

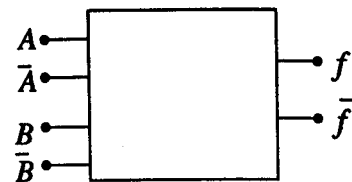
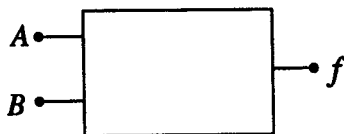
F8: Logisk och fysisk konstruktion av CMOS grindar

- Målsättning:
 - Peka ut de faktorer som har störst betydelse vid optimering av kombinatorik på logisk nivå för hastighet
- Innehåll:
 - CVSL (differentiell logik)
 - TSPC (true single phase clocking)
 - Fan-in / Fan-out
 - Typiska fördröjningar i nand- och nor-grindar

1 (3 3)

CVSL (Cascode Voltage Switch Logic)

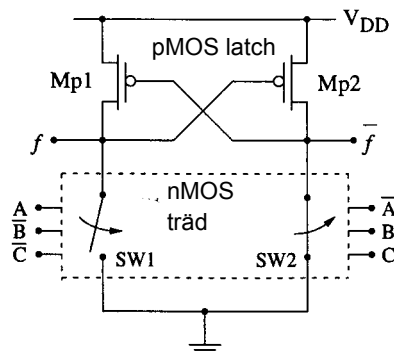
- Single Rail Logic
 - Tills nu har vi sett på single rail logic där ett logisk tillstånd representeras av en variabel som har värdet 0 eller 1
 - En logisk grind med funktionen f så att $f(a)$ är ingången till nästa grind
- Dual Rail Logic
 - använder både variabeln och dess komplement ($a, \bar{a}$) som ingångar
 - utgången från en dual rail grind är också paret ($f, \bar{f}$) som driver nästa grind
 - Dual rail logic tolkar differensen ($f - \bar{f}$) som det logiska värdet (och inte var och en var för sig)
 - Kallas också för differentiell logik (DVSL)



2 (3 3)

CVSL

• Grundläggande grind



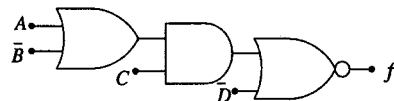
- då $\bar{f} = 0$:
nMOS trädets leder för $\bar{f}$. $\rightarrow$ Mp1 sluts och $f \rightarrow 1$
- då $f = 0$:
nMOS trädets leder för f . $\rightarrow$ Mp2 sluts och $\bar{f} \rightarrow 1$

3 (3 3)

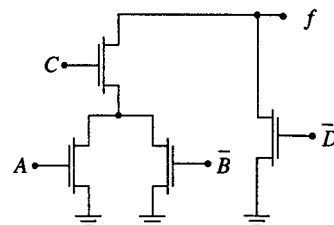
DCVSL exempel (OAOI grind)

Funktion: $f = \overline{(A + \bar{B}) \cdot C + D}$

Skapa $\bar{f}$ med komplementen av invariablerna: $\bar{f} = (A + \bar{B}) \cdot C + D$



OAOI logikschema

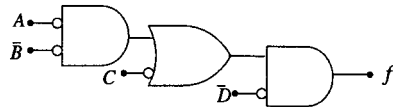


nMOS logikekvivalent

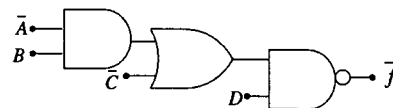
4 (3 3)

DCVSL exempel forts.

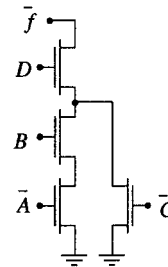
Skapa komplementet till f : $\bar{f} = (A + \bar{B}) \cdot C + \bar{D}$



Efter deMorgans reduktion

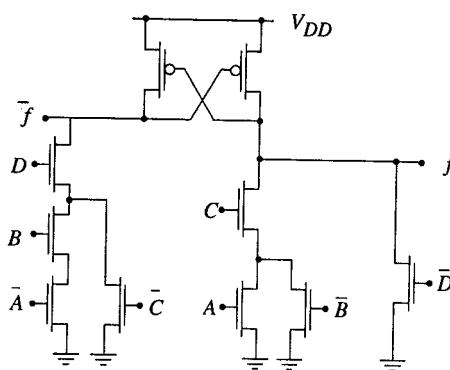


komplementen av invariablerna



nMOS logikekvivalent

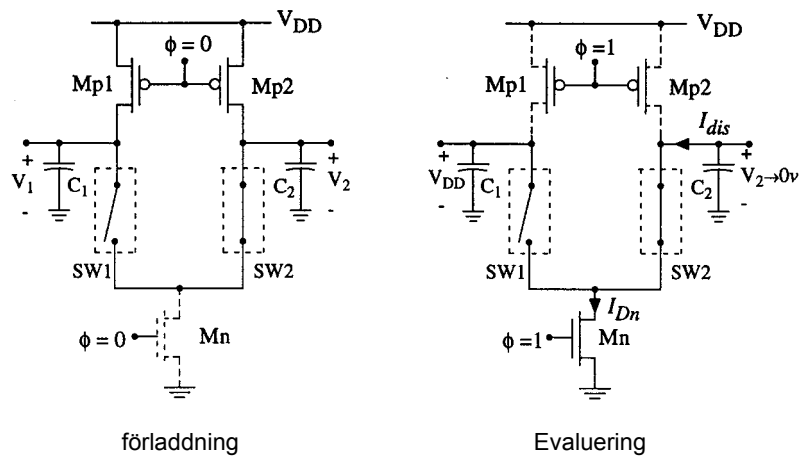
DCVSL exempel forts.



Komplett logisk grind

DCVSL varianter

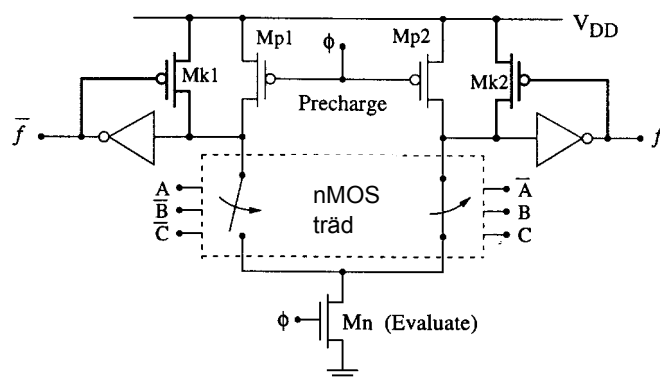
- Dynamisk (klockad)



7 (3 3)

DCVSL varianter

- Statisk (klockad)



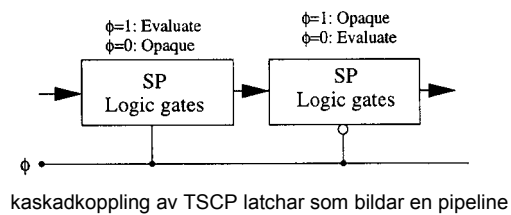
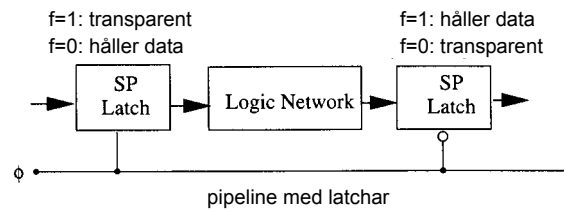
M_{k1} och M_{k2} är "keeper" transistorer som gör de dynamiska utgångarna statiska.

M_{k1} och M_{k2} måste vara svaga, d.v.s små W/L förhållanden.

8 (3 3)

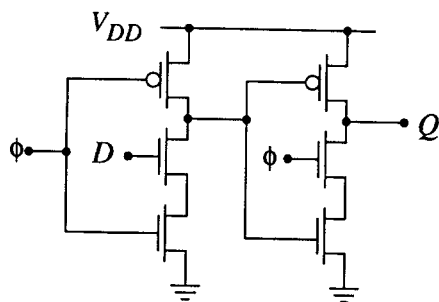
TSPC (True Single Phase Clocking)

- Med enfasklockning förenklas
 - klockgenereringen
 - klockdistribution
- Kan leda till högre klockfrekvenser

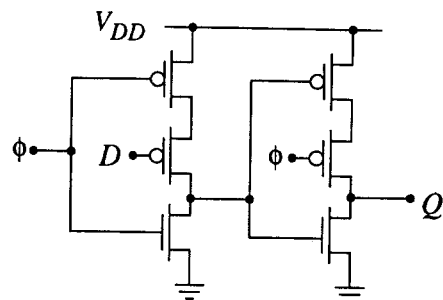


9 (3 3)

TSPC latchar



nMOS latch

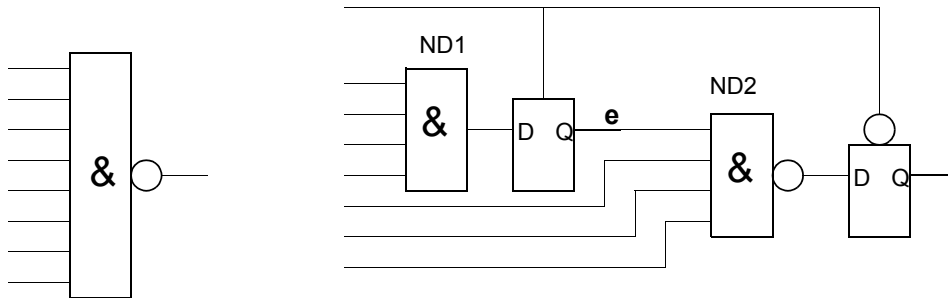


pMOS latch

10 (3 3)

Exempel: 8-ingångars NAND grind i TSPC

- TSPC medger halv klockcykels pipelining

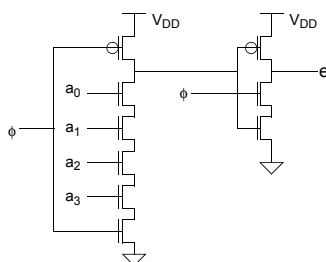


- Beräkningen av NAND funktionen delas upp i två steg
 - under halva klockcykeln beräknar ND1 fram delresultatet e
 - under nästa halva klockcykel beräknar ND2 fram slutresultatet

13 (33)

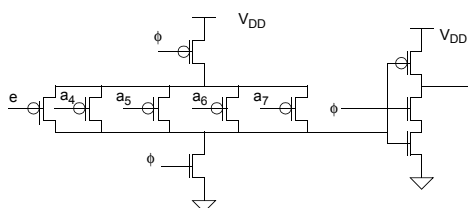
Exempel: 8-ingångars NAND grind i TSPC

- Dela upp grinden i två delar: en nMOS latch och en pMOS latch
 - nMOS grind



$$e = a_0 \cdot a_1 \cdot a_2 \cdot a_3$$

- pMOS grind



$$f = \overline{\overline{a_4 \cdot a_5 \cdot a_6 \cdot a_7 \cdot e}}$$

→ ger en "OR" struktur i pMOS trädet.

14 (33)

Makromodellering

- Grindarna betraktas som fördröjningselement
- Grindarna karakteriseras i en kretssimulator (SPICE)
- Förenklad ekvation för grindfördröjningen ges av:

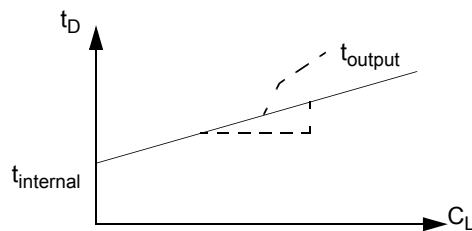
$$t_D = t_{\text{internal}} + t_{\text{output}} \times C_L$$

$$t_D = t_{\text{internal}} + t_{\text{output}} \cdot C_L$$

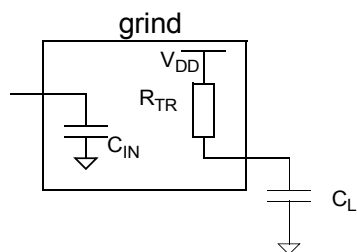
t_D = grindfördröjning

t_{internal} = grindfördröjning utan last ($C_L=0$)

t_{output} = grindfördröjning som kommer sig av pålagd last



Makromodellering



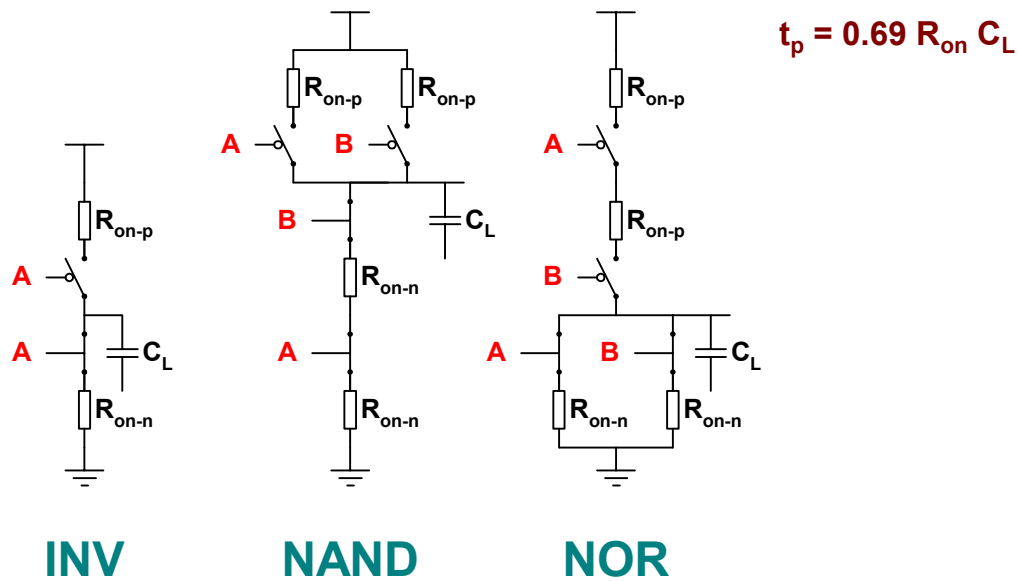
C_{in} är gatekapacitansen för nMOS (C_{Gn}) och pMOS (C_{Gp}) transistorerna: $C_{\text{in}} = C_{\text{g}} = C_{\text{Gn}} + C_{\text{Gp}}$

C_{intern} är summan av kapacitanserna på drain för pMOS och nMOS transistorerna som finns på utgången. $C_{\text{intern}} = C_{\text{Dn}} + C_{\text{Dp}}$

$$R_{\text{drive}} = 1/\beta(V_{\text{GS}} - V_{\text{T}})$$

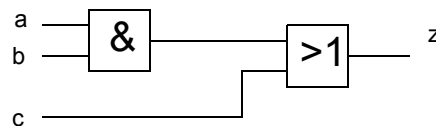
C_{g} och C_{D} är direkt proportionella mot transistorernas kanalbredder (W). (Naturligtvis en approximation)

Analys av grindfördröjning



17 (33)

Exempel på användning av makromodell



Varje nät har en last på 10fF utom utgången z som har 20fF

	$t_{i,rise}[ns]$	$t_{i,fall}[ns]$	$k_{load,rise}[ns/pF]$	$k_{load,fall}[ns/pF]$	$C_{in}[fF]$
and	0.24	0.26	2.12	3.83	7
or	0.20	0.30	2.40	3.50	9

and: $t_{Dr} = 0.24 + 2.12 \cdot (9+10) \cdot 10^{-3}$ $t_{Df} = 0.26 + 3.83 \cdot (9+10) \cdot 10^{-3}$

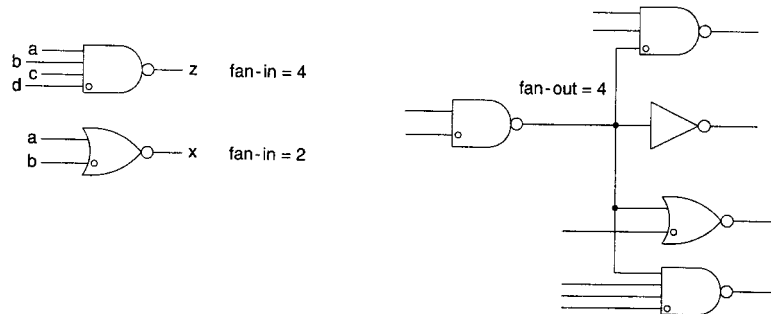
or: $t_{Dr} = 0.20 + 2.40 \cdot (20) \cdot 10^{-3}$ $t_{Df} = 0.30 + 3.50 \cdot (20) \cdot 10^{-3}$

$t_{Da-z,1} = 0.28 + 0.33 = 0.61 \text{ ns}$

$t_{Da-z,2} = 0.25 + 0.37 = 0.62 \text{ ns}$

18 (33)

Fan-in Fan-out

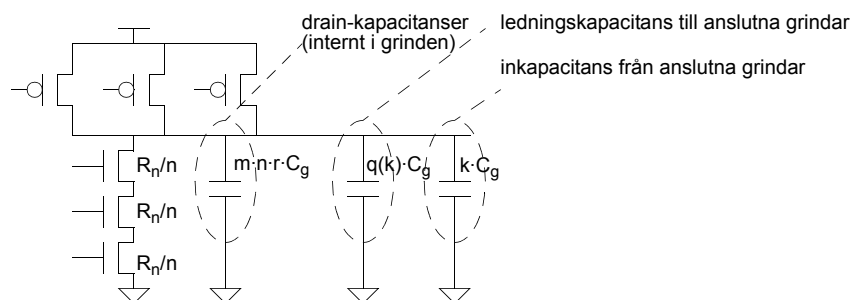


Falltid för NAND grindar

$$t_{Df} = m \cdot \frac{R_n}{n} \cdot (m \cdot n \cdot r \cdot C_g + q(k) \cdot C_g + k \cdot C_g)$$

$$t_{Df} = R_n \cdot C_g \cdot m^2 \cdot r + m \cdot k \cdot \frac{R_n \cdot C_g}{n} \cdot \left(1 + \frac{q(k)}{k}\right)$$

Exempel: 3-ingångars NAND (m=3)

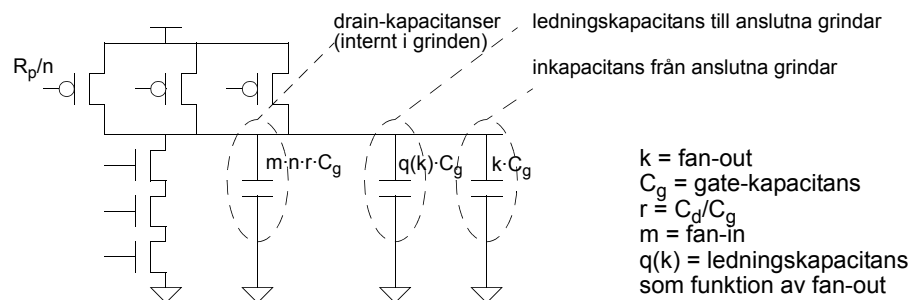


Stigtid för NAND grindar

$$t_{Dr} = \frac{R_p}{n} (m \cdot n \cdot r \cdot C_g + q(k) \cdot C_g + k \cdot C_g)$$

$$t_{Dr} = R_p \cdot C_g \cdot m \cdot r + k \cdot \frac{R_p \cdot C_g}{n} \cdot \left(1 + \frac{q(k)}{k}\right)$$

Exempel: 3-ingångars NAND (m=3)



21 (33)

Summering stig/falltider för NOR och NAND

• NAND

$$t_{Dr} = R_p \cdot C_g \cdot m \cdot r + k \cdot \frac{R_p \cdot C_g}{n} \cdot \left(1 + \frac{q(k)}{k}\right)$$

$$t_{Df} = R_n \cdot C_g \cdot m^2 \cdot r + m \cdot k \cdot \frac{R_n \cdot C_g}{n} \cdot \left(1 + \frac{q(k)}{k}\right)$$

• NOR

$$t_{Dr} = m \cdot \frac{R_p}{n} \cdot (m \cdot n \cdot r \cdot C_g + q(k) \cdot C_g + k \cdot C_g)$$

$$t_{Df} = \frac{R_n}{n} \cdot (m \cdot n \cdot r \cdot C_g + q(k) \cdot C_g + k \cdot C_g)$$

• Dimensionering av NAND-grind för symmetriskt omslag ($t_{Dr} = t_{Df}$)

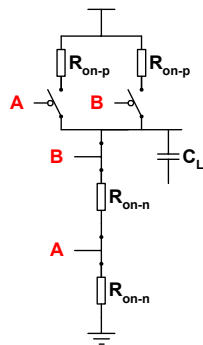
$$R_p = m \cdot R_n$$

alltså

$$\beta_p \cdot W_p = \frac{\beta_n \cdot W_n}{m}, \quad \text{där } L_n = L_p$$

22 (33)

Analys av grindfördröjning



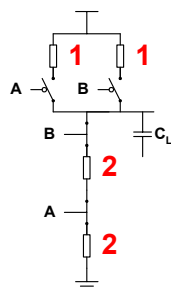
NAND

Three cases

1. Pull up of one PMOS, Worst case
 $t_{pLH} = 0.69R_pC_L$
2. Pull up of two PMOS at the same time
 $t_{pLH} = 0.69(R_p/2)C_L$
3. Pull down of NMOS
 $t_{pLH} = 0.69(2R_n)C_L$

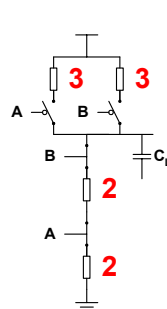
23 (33)

Konstruera för "värsta fallet"



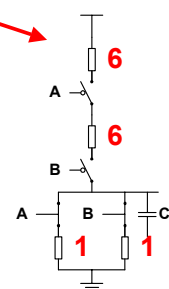
NAND

$$R_{on-p} = R_{on-n}$$



NAND

$$R_{on-p} = 3R_{on-n}$$



NOR

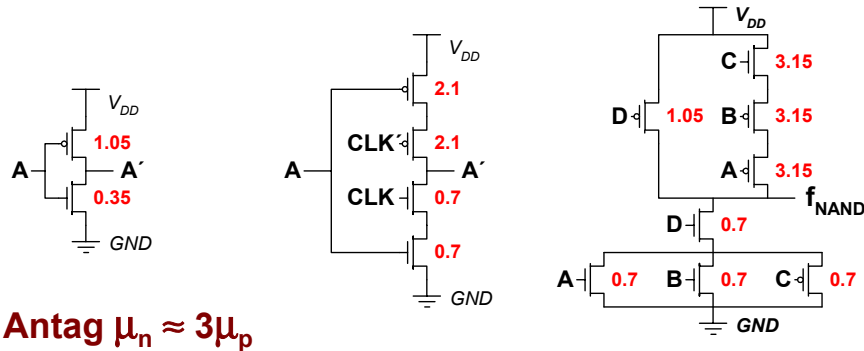
$$R_{on-p} = 3R_{on-n}$$

(For minimum sized transistors)

24 (33)

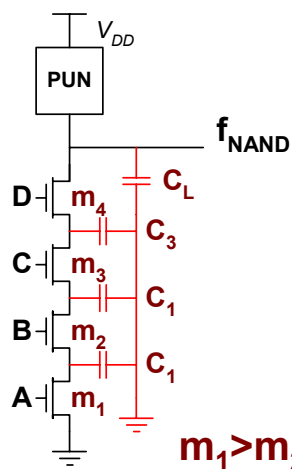
Transistordimensionering

- för samma drivförmåga för alla grindar
- för symmetriskt svar



25 (33)

Reducering av fördröjningen i komplexa grindar



Progressiv dimensionering:

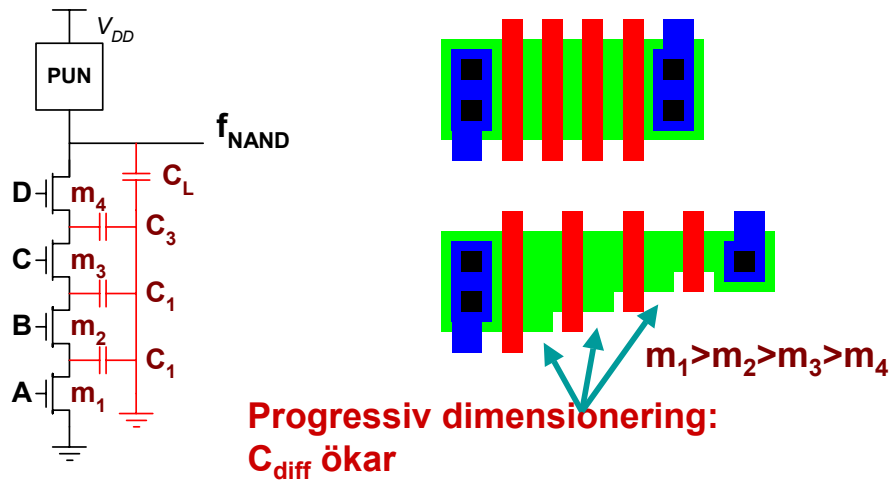
- m_4 laddar ur C_L
- m_3 laddar ur $C_L + C_3$
- m_2 laddar ur $C_L + C_3 + C_2$
- m_1 laddar ur $C_L + C_3 + C_2 + C_1$

Minska storlekarna nedåt

Kan ge reduction i
grindfördröjning med 30%!

26 (33)

Reducering av fördröjningen i komplexa grindar

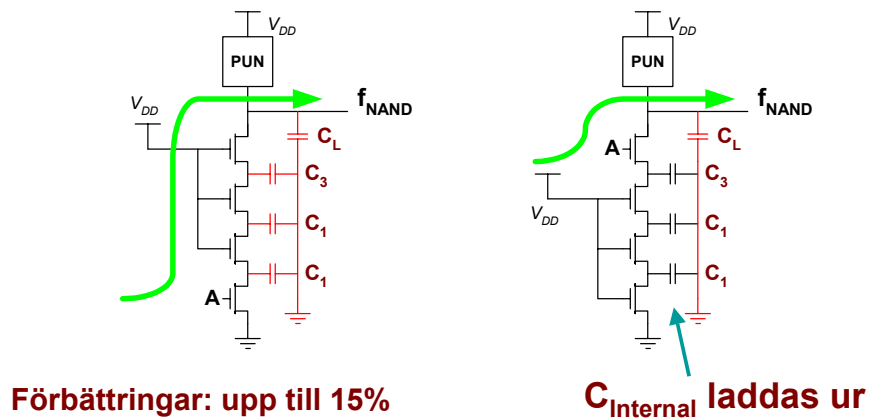


27 (33)

Reducering av fördröjningen i komplexa grindar

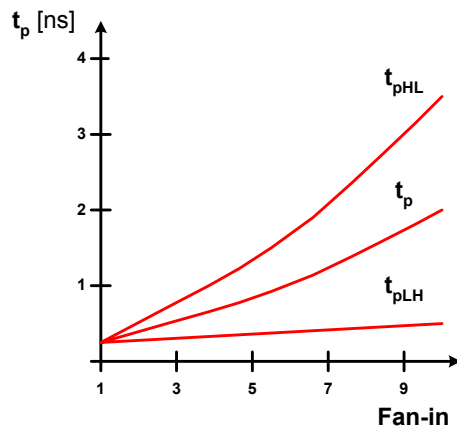
Transistor Ordning

Grindar i den kritiska signalvägen



28 (33)

Fördröjning som en funktion av fan-in



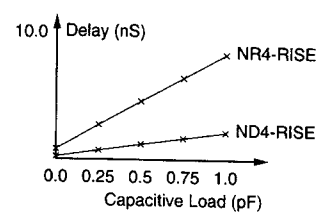
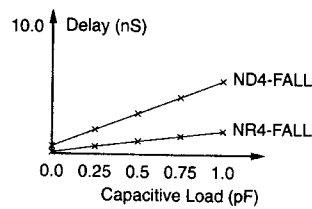
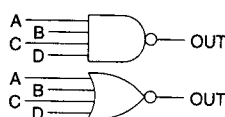
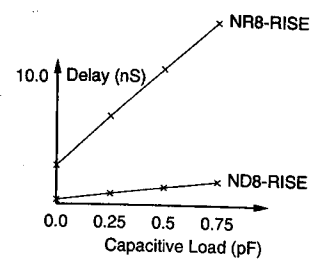
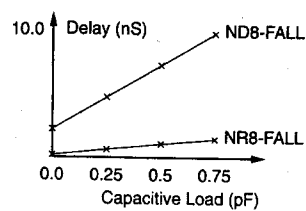
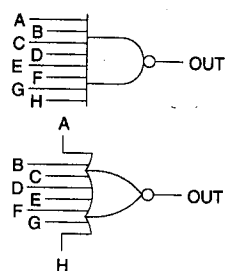
Prestanda försämras snabbt vid stora fan-in

Fan-in större än 3 till 4 undviks normalt

N-input NAND

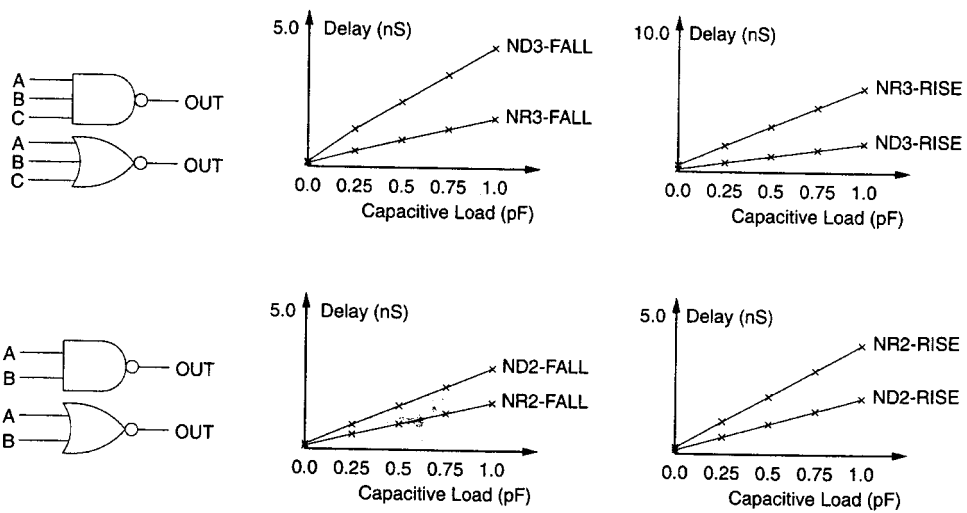
29 (33)

Typiska grindfördröjningar i NAND och NOR



30 (33)

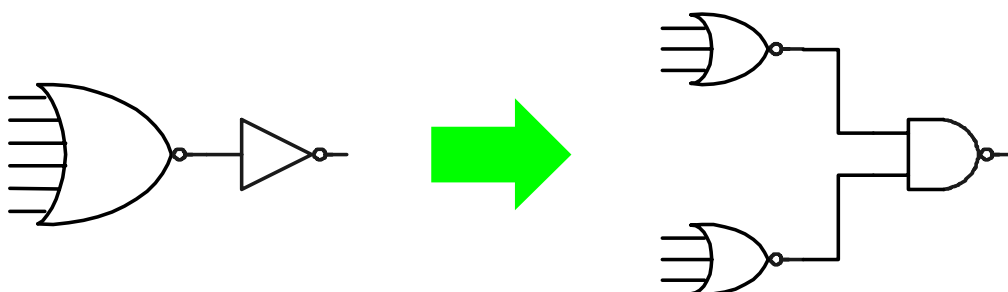
Typiska grindfördröjningar i NAND och NOR



31 (33)

Reducera fördröjningen med lågt fan-in

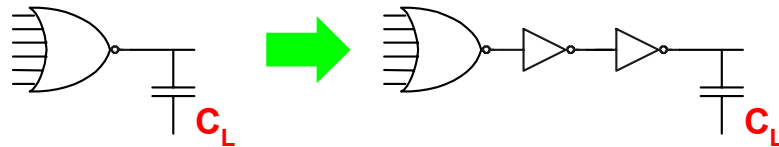
Reducera fan-in



32 (33)

Buffring av komplexa grindar

Om C_L (Fan-out) är stor alla transistorer måste skalas upp



Buffra: Isolera Fan- in från Fan- out

Endast buffer transistorer behöver skalas