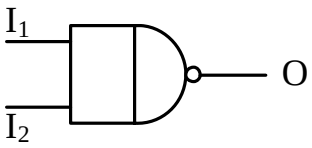
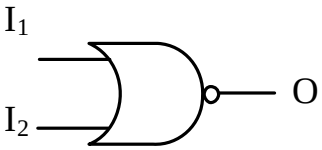
Combinational Logic

NAND



I_1	I_2	O
0	0	1
0	1	1
1	0	1
1	1	0

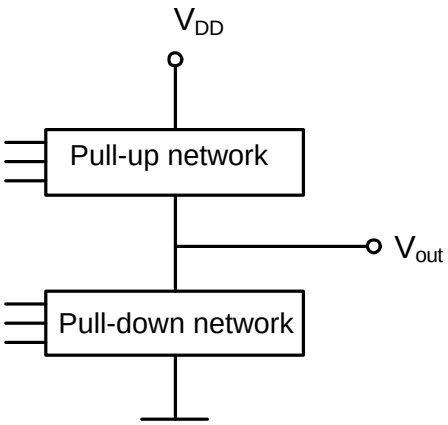
NOR

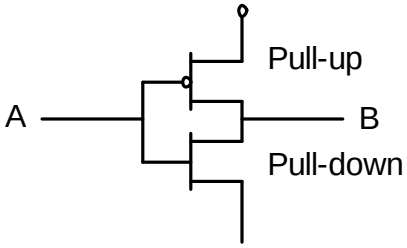


I_1	I_2	O
0	0	1
0	1	0
1	0	0
1	1	0

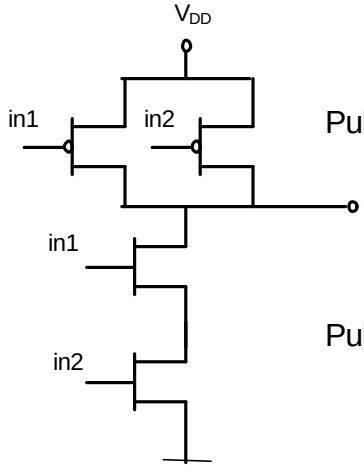
NAND and NOR gates are easy to realize in CMOS

Complementary CMOS Logic





*Two input NAND



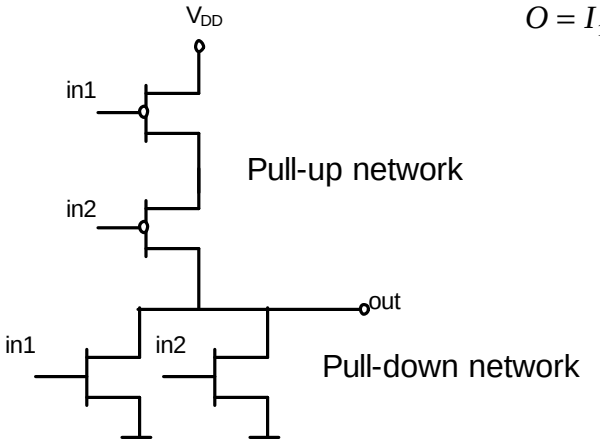
Pull-up network

Pull-down network

$$O = \overline{I_1 \cdot I_2}$$

I_1	I_2	O
0	0	1
0	1	1
1	0	1
1	1	0

*Two input NOR



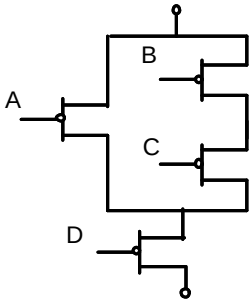
$$O = \overline{I_1 + I_2}$$

I_1	I_2	O
0	0	1
0	1	0
1	0	0
1	1	0

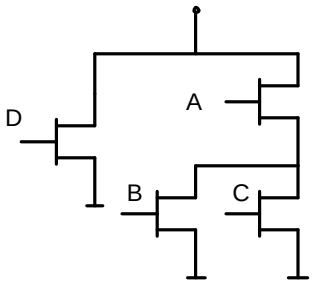
How to realize a complex function?

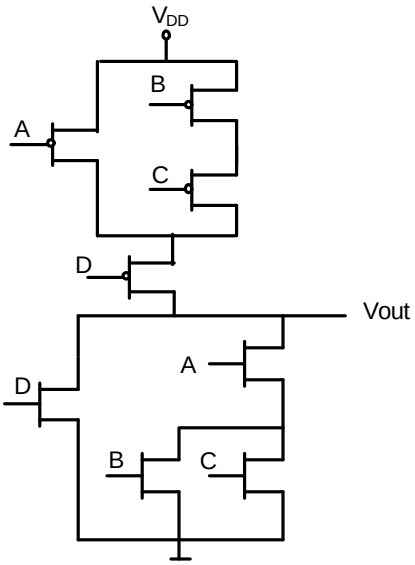
$$F = \overline{D + A \cdot (B + C)}$$

Pull-up network

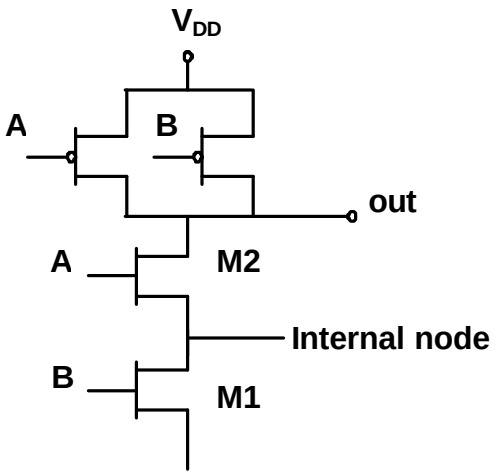


Pull-down network

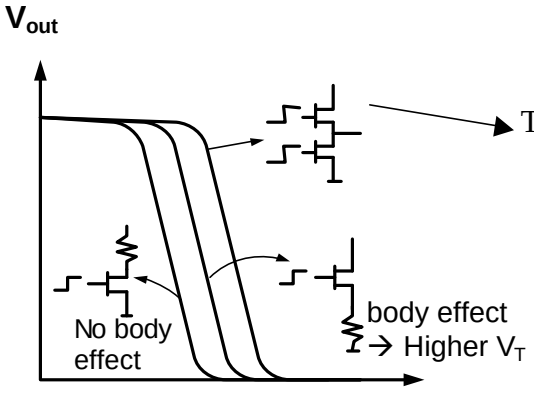




Transfer Characteristic → depends on the data input patterns assume 2-in nand



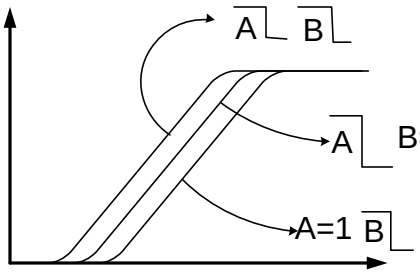
→ M2 has higher Threshold voltage than M1 due to body effect



This has to become higher than $\frac{V_{DD}}{2}+V_T$ for the gate to activate output

Propagation delay of Complementary CMOS gate

Propagation delay → depends on the data input pattern



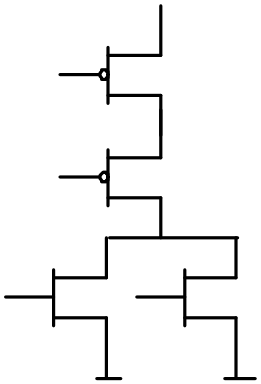
→ delay = $0.69\left(\frac{Rp}{2}\right)C_L$

→delay = $0.69RpC_L$

→ delay = $0.69Rp(C_L+C_{\text{internal node}})$

- For NAND gate to have same pull-down delay(t_{phl}) as min-sized inverter, NMOS devices must be twice as wide to get same R_N and PMOS devices remain unchanged.
- Widening NMOS increase capacitance so NAND gate will be always slower than inverter.
- For NOR gate to be as fast as inverter PMOS devices 2X larger than inverter and NMOS devices remain unchanged.

NOR gate



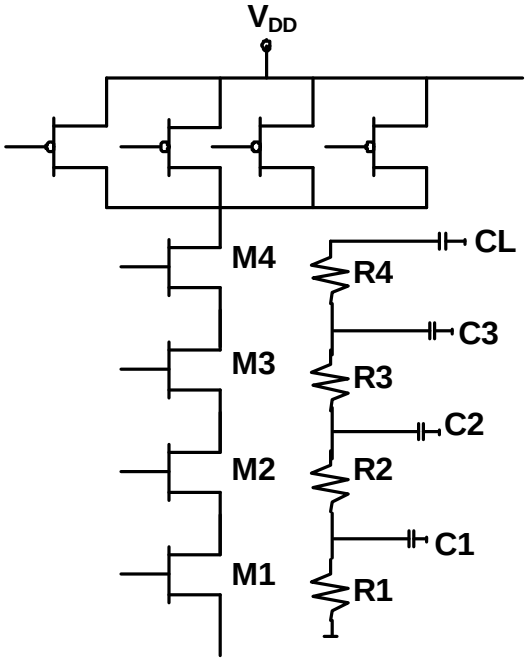
→ stacking PMOS slows down the performance

→ NOR gates is generally slower than NAND gates

How to account for internal node

Use Elmore delay model

Assume a 4-input NAND gates

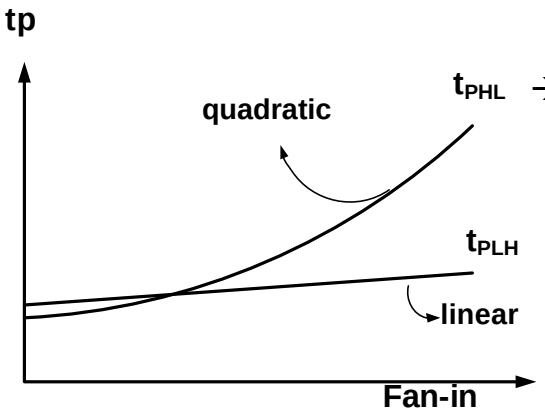


$t_{PHL} = 0.69(R_1C_1+(R_1+R_2)C_2+$

$(R_1+R_2+R_3)C_3+(R_1+R_2+R_3+R_4)C_L)$

assuming identical NMOS devices

$t_{PH}L = 0.69R_N(C_1+2C_2+3C_3+4C_L)$



→ t_{PLH} goes up linearly as capacitance increase linearly with fan-in but resistance remains the same.

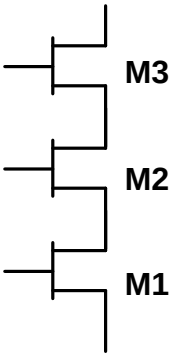
→ t_{PHL} goes up quadratically as capacitance and resistance both increase (Elmore formula)

How to solve the problem of increased delay

Large fan-in NAND gate

Transistor sizing →widening NMOS 's in series also increase the self loading Effect

Progressive transistor sizing



$R_1C_1+R_1C_2+R_1C_3$
$+R_2C_3+R_2C_3$
$+R_3C_3$

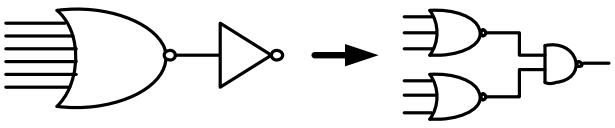
→ $M_1>M_2>M_3$

Input reordering

Identify the critical path (signal that comes last)

→ connect it closer to the output

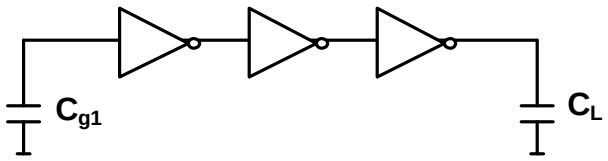
Logic restructuring



Less fan-in → faster

Optimizing Performance in Combinational Networks

Inverter



Case I : N is fixed

Effective fan-out, $F = \frac{C_L}{C_{g1}}$, fan-out of each stages, $f = \sqrt[N]{F}$

Case II : N is not fixed

$t_p = t_{p0}\left(1 + \frac{f}{g}\right)$ for each inverter

same thing applied to combinational Networks

$t_p = t_{p0}\left(p + g\frac{f}{g}\right)$ p : electrical effort, g : logical effort

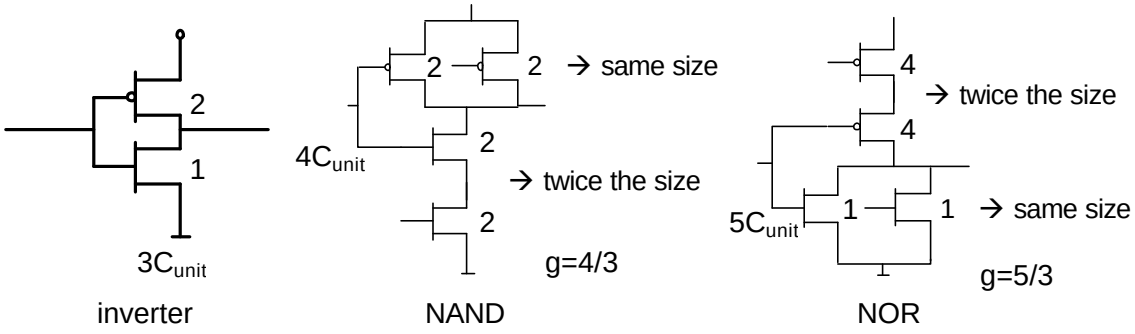
gate	p
inverter	1
n-input NAND	n
n-input NOR	n
n-way MVX	2n
XOR, XNOR	$n2^{n-1}$

electrical effort → p represent the ratio of intrinsic delay of a gate to intrinsic delay of inverter

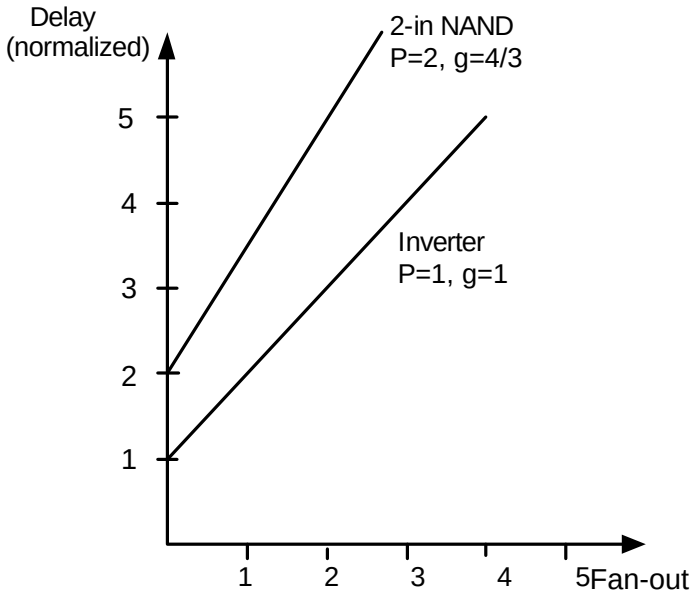
gate	g			
	1-input	2-input	3-input	n-input
inverter	1			
NAND		4/3	5/3	(n+2)/3
NOR		5/3	7/3	(2n+1)/3
MVX		2	2	2
XOR		4	12	?

Logic effort, g → for a given load complex gate has to work harder to produce same results as an inverter(internal nodes) → either same cap, less drive current or same drive current more capacitance

Examples



Inverter delay : $t_p = t_{p0}\left(1+\frac{f}{g}\right)$ Gate delay : $t_p = t_{p0}\left(p+g\frac{f}{g}\right)$



For optimum performance (max speed of chain of complex gates)

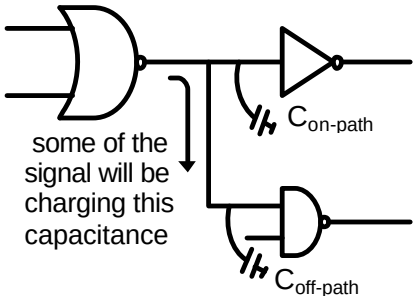
Gate effort, $h = f_1g_1 = f_2g_2 = f_3g_3 = \ldots\ldots = f_ng_n$

So, gate effort = logical effort x effective fan-out

Branching effort

$b = \frac{C_{on-path} + C_{off-path}}{C_{on-path}}$ $C_{on\text{-}path}$: load caps of the gates along the path

$C_{off\text{-}path}$: caps of the connections that leads odd the path



Branching effort along the path , B = $b_1 * b_2 * b_3 * \ldots\ldots$

Total Fan-out, $F = \frac{C_L}{C_{g1}}$

Path effort, $H = GFB = \underbrace{g_1 g_2 g_3}_{\text{logical effort}} \cdots \times \underbrace{\frac{C_L}{C_{g1}}}_{\text{total fan-out}} \times \underbrace{b_1 b_2 b_3}_{\text{branching effort}}$

To minimize the path delay

gate effort of each gate, $h = \sqrt[N]{H} = g_1 f_1 = g_2 f_2 = g_3 f_3 = \cdots$

N: number of gates along the path

Minimum delay of the path

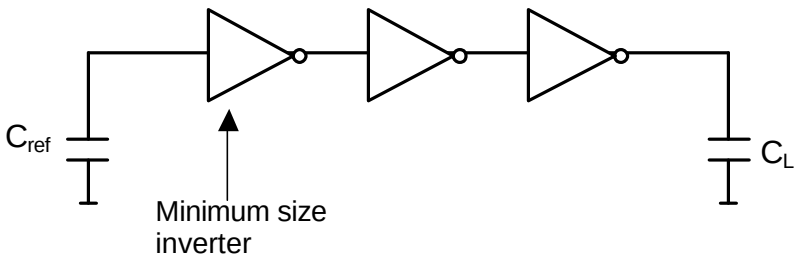
$$D = t_{p1} + t_{p2} + \cdots = t_{p0}\left(\sum_{j=1}^{N} p_j + \frac{N\sqrt[N]{H}}{g}\right)$$

$\sum_{j=1}^{N} p_j$: this part of the delay depends on particular gate's electrical effort

$\frac{N\sqrt[N]{H}}{g}$: this part of the delay is identical for all the gates

D does not depend on sizing of transistors but on types of gates in the path (p,b, g) and total fan-out(F).

To find transistor sizing



Imagine minimum size inverter, input cap → C_{ref}

In the chain of inverter the first one is usually min. size. Based on total fan-out we were able to find sizing of each transistor.

Assume that unit size gate driving capability = min size inverter driving capability

→ input cap of unit size gate = g* C_{ref}

sizing factor of the first gate of the chain → S_1

$$C_{g1} = S_1 g_1 C_{ref}$$

Now the sizing factors of individual chain can be derived by working from front to end.

Input capacitance of gate 2 (include branching effort of gate 1)

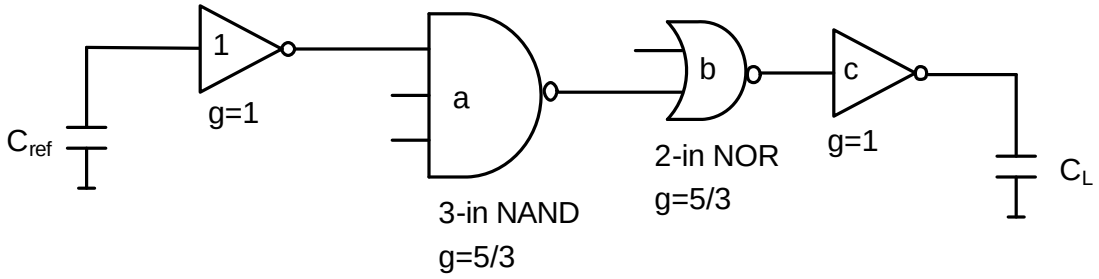
$$g_2 S_2 C_{ref} = \left(\frac{f_1}{b_1} \right) g_1 S_1 C_{ref}$$

$$g_2 S_2 = \frac{f_1}{b_1} g_1 S_1$$

$$\left.\begin{aligned} g_3 S_3 &= \frac{f_2}{b_2} g_2 S_2 = \frac{f_2}{b_2}\frac{f_1}{b_1} g_1 S_1 \\ g_3 S_3 &= \frac{f_{n-1}}{b_{n-1}} \cdots\cdots \frac{f_2}{b_2}\frac{f_1}{b_1} g_1 S_1 \end{aligned}\right\}$$ all the sizing can be formed

Example 1

Minimize the delay on the critical path, assuming F=5



$$F = \frac{C_L}{C_{g1}} = 5 \qquad G = g_1 g_2 g_3 g_4 = 1 \times \frac{5}{3} \times \frac{5}{3} \times 1 = \frac{25}{9}$$

$B = 1 \leftarrow$ no branching in this example

$$H = GFB = \frac{25}{9} \times 5 \times 1 = \frac{125}{9} \qquad h = \sqrt[N]{H} = \sqrt[4]{\frac{125}{9}} = 1.93$$

$$h = f_1 g_1 \rightarrow f_1 = \frac{h}{g_1} = 1.93 \qquad f_2 = \frac{h}{g_2} = \frac{1.93}{5/3} = 1.16$$

$$f_3 = \frac{h}{g_3} = \frac{1.93}{5/3} = 1.16 \quad f_4 = \frac{h}{g_4} = 1.93$$

- > inverter sizing is larger than complex gate because inverter can drive better

Now find gate sizes with respect to min size

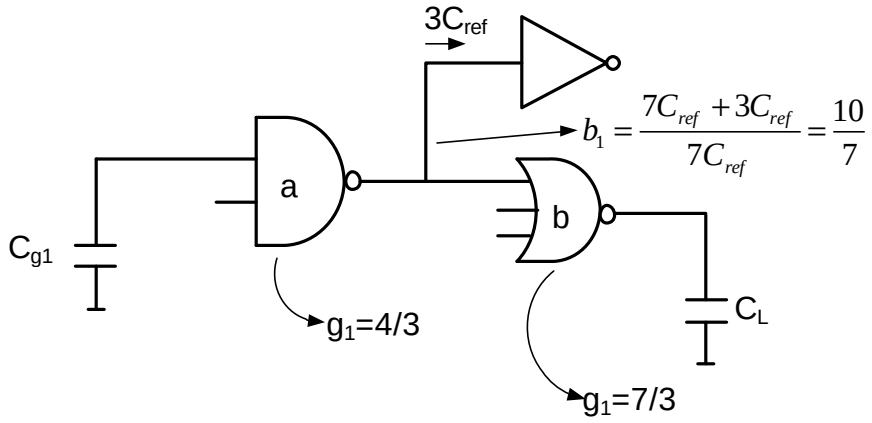
Min size → $S_1 = 1$

$$b_1 = b_2 = b_3 = 1, \qquad g_2 S_2 = f_1 g_1 S_1 \rightarrow S_2 = \frac{f_1 g_1}{g_2} = \frac{1.93}{5/3} = 1.16$$

$$g_3 S_3 = f_2 g_2 S_2 \rightarrow S_3 = \frac{1.16 \times 5/3 \times 1.16}{5/3} = 1.34$$

$$g_4 S_4 = f_3 g_3 S_3 \rightarrow S_4 = \frac{1.16 \times 5/3 \times 1.16}{1} = 2.6$$

Example 2



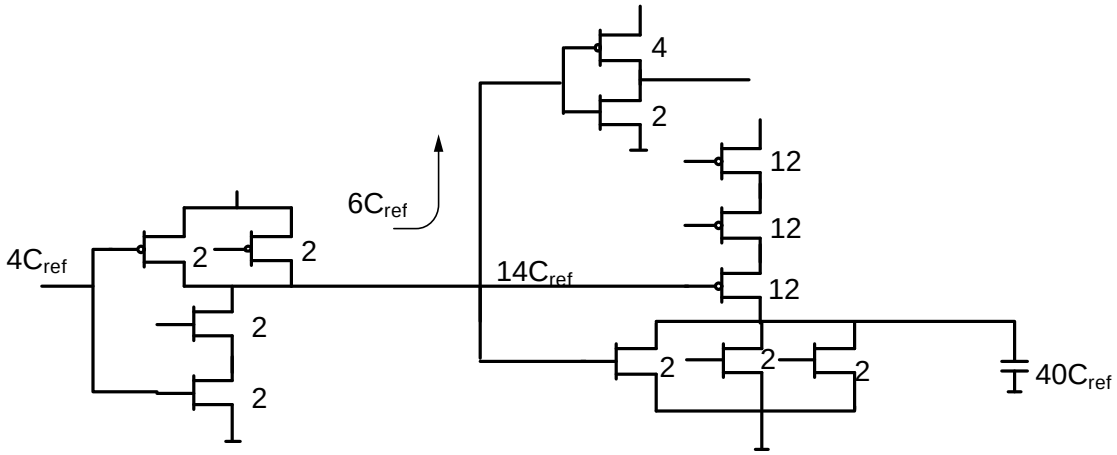
$$\left.\begin{aligned} &F = 10 \\ &G = g_1 \times g_2 = \frac{4}{3} \times \frac{7}{3} = \frac{28}{9} \\ &B = b_1 \times 1 = \frac{10}{7} \end{aligned}\right\} H = GFB = 10 \times \frac{28}{9} \times \frac{10}{7} = \frac{400}{9}$$

$$h = \sqrt[2]{H} = \sqrt[2]{\frac{400}{9}} = \frac{20}{3} = 6.66$$

$$h = f_1 g_1 \quad \rightarrow \quad f_1 = \frac{h}{g_1} = \frac{6.66}{4/3} = \frac{20}{4} = 5$$

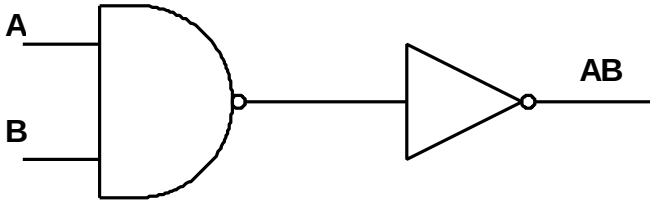
$$h = f_2 g_2 \quad \rightarrow \quad f_2 = \frac{h}{g_2} = \frac{6.66}{7/3} = \frac{20}{7} = 2.85$$

$$\text{sizing} \rightarrow g_2 S_2 = \frac{f_1}{b1}\ g_1 S_1 \quad \rightarrow \quad S_2 = \frac{f_1 g_1}{g_2 b_1} = 2$$



Power Consumption in complex logics

$P = C_L V_{DD}{}^2 a_{0\to1}$ a : activity factor



A	B	Out
0	0	0
0	1	0
1	0	0
1	1	1

$a_{0\to1} = P_0 \cdot P_1 = P_0(1-P_0)$

$a_{0\to1} = \frac{N_0}{2^N} \cdot \frac{N_1}{2^N} = \frac{N_0(2^N - N_0)}{2^{2N}}$ N_0 : # of 0's, N_1 : # of 1's

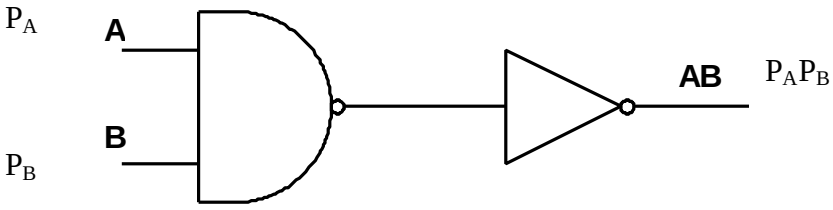
Assumption

* Uniform input probability distribution

* Input signal are not correlated

What if input signal is not uniform?

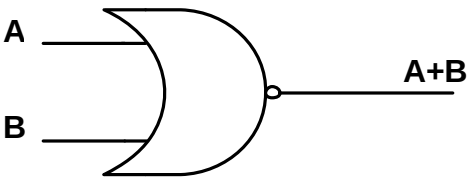
A → P_A = 0.5



P : Probability of signal being "1"

$P_AP_B = P_{output=1}$

$a_0 \to 1 = P_A P_B (1 - P_A P_B)$ for NAND, AND



A	B	Out
0	0	0
0	1	1
1	0	1
1	1	1

$Output = 0 \to (1-P_A)(1-P_B)$

$a_{0\to1} = (1-P_A)(1-P_B)[1-(1-P_A)(1-P_B)]$ for OR

Example

$$P_A = 0.5 \qquad a_{0\to 1 AND} = \mathrm{P_{AND}} = 0.25(1 - 0.25) = 0.18$$
$$P_B = 0.5 \qquad a_{0\to 1 OR} = \mathrm{P_{OR}} = 0.25(1 - 0.25) = 0.18$$

$$P_A = 0.2 \qquad a_{0\to 1 AND} = \mathrm{P_{AND}} = 0.1(1 - 0.1) = 0.09$$
$$P_B = 0.5 \qquad a_{0\to 1 OR} = \mathrm{P_{OR}} = 0.8\times 0.5(1 - 0.8\times 0.5) = 0.24$$