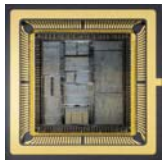
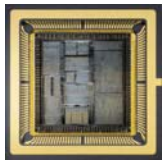


Lecture 7 : Logical Effort

CSCI 5330
Digital CMOS VLSI Design

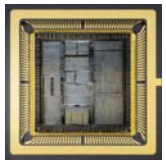
Instructor: Saraju P. Mohanty, Ph. D.

NOTE: The figures, text etc included in slides are borrowed from various books, websites, authors pages, and other sources for academic purpose only. The instructor does not claim any originality.



Outline

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

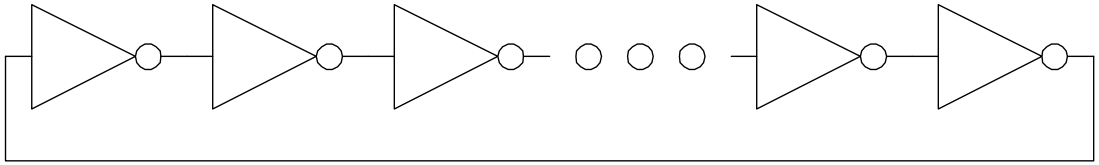


Introduction

- 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

- Example : Estimate the frequency of an N-stage ring oscillator



Logical Effort: g = 1

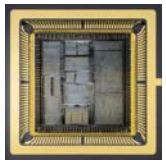
Electrical Effort: h = 1

Parasitic Delay: p = 1

Stage Delay: d = 2

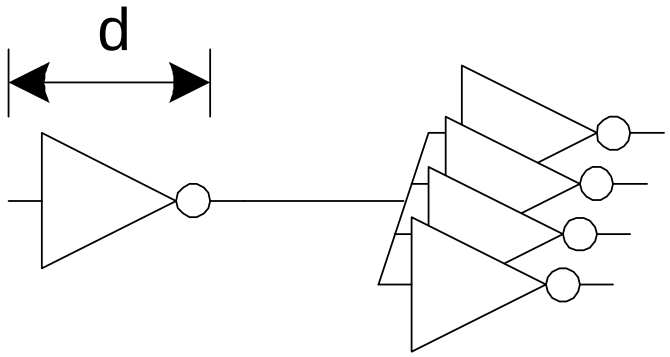
Frequency: $f_{osc} = 1/(2*N*d) = 1/4N$

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



Delay in a Logic Gate

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



Logical Effort: g = 1

Electrical Effort: h = 4

Parasitic Delay: p = 1

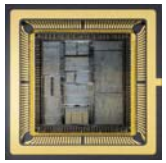
Stage Delay: d = 5

The FO4 delay is about

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*

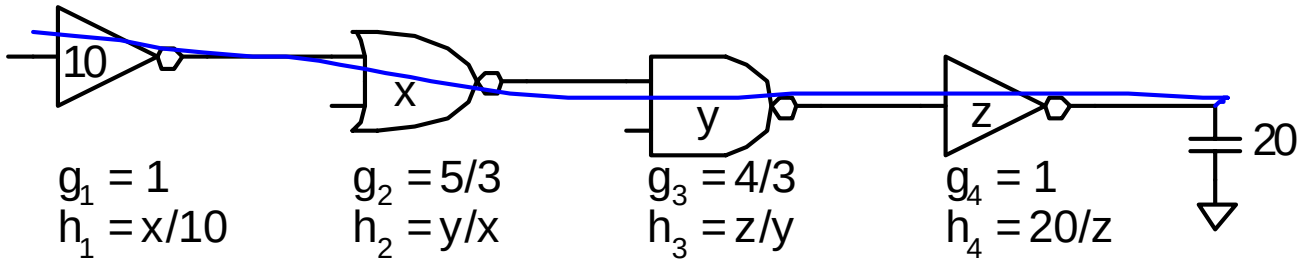
$$G = \prod g_i$$

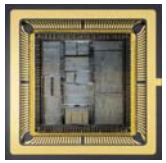
- *Path Electrical Effort*

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

- *Path Effort*

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





Multistage Logic Networks

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

$$G = \prod g_i$$

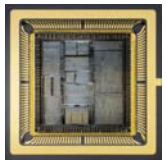
- *Path Electrical Effort*

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

- *Path Effort*

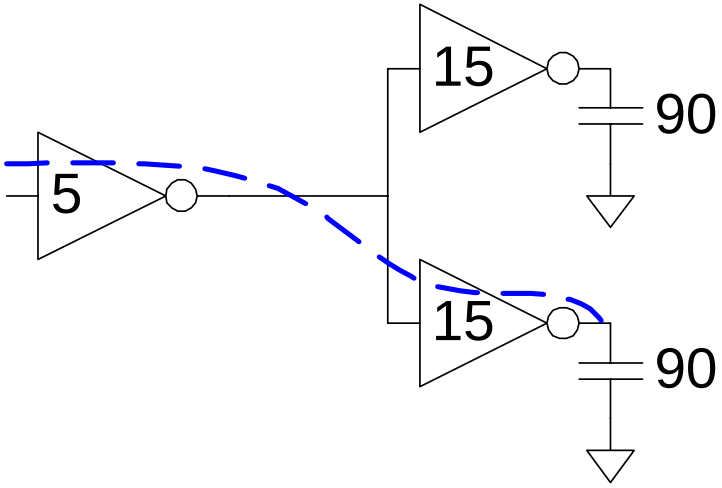
$$F = \prod f_i = \prod g_i h_i$$

- Can we write F = GH?



Multistage Logic Networks : Paths that Branch

- No! Consider paths that branch:



G = 1

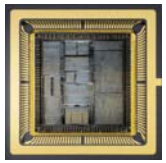
H = 90 / 5 = 18

GH = 18

h_1 = (15 +15) / 5 = 6

h_2 = 90 / 15 = 6

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



Multistage Logic Networks : Branching Effort

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

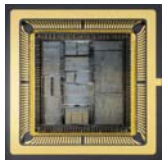
$$b = \frac{C_{\text{on path}} + C_{\text{off path}}}{C_{\text{on path}}}$$

$$B = \prod b_i$$

Note:

$$\prod h_i = BH$$

- Now we compute the path effort
  - F = GBH



Multistage Logic Networks : Multistage Delays

- Path Effort Delay

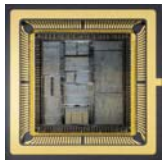
$$D_F = \sum f_i$$

- Path Parasitic Delay

$$P = \sum p_i$$

- Path Delay

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



Multistage Logic Networks : Designing Fast Circuits

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

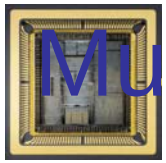
- Delay is smallest when each stage bears same effort

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

- 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



Multistage Logic Networks : Gate Sizes

- How wide should the gates be for least delay?

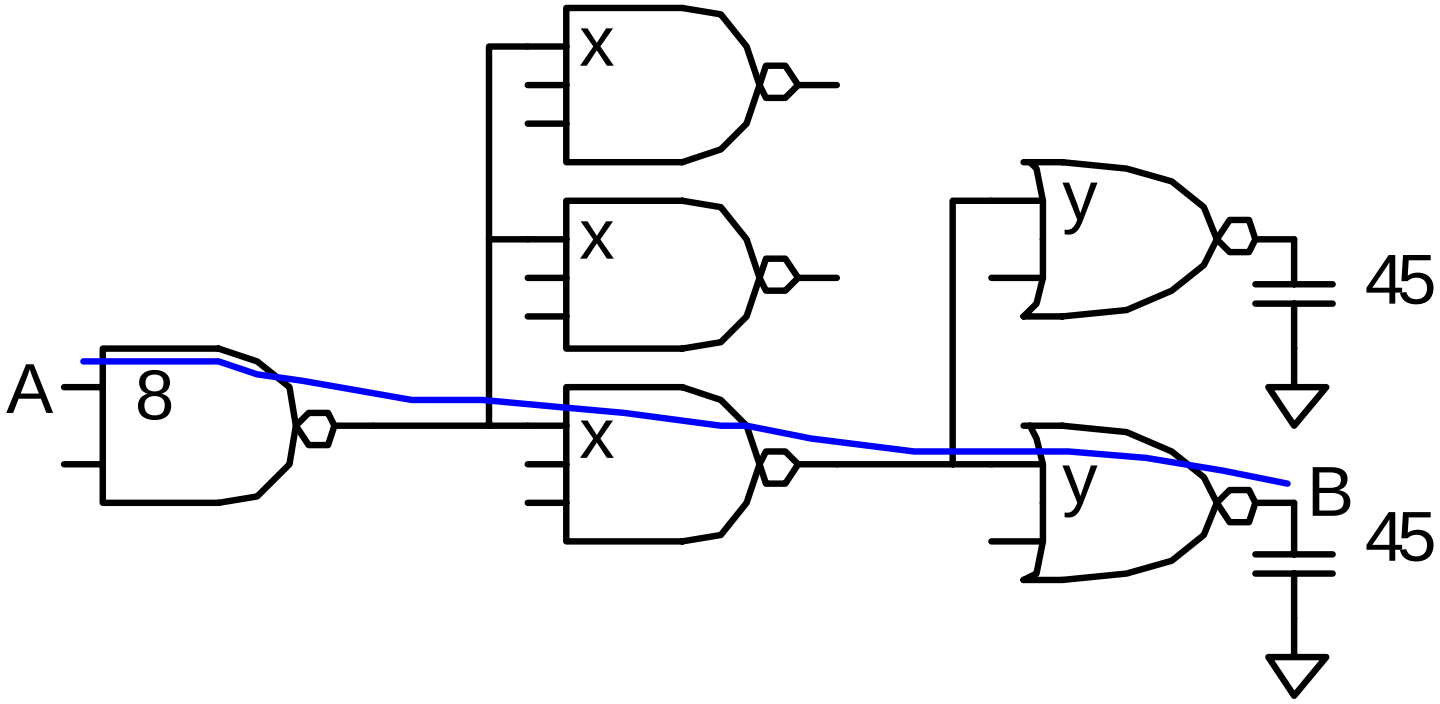
$$\hat{f} = gh = g\frac{C_{out}}{C_{in}}$$

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

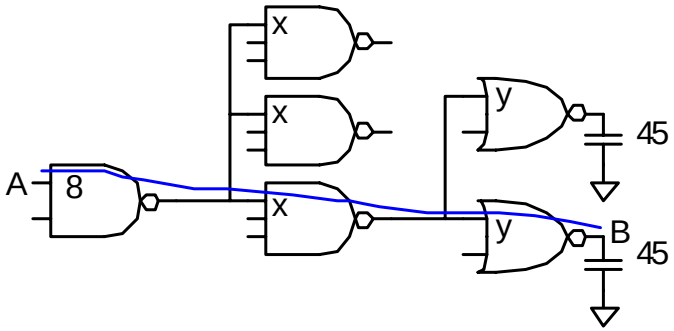
- Working backward, apply capacitance transformation to find input capacitance of each gate given load it drives.
- Check work by verifying input cap spec is met.

Multistage Logic Networks : 3-stage path

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



Multistage Logic Networks : 3-stage path Example



Logical Effort G = (4/3)*(5/3)*(5/3) = 100/27

Electrical Effort H = 45/8

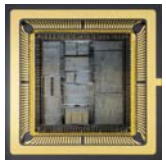
Branching Effort B = 3 * 2 = 6

Path Effort F = GBH = 125

Best Stage Effort $\hat{f} = \sqrt[3]{F} = 5$

Parasitic Delay P = 2 + 3 + 2 = 7

DelayD = 3*5 + 7 = 22 = 4.4 FO4

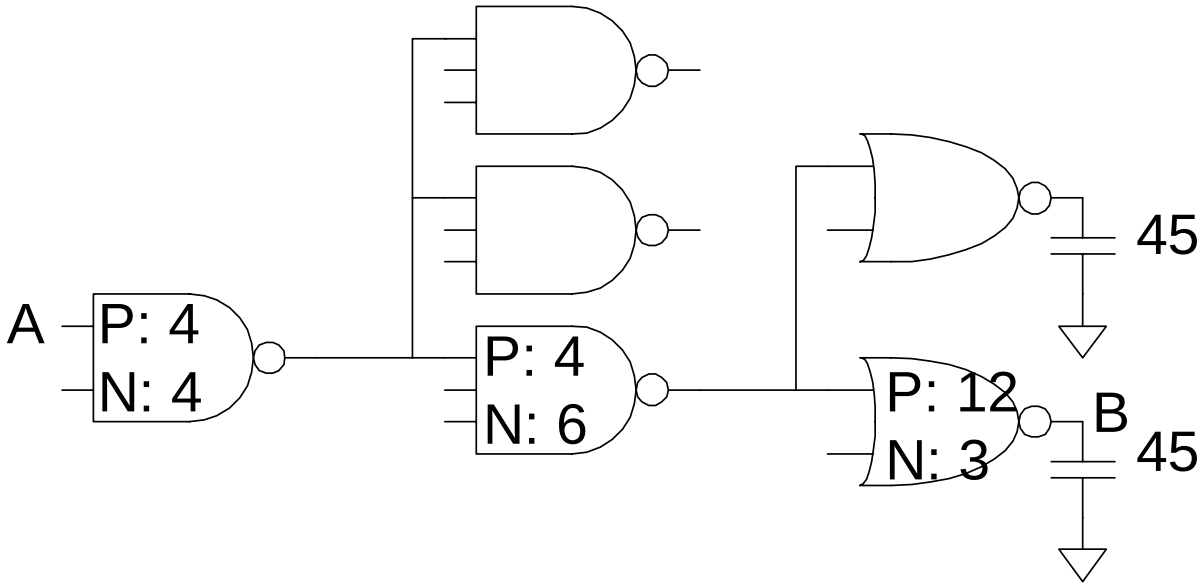


Multistage Logic Networks : 3-stage path Example

- Work backward for sizes

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

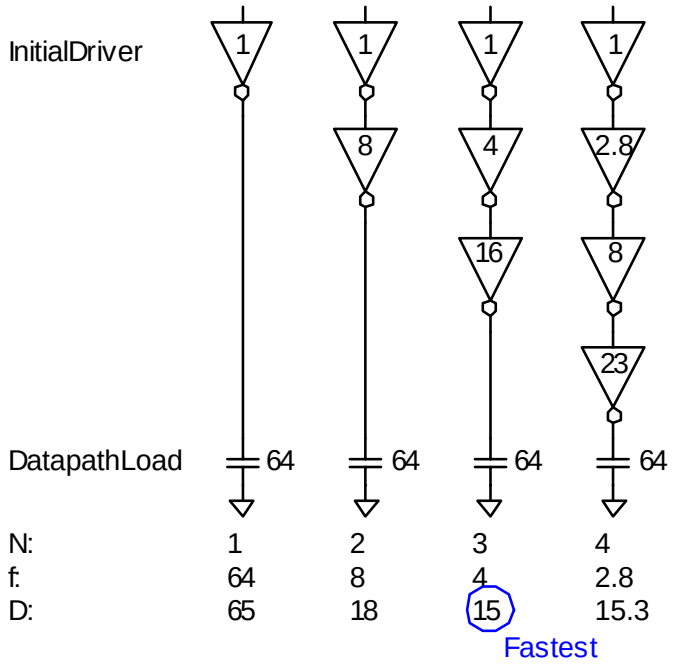
  x = (15*2) * (5/3) / 5 = 10



Choosing Best Number of Stages

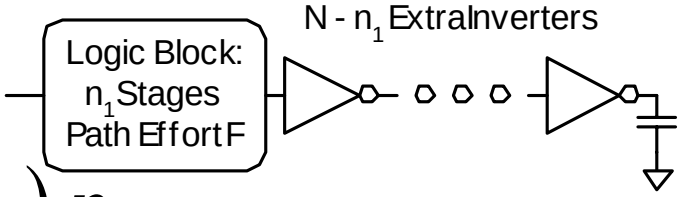
- How many stages should a path use?
  - Minimizing number of stages is not always fastest
- Example: drive 64-bit datapath with unit inverter

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



Choosing Best Number of Stages

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



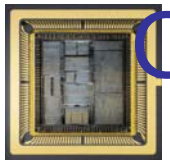
$$D = NF^{\frac{1}{N}} + \sum_{i=1}^{n_1} p_i + \left(N - n_1\right) p_{inv}$$

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

- Define best stage effort

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

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



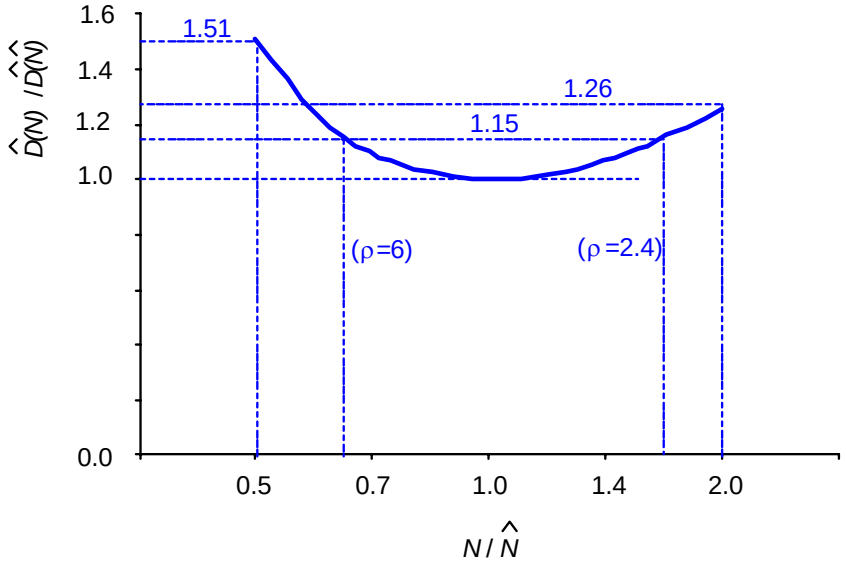
Choosing Best Number of Stages : Best Stage Effort

- $p_{inv} + \rho(1 - \ln \rho) = 0$ has no closed-form solution

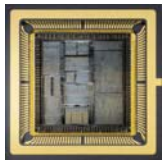
- Neglecting parasitics ($p_{inv} = 0$), we find $\rho = 2.718$ (e)
- For $p_{inv} = 1$, solve numerically for $\rho = 3.59$

Choosing Best Number of Stages : Sensitivity Analysis

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



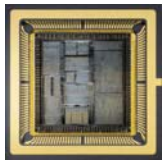
- $2.4 < \rho < 6$ gives delay within 15% of optimal
  - We can be sloppy!
  - I like $\rho = 4$



Spreadsheet Comparison

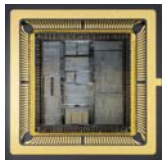
- Compare many alternatives with a spreadsheet

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



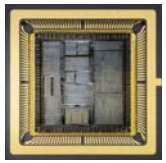
Review of Definitions

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



Limitations 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