EE307 Digital Electronics and Integrated Circuits

HW(week 8)7 - Assigned: 2/25/18 Due: 3/2/18

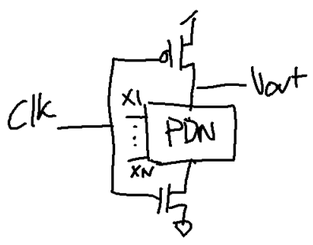
This week's homework is about Elmore delay, setup.and hold times and tclkQ. It also introduces one more ability of LTSpice and builds a couple more blocks of the homework project.

1. Complex CMOS circuit delay.
   a. What is the transistor level circuit for a CMOS circuit that implements the circuit:

      $$Y = \overline{(A \bullet B)} + \overline{(C + \overline{D})}$$

      Check your result against the answer posted on PolyLearn. Use the same order of transistors as in the answer posted on PolyLearn.
   b. Draw in node capacitances on your schematic and name them so I can see which capacitance you are talking about in the rest of the questions. Just name them C1, C2, C3 etc etc. No need to write CGDov or any of that.
   c. Draw a version of the previous circuit after the transition from: A=B=D=1 and C=0. To A=B=C=D=1 using just wires, Cs and Rs. Also mark where the new "input" (Voltage that output is switching to) and Vout (the node that you are finding the delay of).
   d. Assuming each transistor in your circuit has the same R_{ON} of 'R_{ON}', use the Elmore delay technique to find the time constant τ=RC for the transition. Assume the input goes from: A=B=D=1 and C=0. To A=B=C=D=1 Note that I'm saying that D=1. The input to the circuit is actually $\overline{D}$.
   e. Assuming all transistors in your circuit has the same (large signal) R_{ON} of 'R_{ON}', use the Elmore delay technique to find the time constant τ=RC for the transition when the inputs go from A=B=C=D=1, to C→0, A=B=D=1.

2. You may have seen dynamic logic in lab. Dynamic logic looks like this:



The PDN is exactly the same as CMOS. The exact same process is followed to generate the logic for the PDN for CMOS as for dynamic logic. The difference is that dynamic logic uses a clock and there are two steps or phases to each calculation of the output. The first phase is when the clock is low. The PMOS will be conducting and Vout will always be set to VDD. This phase is called the "precharge" phase. The next phase is when the clock goes high. the PMOS is turned off and, if the PDN conducts, the output is pulled down to zero. If the PDN isn't conducting then Vout stays at VDD because the capacitance at Vout doesn't have a path to discharge. Since the pulldown network is designed exactly like the CMOS circuits, you know it will only conduct when the output should be low. There are some good things about dynamic logic:

1. Less transistors in the PUN,
2. Dynamic logic swings rail to rail (just like CMOS).

But its drawbacks are… too much for use these days. Clocks use a ton of power so having to add a clock path to every logic gate in your design will increase your power budget by multiple times on average. Also, to run a dynamic logic circuit faster than CMOS these days, you'd need a terahertz (10^{12}Hz) clock which is close to impossible to create and keep clean.

a. Draw the dynamic logic version of the circuit in question 1. Check out your answer on PolyLearn.
b. For the circuit you found in 2a), For inputs A=B=C=D=1, how fast can the clock go? The duty cycle doesn't need to be 50%. Use, for each node, C=15fF, Ron=145Ω.

3. Use Elmore delay to add in the wire between the inverters to the calculations.
   a. Use Ron_N=145Ω , Ron_P=325Ω, and C on the output of the first inverter Cout=20.2fF. C on the input to the second inverter is 26.82F. The wire is in N+ and N+ has a Ω/□ as shown below. The wire is 2µm long and 0.05µm wide. Ignore the C in the wire (this is OK because the C is much smaller than the transistors and the R is pretty large since we're using N+). The node of interest is at the input of the second inverter since it's that voltage that will start the second inverter's transition to start. The direction of the transition is when Vin goes from 0V to VDD:

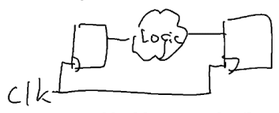
PROCESS PARAMETERS	RR	N+	P+	P+PLY	POLY	N+BLK	P+BLK	N_W	UNITS
Sheet Resistance	1523.7	6.2	8.1	260.7	6.3	72.3	107.7	321	ohms/sq
Contact Resistance		7.2	6.8	6.5	6.7				ohms
Gate Oxide Thickness		46							angstrom

PROCESS PARAMETERS	M1	M2	M3	M4	M5(MT)	M6(ML)	UNITS
Sheet Resistance	57	90	95	92	91	14.29	mohms/sq
Contact Resistance		2.7	2.3	2.25	2.29	0.28	ohms

COMMENTS: BLK is silicide block.

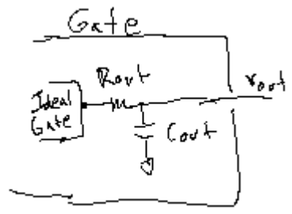
CAPACITANCE PARAMETERS	N+	P+	POLY	D_N_W	N_W	M1	M2	M3	M4	MT	ML	UNITS
Area (substrate)	901	1139	108	220	289	63	41	28	22	19	13	aF/um^2
Area (N+active)			7540									aF/um^2
Area (P+active)			7542									aF/um^2
Area (r well)	873			1088								aF/um^2
Area (MOS varactor@1V)			7776									aF/um^2
Area (HA varactor)		2744										aF/um^2
Area (MiM)			2057									aF/um^2
Area (M1)			191									aF/um^2
Area (M2)						116						aF/um^2
Area (M3)							83					aF/um^2
Area (M4)								91				aF/um^2
Area (MT)									96			aF/um^2
Area (ML)										22		aF/um^2

4. STA: In large IC circuit there may be millions of gates. Simulations are impossible to do and often you just have to trust that the automated tools create a functionally correct circuit (functionally correct means that it follows its truth table if given enough time). Often it would take a prohibitive amount of time to simulate for timing behavior. What you CAN test for is the worst case delay. STA (static timing analysis) uses a technique that black-boxes all of the gates in order to approximate the worst case delay. For each gate the tp, Cout, Cin and Rout (Ron) are given but no information about the functionality of the gate is given. A single type of gate can be used millions of times in a design so it's not as bad as it sounds. You have made symbols for gates and then used a lot of a single gate. For STA to be possible on your designs, you'd have to figure out the Cin, Cout and Rout for the simulator just once and then those values would be used on all instances of that gate. The way the software works is the same as the diagram that we used when talking about setup and hold times:



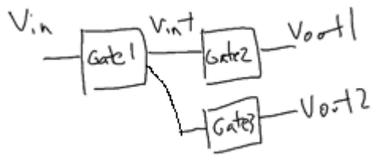
For setup and hold times, the flip-flops are looked at. For the logic, STA is used with the Cin, Cout and Rout of the gates and the tp, Cin, Cout and Rin from the flip-flops.

The Rout and Cout are considered as:



You only see Vout but the R and C are configured as shown internal to the gate.
tp is a delay in addition to the delay caused by Rout, Vout and Cin and is added when you go through a gate.

Inside the logic there will be gates cascaded together in, say, a half adder or a multiplier. The tool also assigns resistance and capacitance to wires if they are longer than 1µm and/or if width is less than 200nm. Assuming that wires all are longer than 1µm and are all 200nm wide and come with the L given as LVin, and/or LVint, come up with a general equation for the delay from Vin to Vout1 of the following gates using Elmore. Ignore wire C. Assume Vin changes instantaneously so that first Cin doesn't affect the delay. Find an equation for the delay to Vout1 and Vout2. (Not a numerical value – Just an equation).



5. Setup and hold times:

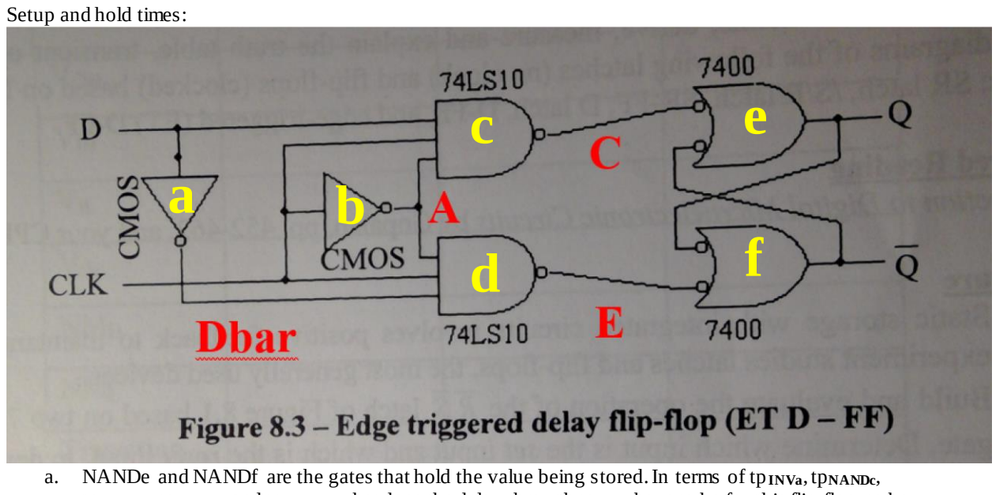
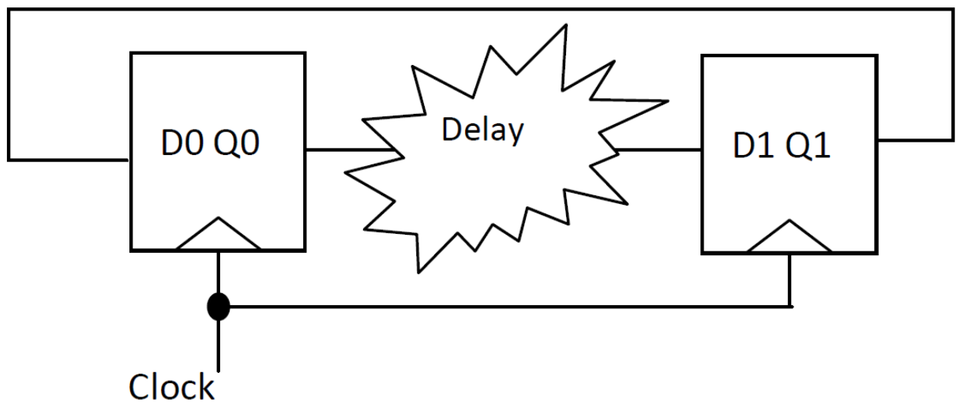


Figure 8.3 – Edge triggered delay flip-flop (ET D – FF)

a. NANDe and NANDf are the gates that hold the value being stored. In terms of tp_{INVa}, tp_{NANDc}, tp_{NANDd}, tp_{NANDe}, and tp_{NANDf}, what does the delay through tp_{INVb} have to be for this flip-flop to be able to completely grab a new value? You might not need all of the delays listed (tp_{INVa}, tp_{NANDc}, tp_{NANDd}, tp_{NANDe}, and tp_{NANDf}).
b. What is the hold time? Express the answer in terms of the values you need from: tp_{INVa}, tp_{INVb}, tp_{NANDc}, tp_{NANDd}, tp_{NANDe}, and tp_{NANDf}.
c. What is the $t_{CLK \rightarrow Q}$ time? Express the answer in terms of the values you need from: tp_{INVa}, tp_{INVb}, tp_{NANDc}, tp_{NANDd}, tp_{NANDe}, and tp_{NANDf}.

d. What is the t_{setup} time? Express the answer in terms of the values you need from: tp_{INVa}, tp_{INVb}, tp_{NANDc}, tp_{NANDd}, tp_{NANDe}, and tp_{NANDf}.
e. Setup and hold time violations.



i. Is there a violation for these two systems? If there is/are, what kind is/are it/they and show the math for it/them: 3ns ≤ tdelay ≤ 6ns. Clock frequency is 50MHz.

	Flip-flop 0	Flip-flop 1
Hold time	6ns	8ns
Setup time	11ns	20ns
TclkQ	2ns ≤ TclkQ ≤ 8ns	5ns ≤ TclkQ ≤ 10ns

ii. Will this ever work even with an adjustment of the clock speed? If it won't, what is the name of the violation that can't be fixed with an adjustment of the clock? (HINT: The next question tells you the answer to the first part of this question).
iii. How can you fix the violation you found in question 5eii? Assume all delays are in 1ns steps. No fractions of a ns. Assume you have buffers with a 1ns delay. Draw the new circuit.
iv. With the fix you suggested in 5eiii, what is the maximum clock speed you can use? Assume all delays are in 1ns steps. No fractions of a ns.
v. What violation can be fixed by slowing the clock?