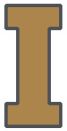
# ECE 445 INTRO TO VLSI DESIGN

## DIGITAL DESIGN USING LOGICAL EFFORT

VISHAL SAXENA
VSAXENA@UIDAHO.EDU





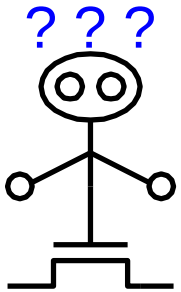
University of Idaho
College of Engineering

# 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

# DELAY IN A LOGIC GATE

Express delays in process-independent unit

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

$$d = \frac{d_{abs}}{\tau}$$

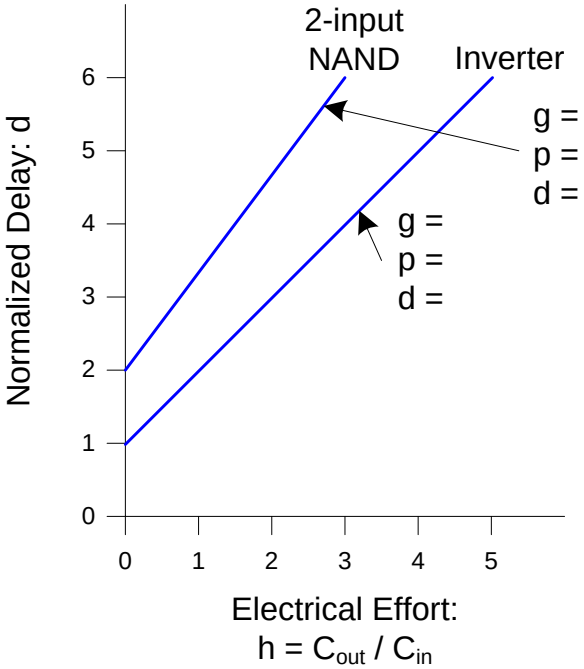
$\tau$ = 3RC

≈ 3 ps in 65 nm process

60 ps in 0.6 μm process

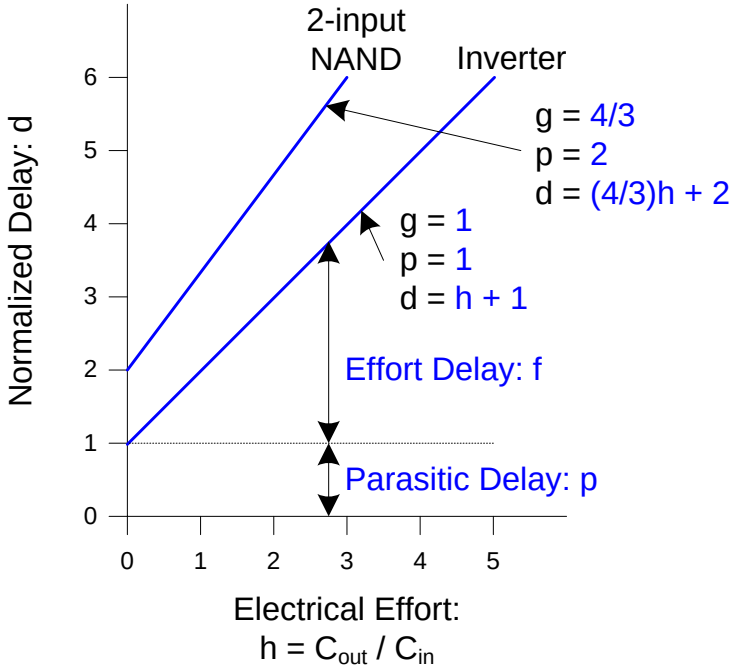
# DELAY PLOTS

- $d = f + p$
  - $= gh + p$
- What about
- NOR2?



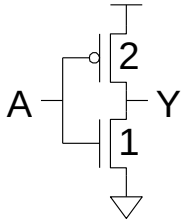
# DELAY PLOTS

- $d = f + p$
  - $= gh + p$
- What about
- NOR2?

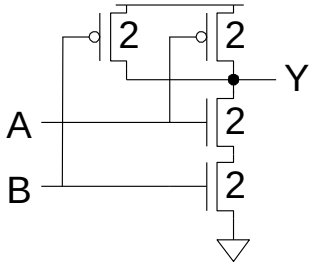


# COMPUTING LOGICAL EFFORT

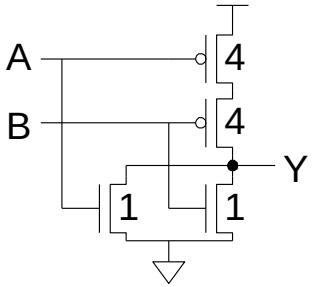
- DEF: *Logical effort is the ratio of the input capacitance of a gate to the input capacitance of an inverter delivering the same output current*.
- Measure from delay vs. fanout plots
- Or estimate by counting transistor widths




$C_{in}$ = 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 | |

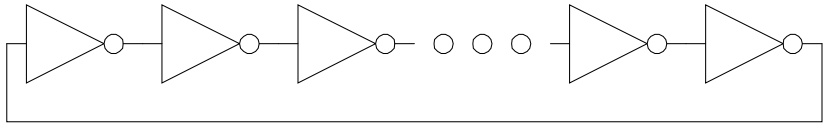
# CATALOG OF GATES

- 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 | 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

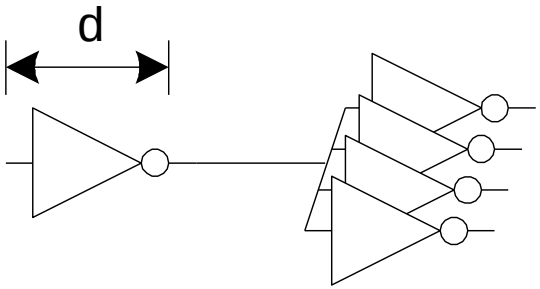


- Logical Effort: g = 1
- Electrical Effort: h = 1
- Parasitic Delay: p = 1
- Stage Delay: d = 2
- Frequency: $f_{osc} = 1/(2*N*d) = 1/4N$

31 stage ring oscillator in 0.6 μm process has frequency of ~ 200 MHz

# 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 μm process

15 ps in a 65 nm process

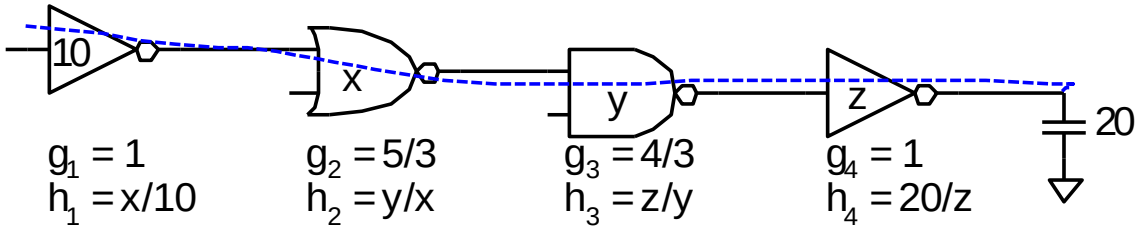
# MULTISTAGE LOGIC NETWORKS

- Logical effort generalizes to multistage networks
- *Path Logical Effort*
- *Path Electrical Effort*
- *Path Effort*

$$G = \prod g_i$$

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

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



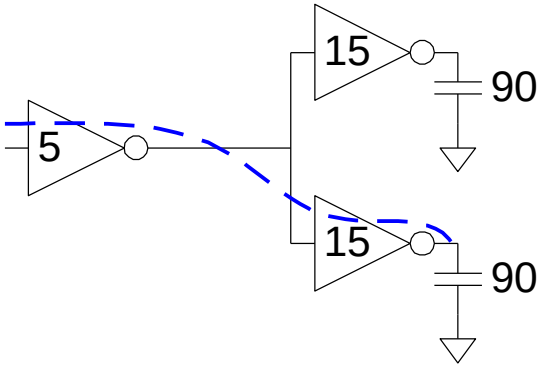
# MULTISTAGE LOGIC NETWORKS

- 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

- Delay is smallest when each stage bears same effort

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

- Thus minimum delay of N stage path is

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

- 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?

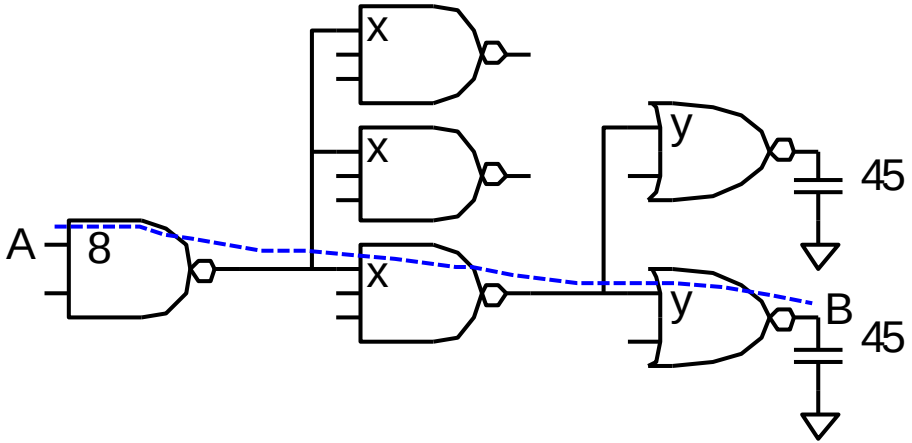
$$\hat{f} = gh = g\frac{C_{out}}{C_{in}}$$

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

- 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.

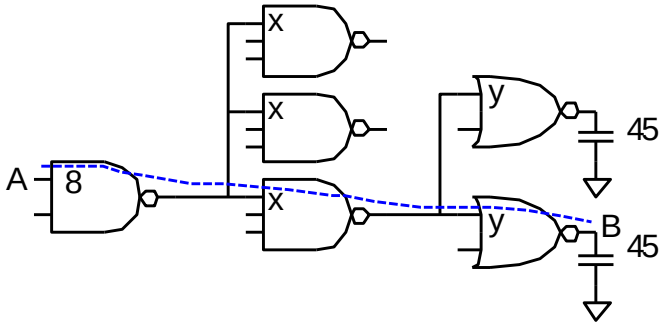
# EXAMPLE: 3-STAGE PATH

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



# 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 | $\hat{f} = \sqrt[3]{F} = 5$ |
| Parasitic Delay | P = 2 + 3 + 2 = 7 |
| Delay | D = 3*5 + 7 = 22 = 4.4 FO4 |

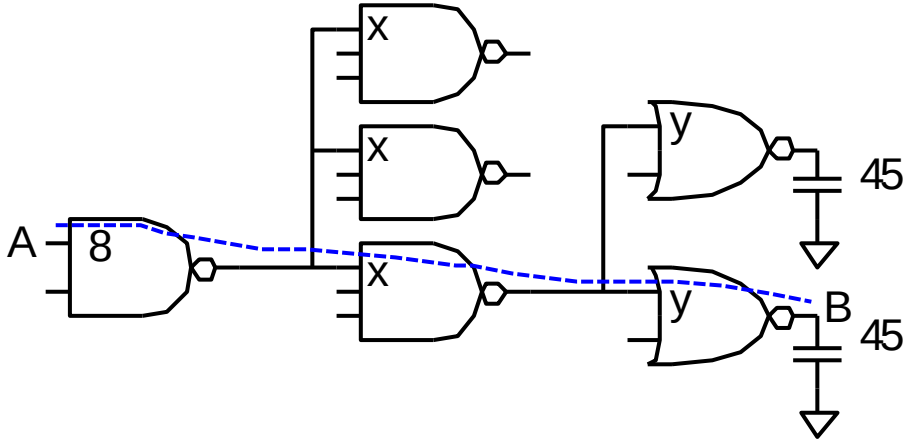


# EXAMPLE: 3-STAGE PATH

- Work backward for sizes
- y = 45 * (5/3) / 5 = 15
- x = (15*2) * (5/3) / 5 = 10

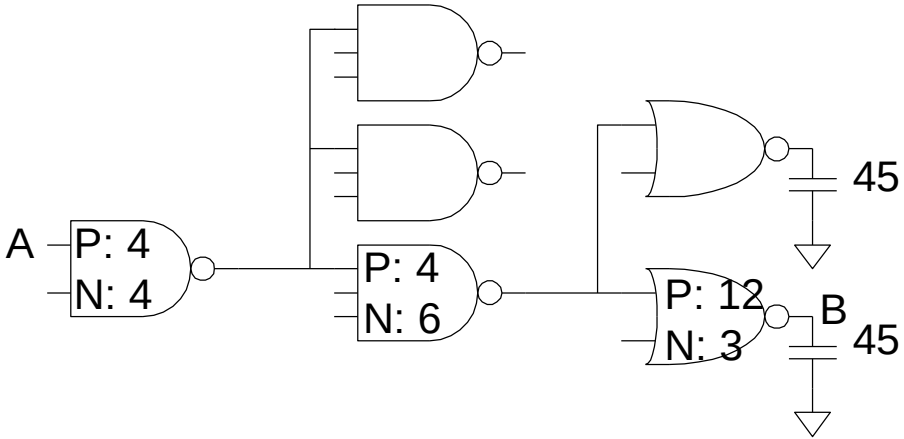
$$\hat{f} = gh = g\frac{C_{out}}{C_{in}}$$

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



# EXAMPLE: 3-STAGE PATH

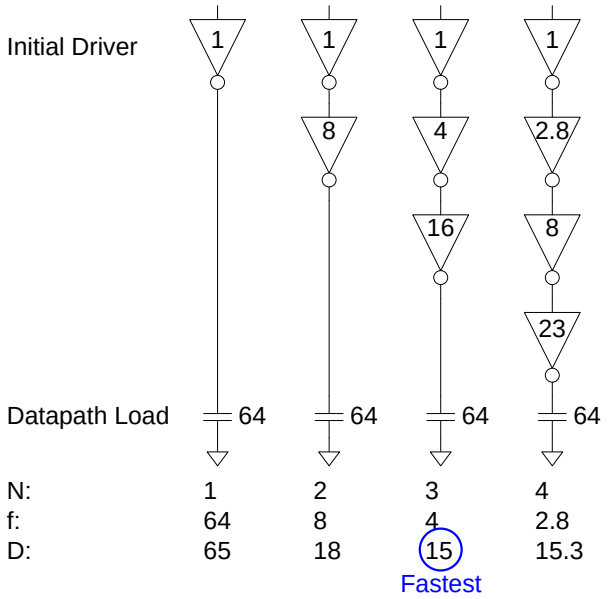
- Work backward for sizes
- y = 45 * (5/3) / 5 = 15
- x = (15*2) * (5/3) / 5 = 10



# 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

- $D = NF^{1/N} + P$
- $= N(64)^{1/N} + N$

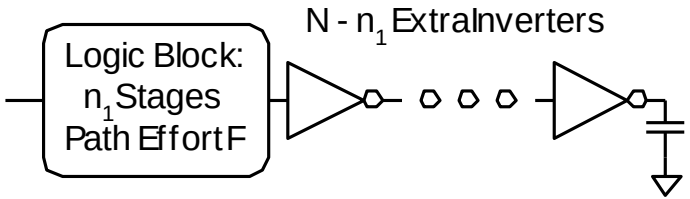


# DERIVATION

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

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

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




- Define best stage effort

$$\rho = F^{\frac{1}{N}}$$

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

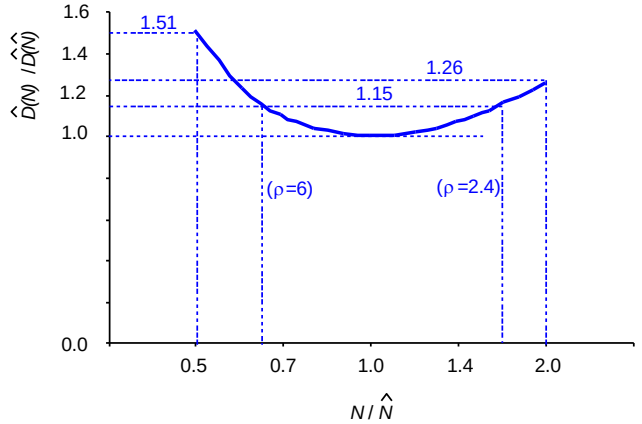
# BEST STAGE EFFORT

- $p_{inv} + \rho(1 - \ln \rho) = 0$ 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 $\rho = 4$

# REVIEW OF DEFINITIONS

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

# METHOD OF LOGICAL EFFORT

1) Compute path effort
2) Estimate best number of stages
3) Sketch path with N stages
4) Estimate least delay
5) Determine best stage effort

6) Find gate sizes

$$F = GBH$$

$$N = \log_4 F$$

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

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

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

# 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