# EECS 151/251A Homework 6

Instructor: Prof. John Wawrzynek, TAs: Christopher Yarp, Arya Reais-Parsi

Due Monday, Mar 11th, 2019

## Problem 1: Flip-Flop Malfunction [5 pts]

The positive edge triggered flip-flop presented in lecture could malfunction if the output load capacitance is too high. Explain in detail how this failure could occur. Suggest a modification to the circuit that would fix this problem.

Solution:

The effective resistance through a transistor is an inverse function of its width. The wider it is, the less effective reisistance. The time required to bring a capacitive load to a set voltage is related to the time constant $\tau = RC$. If the resistance is too high, the capacitive load may not reach the required voltage by the next clock edge. For the Flip-Flop to function properly, the output node needs to be stable by the time the negative clock edge comes and the second stage transmission gate is made transparent. If it is not, the wrong logic level may be passed to the inverter chain which would ultimately result in the proper value not being held for the entire half clock cycle.

One method of addressing this problem is to size the last inverter in the Flip-Flop to have less effective resistance. However, increasing the transistor width also increases the gate capacitance, which is undesirable.

An alternative solution is to create a chain of inverters or buffers after the Flip-Flop. The optimal sizing of the buffer in this chain is the focus of Problem 3.

## Problem 2: Buffering Long Wires [20 pts]

Assume an inverter (of size 1) is driving another inverter (size 1) through a wire of length $L$. Because of wire delay you plan to divide the wire into $N$ sections of equal length and insert size 1 inverters as buffers to minimize the total delay. The wire capacitance per unit length is $c_w$ and resistance per unit length is $r_w$. Assume that the inverter input capacitance $C_g = 10c_w$ and the inverter drive resistance $R_{dr} = 100r_w$. Also assume that the internal capacitance $C_{int} = C_g$.

Derive an expression for $N$ as a function of $L$ that minimizes the delay.

Solution:

From Lecture 11, slide 11. The propagation delay of a gate driving a wire as well as a load capacitance can be expressed with the following equation:

$$t_p = 0.69R_{dr}(C_{int} + C_{fan}) + 0.69(R_{dr}c_w + r_wC_{fan})L + 0.38r_wc_wL^2$$

where $R_{dr}$ is the drive resistance, $r_w$ is the resistance per unit length of wire, $c_w$ is the capacitance per unit length of wire, $L$ is the lenngth of the wire, and $C_{fan}$ is the capacitance of the fanout (the load).

Using the given information (inverter input capacitance $C_g = 10c_w$, $R_{dr} = 100r_w$, $C_{int} = C_g$) and noting that the load capacitance of 1 section is the input capacitance of the next inverter, let us derive the delay for a length $l$ section of wire:

$$\begin{aligned}
t_p &= 0.69(100r_w)(C_g + C_g) + 0.69(100r_wc_w + r_wC_g)L + 0.38r_wc_wL^2\\
&= 138r_wC_g + (69r_wC_w + 0.69r_wC_g)l + 0.38r_wc_wl^2\\
&= 138r_w(10c_w) + (69r_wC_w + 0.69r_w(10c_w))l + 0.38r_wc_wl^2\\
&= 1380r_wc_w + (69r_wC_w + 6.9r_wc_w)l + 0.38r_wc_wl^2\\
&= 1380r_wc_w + 75.9r_wc_wl + 0.38r_wc_wl^2
\end{aligned}$$

Because we are dividing the wire into N equally sized sections, let us calculate the delay for a section of length $l = L/N$.

$$t_{p_l} = 1380r_wc_w + 75.9r_wc_w\left(\frac{L}{N}\right) + 0.38r_wc_w\left(\frac{L}{N}\right)^2$$

Summing the delay through all sections

$$\begin{aligned}
t_{p_{tot}} &= Nt_{p_l}\\
&= 1380r_wc_wN + 75.9r_wc_wL + 0.38r_wc_w\left(\frac{L^2}{N}\right)
\end{aligned}$$

We want to find a number of stages N that minimizes the total delay for a given $L$

Find local minima of $t_{p_{tot}}$ by finding when the slope of the function is 0:

$$\begin{aligned}
\frac{dt_{p_{tot}}}{dN} &= 1380r_w c_w + 0 - 0.38r_w c_w \frac{L^2}{N^2} \\
0 &= 1380r_w c_w - 0.38r_w c_w \frac{L^2}{N^2} \\
0.38\frac{L^2}{N^2} &= 1380 \\
1380N^2 &= 0.38L^2 \\
N^2 &= \frac{19}{69000}L^2 \\
N &= \pm\sqrt{\frac{19}{69000}}L
\end{aligned}$$

Note that $L$ cannot be negative and therefore:

$$N = \sqrt{\frac{19}{69000}}L$$

We next need to confirm that this point is in fact a local minimum. We can do this using the 2nd derivative test.

$$\frac{d^2t_{p_{tot}}}{d^2N} = 0 + 0.76r_w c_w \frac{L^2}{N^3}$$

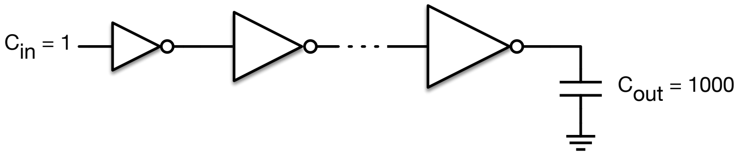
At $N = \sqrt{\frac{19}{69000}}L$, the second derivative is positive, so this is a local minimum.

The function $t_{p_{tot}}$ does have a discontinuity for $N \to 0^+$ where $t_{p_{tot}} \to \infty$.

Therefore, the delay is minimized when the wire is split into $N = \sqrt{\frac{19}{69000}}L$ segments.

## Problem 3: Buffer Chain [5 pts]

Assume that we have the situation shown below with an inverter chain used to drive a large capacitive load (F = 1000) with minimal delay. How many buffers (inverter stages) would be optimal (or near optimal) in this case? What should be the fanout, f, be at each stage?



Solution:

We know from lecture that a simple, common choice of fanout is $f = 4$ close to the optimal for $\gamma = 1$. We also know that the fanout of each stage should be roughly equal. With that in mind, we can come up with the number of stages necessary:

$$\begin{aligned} f^N &= 1000 \\ N &= \log_f(1000) \\ &= 4.9829 \end{aligned}$$

We can't have fractional inverters, so we round this up to $N = 5$. Now we can determine the fanout of each stage again either (a) assuming that each stage is equal:

$$\begin{aligned} f &= \sqrt[5]{1000} \\ &= 3.9811 \end{aligned}$$

or (b) setting as many stages as possible to $f = 4$ and adjusting the last one, $f'$:

$$\begin{aligned} f' &= 1000/4^4 \\ &= 3.9062 \end{aligned}$$

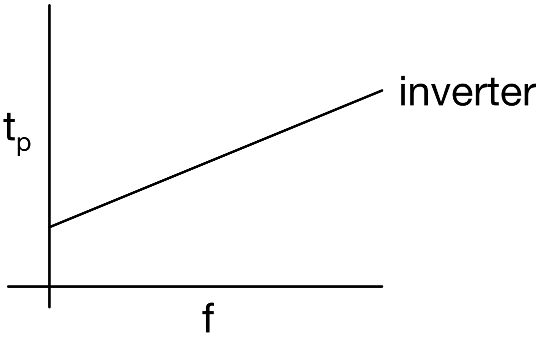
## Problem 4: Gate Propagation Delay Derivation [30 pts]

Derive the formulas for gate propagation as a function of fanout, $f$, for a 2-input NAND gate and a 2-input NOR gate as we did in lecture for the inverter. In both cases, derive the equations based on the input connected to the transistor closest to the output. In the case of the NAND gate assume the other input had been set to 1 (for a long time), and for the NOR, assume the other input had been set to 0. Size the transistors so that the capacitance of each gate input is equivalent to the input capacitance of the inverter. Also assume that the resistance of the pFET is twice that of the nFET ($Rp = 2Rn$) if the pFET and nFET have the same width. Size the transistors so that the rise time and fall times are equivalent. (Note: For this problem, when 2 transistors are in series, ignore the capacitance at their shared node.)

(a) 2-input NAND equation: $t_p =$?

(b) 2-input NOR equation: $t_p =$?

(c) Sketch the curves for the NAND and NOR

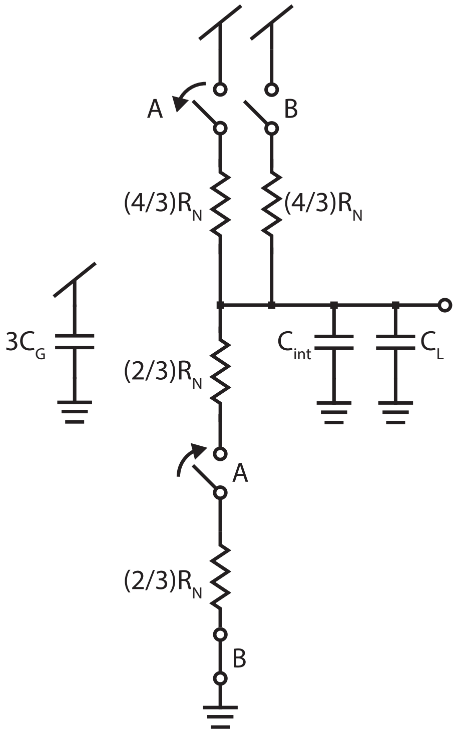


(d) **251A only — *Optional* Challenge Question for 151:** Without working it out in detail, sketch the curve you would expect for a 4-input NAND.

(e) **251A only — *Optional* Challenge Question for 151:** Now assume that for the 2-input NAND we were to consider the other input (the one furthest away from the output). Without working it out in detail, explain how it would effect the result.

Solution:

(a) We can model the 2-input NAND as follows, using the assumptions given in the question ($B = 0$ and ignoring the capacitance at the shared node between transistors in series):



We're told to assume that PMOS resistance is twice NMOS resistance. Since a single PMOS ends up filling the output capacitance while two series NMOS transistors drain it, we just have to set the PMOS and NMOS transistors to be the same size to achieve equal rise and fall times. To see why, consider this means we require $R_P = 2R_N$ in terms of their unit resistances, but $R_P$ already *is* $2R_N$ due to the differing PMOS/NMOS charge mobility.

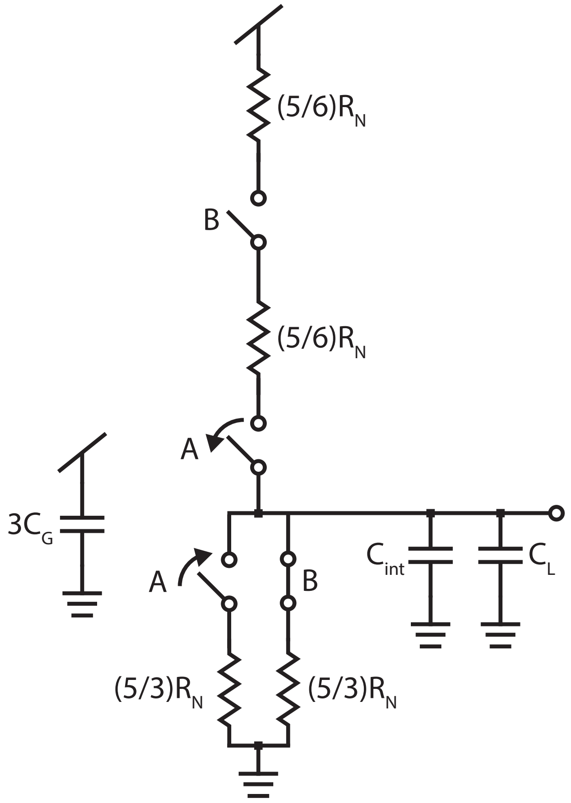
We're also told to size the input capacitances so that they equal an inverter (*per input*). Since our NMOS and PMOS transistors are the same size, we only have $2C_G$ units of capacitance at the input. We have to increase the dimensions of each transistor by $\frac{3}{2}$ relative to the inverter. This decreases the series resistances in the charge and discharge paths to $2 \cdot (2/3)R_N$ each, and makes $C_{IN} = 3C_G$.

The internal capacitance, $C_{int}$, is the sum of drain capacitances in transistors whose drains share the output node (2 PMOS and 1 NMOS): $C_{int} = (6/2)C_D + (3/2)C_D = (9/2)\gamma C_D$

Now, also adding in an additional (but arbitrary) scaling factor $W$ as in lecture,

$$\begin{aligned}
t_p &= 0.69 \cdot 2\left(\frac{2R_N}{3W}\right)(C_{int} + C_L) \\
&= 0.69\left(\frac{4R_N}{3W}\right)\left(\frac{9}{2}\gamma W C_G + C_L\right) \\
&= 0.69\left(\frac{R_N}{W}\right)\left(6\gamma W C_G + \frac{4}{3}C_L\right) \\
&= 0.69\left(\frac{R_N}{W}\right)3\gamma W C_G\left(2 + \frac{4C_L}{3\gamma\,(3WC_G)}\right) \\
&= [0.69 \cdot 3R_N\gamma C_G]\left(2 + \frac{4C_L}{3\gamma C_{IN}}\right) \\
&= t_{p0}\left(2 + \frac{4f}{3\gamma}\right)
\end{aligned}$$

(b) We can model the 2-input NOR as follows:



To match rise and fall times, the series resistance of the PUN and PDN in the charge and discharge cases must be the same. Since we now have two PMOS rtransistors in series with only one NMOS, we have to set $2R_P/W_T = R_N$, where $W_T$ is the relative sizing difference between the P- and NMOS devices. Since $R_P = 2R_N$, we solve for $W_T = 4R_N/R_N = 4$. That is, the PMOS devices must be 4 times larger than the NMOS ones with a gate capacitance of $4C_G$. The total input capacitance is thus $C_{IN} = 5C_G$.

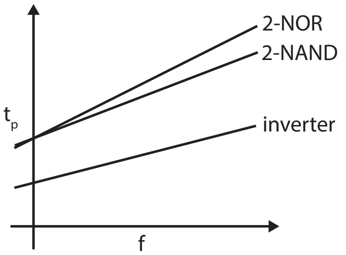
Repeating the solution to part (a), we need to scale all the transistor widths by 3/5 to ensure that the total input capacitance is the same as the inverter ($3C_G$). Now the resistance in the charge/discharge paths is $5R_N/3$, $C_{IN} = 3C_G$.

The internal capacitance comes from the 2 NMOS and 1 PMOS transistors sharing the output node: $C_{int} = 2 \times (3/5)C_G + (12/5)C_G = (18/5)C_G$.

Again,

$$\begin{aligned}
t_p &= 0.69\left(\frac{3R_N}{5W}\right)(C_{int} + C_L) \\
&= 0.69\left(\frac{5R_N}{3W}\right)\left(\frac{18}{5}\gamma W C_G + C_L\right) \\
&= 0.69\left(\frac{R_N}{W}\right)\left(6\gamma W C_G + \frac{5C_L}{3}\right) \\
&= 0.69\left(\frac{R_N}{W}\right)3\gamma W C_G\left(2 + \frac{5C_L}{3\,(3\gamma W C_G))}\right) \\
&= [0.69 \cdot 3R_N\gamma C_G]\left(2 + \frac{5C_L}{3\gamma C_{IN}}\right) \\
&= t_{p0}\left(2 + \frac{5f}{3\gamma}\right)
\end{aligned}$$

(c) The y-intercepts for NAND and NOR are both twice that of the inverter. The NAND line has a gradient 4/3 that of the inverter (steeper); for NOR it is 5/3 (steepest).



(d) Again consider the input to the PDN closest to the output. To balance pull-up and pull-down times, we now need the PMOS FETs to be half the width of the NMOS FETs (assuming still that in terms of unit resistance $R_P = 2R_N$). The total capacitance for this input is now $1.5C_G$ - in order to make it equal the inverter, we have to double the size of both the P- and NMOS devices. With the arbitrary scaling factor W, the effective resistance of the PMOS FET becomes $R_{PFET} = \frac{2R_N}{W}$, and that of the NMOS FET becomes $R_{NFET} = \frac{R_N}{2W}$.

The internal capacitance becomes $C_{int} = 4WC_D + 2WC_D = 6W\gamma C_G$.

Solving for $t_p$,

$$\begin{aligned} t_p &= 0.69\left(\frac{2R_N}{W}\right)(C_{int}+C_L) \\ &= 0.69\left(\frac{2R_N}{W}\right)(6\gamma W C_G + C_L) \\ &= 0.69\left(\frac{3R_N}{W}\right)\gamma W C_G\left(4+\frac{2C_L}{3\gamma W C_G}\right) \\ &= t_{p0}\left(4+\frac{2f}{\gamma}\right) \end{aligned}$$

What we've done is derive logical effort $g$, where $t_p = t_{p0}\,(p + gf/\gamma)$. $g$ for a 4-input NAND is[c] $(n+2)/3 = 2$, consistent with our solution. As a result we'd expect the "curve" for 4-NAND to have twice the gradient of the inverter.

(e) Consider the other input in the PDN. When switching the lower NMOS FET, the node between the two NMOS FETs will need to be discharged before discharging the common drain node at the output. Therefore the gate will have greater delay.

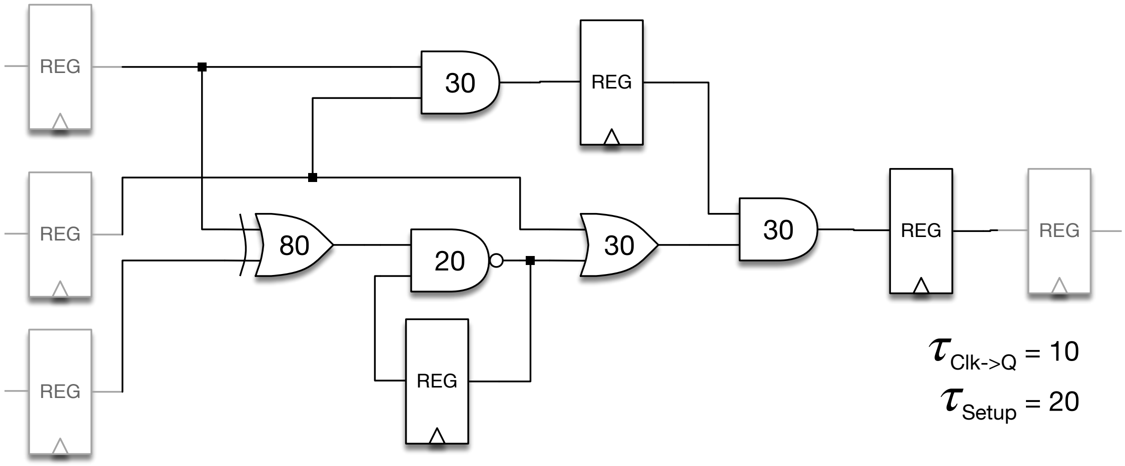
[c] See Rabaey et. al, *Digital Integrated Circuits, A Design Perspective, 2nd Edition, Table 6-5.*

## Problem 5: Critical Path & Retiming [10 pts]

A circuit for a hypothetical process is shown below. The gate delay (in ps) is written in each gate.

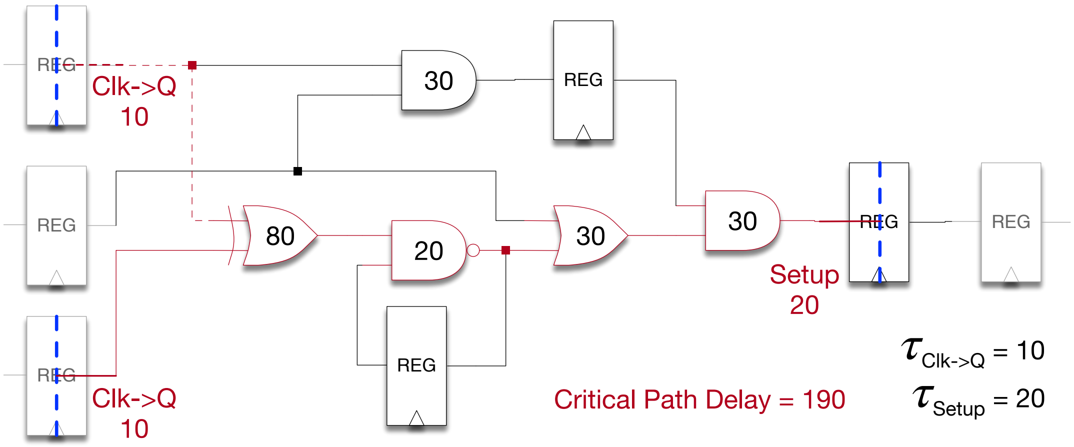
(a) Without modifying the circuit, derive the maximum clock frequency.

(b) Now retime the circuit (without changing the latency between any input and the output) to maximize the clock frequency. Assume that you cannot move the gray registers on the border of the circuit. Draw the new circuit and derive the new maximum clock frequency.

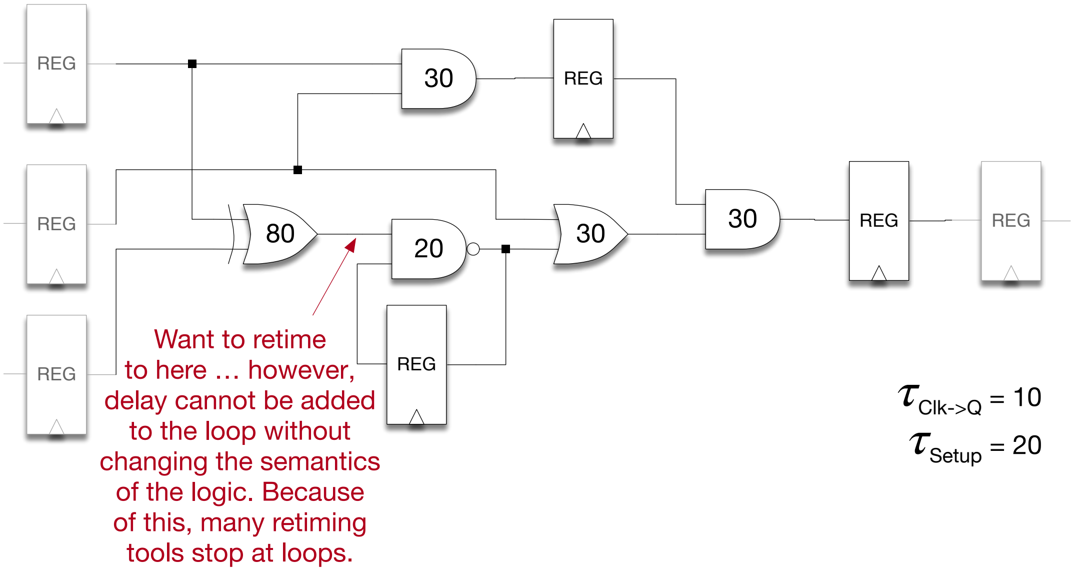


Solution:

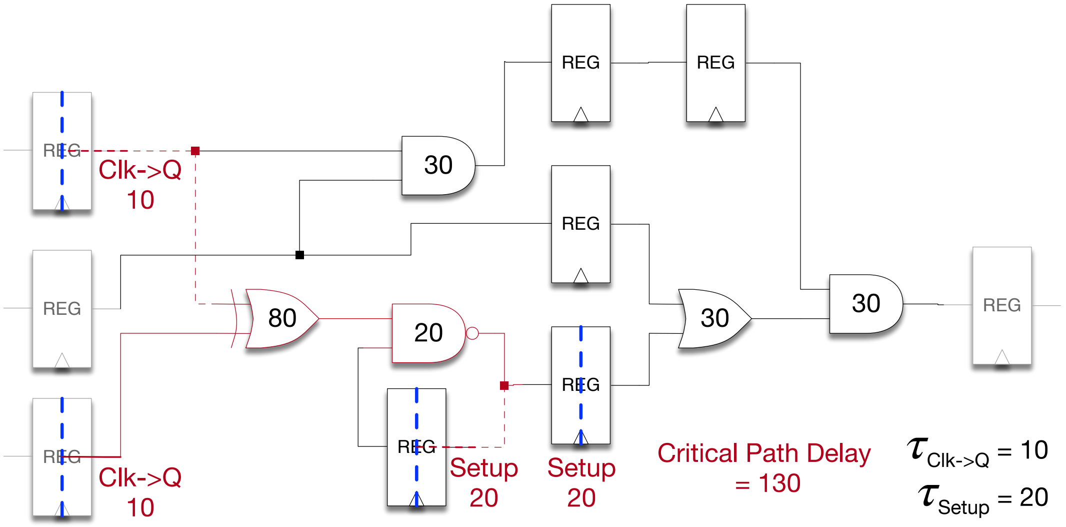
The critical path(s) for the circuit are shown below. The critical path is 190 ps supporting a maximum clock rate of 5.26 GHz.


REG
REG
REG
REG
REG
REG
REG
Clk->Q
10
Clk->Q
10
30
80
20
30
30
Setup
20
Critical Path Delay = 190
$\tau_{\text{Clk->Q}} = 10$
$\tau_{\text{Setup}} = 20$

We would ideally like to re-time the circuit so that the delay is split evenly. Unfortunately, that is not possible without splitting apart gates. Our second choice would be to re-time a register to the node pointed to in the diagram below.


REG
REG
REG
REG
REG
REG
REG
30
80
20
30
30
Want to retime to here … however, delay cannot be added to the loop without changing the semantics of the logic. Because of this, many retiming tools stop at loops.
$\tau_{\text{Clk->Q}} = 10$
$\tau_{\text{Setup}} = 20$

Because that involves retiming across a loop, many re-timing aware tools would stop before that. This is because the delay in the feedback loop cannot be changed without changing the semantics of the problem. The tool would therefore settle for the less optimal solution which is shown below. Note that registers were added to the design to match the delay in parallel paths. The new critical path is 130 ps supporting a maximum clock frequency of 7.69 GHz.

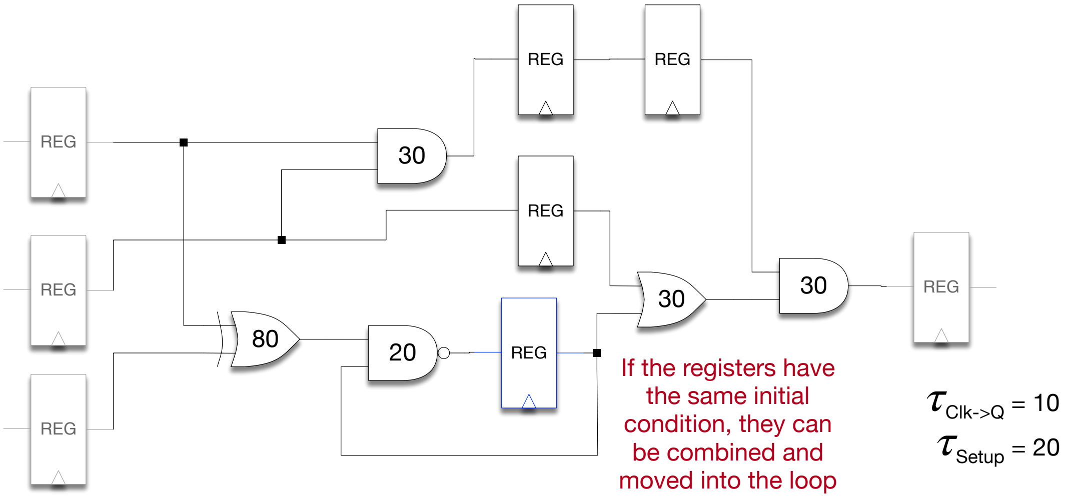


This is the retimed solution that
many retiming aware tools will stop at.

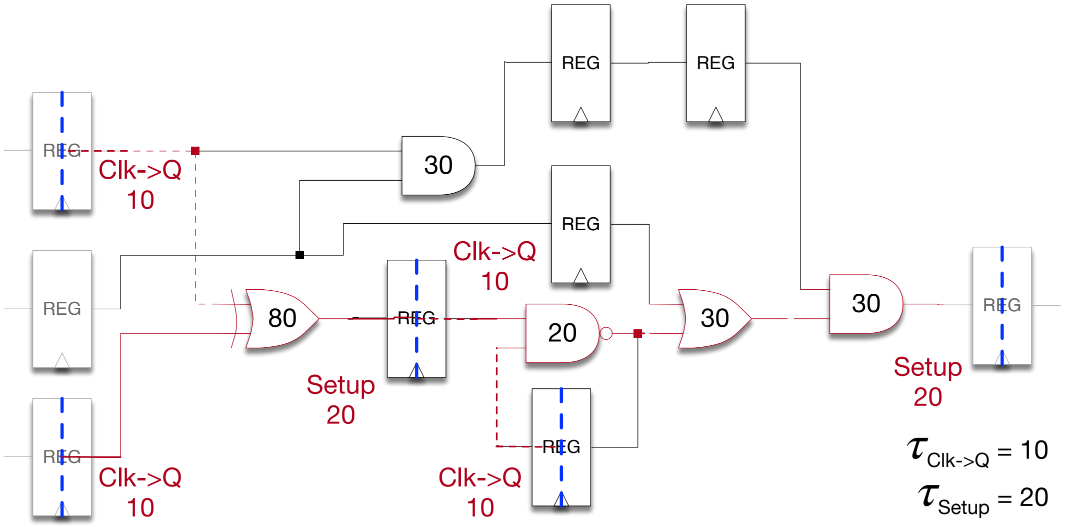
*This is also the optimal solution when
the initial values of the registers are not given.*

Retiming across the entire loop may possible, however, if some conditions are met. Given the retimed solution above, if the register in the loop and the register after the loop have the same initial values, they can be combined inside the loop. This transformation keeps the same delay in the loop and maintains the same delay on the output path from the loop. *Note that we did not tell you what the initial values of the registers were in this problem, so you would be correct if you stopped at this point.*

*Assuming the registers have the same initial value*, the circuit can be modified as shown below:



At this point, the register can be moved through the NAND gate. It is *imperative* that the initial conditions of the registers are set properly as the result after the first cycle will potentially propagate forever in the feedback loop.



The register can be moved through the NAND gate, producing, a register at each input. The total delay in the loop is unchanged. Care must be taken to properly set the initial conditions of the registers.

Critical Path Delay = 110

This version of the circuit has 4 paths tied for the critical path of 110 ps supporting a max clock frequency of 9.09 GHz.