Okay, now that we understand how to build combinational logic gates using CMOS, let's turn our attention to the timing specifications for the gates.

Here's a simple circuit consisting of two CMOS inverters connected in series, which we'll use to understand how to characterize the timing of the inverter on the left.

It will be helpful to build an electrical model of what happens when we change V_IN, the voltage on the input to the left inverter.

If V_IN makes a transition from a digital 0 to a digital 1, the PFET switch in the pullup turns off and the NFET switch in pulldown turns on, connecting the output node of the left inverter to GROUND.

The electrical model for this node includes the distributed resistance and capacitance of the physical wire connecting the output of the left inverter to the input of the right inverter.

And there is also capacitance associated with the gate terminals of the MOSFETs in the right inverter.

When the output node is connected to GROUND, the charge on this capacitance will flow towards the GROUND connection through the resistance of the wire and the resistance of the conducting channel of the NFET pulldown switch.

Eventually the voltage on the wire will reach the potential of the GROUND connection, 0V.

The process is much the same for falling transitions on V_IN, which cause the output node to charge up to V_DD.

Now let's look at the voltage waveforms as a function of time.

The top plot shows both a rising and, later, a falling transition for V_IN.

We see that the output waveform has the characteristic exponential shape for the voltage of a capacitor being discharged or charged though a resistor.

The exponential is characterized by its associated R-C time constant, where, in this case, the R is the net resistance of the wire and MOSFET channel, and the C is the net capacitance of the wire and MOSFET gate terminals.

Since neither the input nor output transition is instantaneous, we have some choices to make about how to measure the inverter's propagation delay.

Happily, we have just the guidance we need from our signaling thresholds!

The propagation delay of a combinational logic gate is defined to be an upper bound on the delay from valid inputs to valid outputs.

Valid input voltages are defined by the V_IL and V_IH signaling thresholds, and valid output voltages are defined by the V_OL and V_OH signaling thresholds.

We've shown these thresholds on the waveform plots.

To measure the delay associated with the rising transition on V_IN, first identify the time when the input becomes a valid digital 1, i.e., the time at which V_IN crosses the V_IH threshold.

Next identify the time when the output becomes a valid digital 0, i.e., the time at which V_OUT crosses the V_OL threshold.

The interval between these two time points is the delay for this particular set of input and output transitions.

We can go through the same process to measure the delay associated with a falling input transition.

First, identify the time at which V_IN cross the V_IL threshold.

Then find the time at which V_OUT crosses the V_OH threshold.

The resulting interval is the delay we wanted to measure.

Since the propagation delay, t_PD, is an upper bound on the delay associated with *any* input transition, we'll choose a value for t_PD that's greater than or equal to the measurements we just made.

When a manufacturer selects the t_PD specification for a gate, it must take into account manufacturing variations, the effects of different environmental conditions such as temperature and power-supply voltage, and so on.

It should choose a t_PD that will be an upper bound on any delay measurements their customers might make on actual devices.

From the designer's point of view, we can rely on this upper bound for each component of a larger digital system and use it to calculate the system's t_PD without having to repeat all the manufacturer's measurements.

If our goal is to minimize the propagation delay of our system, then we'll want to keep the capacitances and resistances as small as possible.

There's an interesting tension here: to make the effective resistance of a MOSFET switch smaller, we would increase its width.

But that would add additional capacitance to the switch's gate terminal, slowing down transitions on the input node that connects to the gate!

It's a fun optimization problem to figure out transistor sizing that minimizes the overall propagation delay.

Although not strictly required by the static discipline, it will be useful to define another timing specification, called the "contamination delay".

It measures how long a gate's previous output value remains valid after the gate's inputs start to change and become invalid.

Technically, the contamination delay will be a lower bound on the delay from an invalid input to an invalid output.

We'll make the delay measurements much as we did for the propagation delay.

On a rising input transition, the delay starts when the input is no longer a valid digital 0, i.e., when V_IN crosses the V_IL threshold.

And the delay ends when the output becomes invalid, i.e., when V_OUT crosses the V_OH threshold.

We can make a similar delay measurement for the falling input transition.

Since the contamination delay, t_CD, is a lower bound on the delay associated with *any* input transition, we'll choose a value for t_CD that's less than or equal to the measurements we just made.

Do we really need the contamination delay specification?

Usually not.

And if not's specified, designers should assume that the t_CD for a combinational device is 0.

In other words a conservative assumption is that the outputs go invalid as soon as the inputs go invalid.

By the way, manufacturers often use the term "minimum propagation delay" to refer to a device's contamination delay.

That terminology is a bit confusing, but now you know what it is they're trying to tell you.

So here's a quick summary of the timing specifications for combinational logic.

These specifications tell us how the timing of changes in the output waveform (labeled B in this example) are related to the timing of changes in the input waveform (labeled A).

A combinational device may retain its previous output value for some interval of time after an input transition.

The contamination delay of the device is a guarantee on the minimum size of that interval, i.e., t_CD is a lower bound on how long the old output value stays valid.

As stated in Note 2, a conservative assumption is that the contamination delay of a device is 0, meaning the device's output may change immediately after an input transition.

So t_CD gives us information on when B will start to change.

Similarly, it would be good to know when B is guaranteed to be done changing after an input transition.

In other words, how long do we have to wait for a change in the inputs to reflected in an updated value on the outputs?

This is what t_PD tells us since it is a upper bound on the time it takes for B to become valid and stable after an input transition.

As Note 1 points out, in general there are no guarantees on the behavior of the output in the interval after t_CD and before t_PD, as measured from the input transition.

It would legal for the B output to change several times in that interval, or even have a non-digital voltage for any part of the interval.

As we'll see in the last video of this chapter, we'll be able to offer more insights into B's behavior in this interval for a subclass of combinational devices.

But in general, a designer should make no assumptions about B's value in the interval between t_CD and t_PD.

How do we calculate the propagation and contamination delays of a larger combinational circuit from the timing specifications of its components?

Our example is a circuit of four NAND gates where each NAND has a t_PD of 4 ns and t_CD of 1 ns.

To find the propagation delay for the larger circuit, we need to find the maximum delay from an input transition on nodes A, B, or C to a valid and stable value on the output Y.

To do this, consider each possible path from one of the inputs to Y and compute the path delay by summing the t_PDs of the components along the path.

Choose the largest such path delay as the t_PD of the overall circuit.

In our example, the largest delay is a path that includes three NAND gates, with a cumulative propagation delay of 12 ns.

In other words, the output Y is guaranteed be stable and valid within 12 ns of a transition on A, B, or C. To find the contamination delay for the larger circuit, we again investigate all paths from inputs to outputs, but this time we're looking for the shortest path from an invalid input to an invalid output.

So we sum the t_CDs of the components along each path and choose the smallest such path delay as the t_CD of the overall circuit.

In our example, the smallest delay is a path that includes two NAND gates with a cumulative contamination delay of 2 ns.

In other words, the output Y will retain its previous value for at least 2 ns after one of the inputs goes invalid.