

電子回路：第13回 回路遅延

基礎工学部情報科学科

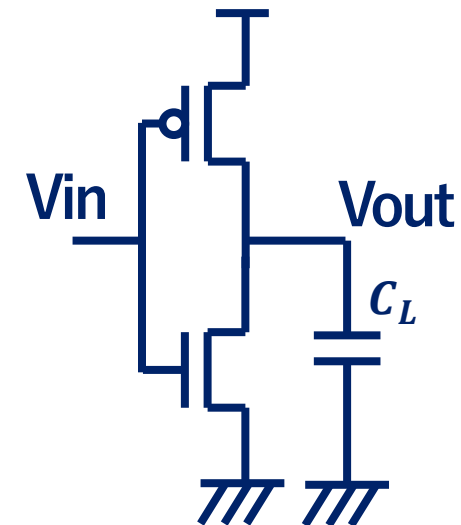
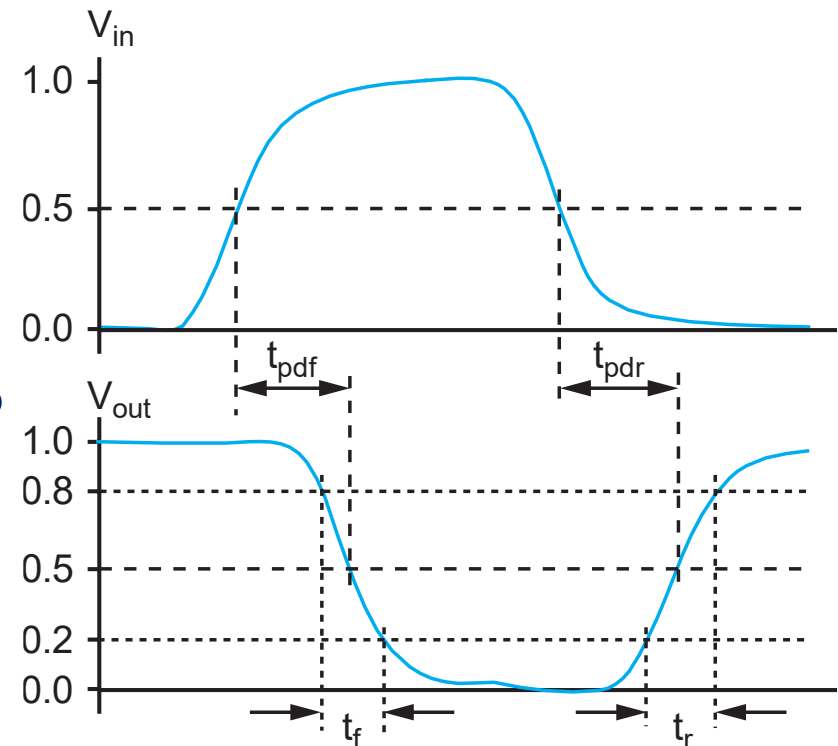
栗野 皓光

awano@ist.osaka-u.ac.jp



インバータの過渡応答

- t_{pdr} : 立ち上がり遅延
 - 入力が $V_{DD}/2$ を横切ってから出力が $V_{DD}/2$ を超えるまでの時間差
- t_{pdf} : 立ち下がり遅延
 - 入力が $V_{DD}/2$ を横切ってから出力が $V_{DD}/2$ を下回るまでの時間差
- t_{pd} : 平均遅延時間
 - $t_{pd} = (t_{pdr} + t_{pdf})/2$



インバータの遅延

立下り遅延を計算してみる

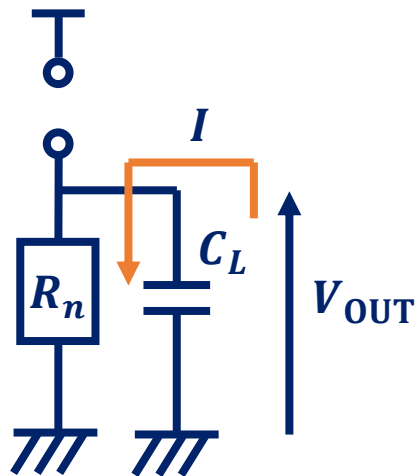
- トランジスタを理想的なスイッチとして考えれば
pMOS->オフ / nMOS->抵抗
- キャパシタ両端の電圧を V_{OUT} とすると、キャパシタに蓄えられている電荷は $Q = C_L \cdot V_{OUT}$
- よってキャパシタに流れる電流は

$$I = \frac{dQ}{dt} = C_L \frac{dV_{OUT}}{dt}$$

- 以上を使って回路方程式を立てると

$$R_n \cdot I = -V_{OUT}$$

$$\Rightarrow R_n C_L \frac{dV_{OUT}}{dt} = -V_{OUT}$$



- 変数分離形なので

$$\frac{dV_{OUT}}{V_{OUT}} = -\frac{dt}{R_n C_L}$$

と変形して両辺を積分すると

$$\ln|V_{OUT}| = -\frac{t}{R_n C_L} + A \quad (A: \text{積分定数})$$

$$V_{OUT} = A \cdot \exp\left(-\frac{t}{R_n C_L}\right)$$

- 時刻 $t = 0$ で $V_{OUT} = V_{DD}$ なので $A = V_{DD}$
- t_{pdf} は出力が $V_{DD}/2$ を横切るまでの時間だったので

$$\frac{V_{DD}}{2} = V_{DD} \cdot \exp\left(-\frac{t_{pdf}}{R_n C_L}\right)$$

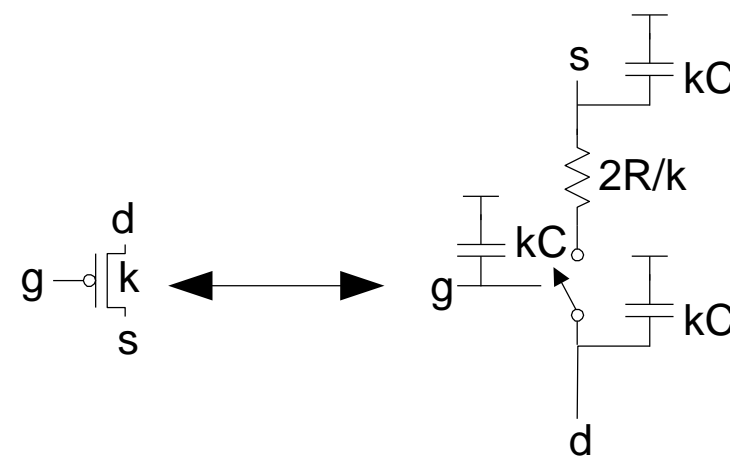
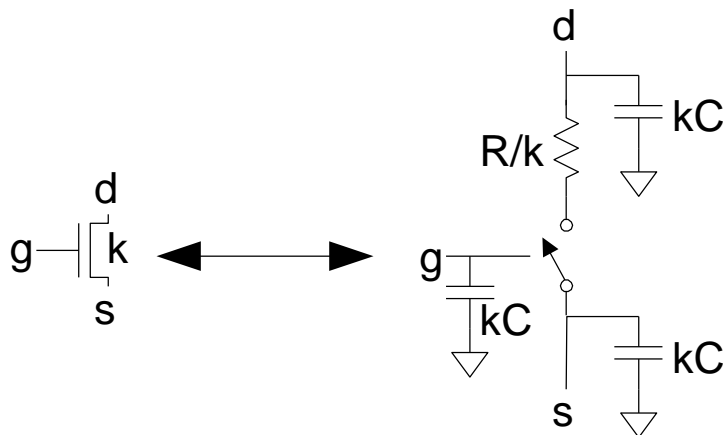
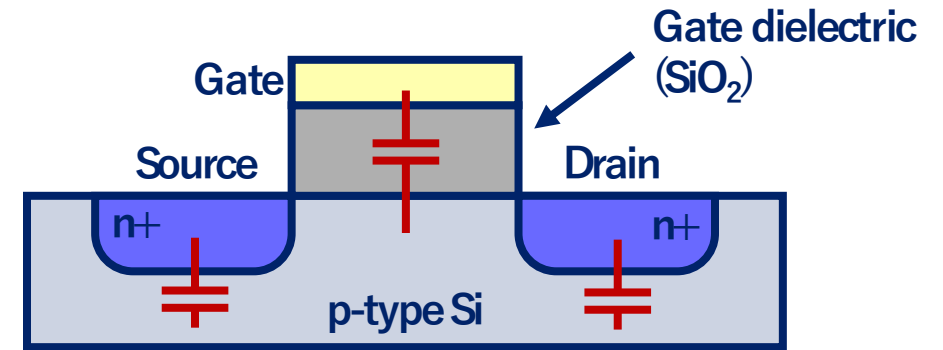
よって立下り遅延は

$$t_{pdf} = \ln 2 \cdot R_n C_L$$



トランジスタの詳細なモデル化

- 現実のトランジスタにはソース・ドレイン・ゲートに寄生容量が出来ている
- ゲート容量 (C_{gate})
 - ゲート面積に比例: $C_{gate} \propto WL$
- 拡散容量 (C_{diff})
 - ソース・ドレインの面積に依存
 - ゲート容量と同程度として近似 ($C_{diff} \doteq C_{gate}$)
- 寄生容量を組み込んだトランジスタの等価回路は以下のようなになる

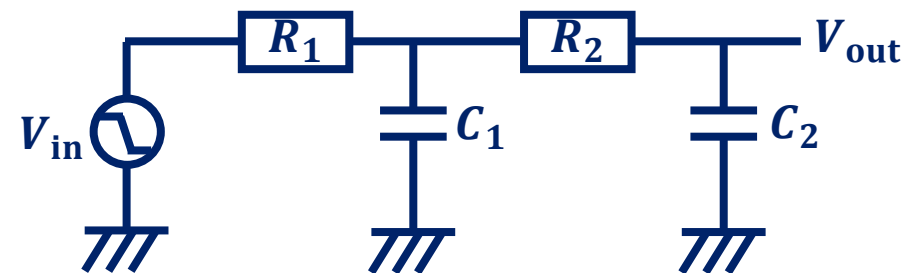


Elmore遅延モデル

- 右の回路で V_{in} が V_{DD} から0に立ち下がった時の V_{out} は

$$V_{out} = V_{DD} \frac{\tau_1 e^{-t/\tau_1} - \tau_2 e^{-t/\tau_2}}{\tau_1 - \tau_2}$$

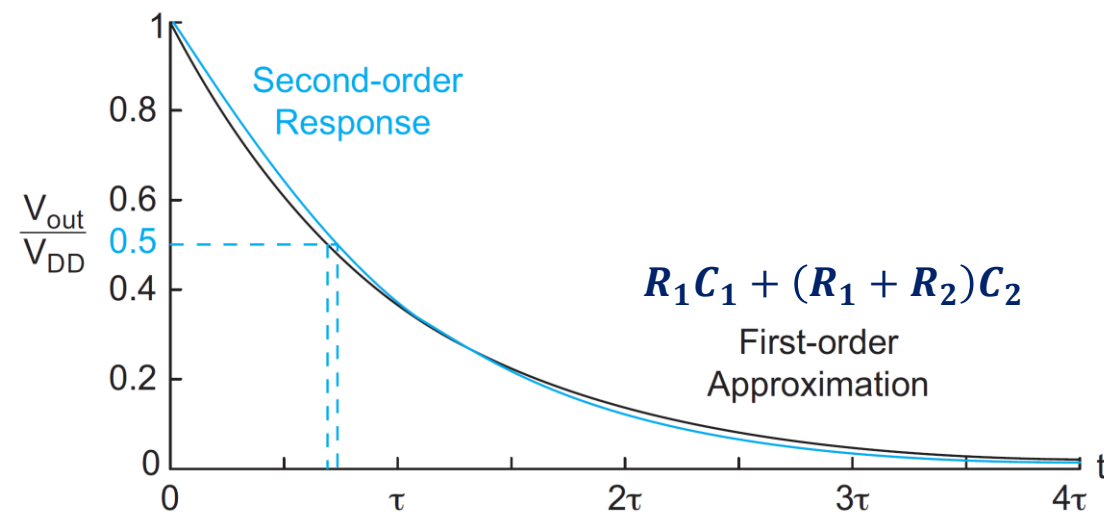
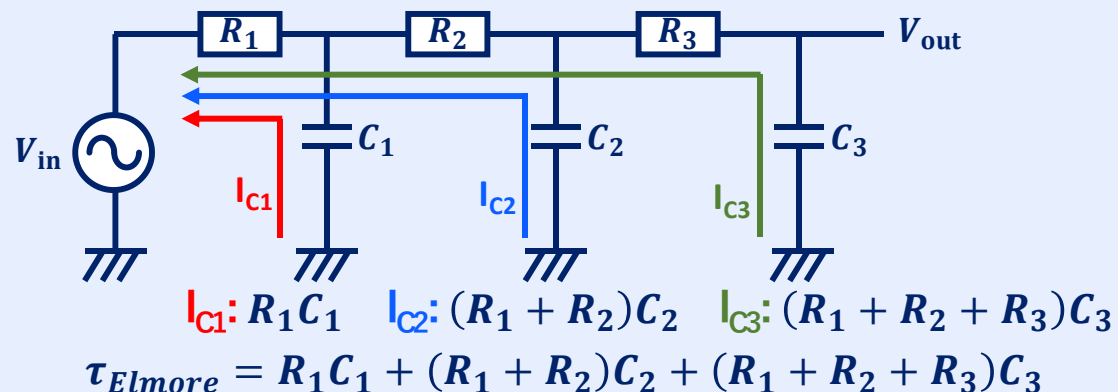
$$\tau_{1,2} = \frac{R_1 C_1 + (R_1 + R_2) C_2}{2} \left(1 \pm \sqrt{1 - \frac{4\hat{R}\hat{C}}{[1 + (1 + \hat{R})\hat{C}]^2}} \right) \quad \hat{R} = \frac{R_2}{R_1} \quad \hat{C} = \frac{C_2}{C_1}$$



- 手計算で解けないほど複雑になってしまうので1段のRCで遅延を近似する方法が考案された⇒Elmore遅延モデル

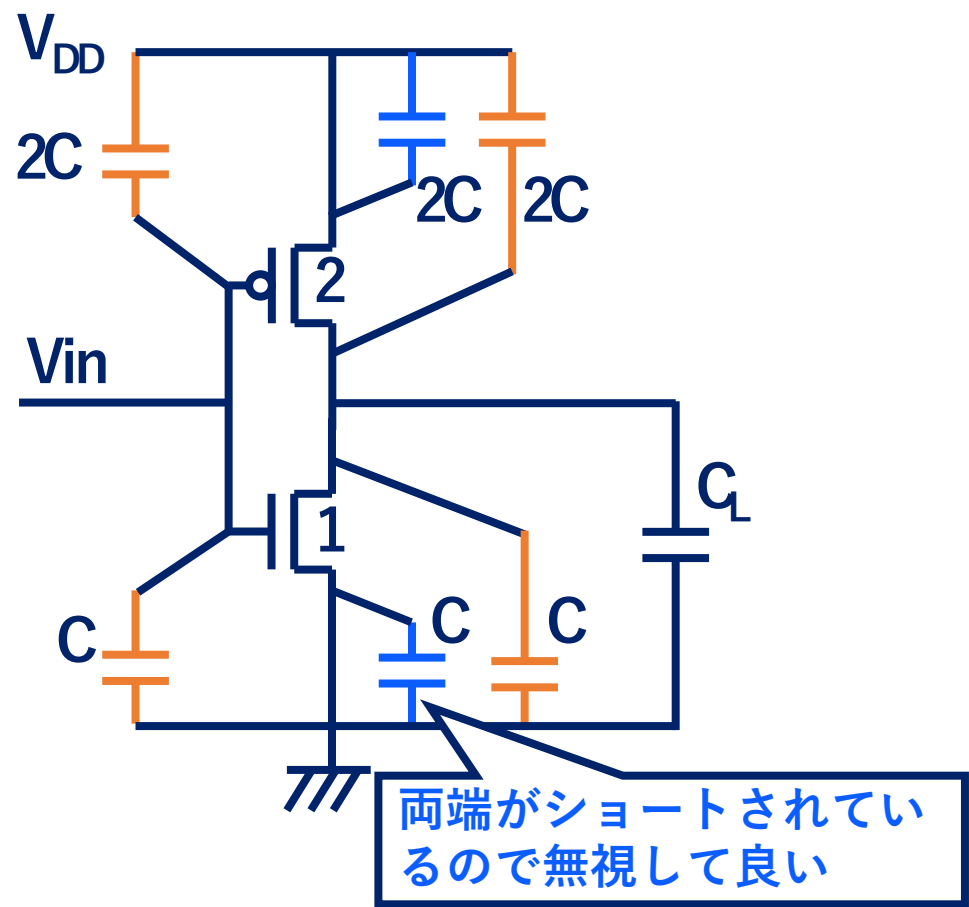
Elmore遅延の計算方法:

1. キャパシタから電圧源に至る経路を描く
2. 静電容量と1.で描いた経路上の抵抗値の和を掛ける
3. 全ての経路に対して2.で求めたRC積を足し合わせる

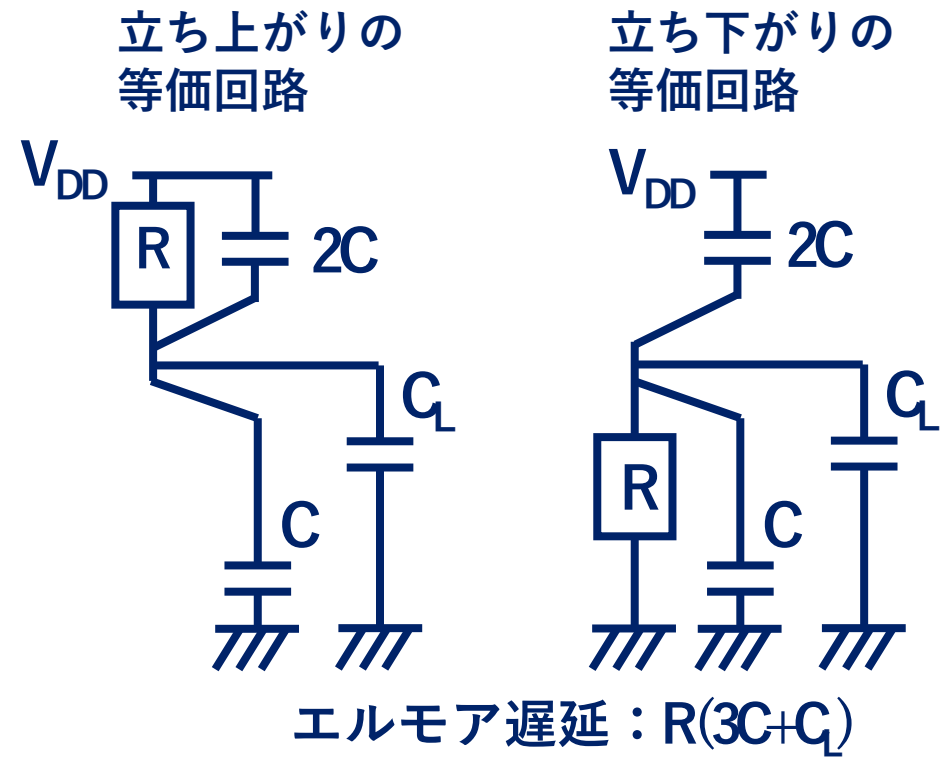


近似精度比較

Elmore遅延の計算例#1：インバータ遅延計算の別解



- 寄生容量を考慮したインバータの等価回路
- 色付きのキャパシタが寄生容量



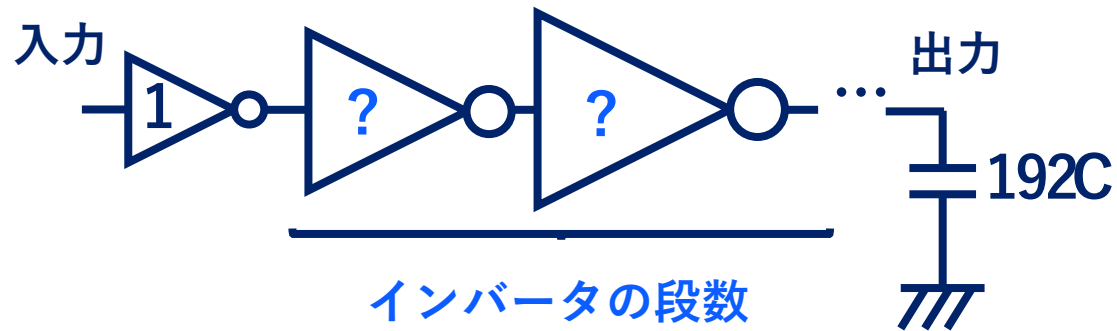
Process node [um]	C [fF/um]	R [kΩ·um]	Time constant (RC) [ps]
1.2	1.79	14.5	26
0.8	1.91	11.0	21
0.6	1.64	7.9	13
0.35	1.69	7.1	12

代表的なRC時定数の例

大きな負荷容量を駆動する場合の最適化

- 単位サイズのインバータを用いて192Cの容量を駆動する場合を考える

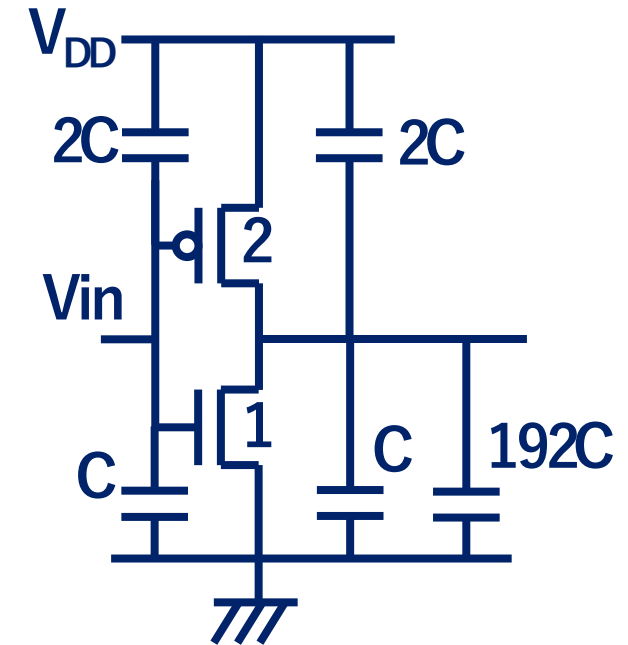
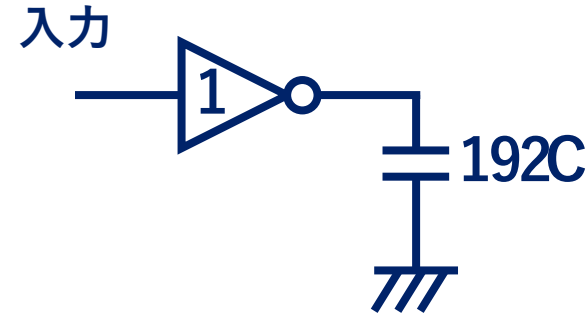
想定するシステム



最適化すべきパラメータ

- 各インバータの駆動力（サイズ）はどのように決めるか？
- 遅延を最小化するためには何段のインバータが必要か？

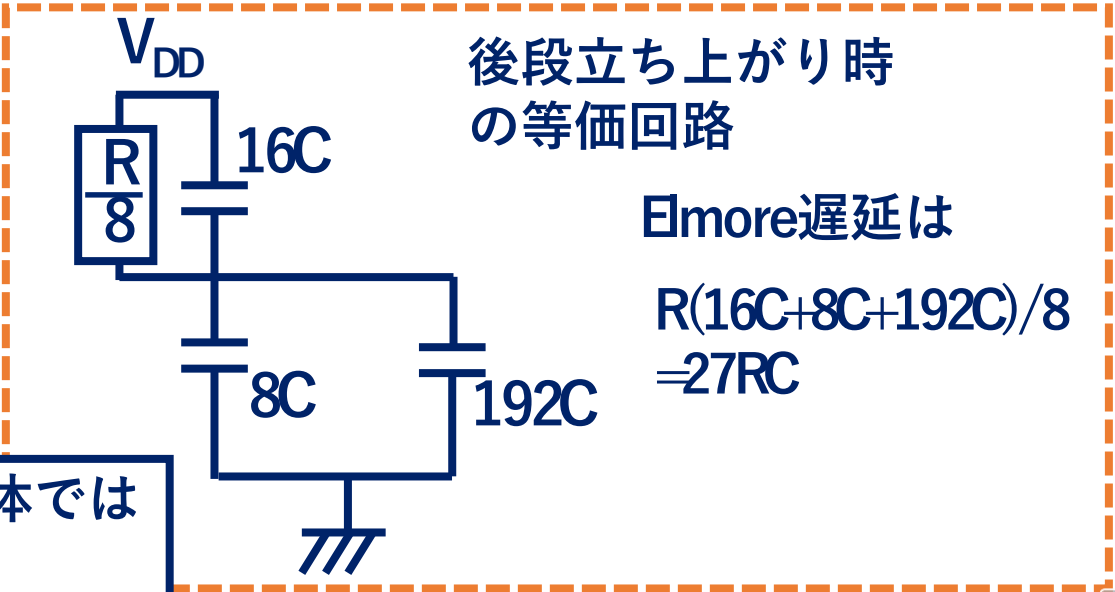
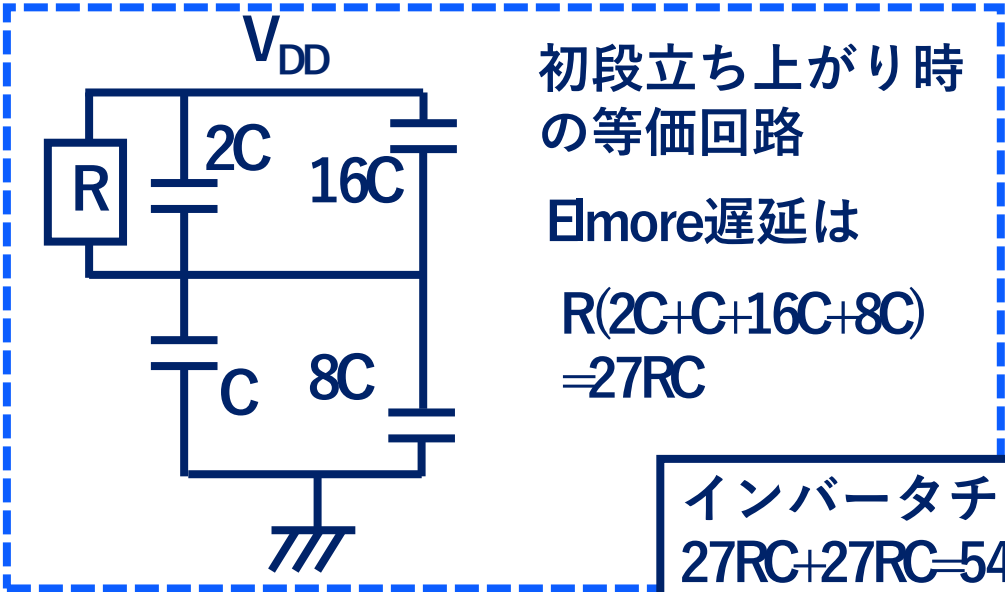
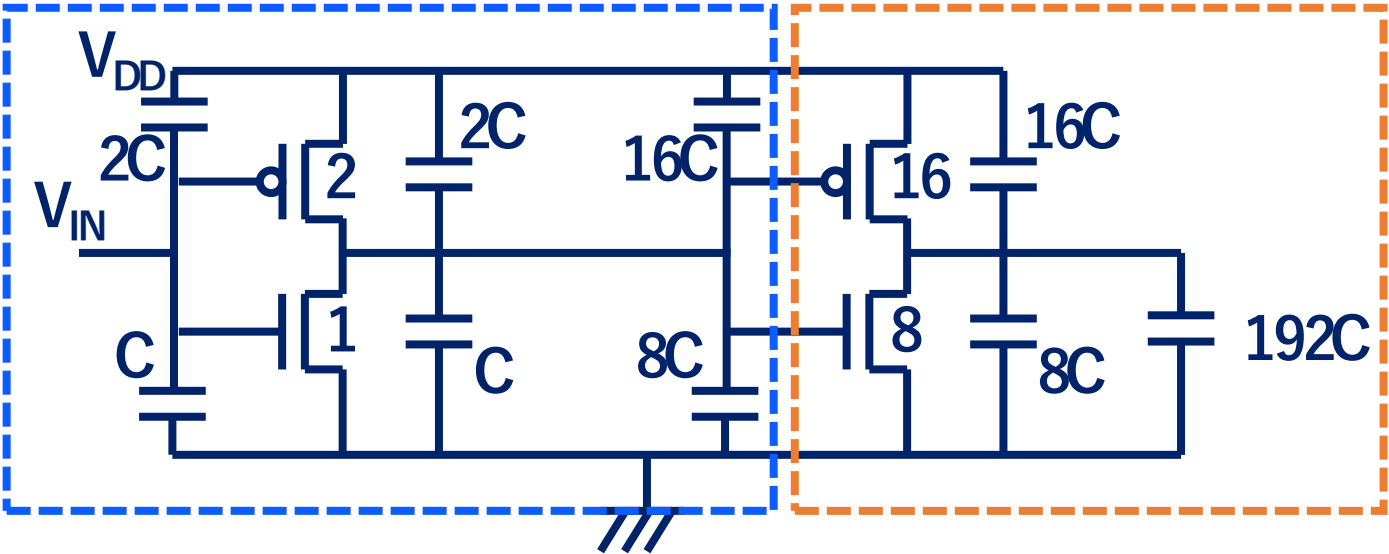
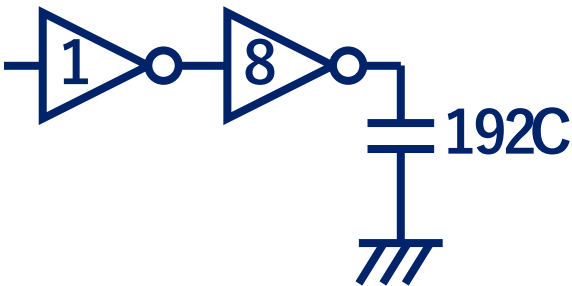
□ 1段で駆動する場合



- 立ち上がり遅延 $(3C+64\cdot 3C)R=195RC$
- 立ち下がり遅延 $(3C+64\cdot 3C)R=195RC$

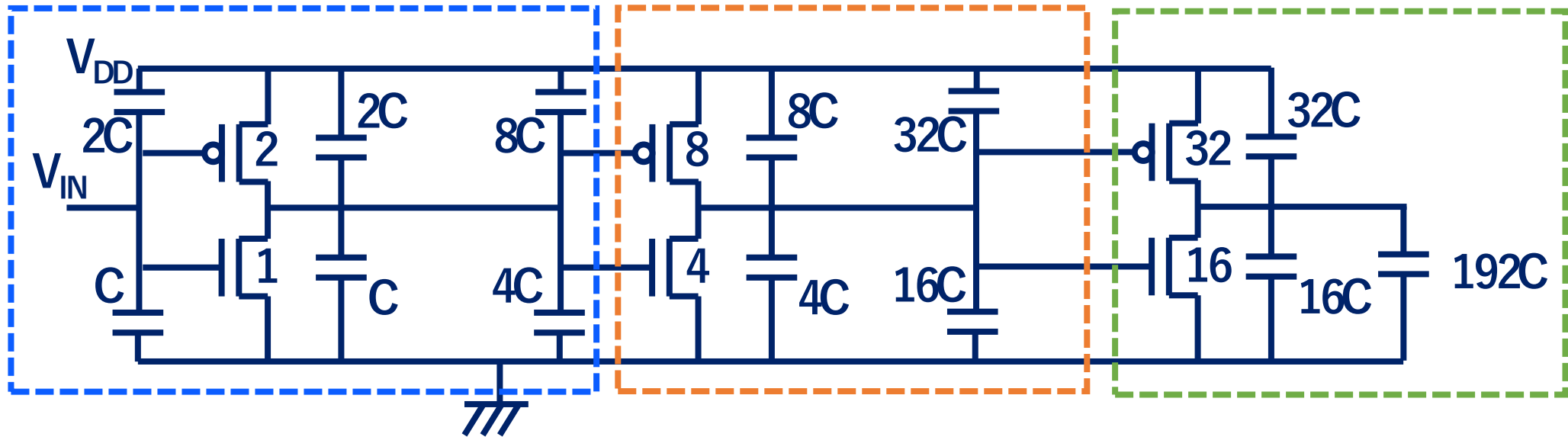
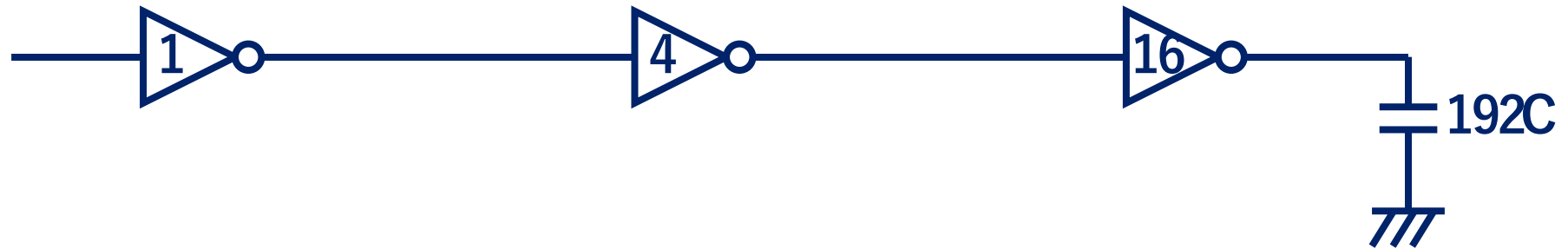
〔nMOSとpMOSの等価抵抗が等しくなるようにサイジングしているので立ち上がり・立ち下がり遅延は等しくなるはず〕

2段で駆動する場合



インバータチェーン全体では
 $27RC+27RC=54RC$ の遅延

3段で駆動する場合



$$(2C + C + 8C + 4C)R = 15RC$$

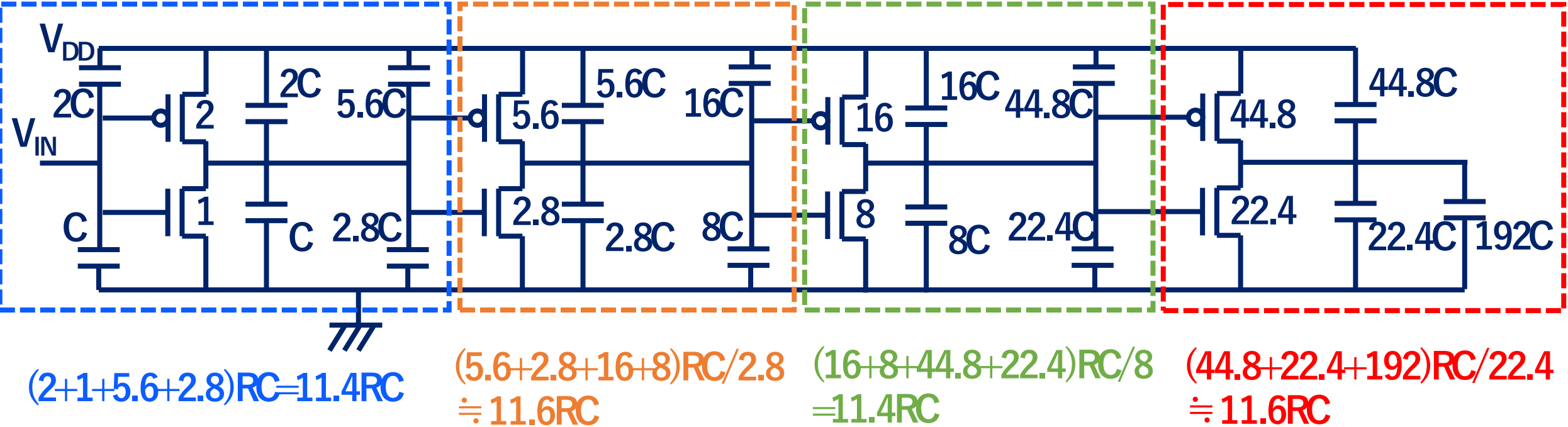
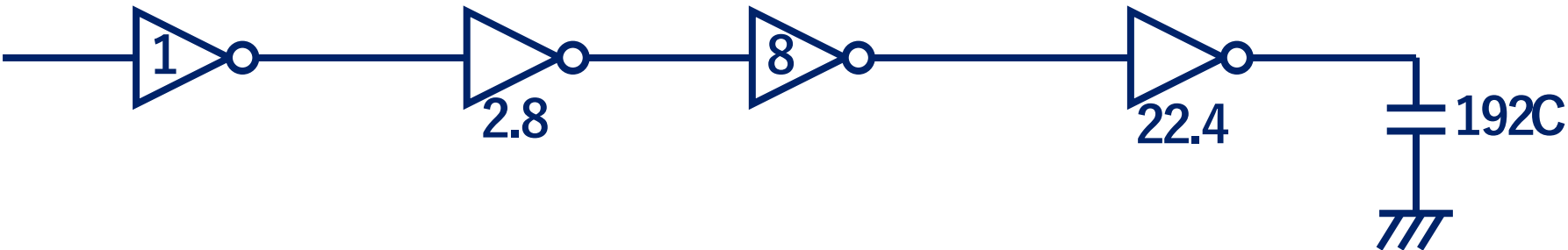
$$(8C + 4C + 32C + 16C)R/4 = 15RC$$

$$(32C + 16C + 192C)R/16 = 15RC$$

全体の遅延は $15RC \times 3 = 45RC$

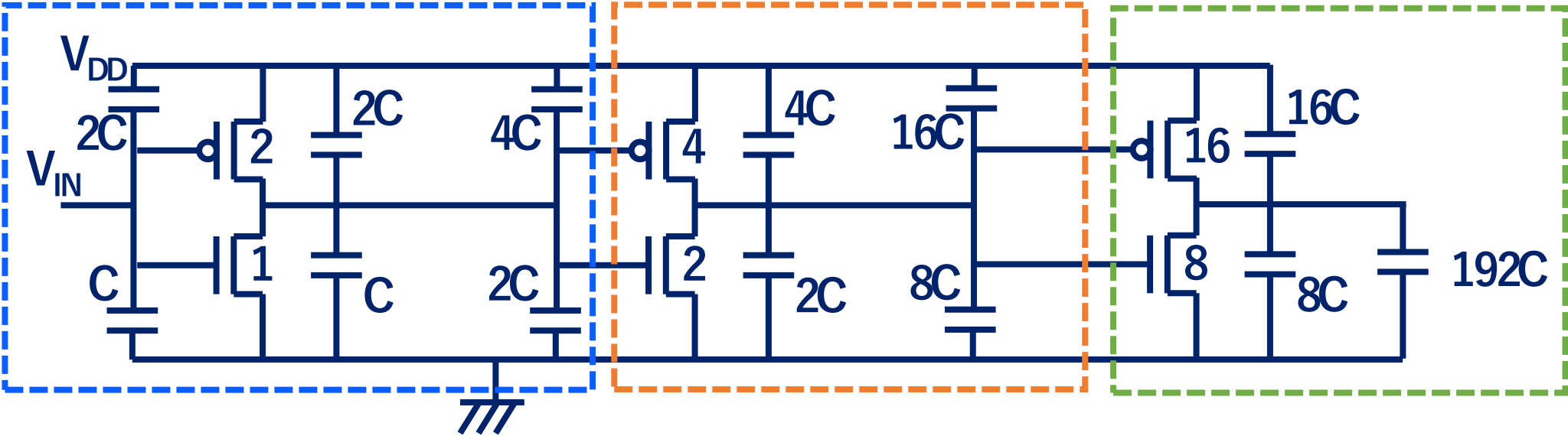
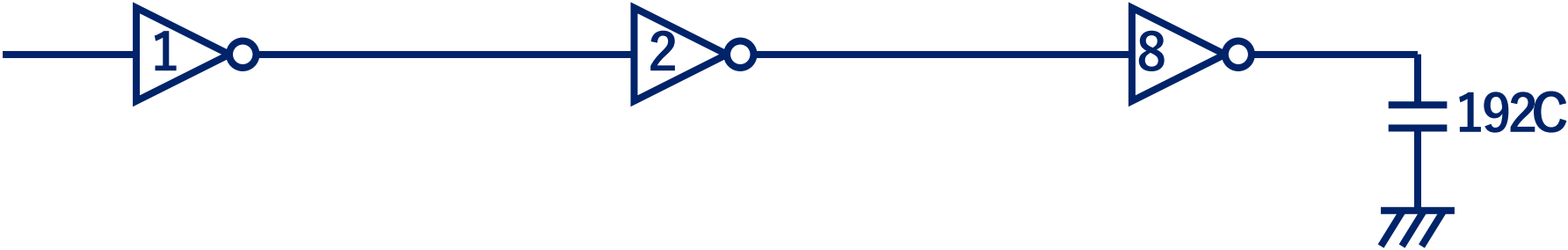


4段で駆動する場合



全体の遅延は46RC

3段で非均一なfan-outで駆動した場合



$(2C+C+4C+2C)R=9RC$

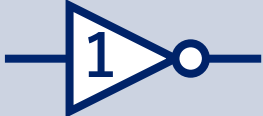



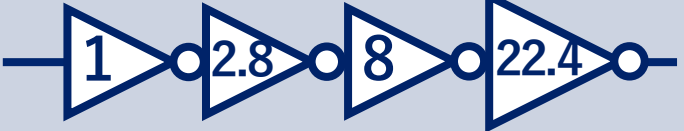
$(4C+2C+16C+8C)R/2=15RC$

$(16C+8C+192C)R/8=27RC$

全体の遅延は 51RC



結果をまとめると

回路	段数	fan-out	1段あたりの遅延	遅延合計
	1	64	195	195RC
	2	8	27	54RC
	3	4	15	45RC
	3	2/4/8	9/15/27	51RC
	4	2.8	11.4	45.8RC

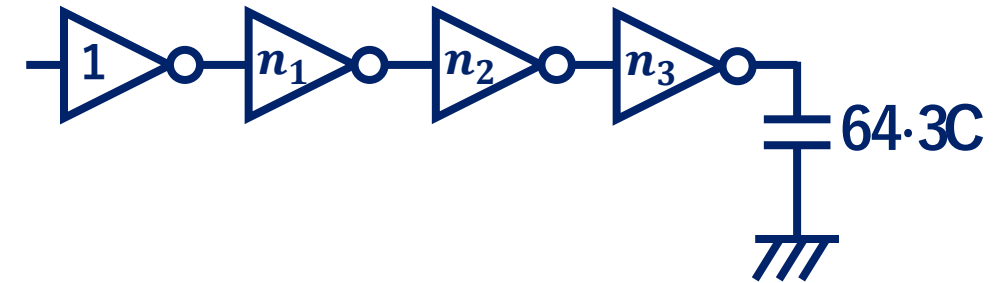
最短の遅延を得るためには

(1) 各段のfan-outを揃える ・ (2) fan-outは4程度に抑える のが良さそう



4段で駆動する場合のサイジングを真面目に解いてみる

- 右図のような状況で遅延が最小となるように n_1, n_2, n_3 を決める



- Elmore遅延は

$$t = (3C + 3Cn_1)R + (3Cn_1 + 3Cn_2)\frac{R}{n_1} + (3Cn_3 + 64 \cdot 3C)\frac{R}{n_3}$$

となるので、Elmore遅延の n_1, n_2, n_3 に関する微分=0を解けば良い

$$\square \frac{\partial t}{\partial n_1} = 3RC \left(1 + n_1 + \frac{1}{n_1} (n_1 + n_2) \right) = 0$$

$$\square \frac{\partial t}{\partial n_2} = 3RC \left(\frac{1}{n_1} - \frac{1}{n_2^2} (n_2 + n_3) + \frac{1}{n_2} \right) = 0$$

$$\square \frac{\partial t}{\partial n_3} = 3RC \left(\frac{1}{n_2} - \frac{1}{n_3^2} (n_3 + 64) + \frac{1}{n_3} \right) = 0$$

$$\Rightarrow \begin{cases} (1) n_2 = n_1^2 \\ (2) n_3 = \frac{1}{n_1} n_2^2 \\ (3) n_3^2 = 64 n_2 \end{cases}$$

(2)を(3)に代入すると

$$\left(\frac{n_2^2}{n_1} \right)^2 = 64 n_2 \Rightarrow n_2^3 = 64 n_1^2 \dots (4)$$

(1)を(4)に代入すると

$$n_1^6 = 64 n_1^2 \Rightarrow \begin{aligned} n_1 &= 64^{1/4} \\ n_2 &= 64^{1/2} \\ n_3 &= 64^{3/4} \end{aligned}$$

前ページの予想通り、インバータを同じ比率 ($\sqrt[4]{64}$) で大きくすると遅延を最小化出来る



駆動段数の最適化

- fan-outが一定 ($f = \sqrt[N]{F}$) という条件下でインバータ段数の最適値を求める

- インバータ1段分のElmore遅延は

$$(3Cf^i + 3Cf^{i+1}) \frac{R}{f^i} = 3CR(1 + f)$$

- 全体のElmore遅延は

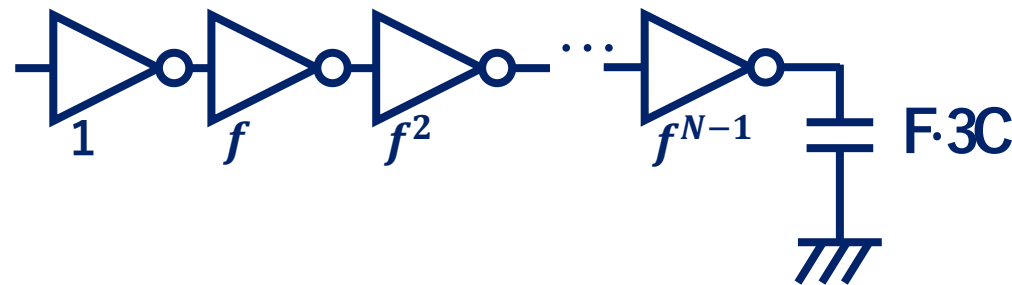
$$t = 3CRN(1 + f) = 3CRN(1 + \sqrt[N]{F}) = 3CRN \left(1 + \exp \left(\frac{\ln F}{N} \right) \right)$$

これを段数 N について最小化したいので、 t の N に関する微分が0となる N を求めれば良い

$$\frac{dt}{dN} = 3CR(1 + \sqrt[N]{F}) + 3CRN \cdot \left(-\frac{1}{N^2} \right) \cdot \ln F \cdot \sqrt[N]{F} = 0 \quad 1 + \sqrt[N]{F} - \sqrt[N]{F} \cdot \ln \sqrt[N]{F} = 0$$

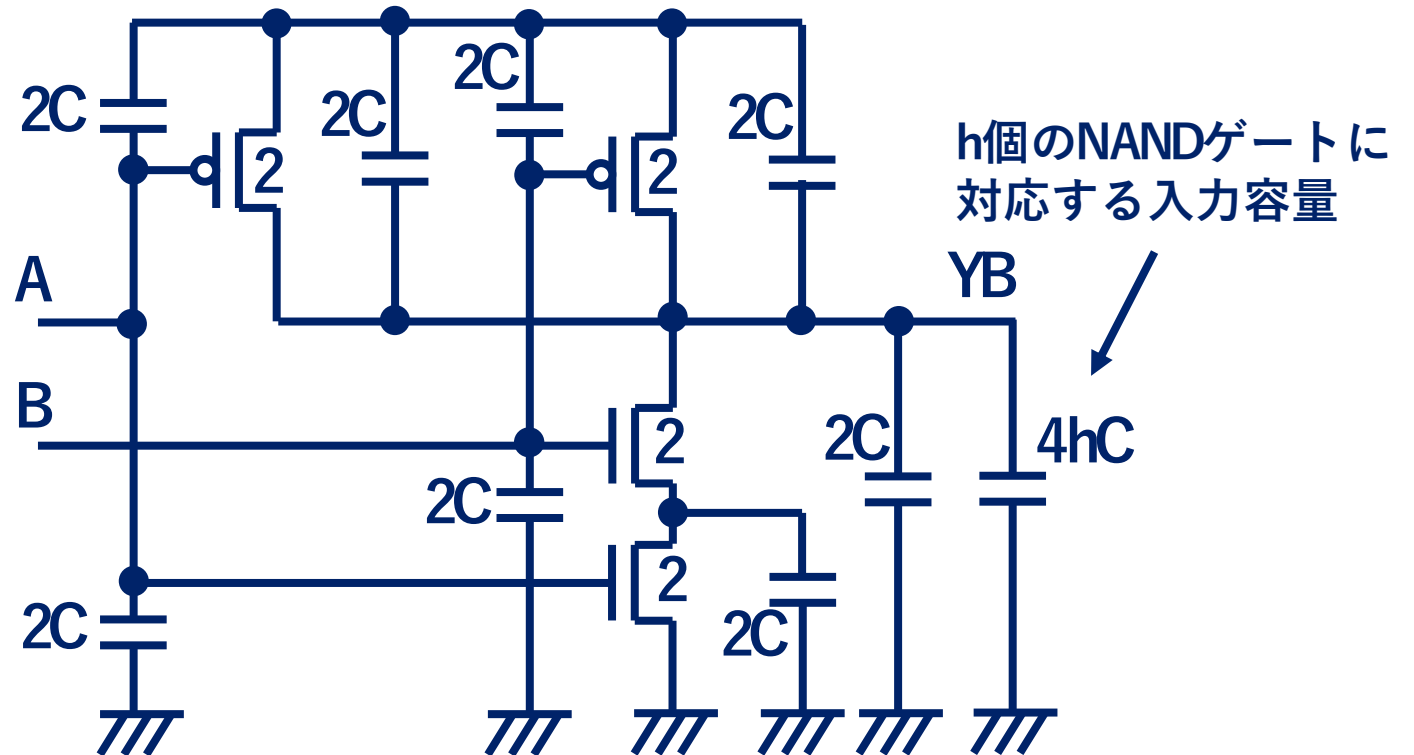
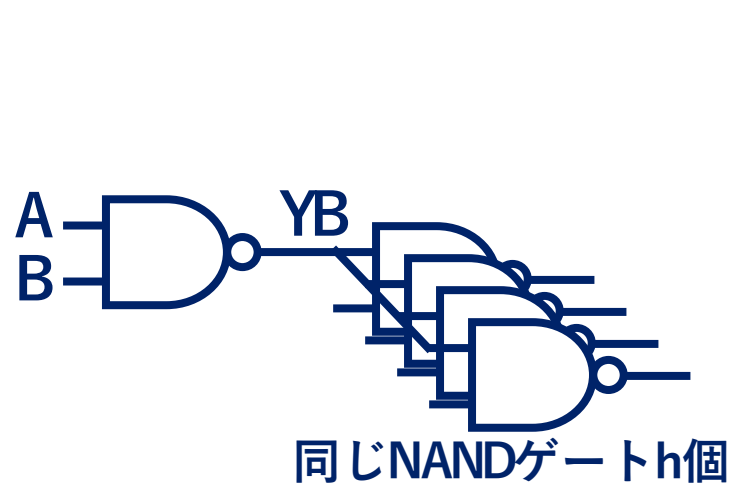
これは解析的な形で解けないので、“1”を無視する（インバータのソース・ドレイン容量に対応）すると

$\sqrt[N]{F} - \sqrt[N]{F} \cdot \ln \sqrt[N]{F} = 0$ となって、インバータ段数の最適値は $N_{opt} = \ln F$ となる



インバータ以外の遅延：2入力NAND

- 自分と同じゲートを h 個駆動しているNANDの遅延を求めてみる



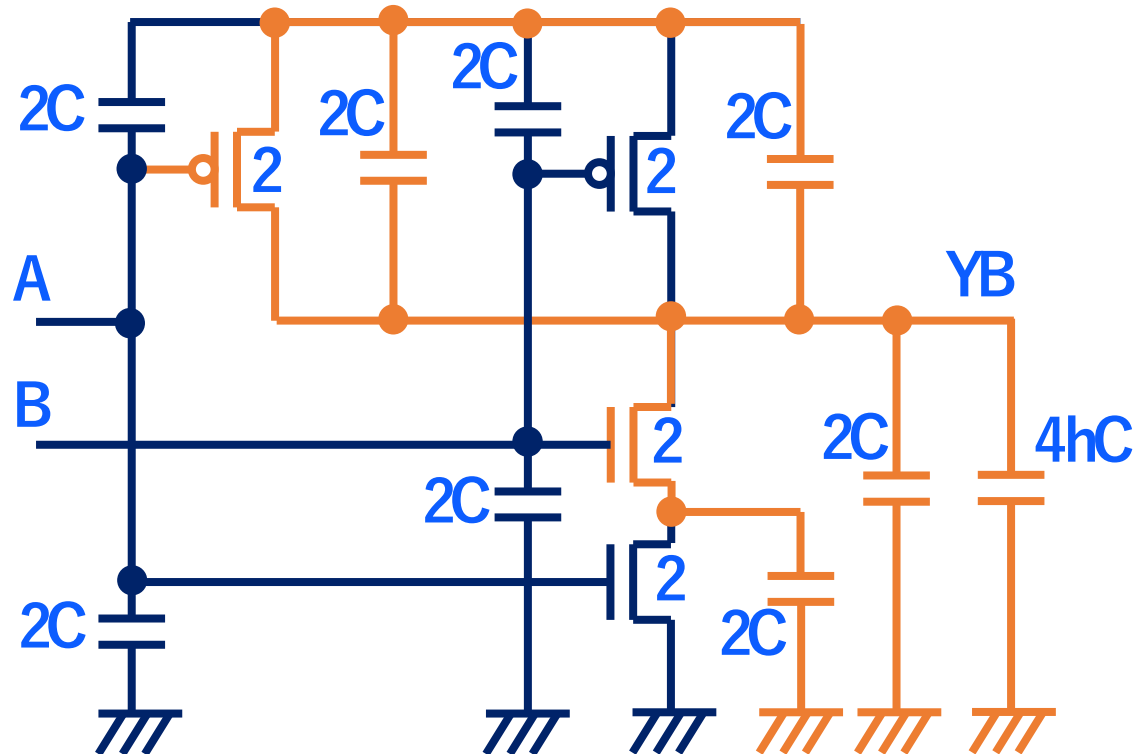
寄生容量を考慮した等価回路



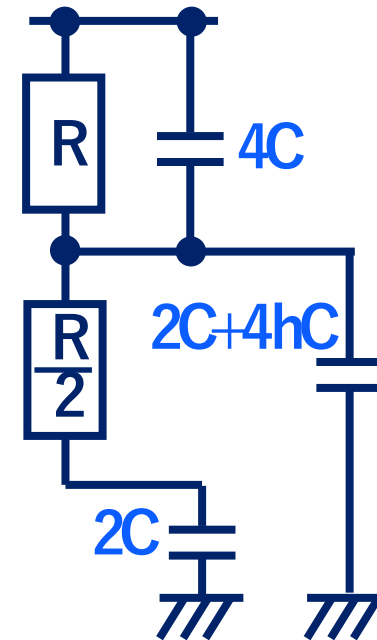
インバータ以外の遅延：2入力NANDの立ち上がり遅延

最悪遅延を考える（B="1"の状態ではAが"1"→"0"と変化）

- 並列接続されたpMOSは1つだけON
- nMOSも1つON⇒nMOSのドレイン容量も遅延に影響



単純化
すると

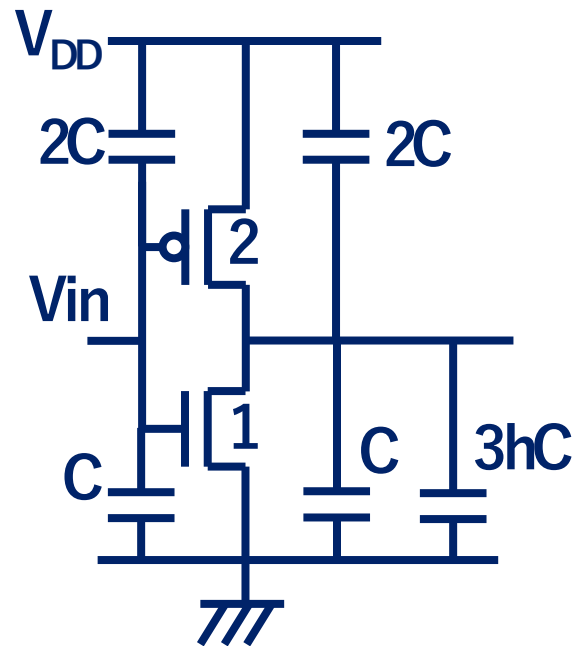


Elmore遅延：

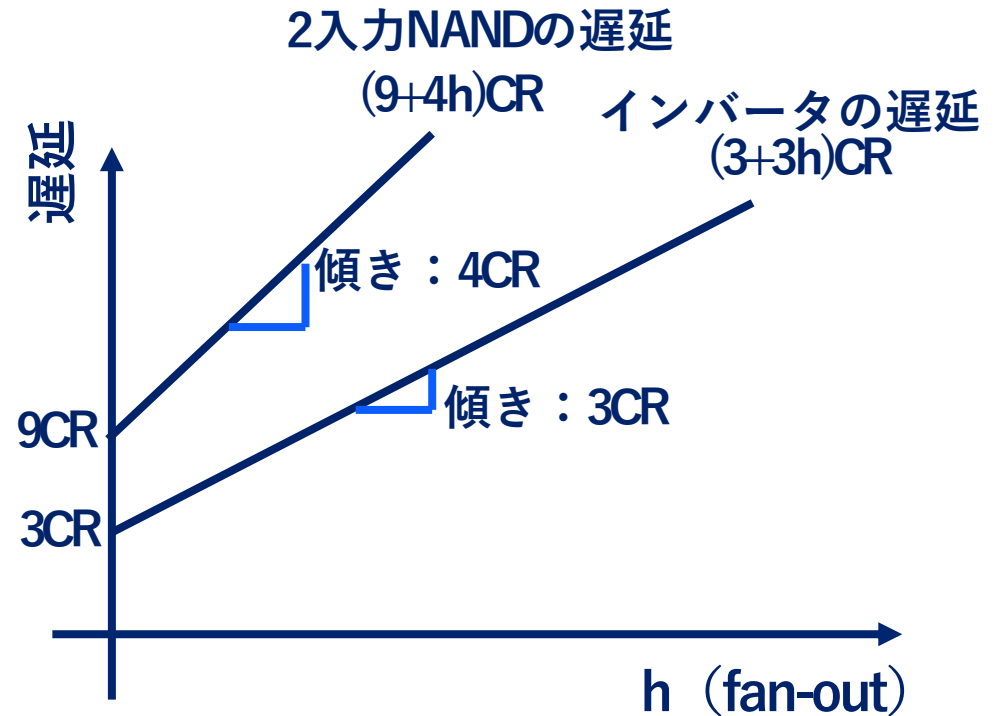
$$(6 + 4h)CR + 2C \left(R + \frac{R}{2} \right) = (9 + 4h)CR$$

インバータの遅延と比較してみる

- 自分と同じゲートを h 個駆動しているインバータの遅延



インバータのElmore
遅延： $(3+3h)CR$



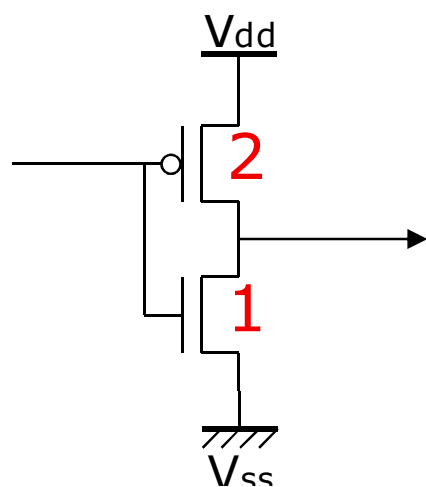
- 2入力NANDの方がfan-out数の増加に対する遅延増加割合が大きい
⇒論理反転するだけのインバータに比べて、複雑な論理演算をすることの代償
- インバータに比べて、どの程度“論理的な努力”を必要としているかの指標：Logical Effort

Logical Effortの定義

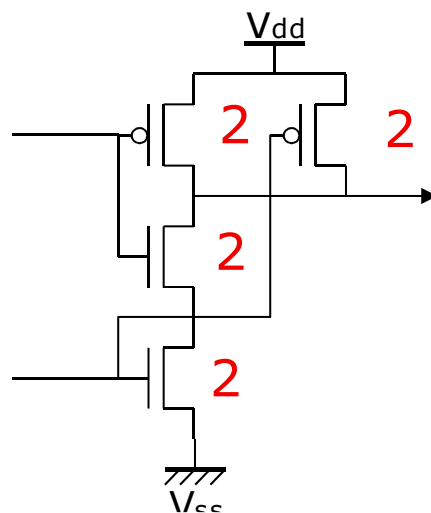
Logical Effort (LE) :

同じ駆動力を持つインバータと比較した時の入力容量の比

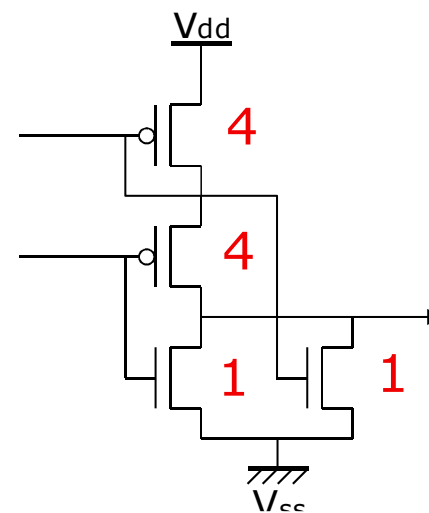
$$= \frac{\text{対象となる論理ゲートの入力容量}}{\text{インバータの入力容量}}$$



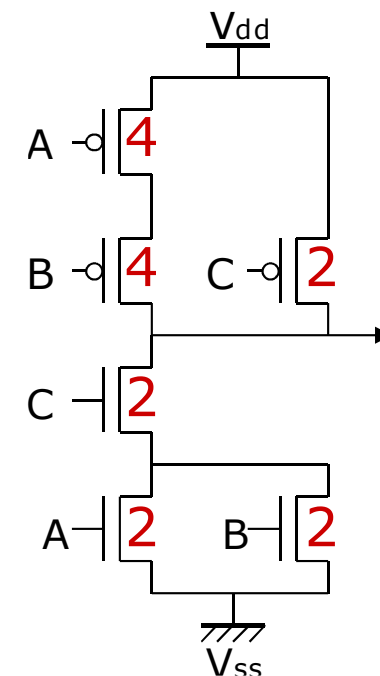
インバータ
LE=1



2入力NAND
LE=4/3



2入力NOR
LE=5/3



OR-AND-INV
LE=2 (A · B)
LE=4/3 (C)

- Logical Effortはゲートの複雑度に応じて増加
- また、NANDよりもNORの方が一般的には遅い

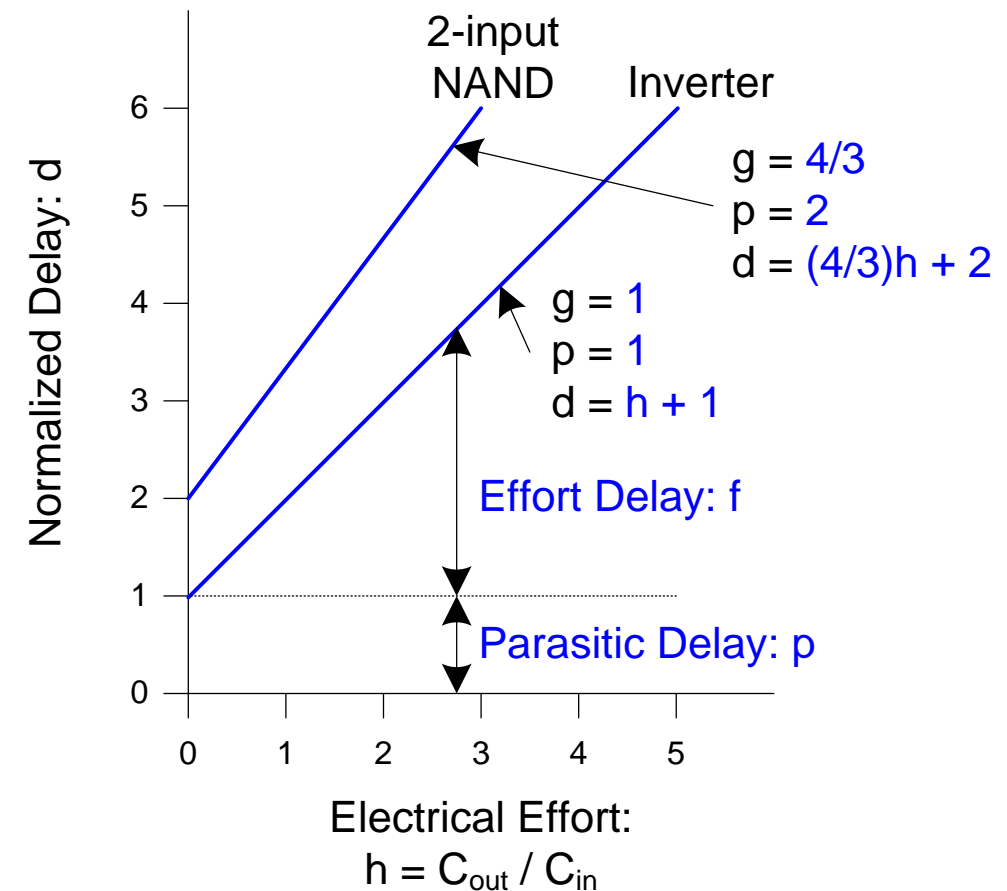


一般の論理ゲートに使える遅延モデル

- 論理ゲートの遅延時間 d はfan-outに比例する成分とfan-outとは無関係に決まる成分で表現できる

$$\text{遅延: } d = gh + p$$

- g : logical effort
- h : fan-out (electrical effortとも言う)
 - 入力容量 C_{in} に対する出力容量 C_{out} の比 ($=C_{out}/C_{in}$)
- p : parasitic delay



代表的な論理ゲートのLogical EffortとParasitic Delay

Logical Effort

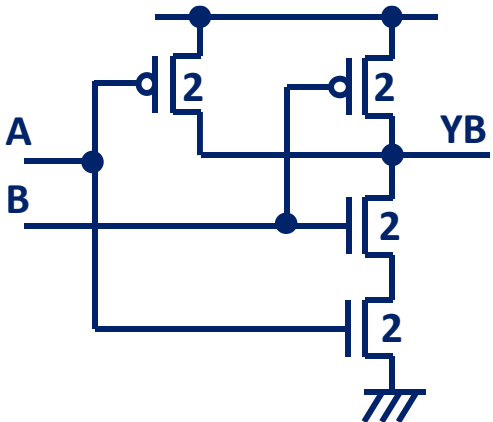
Gate type	Number of inputs				
	1	2	3	4	n
Inverter	1				
NAND		4/3	5/3	6/3	$(n+2)/3$
NOR		5/3	7/3	9/3	$(2n+1)/3$
XOR, XNOR		4, 4	6, 12, 6	8, 16, 16, 8	

入力によってLogical Effortが異なる

Parasitic delay

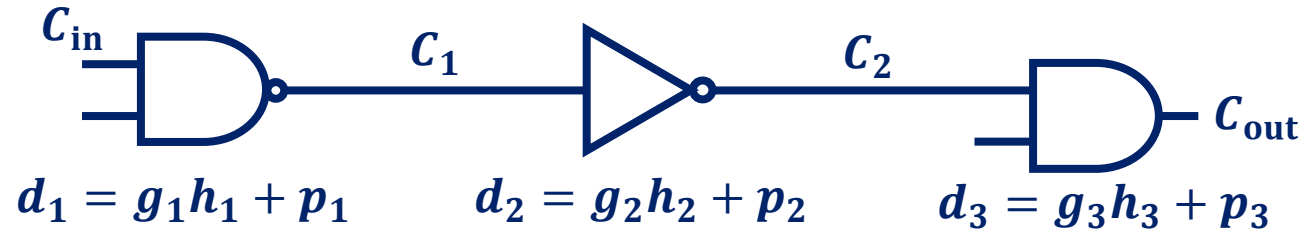
Gate type	Number of inputs				
	1	2	3	4	n
Inverter	1				
NAND		2	3	4	n
NOR		2	3	4	n
XOR, XNOR		4	6	8	

手計算向きな簡易的な方法での推定値



- 出力ノード（YB）に繋がっているドレイン容量を加算
- 左の2入力NANDの場合は2倍のサイズのpMOS・nMOSが3個繋がっているの、6RC
- インバータの寄生容量である3RCで正規化すると“2”

直列接続されたゲートの遅延時間



$$\text{全体の遅延: } D = \sum_i \underline{g_i h_i} + \sum_i \underline{p_i}$$

Effort delay :
トランジスタのサイジングで小さくできる部分

Parasitic delay :
使うゲートの種類によって一意に決まる部分

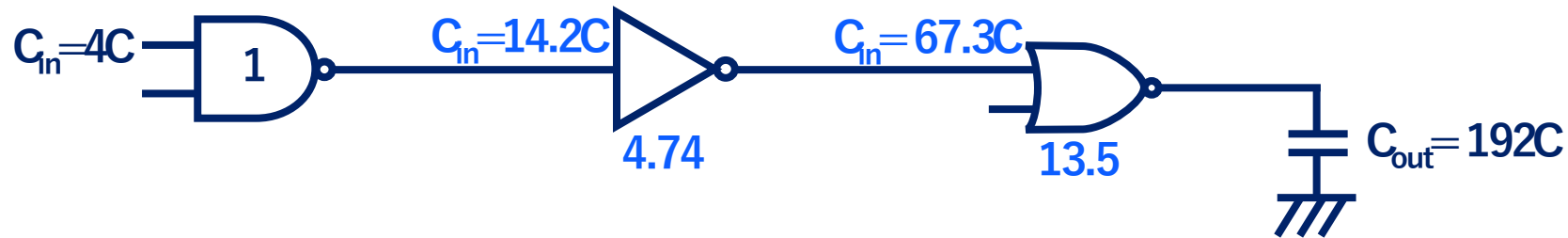
- 相加相乗平均の関係から $\frac{1}{N} \sum_i g_i h_i \geq \left(\prod_i g_i h_i \right)^{1/N}$ であるので、遅延Dが最小になるのは等号が成立時

$$g_1 h_1 = g_2 h_2 = \dots = g_N h_N = \left(\prod_i g_i \prod_i h_i \right)^{1/N} \text{ である}$$

- 以上から g と $\prod_i h_i \left(= \frac{C_1}{C_{in}} \frac{C_2}{C_1} \dots \frac{C_{out}}{C_{N-1}} = \frac{C_{out}}{C_{in}} \right)$ なので入力容量と出力容量の比 $\left[\right]$ が既知の時にDを最小にする h_i の値を決めることが出来る

遅延最小化の例

- 入力容量が“4C”のNANDで64倍の出力容量を駆動する場合を考える



- NAND・INV・NORのLogical Effortは $4/3 \cdot 1 \cdot 5/3$ なので1段あたりのEffort delayは

$$\sqrt[N]{\frac{C_{out}}{C_{in}} \prod_i g_i} = \sqrt[3]{\frac{192}{4} \cdot \frac{4}{3} \cdot 1 \cdot \frac{5}{3}} = 4.74$$

- 初段のfan-out: $h_1 = 4.74 \div \frac{4}{3} = 3.55$
 - 2段目の入力容量は $4C \times 3.55 = 14.2C$
 - 単位インバータの入力容量は $3C$ だったので、入力容量が $14.2C$ となるためには $14.2C/3C = 4.74$ 倍のサイズであれば良い
- 2段目のfan-out: $h_2 = 4.74 \div 1 = 4.74$
 - 3段目の入力容量は $14.2C \times 4.74 = 67.3C$
 - 単位NORの入力容量は $5C$ なので、 $67.3C/5C = 13.5$ 倍のサイズ
- 3段目のfan-out: $h_3 = 4.74 \div \frac{5}{3} = 2.84$

