Clock Characteristics

Clock signals must toggle twice (full cycle) for each single transition for data.

Clock wires also tend to be very heavily loaded nets because they typically fan-out to every FF in the design.

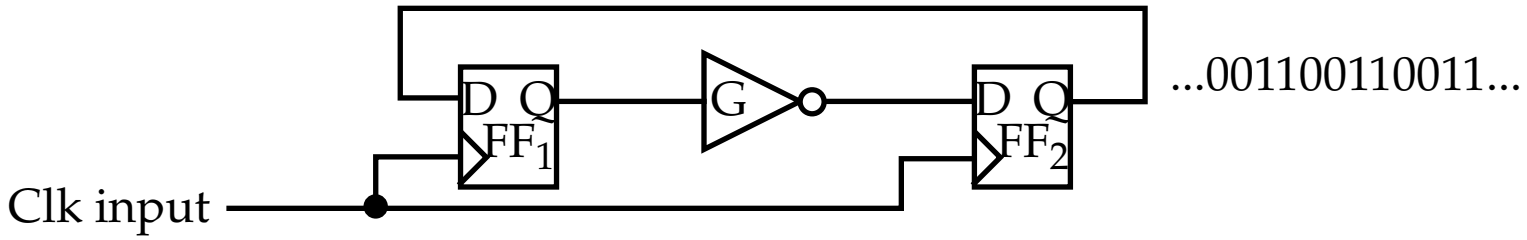
Data wires typically fan-out to only a couple of devices.

Here, we look at clock drivers, special clock routing rules and other circuits for distribution of clock signals.

Timing Margin

We've discussed this concept in VLSI.

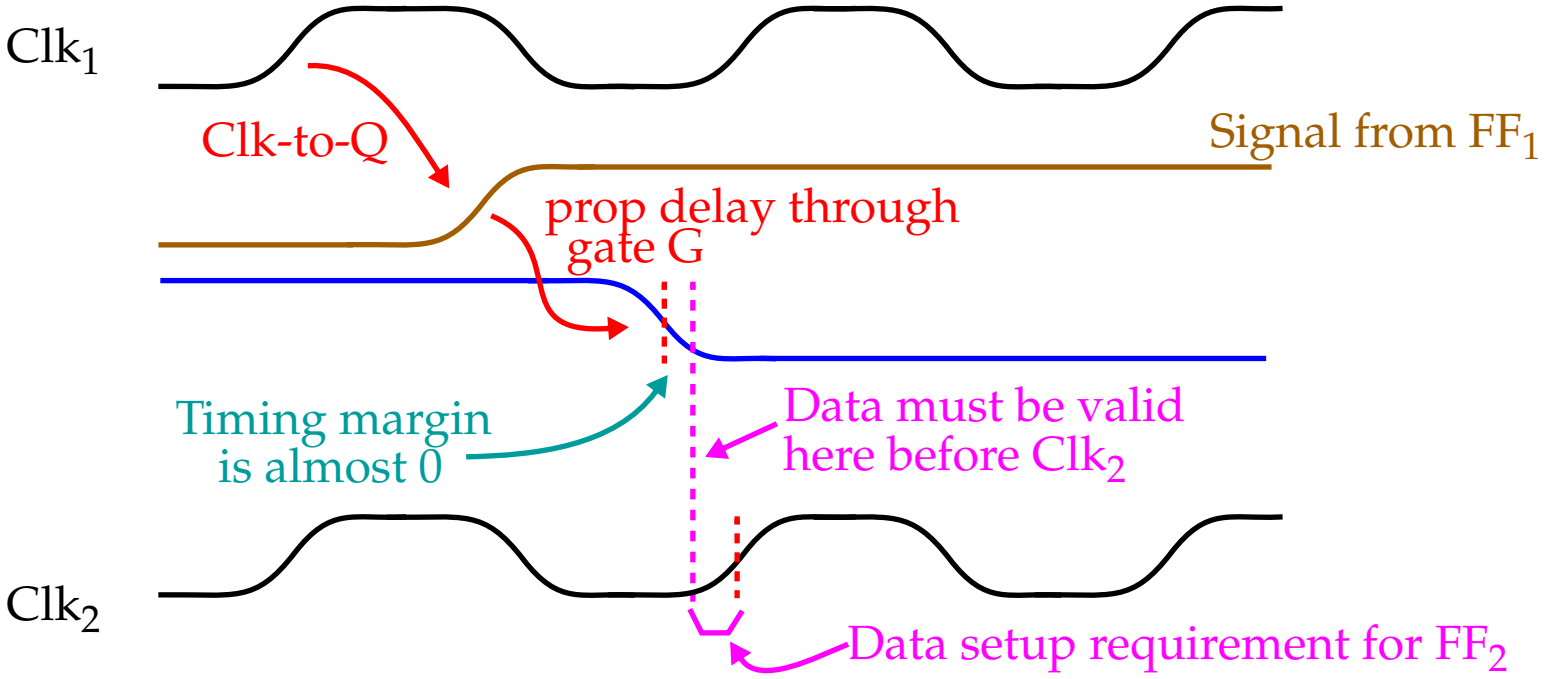
Consider the 2-bit *ring counter*.



Timing Margin

As the frequency is raised, the circuit continues to output the same pattern and then fails.

It fails at the input of FF_2 because of a setup time violation.



The *timing margin* is defined, in this example, as the amount of time between the emergence of signal from G and time when data for FF_2 must be valid.

It is the slack remaining in each clk cycle.

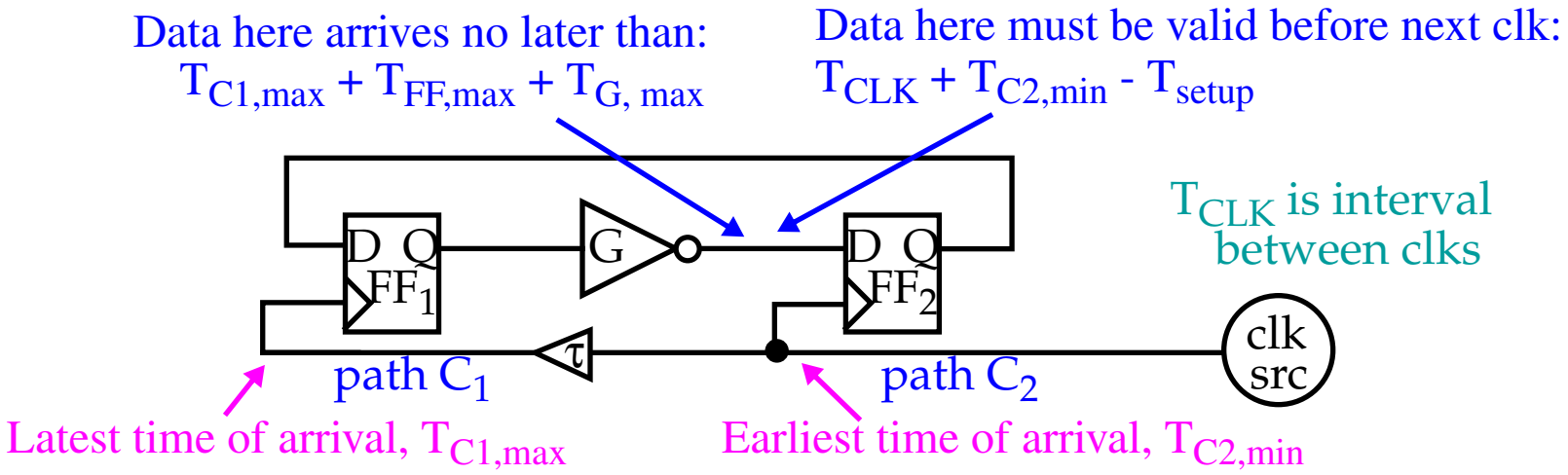
Timing Margin

Timing margins should be positive under all operating conditions, temp, etc.

A good target is to allow one gate delay.

Clock Skew

Again, completely analogous to clock skew in VLSI.



Here, we've analyzed the worst case timing margin.

$$T_{CLK} > T_{FF,max} + T_{G,max} + T_{setup} + (T_{C1,max} - T_{C2,min})$$

Clock Skew

The Clock interval, T_{CLK}, must be greater than the sum of the intervals along the path from FF_1 through gate G + the *clock skew* (last subterm on right).

Late arrival of clk to FF_1, T_{C1}, or early arrival at FF_2, T_{C2}, both deteriorate the timing margin and requires the clk interval to be increased.

Also note that the uncertainty in T_{CLK} (which needs to be considered) is usually minimal because T_{CLK} is usually crystal-controlled.

Lastly, it is apparent from this equation that *clock skew* has as much impact on system performance as any other delay.

Since there is typically only a small number of clk nets on the board, focusing here is an easy way to increase timing margin.

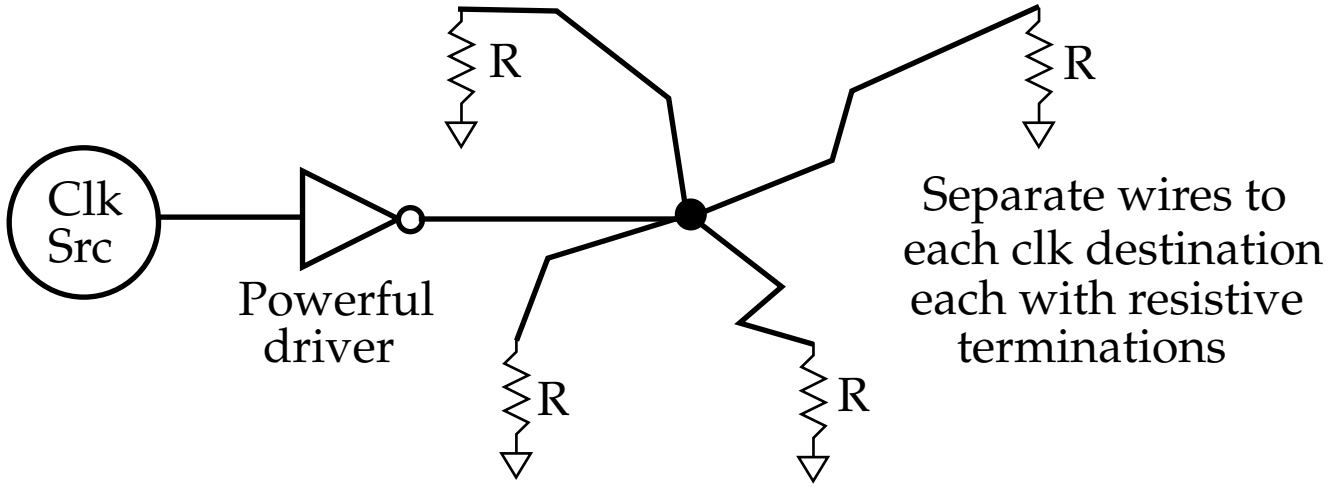
Straightforward way of minimizing clock skew:

- Position all clk inputs close together on the board
- Drive them all from the same source.

Clock Skew

This fails, of course, if the system has many chips requiring clk signals.

In this case, try a spider distribution network:



Total resistive load is R/N, where N is number of branches.

For a 75 Ω transmission line impedance, 3 legs present a 25 Ω load to driver.

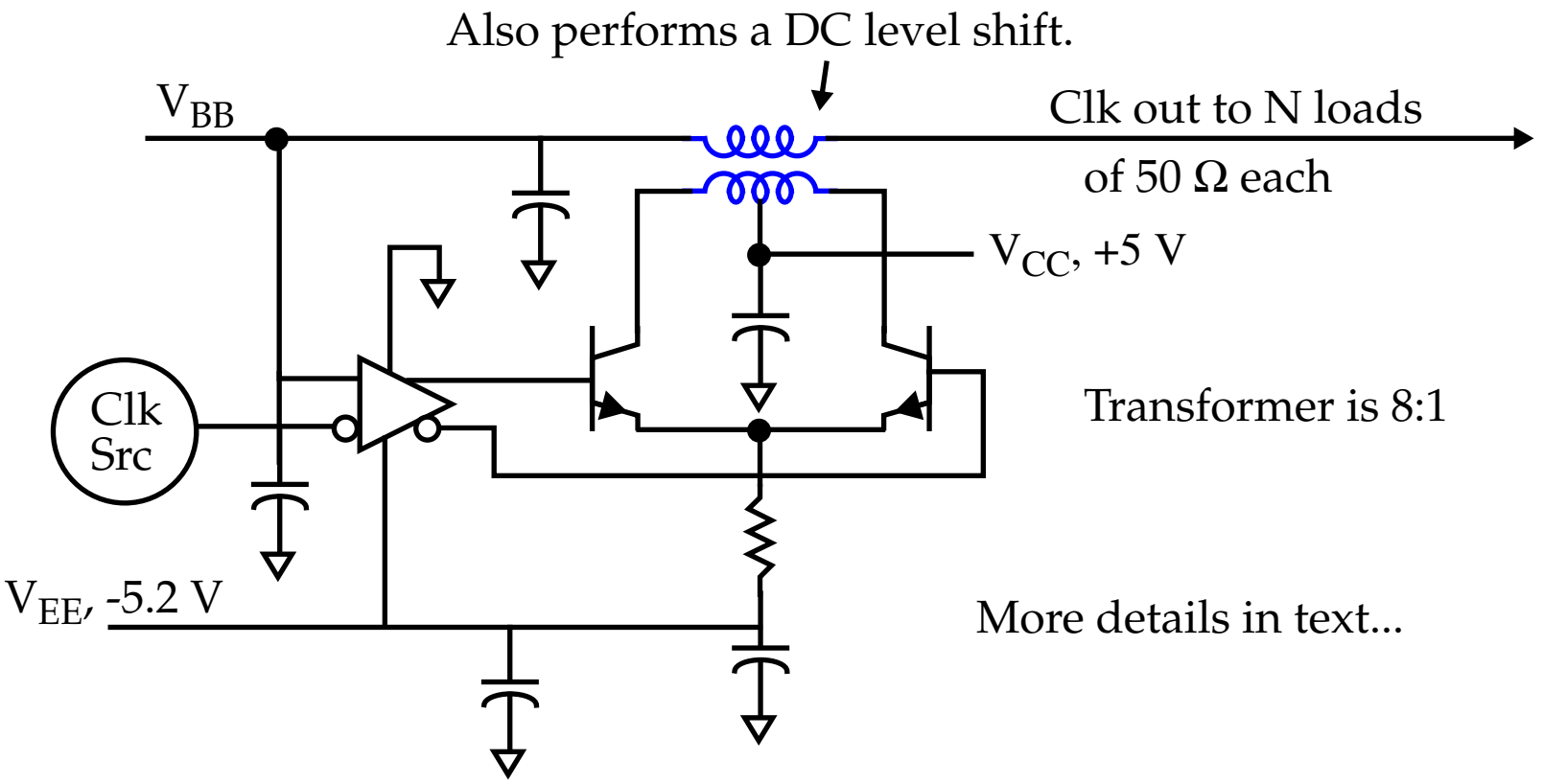
This is a difficult load to drive for any driver.

You can connect several outputs together (parallel arrangement) from a single IC, multiple output driver chip.

Clock Skew

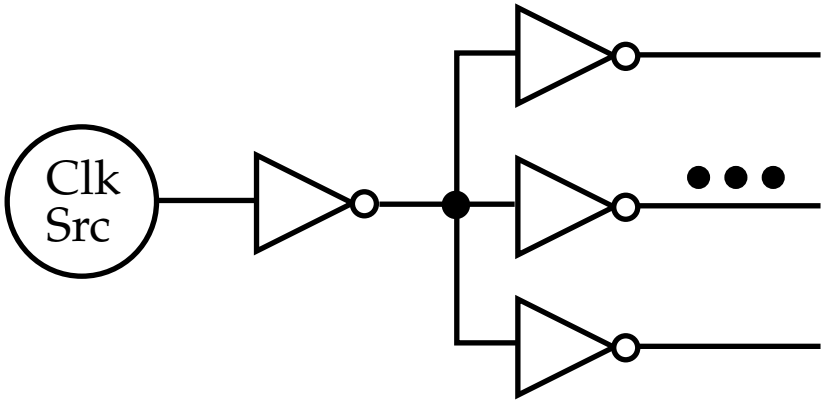
A discrete, low-impedance amp can drive many spider legs

The **ECL power driver** uses a transformer to convert from a high-impedance, high-voltage output to a low-impedance, high-current output.



Clock Skew

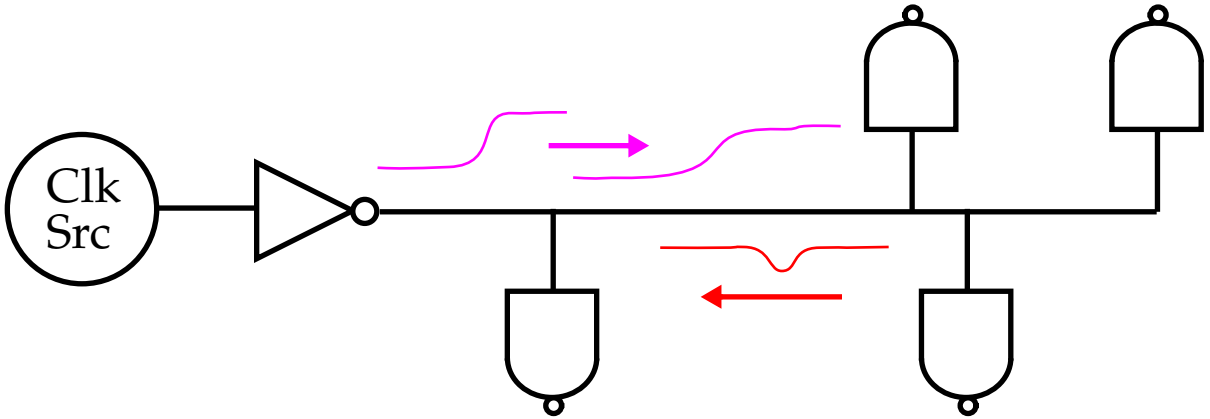
Another alternative is to use a clk distribution tree.



Every path traverses the same number of gates.

Low impedance clock distribution lines

As previously discussed, capacitance causes rise time to degrade and introduces reflections.



Low Impedance Clock Distribution Lines

The reflected pulses follow the derivative of the input signal and are proportional to -jωC(Z_0/2).

From this expression, reducing them requires:

- Slowing down the rise time of the driver.
- Lower the capacitance at each tap.
- Lower the characteristic impedance of the clk distribution line.

For item three, a **doubled** clock driver chip feeding a 20 Ω clock line is 2.5 times less sensitive to the capacitance of the clock taps than a 50 Ω line.

This also helps in situations where the load capacitance is dynamic. Lower impedance makes the line less sensitive to changes in load.

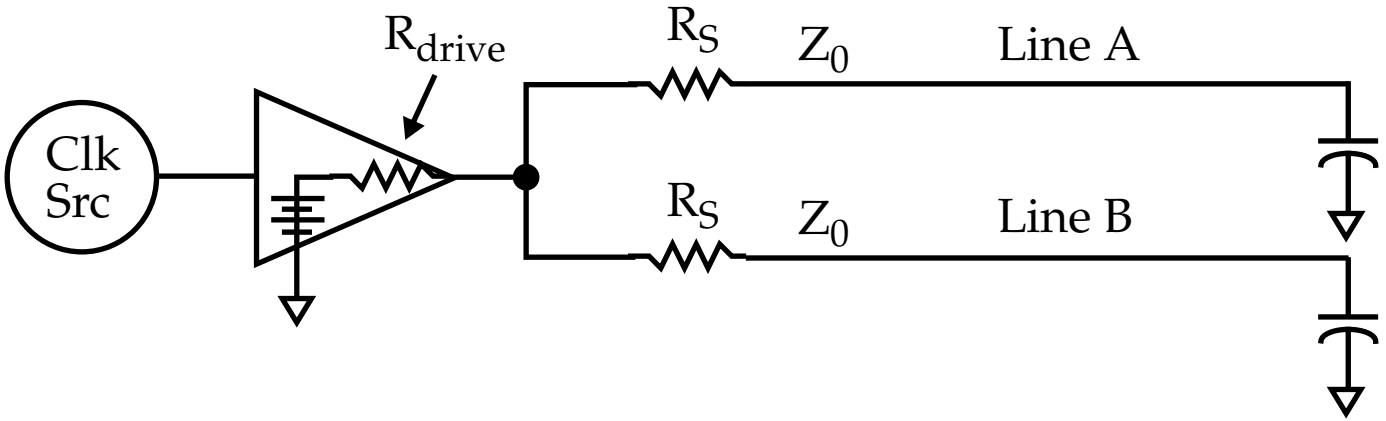
Source Termination of Multiple Clock Lines

We discussed the advantages of a source terminated line:

- Impedance is twice that of an end terminated line
- Current goes to zero after 2T seconds (round trip delay).

Source Termination of Multiple Clock Lines

It is possible to drive multiple src-terminated lines from a single driver, but only under a limited configuration.



Here, the lengths of the lines *must be the same* and the *loads must be balanced*.

These lines are actually coupled together in a jointly resonant structure, because of the **finite** output impedance of the driver.

Therefore, it's not possible to treat them independently.

In order for this to work, the src-termination resistors must equal:

$$R_S = Z_0 - R_{drive}N$$

Source Termination of Multiple Clock Lines

When driving one line, then this reduces to matching $Z_0 = R_s + R_{drive}$, as we discussed previously.

Under multiple lines, this requires smaller src-terminating resistors.

At some value of N, this becomes negative, indicating no solution exists.

Controlling Crosstalk on Clock Lines

Leave extra gaps around clock traces or put them on a separate plane.

The logistics can be tricky. One approach is to make the traces wider than needed, and then, after routing, reduce their width.

Delay Adjustments

Sometimes it is desirable to retard (or advance) the clock along one path over another path.

This usually improves timing margins in one part of the circuit and worsens them in another part.

Clock Delay (or Clock Phase) Adjustments

Or you may desire in other situations to make clock adjustments as a means of lowering the clock skew.

The simplest form is a circuit that adds a **fixed delay**.

Used to compensate for nominal delays in other parts of the circuit.

It is immutable once the board is designed and therefore, cannot be used to cancel variations introduced by board fabrication, chip delay, etc.

Fixed delays are built from 3 basic building blocks:

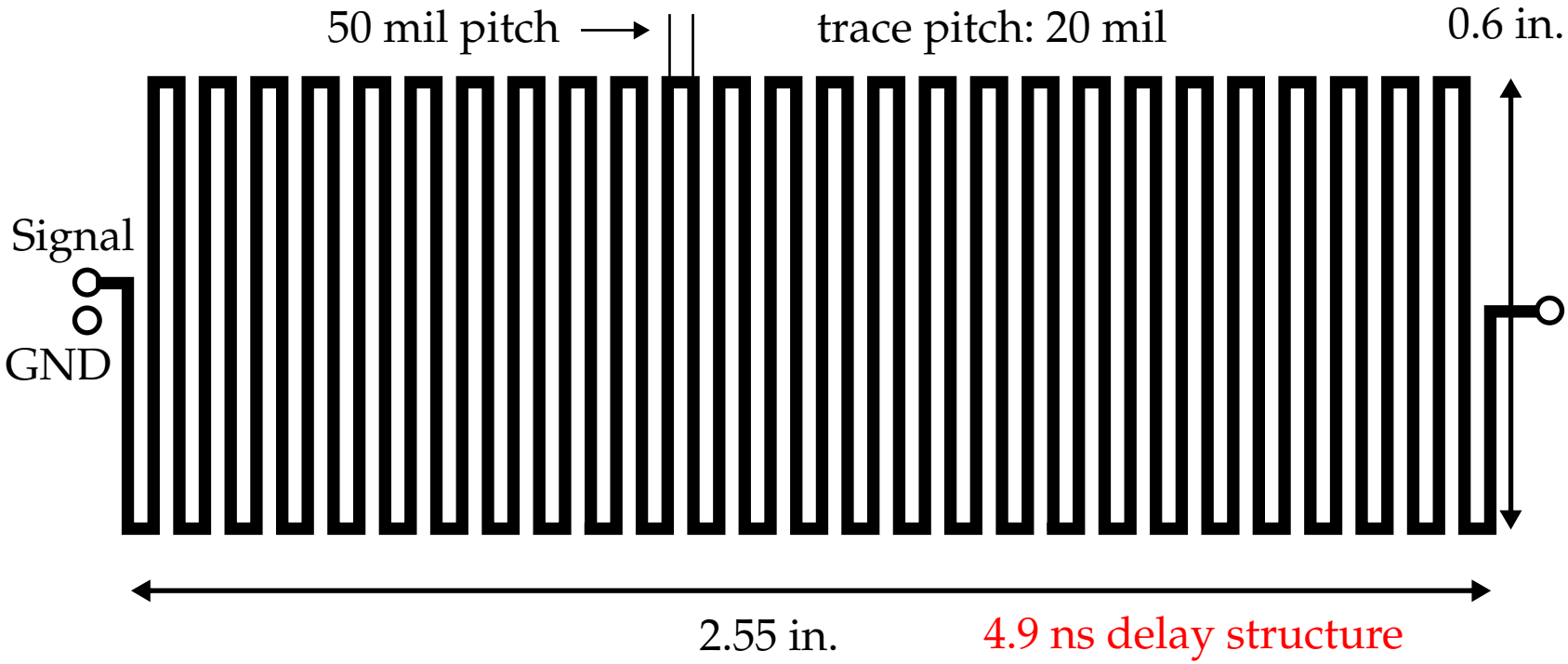
Delay type	**Amount of delay (ns)**	**Variation in delay (%)**
Delay line (trans. line)	0.1-5	10
Gate delay	0.1-20	300
Lumped-circuit delay	0.1-1000	5-20

Here, delay lines are good for short delays and are very accurate, gate delays are less board hungry but much less accurate.

Lumped-circuit delay elements cover the widest range.

Clock Delay (or Clock Phase) Adjustments

Delay lines printed directly on the board are expensive in area:



Each ns of delay consumes about 0.135 in.^{2} of board area, with a 7 ns delay occupying 1 in.^{2}!

Also, with FR-4, expect about a 10% change in prop. delay over the temperature range of 0-70 degrees.

Clock Delay (or Clock Phase) Adjustments

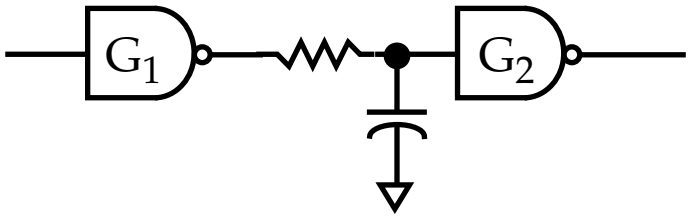
Some commercial lines surround the transmission line with magnetically permeable material that radically increases the delay/inch.

Spare gates can also be used as a delay element.

Problem: manufacturers specify max delay, but rarely disclose min. delay.

The resulting variation can be very large and can actually hinder efforts to control clock skew.

Lumped circuit elements produce clean, repeatable delays.



Delay = 0.69 RC

Accuracy and stability depend on the accuracy of the R and C components and the accuracy of the switching threshold of G_2.

Asymmetric threshold changes the delay for rising and falling edges.

Clock Delay (or Clock Phase) Adjustments

To achieve better symmetry, use a differential receiver with its negative input terminal connected to $V_{DD}/2$.

You should not try to shift the clock by more than 12% of the clock cycle in a single stage, but rather cascade several stages to get more.

When an RC circuit delays a square wave by more than 12%, the RC response does not have time to decay fully between pulse edges.

The receiver sees a slurred wfm between 10 and 90% instead of rail to rail.

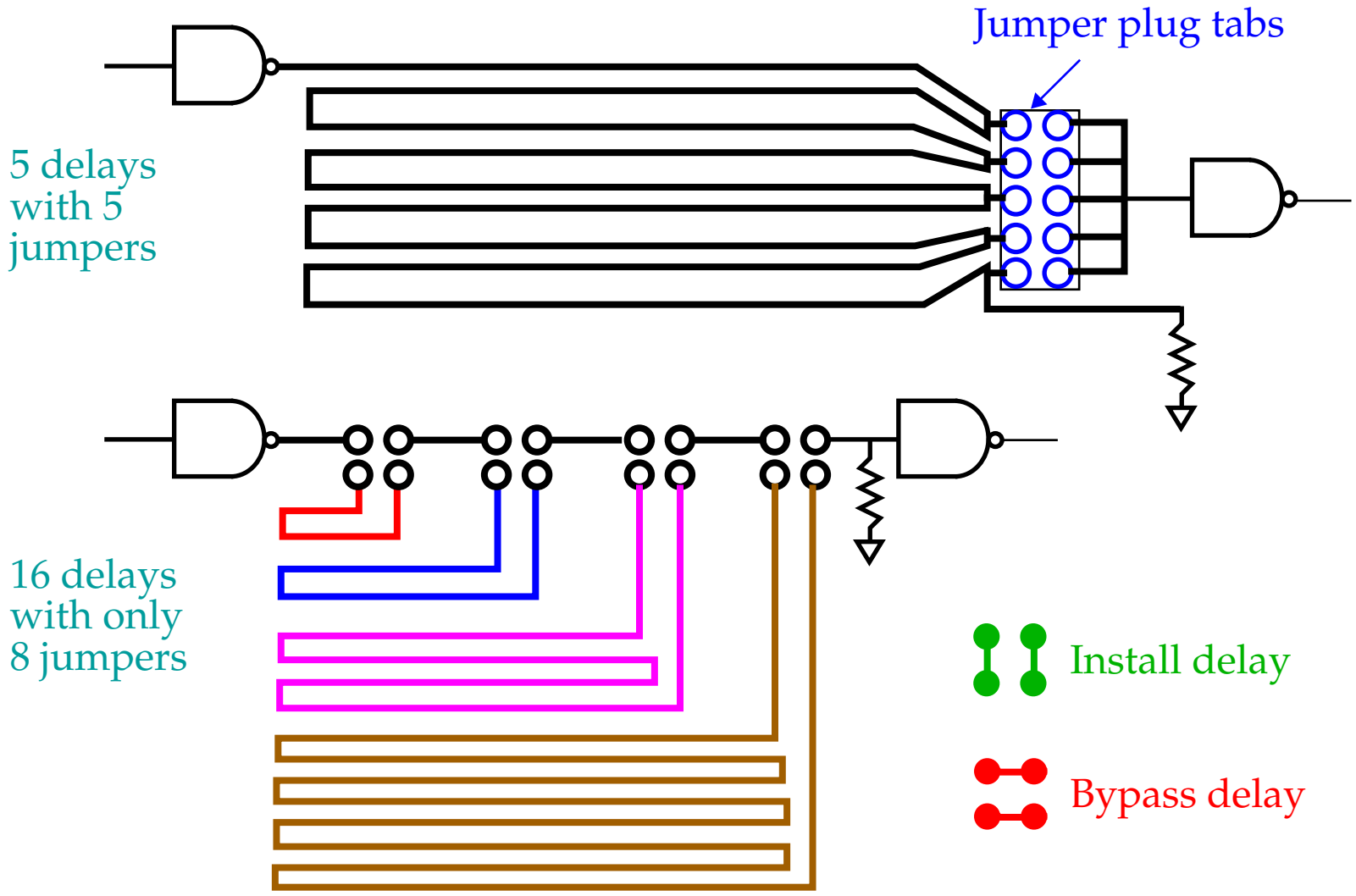
Adjustable Delays

Allows you to adjust for actual delays, which are performed after assembly.

The same three building blocks, transmission lines, logic gates and passive lumped components, are available here as well.

Adjustable Delays

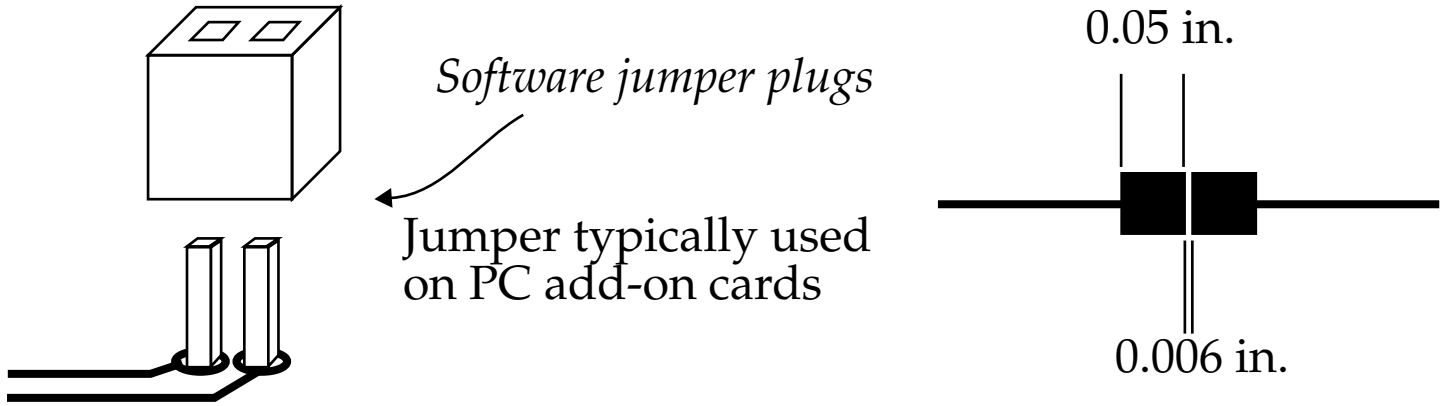
The delay line adjusts in quantized steps.



Adjustable Delays

Note the lengths in the second scheme are *tuned* to provide 1, 2, 4 and 8 times a basic delay *T*.

Tap connections can be implemented with a **shorting jumper bar**.



Shorting block inductance will be a problem above 100 MHz.

Alternative is to use the **solder blob jumper**.

Two 50 mil square pads separated by 6 mils, solder blob is easily cleared using solder wick.

Much better with regard to inductance and are more area efficient.

Adjustable Delays

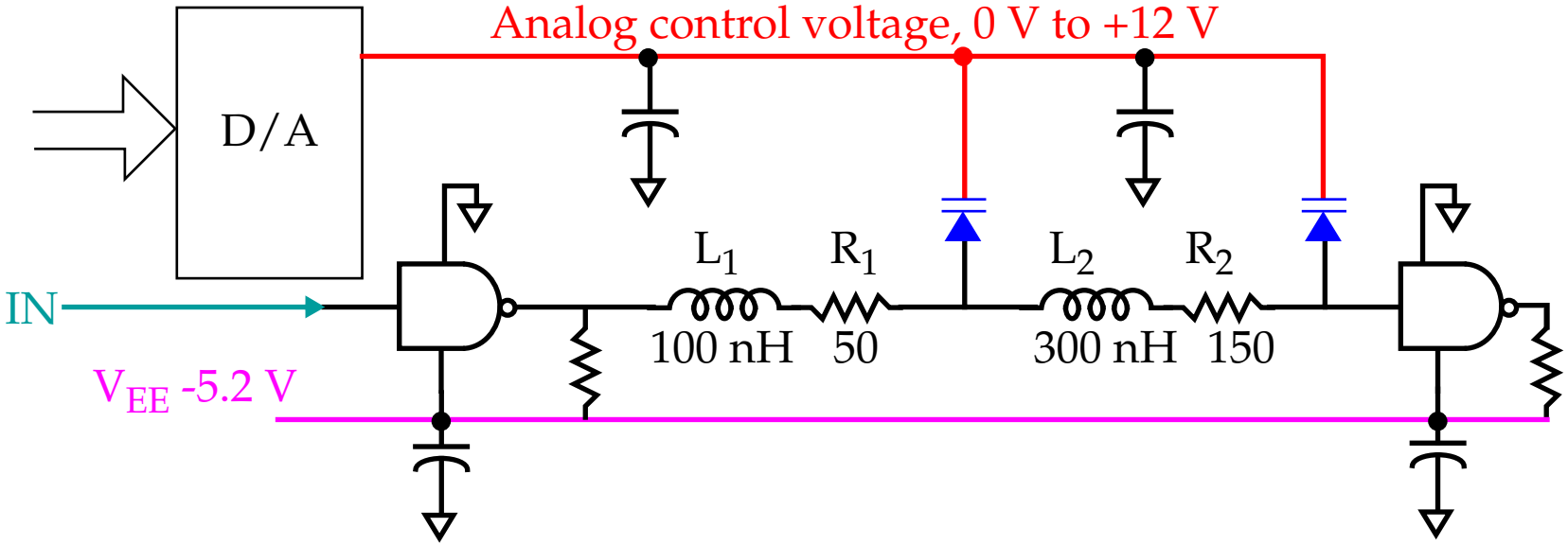
A chain of gates, tapped at discrete points, also allows delay adjustments in quantized steps.

As indicated, basic problem is accuracy.

The lumped circuit delay adjusts by varying R and C.

It's cheaper and easier to get adjustable resistors than capacitors.

An ideal delay circuit is continuously variable, stable over a wide range of temperatures and would adjust *itself* in production.



Adjustable Delays

There are two approaches, first uses **varactor diodes** (previous slide).

The parasitic capacitance associated with a varactor diode changes as a function of the applied voltage.

The preceding circuit uses *LC delay elements*, allowing a wider range of delay adjustment without attenuation.

Also, this circuit cascades two passive delay sections *without* buffering.

The impedance of the 2nd stage is 3 times that of the 1st, which reduces loading on the first and distortion.

There are only two buffers in this design.

A small number is good because their delay is temperature and supply voltage dependent.

The second approach to programmable delay uses a chain of gates within a single IC.

Adjustable Delays

The chain can be very long within the chip.

Programming is accomplished via control of a giant multiplexer that selects the taps to connect the output pins.

One way to automate the adjustment of delay, using either scheme, is to sense the switching times of *data* signals on the bus.

The clock is then adjusted to match specific transition times in the data waveform.

This is similar to *clock recovery* methods in serial data transmission.