ECE 124d/ ECE 256c
Lab 2

This lab is in two parts, in the first you characterize some representative RC buffering cases, and in the second part, you optimize buffer insertion for practical case.
Part a (due Jan. 26, 2011), Part b (due Feb., 3, 2011).

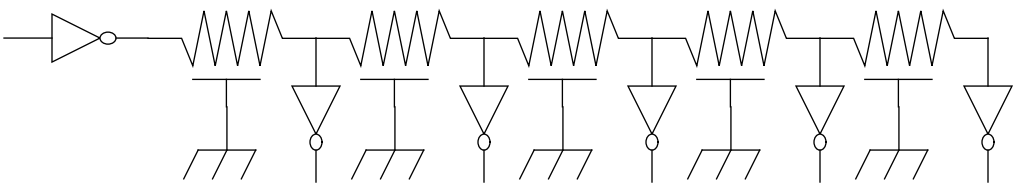
Part a: Modeling

In this lab, you will start with a general distributed LRC model and reduce it to a much simpler model while determining the accuracy of the approximation. First, build a distributed wire model as was done in the first lab, using an LRC distributed model (neglect G). The physical parameters of the wire are as follows:

1. Wire width 210nm
2. Wire thickness: 400nm
3. Material: Cu
4. Distance to substrate from base of wire: 1.2um
5. Dielectric Constant of media: (effective) $2.9\varepsilon_0$

As an approximation of distributed RC lines, Pi-models are commonly used, as are simple lumped Elmore delay models. A pi model for a distributed RC segment models the total resistance as three serial resistors (R/3), with two (C/2) capacitors to ground at the two internal junctions. This model has the advantage that any other model connected to it has to pass current through a resistor before it can get to one of the capacitor nodes (it is generally a really bad idea to have infinite currents present in a simulation). The general practice is to simply substitute a pi-model wherever a distributed RC element is desired, distributing the resistance and capacitance of the segment into the Pi-models. Finally, one can make even a simpler model by lumping each wire segment into a single lumped circuit model consisting of a single resistor and capacitor.

For Buffer models, please use the MMI INV_e extracted spice model for 0.18um technology. These are available on the website. These models are derived from a real 0.18um cell with parasitics extracted. The circuit topology to simulate consists of an inverter-driven distributed line with inverter taps every 1mm along the 5mm long line (shown below). Build 3 models of this network, the first using 1mm segments of distributed LRC model for each RC segment, the second using a Pi-model for each RC segment and the third using a simple RC segment for each RC wire segment.



Simulate these three cases for fast rising, falling and pulse transitions driven from a 50pS source rise and fall time and a total pulse width of 200pS (including the 50pS rising and falling edges). How do the 50% delays at the tap outputs compare for the three models? Replace the 1mm segments with 2mm segments and redo the LRC and Pi-models. How does the accuracy change? Compare these results to Elmore delay for the system where the inverter is modeled as a capacitor on the input and a resistor output as was done in Bakoglu's model.

Part b: Optimization:

Here, you will do the first design lab of the class. Your goal is to propagate a 40pS rising edge arriving at a 1x inverter (defined below) across 5mm of wire to another 1x inverter load located 5mm away. You can add as many buffers of any size at any position you want to minimize the delay (measured at 50% transitions from input pulse to the output of the final buffer). You may not modify the wire parameters at all.

Wire Parameters:

Tech:	45nm	130nm
Vdd	0.8V	1.3V
Width	110nm	260nm
Space	110nm	280nm
Height	350nm	500nm
ILD	150nm	450nm
k	2.4	3.0
R/mm	575Ω	190Ω
L/mm	1.7nH	1.7nH
C/mm	195fF	178fF

1X Inverter Parameters:

Tech:	45nm	130nn
Length	45nm	130nm
Width	90nm	220nm
sdd	200nm	500nm

For this model, you must use at least pi-model segment for each distributed part of the wire. Choose P-transistor widths so that your inverter has equal rise and fall times (otherwise you will get two different solutions for the rising and falling edge cases). Simulate your designs for rising, falling and a 200pS pulse. How do the results differ? Is there any change in pulse width in your design? How does your overall delay compare to the prediction of Bakolgu and to the speed of light limit? (note that k is effectively ε_0).