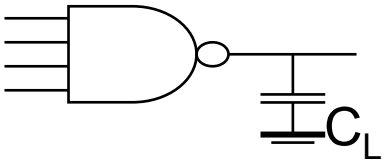
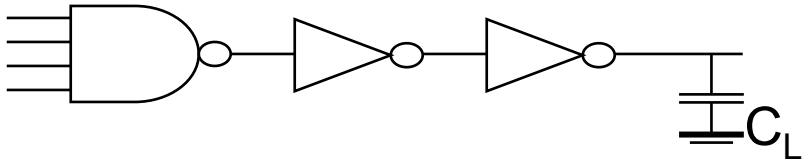
Logical Effort

Question #1

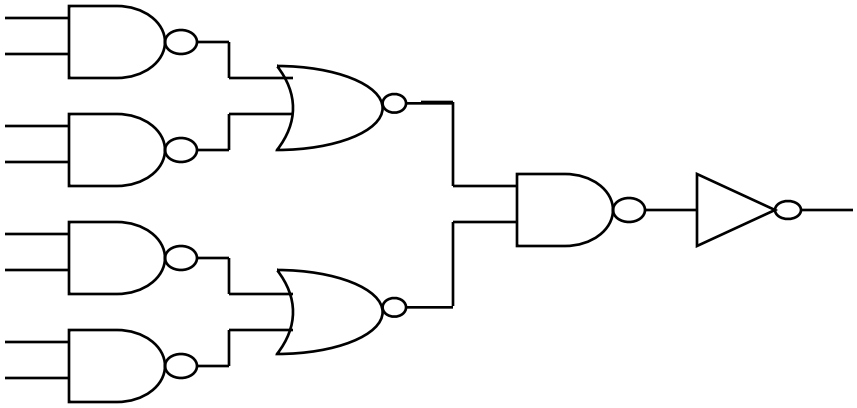
- How to best combine logic and drive for a big capacitive load?

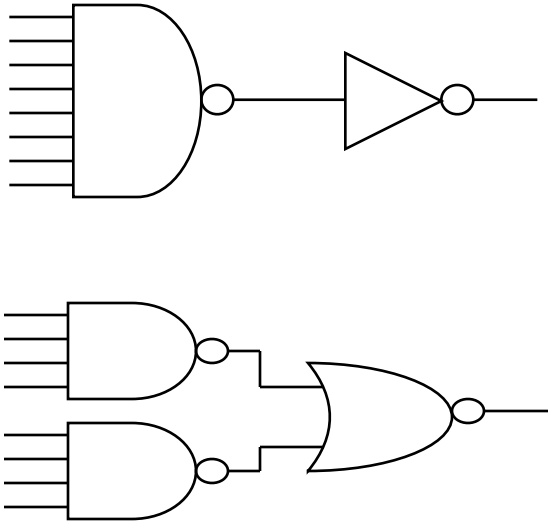




Question #2

- All of these are “decoders”
  - Which one is “best”?





Method to answer both of these questions

- Extension of buffer sizing problem
- Logical effort

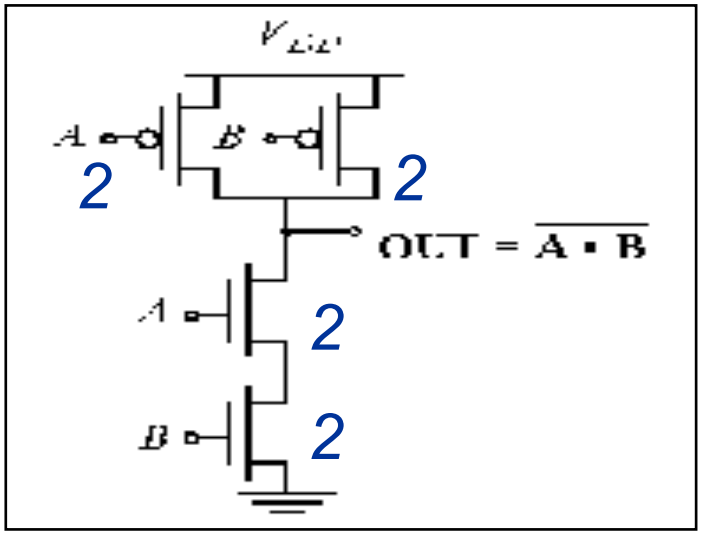
Complex Gate Sizing

Complex Gate Sizing: NAND-2 Example

$$C_{gnand} = 4C_G = (4/3)\, C_{ginv}$$

$$C_{dnand} = 6C_D = 6\gamma C_G = 2\gamma C_{ginv}$$

$$f = C_L/C_{gnand} = (3/4)\, C_L/C_{ginv}$$



$$t_{pNAND} = kR_N(C_{dnand} + C_L)$$

$$= kR_N(2\gamma C_{ginv} + C_L)$$

$$= kR_N C_{ginv}\,(2\gamma + C_L/C_{ginv})$$

$$= t_{inv}\,(2\gamma + (4/3)f)$$

Logical Effort

- Defines ease of gate to drive external capacitance
- Inverter has the smallest logical effort and intrinsic delay of all static CMOS gates
- Logical effort LE is defined as:
  - $(R_{eq,gate}C_{in,gate})/(R_{eq,inv}C_{in,inv})$
  - Easiest way to calculate (usually):
    - Size gate to deliver same current as an inverter, take ratio of gate input capacitance to inverter capacitance
- LE increases with gate complexity

Logical Effort

$$t_{pgate} = t_{inv}\,(p + LEf)$$

Measure everything in units of t_{inv} (divide by t_{inv}):

p – intrinsic delay - gate parameter ≠ f(W)
LE – logical effort – gate parameter ≠ f(W)
f – electrical fanout = C_L/C_{in} = f (W)

Normalize everything to an inverter:
LE_{inv} =1, p_{inv} = γ

Delay of a Logic Gate

Gate delay:

$$Delay = EF + p \quad \text{(measured in units of } t_{inv}\text{)}$$

effective fanout — intrinsic delay

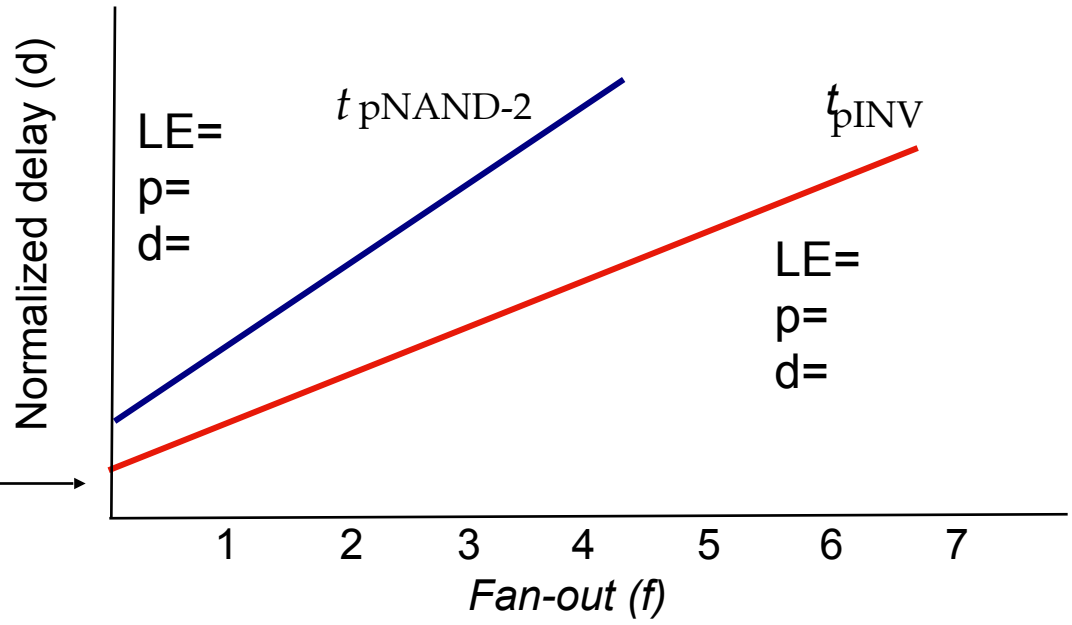
Effective fanout:

$$EF = LE\ f$$

logical effort — electrical fanout $= C_L/C_{in}$

Logical effort is a function of topology, independent of sizing
Effective fanout is a function of load/gate size

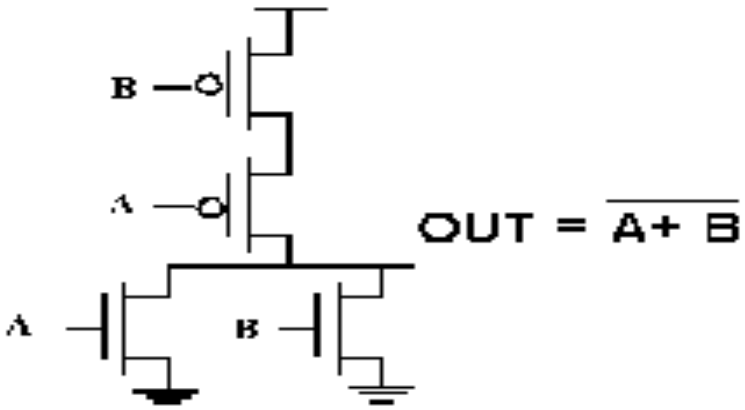
Logical Effort of Gates



p = γ·Fan-in
(for top input)

Delay Of NOR-2 Gate

1. Size for same resistance as inverter
2. LE = ratio of input cap of gate versus inverter



Intrinsic capacitance (C_{dnor}) =

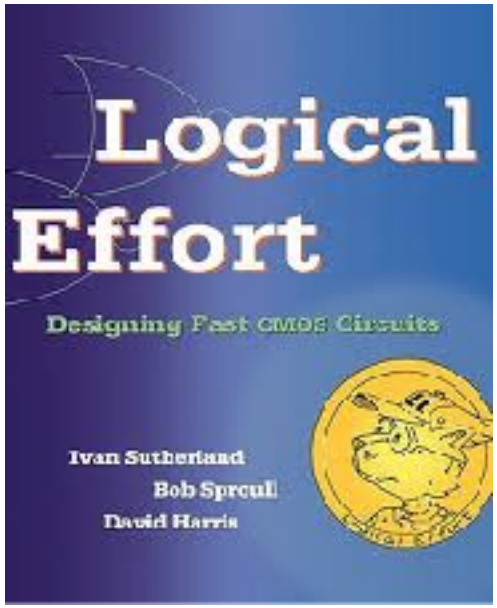
t_{pint} (NOR) =

Question

Any logic function can be implemented using NOR gates only or NAND gates only!

Which of the two approaches is preferable in CMOS (from a performance perspective)?

Logical Effort



Gate Type	Number of Inputs			
	1	2	3	n
Inverter	1			
NAND		4/3	5/3	(n + 2)/3
NOR		5/3	7/3	(2n + 1)/3
Multiplexer		2	2	2
XOR		4	12	

[From Sutherland, Sproull, Harris]

Optimizing Complex Combinational Logic

Multistage Networks

$$Delay = \sum_{i=1}^{N} \left(p_i + LE_i \cdot f_i \right)$$

Effective fanout: $EF_i = LE_i f_i$

Path delay $D = \Sigma d_i = \Sigma p_i + \Sigma EF_i$

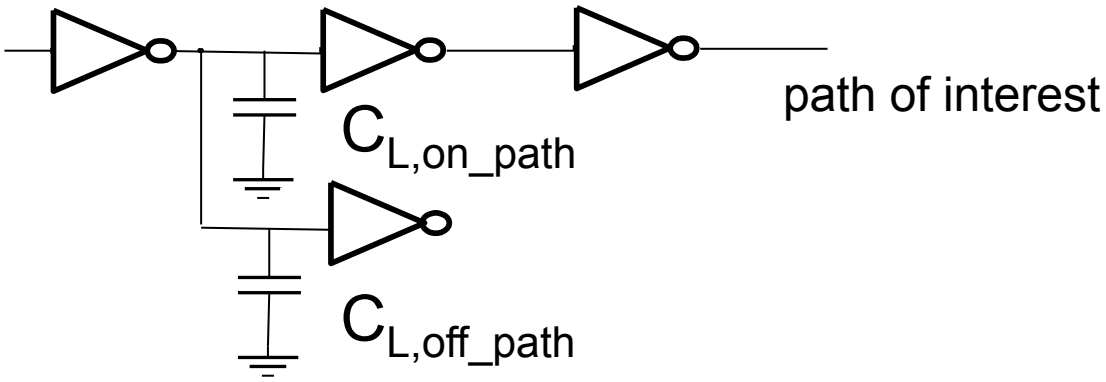
Path electrical fanout: $F = C_L / C_{in} = \Pi f_i$

Path logical effort: $\Pi LE = LE_1 LE_2 \ldots LE_N$

Path effort: $PE = \Pi LE \; F$

Only for tree networks

Adding branching



Branching effort: $$b = \frac{C_{L,on-path} + C_{L,off-path}}{C_{L,on-path}}$$

Multistage Networks

$$Delay = \sum_{i=1}^{N} \left(p_i + LE_i \cdot f_i \right)$$

Effective fanout: $EF_i = LE_i f_i$

Path delay $D = \Sigma d_i = \Sigma p_i + \Sigma EF_i$

Path electrical fanout: $F = C_L / C_{in}$

Branching effort: $\Pi B = b_1 b_2 \ldots b_N$

$\Pi f_i = \Pi B \; F$ *(assuming all paths in the tree are important)*

Path logical effort: $\Pi LE = LE_1 LE_2 \ldots LE_N$

Path effort: $PE = \Pi LE \; \Pi B \; F$

Optimum Effort per Stage

When each stage bears the same effort (effective fanout):

$$EF^N = PE$$

$$EF = \sqrt[N]{PE}$$

Effective fanouts: $LE_1 f_1 = LE_2 f_2 = \ldots = LE_N f_N$

Minimum path delay

$$\hat{D} = \sum_{i=1}^{N} \left(LE_i f_i + p_i \right) = N \cdot PE^{1/N} + \sum_{i=1}^{N} p_i$$

Optimal Number of Stages

For a given load,
and given input capacitance of the first gate
Find optimal number of stages and optimal sizing

$$D = N \cdot PE^{1/N} + \sum p_i$$

Remember: we can always add inverters to the end of the chain

The 'best effective fanout' $EF = PE^{1/\hat{N}}$ is still around 4
(3.6 with γ=1)

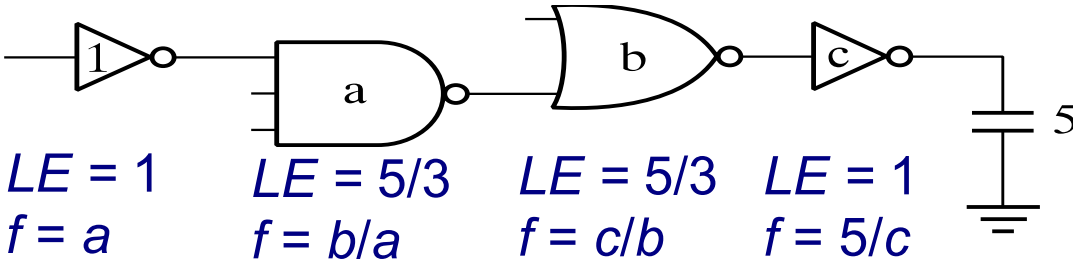
Method of Logical Effort: Summary

- Compute the path effort: PE = (ΠLE)*BF*
- Find the best number of stages $N \sim \log_4 PE$
- Compute the effective fanout/stage EF = $PE^{1/N}$
- Sketch the path with this number of stages
- Work either from either end, find sizes: $C_{in} = C_{out}$*LE/EF

Reference: Sutherland, Sproull, Harris, “Logical Effort”, Morgan-Kaufmann 1999.

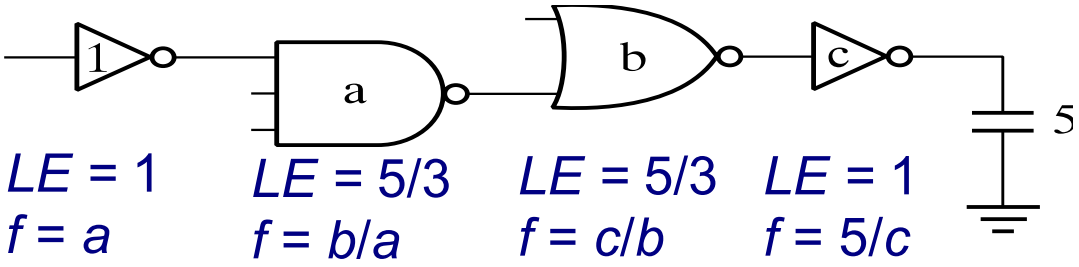
Optimizing Complex Combinational Logic: Examples

Example 1: No branching



Electrical fanout, F =
Π *LE =*
PE =
EF/stage =
a =
b =
c =

Example 1: No branching



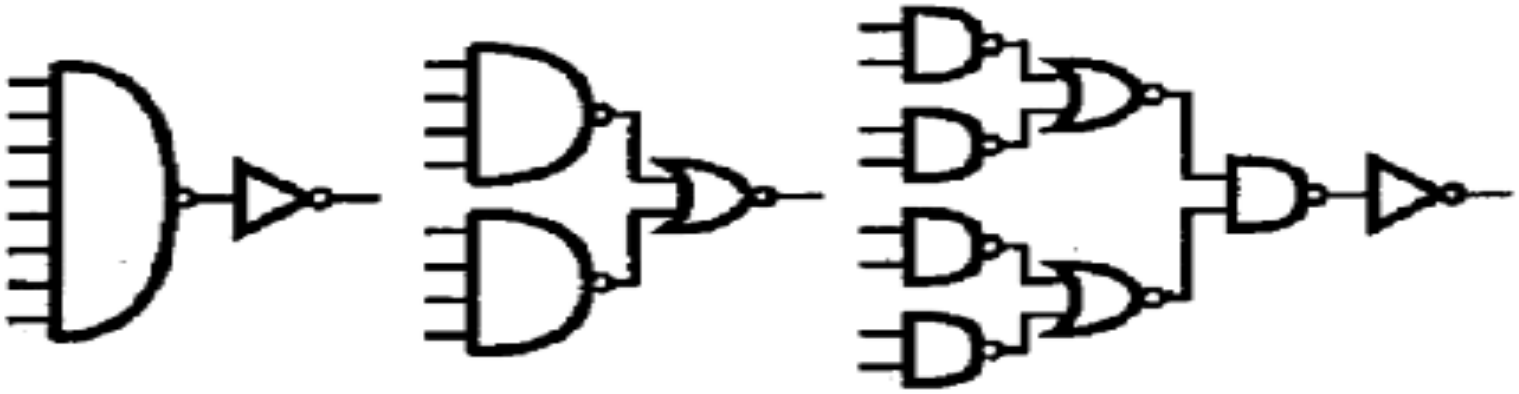
a, b, c are input capacitances normalized to the unit inverter

Electrical fanout, F = 5
Π *LE =* 25/9
PE = 125/9
EF/stage = 1.93
a = 1.93
b = 2.23
c = 2.59

From the back

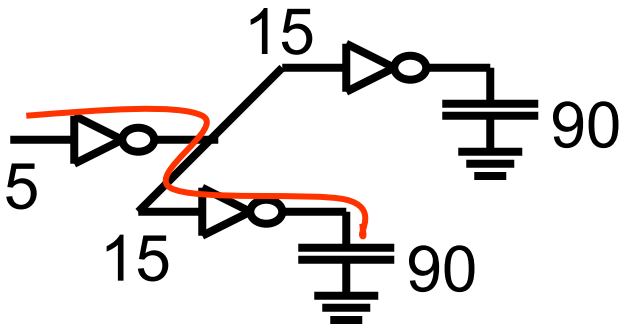
5/c = 1.93
(5/3)c/b = 1.93
(5/3)b/a = 1.93

Our old problem: which one is better?



LE=10/3 1	*LE*=2 5/3	*LE*=4/3 5/3 4/3 1
ΠLE = 10/3	ΠLE = 10/3	ΠLE = 80/27
P = 8 + 1	P = 4 + 2	P = 2 + 2 + 2 + 1

Adding Branching



$LE = 1$

$F = 90/5 = 18$

$PE = 18$ (wrong!)

$EF_1 = (15+15)/5 = 6$

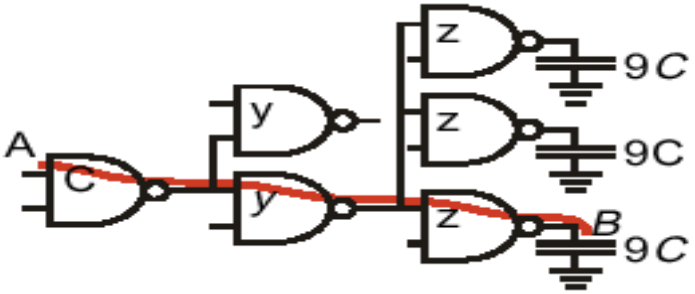
$EF_2 = 90/15 = 6$

$PE =$ 36, not 18!

Better: $PE = F \cdot LE \cdot B = 18 \cdot 1 \cdot 2 = 36$

Example 2 with Branching

Select gate sizes *y* and *z* to minimize delay from *A* to *B*



Logical Effort: $LE =$

Electrical Fanout: $F =$

Branching Effort: $B =$

Path Effort: $PE =$

Best Effective Fanout: $EF =$

Delay: $D =$

Example 2 with Branching

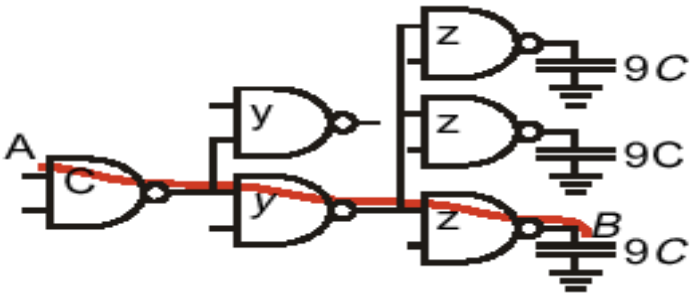
Select gate sizes *y* and *z* to minimize delay from *A* to *B*

Logical Effort: $LE = (4/3)^3$

Electrical Fanout: $F = C_{out}/C_{in} = 9$

Branching Effort: $B = 2 \cdot 3 = 6$

Path Effort: $PE = \prod LE \cdot F \cdot B = 128$



Best Effective Fanout: $EF = PE^{1/3} \approx 5$

Delay: $D = 3 \cdot 5 + 3 \cdot 2 = 21$

Work backward for sizes:

$$z = \frac{9C \cdot (4/3)}{5} = 2.4C$$

$$y = \frac{3z \cdot (4/3)}{5} = 1.9C$$