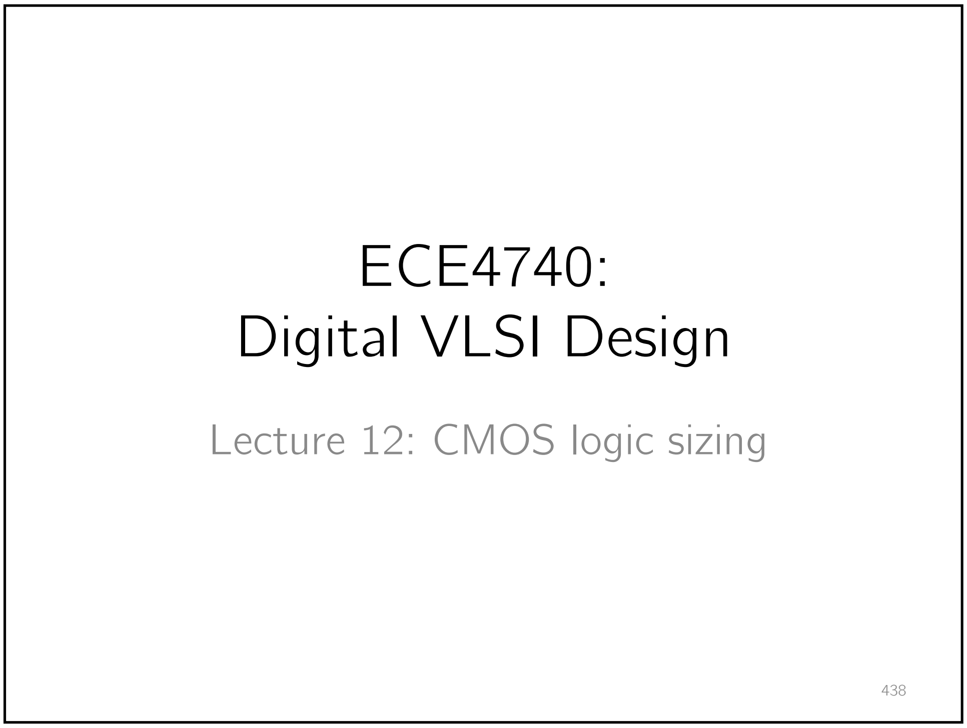


ECE4740:
Digital VLSI Design

Lecture 12: CMOS logic sizing

Needed for sizing CMOS logic gates

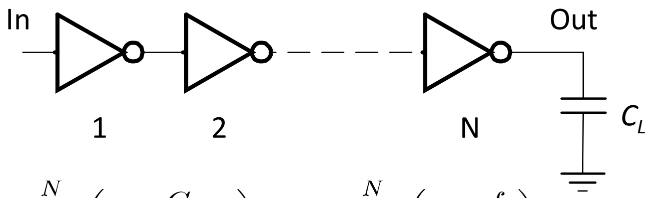
Logical effort

Sizing logic paths for speed

- Input capacitance of logic path is often constrained
  - E.g., ALU in old Intel microprocessor is 0.5pF
- Logic also has to drive some load
- How do we size logic for maximum speed?
- We have solved this for the inverter chain
- Can we generalize it for any type of logic?
  - Determine transistor sizes for complex logic
  - Hand-calculation (=approximation)

440

Remember? Simple buffer example



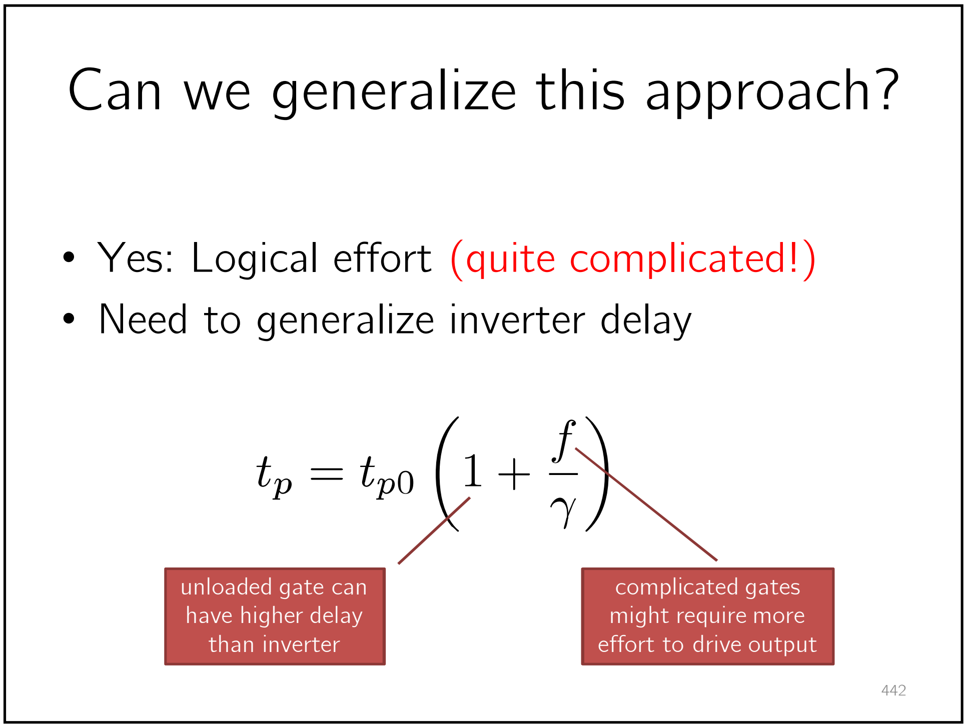
$$t_p = t_{p0} \sum_{i=1}^{N} \left(1 + \frac{C_{i+1}}{\gamma C_i}\right) = t_{p0} \sum_{i=1}^{N} \left(1 + \frac{f_i}{\gamma}\right)$$

effective fan-out

- Optimality condition: $C_i/C_{i-1} = C_{i+1}/C_i$
- Optimum fan-out: $f_{opt} = \sqrt[N]{C_L/C_{in}}$

441

Image adapted from: Digital Integrated Circuits (2nd Edition) by Rabaey, Chandrakasan, Nikolic



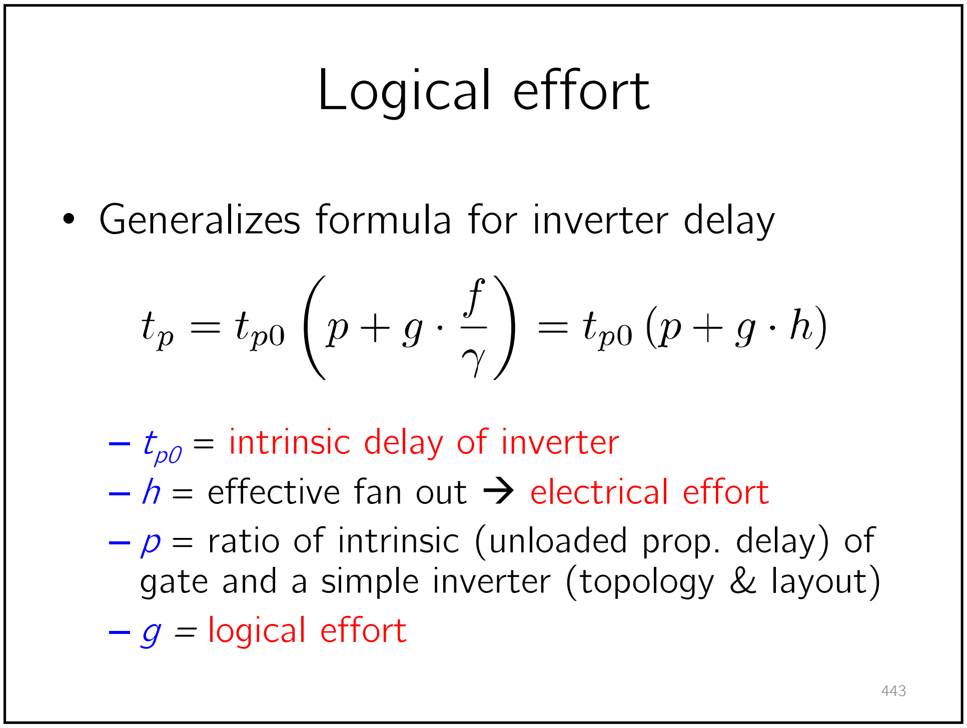
Can we generalize this approach?

- Yes: Logical effort (quite complicated!)
- Need to generalize inverter delay

$$t_p = t_{p0}\left(1 + \frac{f}{\gamma}\right)$$

unloaded gate can have higher delay than inverter

complicated gates might require more effort to drive output



Logical effort

- Generalizes formula for inverter delay

$$t_p = t_{p0}\left(p + g \cdot \frac{f}{\gamma}\right) = t_{p0}\left(p + g \cdot h\right)$$

  - t_{p0} = intrinsic delay of inverter
  - h = effective fan out → electrical effort
  - p = ratio of intrinsic (unloaded prop. delay) of gate and a simple inverter (topology & layout)
  - g = logical effort

Detailed explanation

- Consider loaded delay of NAND gate:

external load capacitance

$$t_{\mathrm{p,NAND}} = 0.69 R_{\mathrm{NAND}} (C_{\mathrm{int,NAND}} + C_{\mathrm{ext}})$$

- Consider unloaded delay of INV gate:

$$t_{\mathrm{p,INV}} = 0.69 R_{\mathrm{INV}} C_{\mathrm{int,INV}}$$

- How much higher is propagation delay of NAND gate than INV gate: $D = \dfrac{t_{\mathrm{p,NAND}}}{t_{\mathrm{p,INV}}}$

Detailed explanation (cont'd)

- Let us evaluate that factor:

$$D = \frac{t_{\mathrm{p,NAND}}}{t_{\mathrm{p,INV}}} = \frac{R_{\mathrm{NAND}}}{R_{\mathrm{INV}}} \left(\frac{C_{\mathrm{int,NAND}}}{C_{\mathrm{int,INV}}} + \frac{C_{\mathrm{ext}}}{C_{\mathrm{int,INV}}} \right)$$

- We assume gate has been sized so that worst-case resistances match, i.e., $\dfrac{R_{\mathrm{NAND}}}{R_{\mathrm{INV}}} = 1$

- Then, we get $D = \underbrace{\dfrac{C_{\mathrm{int,NAND}}}{C_{\mathrm{int,INV}}}}_{= p} + \dfrac{C_{\mathrm{ext}}}{C_{\mathrm{int,INV}}}$

intrinsic delay: how much slower unloaded NAND is than unloaded INV

Detailed explanation (cont'd)

- Focus on the second term: $D = p + \boxed{\frac{C_{\text{ext}}}{C_{\text{int,INV}}}}$
- We want to know how much more this is than for standard sized inverter:

$$
\begin{aligned}
D = p + \frac{C_{\text{ext}}}{C_{\text{int,INV}}} &= p + \frac{C_{\text{g,NAND}}}{C_{\text{g,NAND}}} \frac{C_{\text{ext}}}{C_{\text{int,INV}}} \\
&= p + \frac{C_{\text{g,NAND}}}{C_{\text{int,INV}}} \frac{C_{\text{ext}}}{C_{\text{g,NAND}}} \\
&= p + \underbrace{\frac{C_{\text{g,NAND}}}{C_{\text{g,INV}}}}_{=g} \underbrace{\frac{C_{\text{ext}}}{\gamma C_{\text{g,NAND}}}}_{=h}
\end{aligned}
$$

- We used:

$$\gamma C_{\text{g,INV}} = C_{\text{int,INV}}$$

logical effort: how much more work is needed compared to INV to drive external load

fan out

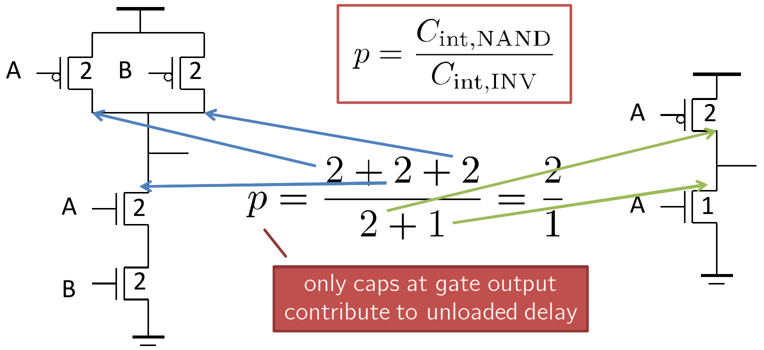
Summary

$$D = \underbrace{\frac{C_{\text{int,NAND}}}{C_{\text{int,INV}}}}_{=p} + \underbrace{\frac{C_{\text{g,NAND}}}{C_{\text{g,INV}}}}_{=g} \underbrace{\frac{C_{\text{ext}}}{\gamma C_{\text{g,NAND}}}}_{=h}$$

- Intrinsic delay: *p*
- Logical effort: *g*
- Effective fan out: *h* (aka. electrical effort)
- Final delay: $\boxed{t_{\text{p,NAND}} = t_{\text{p0}} \cdot D = t_{\text{p,INV}} \cdot D}$

Intrinsic delay p $t_p = t_{p0}\,(p + g \cdot h)$

- Consider min-sized symmetrical NAND2
- Depends on output capacitances*



$$p = \frac{C_{\text{int,NAND}}}{C_{\text{int,INV}}}$$

$$p = \frac{2+2+2}{2+1} = \frac{2}{1}$$

only caps at gate output contribute to unloaded delay

*always in comparison to standard-sized inverter!

Intrinsic delay p $t_p = t_{p0}\,(p + g \cdot h)$

- "How much slower than a CMOS inverter"
- More complex a logic gates → higher the intrinsic delay (compared to INV)

Gate Type	p
Inverter	1
n-input NAND	n
n-input NOR	n
n-way mux	2n
XOR, XNOR	$n\,2^{n-1}$

depends on how you build them

Ignoring second order effects such as internal node capacitances

Logical effort *g* for logic gates

- Inverter has smallest logical effort (g=1) and intrinsic delay (p=1)

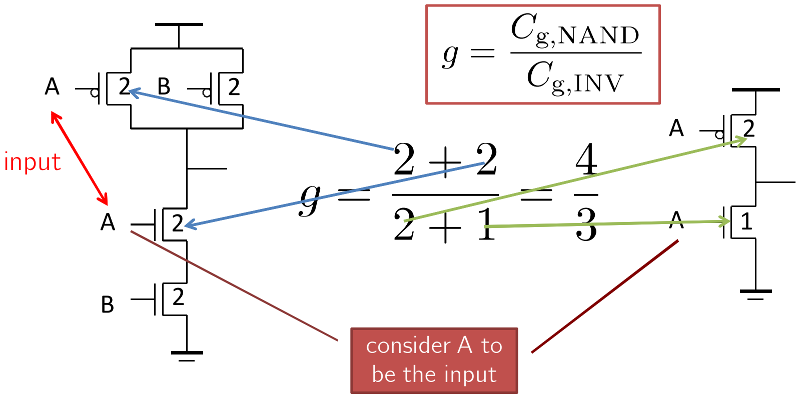
$$t_p = t_{p0}\left(p + \enclose{circle}{g} \cdot h\right)$$

- Logical effort is ratio of input capacitance of given gate to inverter capacitance
  - gate must be sized to deliver same current
  - logical effort refers to only one input
- Logical effort increases with gate complexity
  - function of topology, independent of sizing

450

Logical effort g $t_p = t_{p0}\left(p + \enclose{circle}{g} \cdot h\right)$

- Consider min-sized symmetrical NAND2
- Depends on input capacitances



$$g = \frac{C_{\mathrm{g,NAND}}}{C_{\mathrm{g,INV}}}$$

$$g = \frac{2+2}{2+1} = \frac{4}{3}$$

consider A to be the input

451

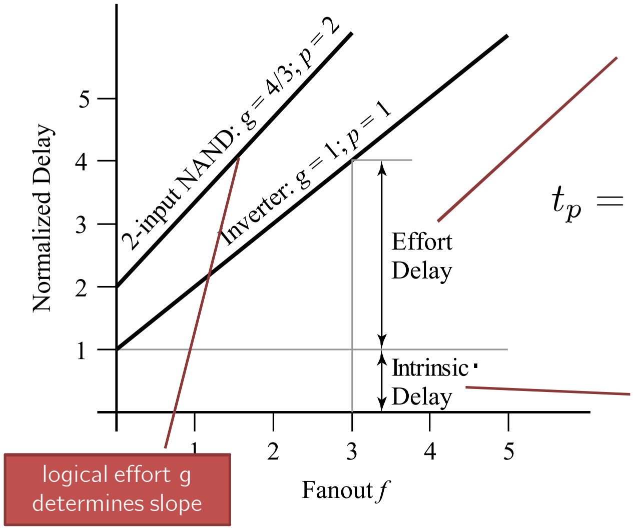
Logical effort g $t_p = t_{p0}\,(p + g \cdot h)$

- Logical effort: given a load, complex gates must work harder to produce similar speed

Gate Type	g (for 1 to 4 input gates)			
	1	2	3	n
Inverter	1			
NAND		4/3	5/3	(n+2)/3
NOR		5/3	7/3	(2n+1)/3
mux		2	2	
XOR		4	12	

Always in comparison to inverter!

Logical effort illustrated



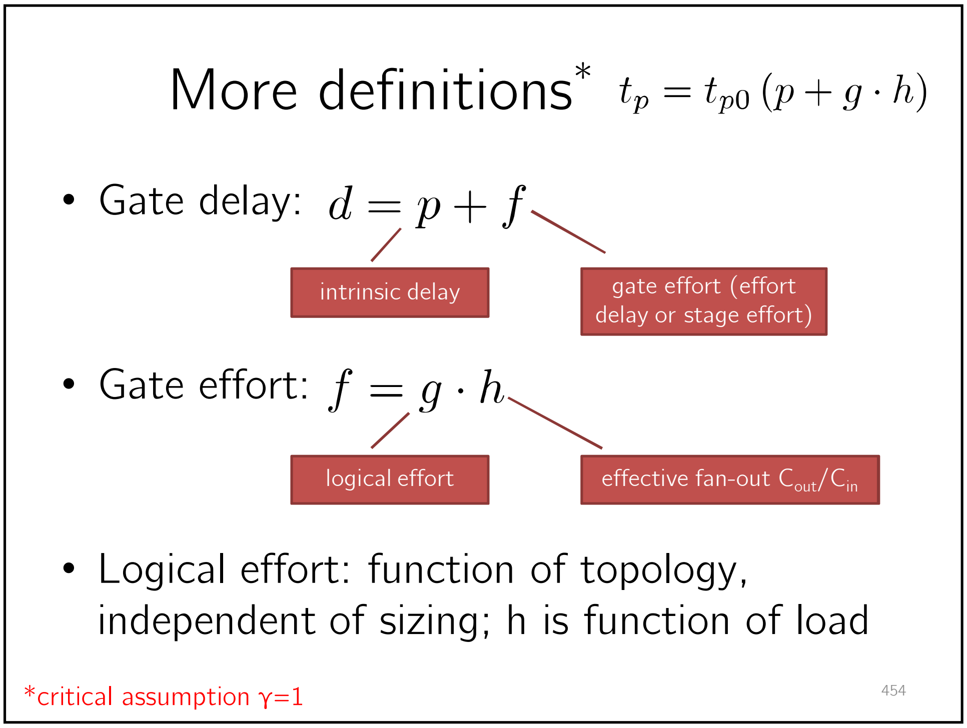
gate effort (or effort delay)

$t_p = t_{p0}\,(p + g \cdot h)$

intrinsic delay

logical effort g determines slope

Image taken from: Digital Integrated Circuits (2nd Edition) by Rabaey, Chandrakasan, Nikolic



More definitions* $t_p = t_{p0}\left(p + g \cdot h\right)$

- Gate delay: $d = p + f$

intrinsic delay

gate effort (effort delay or stage effort)

- Gate effort: $f = g \cdot h$

logical effort

effective fan-out C_{out}/C_{in}

- Logical effort: function of topology, independent of sizing; h is function of load

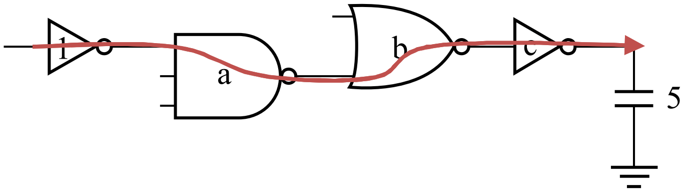
*critical assumption γ=1

454

More definitions...

Sizing of multistage networks

Example



- Assume we want to
  - compute the propagation delay
  - find the minimum propagation delay
  - manually size the gates to achieve minimum propagation delay

Path delay of complex gate network

- Total path delay: $t_p = t_{p0} \sum_{j=1}^{N} (p_j + g_j h_j)$
- Minimum delay through path determines that each stage bears same gate effort
  - take partial derivatives to obtain:

$$g_1 h_1 = g_2 h_2 = \ldots = g_N h_N = f_{opt}$$

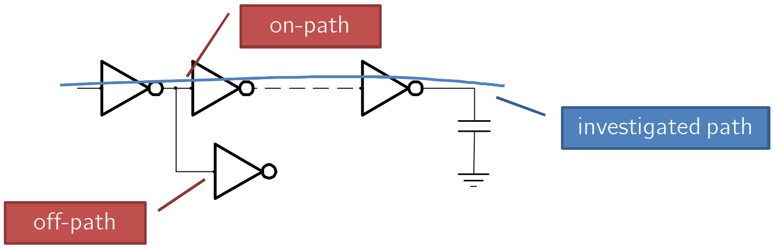
In what follows, we assume γ=1

458

We also need branching effort

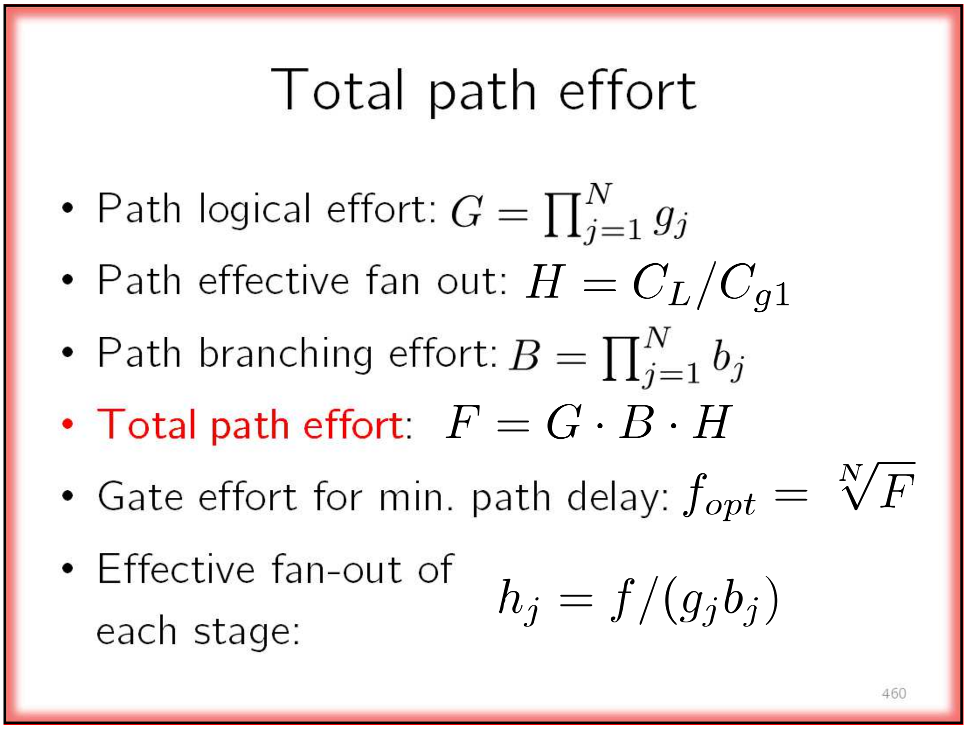
- Some drive current is directed on considered path, some is directed off path
- Branching effort

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



459

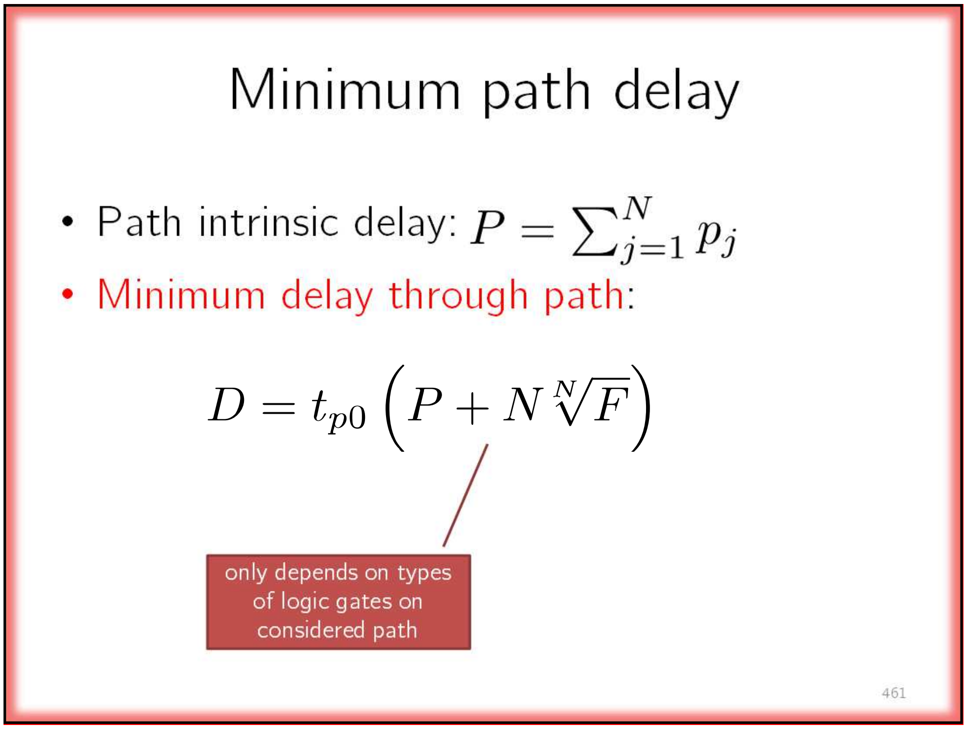
Image adapted from: Digital Integrated Circuits (2nd Edition) by Rabaey, Chandrakasan, Nikolic



Total path effort

- Path logical effort: $G = \prod_{j=1}^{N} g_j$
- Path effective fan out: $H = C_L / C_{g1}$
- Path branching effort: $B = \prod_{j=1}^{N} b_j$
- Total path effort: $F = G \cdot B \cdot H$
- Gate effort for min. path delay: $f_{opt} = \sqrt[N]{F}$
- Effective fan-out of each stage: $h_j = f / (g_j b_j)$

460



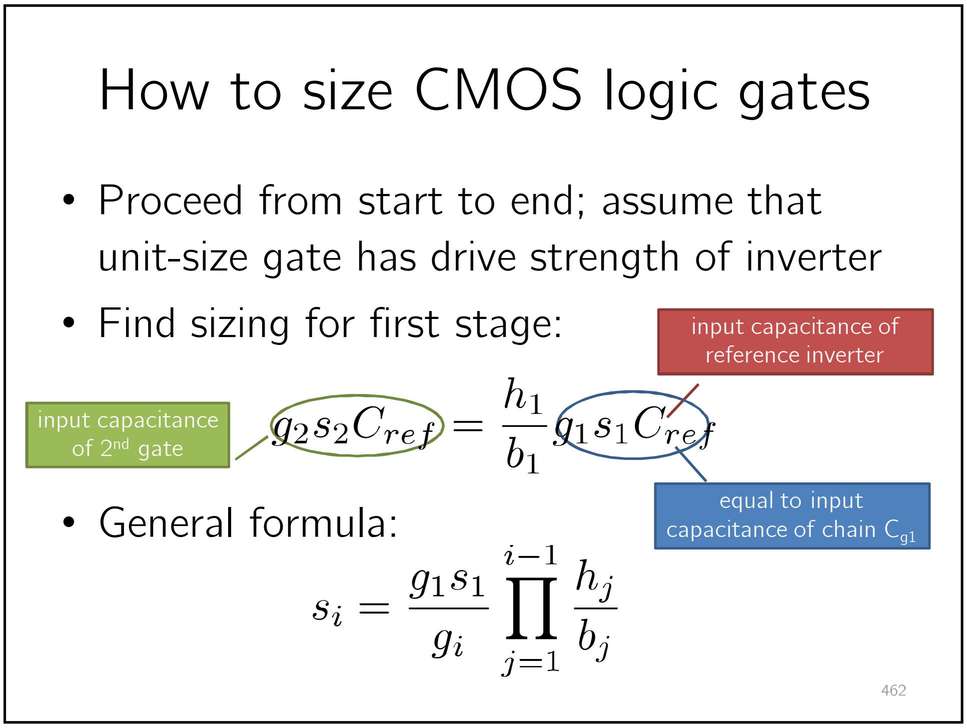
Minimum path delay

- Path intrinsic delay: $P = \sum_{j=1}^{N} p_j$
- Minimum delay through path:

$$D = t_{p0} \left(P + N \sqrt[N]{F} \right)$$

only depends on types of logic gates on considered path

461



How to size CMOS logic gates

- Proceed from start to end; assume that unit-size gate has drive strength of inverter
- Find sizing for first stage:

input capacitance of reference inverter

$$g_2 s_2 C_{ref} = \frac{h_1}{b_1} g_1 s_1 C_{ref}$$

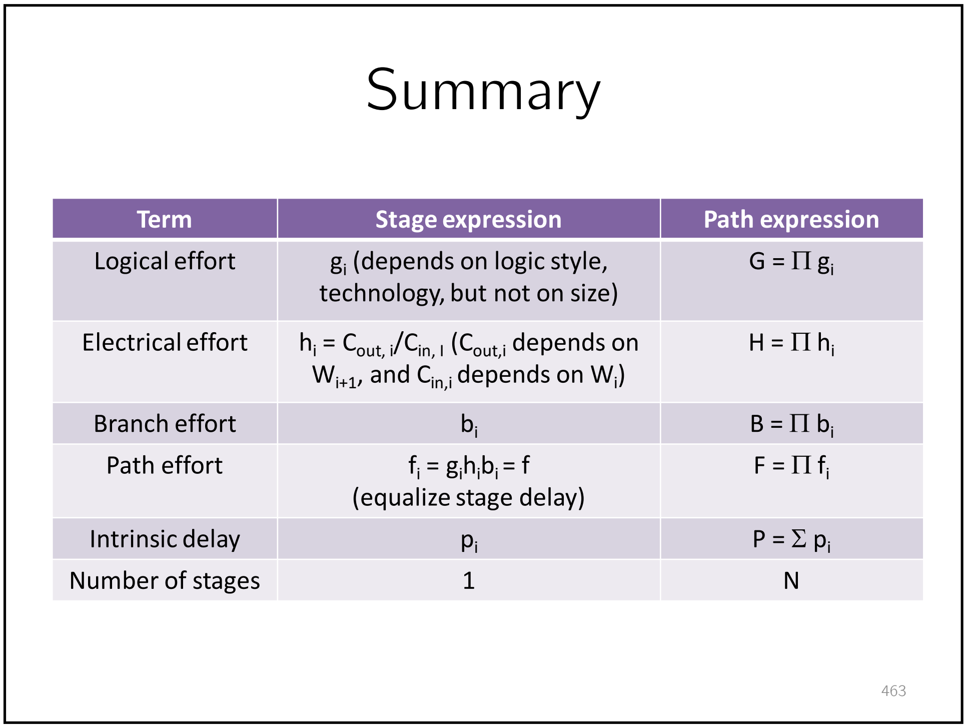
input capacitance of 2nd gate

equal to input capacitance of chain C_{g1}

- General formula:

$$s_i = \frac{g_1 s_1}{g_i} \prod_{j=1}^{i-1} \frac{h_j}{b_j}$$

462



Summary

Term	Stage expression	Path expression
Logical effort	g_i (depends on logic style, technology, but not on size)	$G = \Pi g_i$
Electrical effort	$h_i = C_{out,\,i}/C_{in,\,I}$ ($C_{out,i}$ depends on W_{i+1}, and $C_{in,i}$ depends on W_i)	$H = \Pi h_i$
Branch effort	b_i	$B = \Pi b_i$
Path effort	$f_i = g_i h_i b_i = f$ (equalize stage delay)	$F = \Pi f_i$
Intrinsic delay	p_i	$P = \Sigma p_i$
Number of stages	1	N

463

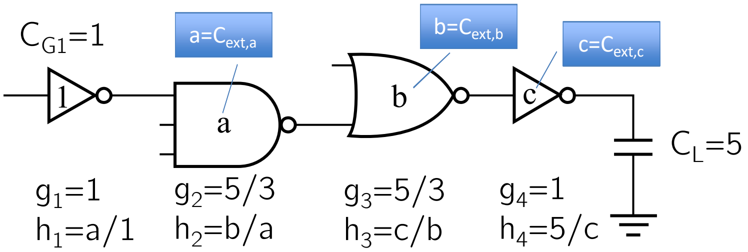
The delay calculation recipe

- Path logical effort: $G = \prod_{j=1}^{N} g_j$
- Path effective fan out: $H = C_L / C_{g1}$
- Path branching effort: $B = \prod_{j=1}^{N} b_j$
- Total path effort: $F = G \cdot B \cdot H$
- Path intrinsic delay: $P = \sum_{j=1}^{N} p_j$
- Minimum delay: $D = t_{p0}\left(P + N\sqrt[N]{H}/\gamma\right)$

sizing not required for min. delay!

464

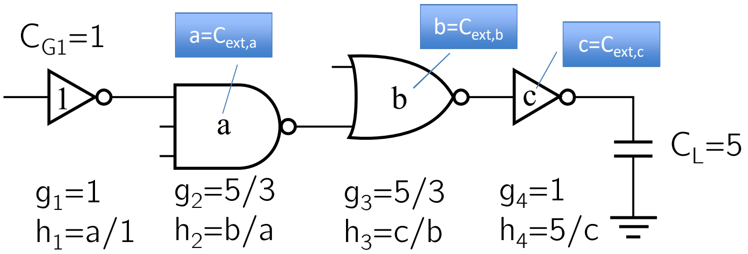
Example 1: optimize delay



- Effective fan-out: H=5/1
- Path logical effort: G=1*5/3*5/3*1=25/9
- Path branching effort: B=1 (no branching)
- Total path effort: F=GBH=125/9

465

Example 1: optimize delay (cont'd)



$g_1=1$ $h_1=a/1$

$g_2=5/3$ $h_2=b/a$

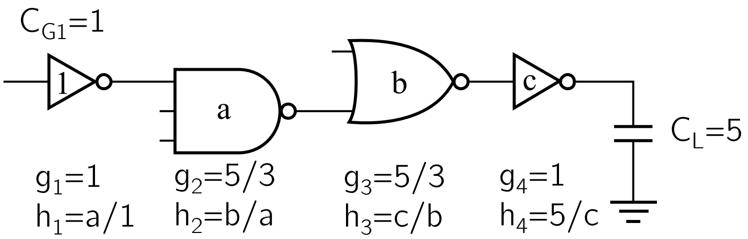
$g_3=5/3$ $h_3=c/b$

$g_4=1$ $h_4=5/c$

- Total path effort: F=GDB=125/9
- Optimal gate effort: $f_{opt}=(125/9)^{1/4}=1.93$
- $f_{opt}=g_1*h_1=1*a/1$ → $a=1.93$
- $f_{opt}=g_2*h_2=5/3*b/a$ → $b=2.23$

466

Example 1: optimize delay (cont'd)



$g_1=1$ $h_1=a/1$

$g_2=5/3$ $h_2=b/a$

$g_3=5/3$ $h_3=c/b$

$g_4=1$ $h_4=5/c$

- $f_{opt}=g_2*h_2=5/3*b/a$ → $b=2.23$
- $f_{opt}=g_3*h_3=5/3*c/b$ → $c=2.59$
- Final check: $C_L/c=f_{opt}$? $5/2.59=1.93=f_{opt}$
- Now we can size the gates…

467

Sizing is done iteratively

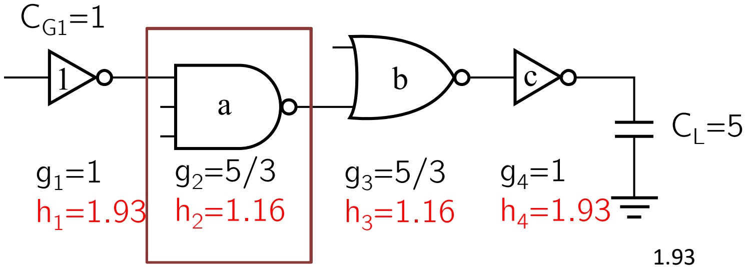
- Remember the recursive sizing formula:

$$g_{i+1}s_{i+1} = \frac{h_i}{b_i} g_i s_i$$

- s = sizing factor
- g = logical effort
- b = branching effort
- E.g., proceed from beginning to end

468

Example 1: 3-NAND sizing



g_1=1 h_1=1.93 g_2=5/3 h_2=1.16 g_3=5/3 h_3=1.16 g_4=1 h_4=1.93

- PMOS
  - (W_p=2 to match inverter) * h_2 = 2.32
  - or: 2 * s_2 = 2.32
- NMOS
  - (W_n=3 to match inverter) * h_2 = 3.48
  - or: 3 * s_2 = 3.48

$$s_2 = \frac{h_1}{b_1 g_2} g_1 s_1$$

1.93

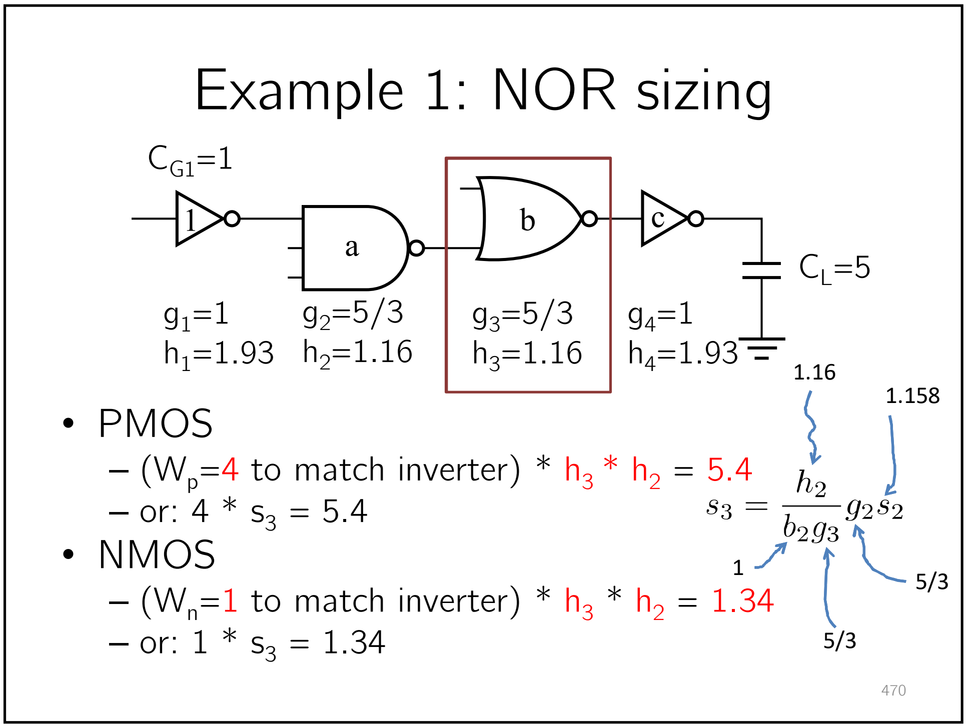
1

1

5/3

1

469



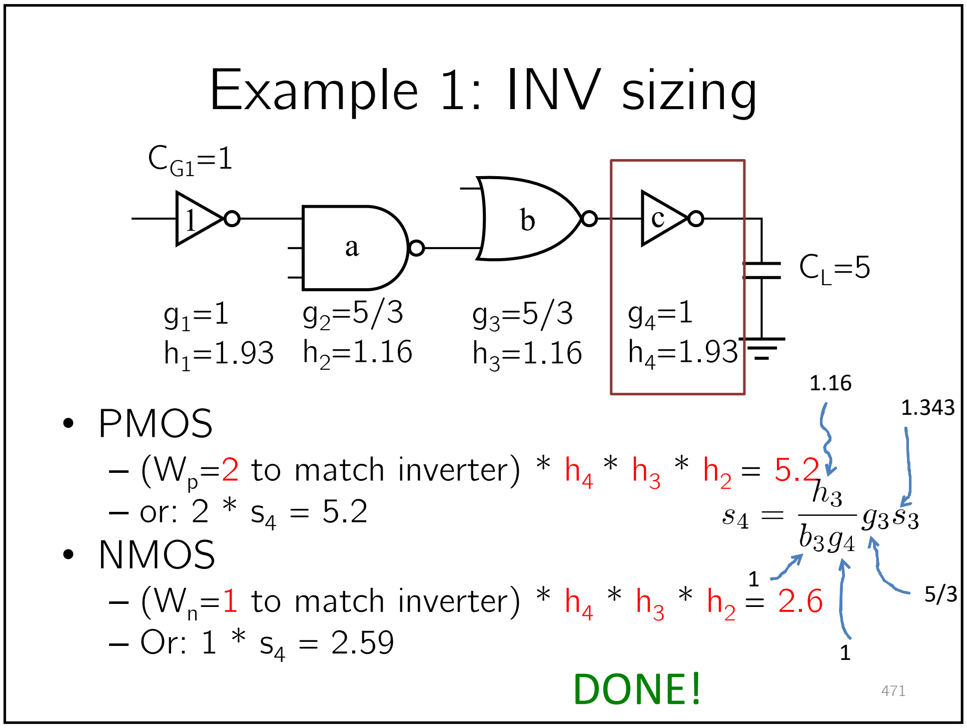
Example 1: NOR sizing

$g_1=1$, $h_1=1.93$ | $g_2=5/3$, $h_2=1.16$ | $g_3=5/3$, $h_3=1.16$ | $g_4=1$, $h_4=1.93$

- PMOS
  - ($W_p=4$ to match inverter) * h_3 * h_2 = 5.4
  - or: 4 * s_3 = 5.4
- NMOS
  - ($W_n=1$ to match inverter) * h_3 * h_2 = 1.34
  - or: 1 * s_3 = 1.34

$$s_3 = \frac{\overset{1.16}{h_2}}{\underset{1}{b_2}\underset{5/3}{g_3}}\underset{5/3}{g_2}\overset{1.158}{s_2}$$

470



Example 1: INV sizing

$g_1=1$, $h_1=1.93$ | $g_2=5/3$, $h_2=1.16$ | $g_3=5/3$, $h_3=1.16$ | $g_4=1$, $h_4=1.93$

- PMOS
  - ($W_p=2$ to match inverter) * h_4 * h_3 * h_2 = 5.2
  - or: 2 * s_4 = 5.2
- NMOS
  - ($W_n=1$ to match inverter) * h_4 * h_3 * h_2 = 2.6
  - Or: 1 * s_4 = 2.59

$$s_4 = \frac{\overset{1.16}{h_3}}{\underset{1}{b_3}\underset{1}{g_4}}\underset{5/3}{g_3}\overset{1.343}{s_3}$$

DONE!

471