CAL POLY
SAN LUIS OBISPO

EE 431: COMPUTER-AIDED DESIGN OF VLSI DEVICES

Logical Effort

Nishith N. Chakraborty

October, 2024

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

$$d = f + p$$

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

$$d = f + p$$

- 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 = 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

DELAY IN A LOGIC GATE

- Express delays in process-independent unit

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

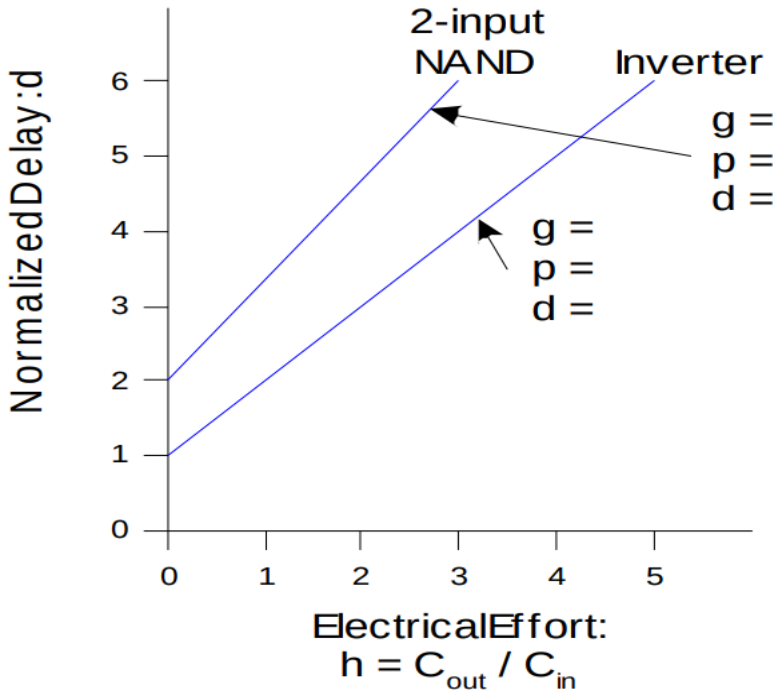
- Delay has two components

$$d = f + p$$

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

DELAY PLOTS

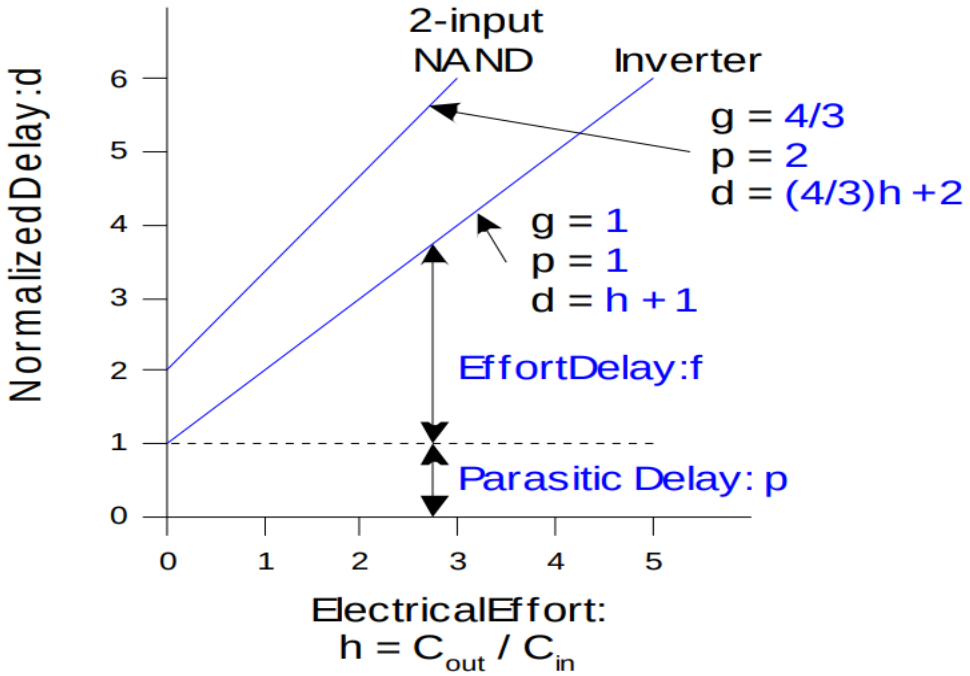
$$d = f + p$$
$$= gh + p$$



DELAY PLOTS

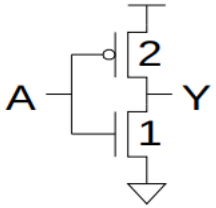
$$d = f + p$$
$$= gh + p$$

- Plot shows delay components for NAND2

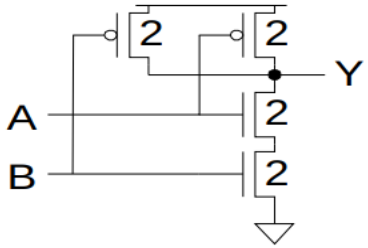


COMPUTING LOGICAL EFFORT

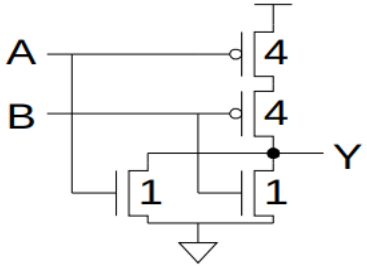
- DEF: *Logical effort is the ratio of the input capacitance of a gate to the input capacitance of an inverter delivering the same output current.*
- Measure from delay vs. fanout plots
- Or estimate by counting transistor widths



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



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



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

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	

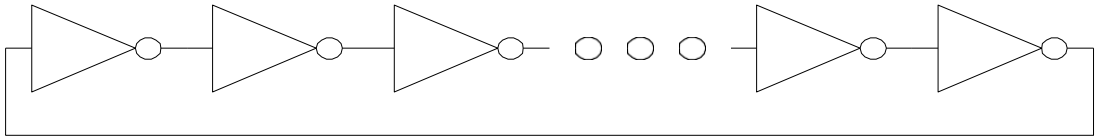
PARASITIC DELAY OF COMMON GATES

- 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 =

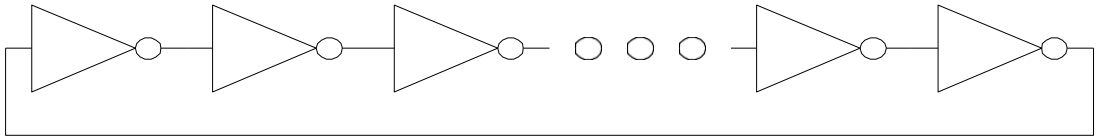
Parasitic Delay: p =

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

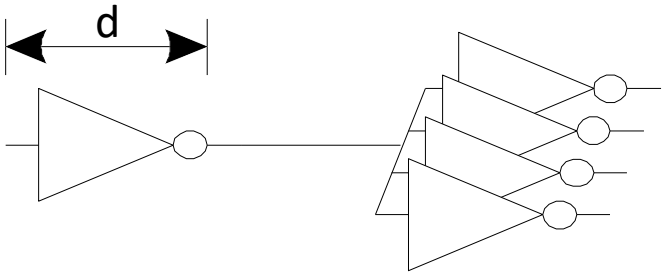
Stage Delay: d = 2

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

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

EXAMPLE: FO-4 INVERTER

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



Logical Effort: g =

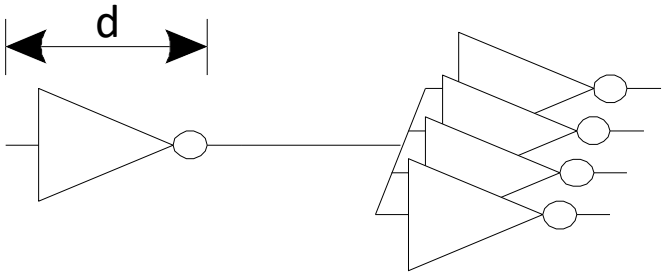
Electrical Effort: h =

Parasitic Delay: p =

Total Delay: d =

EXAMPLE: FO-4 INVERTER

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



Logical Effort: g = 1

Electrical Effort: h = 4

Parasitic Delay: p = 1

Total Delay: d = 5

The FO4 delay is about

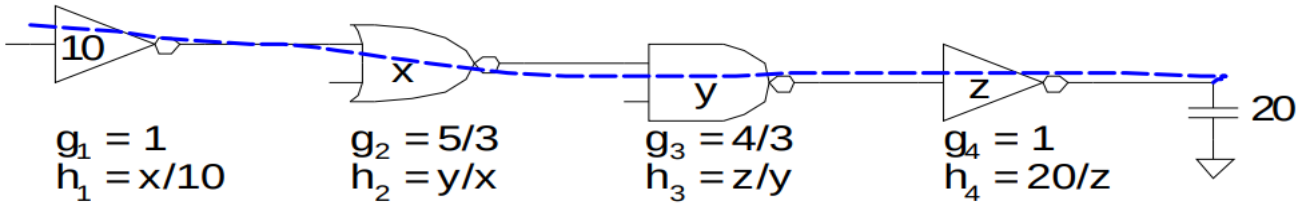
200 ps in 0.6 µm process

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

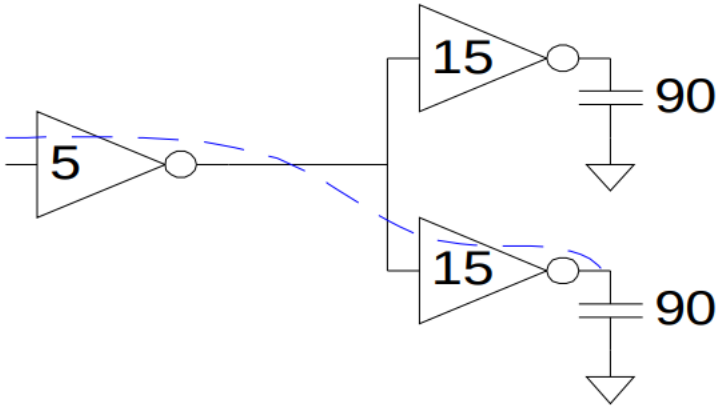
  H =

  GH =

  h_1 =

  h_2 =

  F = GH?



PATHS THAT BRANCH

- No! Consider paths that branch

  G = 1

  H = 90/5 = 18

  GH = 18

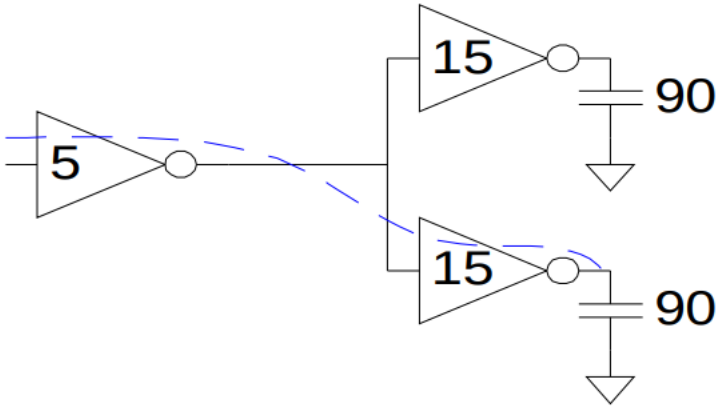
  g_1 = 1

  g_2 = 1

  h_1 = (15 +15) / 5 = 6

  h_2 = 90 / 15 = 6

  F = $g_1g_2h_1h_2$ = 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

GATE SIZES

- How wide should the gates be for least delay?

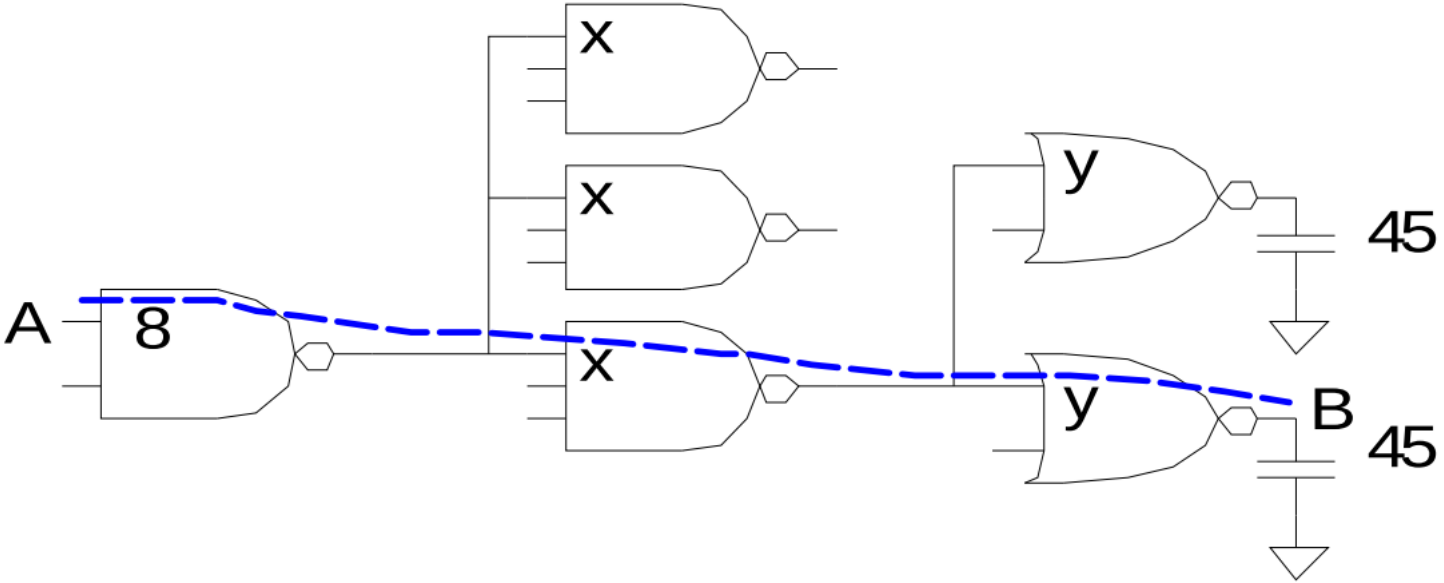
$$\hat{f} = gh = g\frac{C_{out}}{C_{in}}$$

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

- Working backward, apply capacitance transformation to find input capacitance of each 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 =

Electrical Effort H =

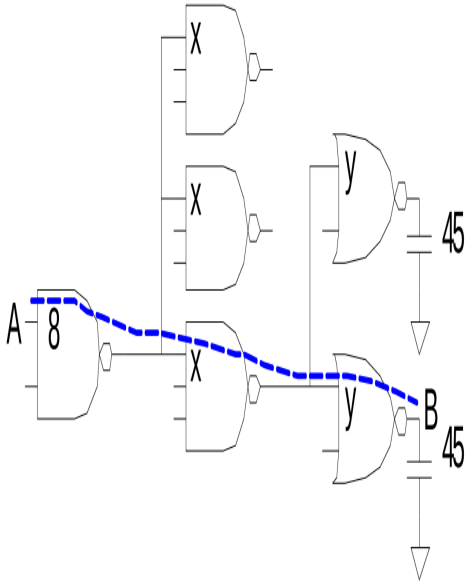
Branching Effort B =

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

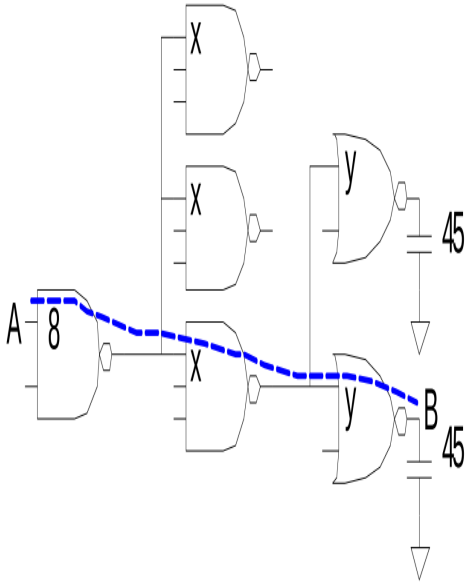
Branching Effort B = B = 3 * 2 = 6

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

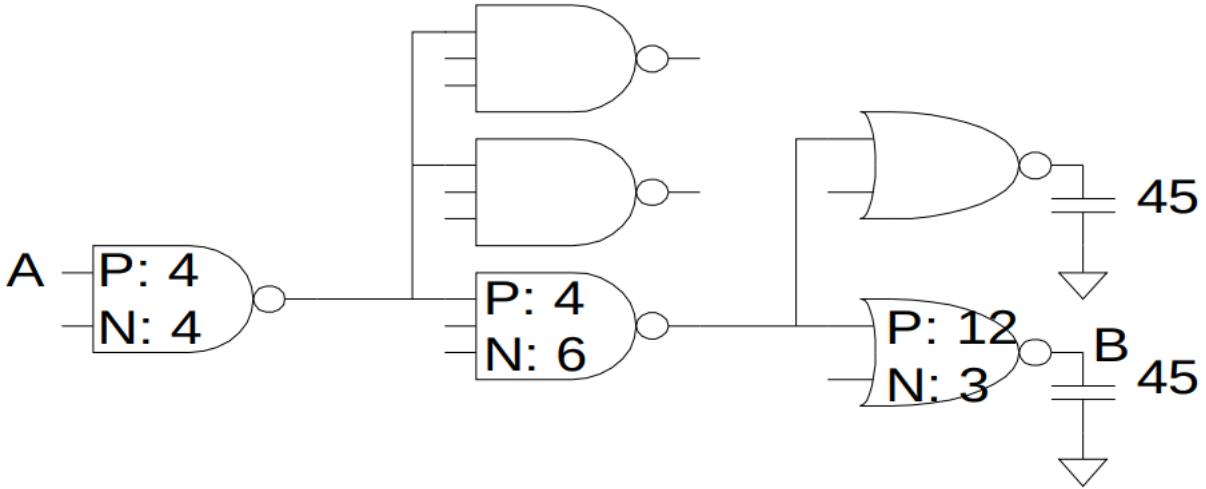


EXAMPLE: 3-STAGE PATH

- Work backward for sizes

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

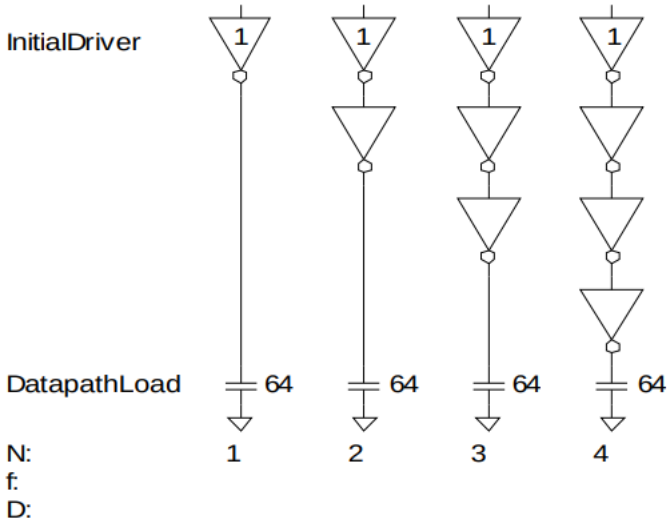
  x = (15*2) * (5/3) / 5 = 10



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 =

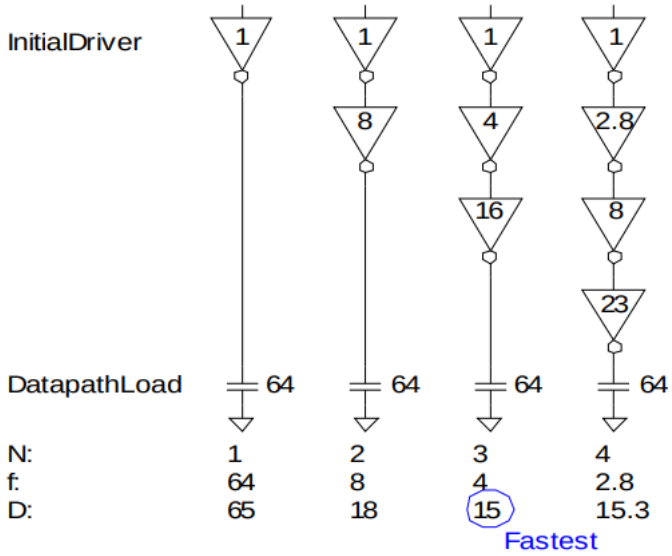


BEST NUMBER OF STAGES

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

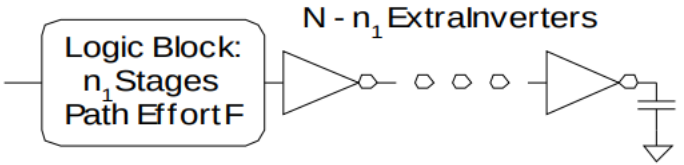
$D = NF^{1/N} + P$

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



DERIVATION

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



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

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

- Define best stage effort

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

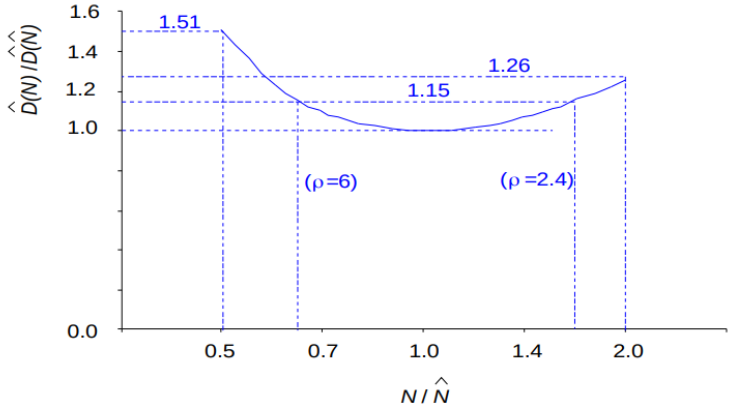
$$p_{inv} + \rho(1 - \ln \rho) = 0$$

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 $\rho = 3.59$
- Ideal number of stages: $\hat{N} = \log_{\rho} F$

SENSITIVITY ANALYSIS

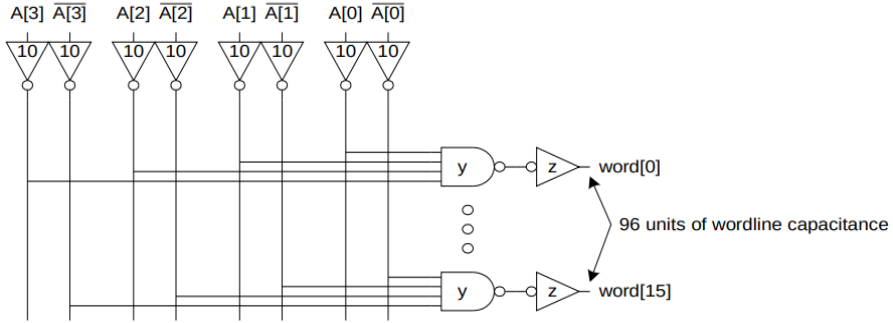
- How sensitive is delay to using exactly the best number of stages?



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

NUMBER OF STAGES: DECODER

- 16-word register file
- 32-bit words
- Each register bit presents a load of three unit-sized transistors on the word line (two unit-sized access transistors plus some wire capacitance)
- True and complementary versions of the address bits A[3:0] are available
- Each address input can drive 10 unit-sized transistors



NUMBER OF STAGES: DECODER

- A 2^N-word decoder consists of 2^N N-input AND gates.
- Therefore, the problem is reduced to designing a suitable 4-input AND gate.
- We need to know the path logical effort to calculate the path effort and best number of stages. However, without knowing the best number of stages, we cannot sketch a path and determine the logical effort for that path. There are two ways to resolve the dilemma. One is to sketch a path with a random number of stages, determine the path logical effort, and then use that to compute the path effort and the actual number of stages. The path can be redesigned with this number of stages, refining the path logical effort. If the logical effort changes significantly, the process can be repeated. Alternatively, we know that the logic of a decoder is rather simple, so we can ignore the logical effort (assume G = 1). Then we can proceed with our design, remembering that the best number of stages is likely slightly higher than predicted because we neglected logical effort.

NUMBER OF STAGES: DECODER

- Decoder effort is mainly electrical and branching

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

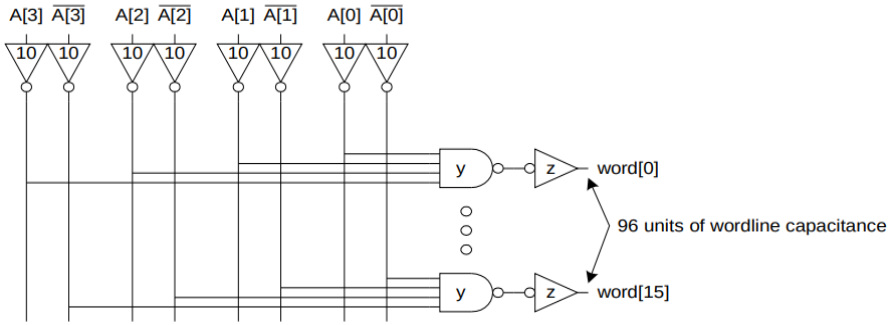
  Branching Effort: B = 8

- 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 AND DELAY

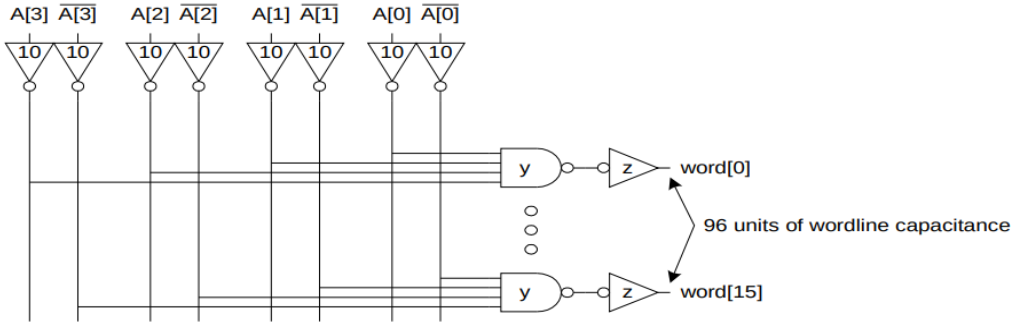
Logical Effort: G =

Path Effort: F =

Stage Effort: $\hat{f}$ =

Path Delay: D =

Gate sizes: z= y =



GATE SIZES AND DELAY

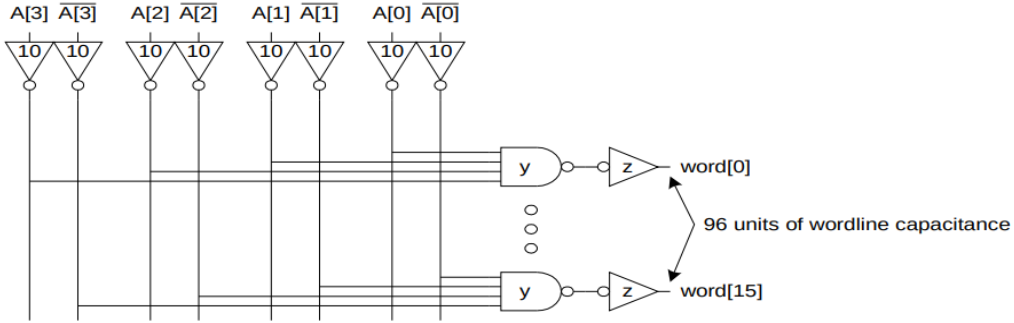
Logical Effort: G = 1 * 6/3 * 1 = 2

Path Effort: F = GBH = 154

Stage Effort: $\hat{f} = F^{1/3}$ =5.36 (Within 2.4< $\hat{f}$ <6, so good!)

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



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 — $D = NF^{1/N} + P$
5) Determine best stage effort — $\hat{f} = F^{1/N}$
6) Find gate sizes — $C_{ini} = g_i C_{outi} / \hat{f}$

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

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
  - Using fewer stages doesn't mean faster paths
  - Delay of path is about log4(F) FO4 inverter delays
  - Inverters and NAND2 best for driving large caps
- Provides language for discussing fast circuits
  - But requires practice to master

Thank you!