of buffers that will minimize the delay. In this section we will approach the general problem of finding the optimum tapering factor. Once the optimum tapering factor is known the optimum number of inverters in the buffer can be found. The problem we are going to discuss here is illustrated in Fig. 4.23.

In this figure, a buffer of \( N-1 \) inverters is inserted between a small reference inverter and a large capacitive load. Again, it is easy to show that minimum delay is obtained for equal tapering factors, i.e. for \( f = \sqrt{x} \). The total delay is then found by adding \( N \) identical stage delays corresponding to the first inverter and the \( N-1 \) stage buffer delays,

\[
d = N(p + f),
\]

As we found above, the number of stages \( N \) is related to the tapering factor through \( f = \frac{\sqrt{x}}{N} \). Solving for \( N \) we obtain \( N = \frac{\ln x}{\ln f} \). Inserting this expression for \( N \) into (4.29) we obtain,

\[
d = \frac{\ln x}{\ln f}(p + f).
\]

Taking the derivative with respect to \( f \) we obtain the following implicit expression for the optimum tapering factor \( f \):

\[
\frac{\ln x}{\ln f^2} = \frac{p}{f^2}.
\]
\[
\ln f = \frac{p + f}{f}.
\] (4.30)

For negligible parasitic capacitances on the inverter output, i.e. \(p=0\), we obtain the now “classical” solution for the optimum tapering factor \(f=e\). Here, \(e\) is the base of the natural logarithm - an irrational and transcendental number approximately equal to 2.72 [Wikipedia]. For the “generic value” of \(p=1\), a tapering factor of \(f=3.6\) is more appropriate. For a modern CMOS process with shallow trench isolation, where \(p=0.5\) is a more typical value we find an optimum tapering factor of 3.2. For simplicity, one usually regards \(f=4\) as the optimum tapering factor, a value that is more in line with the FO4 delay.

Once \(f\) is known we also know how many buffer inverters to insert. Since this number has to be an integer, and preferably an even number, the practical value for the optimum tapering factor to be used must be adjusted to match the buffer inserted for the problem at hand.

The optimum tapering factor as a function of the parasitic delay \(p\) is shown in the left-hand diagram of Fig. 4.24. Furthermore, the delay has been plotted versus the tapering factor for the generic case of \(p=1\) in the right-hand diagram. It is clear from this plot that the delay dependence on the tapering factor is rather weak. Not much speed is lost even if the tapering factor is doubled from, say, four to eight. The delay increase is only 20%.

**Fig. 4.24.** a) Optimum tapering factor vs. \(p\), and b) normalized buffer area and normalized delay vs. the tapering factor, \(f\).
The buffer area is much more critically dependent on the tapering factor chosen. In fact the area decreases to less than half if the tapering factor is doubled from four to eight.

**Example 4.1.** Assume that $x=600$ and $p=1$. What is the optimum tapering factor and how many inverters should there be in the buffer?

**Solution.** According to the previous discussion, the optimum tapering factor is $f=3.6$. The buffer contains $N-1=\ln 600/\ln 3.6-1=4$ inverters, a number that fortunately enough is an even integer. Hence, the delay is given by

$$d = 5(p + f) = 23 \text{ delay units.}$$

Concerning the buffer area $A$, let us assume for simplicity that the area of each inverter is proportional to its size as determined by the tapering factors, i.e.,

$$A \sim f + f^2 + f^3 + f^4 = 230 \text{ area units.}$$

**Example 4.2.** How much speed would we lose and how much area would we save if we instead choose $N=2$?

**Solution.** For a buffer with two inverters instead of four we obtain an optimum tapering factor $f=\sqrt[3]{600}=8.4$. For a two-stage buffer the delay is given by

$$d = 3(p + f) = 28 \text{ delay units.}$$

Actually, this delay is only 23% longer than the minimum delay in the previous example. Simultaneously, the area would be reduced from 230 to 80 area units, an area that is only 35% of the original.

**Example 4.3.** Now, assume that your boss has given you an area restriction. The area $A$ of your buffer must be less than a certain number, say $A_{\text{max}}=30$ area units and only two-stage buffers are allowed. How would you size this buffer, if again $x=600$?

**Solution.** For a two-stage buffer the delay is given by

$$d = p + f_1 + p + f_2 + p + \frac{x}{f_1 f_2}.$$
Concerning the buffer area \( A \), let us assume for simplicity that the inverter cell area is proportional to the cell driving capability, i.e.

\[
A \sim f_1 f_2 \leq A_{\text{max}}.
\]

Now, we have two equations and two unknowns. Solving this problem numerically, we obtain the optimum tapering factors \( f_1 \approx 4 \), and \( f_2 \approx 6.5 \). These tapering factors result in a relative delay of

\[
d = 3 p + 4 + 6.5 + \frac{600}{4 \cdot 6.5} = 36.6 \text{ delay units},
\]

at the maximum allowed area of 30 area units. This delay is about 30% longer than the minimum delay for a two-stage buffer in Example 4.2, but its area is less than 40% of the area of the optimized two-stage buffer.

In retrospect, the most convenient solution would have been to choose \( f_1 = f_2 = f \), resulting in \( f = 5 \) and a delay of 37 delay units. ■

### 4.4.5 Branching

Now, let us conclude this chapter by looking at the branching example shown in Fig. 4.25. Here we see an inverter \((C_1, R_1)\) driving \( b_1 \) identical inverters \((C_2, R_2)\), where each of these inverters in their turn drives \( b_2 \) identical capacitors \( C_L \). These load capacitors are \( x \) times larger in capacitance than the leftmost reference inverter. Let assume that the parasitic delay of these inverters is \( p = 1 \).

![Fig. 4.25. Introducing branching factors.](image_url)
If the stage fanouts are denoted $f_1$ and $f_2$, the relative propagation delay can be written

$$d = p + f_1 + p + f_2.$$  
(4.31)

Since $f_1 = b_1 C_2 / C_1$ and $f_2 = b_2 C_L / C_2$, the product of the two tapering factors is given by

$$f_1 f_2 = b_1 b_2 C_L / C_1 = b_1 b_2 x.$$  
(4.32)

Now, let us introduce the path branching factor, $B = b_1 b_2$, and the path electrical effort $H = x$. Using this nomenclature, the relationship between the two tapering factors can be written,

$$f_1 f_2 = F = B H,$$  
(4.33)

where $F = B H$ is the path fanout, or path effort. The delay can now be written on the simple form

$$d = p + f_1 + p + \frac{F}{f_1}.$$  
(4.34)

As before, minimum delay is obtained for $f_1 = f_2 = \sqrt{F} = \sqrt{B H}$.

**Example 4.4.** Calculate the optimum tapering factor and the buffer driving capability if $b_1 = b_2 = 4$ and $x = 64$!

**Solution.** The path branching factor is $B = b_1 b_2 = 16$, and consequently the path fanout is given by $F = B H = 1024$. The optimum tapering factor is given by $f_1 = f_2 = \sqrt{1024} = 32$.

Inverter sizes can now be calculated with capacitance transformation working backward along the path.

Since, $f_2 = b_2 C_L / C_2$, we obtain $C_2 = b_2 C_L / f_2 = C_L / 8$, and from $C_L = 64 C_1$ we obtain $C_2 = 8 C_1$. Similarly, $C_1 = b_1 C_2 / f_1 = C_2 / 8 = C_1$. This last calculation is done just to verify that we have not done any mistakes in our calculations.

The buffer driving capability now is eight times that of the reference inverter since the driving capability scales with the input capacitance, or maybe it is the input capacitance that scales with the driving capability. Anyway, the buffer source resistance is one eighth of that of the reference inverter, i.e. $R_2 / R_1 = 1/8$ and $C_2 / C_1 = 8$. Note, that the RC product remain constant. ■
4.5 Summary.

In this chapter the dynamic switching properties of the CMOS inverter were discussed. The propagation delay was defined and a simple RC delay model was derived. The RC model and its effective resistance were discussed and a simple RC two-port model of the inverter was derived. After some RC delay examples, the p-channel device was sized with respect to the n-channel device for minimum delay.

The linear delay model and the limitations to the linear delay model were discussed, and comparisons to a nonlinear delay model were made. These limitations were found to be due to the dependence of the propagation delay on the input edge rates. Finally, the usefulness of the relative delay model was demonstrated by a number of technology independent examples where the optimum tapering factor when driving large capacitive loads was found. Buffer insertion was discussed and trade-offs between speed and area were discussed.