EECS 151/251A
SP2022 Discussion 7

GSI: Yikuan Chen, Dima Nikiforov

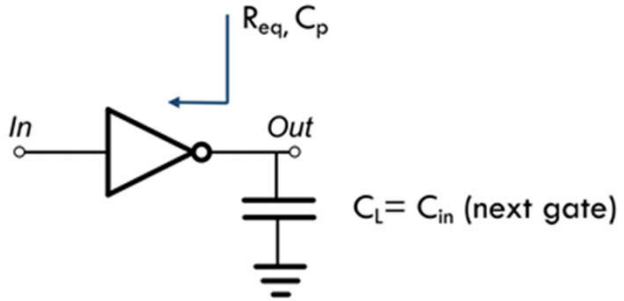
Slides modified from Zhenghan Lin's and Harrison Liew's slides

Agenda

- Inverter Delay – continue from last week's
- Logical Effort
- Elmore Delay

Loaded Inverter Delay

$$
\begin{aligned}
t_{p,inv} &= ln2 \cdot R_{eq}\left(C_{p,tot} + C_L\right) \\
&= ln2 \cdot R_{eq} C_{p,tot}\left(1 + \frac{C_L}{C_{p,tot}}\right) \\
&= ln2 \cdot R_{eq} C_{p,tot}\left(1 + \frac{C_L}{\gamma C_{in}}\right) \\
&= ln2 \cdot R_{eq} C_{p,tot}\left(1 + \frac{f}{\gamma}\right) \\
&= ln2 \cdot \tau_{inv}\left(1 + \frac{f}{\gamma}\right)
\end{aligned}
$$



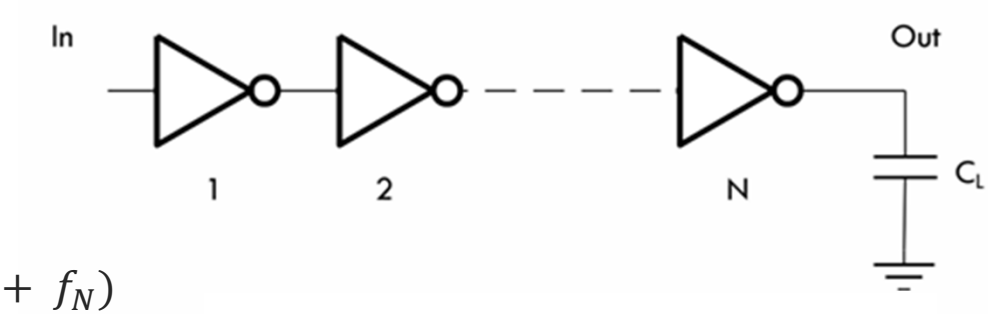
- Intrinsic vs. Extrinsic delay
  - Intrinsic τ_{inv} independent of sizing
- Fanout **f = C_L/C_{in}**
- Generalizable to any CMOS gate

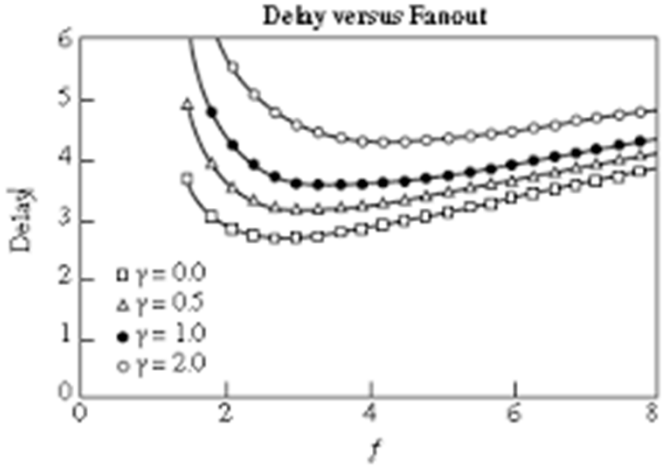
Inverter Chain Sizing

- Goal: minimize path delay
  - Assume 1st inverter has unit size ($C_{in,1} = 1$)
- Path delay:

$$D = t_{p1} + \ldots + t_{pN} = (1 + f_1) + \ldots + (1 + f_N)$$

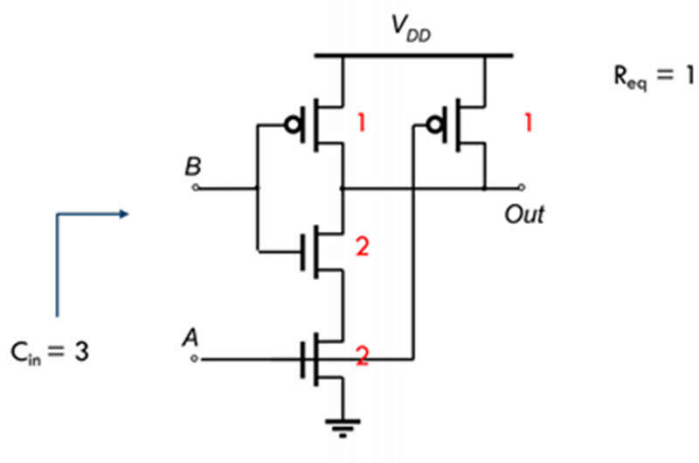
- Path fanout $F = \frac{C_L}{C_{in}}$
- Solution
  - Take partial derivatives w.r.t $C_2, \ldots, C_N$
  - Get $f_1 = f_2 = \ldots = f_N$
  - Min. path delay = $Nf + N = N\sqrt[N]{F} + N$
  - Size backwards





Logical Effort

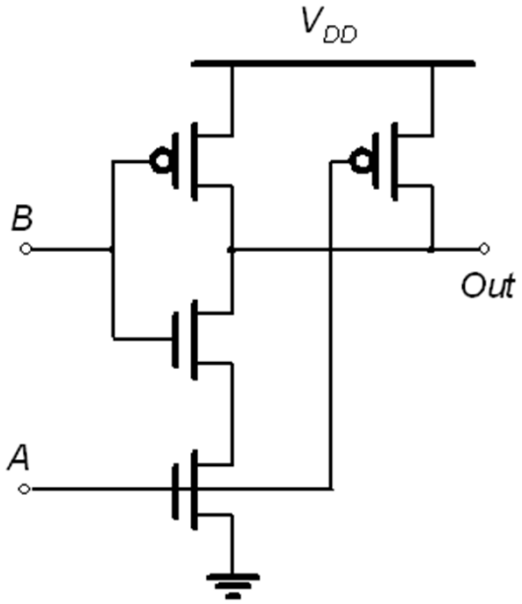
Logical Effort, Parasitic Delay

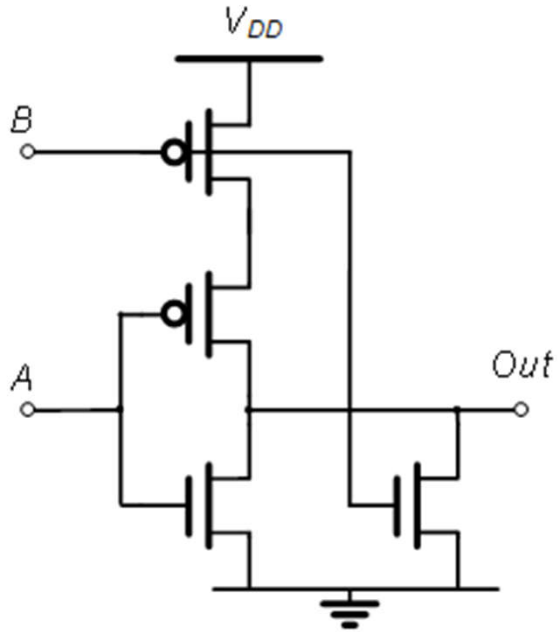


- For any gate, D(gate) = LE · FO + P
- Logical Effort
  - LE, Also defined as g in $t_{p,gate} = \tau_{inv}(p + \frac{gf}{\gamma})$
  - $LE \equiv \frac{R_{eq,gate}C_{in,gate}}{R_{eq,equiv}C_{in,inv}}$, i.e. ratio of $C_{in,gate}$ to $C_{in,inv}$ of a unit inverter *with the same output current*
  - Key to calculating LE: size the transistors to have the same R_{eq} as a unit inverter
- Parasitic Delay
  - Also defined as p in $t_{p,gate} = \tau_{inv}(p + \frac{gf}{\gamma})$
  - Only difference from LE is that p looks at the ratio of $C_{out,gate}$ to $C_{out,inv}$ (intrinsic capacitance)

PMOS/NMOS Resistance

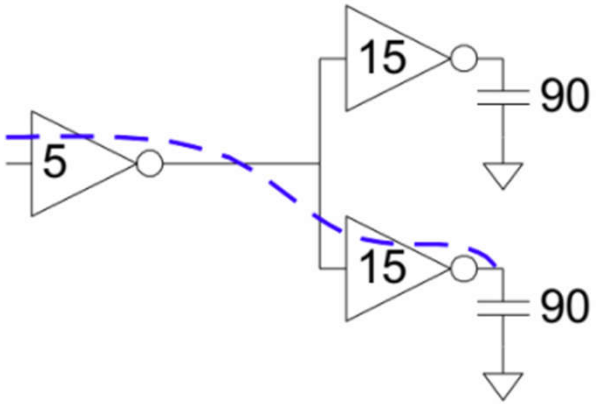
- In lecture, $LE_{NAND2} = LE_{NOR2} = \frac{3}{2}$. What if $R_{on,p} = K \cdot R_{on,n}$?





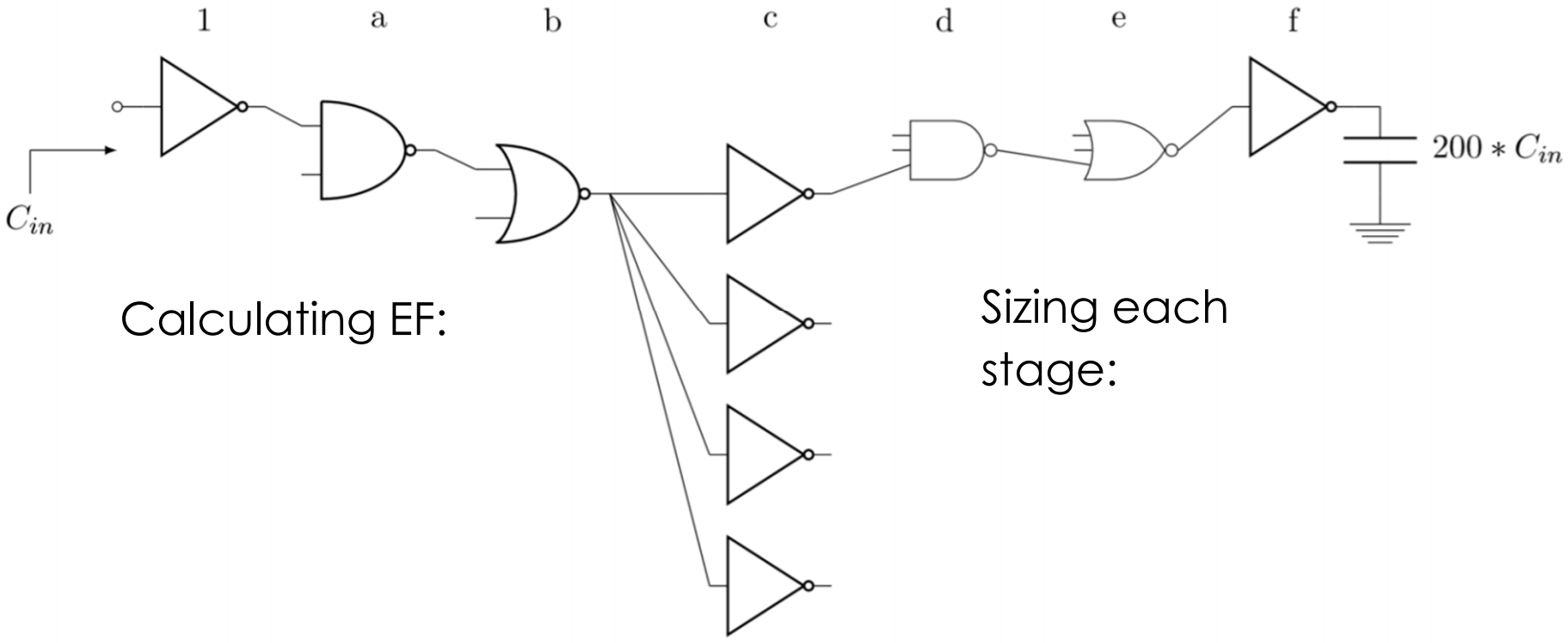
Minimizing Path Delay

- Path Logical Effort: $G = g_1 \cdot g_2 \cdot \ldots \cdot g_N$
- Path fanout $F = C_L / C_{in}$
- Branching, b
  - Extra loading factor $b = \frac{C_{onpath} + C_{offpath}}{C_{onpath}}$
  - Add b to total path effort $H = GFB$
- Solution is similar to single-branch inverter chain, except:
  - Calculate an "effective fanout" for each stage: $EF = \sqrt[N]{H}$
  - Size each stage such that $EF = g_i \cdot f_i$. You can control f_i by sizing!
  - Min. path delay = $N * \sqrt[N]{H} + \sum p_i$



Logic Path with Branch - Example

$$LE_{NAND2} = LE_{NOR2} = \frac{3}{2}$$
$$LE_{NAND3} = LE_{NOR3} = 2$$

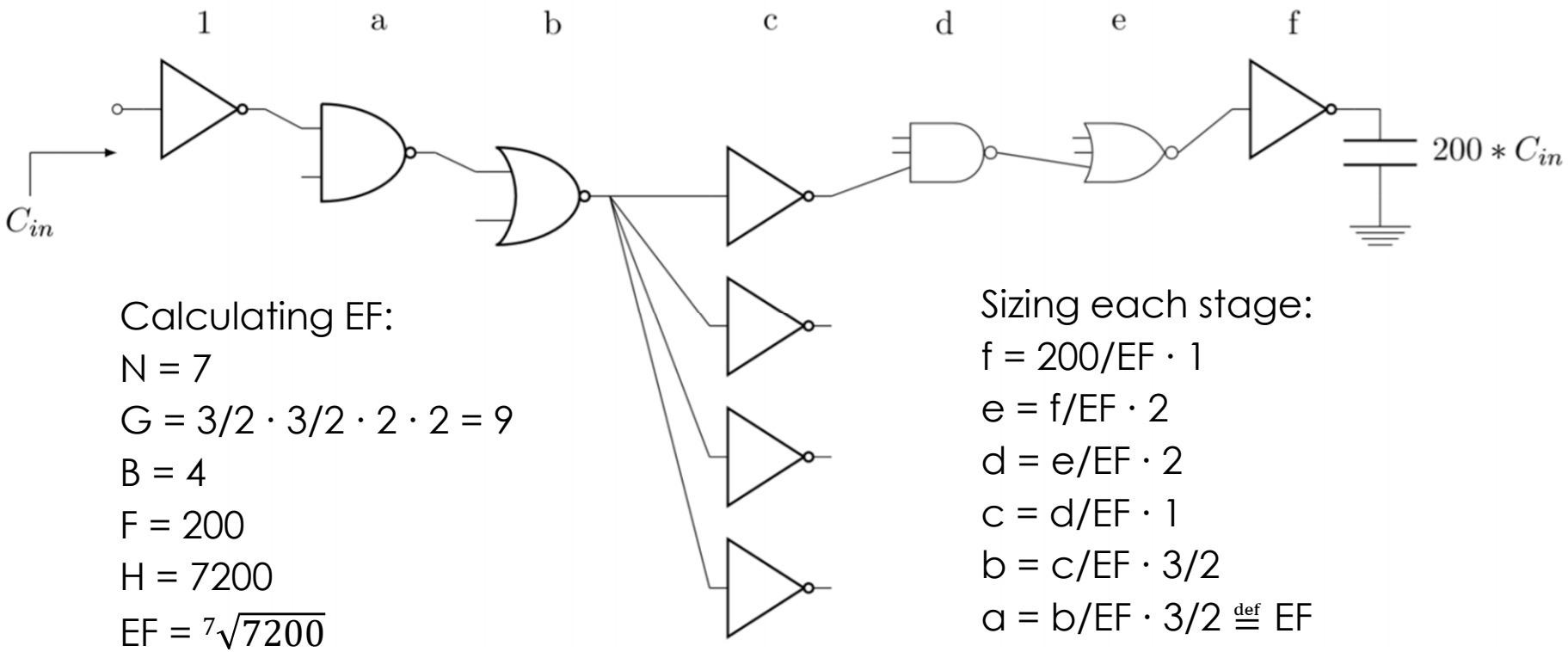


Calculating EF:

Sizing each stage:

Logic Path w/ Branch Example

$$LE_{NAND2} = LE_{NOR2} = \frac{3}{2}$$
$$LE_{NAND3} = LE_{NOR3} = 2$$



Calculating EF:

N = 7

G = 3/2 · 3/2 · 2 · 2 = 9

B = 4

F = 200

H = 7200

EF = $\sqrt[7]{7200}$

Sizing each stage:

f = 200/EF · 1

e = f/EF · 2

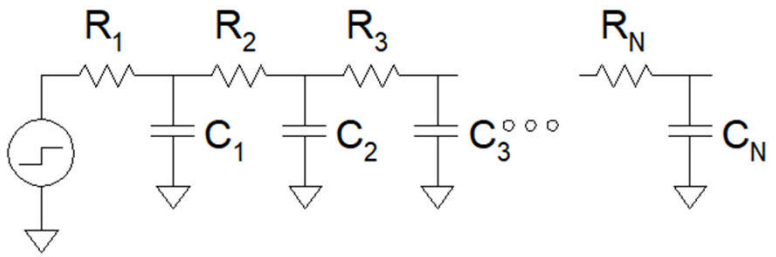
d = e/EF · 2

c = d/EF · 1

b = c/EF · 3/2

a = b/EF · 3/2 $\stackrel{\text{def}}{=}$ EF

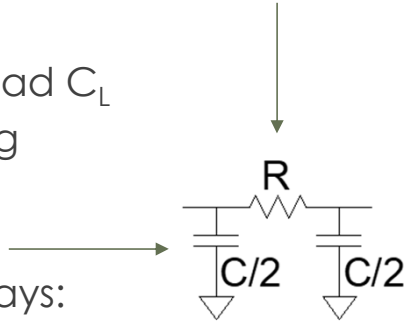
Elmore Delay – "All the R's for that C" or "All the C's by that R"



$$t_{pd} \approx \sum_{\text{nodes } i} R_{i-to-source} C_i$$

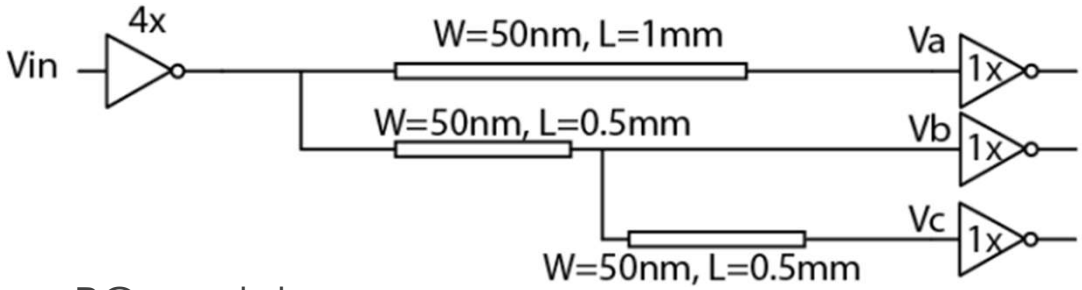
$$= R_1 C_1 + (R_1 + R_2) C_2 + \ldots + (R_1 + R_2 + \ldots + R_N) C_N$$

- Approximately the dominant RC time constant of a network
- R's can be:
  - Gate equivalent resistance R_{eq}
  - Wire resistance R = R□ · # squares
- C's can be:
  - Gate C_{in} and C_p, load C_L
  - Wire plate & fringing capacitance
    - $\pi - model$



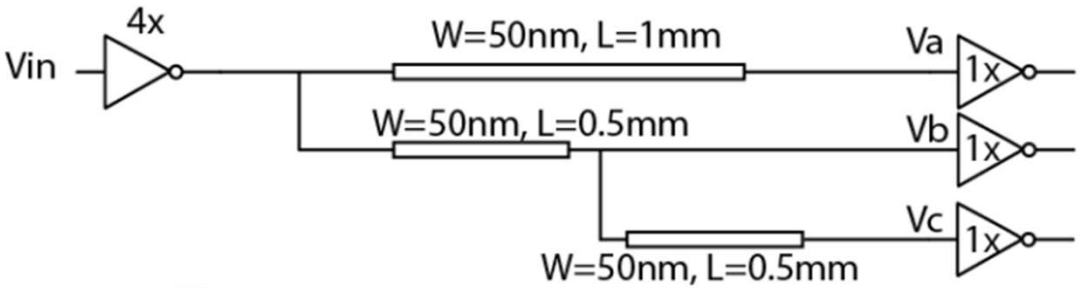
- Think of it 2 equivalent ways:
  - $\sum[C_i \cdot (sum\ of\ R's\ charging\ C_i)]$
  - $\sum[R_i \cdot (sum\ of\ C's\ that\ R_i\ charges)]$

Elmore Example



- What parameters do we need?
- Draw the equivalent circuit Elmore RC model
- Calculate delay to V_a

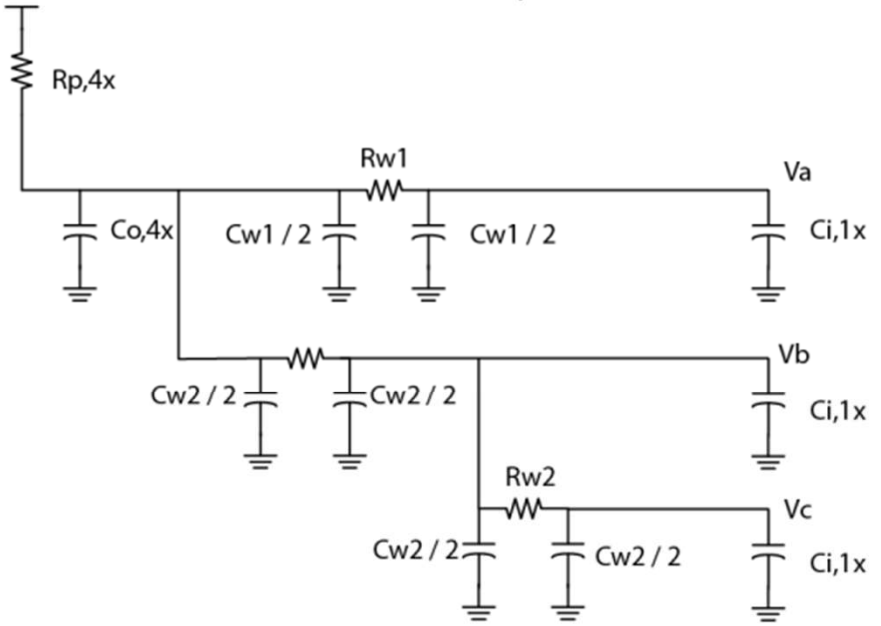
Elmore Example



- What parameters do we need?

$$C_{in}, C_p, R_{on,p}, R_{on,n}, R_{wire}, C_{pp}, C_{fr}$$

- Draw the equivalent circuit Elmore RC model



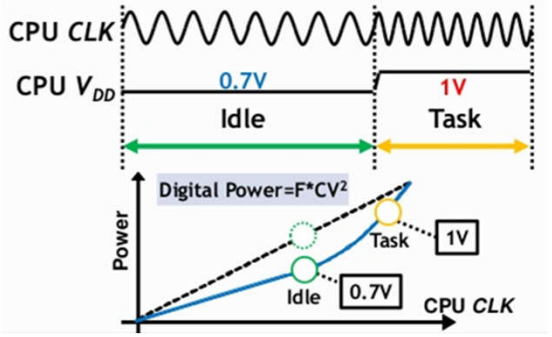
- Calculate delay to V_a

$$t_{p,a} = \ln 2 \cdot R_{w1} \cdot \left(C_{i,1x} + \frac{C_{w1}}{2} \right) +$$
$$\ln 2 \cdot R_{p,4x} \cdot \left(C_{i,1x} + \frac{C_{w1}}{2} + C_{i,1x} + \frac{C_{w2}}{2} + \frac{C_{w2}}{2} + \right.$$
$$\left. C_{i,1x} + \frac{C_{w2}}{2} + \frac{C_{w2}}{2} + \frac{C_{w1}}{2} + C_{o,4x} \right)$$

Power/Energy in Digital Circuits

- Fundamentally, charging/discharging capacitors (gate, parasitic, load) through resistances (PMOS, NMOS, wires)
  - Capacitors draw CV^2 joules from supply over 1 charge/discharge cycle
    - $\frac{1}{2}CV^2$ dissipated in PMOS as heat when charging
    - $\frac{1}{2}CV^2$ stored on capacitor, then dissipated in NMOS when discharging
- Dynamic power = $P_{switching} = \alpha CV^2 f$
  - How to minimize each term?
  - Minimizing which terms reduces total energy consumed?
- Static power = leakage → wasted energy!

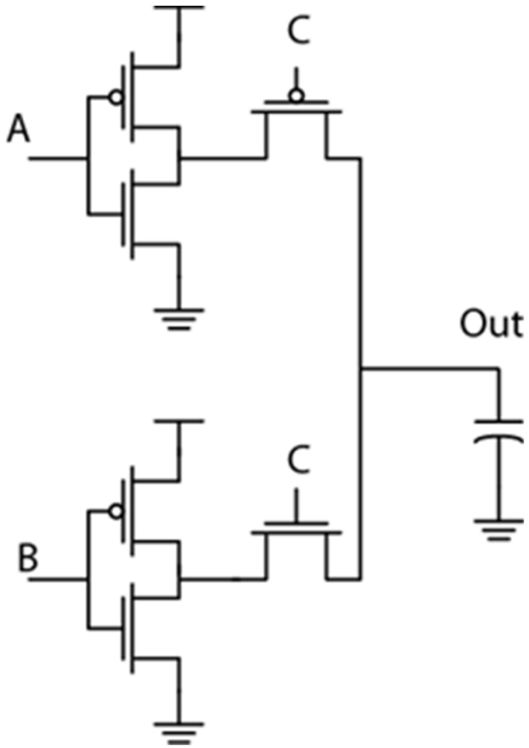
Dynamic Voltage and Frequency Scaling (DVFS)



Energy Example

- Initially: A = 1, C = 1, Out = 0
- Energy pulled from supply when B = 1 → 0?

- Then, how much energy dissipated when C = 1 → 0?



Energy Example

- Initially: A = 1, C = 1, Out = 0
- Energy pulled from supply when B = 1 → 0?

Out can only be charged to $V_{DD} - V_{th,n}$ because NMOS passes "bad 1"

$$E = C_L V_{DD} \left(V_{DD} - V_{th,n}\right)$$

- Then, how much energy dissipated when C = 1 → 0?

Out is discharged to $|V_{th,p}|$ because PMOS passes "bad 0"

$$E = \frac{1}{2} * C_L * \left[\left(V_{DD} - V_{th,n}\right)^2 - V_{th,p}^2\right]$$