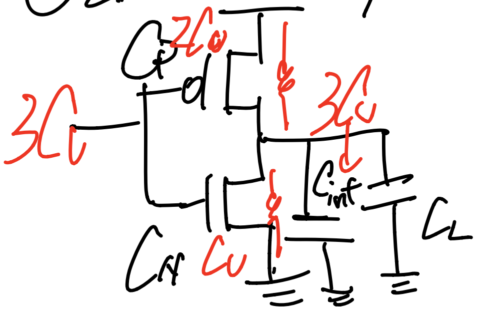
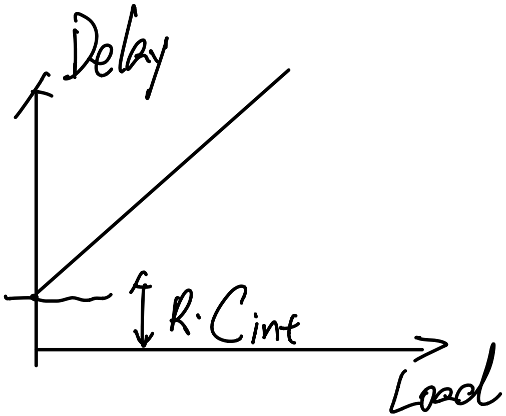
L02:

(1) Linear Delay Model





Tend to use inverter as standard Linear delay model. P/N ratio of unit inverter is 2:1. due to the speed of PMOS is faster than NMOS. and in CMOS we want the speed of PMOS & NMOS equal

C_{int}: Intrinsic Capacitance of the Model, from PMOS & NMOS

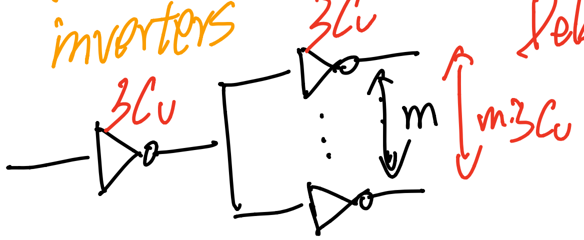
C_L: Load capacitance from wire and gate capacitance of next logic circuitry.

$C_P : 2 \cdot C_{unit}$ $C_N : 1 \cdot C_{unit}$

$$\begin{aligned} Delay &= K R_u (C_{int} + C_L) = K R_u C_{int} + K R_u C_L \\ &= K R_u C_{int} (1 + C_L / C_{int}) \\ &= Delay(Internal) + Delay(Load) \end{aligned}$$

Some divations of Linear Delay Model

- A unit inverter driving m identical unit inverters

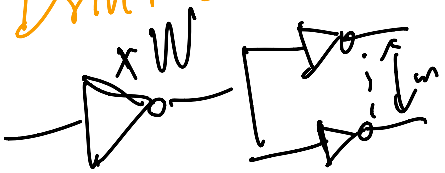


$$\text{Delay} \simeq \boxed{R \cdot 3C_u} + \boxed{m \cdot R \cdot 3C_u}$$

C_{int} C_L

$$\simeq 3RC_u + 3mRC_u$$

- Driver sized W times unit size



*Recall two important equations:

$$R = \rho \cdot \frac{L}{W} \qquad C = \epsilon \cdot \frac{A}{d}$$

Size up driver means we size up W (width), because L is fixed by fab tech

size up W times. $R\downarrow = \frac{R}{W}$ $C\uparrow = W \cdot C$

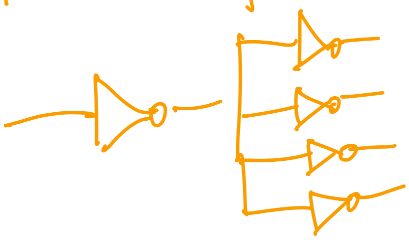
So, size up W times

$$\text{Delay} \simeq 3C_u \cdot \cancel{W} \cdot \frac{R}{\cancel{W}} + \frac{R}{W} \cdot 3C_u \cdot m$$

$$\simeq 3C_u \cdot R + \frac{m \cdot R \cdot 3C_u}{W}$$

So. sizing up driver actually doesn't effect intrinsic delay, it minimized load delay

· Fanout-of-4 (FO4)



FO4 is standard data, will be provided by the fab vendor to give you a taste of their current process standard speed.

$\text{Delay} \approx R \cdot 3C_i + 4 \cdot R \cdot 3C_i$

$\approx 5 \cdot \boxed{R \cdot 3C_i} \longrightarrow$ We refer $R \cdot 3C_i$ as τ, a timing constant, as a delay unit to simplify delay analysis.

② Delay in a logic gate.

Delay has two components:

$$d = f + p.$$

$$f = g \cdot h$$

f: effort delay (a.k.a. stage effort)

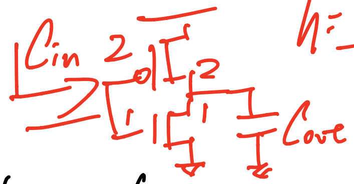
g: logical effor

- Measures relative ability of gate to deliver current
- $g = 1$ for inverter

h: electrical effort $= \frac{C_{out}}{C_{in}}$

- Ratio of output to input capacitance
- Sometimes h refer as "fan out"

E.x.



$h = \frac{2+1}{2+1} = 1$ (Unit inverter)

p: parasitic delay

- Represents delay of gate driving no load
- Set by internal parasitic capacitance

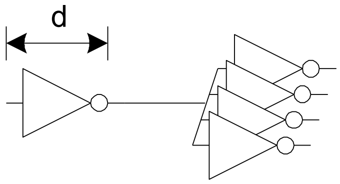
Catalog of Gates

- Parasitic delay of common gates
  - In multiples of p_{inv} (≈ 1)

Gate type	Number of inputs				
	1	2	3	4	n
Inverter	1				
NAND		2	$9/3 = 3$		n
NOR		2			n
Tristate / mux	2	4			$2n$

Example: FO4 Inverter

- Estimate the delay of a fanout-of-4 (FO4) inverter



Logical Effort: $g = 1$

Electrical Effort: $h = C_{out}/C_{in} = \frac{4 \cdot C_{in}}{C_{in}} = 4$

Parasitic Delay: $p = 1$

Stage Delay: $d = g \cdot h + p = 4 \cdot 1 + 1 = 5$

$\because FO4 = 15RC. \; d_{abs} = d \cdot \tau = 3RC \cdot d = 15RC.$

Multistage Delays

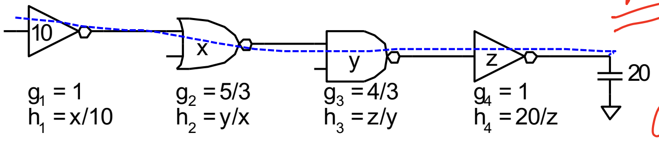
- Path Effort Delay : the absolute delay. $D_F = \sum f_i$
- Path Parasitic Delay $P = \sum p_i$
- Path Delay $D = \sum d_i = D_F + P$

12

Multistage Logic Networks

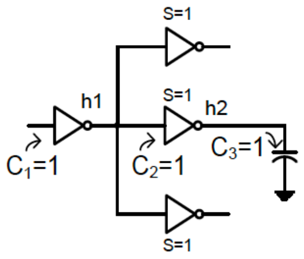
- Logical effort generalizes to multistage networks
- *Path Logical Effort* $G = \prod g_i$
- *Path Electrical Effort* $H = \frac{C_{\text{out-path}}}{C_{\text{in-path}}}$
- *Path Effort* $F = \prod f_i = \prod g_i h_i \underline{= G \cdot H}$ $\rightarrow D_F = \sum f_i$ will reach minimum, when f_i in each stage is equal. e.g. $f_1 = f_2 = \cdots = f_n$

due to branching



13

Branching



Without branching: $h_1 h_2 = H$

With branching: $h_2 = \frac{C_3}{C_2}$

$h_1 = \frac{3C_2}{C_1}$

$h_1 h_2 = 3H$

Need to account for branching!

$$b_i = \frac{C_{on-path} + C_{off-path}}{C_{on-path}} \Rightarrow$$

$C_{on-path,2} = C_2$

$C_{off-path,2} = 2C_2$

$b_2 = \frac{3C_2}{C_2} = 3$

14

Equivalent Path Efforts

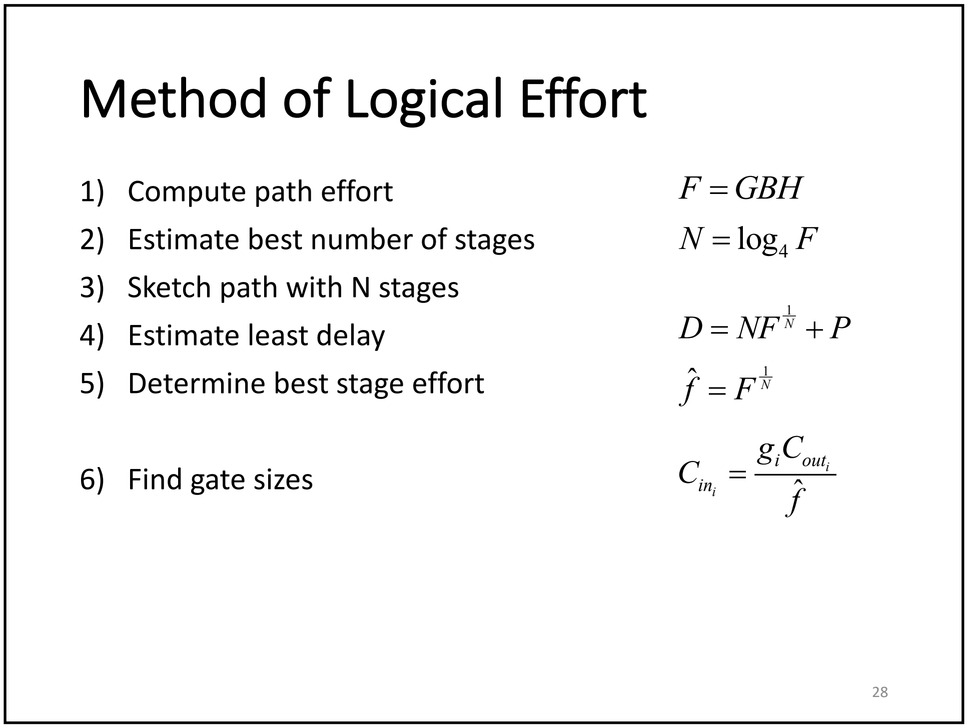
$$H = \frac{C_{out}}{C_{in}}$$
$$B = \prod b_i$$
$$BH = \prod h_i$$
$$G = \prod g_i$$
$$P = \sum p_i$$

Path Effort

$$F = GBH = \prod g_i h_i$$

- Path Effort
  - Does not change with added inverters
  - Does not depend on sizes, but on topology

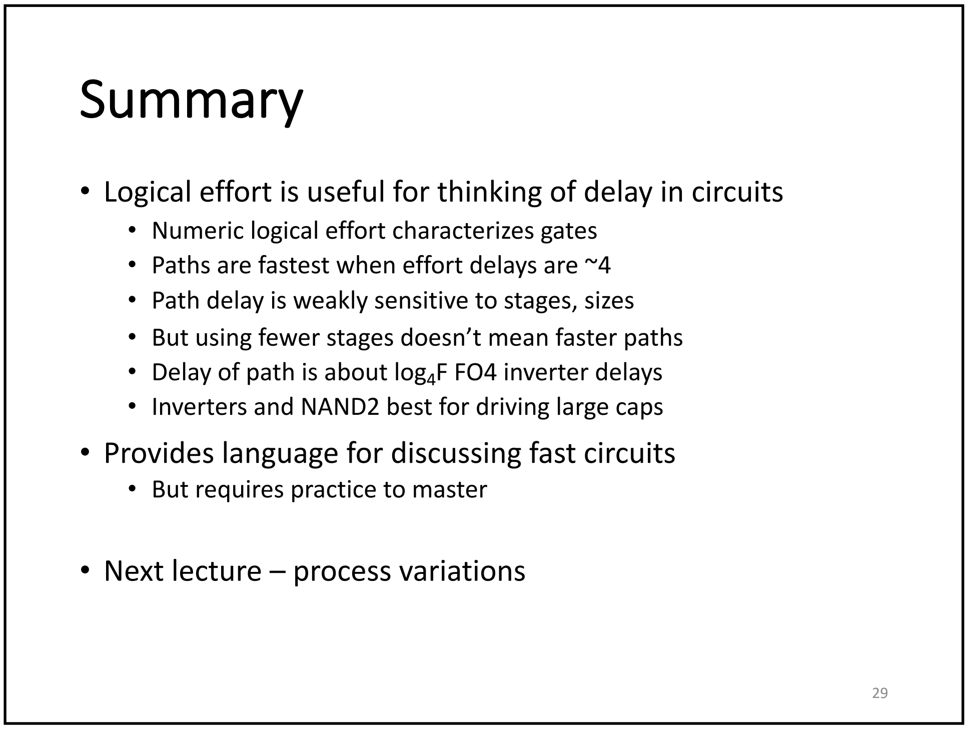
15



Method of Logical Effort

1) Compute path effort $F = GBH$
2) Estimate best number of stages $N = \log_4 F$
3) Sketch path with N stages
4) Estimate least delay $D = NF^{\frac{1}{N}} + P$
5) Determine best stage effort $\hat{f} = F^{\frac{1}{N}}$
6) Find gate sizes $C_{in_i} = \frac{g_i C_{out_i}}{\hat{f}}$

28



Summary

- Logical effort is useful for thinking of delay in circuits
  - Numeric logical effort characterizes gates
  - Paths are fastest when effort delays are ~4
  - Path delay is weakly sensitive to stages, sizes
  - But using fewer stages doesn't mean faster paths
  - Delay of path is about $\log_4$F FO4 inverter delays
  - Inverters and NAND2 best for driving large caps
- Provides language for discussing fast circuits
  - But requires practice to master
- Next lecture – process variations

29