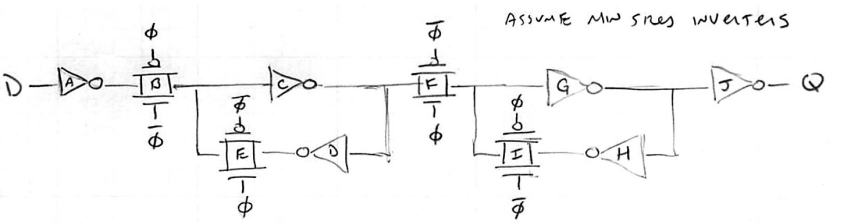
ECE 6745 Complex Digital ASIC Design
Topic 10: CMOS Sequential State

School of Electrical and Computer Engineering
Cornell University

revision: 2025-02-27-12-01

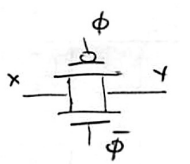
1. Basic Flip-Flop

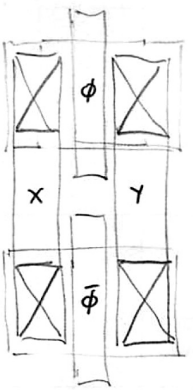
- Basic transmission gate master/slave flip-flop



- Why do we add extra inverters at the input and output?
- We want to characterize the: setup time, clock-to-Q propagation delay, and hold time
- Logical effort will not directly be of much use since transmission gates couple the delay of various gates together!
- We will need to directly use RC modeling, but first we need an RC model of a transmission gate

RC Model for Transmission Gate





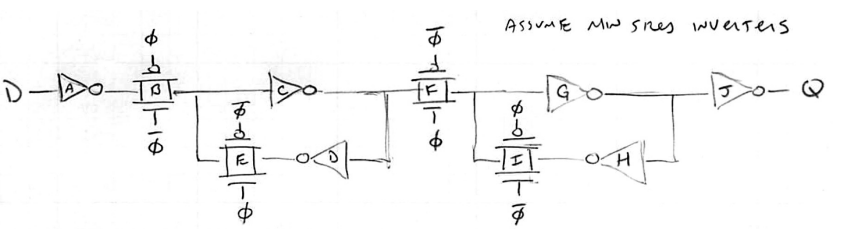
- Assume size of NMOS and PMOS are both minimum width
- Assume a transistor passing a weak value has 2x worse effective resistance
- Derive an RC model for this transmission gate

2. Delay

Setup Time	Latest the input can change before edge while still capturing the value
Hold Time	Earliest the input can change after edge while still capturing the value
Propagation Delay	Delay after clock rises until output is stable
Contamination Delay	Minimum delay after clock rises until output changes

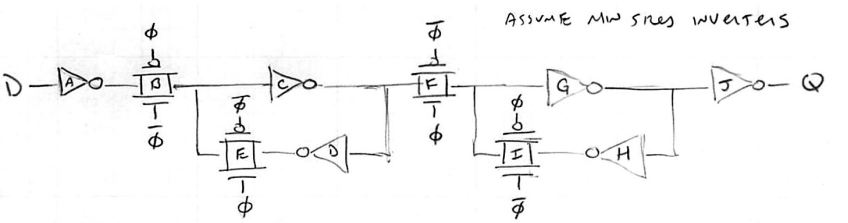
2.1. Setup Time

- To calculate the setup time we ask ourselves, "How far does the input signal need to propagate so that we can reliably flip the master latch before the clock edge?"



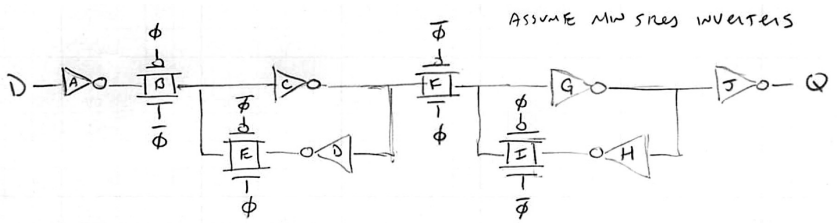
2.2. Clock-to-Q Propagation Delay

- To calculate the propagation delay, we ask ourselves, "How long does it take after the rising edge to propagate the internal state?"



2.3. Hold Time

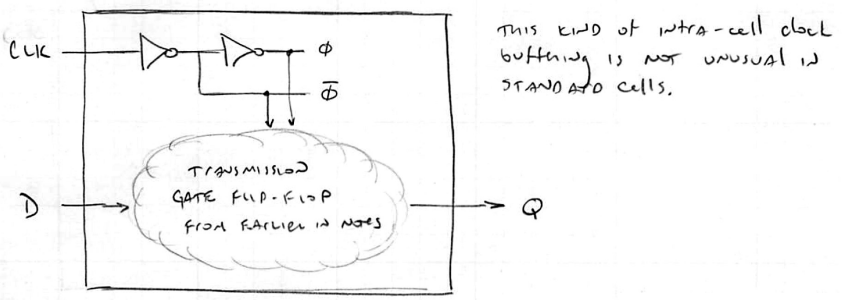
- Hold time is how long we need to keep the input stable *after* the rising edge in order to prevent corrupting the state



- If we assume θ and $\overline{\theta}$ change instantaneously on the rising clock edge, then we actually have a *negative* hold time
- If the input changes right after the edge then by the time it gets to the first transmission gate (gate B) that gate is already open and input signal cannot corrupt the state
- In fact, the input can change a little *before* the edge since it takes some time to propagate through the first inverter

2.4. Internal Clock Delay

- What if we assume there is some delay between the clock input pin of our cell and the actual θ and $\overline{\theta}$ signals?

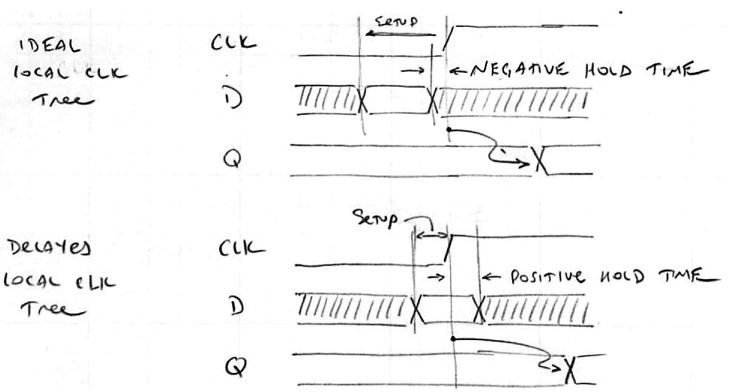


- How does this impact setup time, propagation delay, hold time?
- Remember that all three metrics are defined with respect to the clock pin of the cell *not* the internal θ and $\overline{\theta}$ signals
- First calculate the delay through the local clock tree (clock insertion delay), then factor this into these three metrics

Calculate the delay through the local clock tree

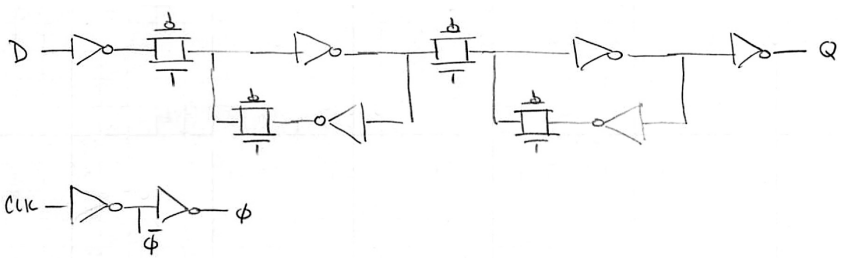
- What is the output load on the local clock tree?

- Essentially what we have done is shift the sampling window



3. Energy

- Dynamic energy in a flip-flop comes from two sources:
  - toggling the data lines
  - toggling the clock
- First, let's label all of the gate and parasitic caps



- Estimate worst case energy per write/read access by calculating the maximum switched cap while assuming every node toggles

- Let's assume the following:
  - clock rate of 500MHz
  - data activity factor of 0.1
  - clock activity factor of 2 (toggles twice per cycle!)

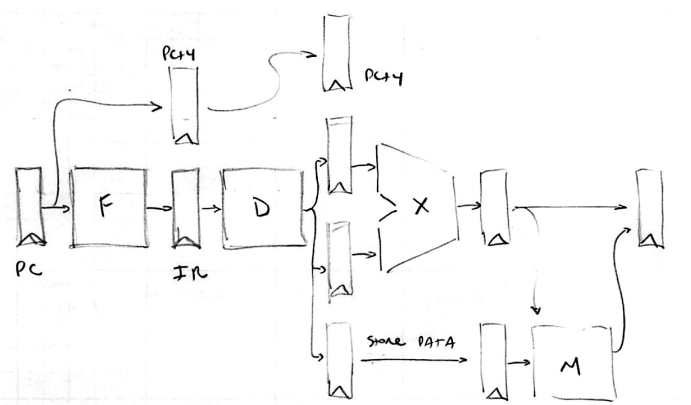
$$E_{\text{data}} = \alpha \frac{1}{2} C {V_{dd}}^2 = (0.1)\frac{1}{2} \times 52\text{C} \times 0.5\frac{\text{fF}}{\text{C}} \times (1\text{V})^2 \qquad = 1.3 \text{ fJ}$$

$$P_{\text{data}} = \alpha f \frac{1}{2} C {V_{dd}}^2 = (0.1)(0.5 \times 10^9)\frac{1}{2} \times 52\text{C} \times 0.5\text{fF/C} \times (1\text{V})^2 \qquad = 0.65 \text{ µW}$$

$$E_{\text{clock}} = \alpha \frac{1}{2} C {V_{dd}}^2 = (2.0)\frac{1}{2} \times 20\text{C} \times 0.5\text{fF/C} \times (1\text{V})^2 \qquad = 10 \text{ fJ}$$

$$P_{\text{clock}} = \alpha f \frac{1}{2} C {V_{dd}}^2 = (2.0)(0.5 \times 10^9)\frac{1}{2} \times 20\text{C} \times 0.5\text{fF/C} \times (1\text{V})^2 \qquad = 5 \text{ µW}$$

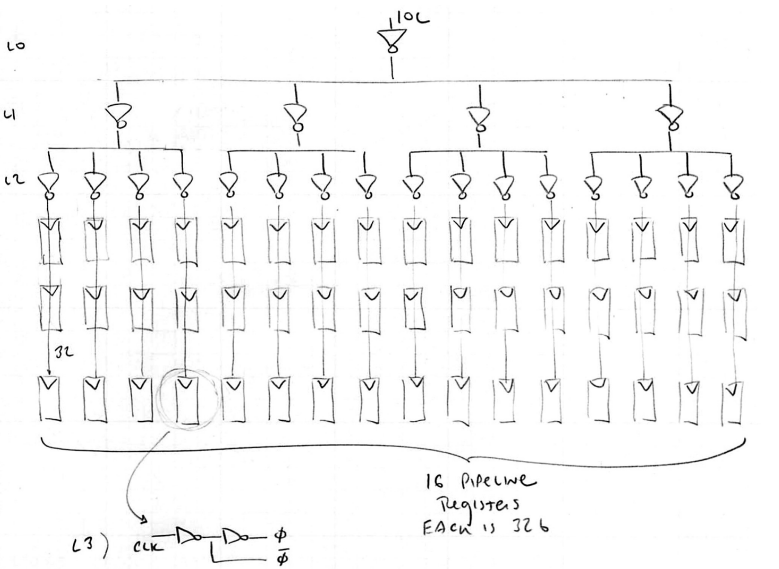
- Once we factor in activity factor, clock energy/power is significantly higher than data energy/power
- Let's estimate the total data/clock energy for a simple processor



- With 512 bits, the total data/clock energy/power is:

$$
\begin{aligned}
E_{\text{data}} &= 512 \times 1.3 \text{ fJ} &&= 666 \text{ fJ} \\
P_{\text{data}} &= 512 \times 0.65 \text{ µW} &&= 333 \text{ µW} \\
E_{\text{clock}} &= 512 \times 10 \text{ fJ} &&= 5.1 \text{ pJ} \\
P_{\text{clock}} &= 512 \times 5 \text{ µW} &&= 2.6 \text{ mW}
\end{aligned}
$$

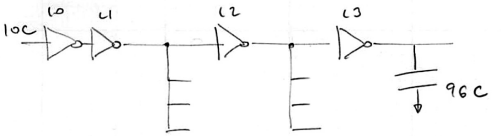
- This ignores the clock tree; let's try and estimate the power consumed in the clock tree



- What is the total clock load?
- How should we size the inverters in the clock tree to reduce delay?
- Note that skew is much more important than absolute delay, but we still need a relatively fast tree to avoid very bad slew rates

- We can use logical effort to size the clock tree

WHAT ABOUT A four-level clock tree?



$F = GBW = 1\,(4\times4)\,96/10 = 153.6$

$\hat{f} = \sqrt[4]{F} = 3.52$

$\hat{D} = 4\,(3.52) + 4 = \underline{18.1\,\tau}$ COMPARED TO 19.2 WITH ONLY 3 LEVELS

SO OUR four-level clock tree IS ~5% FASTER BUT AT WHAT COST?

$C_{IN,C3} = (1/3.52)\ 96C = 27.3C \rightarrow 27C$

$C_{IN,C2} = (1/3.52)\ 27.3C \times 4 = 31C \rightarrow 30C$

$C_{IN,C1} = (1/3.52)\ 31C \times 4 = 35C \rightarrow 36C$

$C_{IN,C0} = (1/3.52)\ 35C = 9.94C \rightarrow 9C$

ROUND TO NEAREST EVEN MULTIPLE OF MINIMUM SIZED INVERTER

TOTAL SWITCHING CAP?

$C_{SW,C3} = 27C \times 2 \times 16 = 864C$

$C_{SW,C2} = 30C \times 2 \times 4 = 240C$

$C_{SW,C1} = 36C \times 2 \times 1 = 72C$

$C_{SW,C0} = 9C \times 2 \times 1 = 18C$

TOTAL C_{SW} for 4 STAGE TREE IS 1194 C

SO 4 STAGE TREE IS 5% FASTER BUT ~1.7x MORE ENERGY/POWER/AREA
LET'S STICK WITH THE THREE STAGE!

So in summary

$P_{DATA} = 333\,\mu W$

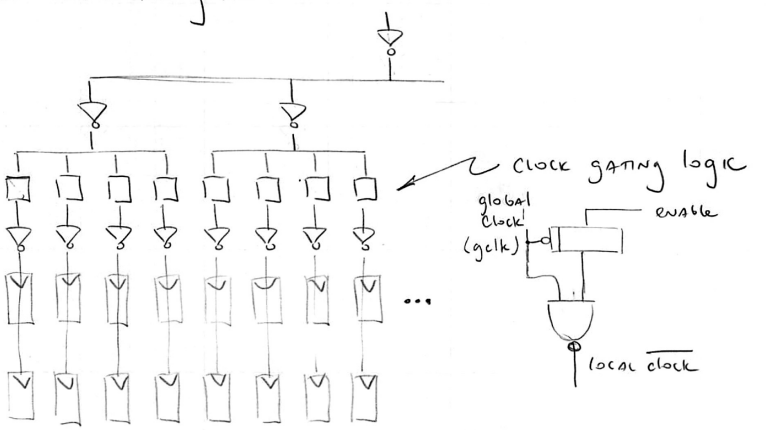
$P_{clock} = 7.6\,mW$

$P_{clockTree} = 173\,\mu W$

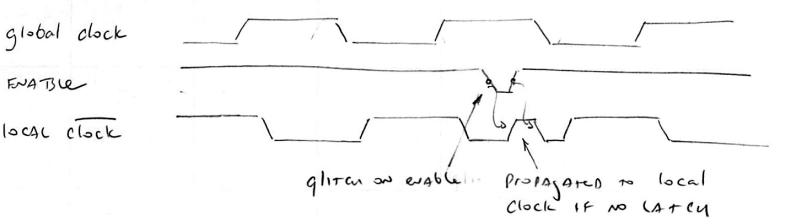
$P_{TOT} = 3.1\,mW$

So almost 84% of total dynamic power is spent in the local "INTEL-cell" clock logic!

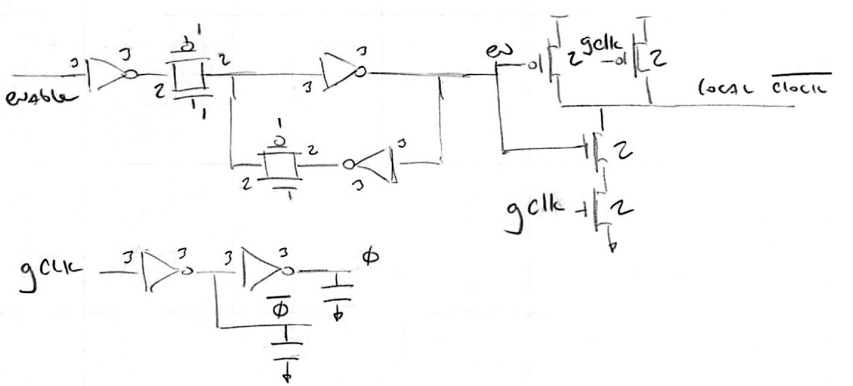
CAN CLOCK GATING HELP?



WHY DO WE NEED THE LATCH? AVOID GLITCHING ON ENABLE SIGNAL FROM PROPAGATING TO LOCAL CLOCK



- Clock gating
  - reduces activity on l2 of clock tree
  - reduces activity on local intra-cell clocking logic
  - increases switched cap when register is enabled
- Estimate extra switched cap due to gating cell on clock path when register is continuously enabled



- Estimate switched cap when continuously disabled
  - Extra 20C switched cap on global clock tree still toggles
  - Final clock inverter in clock tree does not toggle (576C)
  - Intra-cell clock logic does not toggle (512×20C = 10,240C)

- Let's put all of this together to estimate the impact of clock gating

	No Clock Gating	w/ Clock Gating (enabled)	w/ Clock Gating (disabled)
Data			
Intra Cell Clock			
L0,L1 Clk Tree			
L2 Clk Tree			
Gating Logic			
Total			

- In the worst case where we add clock gating but never actually gate any registers, we add a power overhead of:

- In the best case where we add clock gating and we gate the clock every cycle, we reduce the total power by: