

# Review: Logical Effort

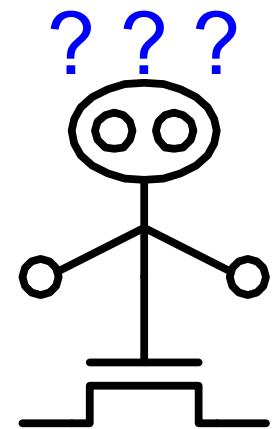
# Outline

---

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



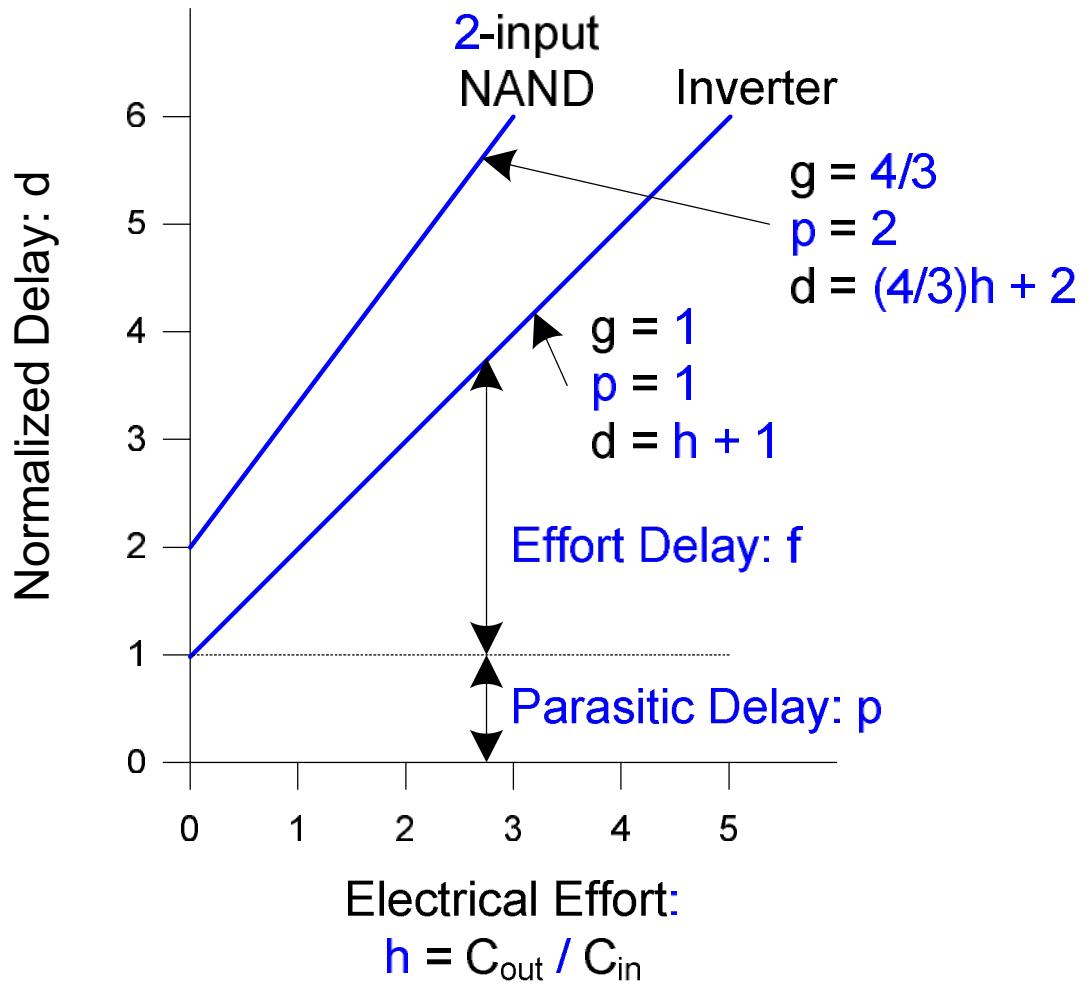
# Delay in a Logic Gate

- Express delays in process-independent unit  $d = \frac{d_{abs}}{\tau}$
- Delay has two components:  $d = f + p$
- $f$ : *effort delay* =  $gh$  (a.k.a. stage effort)
  - Again has two components
- $g$ : *logical effort*
  - Measures relative ability of gate to deliver current
  - $g \equiv 1$  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

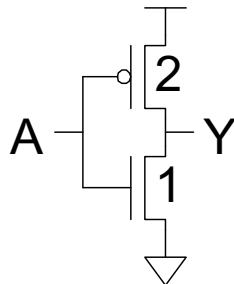
$$\begin{aligned}d &= f + p \\&= gh + p\end{aligned}$$

- What about NOR2?

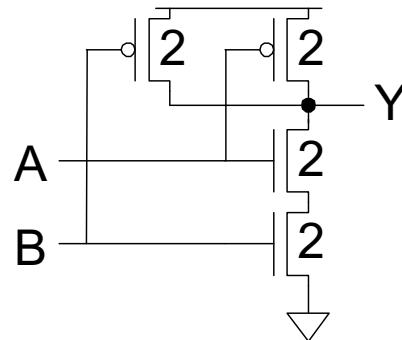


# Computing Logical Effort

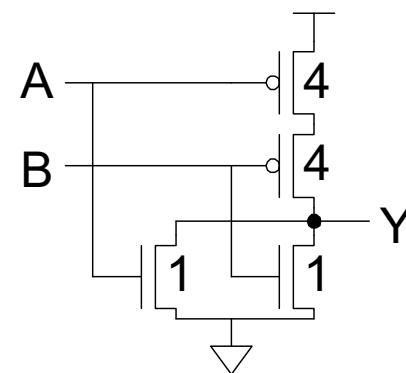
- DEF: *Logical effort is the ratio of the input capacitance of a gate to the input capacitance of an inverter delivering the same output current.*
- Measure from delay vs. fanout plots
- Or estimate by counting transistor widths



$$C_{in} = 3$$
$$g = 3/3$$



$$C_{in} = 4$$
$$g = 4/3$$



$$C_{in} = 5$$
$$g = 5/3$$

# Catalog of Gates

- Logical effort of common gates

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	

- Compare the growth rate for NAND and NOR!

# Parasitic Delay

---

- Represents delay of gate driving no load:
    - Set by internal parasitic capacitance
  - Counts only direct diffusion capacitance on the output node:
    - e.g., unit inverter has 3 units of diffusion capacitance on the output
  - Definition:  $p_{inv}$  is the ratio of diffusion capacitance to gate capacitance of an inverter
    - Also called normalized parasitic delay
    - $p_{inv}$  of an inverter is 1
  - Increasing transistor sizes reduces resistance but increases capacitance:
    - Parasitic delay is independent of gate size
  - This is inaccurate but it gives a general idea about the delay
-

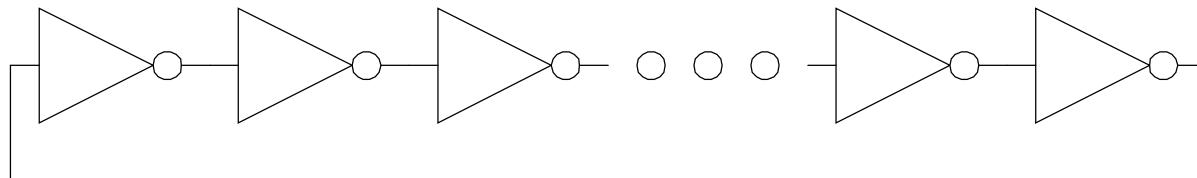
# Catalog of Gates

- Parasitic delay of common gates
  - In multiples of  $p_{inv}$  ( $\approx 1$ )

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:  $g = 1$

Electrical Effort:  $h = 1$

Parasitic Delay:  $p = 1$

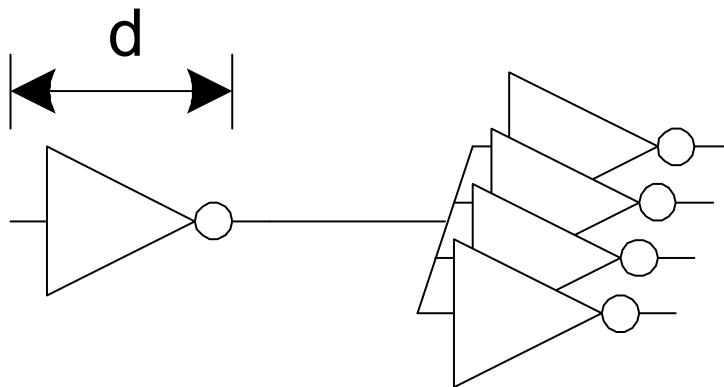
Stage Delay:  $d = 2$

Frequency:  $f_{osc} = 1/(2*N*d) = 1/4N$

31 stage ring oscillator in  
65 nm process has  
frequency of  $\sim 2.7$  GHz

# Example: FO4 Inverter

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



Logical Effort:  $g = 1$

Electrical Effort:  $h = 4$

Parasitic Delay:  $p = 1$

Stage Delay:  $d = 5$

The FO4 delay is about

300 ps in 0.6  $\mu\text{m}$  process

15 ps in a 65 nm process

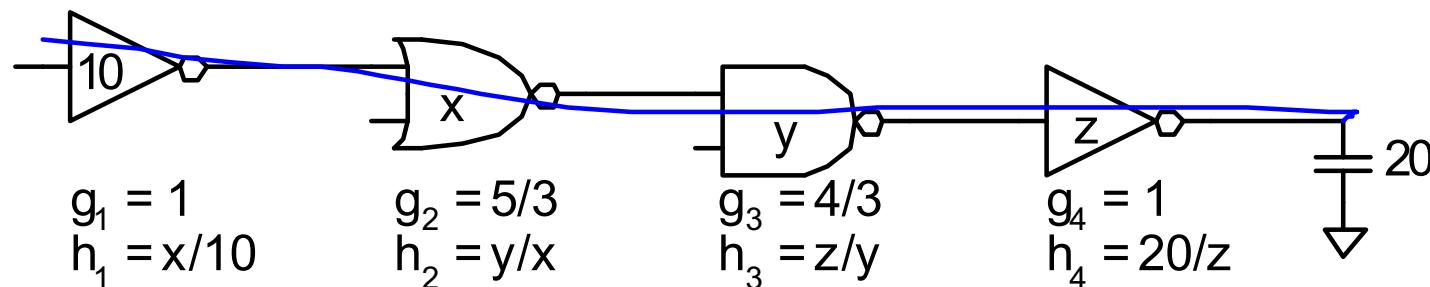
# Multistage Logic Networks

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



# 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:

$$G = 1$$

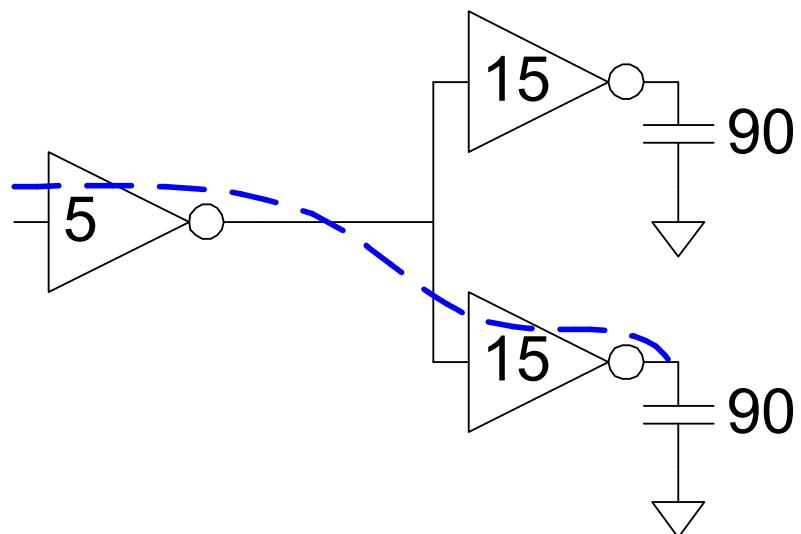
$$H = 90 / 5 = 18$$

$$GH = 18$$

$$h_1 = (15 + 15) / 5 = 6$$

$$h_2 = 90 / 15 = 6$$

$$F = g_1 g_2 h_1 h_2 = 36 = 2GH$$



# Branching Effort

---

- Introduce *branching effort*
  - Accounts for branching between stages in path

$$b = \frac{C_{\text{on path}} + C_{\text{off path}}}{C_{\text{on path}}}$$

$$B = \prod b_i$$

Note:

$$\prod h_i = BH$$

- Now we compute the path effort
  - $F = GBH$

# 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

$$D = NF^{\frac{1}{N}} + P$$

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

# Gate Sizes

- How wide should the gates be for least delay?

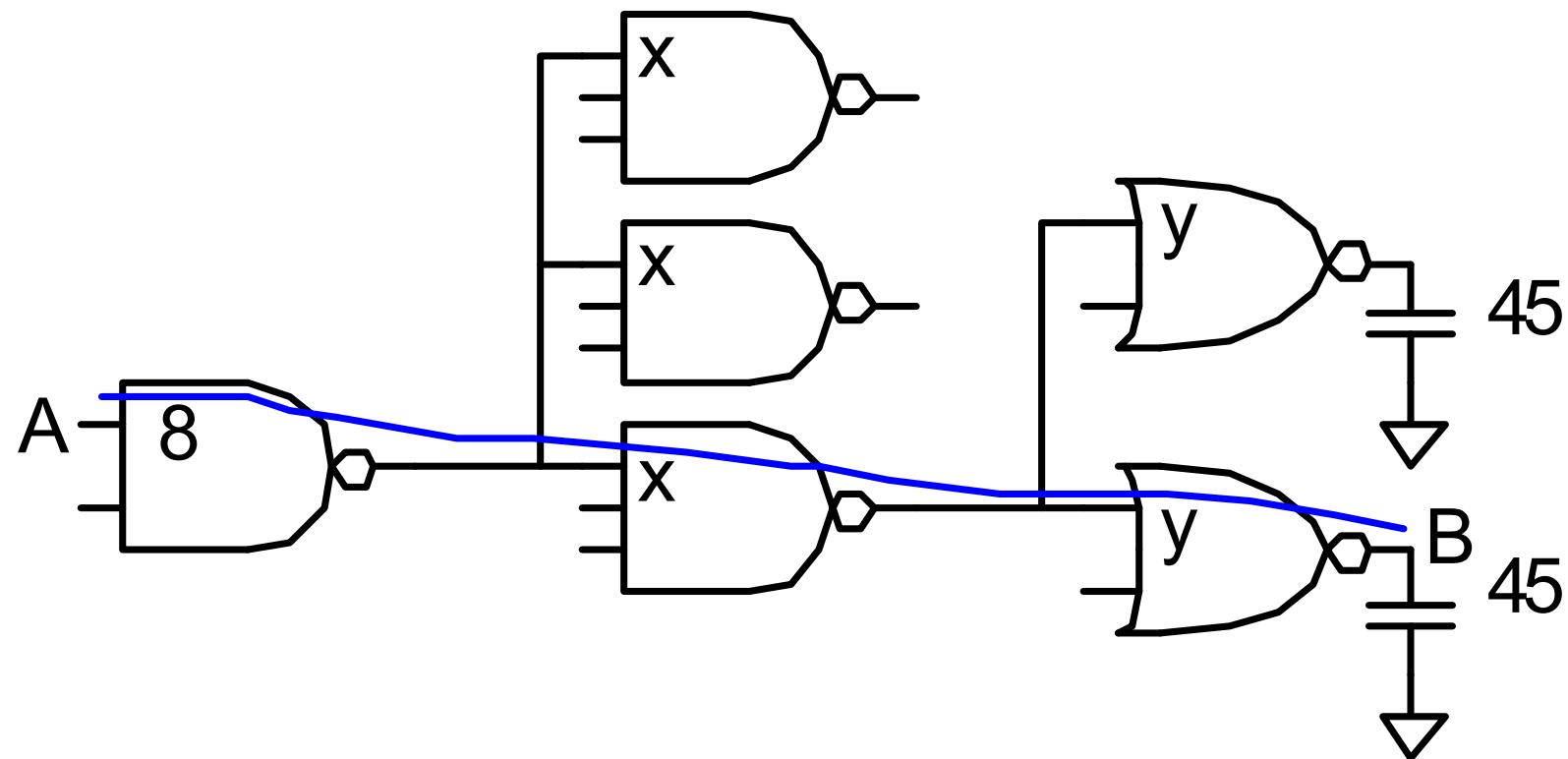
$$\hat{f} = gh = g \frac{C_{out}}{C_{in}}$$

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

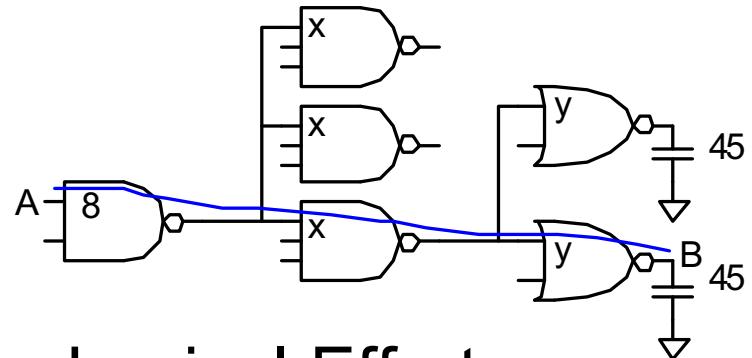
- Working backward, apply capacitance transformation to find input capacitance of each gate given load it drives.
- Check work by verifying input cap spec is met.

# Example: 3-stage path

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



# Example: 3-stage path



Logical Effort

$$G = (4/3) * (5/3) * (5/3) = 100/27$$

Electrical Effort

$$H = 45/8$$

Branching Effort

$$B = 3 * 2 = 6$$

Path Effort

$$F = GBH = 125$$

Best Stage Effort

$$125^{1/3} = 5$$

Parasitic Delay

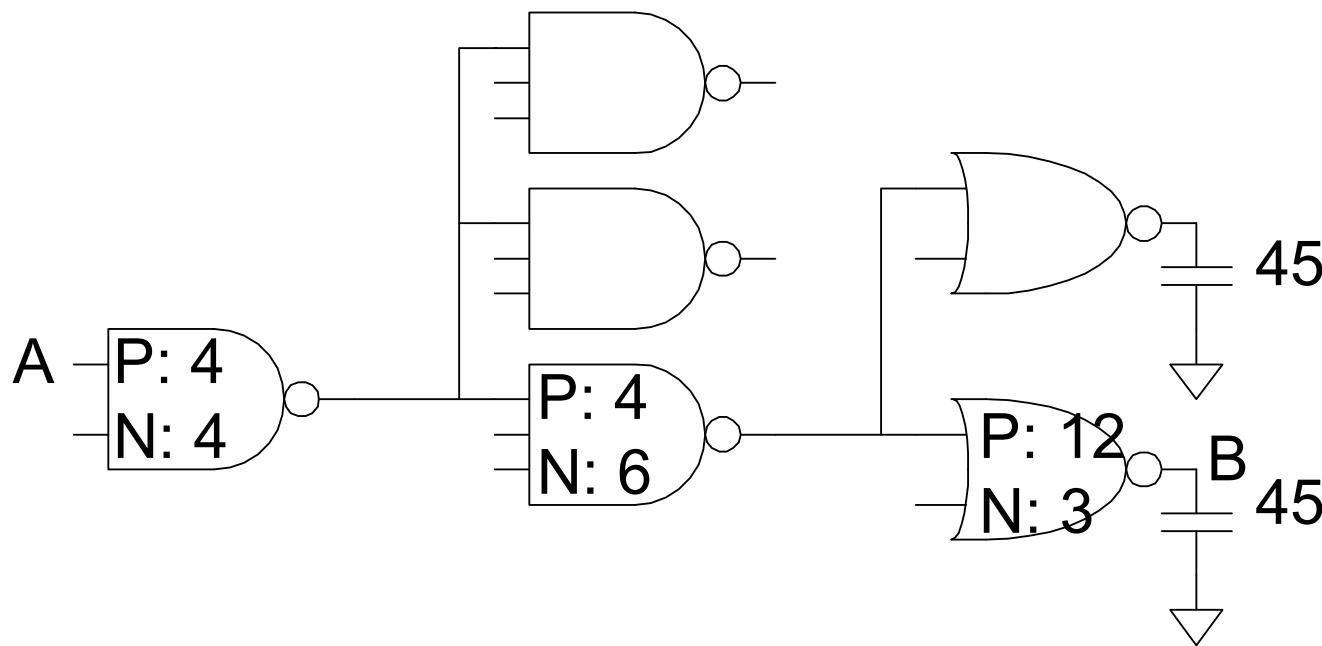
$$P = 2 + 3 + 2 = 7$$

Delay

$$D = 3 * 5 + 7 = 22 = 4.4 \text{ FO4}$$

# Example: 3-stage path

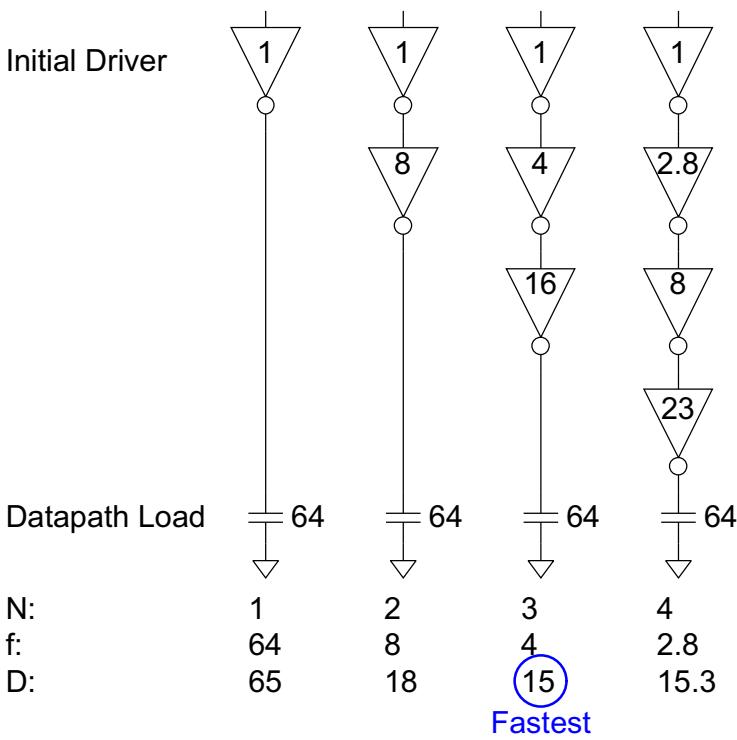
- Work backward for sizes  
 $y = 45 * (5/3) / 5 = 15$   
 $x = (15*2) * (5/3) / 5 = 10$
- Verify the first stage:  
 $(10*3/8) * (4/3) = 5$



# 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

$$\begin{aligned}D &= NF^{1/N} + P \\&= N(64)^{1/N} + N\end{aligned}$$



# Derivation

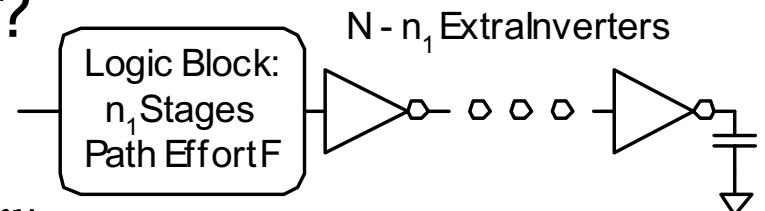
- Consider adding inverters to end of path
  - How many give least delay?

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

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

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

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



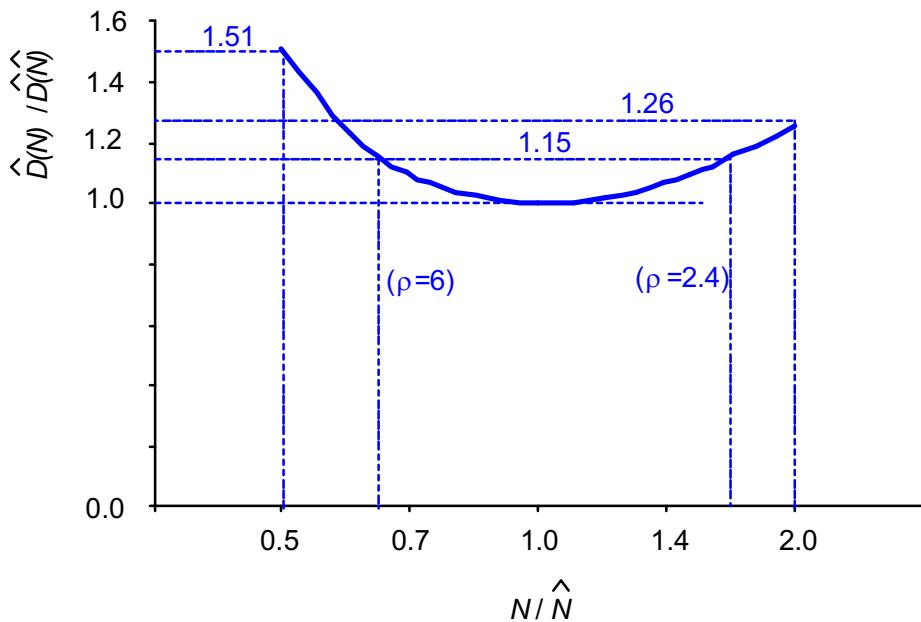
Note: This equation assumes that all added inverters are unit inverters.

# 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  $p_{inv} = 1$ , solving numerically gives  $\rho = 3.59$
- $\hat{N}$  can be found from  $\hat{N} = \log_\rho F$
- Need to find an integer close to it:
  - Increasing  $\rho$  and decreasing  $\hat{N}$  => lower parasitic delay
  - Decreasing  $\rho$  and increasing  $\hat{N}$  => higher parasitic delay
- The first approach is usually taken

# 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 imprecise!
  - Typically  $\rho = 4$

# Summary

## ❑ 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

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

# Homework Assignments

---

- Chapter 4: 4.5 to 4.10
- For Tuesday 1402/9/14