Introduction to CMOS VLSI Design

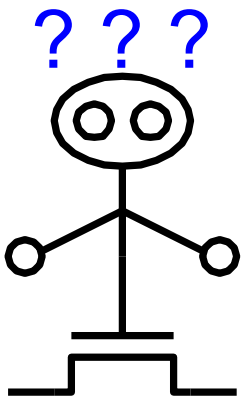
Lecture 5: Logical Effort

Outline

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

Introduction

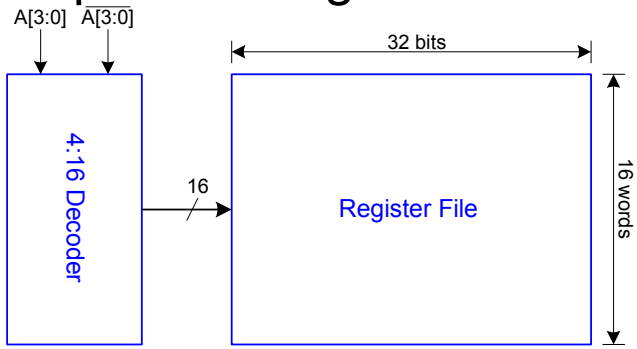
- Chip designers face a bewildering array of choices
  - What is the best circuit topology for a function?
  - How many stages of logic give least delay?
  - How wide should the transistors be?



- 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

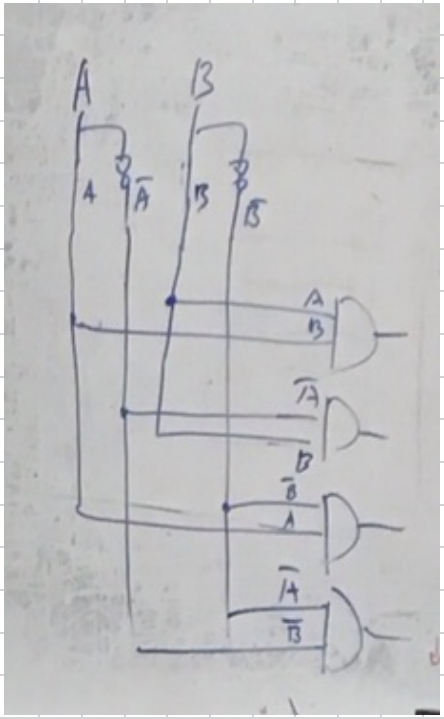
Example

- 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?

這是一個 2 * 4的decoder



Delay in a Logic Gate

- Express delays in process-independent unit

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

τ = 3RC 延遲參數

≈ 12 ps in 180 nm process

40 ps in 0.6 μm process

Delay in a Logic Gate

- Express delays in process-independent unit

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

- Delay has two components

$$d = f + p$$

寄生電容

Delay in a Logic Gate

- Express delays in process-independent unit

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

- Delay has two components

$$d = f + p$$

- *Effort delay f = gh* (a.k.a. *stage effort*)
  - Again has two components

Delay in a Logic Gate

- Express delays in process-independent unit

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

- Delay has two components

$$d = f + p$$

g 和邏輯夾 的種類有關

- Effort delay $f = gh$ (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

Delay in a Logic Gate

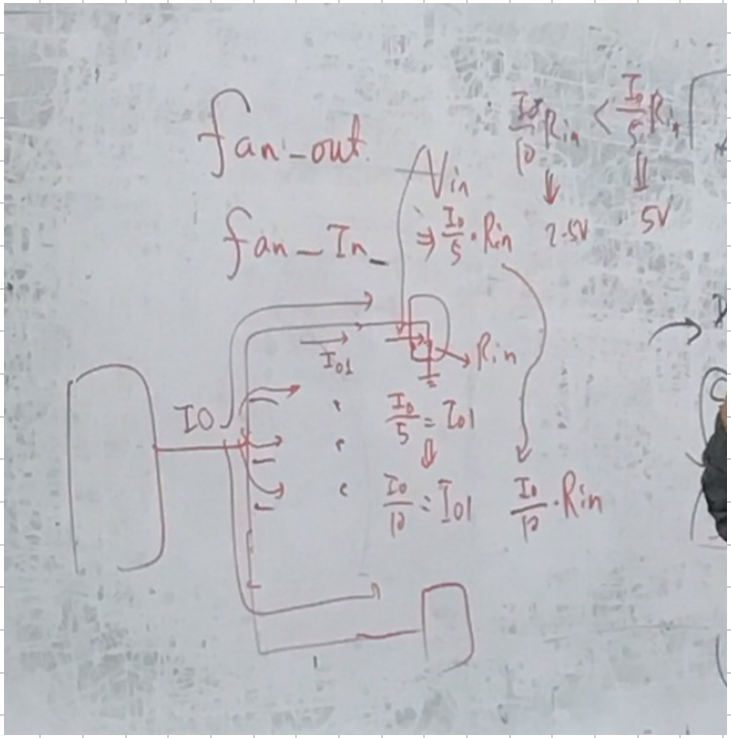
- Express delays in process-independent unit

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

- Delay has two components

$$d = f + p$$

- Effort delay $f = gh$ (a.k.a. stage effort)
  - Again has two components
- *h*: *electrical effort* = C_{out} / C_{in}
  - Ratio of output to input capacitance
  - Sometimes called fanout



可能會因為 接的電路多了
電壓就被分散了 就叫fanout

Fan in 和接地的分散有關

Delay in a Logic Gate

- Express delays in process-independent unit

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

- Delay has two components

$$d = f + p$$

這裡要講p

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

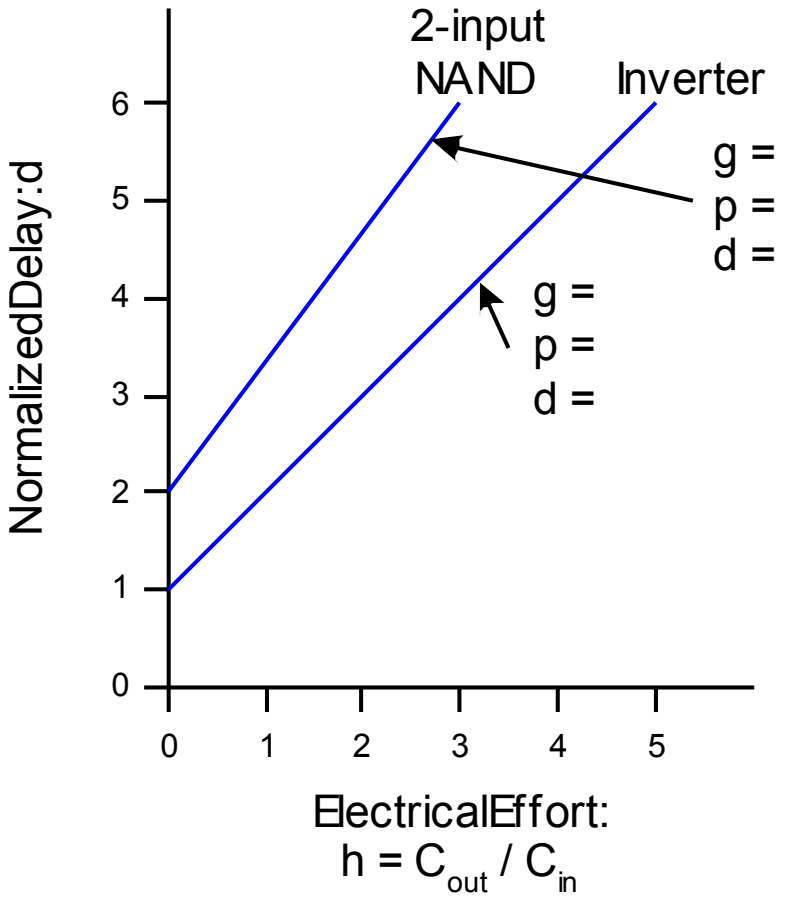
由 寄生電容 來定的

這是 stage delay

Delay Plots

$$d = f + p$$
$$= gh + p$$

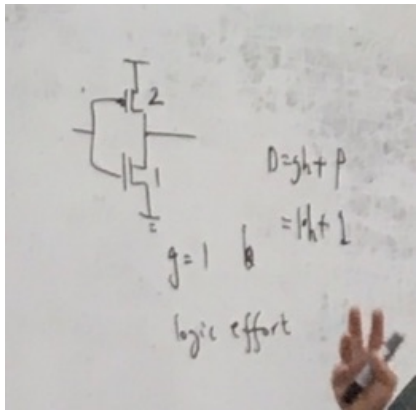
這裡的線從這公式得到的

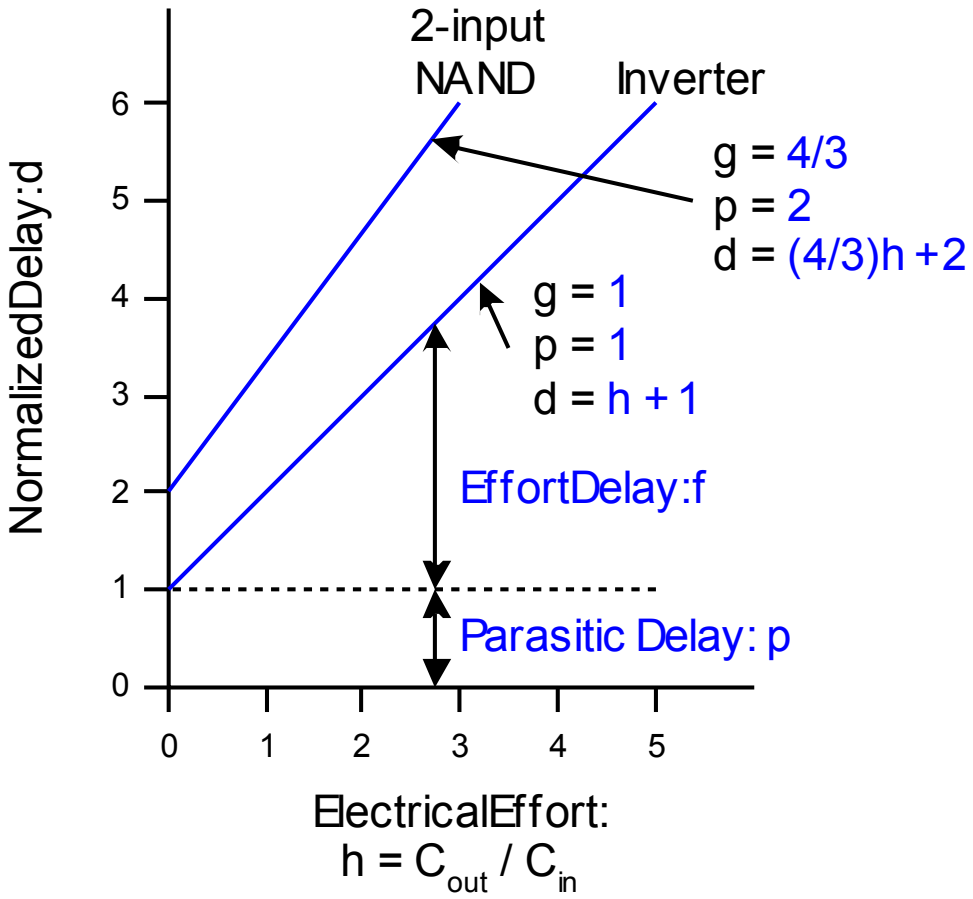


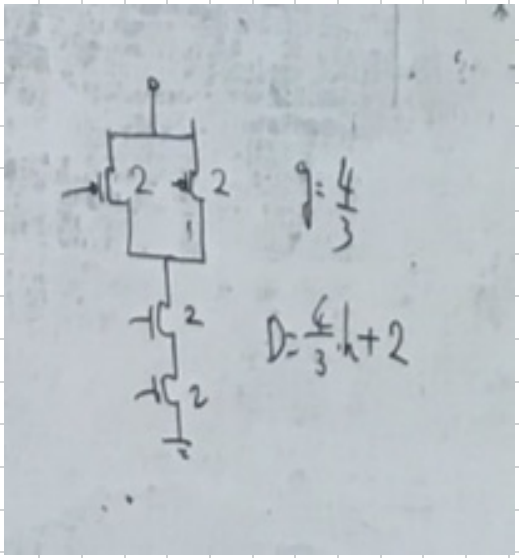
Delay Plots

$$d = f + p$$
$$= gh + p$$

- What about NOR2?

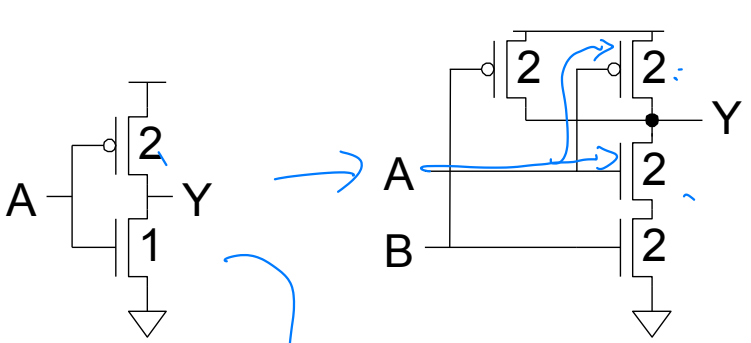





2
2
1
2
2
g = 4/3
D = 4/3 h + 2

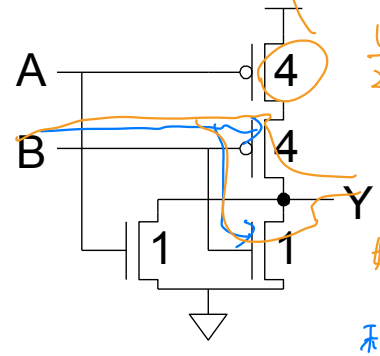
Computing Logical Effort

- 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



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

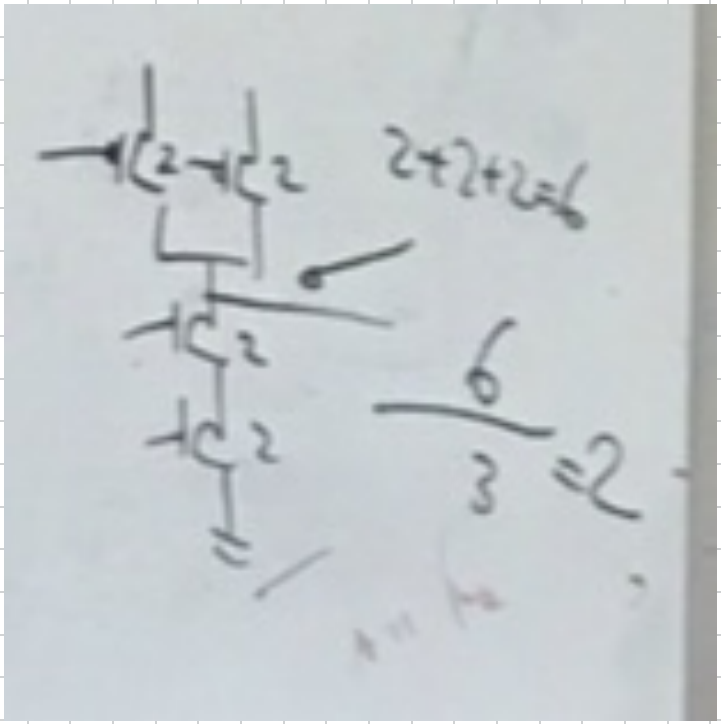
$C_{in} = 4$
$g = 4/3$

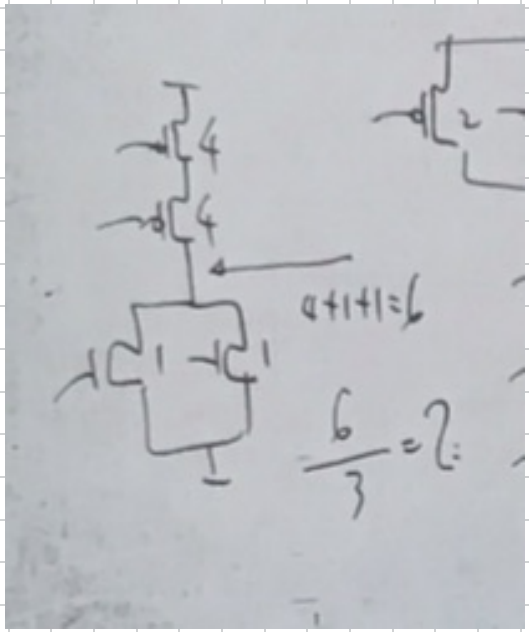


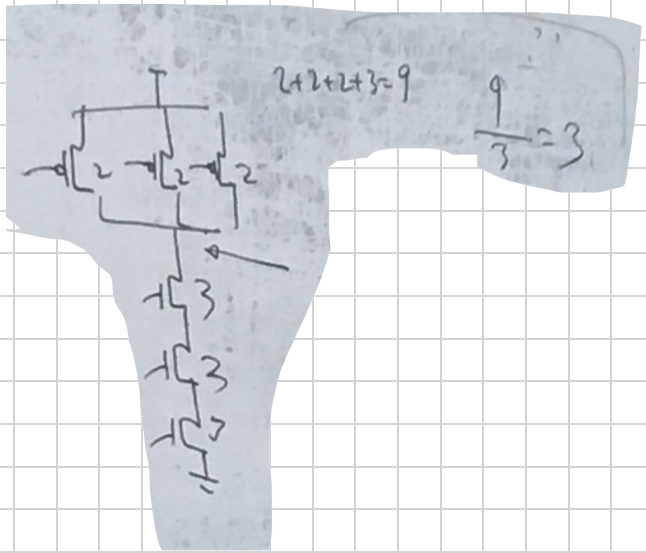
$C_{in} = 5$
$g = 5/3$

串連幾個 放大幾倍

寄生電容
並聯 的 都要算
串連 的 只算一個







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	

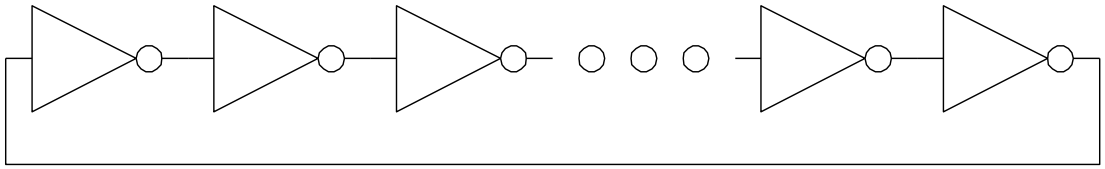
Catalog of Gates

- 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 =

Electrical Effort: h =

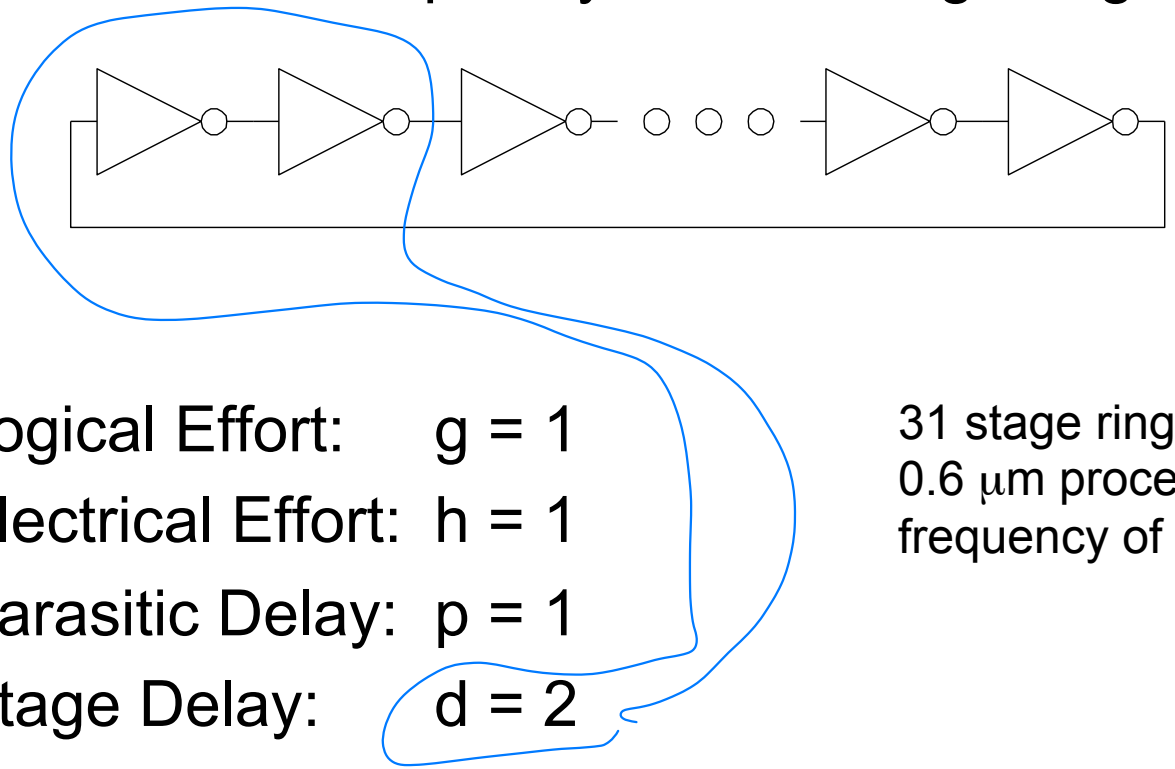
Parasitic Delay: p =

Stage Delay: d =

Frequency: f_{osc} =

Example: Ring Oscillator

- Estimate the frequency of an N-stage ring oscillator



Logical Effort: g = 1

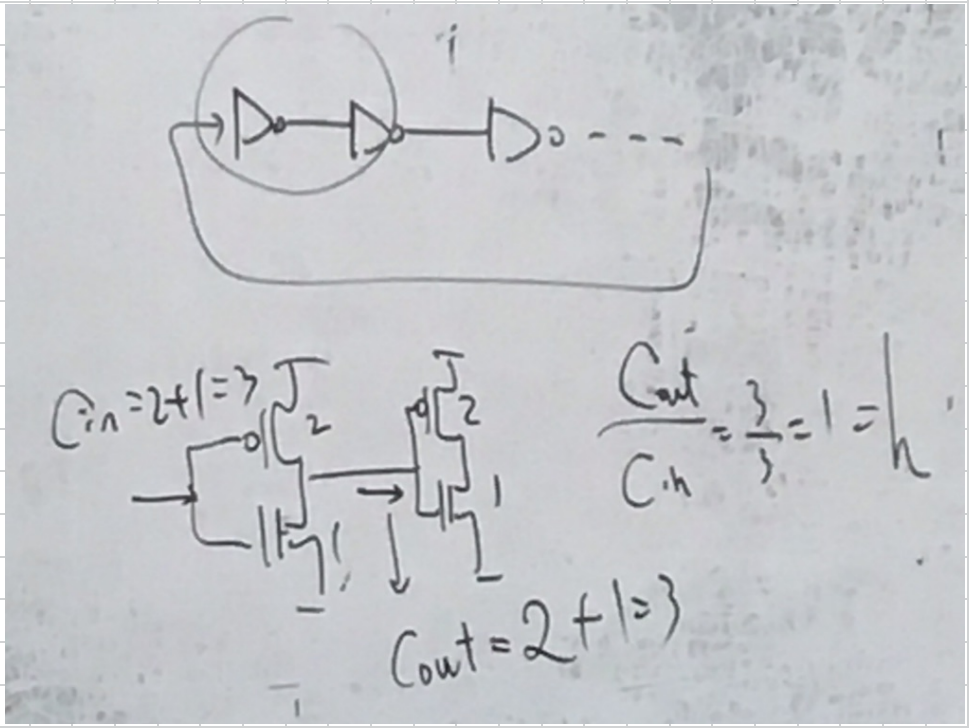
Electrical Effort: h = 1

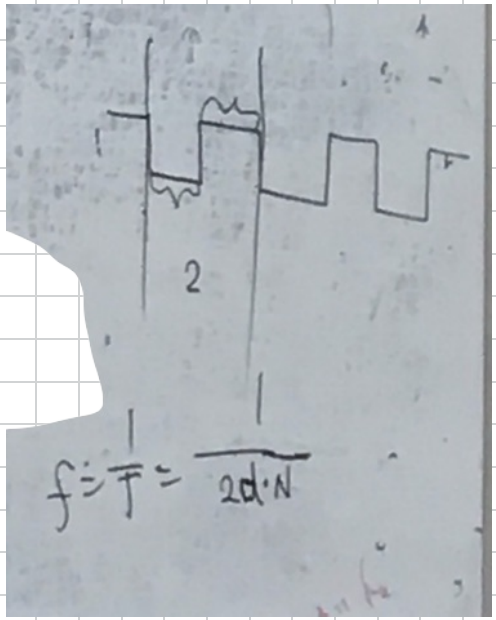
Parasitic Delay: p = 1

Stage Delay: d = 2

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

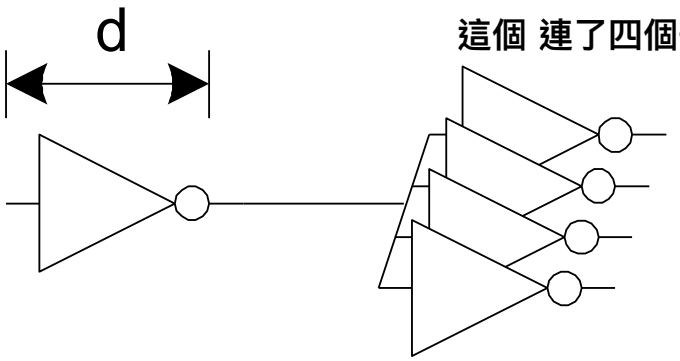
31 stage ring oscillator in 0.6 μm process has frequency of ~ 200 MHz


Cin = 2+1 = 3
2
2
1
1
Cout = 2+1 = 3
Cout / Cin = 3/3 = 1 = h


2
f = 1/T = 1/2d·N

Example: FO4 Inverter

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



Logical Effort: g =

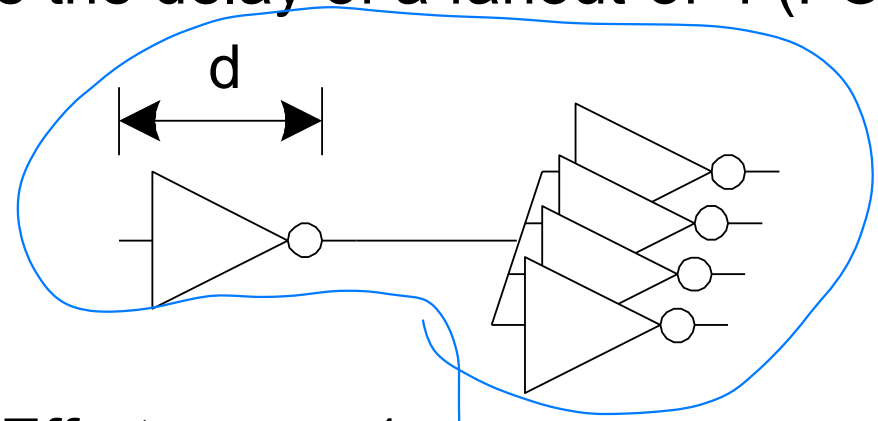
Electrical Effort: h =

Parasitic Delay: p =

Stage Delay: d =

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

$d = gh + p$

The FO4 delay is about

200 ps in 0.6 μm process

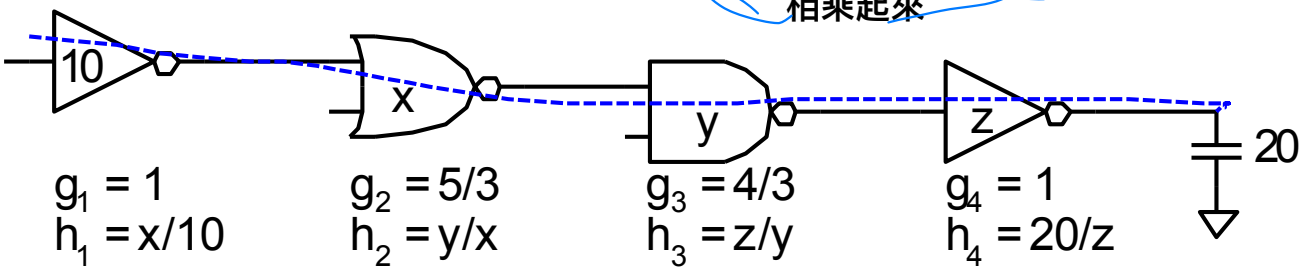
60 ps in a 180 nm process

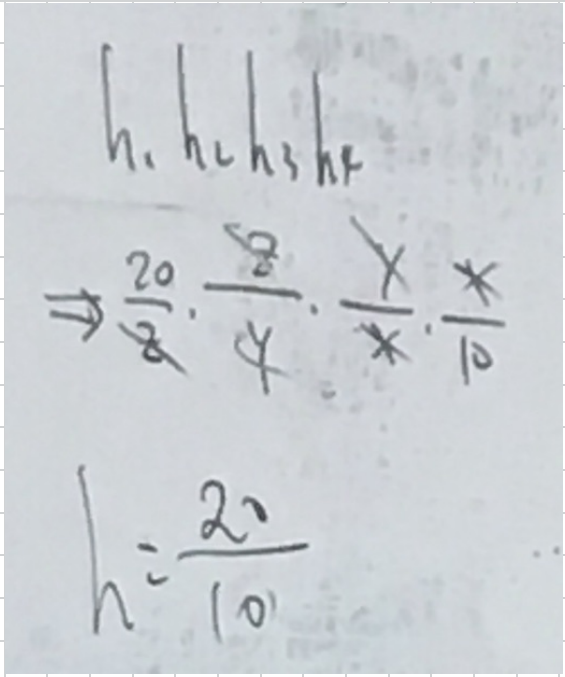
f/3 ns in an *f* μm process

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

相乘起來




$h_1 h_2 h_3 h_4$
$\Rightarrow \frac{20}{2} \cdot \frac{2}{4} \cdot \frac{4}{4} \cdot \frac{4}{10}$
$h = \frac{20}{10}$

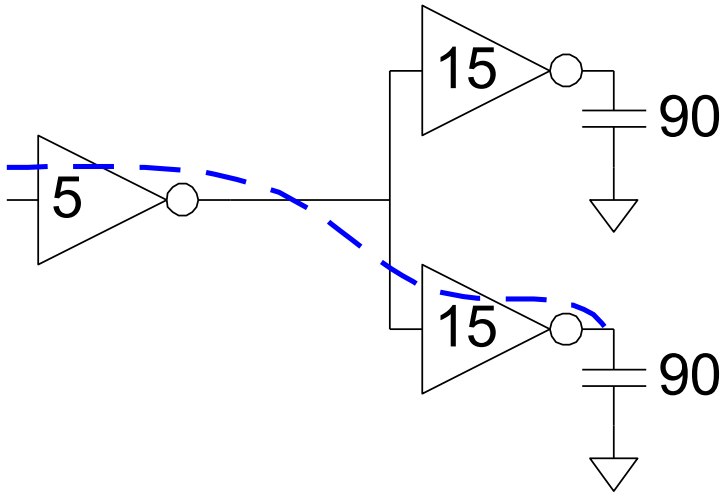
Multistage Logic Networks

- 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?

不可這樣算

Paths that Branch

- No! Consider paths that branch:



G =

H =

GH =

h_1 =

h_2 =

F = GH?

Paths that Branch

- No! Consider paths that branch:

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

$G = 1$

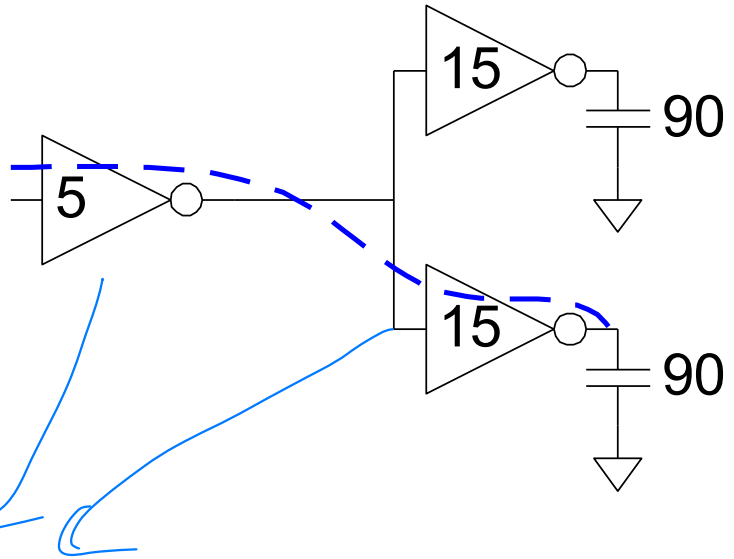
$H = 90 / 5 = 18$

$GH = 18$

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

$h_2 = 90 / 15 = 6$

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



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

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

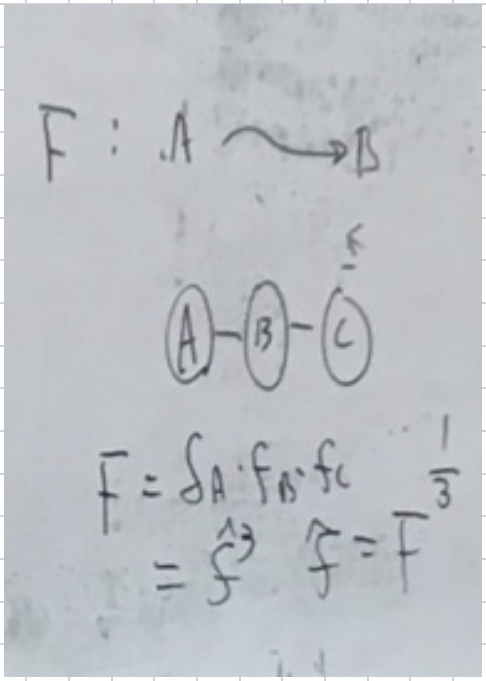
- Delay is smallest when each stage bears same effort

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

- Thus minimum delay of N stage path is

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

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


F : A → B
A
B
C
F = f_A · f_B · f_C
= f^3
f = F^1/3

Gate Sizes

- How wide should the gates be for least delay?

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

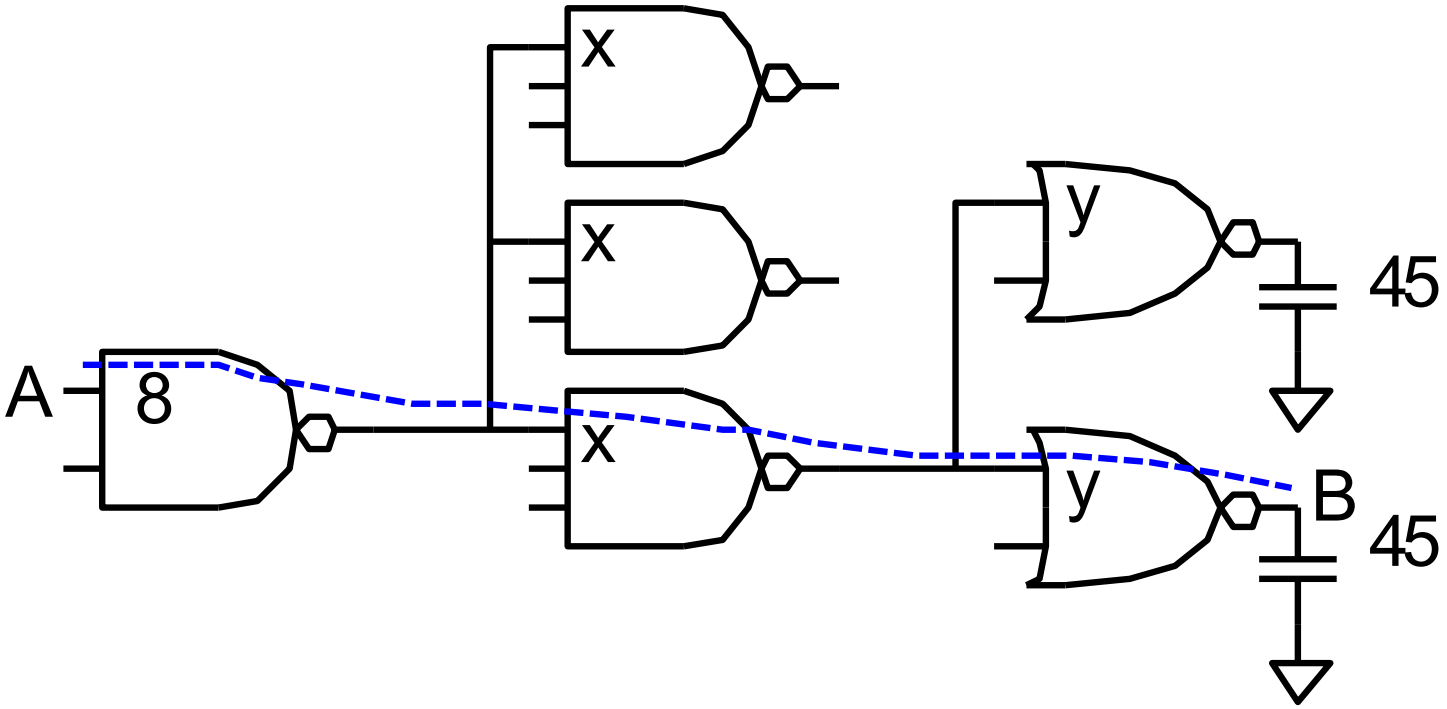
$$\Rightarrow C_{in_i} = \frac{g_i C_{out_i}}{\hat{f}}$$

- 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.

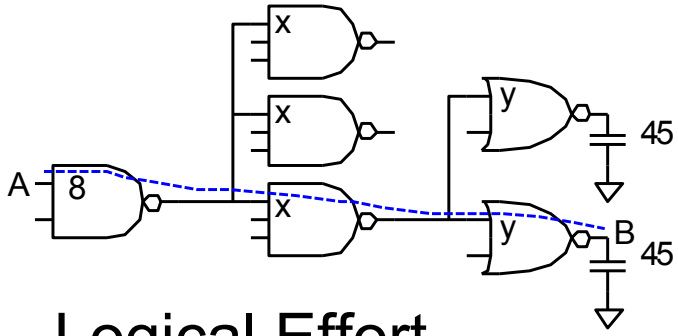
Example: 3-stage path

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

要得到最小的延遲 從a到b x和y要多大

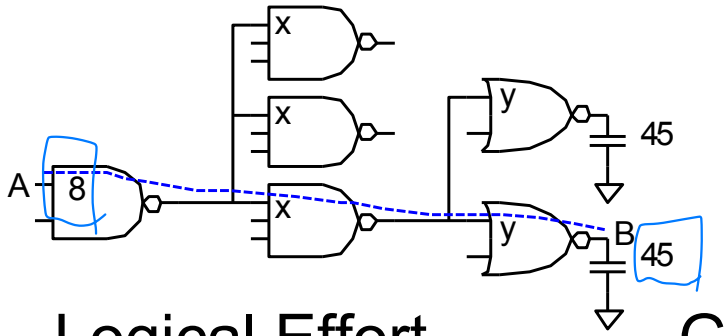


Example: 3-stage path



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



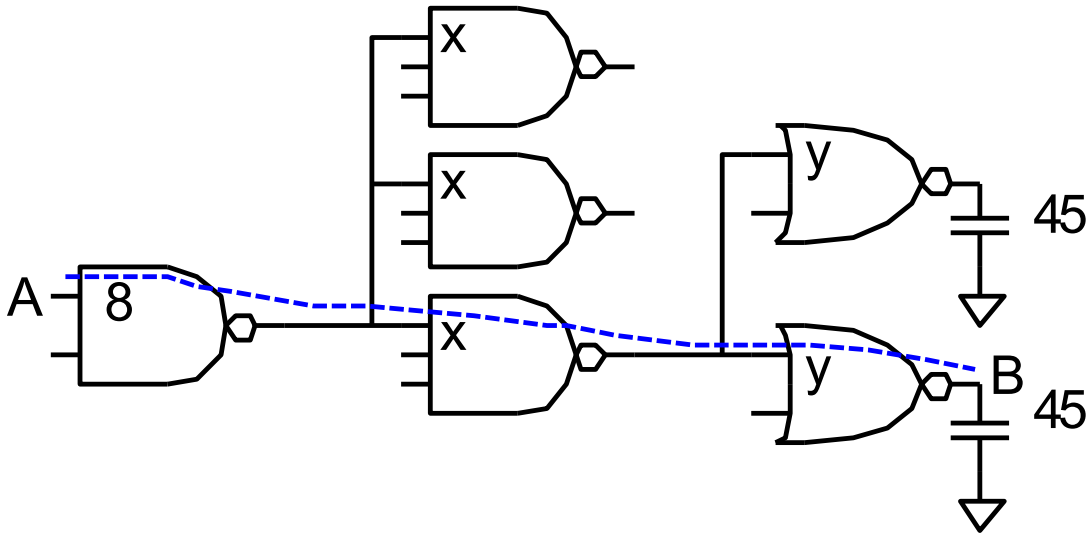
Logical Effort	G = (4/3)*(5/3)*(5/3) = 100/27
Electrical Effort	H = 45/8
Branching Effort	B = 3 * 2 = 6
Path Effort	F = GBH = 125
Best Stage Effort	$\hat{f} = \sqrt[3]{F} = 5$
Parasitic Delay	P = 2 + 3 + 2 = 7
Delay	D = 3*5 + 7 = 22 = 4.4 FO4

Example: 3-stage path

- Work backward for sizes

  y =

  x =

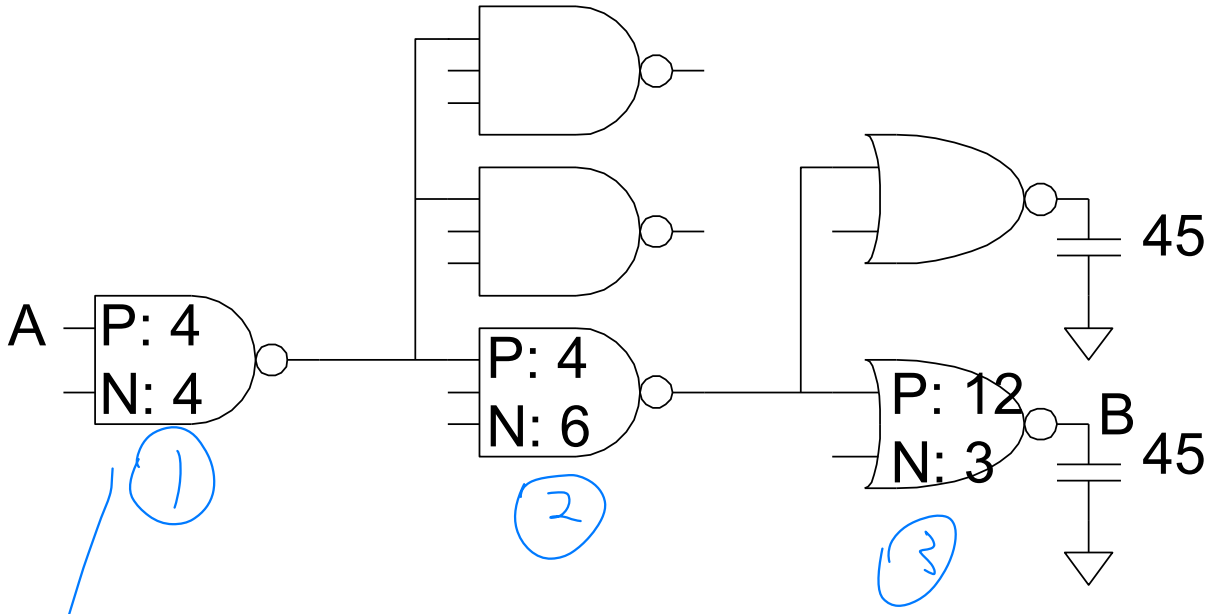


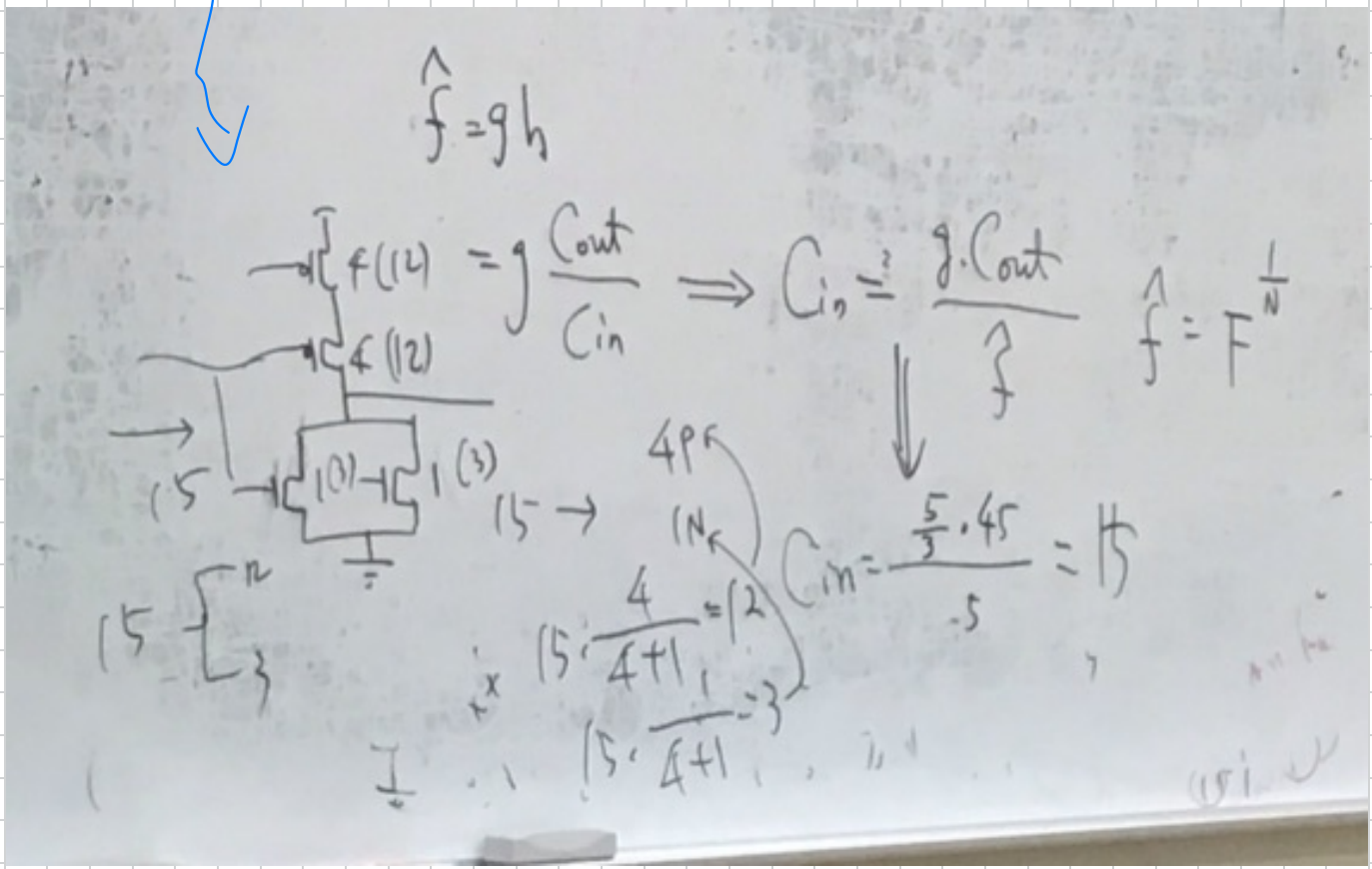
Example: 3-stage path

- Work backward for sizes

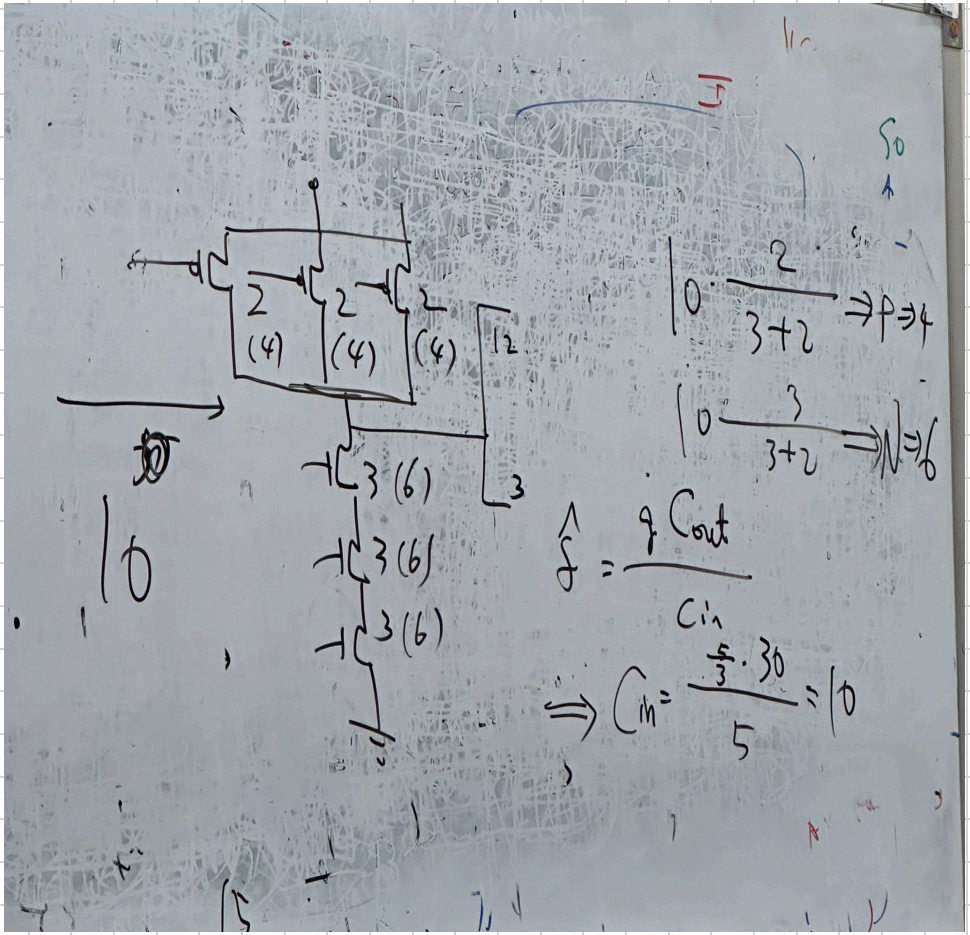
  y = 45 * (5/3) / 5 = 15

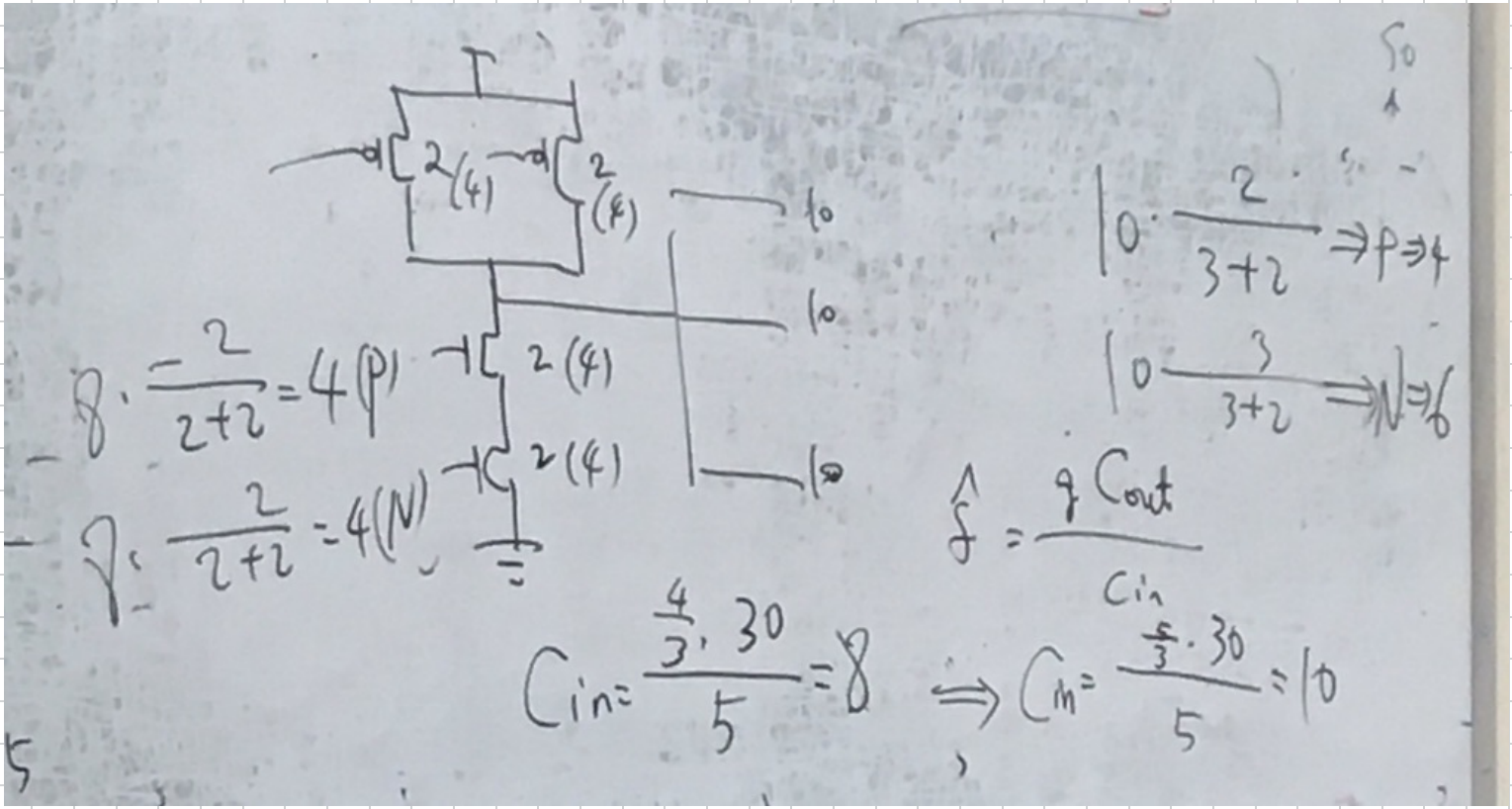
  x = (15*2) * (5/3) / 5 = 10




$\hat{f} = gh$
$f(14) = g\frac{Cout}{Cin} \Rightarrow Cin = \frac{g.Cout}{\hat{f}}$
$\hat{f} = F^{\frac{1}{N}}$
f(12)
10
1
3
15
15 →
4pF
$Cin = \frac{\frac{5}{3}.45}{5} = 15$
$15\frac{4}{4+1} = 12$
$15\frac{1}{4+1} = 3$

(2)

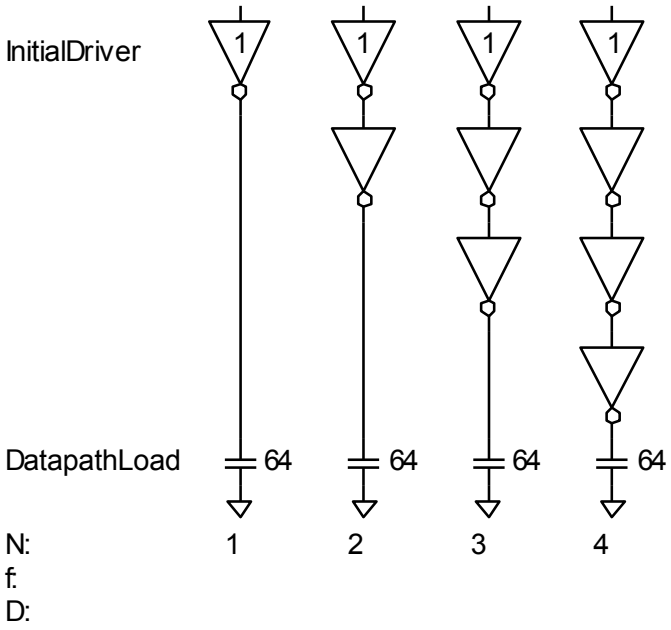

2
(4)
2
(4)
2
(4)
12
3 (6)
3 (6)
3 (6)
3
10
$\hat{f} = \frac{g C_{out}}{C_{in}}$
$\Rightarrow C_{in} = \frac{\frac{5}{3} \cdot 30}{5} = 10$
$10 \xrightarrow{\frac{2}{3+2}} 4 \Rightarrow d \Rightarrow 4$
$10 \xrightarrow{\frac{3}{3+2}} 6 \Rightarrow N \Rightarrow 6$


2 (4)
2 (4)
2 (4)
2 (4)
10
10
10
8: 2/(2+2) = 4 (P)
7: 2/(2+2) = 4 (N)
10: 2/(3+2) ⇒ P ⇒ 4
10: 3/(3+2) ⇒ N ⇒ 6
f = q Cout / Cin
Cin = (4/3 · 30)/5 = 8 ⇒ Cin = (5/3 · 30)/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

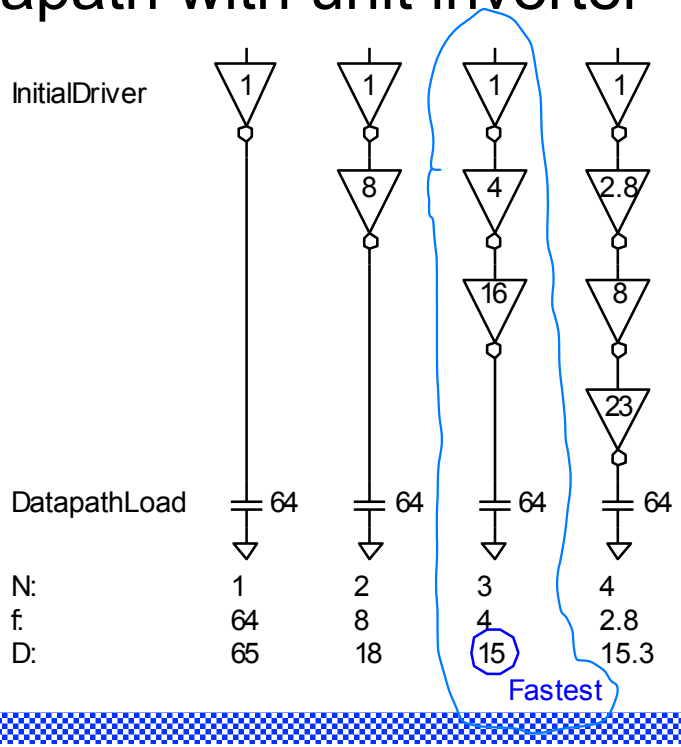
D =



Best Number of Stages

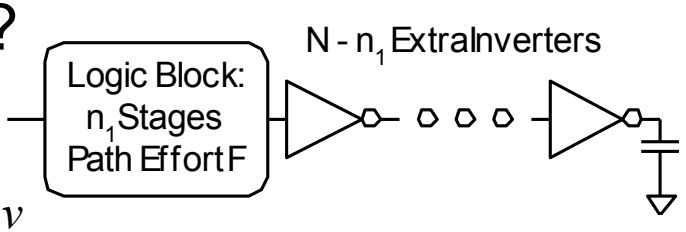
- How many stages should a path use?
  - Minimizing number of stages is not always fastest
- Example: drive 64-bit datapath with unit inverter

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



Derivation

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

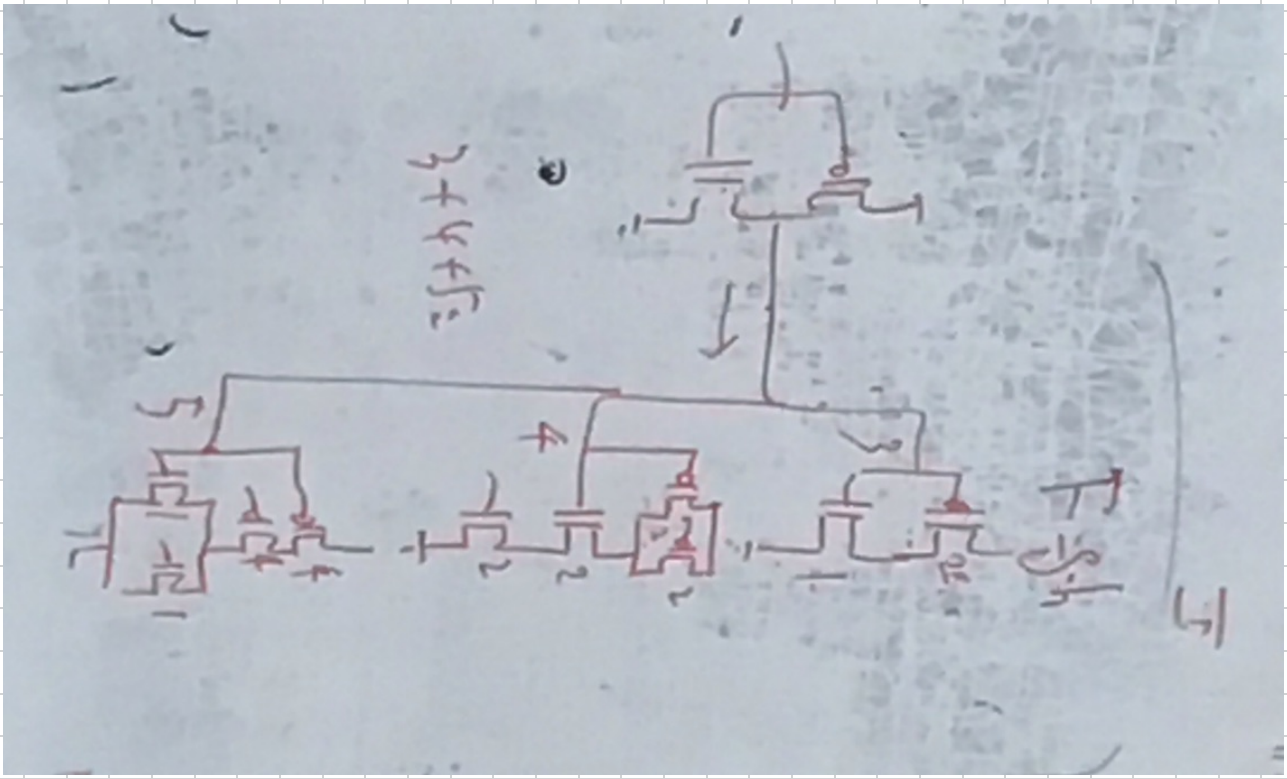


$$D = NF^{\frac{1}{N}} + \sum_{i=1}^{n_1} p_i + \left(N - n_1\right) p_{inv}$$

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

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

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



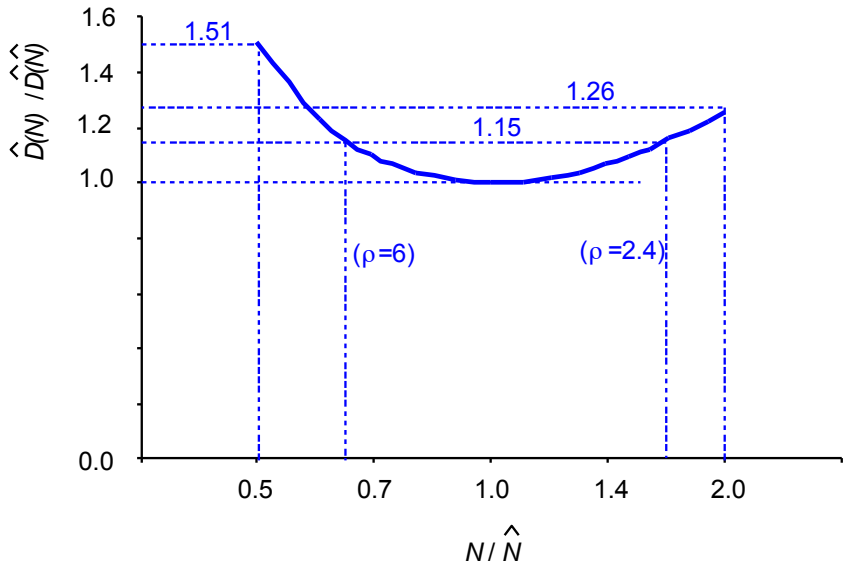
Best Stage Effort

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

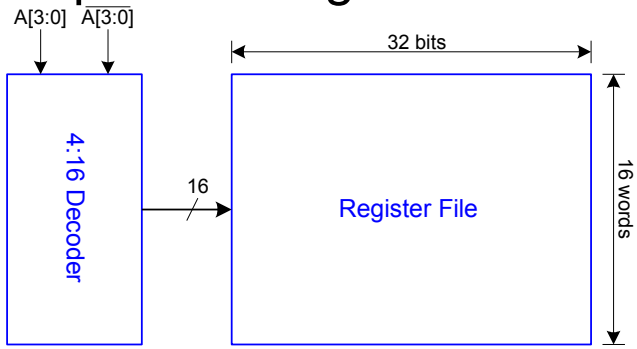
- How sensitive is delay to using exactly the best number of stages?



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

Example, Revisited

- 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?

Number of Stages

- Decoder effort is mainly electrical and branching

  Electrical Effort: H =

  Branching Effort: B =

- If we neglect logical effort (assume G = 1)

  Path Effort: F =

  Number of Stages: N =

Number of Stages

- Decoder effort is mainly electrical and branching

  Electrical Effort: H = (32*3) / 10 = 9.6

  Branching Effort: B = 8

- If we neglect logical effort (assume G = 1)

  Path Effort: F = GBH = 76.8

  Number of Stages: $N = \log_4 F = 3.1$

- Try a 3-stage design

Gate Sizes & Delay

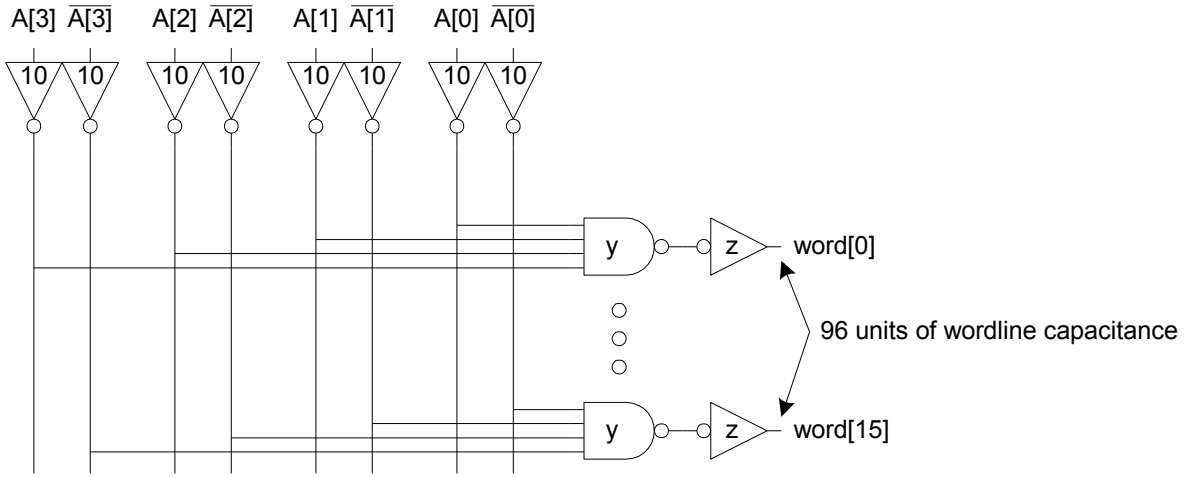
Logical Effort: G =

Path Effort: F =

Stage Effort: $\hat{f}$ =

Path Delay: D =

Gate sizes: z = y =



Gate Sizes & Delay

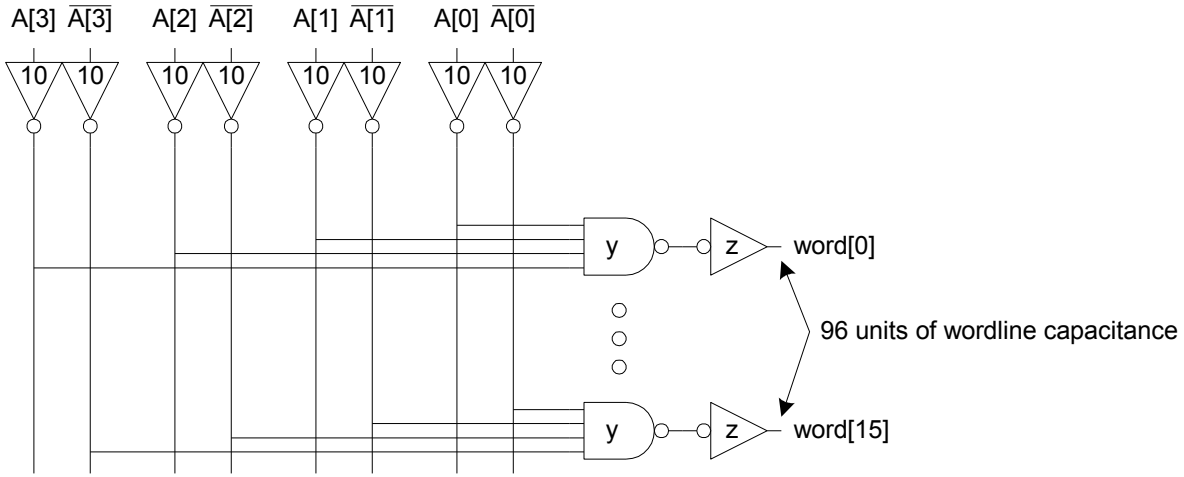
Logical Effort: G = 1 * 6/3 * 1 = 2

Path Effort: F = GBH = 154

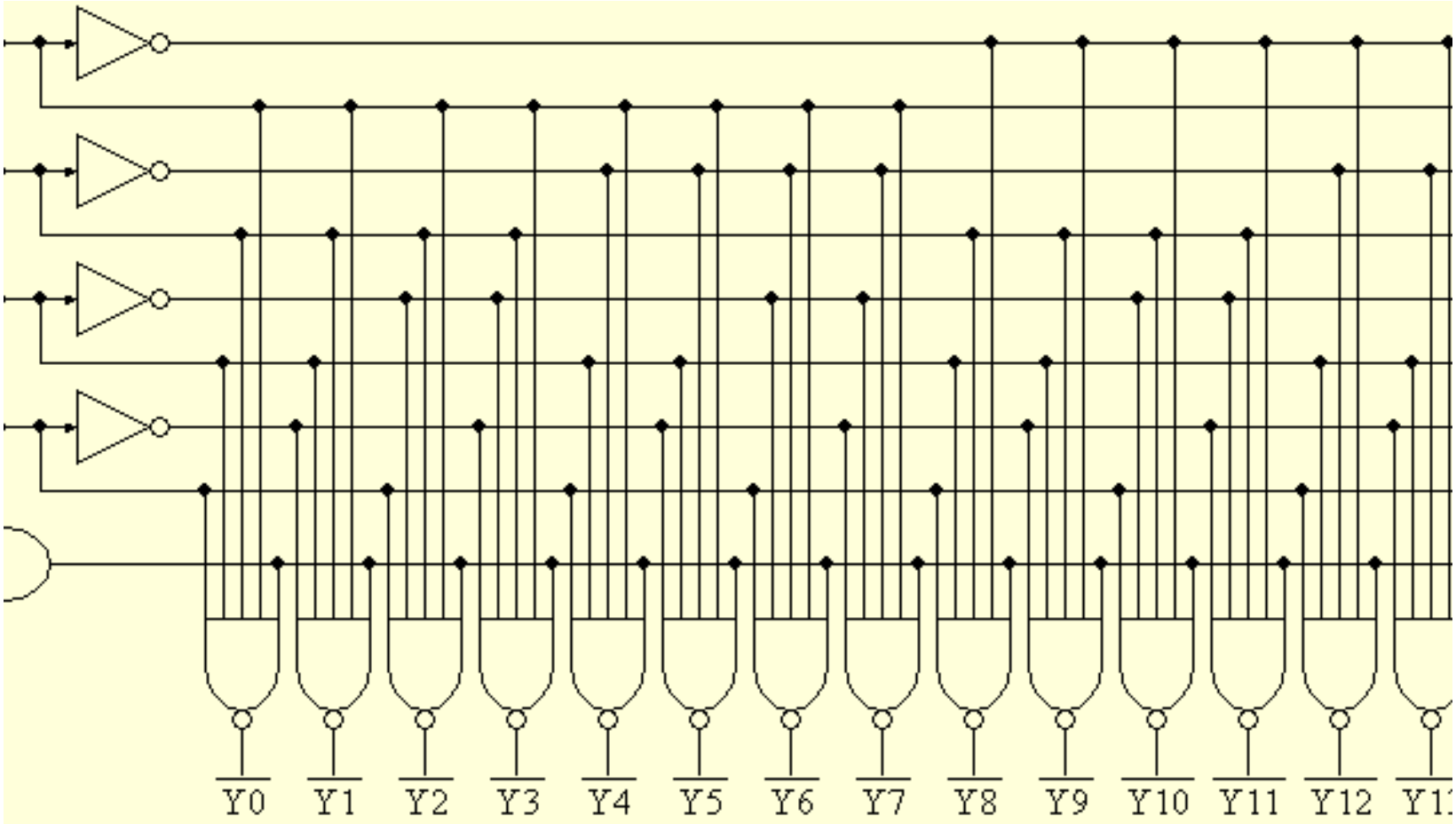
Stage Effort: $\hat{f} = F^{1/3} = 5.36$

Path Delay: $D = 3\hat{f} + 1 + 4 + 1 = 22.1$

Gate sizes: z = 96*1/5.36 = 18 y = 18*2/5.36 = 6.7



74154 Decoder



Comparison

- Compare many alternatives with a spreadsheet

Design	N	G	P	D
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	$h = \frac{C_{out}}{C_{in}}$	$H = \frac{C_{out\text{-}path}}{C_{in\text{-}path}}$
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 $F = GBH$
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 $C_{in_i} = \frac{g_i C_{out_i}}{\hat{f}}$

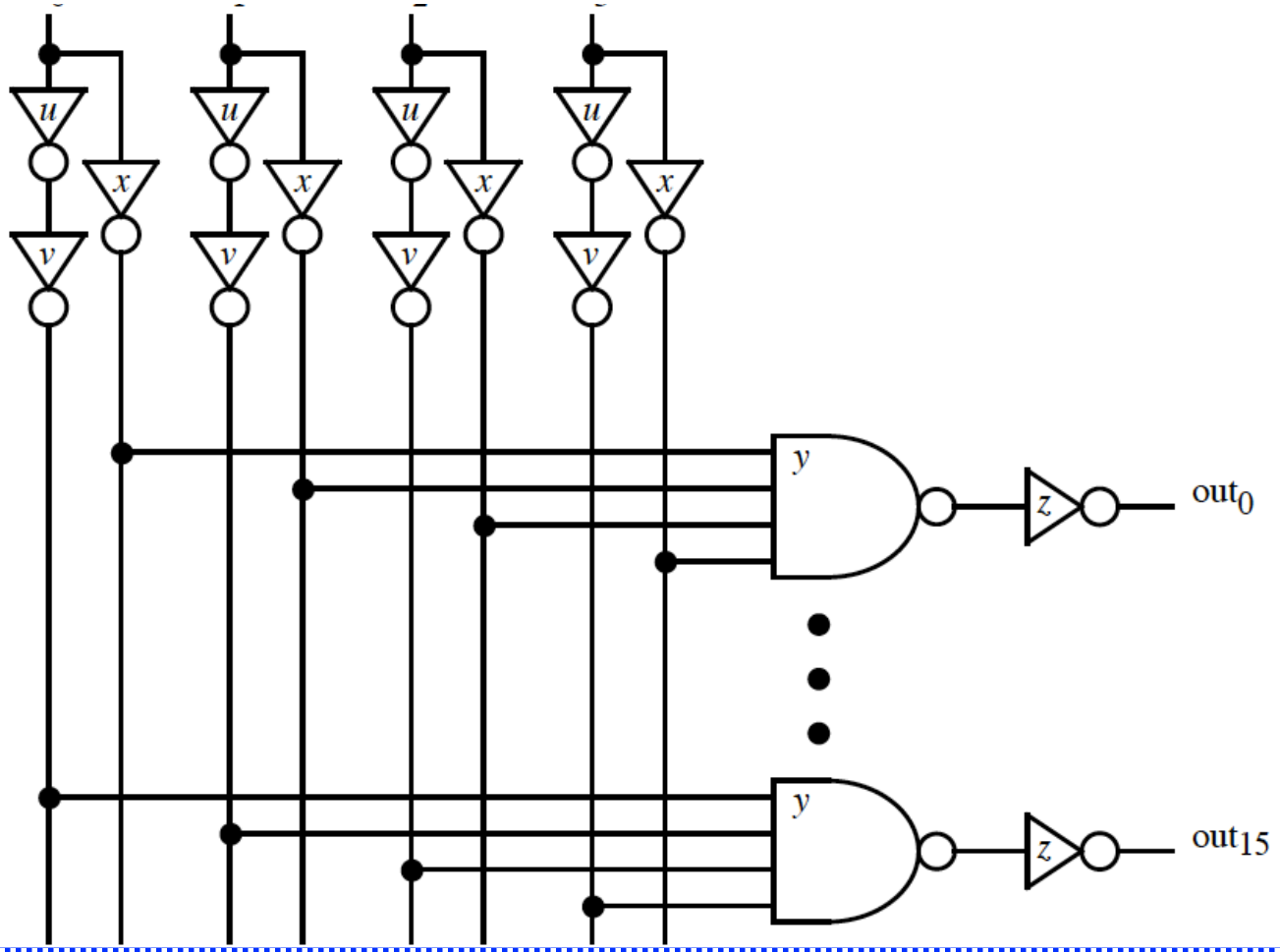
Limits of Logical Effort

- 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

Decoder with one polarity of input



A path with different logical and electrical efforts

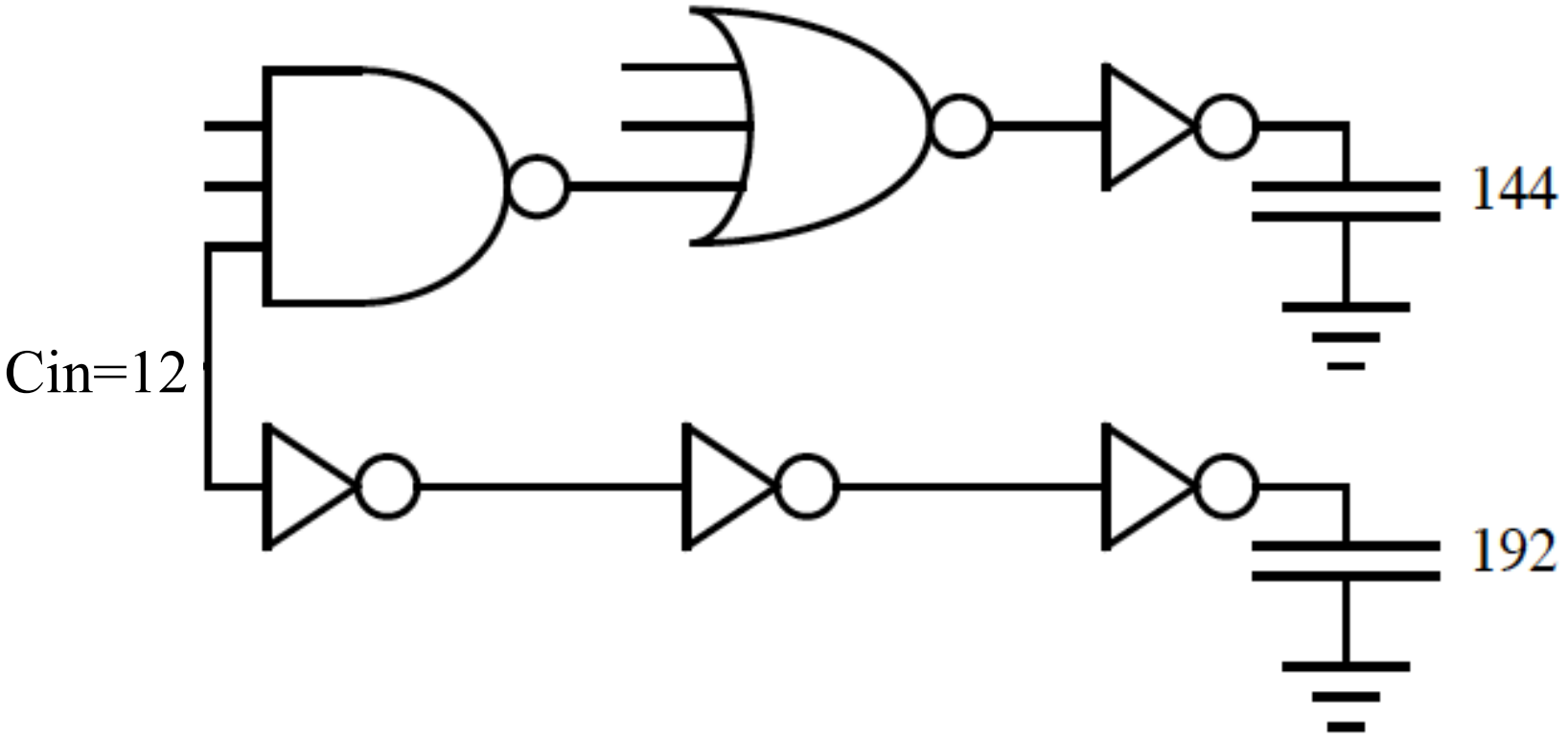
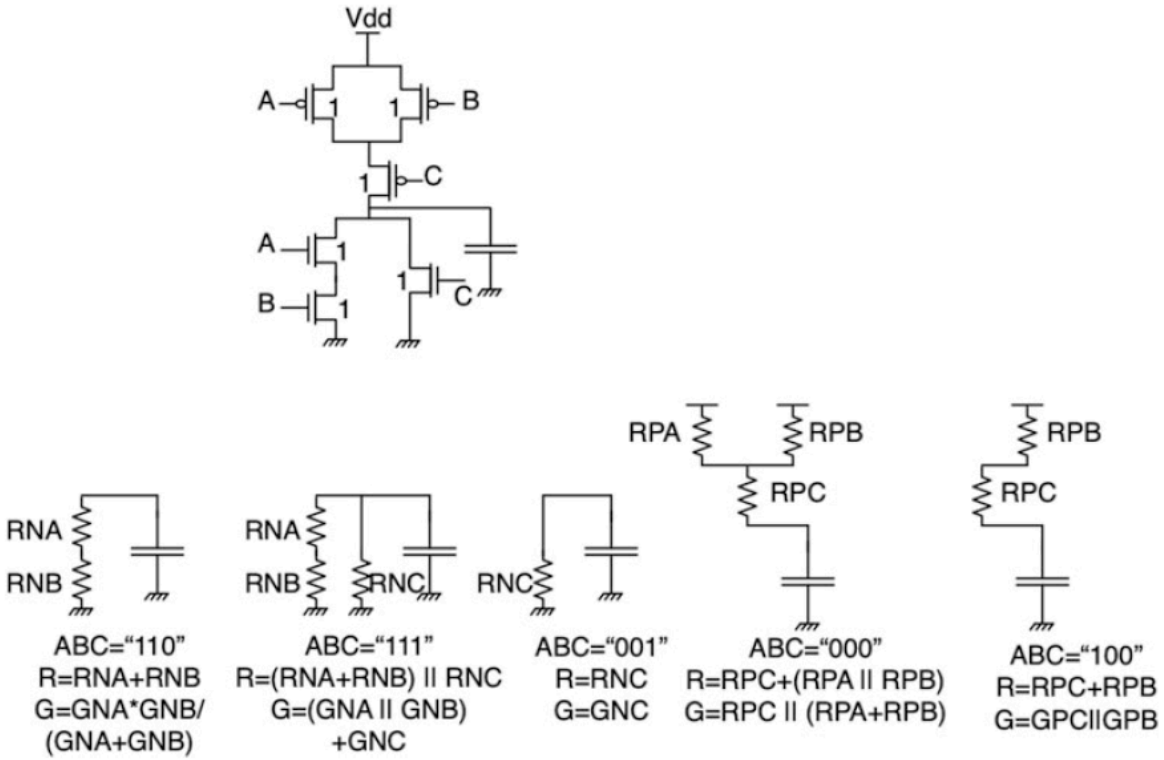


Fig. 3.29 Resistance for inputs "110", "111", "001", "000", "100"



mate of the path effort by assuming the logical effort is unity. The electrical effort is $32 \times 3/10 = 9.6$. The branching effort is 8 because the true and complement address inputs each control half of the outputs. Path effort is $9.6 \times 8 = 76.8$. Hence, we should use about $\log_4 76.8 = 3.1$ stages. Because we neglected logical effort, the actual number of stages will be slightly higher than the number we have estimated. Figure 2.2 shows a three-stage circuit, while Exercise 2.3 considers a four-stage circuit.

The three-stage circuit uses 16 four-input NAND gates. Because each address input must drive eight of the NAND gates, yet can handle only a relatively small input capacitance, we use an inverter to power up the signal. How do we size the decoder, and what is its delay?

Because the logical effort is $1 \times 2 \times 1 = 2$, the actual path effort is 154 and the stage effort is $f = (154)^{1/3} = 5.36$. Working from the output, the final inverter must have input capacitance $z = (32 \times 3) \times 1/5.36 = 18$, and the NAND gate must have input capacitance $y = 18 \times 2/5.36 = 6.7$. The delay is $3f + P =$