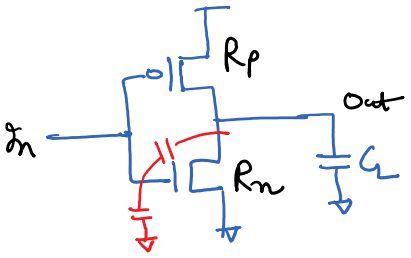
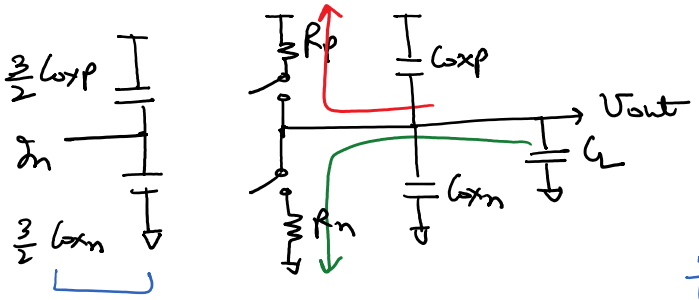
ECE 445 – Lecture 18

Monday, April 1, 2019 7:59 AM

$t_{pHL} = 0.7$

$t_{pLH} =$





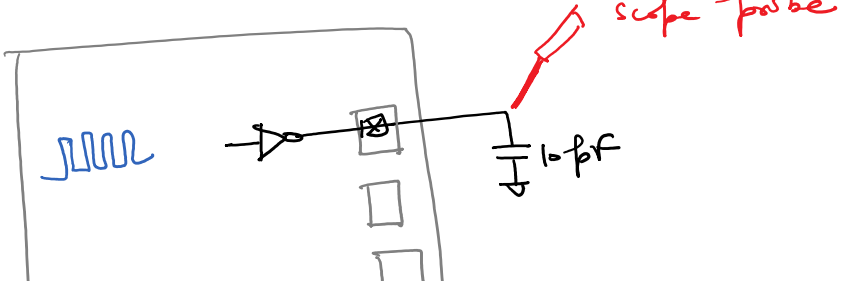
$\tau = R C_{in} = 0$

$t_{pLH} = 0.7 R_p (C_{oxn} + C_{oxp} + C_L)$

$t_{pHL} = 0.7 R_n (C_{oxn} + C_{oxp} + C_L)$

$f_{osc} = ?$

$P_d = C V_{DD}^2 f_{clk}$



10fF

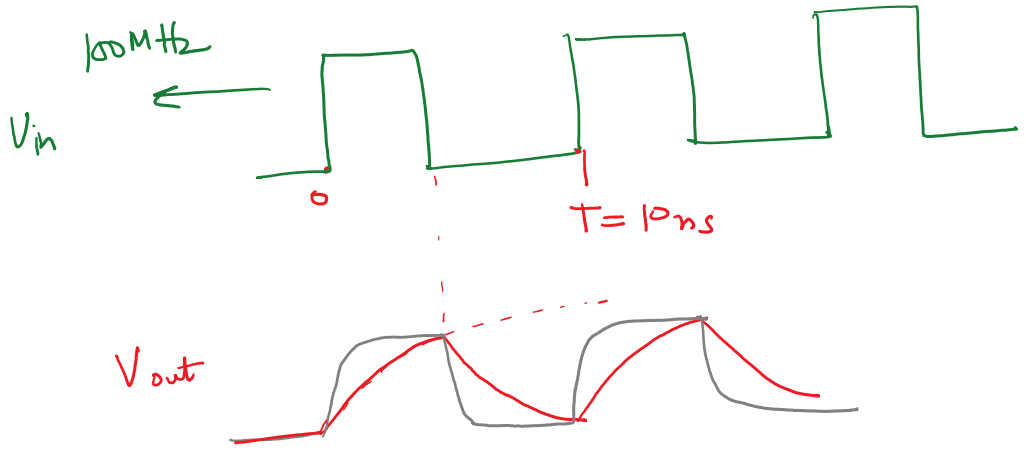
$t_{pLH} = 0.7 R_p (C_{outp} + C_{outn} + C_L)$

C_{outp}: 30fF, C_{outn}: 10fF

Let R_p = 1kΩ

$t_d \simeq 0.7 \times 1 \times 10^3 \times 10 \times 10^{-12}$

$= 0.7 \times 10^{-8} = 7 \times 10^{-9} = 7ns$



t_d = 1ns

$R_p = R_p' \cdot \frac{L}{W}$

W ↑ 10×

$t_d = 0.7 R_p [C_{outp} + C_{outn} + C_L]$

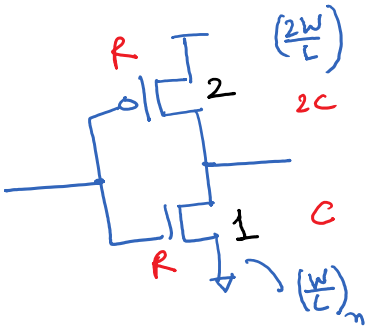
C_{outp}: 10×, C_{outn}: 10×

R_p: $\frac{1}{10\times}$

Logical Effort

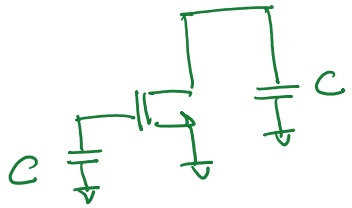
* Systematic method for "sizing" digital circuits.

* Simplified model for calculations, assume short-channel CMOS technology



MOSFET Capacitance Model:

$C \triangleq C_{ox}' WL \, (scale)^2$



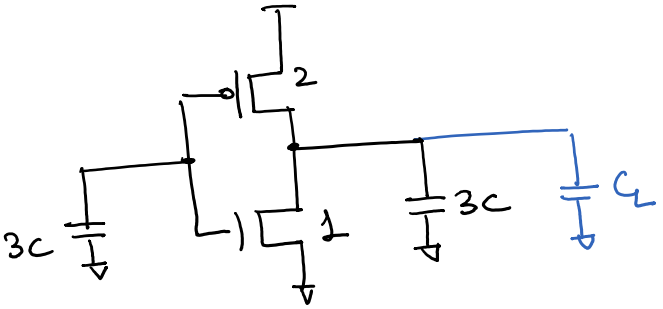
~~3~~

$\tau = R[C + 2C] = 3RC$ ← Unit inverter time constant
↳ delay $0.7(3RC)$

* Express time delays in a process independent unit

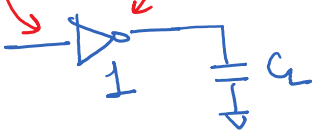
Normalized delay, $d = \frac{d_{abs}}{0.7\tau} = \frac{0.7 \times 3RC}{0.7 \times 3RC} = 1$

baseline for an inverter



$C_{out} = 3c + C_L$

$C_{in} = 3c$



$$d = f + p$$

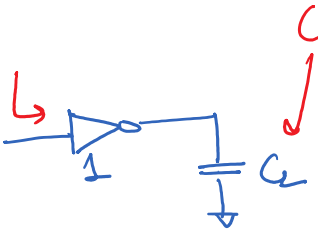
f: effort delay

p: parasitic delay

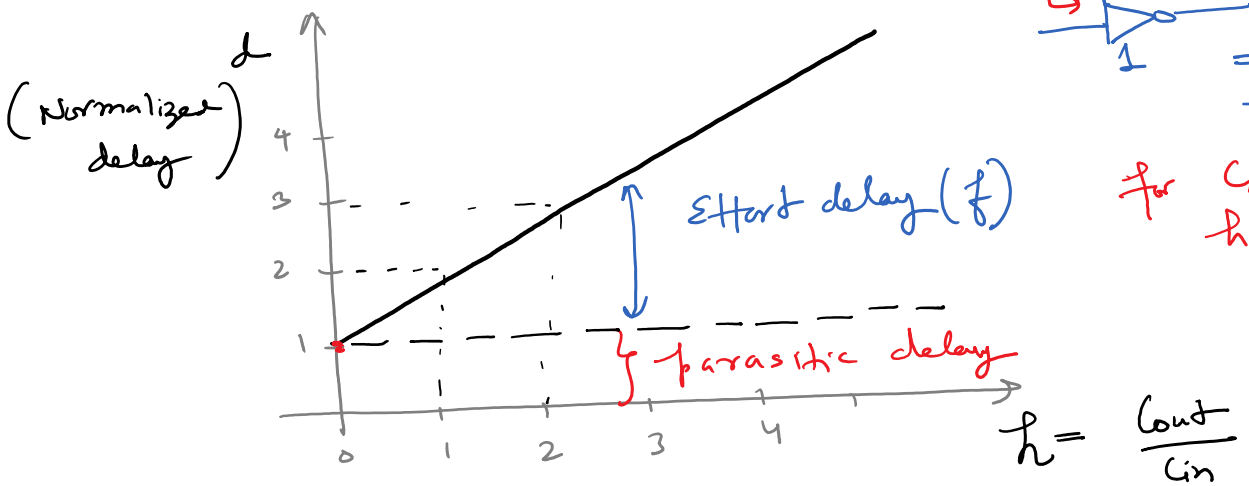
$$= g \times h + p$$

$g \times h$: electrical effort (how big of a load is being driven)

g: logical effort of the gate (=1 for an inverter)



for $C_L = 0$, $h = 0$

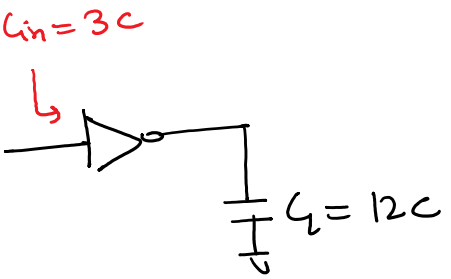


⇒ electrical effort

parasitic delay ⇒ inverter driving its own output parasitic capacitance (3C)

effort delay ⇒ delay due to the inverter driving $C_{out} = C_L$

Ex.



for an inverter
$p = 1$
$g = 1$

$$d = g \cdot h + p$$

$$= 1 \times \left(\frac{12C}{3C}\right) + 1$$

$$= 4 + 1 = 5$$

$C_L = 100C$

$$d = g \times h + p$$

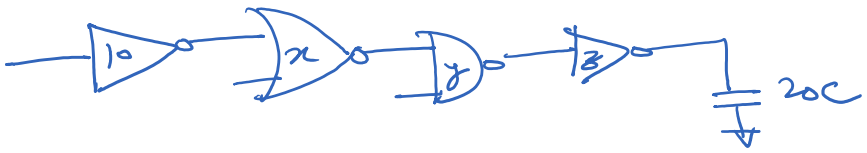
$$= 1 \times \left(\frac{100C}{3C}\right) + 1 = 34$$

Monday, April 1, 2019 8:36 AM

Logical Effort on paths:

* A multistage logic network

* How do we size each of the stages?



Delay is minimized

Single logic stage

Logical effort (g)

Electrical effort ($h = \frac{C_{out}}{C_{in}}$)

↳ Effort delay $\Rightarrow f = gh$

Parasitic delay $\Rightarrow p$

Logic path

Path effort $\Rightarrow G = \prod_i g_i$

Path electrical effort $\Rightarrow H = \frac{C_{out,path}}{C_{in,path}}$

Path effort $\Rightarrow F = \prod_i f_i = \prod_i g_i h_i$

$= G \cdot H$

$P = \sum_i p_i$

$G = g_1 g_2 g_3 g_4$

$H = \frac{C_L}{C_{in_1}}$

$C_{in,1}$

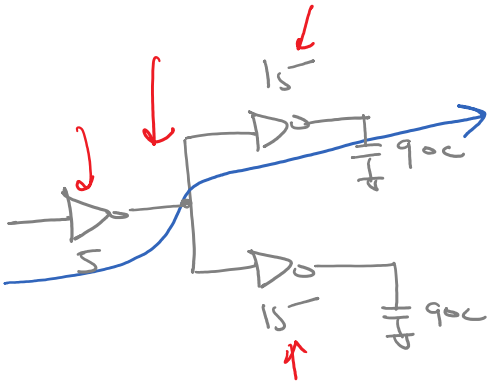
$f = gh$

p

$F = G \cdot H$

P

Branching paths:



Need to account for branching between stages

branching effort, $b = \dfrac{C_{on,path} + C_{off,path}}{C_{on,path}}$

$\Rightarrow b = \dfrac{15+15}{15} = 2$

Path branching effort $\Rightarrow B = \prod_i b_i$

Path effort

$F = G \cdot B \cdot H$ ①

$\dfrac{C_{out,path}}{C_{in,path}}$

$\prod_i b_i$

$\prod_i g_i$

Delay of the multistage network:

path delay: $D = \sum_i d_i$

$= D_F + P$

P → parasitic delay

D_F → path effort delay (due to F)

$D_F = \sum f_i$ ⇒ Sum of the individual effort delays.

$P = \sum p_i$ ⇒ sum of the individual parasitic delays.

objective:

Find f_i's such that $D = \sum f_i + \sum p_i$ is minimized

optimization ⇒ find f_i's to minimize D

Theorem: If a path has N stages with path effort F and path parasitic delay P, then the path delay is minimized when each stage

bears the same effort and must be

$$\Rightarrow \boxed{\hat{f} = F^{1/N}} \quad (2)$$

$\Rightarrow$ The minimum delay for the N-stage path is

$$D = N \cdot \hat{f} + P$$

$$\Rightarrow \boxed{D = N F^{1/N} + P} \quad (3)$$

↳ key result of logical effort

We note that

$F = GBH$ is independent of gate sizes and depend only on the path

G, B, $H = \frac{C_{out,path}}{C_{in,path}}$

$$\hat{f} = F^{1/N}$$

$$\hat{f} = g_i h_i$$

we know $g_i \Rightarrow$ type of the gate

$$\Rightarrow h_i = \frac{\hat{f}}{g_i}$$

$$C_{out,i} \times g_i \quad (4)$$

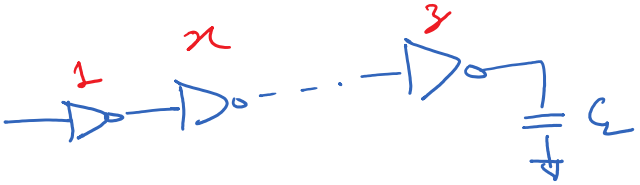
$$\therefore h_i = \frac{C_{out,i}}{C_{in,i}} \Rightarrow \boxed{C_{in,i} = \frac{C_{out,i} \times g_i}{f}}$$

we can back calculate C_{in} and thus the sizing of each of the stages.

Monday, April 1, 2019 8:59 AM

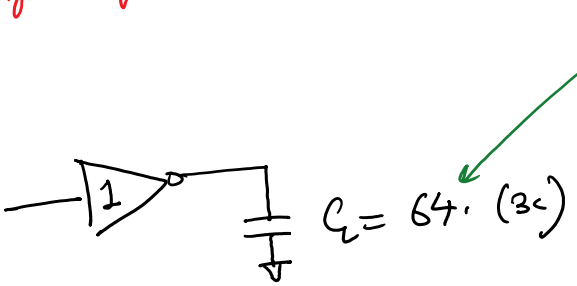
Choosing the best number of stages (N=?)

* Design an inverter buffer to drive external load.



min delay

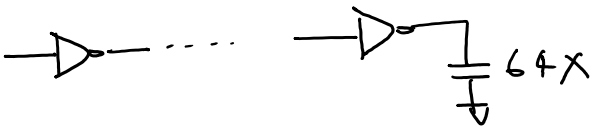
Buffer Design problem:



N=1

$f = gh = 1 \times 64$

$D = f + p = 64 + 1 = \underline{\underline{65}}$



$D = N F^{1/N} + P$ ⟶ $P = \Sigma p_{inv} = N p_{inv}$

$$\Rightarrow D = N \cdot F^{1/N} + N \cdot p_{inv}$$

$$\frac{\partial D}{\partial N} = F^{1/N} - N \cdot \frac{1}{N} F^{1/N} \ln(F^{1/N}) + p_{inv} = 0$$

$$\Rightarrow F^{1/N}\left(1 - \ln(F^{1/N})\right) + p_{inv} = 0$$

$$\Rightarrow p_{inv} + \rho(1 - \ln\rho) = 0$$

where $\rho = F^{1/\hat{N}}$

$\hat{N}$ → best number of stages.

Solving numerically

for $p_{inv} = 2$

$$\rho = 3.59 \Rightarrow \hat{f} = F^{1/\hat{N}}$$

⇒ A path achieves best delay by using

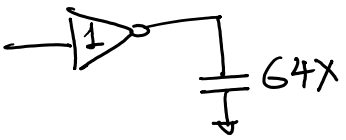
$$\hat{N} = \log_\rho(F) \text{ stages.}$$

* Using $\rho = \hat{f} = 4$ is a convenient choice and simplifies mentally choosing the best number of stages. (< 2% error from minimum delay)

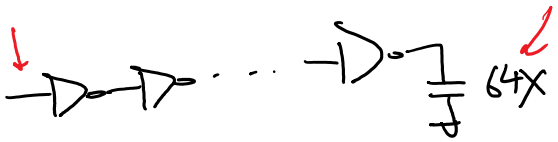
$$\hat{N} = \log_4(F)$$

fan-out or stage effort of 4 gives overall the best delay.

$\Rightarrow$ fan-out of 4 (Fo4) delay.



D = 6τ



$F = GBH = 1 \cdot 1 \cdot 64 = 64$

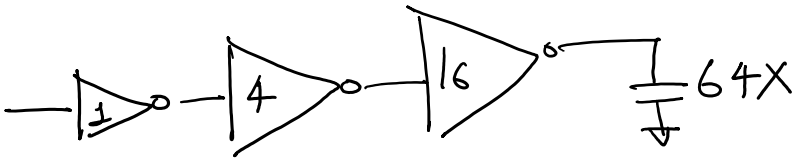
$\hat{N} = \log_4 F = \log_4(64) = \log_4(4 \times 4 \times 4)$

$= 3$ Stages

$\hat{f} = F^{1/\hat{N}} = \sqrt[3]{64} = \underline{4}$ ← by design fo4

$D = N \cdot f + P$

$= 3 \times 4 + 3 = \underline{15}$ fastest delay.


1
4
16
64X