Slide Set 12:
More about Timing

Steve Wilton
Dept. of ECE
University of British Columbia
stevew@ece.ubc.ca

Based on Slides by Res Saleh, 2000
which were based on Slides by Mark Horowitz, Sanford U.

Overview

In this lecture we will look at delay estimation in more detail, as well as explain how to find the delay in more complex situations (like transmission gates). The goal of this lecture is to give you the tools for determining how to size transistors in the cells that you design using a number of different models.

We will still use a simple RC model, but now we will look at more complex RC networks. We will also look at how to use this simple model to size transistors in a circuit.

Board Notes: RC Revisited

Summary of RC-Revisited Board Notes

Delay of a gate depends on:

- Capacitance being charged/discharged
- Full-on resistance of driver
- Speed that driver turns on or off (slope of the input signal)

If the input signal rises in 0 time (step signal), then the delay of a gate can be written as 0.693 R*C.

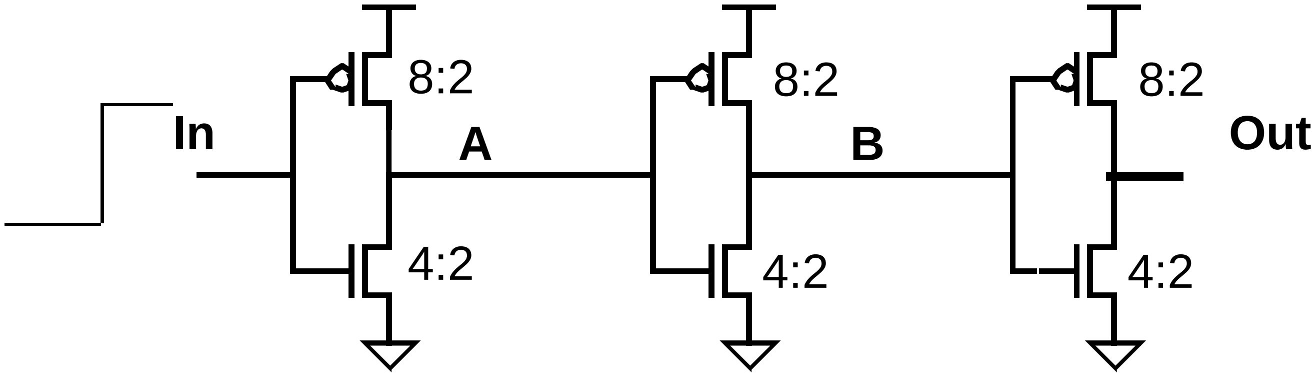
But, real input signals have a non-zero rise time. In that case, it can be shown that this increases the delay of a gate by about 40% (=0.3*R*C).

Thus, if the input signal has a non-zero rise time, we can estimate the gate delay as R*C (which is what we have been doing up to now)

More detailed approximations are sometimes used.

Review: Delay through a Series of Gates

So we can use this to find the delay of a series of gates:



In rises, to **A** falling = 0.693 x 6.5K x 24fF = 0.12ns (step input, single fanout)
A falling to **B** rising = 6.5K x 24fF = .156ns (ramp input, single fanout)
B rising to Out falling = 6.5K x 12fF = .075ns (ramp input, no fanout)

Total Delay = 0.12ns + 0.156ns + 0.07ns = 0.346ns

The only thing new here (compared to what you learned before the midterm) is that we assumed a step input of the input IN.

Transistor Sizing

Before the midterm, we learned how to size NMOS and PMOS transistors so as to make the pull-up and pull-down time equal.

But, this gave relative sizes (eg. Wp = 2*Wn).

The absolute values matter:

- For speed
- For power (we haven't talked about this yet)

Transistor Sizing

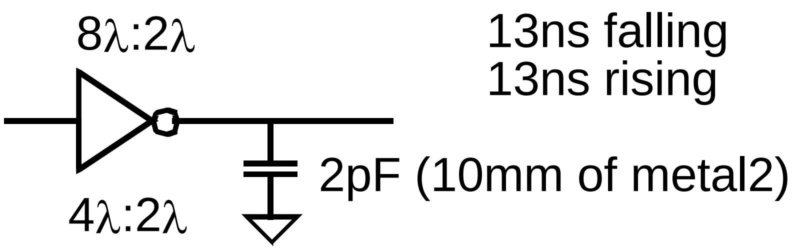
How big do you need to make a device? Depends on the desired timing:

- Need to think about the load you are driving
- Need to think about the load you present to your predecessor

Transistor sizes matter when you are driving a large capacitance

Large capacitance can be from:

Long wire on output, or large fanout (many gates being driven)



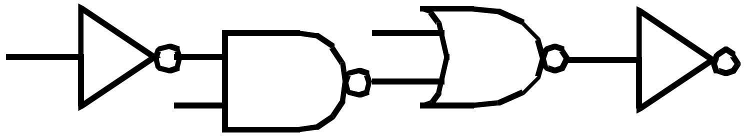
But this increases load to previous gate:

there is an optimum transistor sizing when we want to drive a large cap.

Board Notes: Driver Sizing

Gate Sizing

- What about things other than inverters? Delay of each stage should be equal

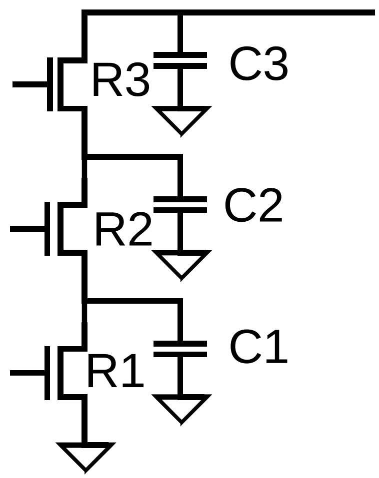


- If any of the delays are not equal,
  - Make the gate with the largest delay larger.
  - Decreases its delay, and increases its predecessor's delay.
  - But since its delay started larger, there will be a net win.
  - Optimal is roughly when the delays are equal
- For design, you don't want tons of SPICE or irsim simulations.
- Don't even really want to write RC equations.
  - Need a simpler way to get delays

Board Notes: Logical Effort

Series Stacks

What if we have transistors in series. How do we calculate more accurate delays? We know that we can add resistances, but what about the capacitance between the transistors?



If C3 is much larger than (C1 + C2) can ignore these smaller capacitors

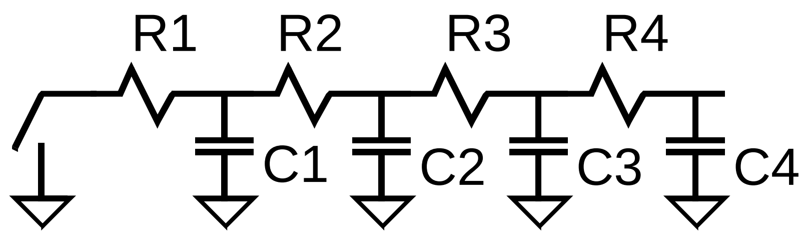
So the delay is approximately (R1 + R2 + R3) C3

But what if all are about the same size?

Have a distributed RC problem: need a different approach

Distributed RC using "Elmore" Delay

This is an old problem, which has a nice solution for RC ladder networks:



Delay = Sum (Cap_i * Resistance from Cap_i to source)
= (R1) C1 + (R1 + R2) C2 + (R1 + R2 + R3) C3 +
(R1 + R2 + R3 + R4) C4.

For RC trees, equation is

$$t_i = \sum_{k=1}^{n} R_{ik} C_k$$

where: t_i is the delay to node i

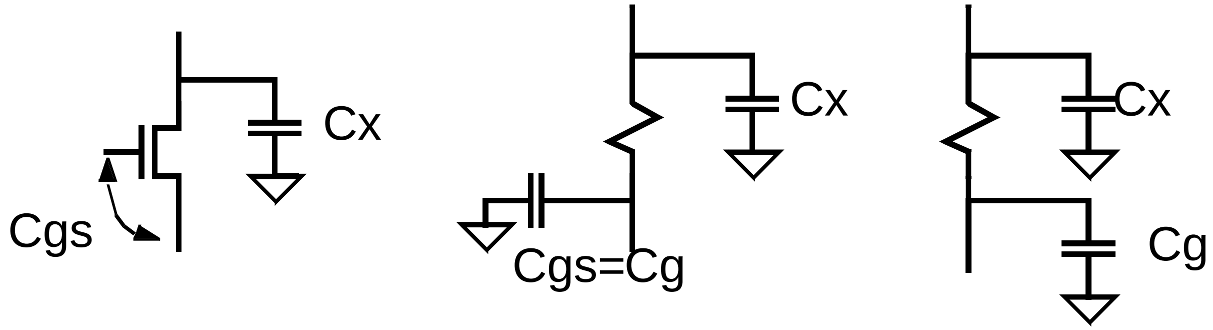
R_{ik} is the resistance of path to ground that is common to nodes i and k

For more details: take EECE 481 next term…

Distributed MOS Networks

We now have the formula we need to use, but we still need to figure out the resistance and capacitance values.

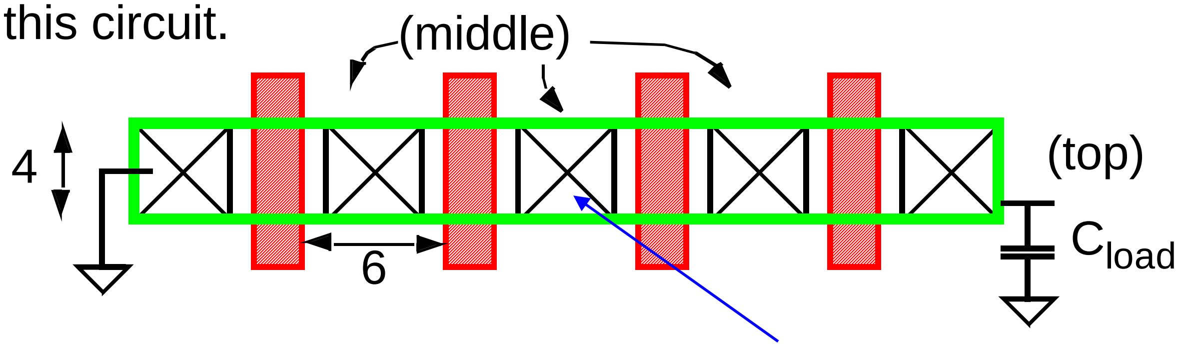
- Resistance is the ‘standard values’
- Need to worry about the gate capacitance



We must include an extra gate capacitance when calculating the source capacitance (we will not include it when finding the drain capacitance, since the whole gate capacitance will be accounted for on the source side of the transistor)

Source Capacitance

Consider this circuit.



Before, we would have estimated the cap. of this region as one of:

(2fF) * (W) = 2 fF * (4/2) = 4fF

or (6/2)*(4/2)*0.2fF/μ2 * (4/2 + 4/2 + 6/2+ 6/2)*0.5fF/μ = 6.2 fF

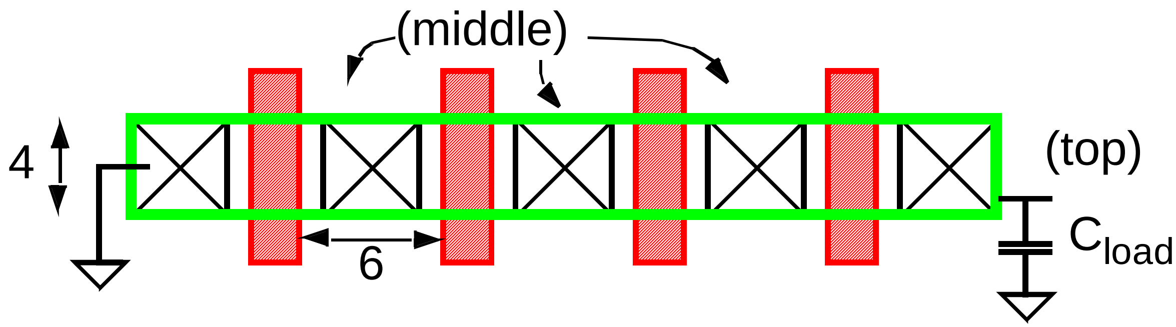
But now, we will add a gate capacitance to the equation (more accurate):

4 fF (from before) + 2fF*W = 8 fF

or 6.2 fF (from before) + 2fF*W = 10.2 fF

Note that in the rest of this course, we will use the simpler 8fF (Rather than 10.2 fF, but just be aware we are making an approximation)

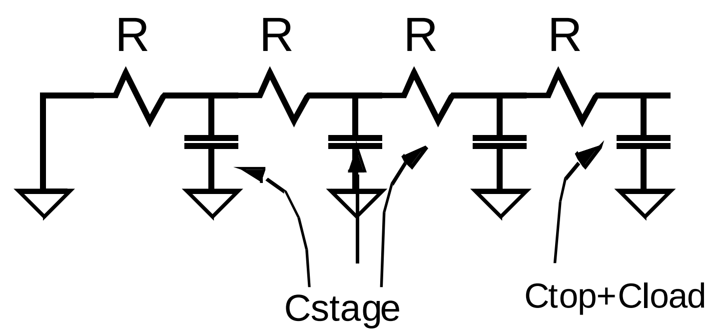
Delay for Series Stacks



Rstage = 1/2 13K

Cstage = 4fF + 4fF = 8fF (middle)

Ctop = 4fF (top)



So delay is:

t = RCstage + 2RCstage + 3RCstage + 4R(Ctop+Cload)

In general for n stages:

t = n(n-1)/2 RCstage+ nR(Cload+Ctop)=0.05ns * n(n-1)/2 + nR(Cload+Ctop)

Delay for a Series Stack

Stack Delay is quadratic in the number of devices.

This for an unloaded stack (only takes into account Ctop)

As we did in the last slide, for true delay we need to add nRCload

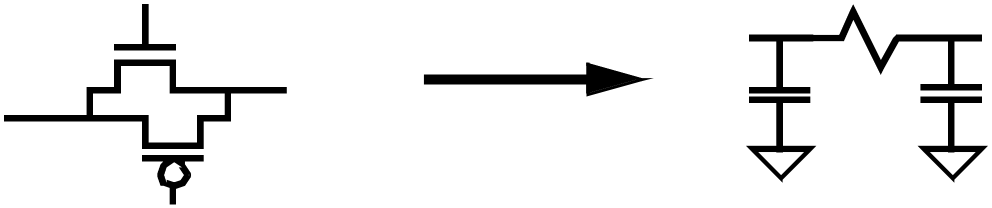
Stack	Delay
1	0.03 ns
2	0.10 ns
3	0.23 ns
4	0.42 ns
5	0.65 ns
6	0.94 ns

This table is only for minimum size devices, with contacts between each stage, but the principle is that tall stacks are slow.

Transmission Gates

CMOS switches are handled using the same method as series stacks.

Question is how to model the resistance



Two transistors in parallel, but one of them is passing its weak value.

Roughly doubles the resistance because Vgs is decreasing

Resistance of nMOS pulling up = 26K/sq

Resistance of pMOS pulling down = 52K/sq

For 1:1 ratio of p to n ratio for the transmission gate

Pull up is 26K (pMOS) || 26K (nMOS) = 13K/sq

Pull down is 13K (nMOS) || 52K (pMOS) = 10K (~ 13K)

So, for our purposes, the resistance is simply 13K/sq

Transmission Gate Model

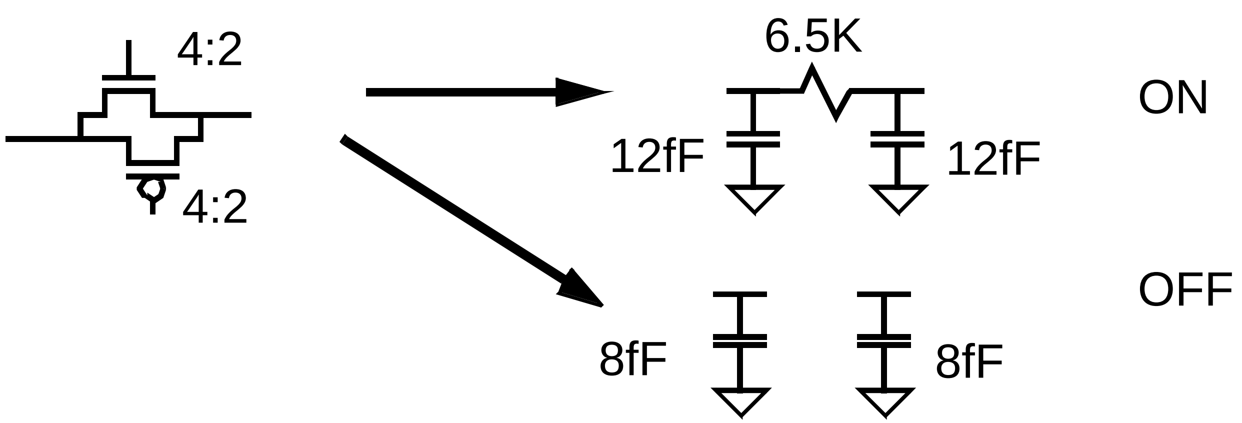
Resistance of 4:2 Transmission Gate:

- 6.5K for pull up and pull down

Capacitance of 4:2 Transmission Gate:

- The source of the pMOS is the drain of nMOS so each diffusion terminal sees one gate capacitance (when the transistors are on -- when off just see the diffusion capacitance)
- 2 diffusion regions + one gate = 2(4fF) + 4fF = 12fF

Model of transmission gate (two cases)



Board Notes: Example of the delay through a transmission gate

Wiring Capacitance

For fastest systems, want the wire capacitance (and hence delay) to be small compared to gate capacitance

- But this leads to very large transistors.
- Compromise is to try to keep ratio from 30% to 70% wire

For a standard cell library how big should the transistors be?

- Want the delay to have some tolerance to placements.
- Implies that the wire capacitance should be a small fraction of total
- Long wires are probably millimeters (.2pF)
- So, transistors should be pretty large (10-20x minimum size)
- OK, since transistors are the free things that fit under wires.

Trend is toward larger transistors. Stick layout diagrams should ‘show’ transistor widths. In industry, you don’t default to minimum size transistors, You should default to 5-10x minimum size so that you can drive the wire and the fanout loading.

Wire Resistance

Previous slides ignored wire resistance

- For short wires this is ok (Rwire « Rtrans)

As wire gets longer

- Rwire gets larger
- Rtrans gets smaller (larger transistor to drive larger capacitance)
- Can become an issue

Wire delay is proportional to length^{2},

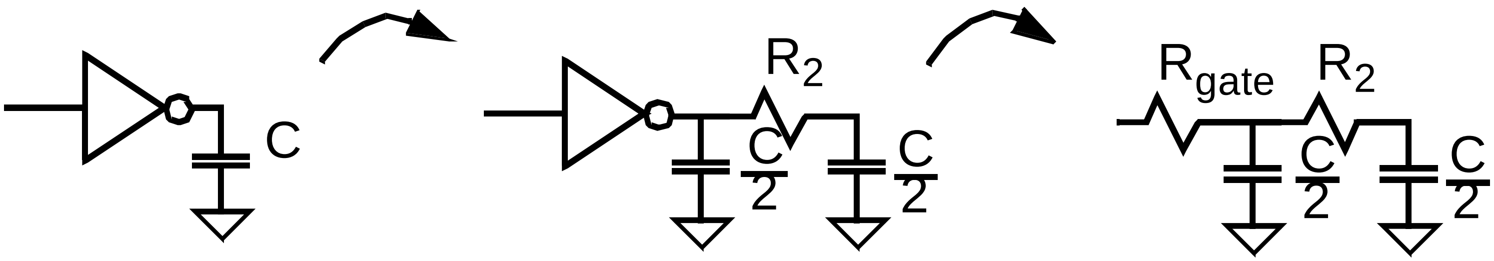
- Capacitance of wire is proportional to length
- Resistance is proportional to length too

Sometimes add repeaters to reduce the total wire delay.

- Break the quadratic increase, but adds buffer delay

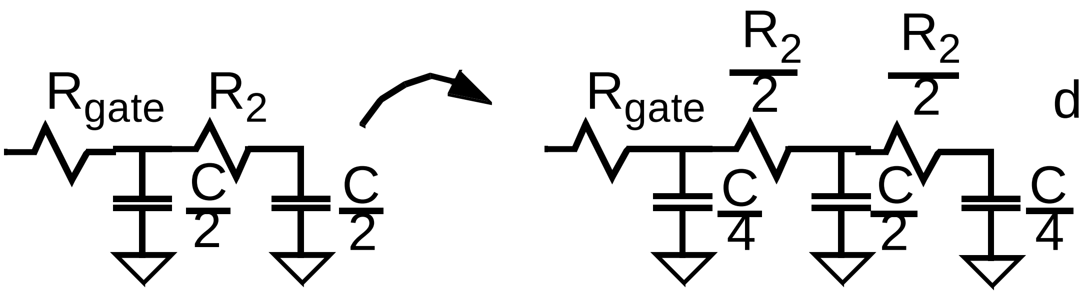
Wires are also Distributed RC; Use “π Model”

The resistance & capacitance of wires are not really lumped at the end. They are really distributed continuously along the length of the wire. One way to model this better is to use a “π model”:



Wires are also Distributed RC; Use "π Model"

We can model the wire by breaking it into any number of lumped elements. In the limit, an infinite number of lumped π segments elements is equivalent to the continuous R and C of the physical wire. But, fortunately, it turns out that the Elmore Delay summing from the previous slide is **independent** of the number of segments chosen.

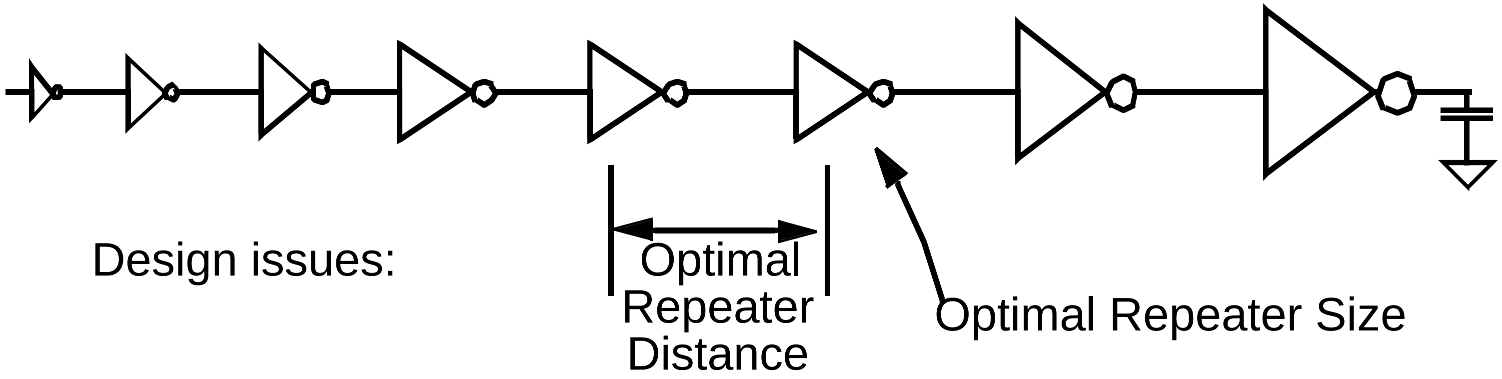


$$\text{delay} = R_{gate}C + \frac{R_2C}{2}$$

Intuitively: we are dividing the wire resistance by 2, because on average, the capacitance has to discharge through half of the wire resistance.

Buffering Long Wires

For long wires, we can insert buffers to reduce the delay:



Added buffer delay is matched by reduced wire delay

- Can use RC model to find optimal length, and repeater size
- This optimum distance is about 1 to 2mm in typical 0.18μ fabs.

General Rules of Thumb (for speed)

- Try to keep the fanout of all gates to be less than 5
- Keep fanin limited to 2, 3 or 4
- Try to keep the delays of the gates in a critical path roughly the same.
  - Large fanin gates should have smaller fanout
- Be flexible on sense of logic (push inversions around)
- Don't use minimum size transistors, unless you know the wire is short