So, how hard can it be to build a communication channel?

Aren't they just logic circuits with a long wire that runs from one component to another?

A circuit theorist would tell you that wires in a schematic diagram are intended to represent the equipotential nodes of the circuit, which are used to connect component terminals.

In this simple model, a wire has the same voltage at all points and any changes in the voltage or current at one component terminal is instantly propagated to the other component terminals connected to the same wire.

The notion of distance is abstracted out of our circuit models: terminals are either connected by a wire, or they're not.

If there are resistances, capacitances, and inductances that need to be accounted for, the necessary components can be added to the circuit model.

Wires are timeless.

They are used to show how components connect, but they aren't themselves components.

In fact, thinking of wires as equipotential nodes is a very workable model when the rate of change of the voltage on the wire is slow compared to the time it takes for an electromagnetic wave to propagate down the wire.

Only as circuit speeds have increased with advances in integrated circuit technologies did this rule-of-thumb start to be violated in logic circuits where the components were separated by at most 10's of inches.

In fact, it has been known since the late 1800s that changes in voltage levels take finite time to propagate down a wire.

Oliver Heaviside was a self-taught English electrical engineer who, in the 1880's, developed a set of "telegrapher's equations" that described how signals propagate down wires.

Using these, he was able to show how to improve the rate of transmission on then new transatlantic telegraph cable by a factor of 10.

We now know that for high-speed signaling we have to treat wires as transmission lines, which we'll say more about in the next few slides.

In this domain, the distance between components and hence the lengths of wires is of critical concern if we want to correctly predict the performance of our circuits.

Distance and signal propagation matter - real-world wires are, in fact, fairly complex components!

Here's an electrical model for an infinitesimally small segment of a real-world wire.

An actual wire is correctly modeled by imagining a many copies of the model shown here connected end-to-end.

The signal, i.e., the voltage on the wire, is measured with respect to the reference node which is also shown in the model.

There are 4 parameters that characterize the behavior of the wire.

R tells us the resistance of the conductor.

It's usually negligible for the wiring on printed circuit boards, but it can be significant for long wires in integrated circuits.

L represents the self-inductance of the conductor, which characterizes how much energy will be absorbed by the wire's magnetic fields when the current flowing through the wire changes.

The conductor and reference node are separated by some sort insulator (which might be just air!) and hence form a capacitor with capacitance C.

Finally, the conductance G represents the current that leaks through the insulator.

Usually this is quite small.

The table shows the parameter values we might measure for wires inside an integrated circuit and on printed circuit boards.

If we analyze what happens when sending signals down the wire, we can describe the behavior of the wires using a single component called a transmission line, which has a characteristic complex-valued impedance Z_0.

At high signaling frequencies and over the distances found on-chip or on circuit boards, such as one might find in a modern digital system, the transmission line is lossless, and voltage changes ("steps") propagate down the wire at the rate of 1/sqrt(LC) meters per second.

Using the values given here for a printed circuit board, the characteristic impedance is approximately 50 ohms and the speed of propagation is about 18 cm (7") per ns.

To send digital information from one component to another, we change the voltage on the connecting wire, and

that voltage step propagates from the sender to the receiver.

We have to pay some attention to what happens to that energy front when it gets to the end of the wire!

If we do nothing to absorb that energy, conservation laws tell us that it reflects off the end of the wire as an "echo" and soon our wire will be full of echoes from previous voltage steps!

To prevent these echoes we have to terminate the wire with a resistance to ground that matches the characteristic impedance of the transmission line.

If the signal can propagate in both directions, we'll need to terminate at both ends.

What this model is telling is the time it takes to transmit information from one component to another and that we have to be careful to absorb the energy associated with the transmission when the information has reached its destination.

With that little bit of theory as background, we're in a position to describe the real-world consequences of poor engineering of our signal wires.

The key observation is that unless we're careful there can still be energy left over from previous transmissions that might corrupt the current transmission.

The general fix to this problem is time, i.e., giving the transmitted value a longer time to settle to an interference-free value.

But slowing down isn't usually acceptable in high-performance systems, so we have to do our best to minimize these energy-storage effects.

If the termination isn't exactly right, we'll get some reflections from any voltage step reaching the end of the wire, and it will take a while for these echoes to die out.

In fact, as we'll see, energy will reflect off of any impedance discontinuity, which means we'll want to minimize the number of such discontinuities.

We need to be careful to allow sufficient time for signals to reach valid logic levels.

The shaded region shows a transition of the wire A from 1 to 0 to 1.

The first inverter is trying to produce a 1-output from the initial input transition to 0, but doesn't have sufficient time to complete the transition on wire B before the input changes again.

This leads to a runt pulse on wire C, the output of the second inverter, and we see that the sequence of bits on A has been corrupted by the time the signal reaches C.

This problem was caused by the energy storage in the capacitance of the wire between the inverters, which will limit the speed at which we can run the logic.

And here see we what happens when a large voltage step triggers oscillations in a wire, called ringing, due to the wire's inductance and capacitance.

The graph shows it takes some time before the ringing dampens to the point that we have a reliable digital signal.

The ringing can be diminished by spreading the voltage step over a longer time.

The key idea here is that by paying close attention to the design of our wiring and the drivers that put information onto the wire, we can minimize the performance implications of these energy-storage effects.

Okay, enough electrical engineering!

Suppose we have some information in our system.

If we preserve that information over time, we call that storage.

If we send that information to another component, we call that communication.

In the real world, we've seen that communication takes time and we have to budget for that time in our system timing.

Our engineering has to accommodate the fundamental bounds on propagating speeds, distances between components, and how fast we can change wire voltages without triggering the effects we saw on the previous slide.

The upshot: our timing models will have to account for wire delays.

In Part 1 of this course, we had a simple timing model that assigned a fixed propagation delay, t_PD, to the time it took for the output of a logic gate to reflect a change to the gate's input.

We'll need to change our timing model to account for delay of transmitting the output of a logic gate to the next components.

The timing will be load dependent, so signals that connect to the inputs of many other gates will be slower than

signals that connect to only one other gate.

The Jade simulator takes the loading of a gate's output signal into account when calculating the effective propagation delay of the gate.

We can improve propagation delays by reducing the number of loads on output signals or by using specially-design gates called buffers (the component shown in red) to drive signals that have very large loads.

A common task when optimizing the performance of a circuit is to track down heavily-loaded and hence slow wires and re-engineering the circuit to make them faster.

Today our concern is wires used to connect components at the system level.

So next we'll turn our attention to possible designs for system-level interconnect and the issues that might arise.