

Lecture 9: Logical Effort

Assignment for 9/26

- Complete lab 4
- Text book reading sections 4.4 Through 4.5

HW # 1 due 10/1

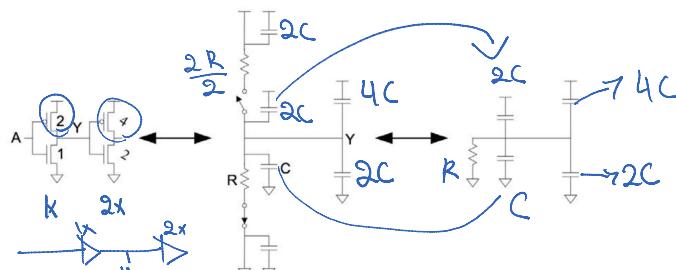
Quiz 4

Tuesday, September 24, 2019 11:04 AM

Quiz 4 Sep. 24, 2019 CPEG460/660

Name: _____

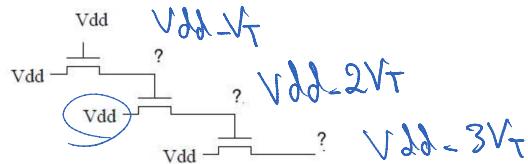
Q1: Label resistances and capacitances below. 1X NMOS transistor has resistance of R and capacitance of C. 1X PMOS transistor has resistance of 2R and capacitance of C.



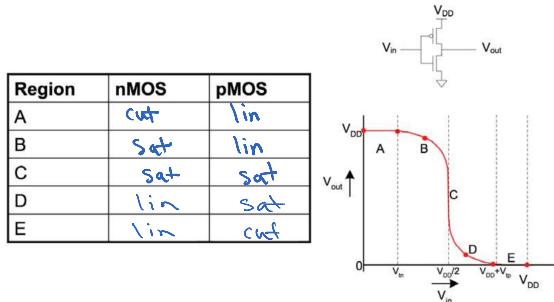
$$A = \frac{P_{df} + P_{dr}}{2}$$

$$R(2C + C + 4C + 2C) = 9RC$$

Q2: Find the output voltages. Assume NMOS threshold voltage is VT.



Q3: Label the region of operation for NMOS and PMOS transistors.



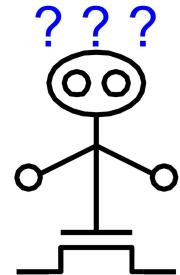
Outline

- Logical Effort
- Delay in a Logic Gate
- Multistage Logic Networks
- Choosing the Best Number of Stages
- Example
- Summary

Introduction

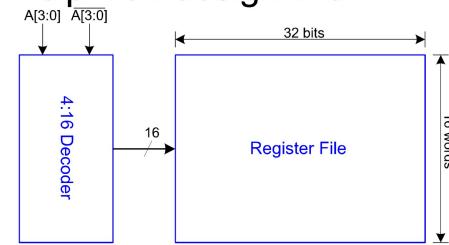
- Chip designers face a bewildering array of choices
 - What is the best circuit topology for a function?
 - How many stages of logic give least delay?
 - How wide should the transistors be?

- Logical effort is a method to make these decisions
 - Uses a simple model of delay
 - Allows back-of-the-envelope calculations
 - Helps make rapid comparisons between alternatives
 - Emphasizes remarkable symmetries



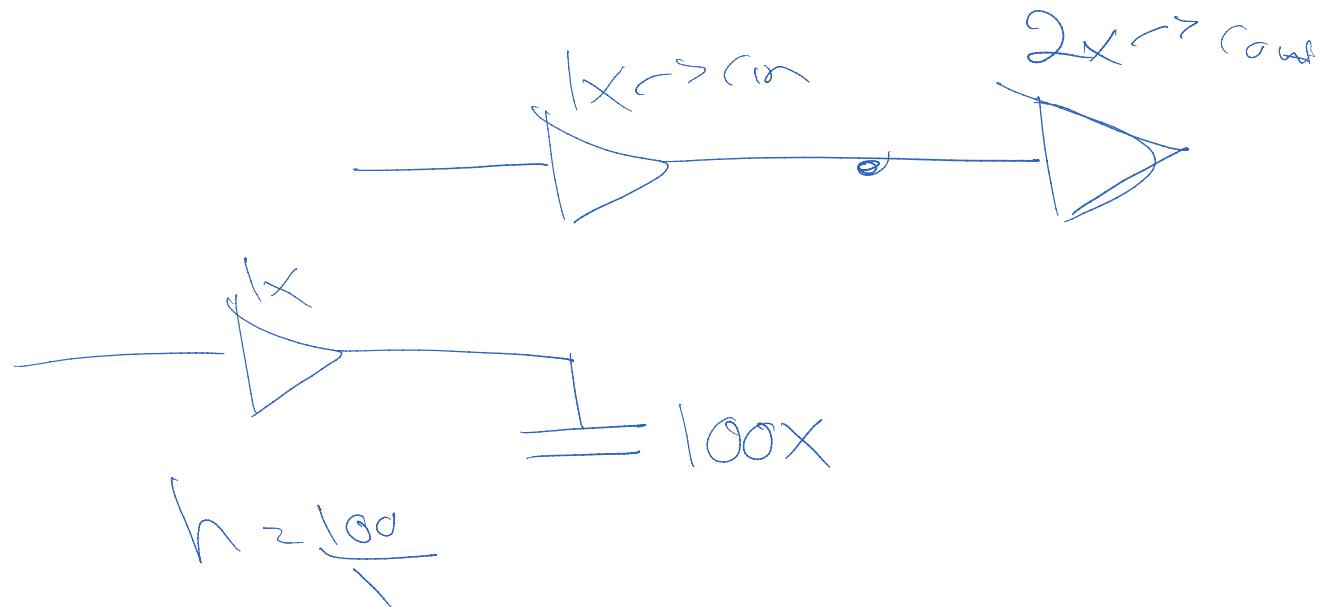
Example

- Ben Bitdiddle is the memory designer for the Motoroil 68W86, an embedded automotive processor. Help Ben design the decoder for a register file.
- Decoder specifications:
 - 16 word register file
 - Each word is 32 bits wide
 - Each bit presents load of 3 unit-sized transistors
 - True and complementary address inputs A[3:0]
 - Each input may drive 10 unit-sized transistors
- Ben needs to decide:
 - How many stages to use?
 - How large should each gate be?
 - How fast can decoder operate?



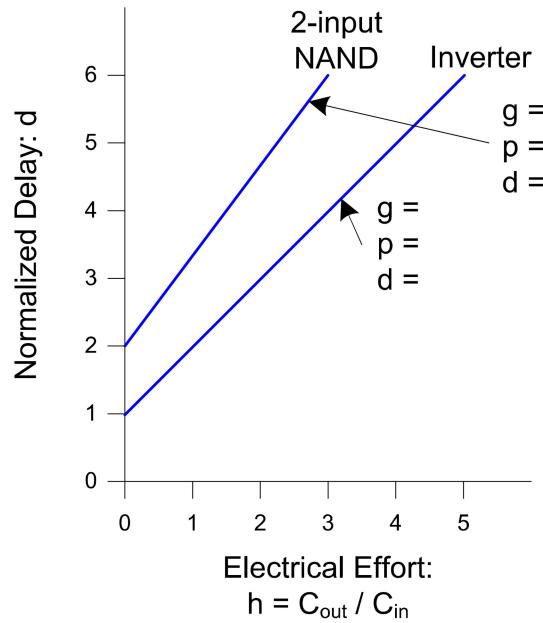
Delay in a Logic Gate

- Express delays in process-independent unit $d = \frac{d_{abs}}{\tau}$
- Delay has two components: $d = f + p$ $\tau = 3RC$
 - f : effort delay = gh (a.k.a. stage effort)
 - Again has two components
 - g : logical effort
 - Measures relative ability of gate to deliver current
 - $g = 1$ for inverter
 - h : electrical effort = C_{out} / C_{in}
 - Ratio of output to input capacitance
 - Sometimes called fanout
 - p : parasitic delay
 - Represents delay of gate driving no load
 - Set by internal parasitic capacitance



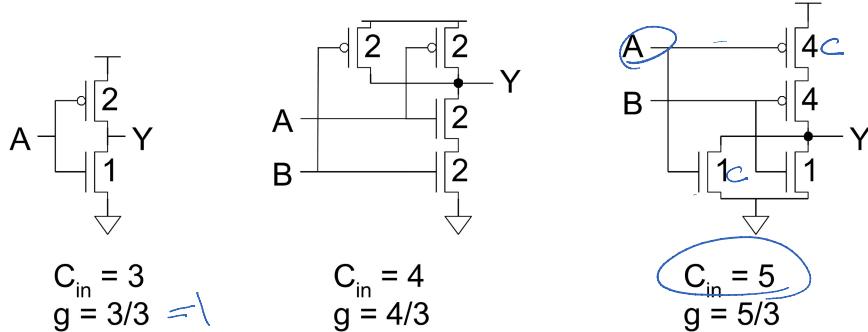
Delay Plots

$$\begin{aligned}d &= f + p \\&= gh + p\end{aligned}$$



Computing Logical Effort

- DEF: *Logical effort is the ratio of the input capacitance of a gate to the input capacitance of an inverter delivering the same output current.*
- Measure from delay vs. fanout plots
- Or estimate by counting transistor widths



Catalog of Gates

□ Logical effort of common gates

g

| Gate type | Number of inputs | | | | |
|----------------|------------------|------|----------|--------------|----------|
| | 1 | 2 | 3 | 4 | n |
| Inverter | 1 | | | | |
| NAND | | 4/3 | 5/3 | 6/3 | (n+2)/3 |
| NOR | | 5/3 | 7/3 | 9/3 | (2n+1)/3 |
| Tristate / mux | 2 | 2 | 2 | 2 | 2 |
| XOR, XNOR | | 4, 4 | 6, 12, 6 | 8, 16, 16, 8 | |

Catalog of Gates

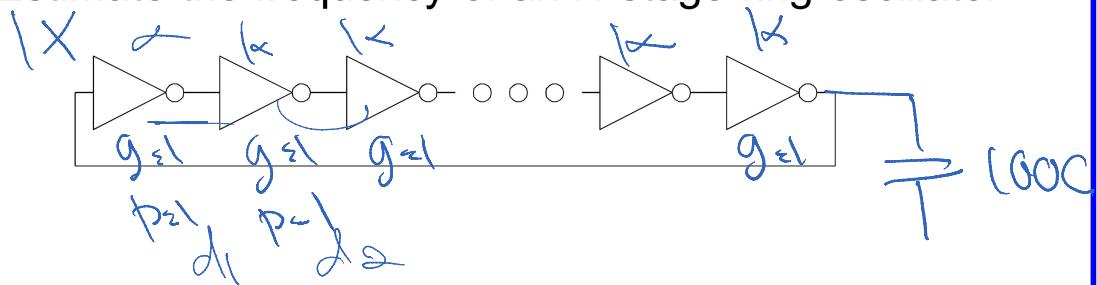
- Parasitic delay of common gates
 - In multiples of p_{inv} (≈ 1)

($3Rc$)

| Gate type | Number of inputs | | | | |
|----------------|------------------|---|---|---|----|
| | 1 | 2 | 3 | 4 | n |
| Inverter | 1 | | | | |
| NAND | | 2 | 3 | 4 | n |
| NOR | | 2 | 3 | 4 | n |
| Tristate / mux | 2 | 4 | 6 | 8 | 2n |
| XOR, XNOR | | 4 | 6 | 8 | |

Example: Ring Oscillator

- Estimate the frequency of an N-stage ring oscillator



Logical Effort: $g =$

Electrical Effort: $h =$

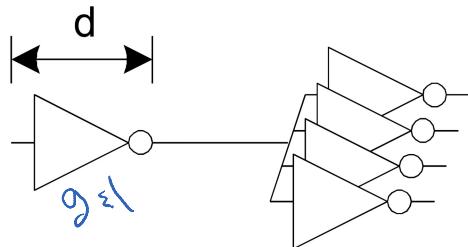
Parasitic Delay: $p =$

Stage Delay: $d = \frac{goh + P}{2} = S + P$

Frequency: $f_{osc} = \frac{1}{d}$

Example: FO4 Inverter

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



Logical Effort: $g =$

Electrical Effort: $h = \frac{C_{out}}{C_N}$

Parasitic Delay: $p =$

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

Multistage Logic Networks

- Logical effort generalizes to multistage networks

- Path Logical Effort

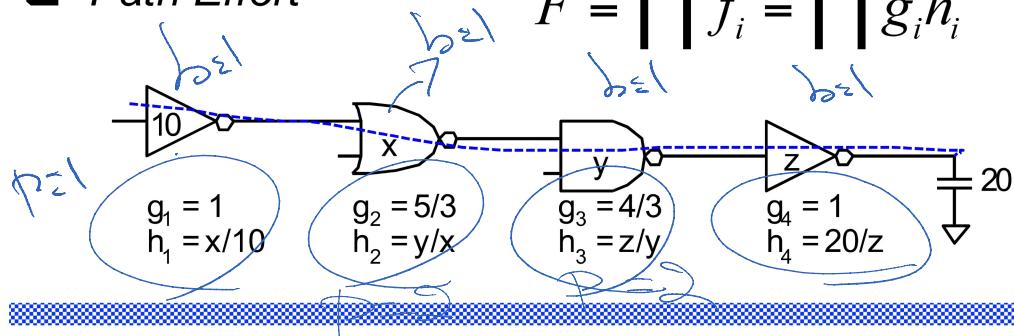
$$G = \prod g_i \rightarrow$$

- Path Electrical Effort

$$H = \frac{C_{\text{out-path}}}{C_{\text{in-path}}}$$

- Path Effort

$$F = \prod f_i = \prod g_i h_i$$



8: Logical Effort

CMOS VLSI Design 4th Ed.

13

$$H = \frac{20}{10} = 2$$

$$G = \prod g_i = g_1 \cdot g_2 \cdot g_3 \cdot g_4 = 1 \cdot \frac{5}{3} \cdot \frac{4}{3} \cdot 1 = \frac{20}{9}$$

$$P = \sum_{i=1}^4 1 + 2 + 2 + 1 = 6$$

$$F = G \cdot H \cdot B = \frac{20}{9} \cdot 2 \cdot 1 = \frac{40}{9}$$

$$\hat{F} = F^{1/N} = \frac{40}{9}^{1/4}$$

$$D = N \hat{F} + P$$

$$d_1 = 1 \cdot \frac{x}{10} + 1$$

$$d_2 = \frac{5}{3} \cdot \frac{y}{x} + 2$$

$$d_3 = \frac{4}{3} \cdot \frac{z}{y} + 2$$

$$d_4 = 1 \cdot \frac{20}{z} + 1$$

Multistage Logic Networks

- Logical effort generalizes to multistage networks

- *Path Logical Effort*

$$G = \prod g_i$$

- *Path Electrical Effort*

$$H = \frac{C_{out-path}}{C_{in-path}}$$

- *Path Effort*

$$F = \prod f_i = \prod g_i h_i$$

- Can we write $F = GH$?

Paths that Branch

□ No! Consider paths that branch:

$$\underline{G} = 1.6 \cdot 1.5$$

$$H = 90 / 5 \approx 18$$

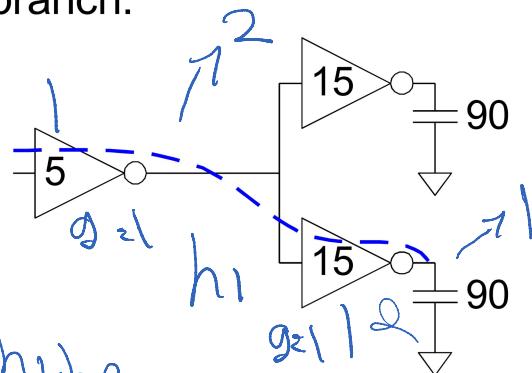
$$GH = 1.6 \cdot 18$$

$$h_1 = 15 / 15 = 3$$

$$h_2 = 90 / 15 = 6 \quad \left. \begin{array}{l} H \approx h_1 + h_2 \\ \approx 3 + 6 = 18 \end{array} \right\}$$

$$F = G \cdot H \cdot B = 1.6 \cdot 18 \cdot 2 = 57.6$$

$$B = 2 \cdot 1$$



Multistage Delays

- Path Effort Delay

$$D_F = \sum f_i$$

- Path Parasitic Delay

$$P = \sum p_i$$

- Path Delay

$$D = \sum d_i = D_F + P$$

Designing Fast Circuits

$$D = \sum d_i = D_F + P$$

- Delay is smallest when each stage bears same effort

$$\hat{f} = g_i h_i = F^{\frac{1}{N}}$$

- Thus minimum delay of N stage path is

$$N \hat{f} + P$$

- This is a **key** result of logical effort
 - Find fastest possible delay
 - Doesn't require calculating gate sizes

Gate Sizes

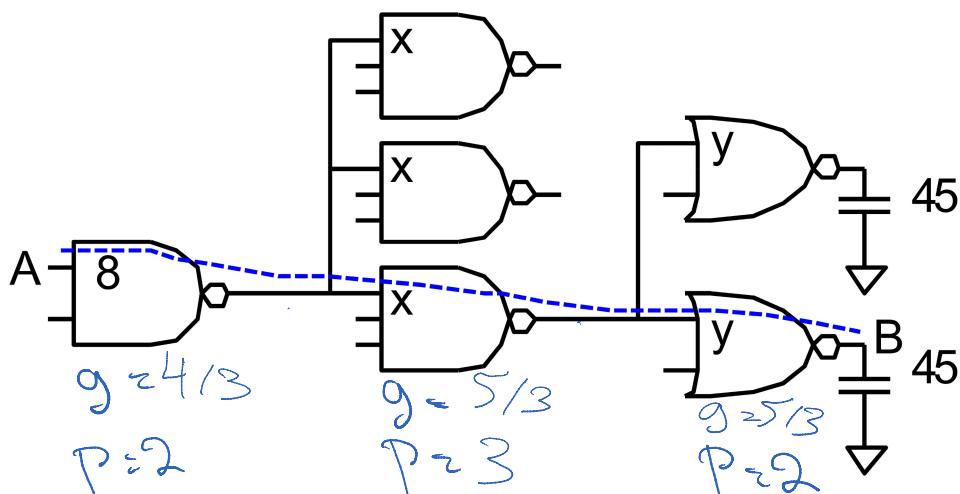
- How wide should the gates be for least delay?

$$\hat{f} = gh = g \frac{C_{out}}{C_{in}}$$
$$\Rightarrow C_{in_i} = \frac{g_i C_{out_i}}{\hat{f}}$$

- Working backward, apply capacitance transformation to find input capacitance of each gate given load it drives.
- Check work by verifying input cap spec is met.

Example: 3-stage path

- Select gate sizes x and y for least delay from A to B



$N = 3$

$$H = \frac{45}{3}$$

$$P = 2 + 3 + 2 = 7$$

$$G = 4/3 \cdot \frac{5}{3} \cdot \frac{5}{3} = \frac{100}{27}$$

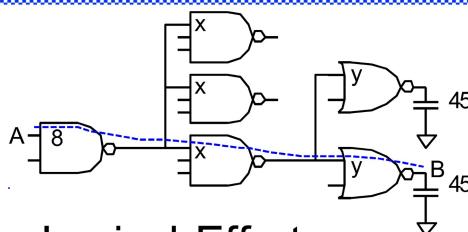
$$B = 3 + 2 + 1 = 6$$

$$F = G \cdot B \cdot H = \frac{45}{27} \cdot \frac{100}{27} \cdot \frac{6}{3} = 125 = 5^3$$

$$S = F^{1/N} = 125^{1/3} \approx 5$$

$$D = N S + P = 3 \times 5 + 7 = 22 \text{ (3RC)} = 66 \text{ RC}$$

Example: 3-stage path



Logical Effort $G =$
Electrical Effort $H =$
Branching Effort $B =$
Path Effort $F =$
Best Stage Effort $\hat{f} =$
Parasitic Delay $P =$
Delay $D =$

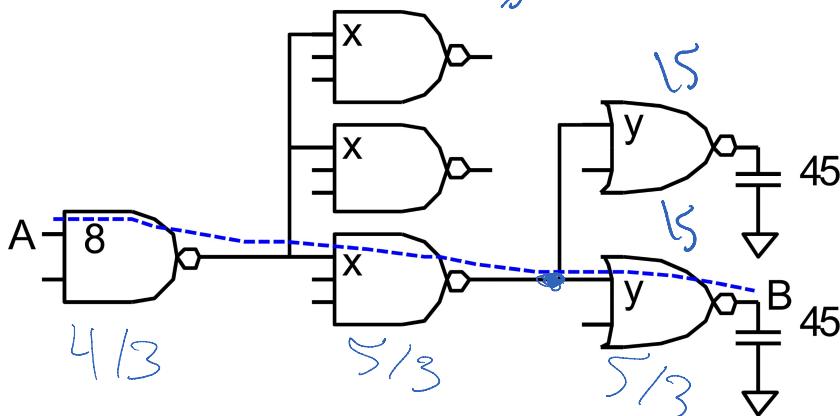
Example: 3-stage path

- Work backward for sizes

$$y = 15$$

$$x = 10$$

$$\frac{C_{in} \approx g; C_{out}}{C_{in}y = \frac{g/3 \cdot 45}{8} \approx 15}$$

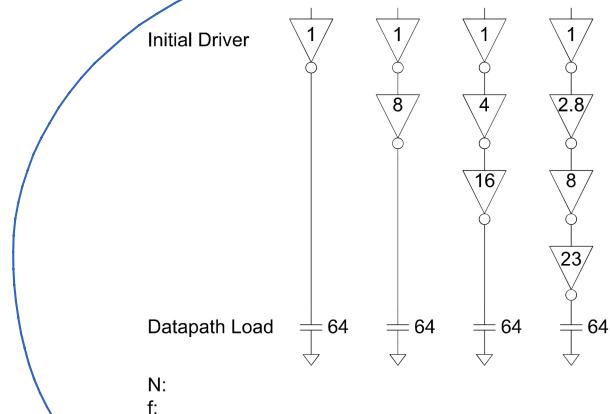


$$X = \frac{g/3 + 3g/10}{8} \approx 10$$

Best Number of Stages

- How many stages should a path use?
 - Minimizing number of stages is not always fastest
- Example: drive 64-bit datapath with unit inverter

D =



Derivation

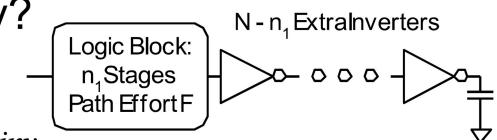
- Consider adding inverters to end of path
 - How many give least delay?

$$D = NF^{\frac{1}{N}} + \sum_{i=1}^{n_1} p_i + (N - n_1) p_{inv}$$

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

- Define best stage effort $\rho = F^{\frac{1}{N}}$

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

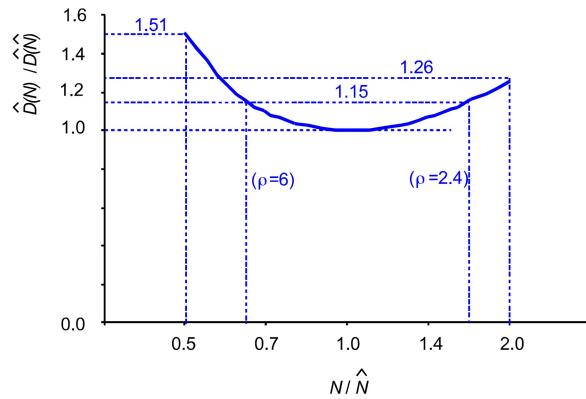


Best Stage Effort

- $p_{inv} + \rho(1 - \ln \rho) = 0$ has no closed-form solution
- Neglecting parasitics ($p_{inv} = 0$), we find $\rho = 2.718$ (e)
- For $p_{inv} = 1$, solve numerically for $\rho = 3.59$

Sensitivity Analysis

- How sensitive is delay to using exactly the best number of stages?



- $2.4 < \rho < 6$ gives delay within 15% of optimal
 - We can be sloppy!
 - I like $\rho = 4$

$$\log_2 1624 = 2^x \stackrel{x=10}{=} 1624 \quad \log_4 F = x$$

Comparison

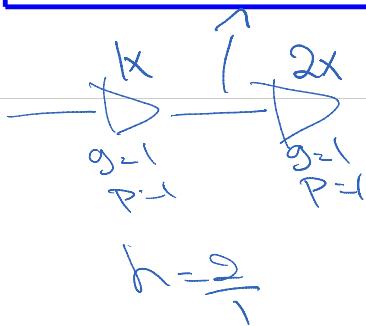
$$4^x = F$$

- Compare many alternatives with a spreadsheet
- $D = N(76.8 G)^{1/N} + P$

| Design | N | G | P | D |
|-----------------------------|---|------|---|------|
| NOR4 | 1 | 3 | 4 | 234 |
| NAND4-INV | 2 | 2 | 5 | 29.8 |
| NAND2-NOR2 | 2 | 20/9 | 4 | 30.1 |
| INV-NAND4-INV | 3 | 2 | 6 | 22.1 |
| NAND4-INV-INV-INV | 4 | 2 | 7 | 21.1 |
| NAND2-NOR2-INV-INV | 4 | 20/9 | 6 | 20.5 |
| NAND2-INV-NAND2-INV | 4 | 16/9 | 6 | 19.7 |
| INV-NAND2-INV-NAND2-INV | 5 | 16/9 | 7 | 20.4 |
| NAND2-INV-NAND2-INV-INV-INV | 6 | 16/9 | 8 | 21.6 |

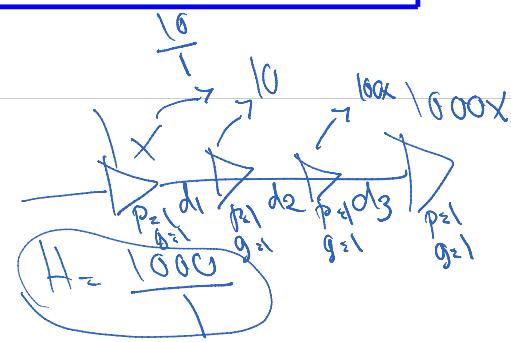
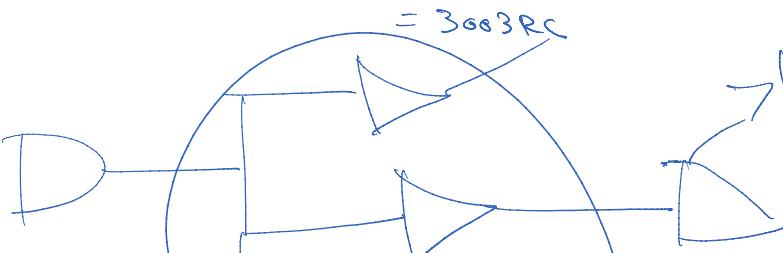
Review of Definitions

| Term | Stage | Path |
|-------------------|--|--|
| number of stages | 1 | N |
| logical effort | g | $G = \prod g_i$ |
| electrical effort | $h = \frac{C_{out}}{C_{in}}$ | $H = \frac{C_{out-path}}{C_{in-path}}$ |
| branching effort | $b = \frac{C_{on-path} + C_{off-path}}{C_{on-path}}$ | $B = \prod b_i$ |
| effort | $f = gh$ | $F = GBH$ |
| effort delay | f | $D_F = \sum f_i \rightarrow \neq N$ |
| parasitic delay | p | $P = \sum p_i$ |
| delay | $d = f + p$ | $D = \sum d_i = D_F + P$ |



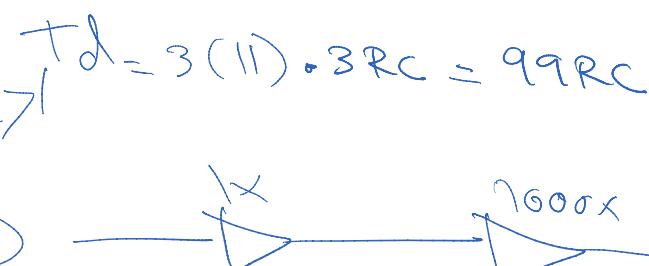
$$d_1 = 1 \cdot 2 + 1 = 3 \quad (3RC) = 9RC$$

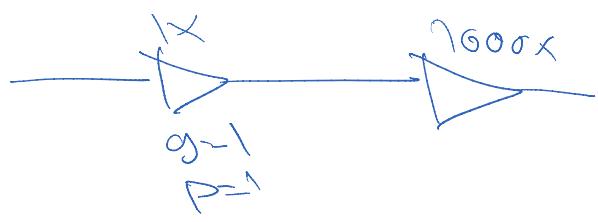
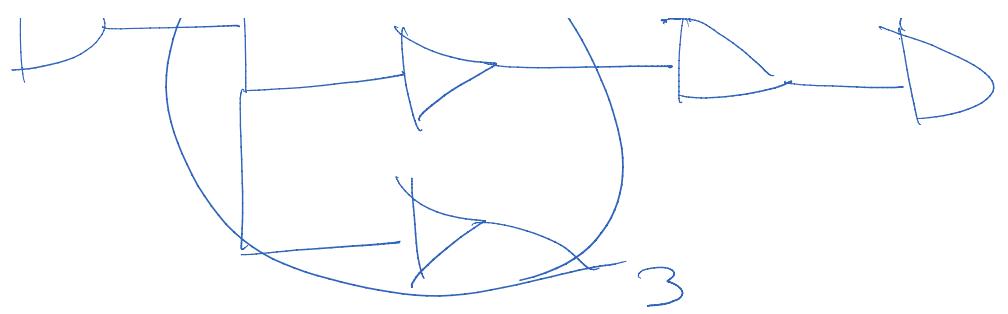
$$D = 1 \cdot \frac{1000}{10} + 1 = 101 \cdot 3RC$$



$$d_2 = 1 \cdot \frac{100}{10} + 1 = 11$$

$$d_3 = 1 \cdot \frac{1000}{100} + 1 = 11$$





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 | $C_{in_i} = \frac{g_i C_{out_i}}{\hat{f}}$ |

Limits of Logical Effort

- Chicken and egg problem
 - Need path to compute G
 - But don't know number of stages without G
- Simplistic delay model
 - Neglects input rise time effects
- Interconnect
 - Iteration required in designs with wire
- Maximum speed only
 - Not minimum area/power for constrained delay

Summary

- Logical effort is useful for thinking of delay in circuits
 - Numeric logical effort characterizes gates
 - NANDs are faster than NORs in CMOS
 - 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