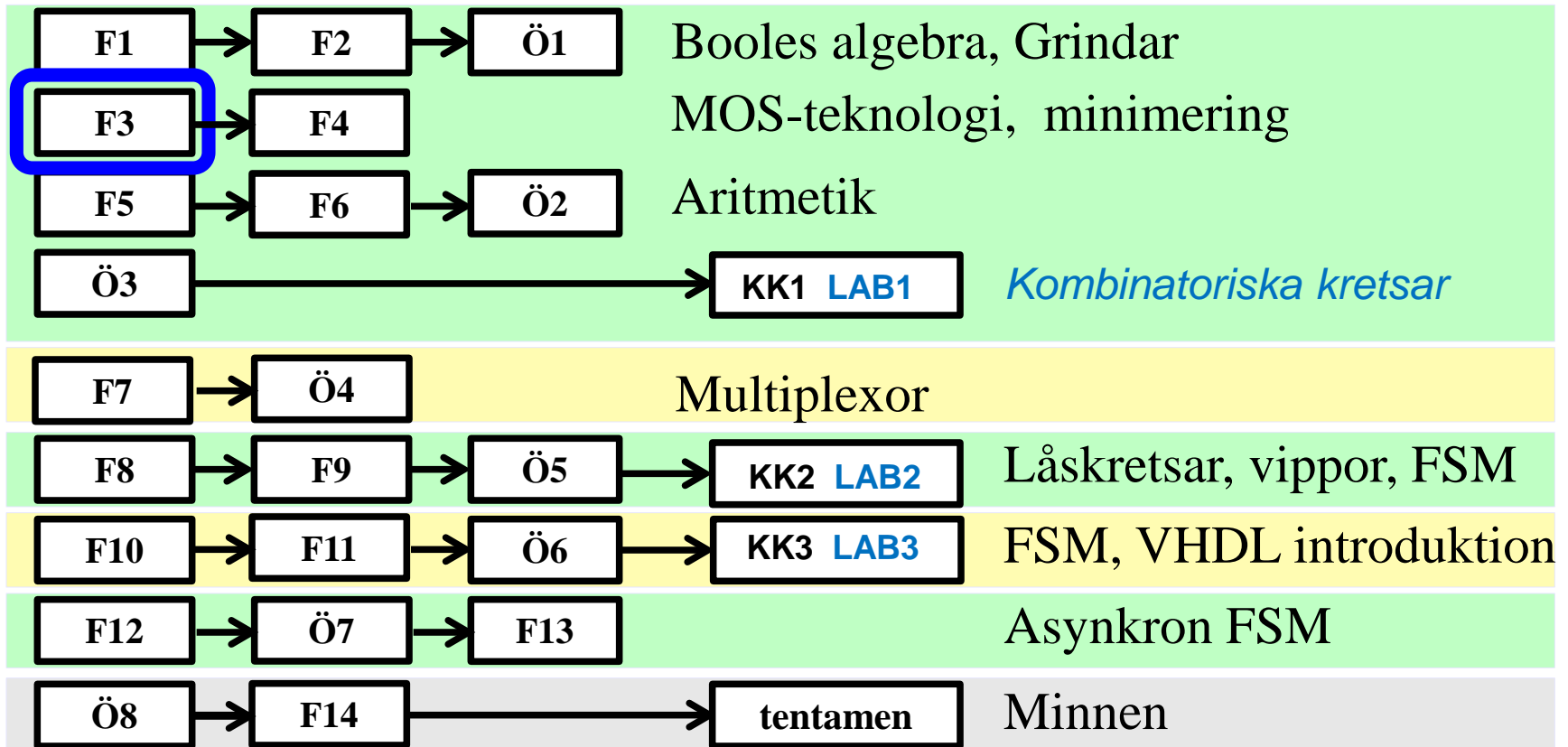


# Digital Design IE1204

**F3 CMOS-kretsen,  
Implementeringsteknologier**

**`william@kth.se`**

# IE1204 Digital Design



*Föreläsningar och övningar bygger på varandra! Ta alltid igen det Du missat!  
Läs på i förväg – delta i undervisningen – arbeta igenom **igen** efteråt!*

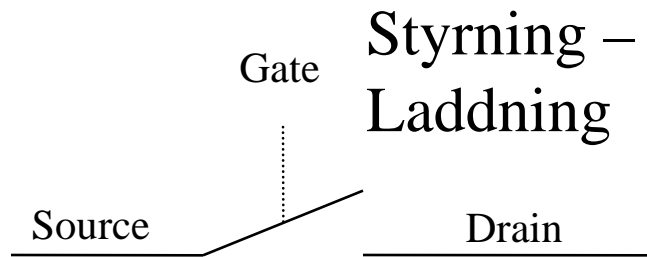
# Detta har hänt i kursen ...

Talsystem: Decimala, hexadecimala, oktala, binära

$$(175,5)_{10} = (AE.8)_{16} = (256.4)_8 = (10101110.1)_2$$

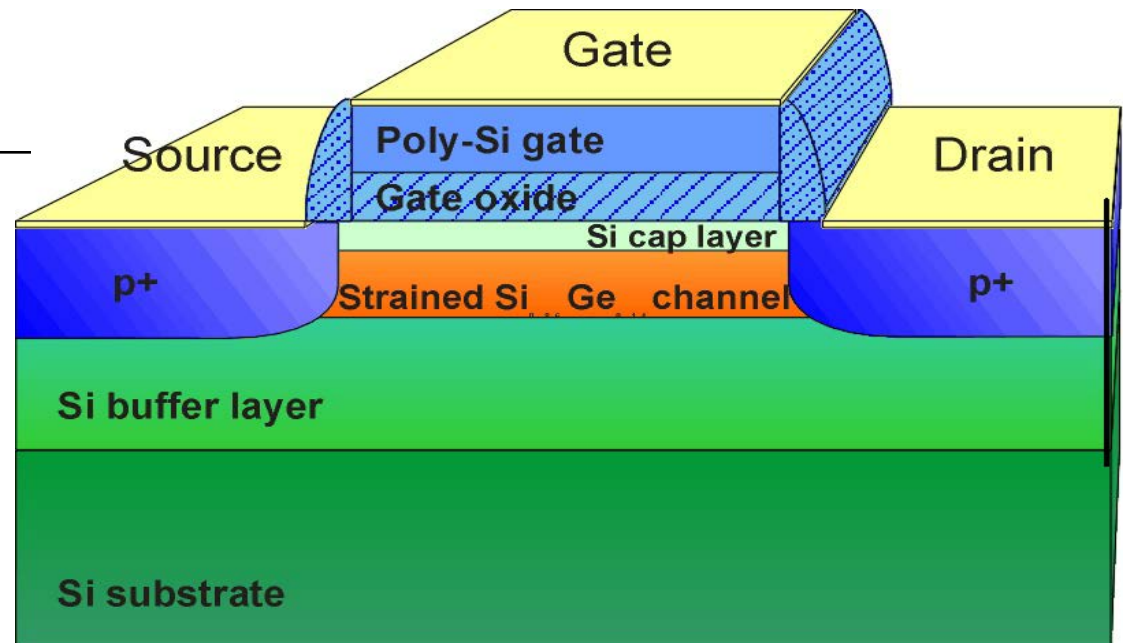
AND OR NOT EXOR EXNOR Sanningstabell,  
mintermer Maxtermer PS-form SP-form  
deMorgans lag Bubbelgrindar  
Fullständig logik NAND NOR

# Transistorn en omkopplare utan rörliga delar



Omkopplare – Kan leda ström  
(ström=laddning/tid)

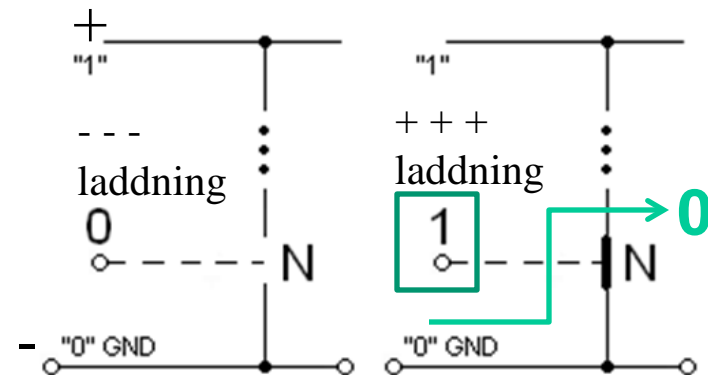
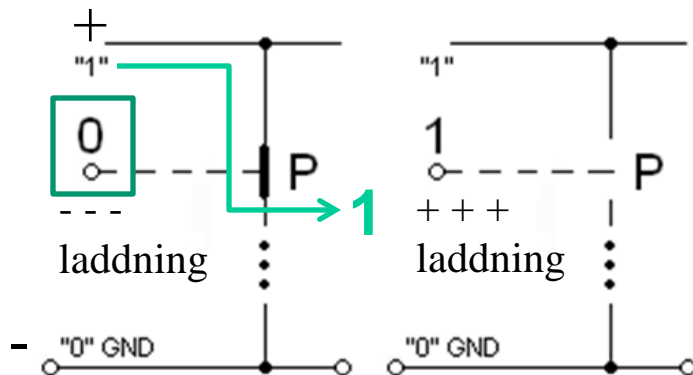
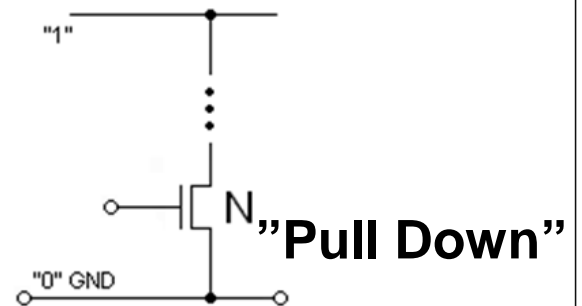
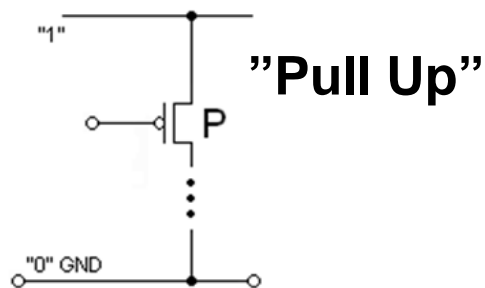
Principskiss för SiGe transistor (KTH)



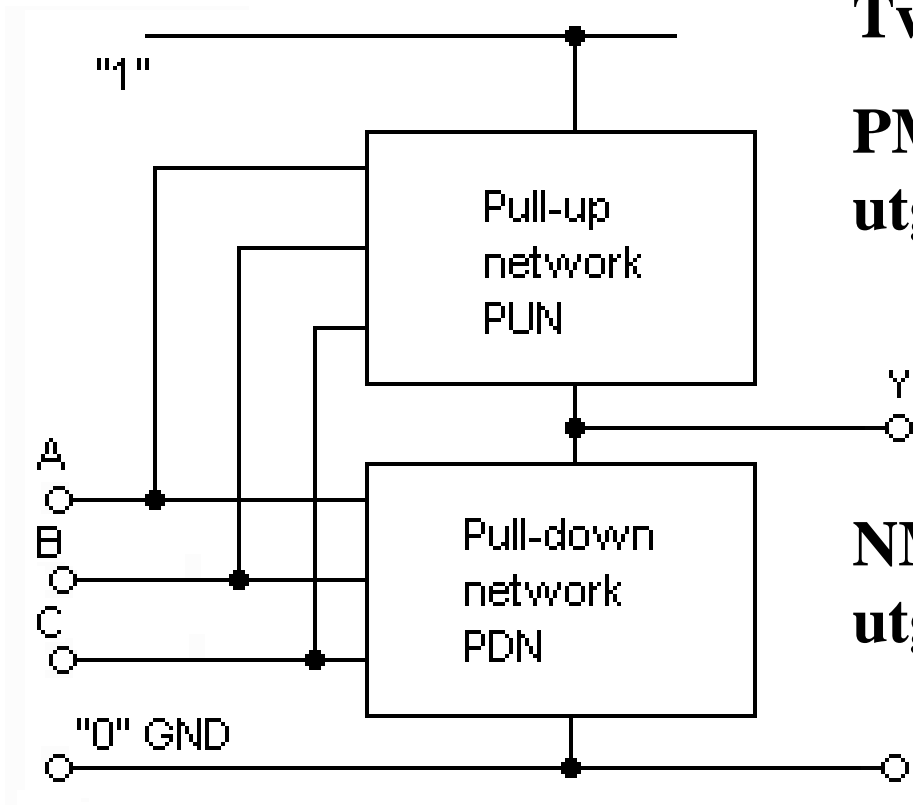
# Varför CMOS?

- CMOS-Transistorer är enkla att tillverka
- CMOS-Transistorer är gjorda av vanlig sand => billigt råmaterial
- En transistor är lätt att få att fungera som en switch (omkopplare)

# P och N MOS-transistorer



# Strukturen av en CMOS-krets

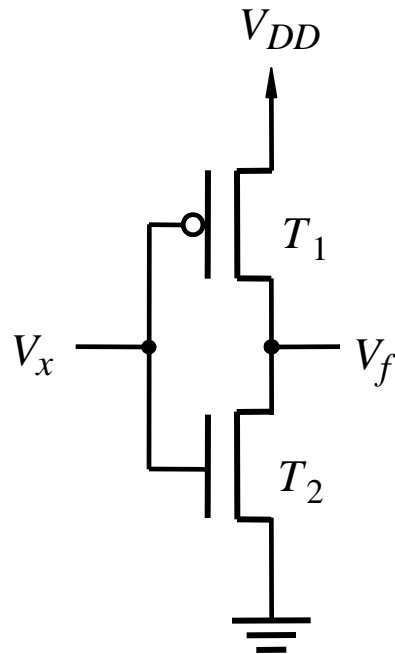


**Två olika nät:**

**PMOS gör kretsens  
utgång "1"**

**NMOS gör kretsens  
utgång "0"**

# Inverteraren



(a) Circuit

**En CMOS-krets består av både PMOS och NMOS-kretsar. CMOS står för (Complementary MOS).**

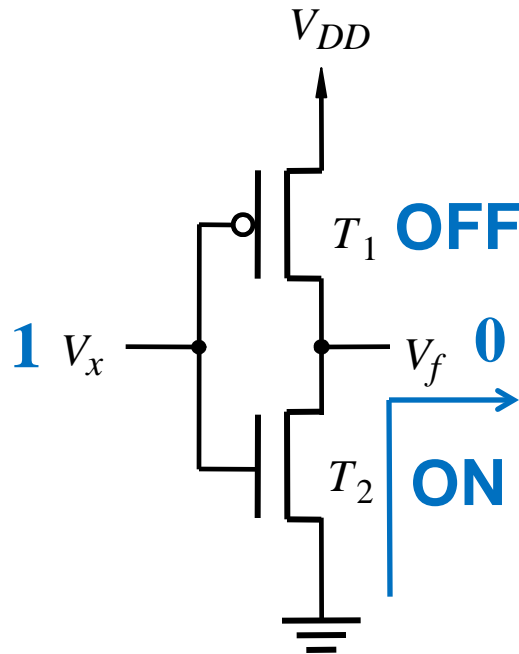
$x$	$T_1$	$T_2$	$f$
0	on	off	1
1	off	on	0

(b) Truth table and transistor states

Area:  $A_{\text{inverter}} = 2$  Transistors



# Inverteraren



(a) Circuit

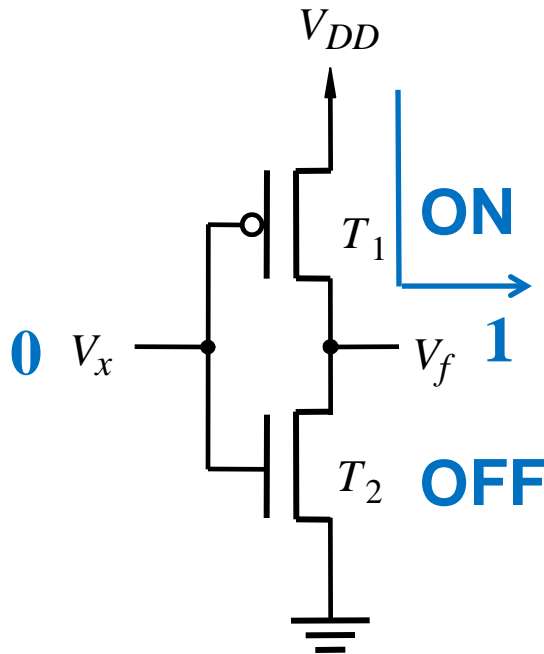
En CMOS-krets består av både PMOS och NMOS-kretsar. CMOS står för (Complementary MOS).

$x$	$T_1$	$T_2$	$f$
0	on	off	1
1	off	on	0

(b) Truth table and transistor states

Area:  $A_{\text{inverter}} = 2$  Transistors

# Inverteraren



(a) Circuit

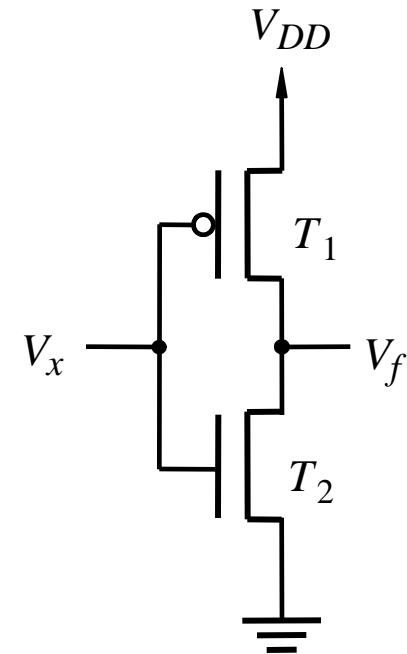
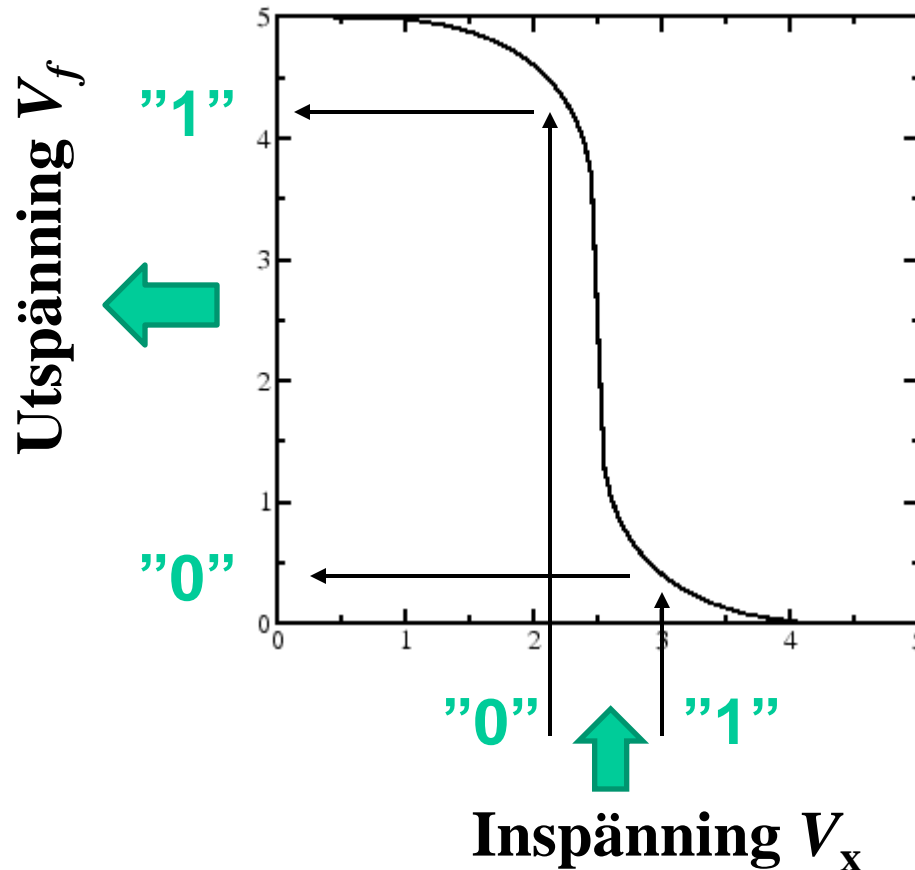
En CMOS-krets består av både PMOS och NMOS-kretsar. CMOS står för (Complementary MOS).

$x$	$T_1$	$T_2$	$f$
0	on	off	1
1	off	on	0

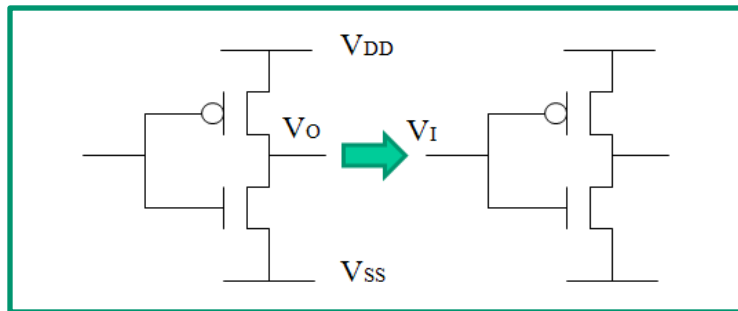
(b) Truth table and transistor states

Area:  $A_{\text{inverter}} = 2$  Transistors

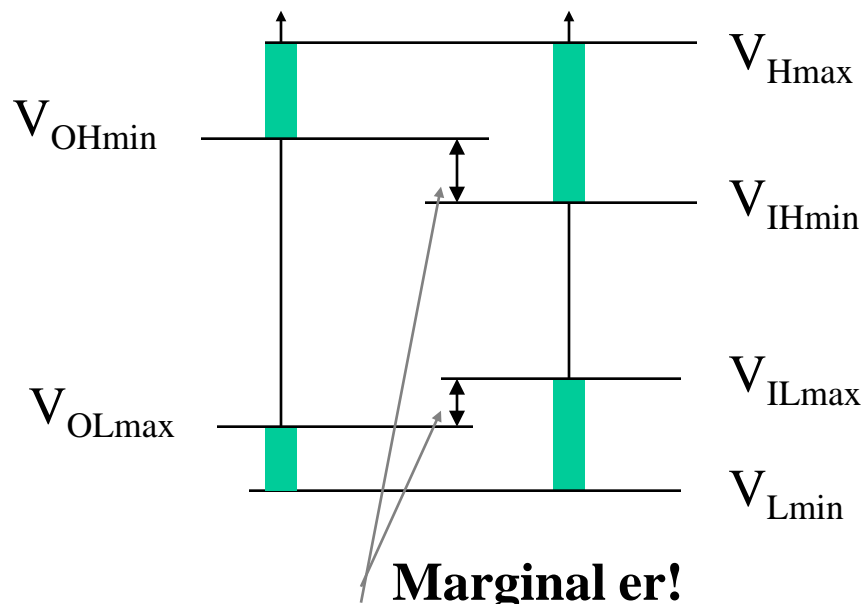
# CMOS-inverterarens spänningsnivåer



# Typiska signalnivåer för CMOS

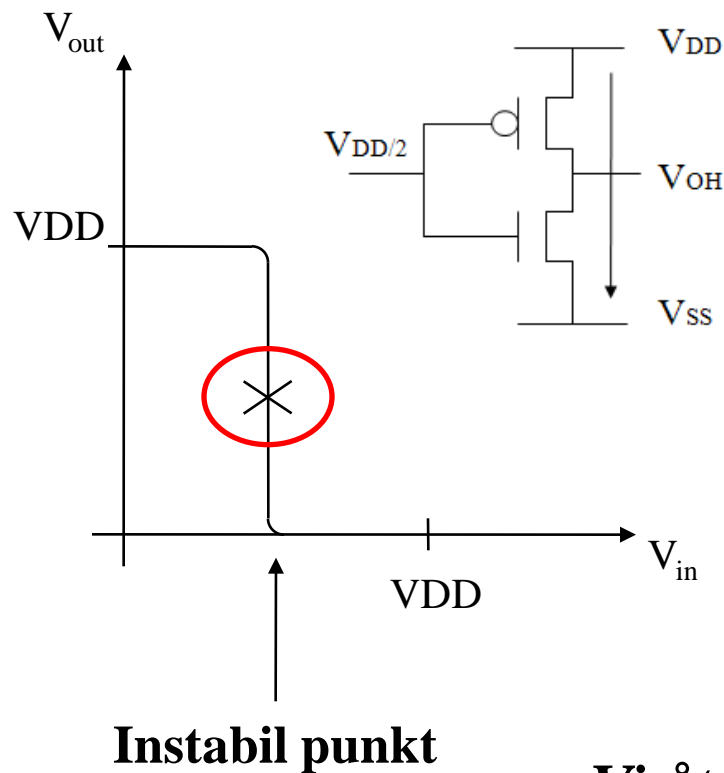


*Utgångsspänningar  $V_O$  och ingångsspänningar  $V_I$  passar varandra som "hand i handske", och med marginal!*



Matningsspänning	5.0V	3.3V	1.8V
$V_{HMAX}$	5.0	3.3	1.8
$V_{IHMIN}$	2.9	1.9	1.0
$V_{LMAX}$	2.1	1.4	0.8
$V_{LMIN}$	0.0	0.0	0.0

# En instabil punkt !

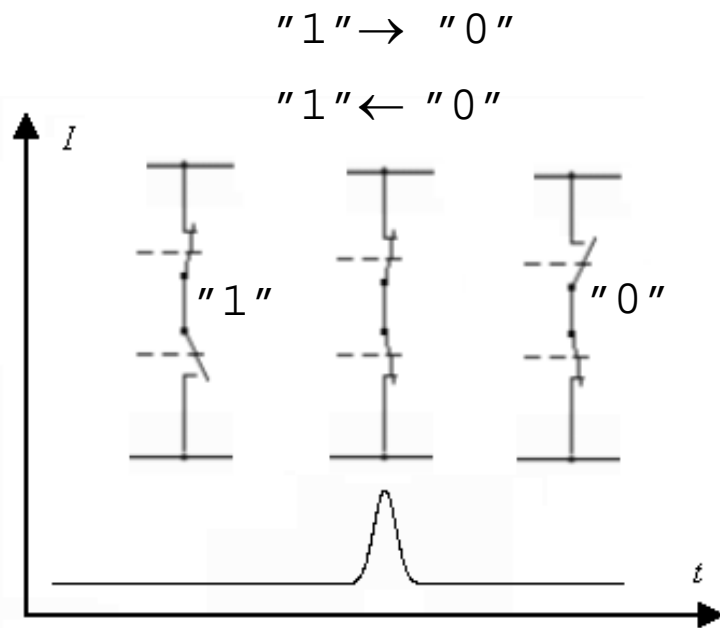


- CMOS-kretsen har en mycket stabil överföringsfunktion
- Vid  $V_{in} = V_{DD}/2$  finns en **instabil punkt**, då både  $T_1$  och  $T_2$  leder.
- Om en krets tillfälligt fastnar i detta läge så inträder ett tillstånd som kallas för metastabilitet.
- Om detta tillstånd varar för länge så kan transistorerna i kretsen skadas pga den höga strömmen.

*Vi återkommer till metastabilitet ...*

# CMOS - Dynamisk förlusteffekt !

Klassisk CMOS har *bara* förlusteffekt precis vid *omslaget*.  
Förlusteffekten  $P_F$  blir proportionell mot klockfrekvensen!



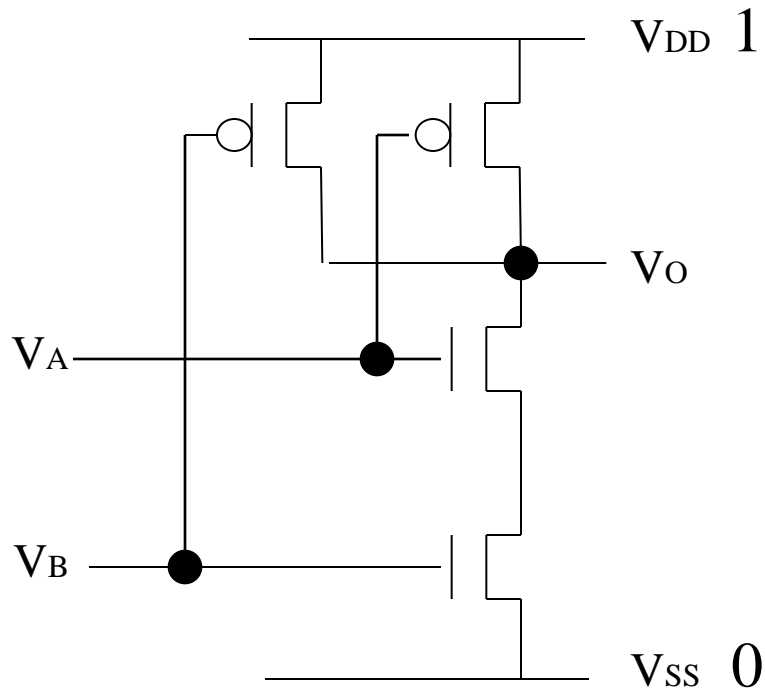
$$P_F \propto f_C \cdot V_{DD}^2$$

$P_F$  Power losses

$f_C$  Clockfrequency

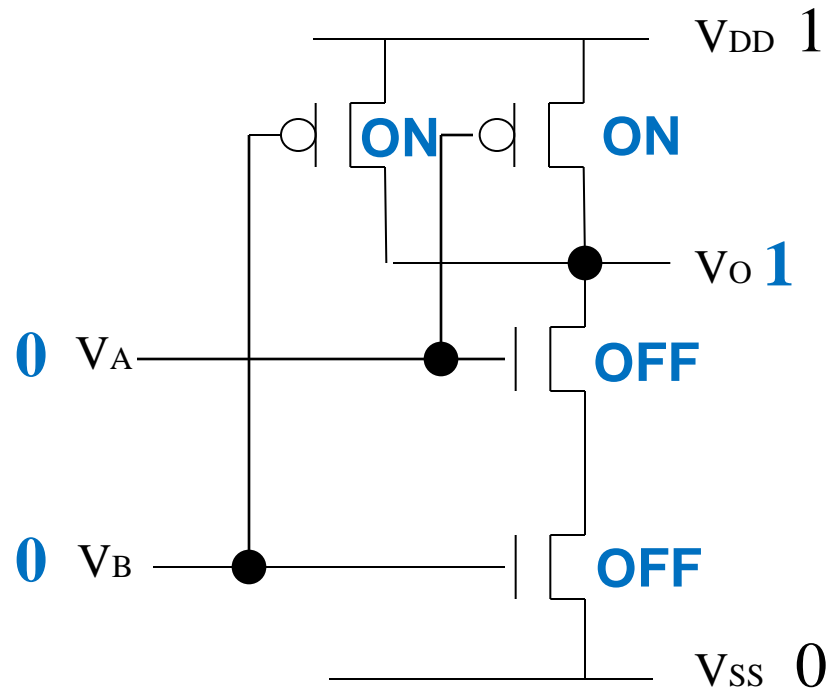
$V_{DD}$  Supply Voltage

# Snabbfråga: Vilken grind?



$V_A$	$V_B$	$V_O$
$V_{SS}(0)$	$V_{SS}(0)$	
$V_{SS}(0)$	$V_{DD}(1)$	
$V_{DD}(1)$	$V_{SS}(0)$	
$V_{DD}(1)$	$V_{DD}(1)$	

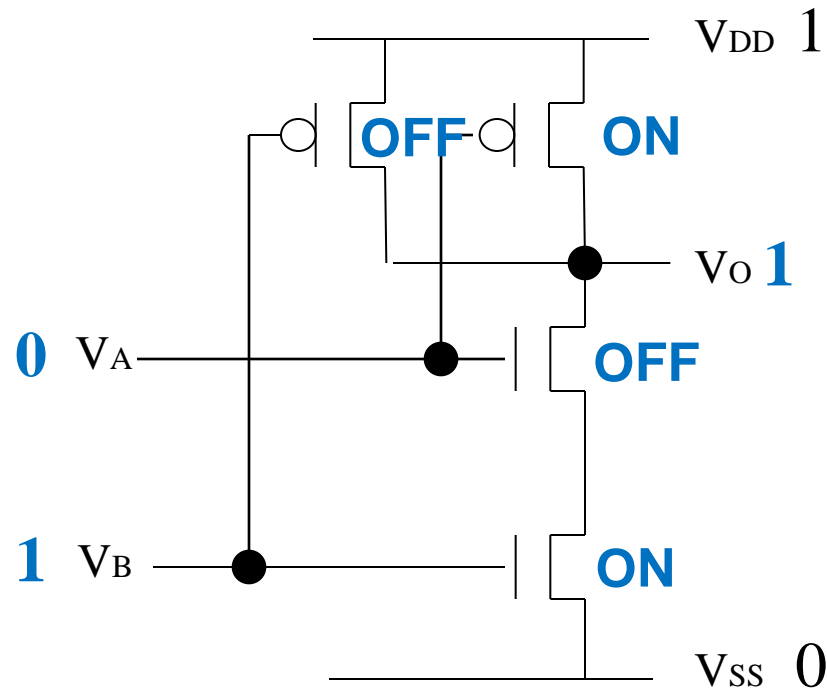
# Snabbfråga: Vilken grind?



$V_A$	$V_B$	$V_O$
$V_{SS}(0)$	$V_{SS}(0)$	$V_{DD}(1)$
$V_{SS}(0)$	$V_{DD}(1)$	
$V_{DD}(1)$	$V_{SS}(0)$	
$V_{DD}(1)$	$V_{DD}(1)$	

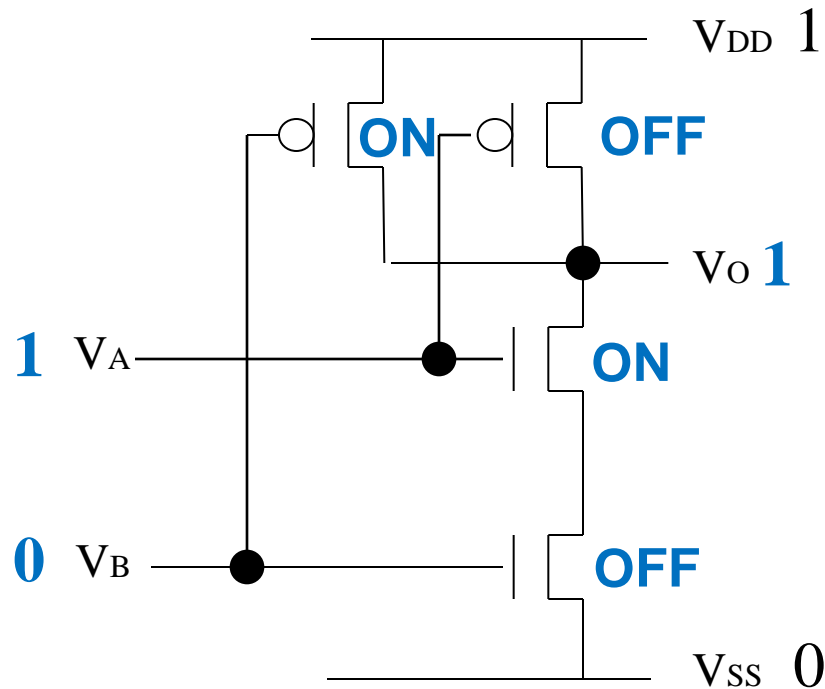


# Snabbfråga: Vilken grind?



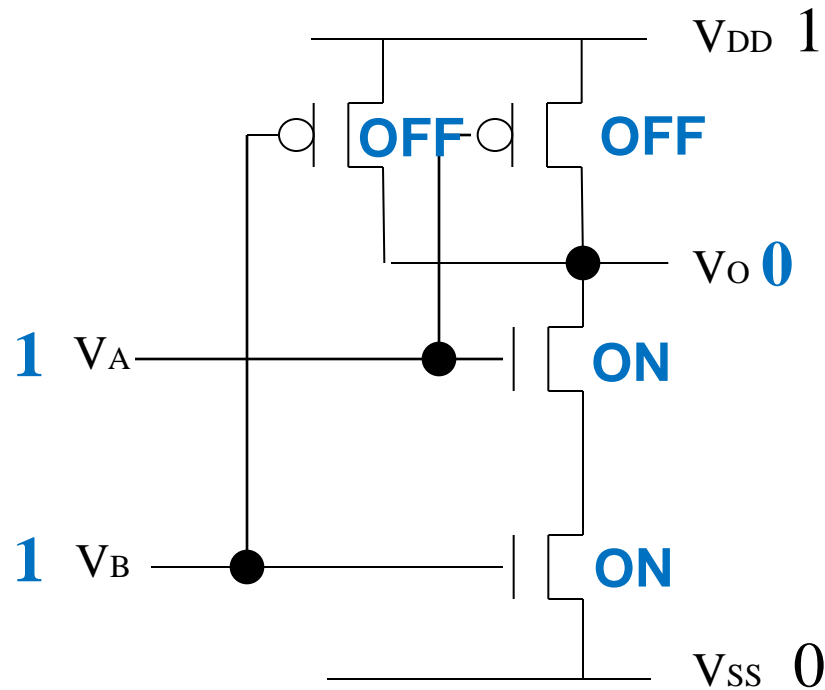
$V_A$	$V_B$	$V_O$
$V_{SS}(0)$	$V_{SS}(0)$	$V_{DD}(1)$
$V_{SS}(0)$	$V_{DD}(1)$	$V_{DD}(1)$
$V_{DD}(1)$	$V_{SS}(0)$	
$V_{DD}(1)$	$V_{DD}(1)$	

# Snabbfråga: Vilken grind?



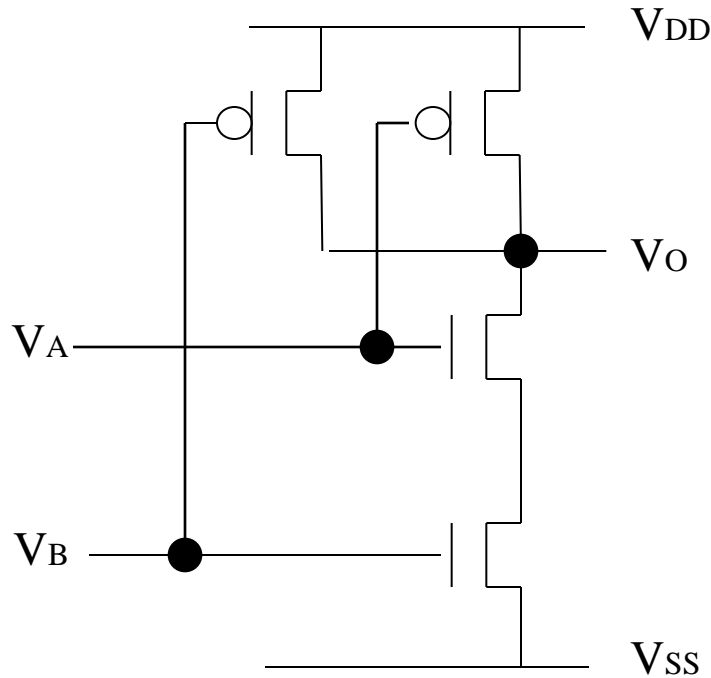
$V_A$	$V_B$	$V_O$
$V_{SS}(0)$	$V_{SS}(0)$	$V_{DD}(1)$
$V_{SS}(0)$	$V_{DD}(1)$	$V_{DD}(1)$
$V_{DD}(1)$	$V_{SS}(0)$	$V_{DD}(1)$
$V_{DD}(1)$	$V_{DD}(1)$	

# Snabbfråga: Vilken grind?



$V_A$	$V_B$	$V_O$
$V_{SS}(0)$	$V_{SS}(0)$	$V_{DD}(1)$
$V_{SS}(0)$	$V_{DD}(1)$	$V_{DD}(1)$
$V_{DD}(1)$	$V_{SS}(0)$	$V_{DD}(1)$
$V_{DD}(1)$	$V_{DD}(1)$	$V_{SS}(0)$

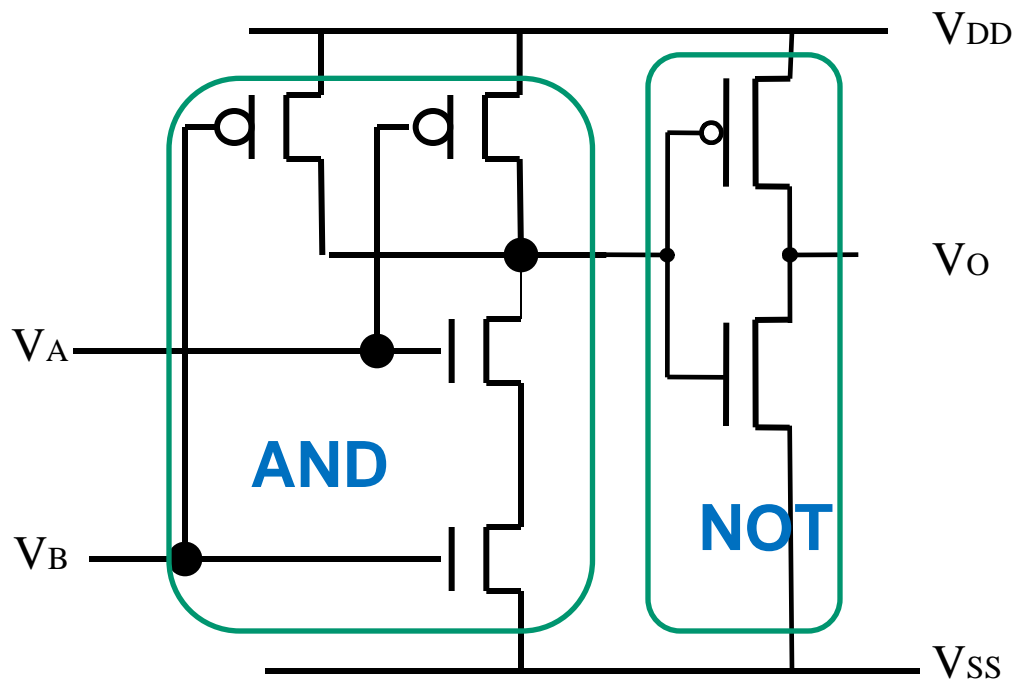
# NAND-grinden



$V_A$	$V_B$	$V_O$
$V_{SS}(0)$	$V_{SS}(0)$	$V_{DD}(1)$
$V_{SS}(0)$	$V_{DD}(1)$	$V_{DD}(1)$
$V_{DD}(1)$	$V_{SS}(0)$	$V_{DD}(1)$
$V_{DD}(1)$	$V_{DD}(1)$	$V_{SS}(0)$

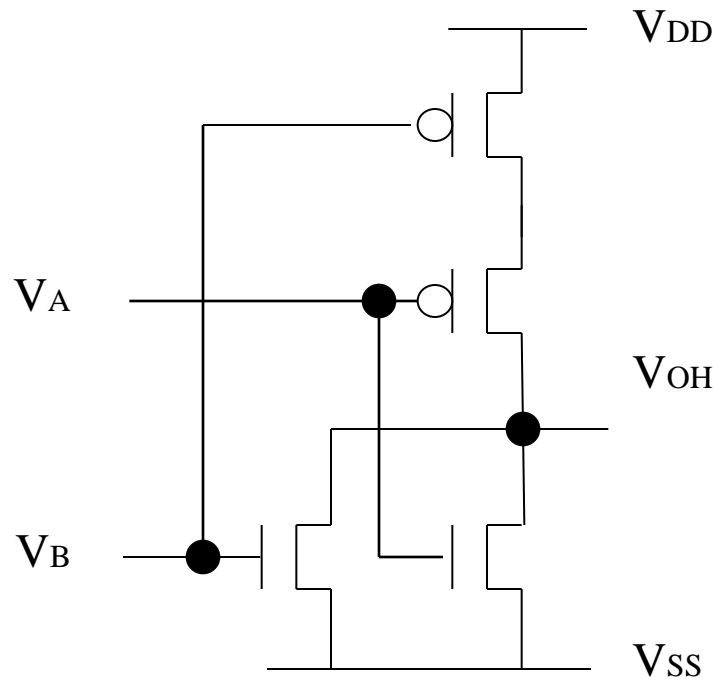
Area:  $A_{\text{NAND}} = 4$  Transistors

# AND-grinden!



Area:  $A_{AND} = 6$  Transistors!

# NOR-grinden



$V_A$	$V_B$	$V_{OH}$
$V_{SS}(0)$	$V_{SS}(0)$	$V_{DD}(1)$
$V_{SS}(0)$	$V_{DD}(1)$	$V_{SS}(0)$
$V_{DD}(1)$	$V_{SS}(0)$	$V_{SS}(0)$
$V_{DD}(1)$	$V_{DD}(1)$	$V_{SS}(0)$

Area:  $A_{NOR} = 4$  Transistors

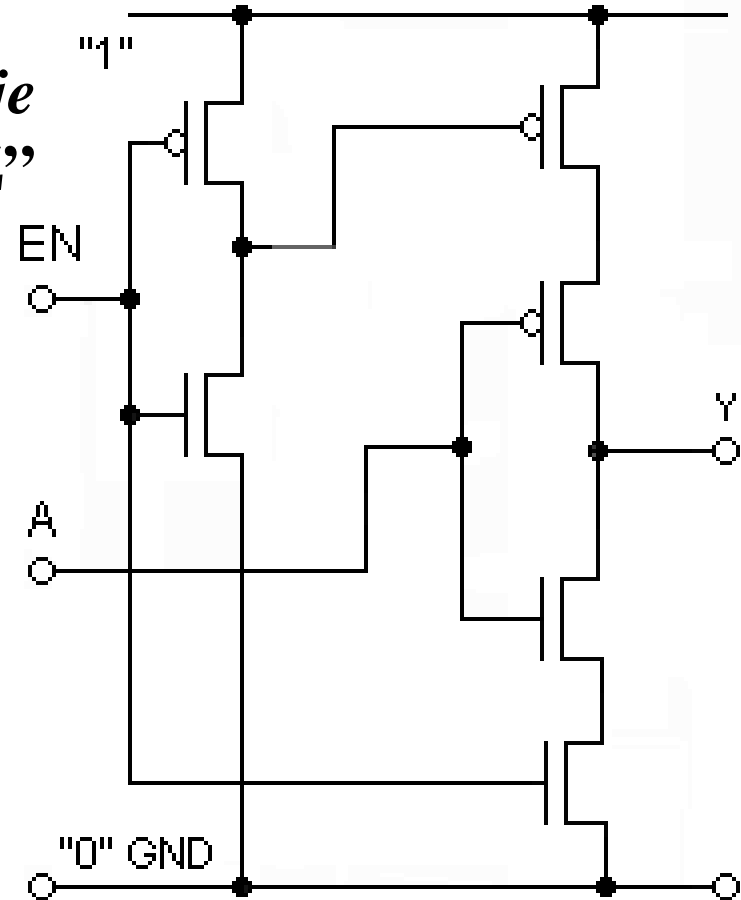
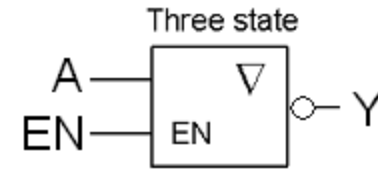
# Negativ logik ?

- Man kan också vända på begreppen och låta **L** (låg spänning) representera en logisk 1:a och låta **H** (hög spänning) representera en logisk 0:a.
  - Detta kallas för negativ logik.
- En AND-funktion blir då en OR-funktion och vice versa.
  - Negativ logik eller positiv logik är egentligen egalt, men av tradition använder man sig av positiv logik.

# Three-state ?

En CMOS-grind kan förutom "1" eller "0" även förses med ett *tredje* utgångstillstånd – **Three-state "Z"** (= fränkopplad utgång ).

Om många utgångar kopplas ihop till **samma tråd** ("buss") så kan ju bara *en* av utgångarna åt gången få vara aktiv. De övriga hålls i Threestate-tillståndet.

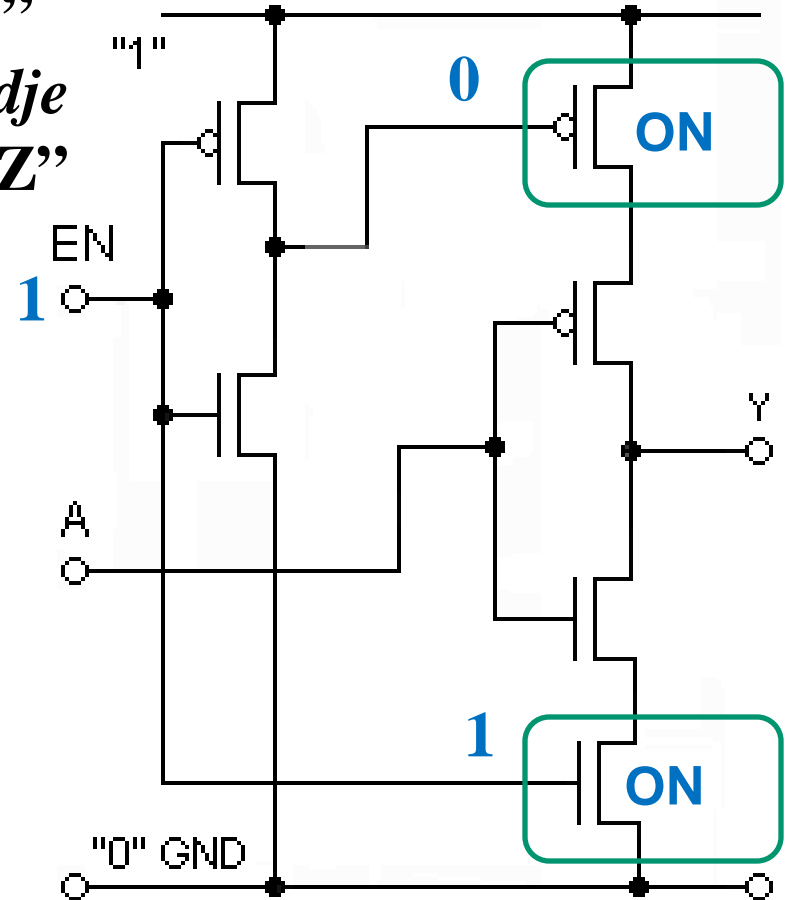
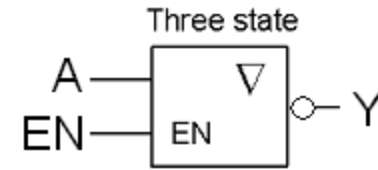




# Three-state ?

En CMOS-grind kan förutom "1" eller "0" även förses med ett *tredje* utgångstillstånd – **Three-state "Z"** (= fränkopplad utgång ).

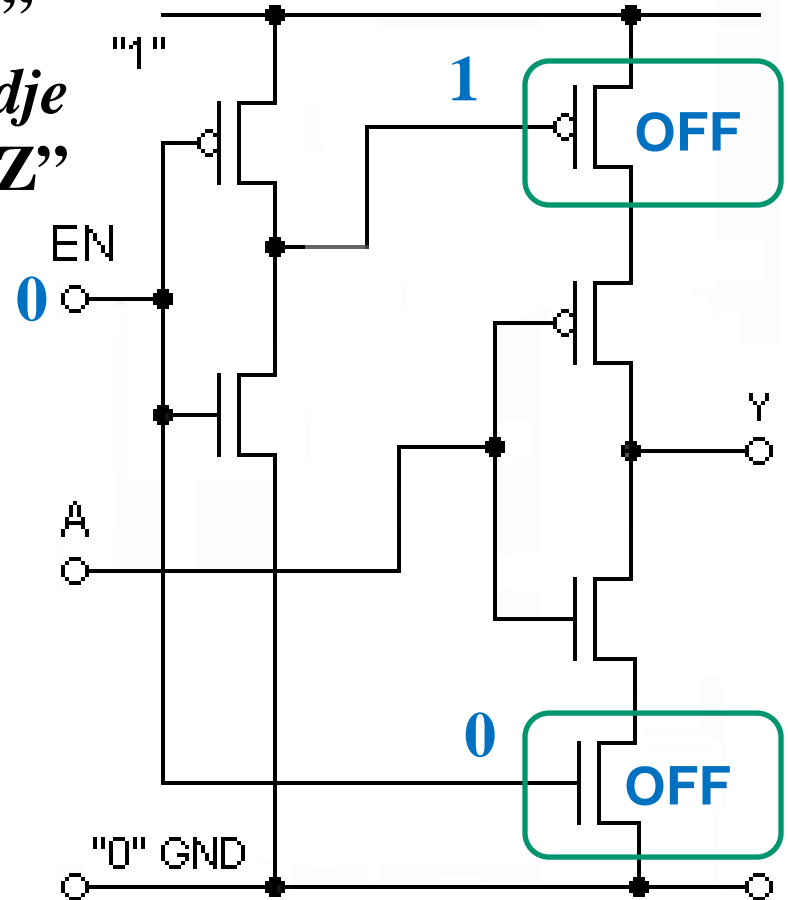
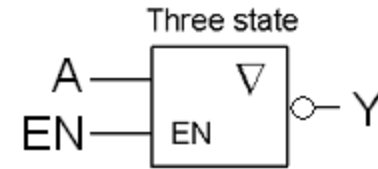
Om många utgångar kopplas ihop till **samma tråd** ("buss") så kan ju bara *en* av utgångarna åt gången få vara aktiv. De övriga hålls i Threestate-tillståndet.



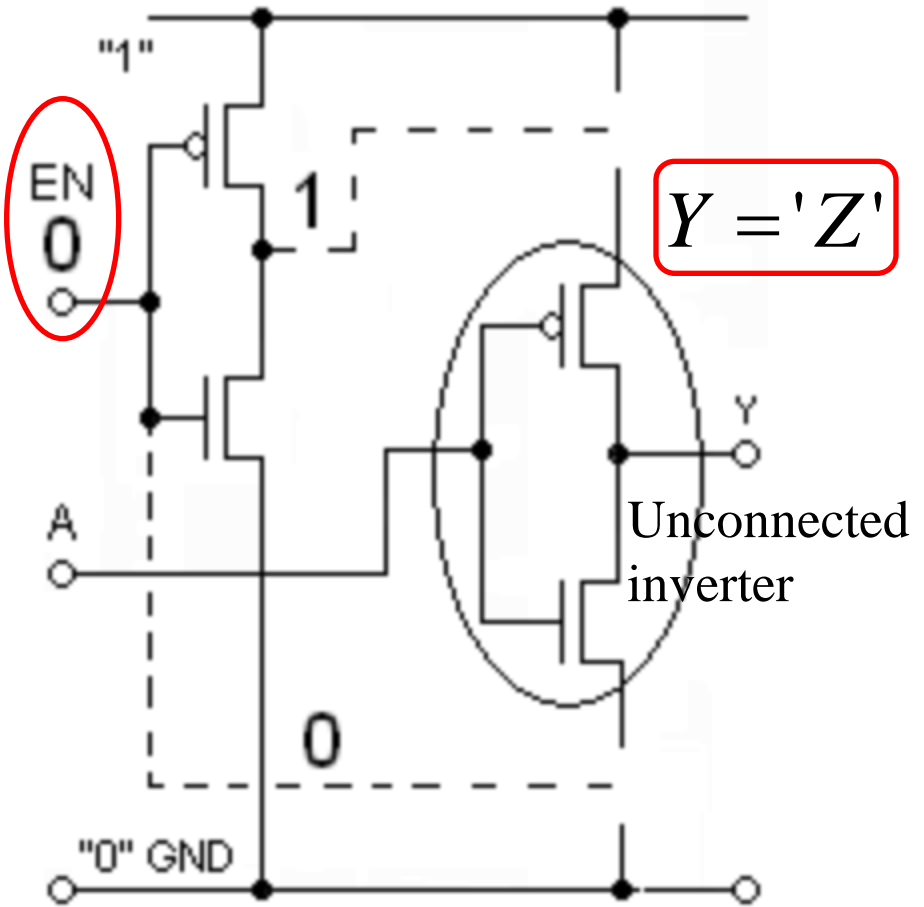
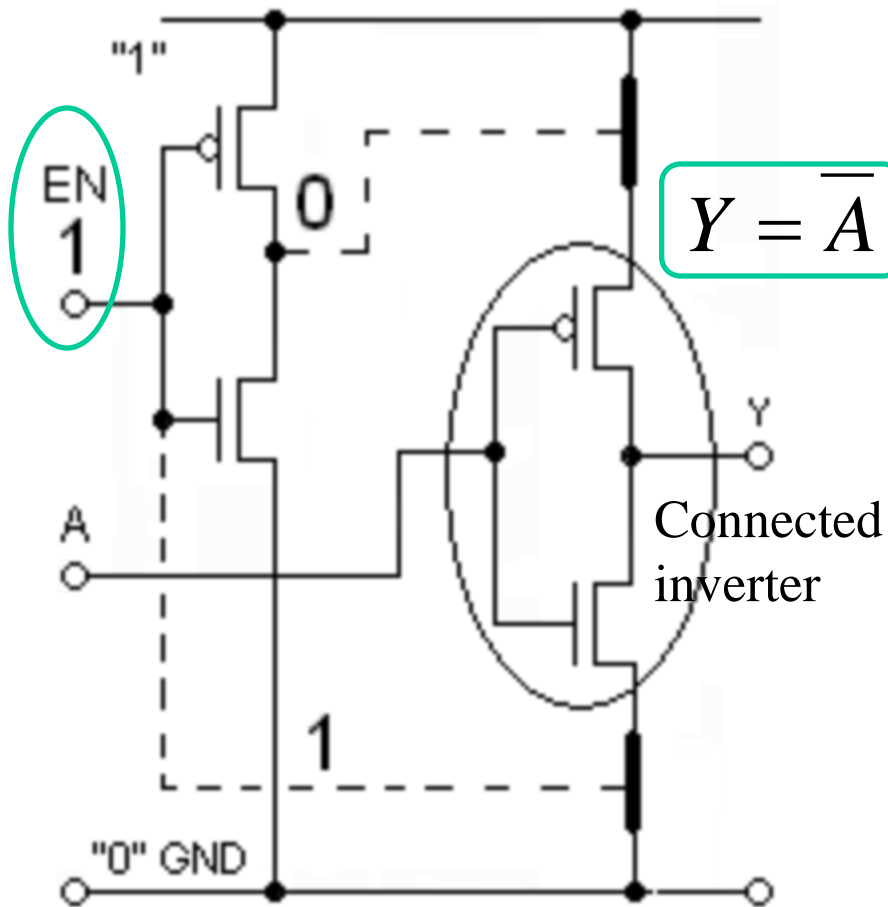
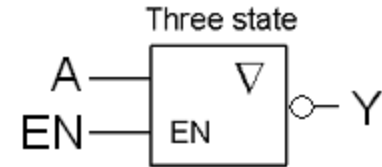
# Three-state ?

En CMOS-grind kan förutom "1" eller "0" även förses med ett *tredje* utgångstillstånd – **Three-state "Z"** (= fränkopplad utgång ).

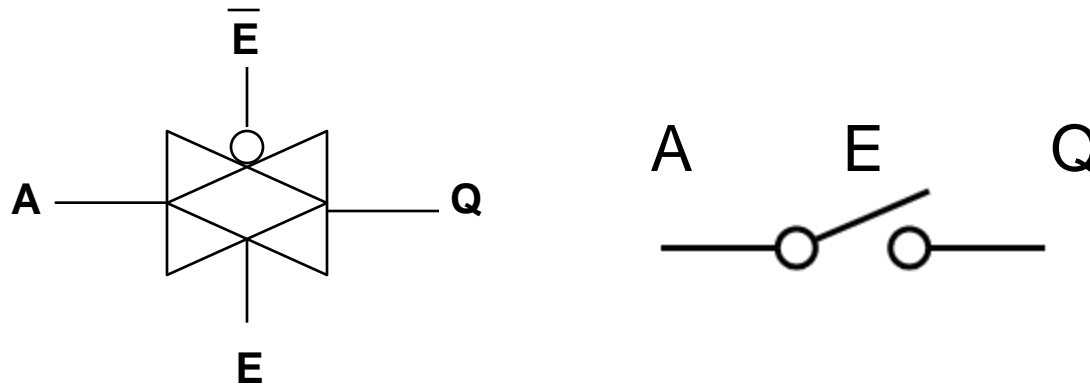
Om många utgångar kopplas ihop till **samma tråd** ("buss") så kan ju bara *en* av utgångarna åt gången få vara aktiv. De övriga hålls i Threestate-tillståndet.



# Three state 'Z'



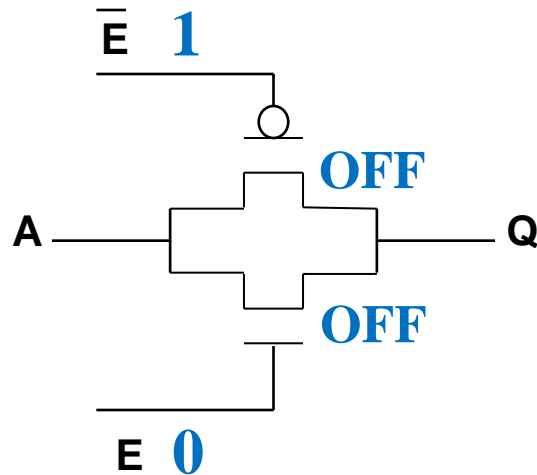
# Transmissionsgrinden (Pass gate)



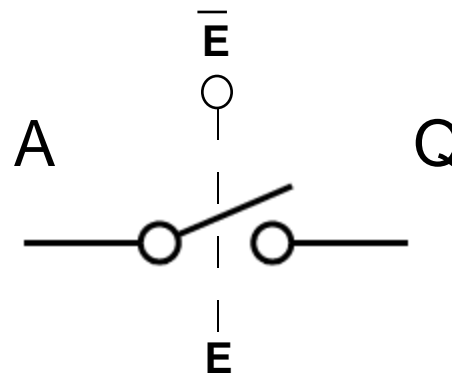
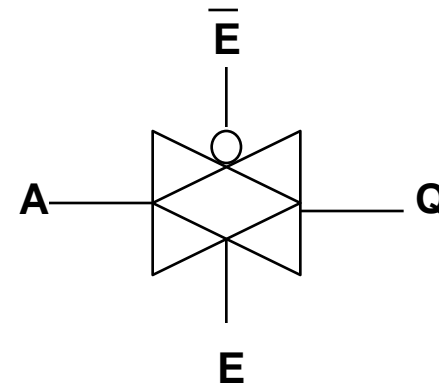
Utan att gå in på kretsdetaljerna så består en transmissionsgrind av en PMOS-transistor i **parallell** med en NMOS-transistor. Grinden styrs med E (och E') och är då att jämföra med en "vanlig" kontakt. En signal kan gå från A till Q, men även baklänges från Q till A. Transmissionsgrindskopplingar utnyttjar färre transistorer än andra grindar, men har sämre drivförmåga.

Area:  $A_{TG} = 2$  Transistors

# ( Transmissionsgrinden )

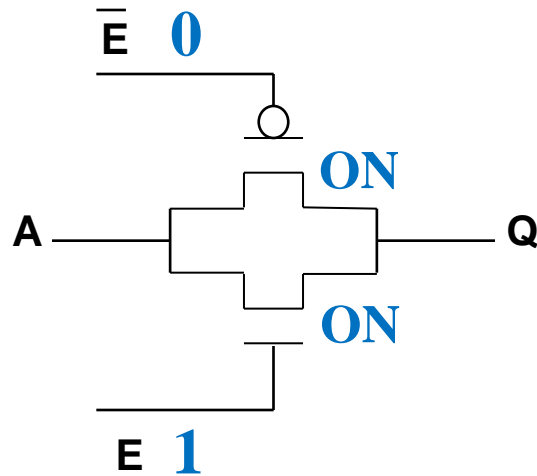


$V_A$	$V_E$	$V_{OH}$
L	L	Z
L	H	L
H	L	Z
H	H	H

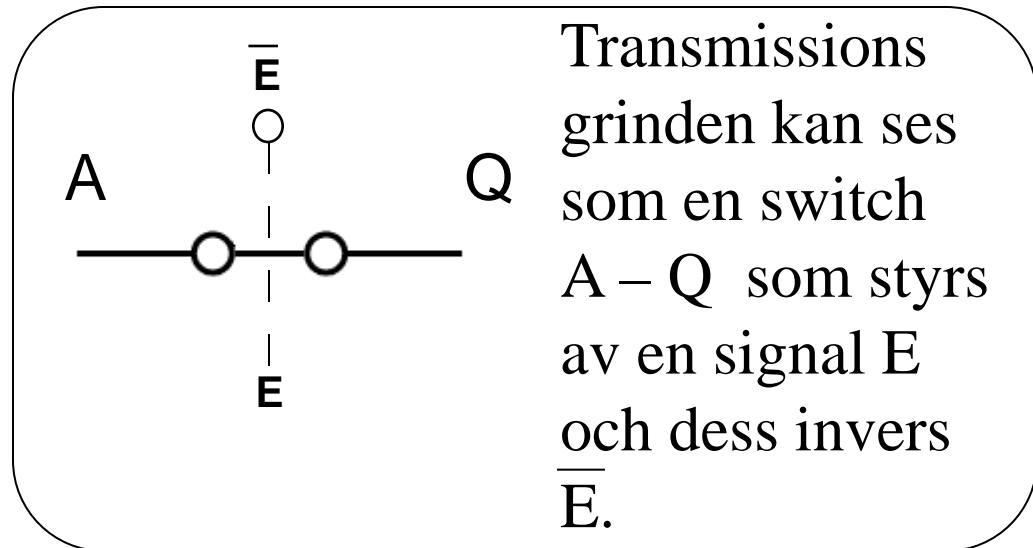
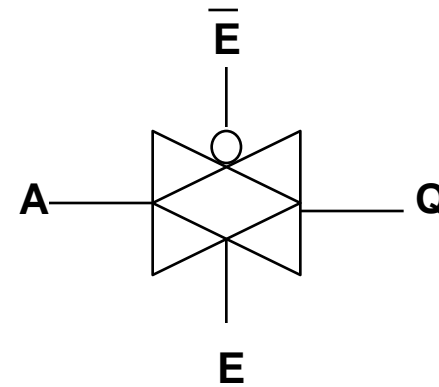


Transmissionsgrinden kan ses som en switch A – Q som styrs av en signal E och dess invers  $\overline{E}$ .

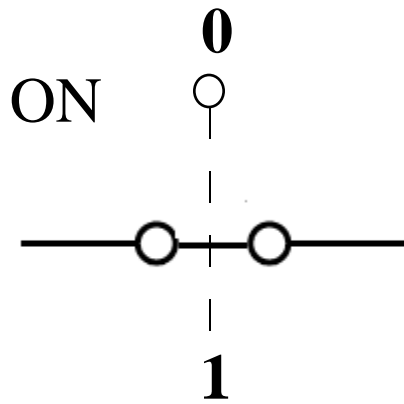
# ( Transmissionsgrinden )



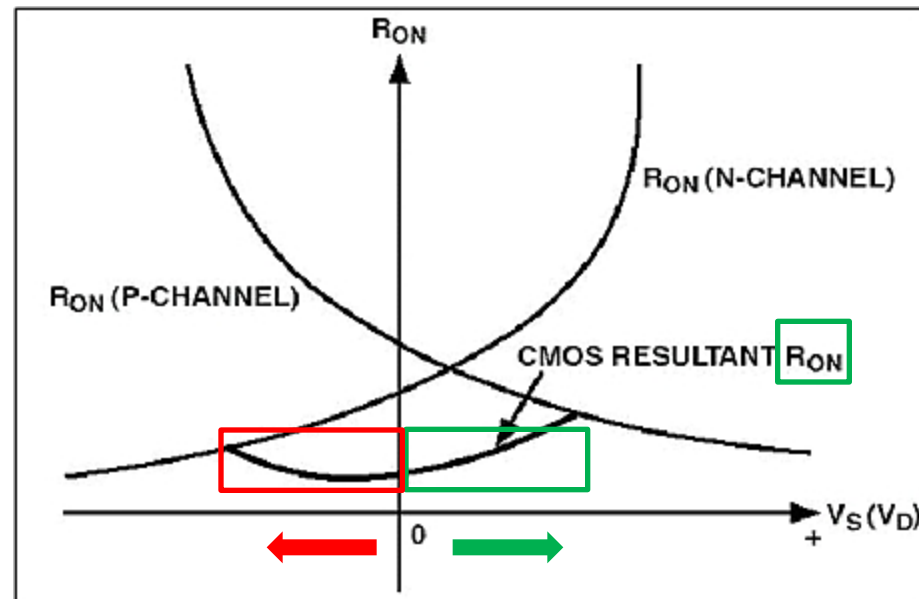
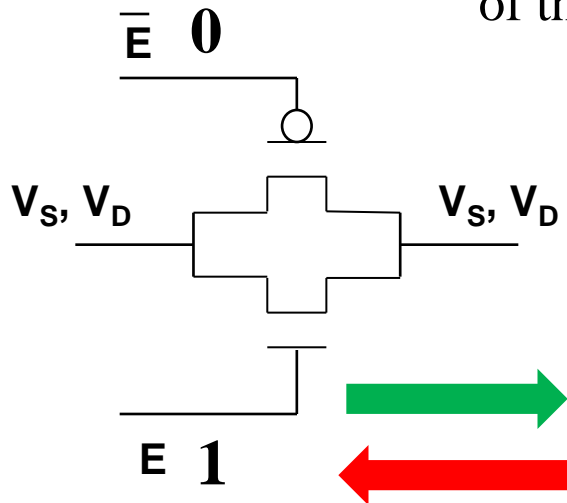
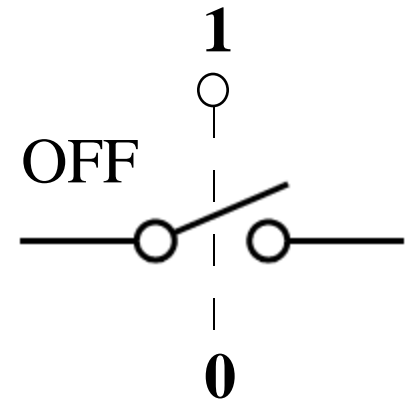
$V_A$	$V_E$	$V_{OH}$
L	L	Z
L	H	L
H	L	Z
H	H	H



# ( Inside story )



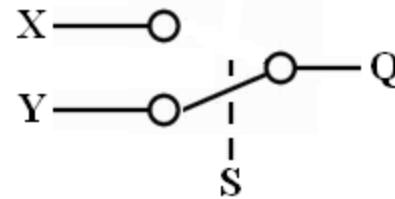
Transmission gate **ON**, both transistors are in parallel and contribute to the low On resistance of the switch.



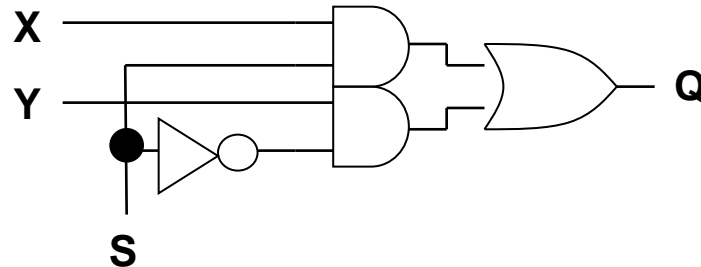
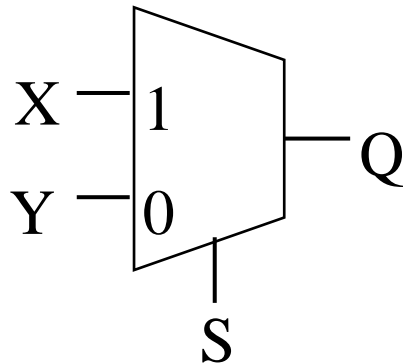
- The switch will work in both directions!

# Vad är en multiplexor, **MUX**?

**En multiplexor är en dataväljare.**



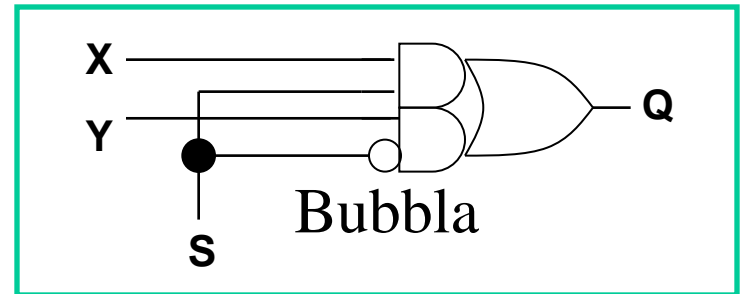
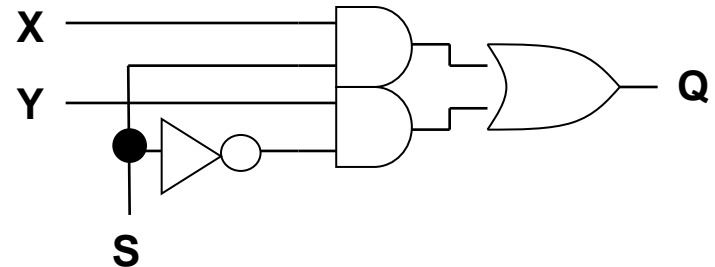
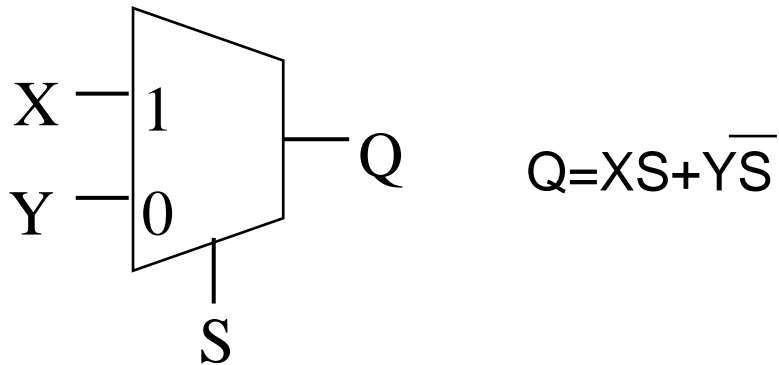
$$Q = XS + Y\bar{S}$$





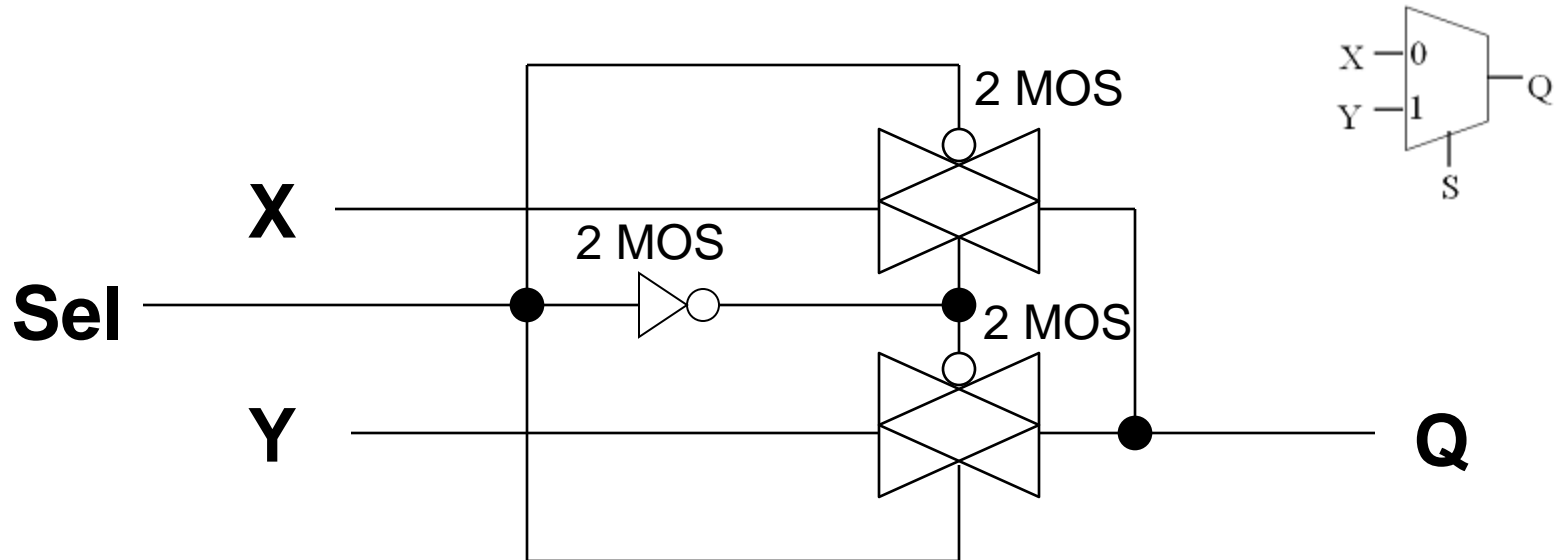
# Förenklat ritsätt

## Exempel: MUX



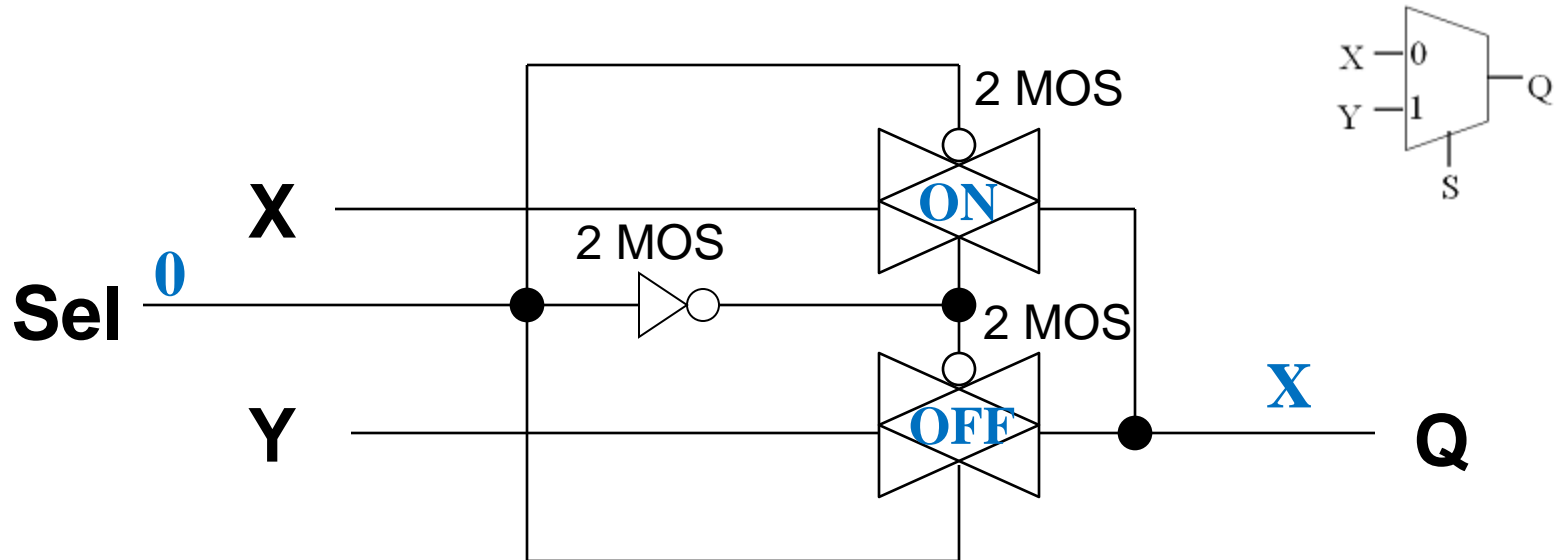
**Av inverteraren blir endast ringen kvar.  
Mellanliggande ledningar underförstås.**

# MUX med transmissionsgrind



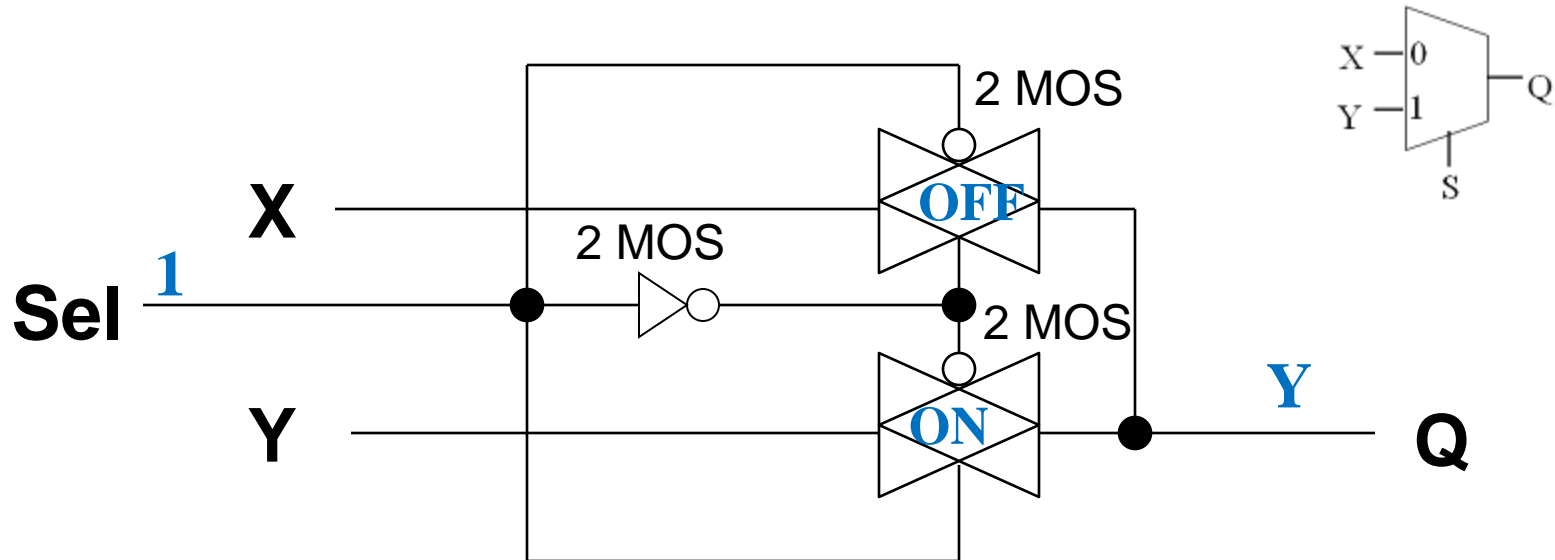
**Area:  $A_{\text{mux}} = 6$  Transistors**

# MUX med transmissionsgrind



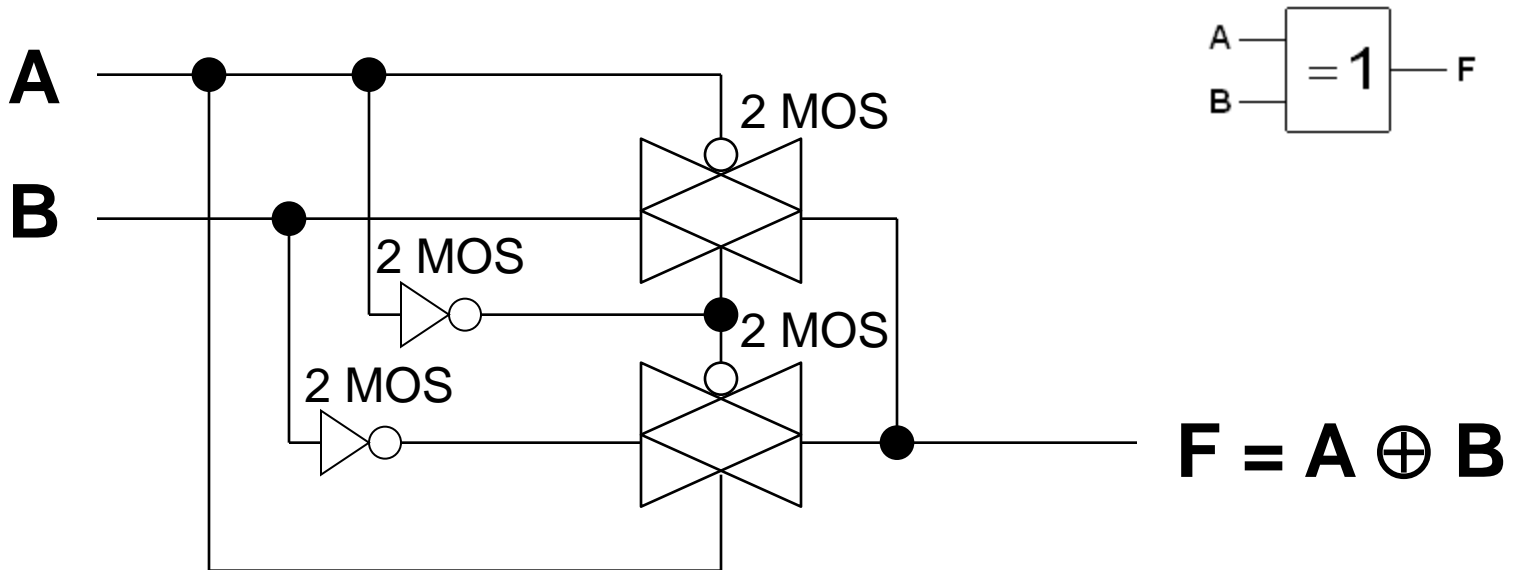
**Area:  $A_{\text{mux}} = 6$  Transistors**

# MUX med transmissionsgrind



**Area:  $A_{\text{mux}} = 6$  Transistors**

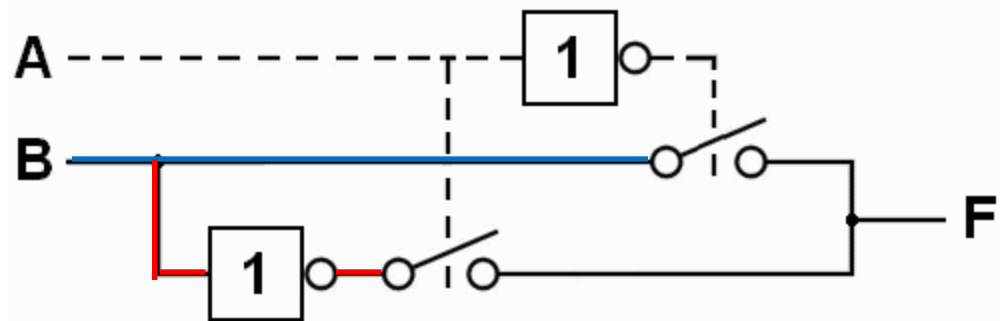
# XOR med transmissionsgrind



Area:  $A_{\text{XOR}} = 8$  Transistors

*Knappast  
självklart?*

# ( **XOR** med transmissionsgrind )



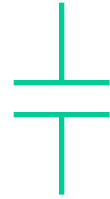
<i>A</i>	<i>B</i>	<i>F</i>
0	0	0
0	1	1
1	0	1
1	1	0

$F = B$

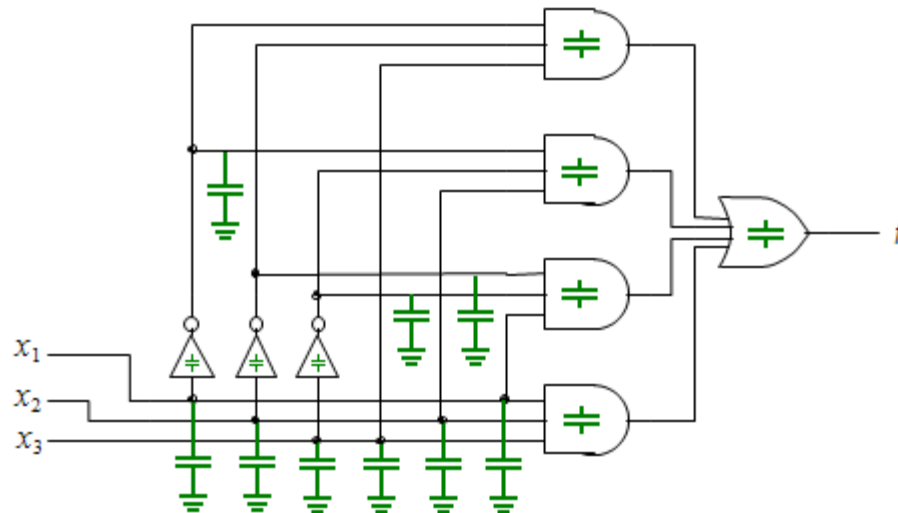
$F = \overline{B}$



# Fördröjningar i kretsar



**Alla ledningar i elektronikkretsar har kapacitans. Det tar ett tag för spänningar att nå slutvärdet. Dessa fördröjningar i kretsar och *mellan* kretsar begränsar snabbheten.**

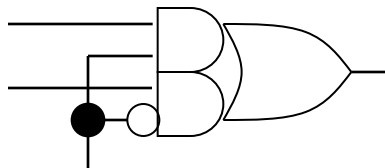




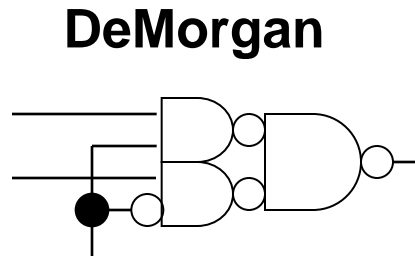
# Typiska fördröjningar

NAND,NOR, NOT	T	NAND=standard T
NOT	$\frac{1}{2} T$ , $1T$	(om NAND-grind)
NAND-NAND	2T	(2 NAND i rad)
AND-OR	4T, 3T	(NAND-NOT+NOR-NOT)
XOR,XNOR,MUX	3...5T	
XOR,MUX (med TG)	2T	

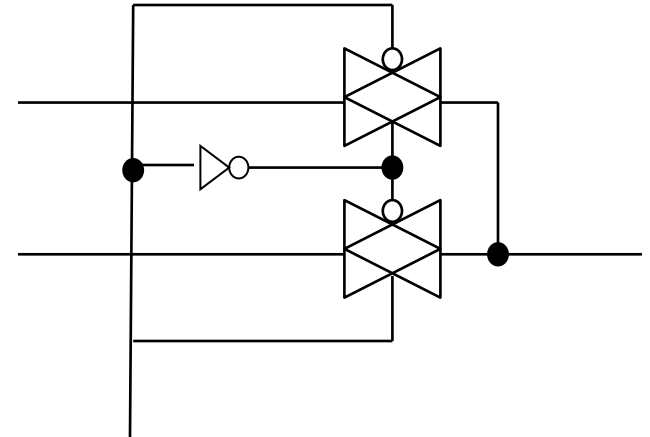
# Optimerade strukturer för MUX



AND-OR



NAND-NAND



Area:  $A_{\text{MUX}} = 2+6+6+6=20$

Transistorer

Delay:  $T_{\text{MUX}} = 5T_{\text{NAND}}$

Area:  $A_{\text{MUX}} = 6$  Transistorer

Delay:  $T_{\text{MUX}} = \sim 2T_{\text{NAND}}$

Area:  $A_{\text{MUX}} = 2+4+4 =$

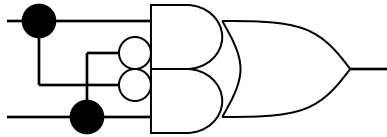
$= 10$  Transistorer

Delay:  $T_{\text{MUX}} = 3T_{\text{NAND}}$



**Bäst!**

# Optimerade strukturer för XOR

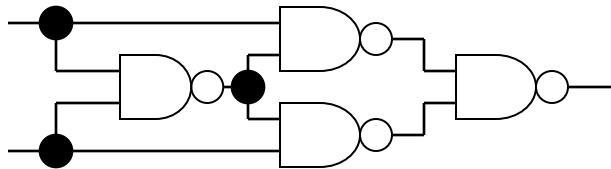


Area:  $A_{XOR}=2+2+6+6+6=22$

Transistorer

Delay:  $T_{XOR}=5T_{NAND}$

Nand only

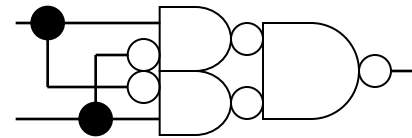


Area:  $A_{XOR}=4+4+4+4=16$

Transistorer

Delay:  $T_{XOR}=3T_{NAND}$

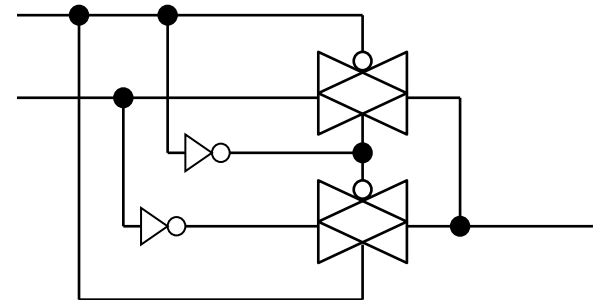
DeMorgan



Area:  $A_{XOR}=2+2+4+4=12$

Transistorer

Delay:  $T_{XOR}=3T_{NAND}$



Area:  $A_{XOR}=8$  Transistorer

Delay:  $T_{XOR} \sim 2T_{NAND}$

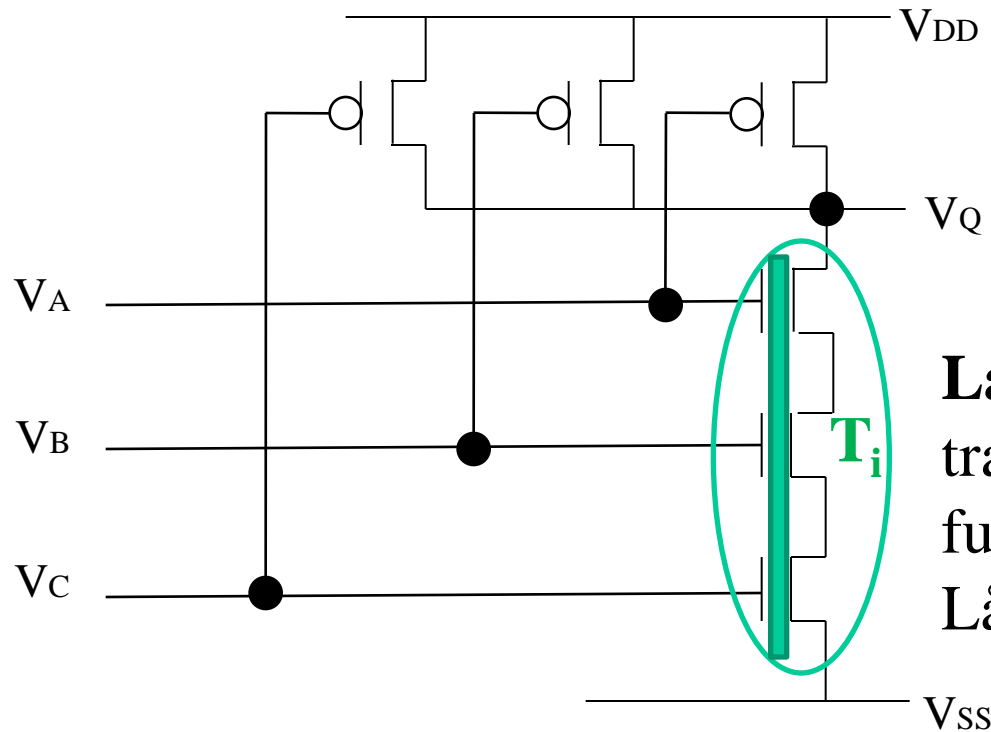
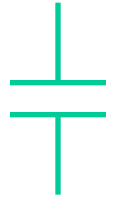


Bäst!

# Fan-out och Fan-in

- **Fan-out** - en utgång driver många ingångar. Utgången lastas ned med summan av ingångarnas kapacitanserna  $\Rightarrow$  fördröjningen **T** blir **last-beroende**.
- **Fan-in** - en grind har många ingångar. Detta medför att den har fler inre kapacitancer  $\Rightarrow$  den inre fördröjningen **T<sub>i</sub>** (även kallad den intrinsiska fördröjningen) blir större.

# Grindar med flera ingångar



3-input NAND

Man använder sällan grindar med fler än **fyra** ingångar.

**Lång rad** av seriekopplade transistorer ger långsam funktion! Spänningsdelning. Låg spänning över varje transistor

# Hög Fan-in löses med trädstrukturer

Bara en  
kontakt?

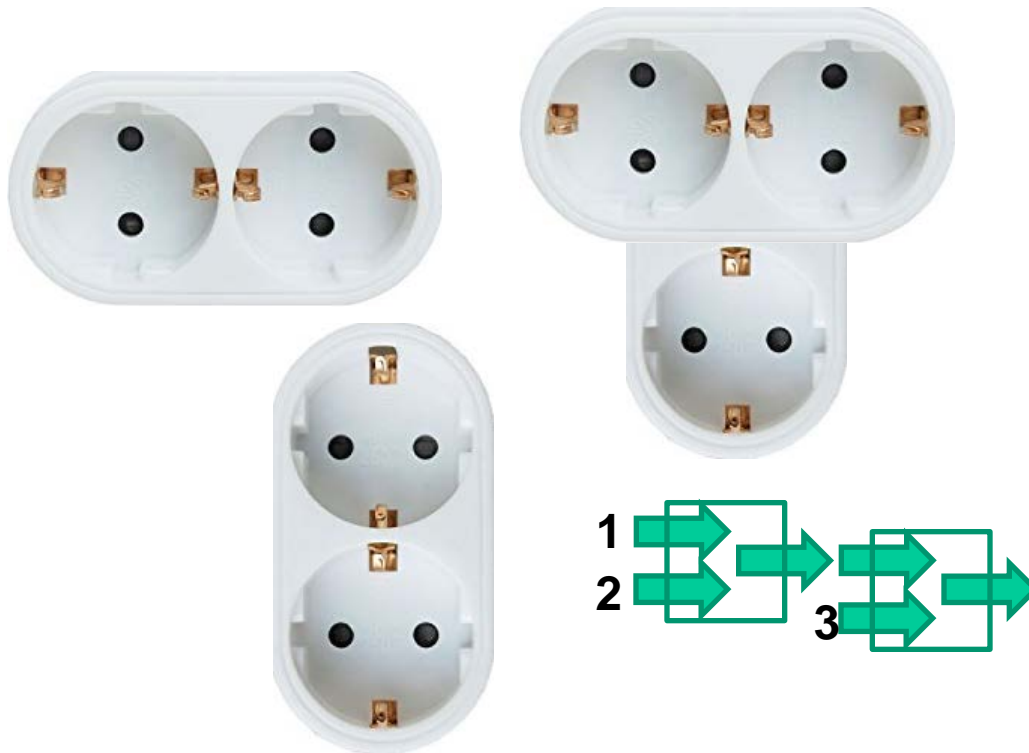


1 →

# Hög Fan-in löses med trädstrukturer

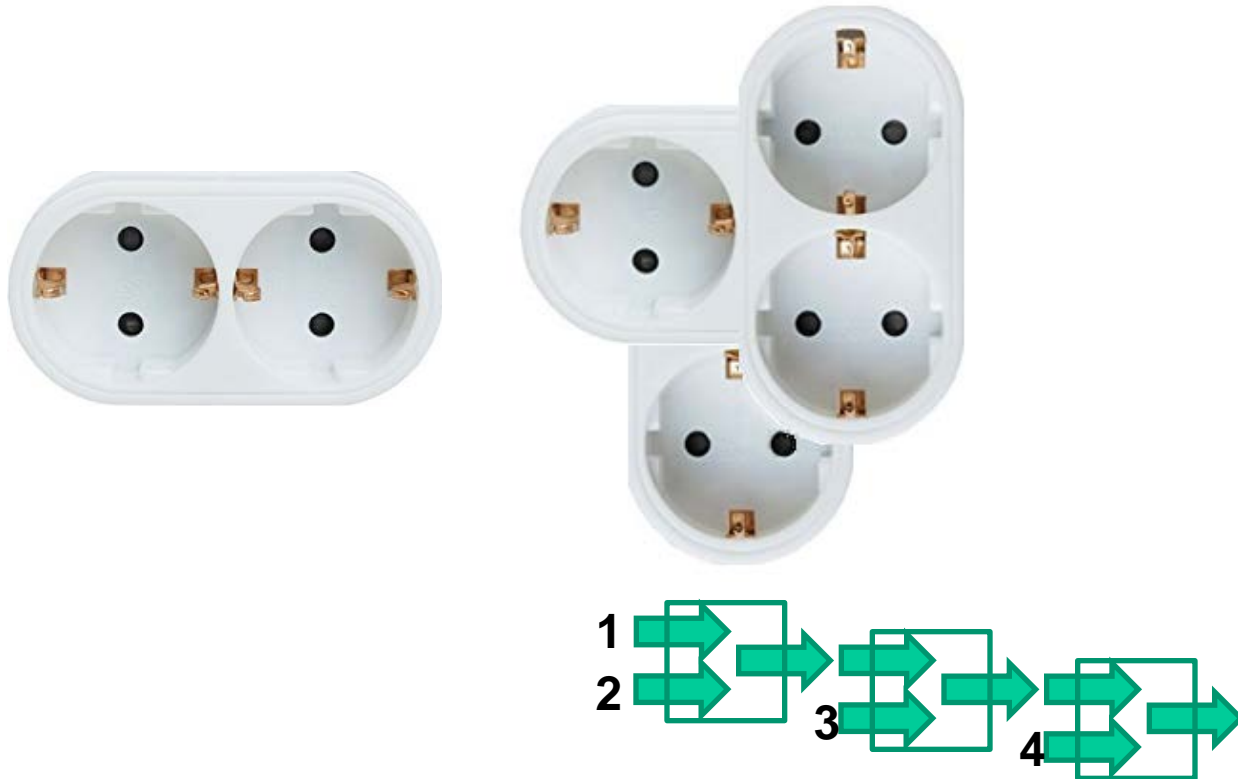


# Hög Fan-in löses med trädstrukturer





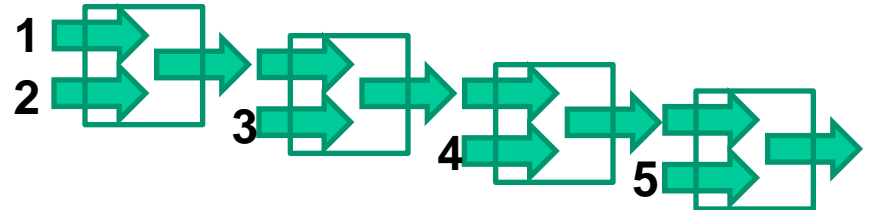
# Hög Fan-in löses med trädstrukturer



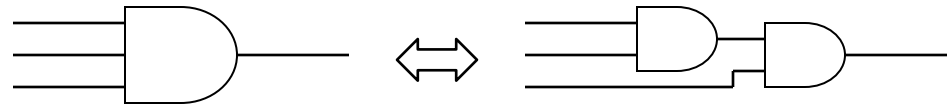
# Hög Fan-in löses med trädstrukturer



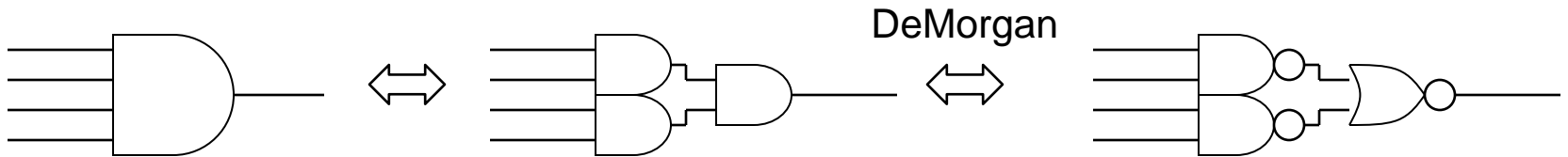
Nu fem  
kontakter!



# Hög Fan-in löses med trädstrukturer



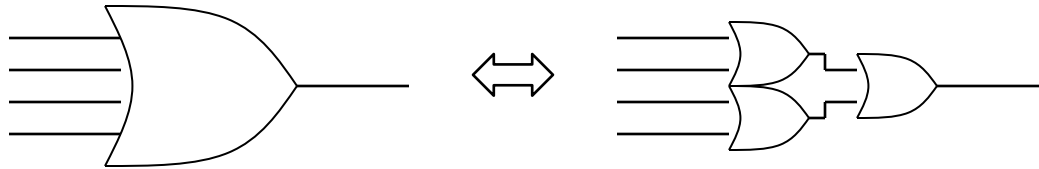
$$a \cdot b \cdot c = a \cdot (b \cdot c)$$



$$a \cdot b \cdot c \cdot d = (a \cdot b) \cdot (c \cdot d) \quad \overline{\overline{(a \cdot b)} + \overline{(c \cdot d)}} = a \cdot b \cdot c \cdot d$$

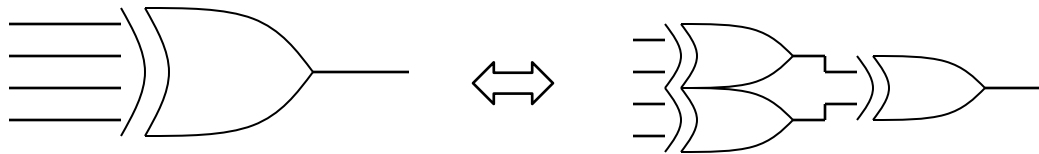
Till priset av ökat grind-djup (fördröjning)

# Fler trädstrukturer

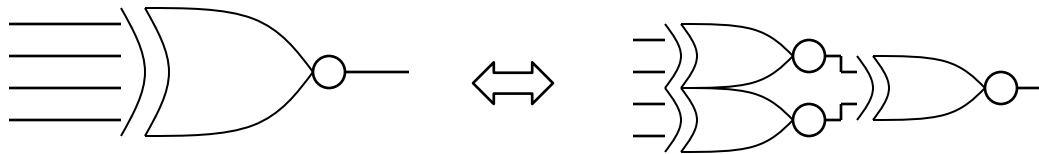


$$a + b + c + d = (a + b) + (c + d)$$

Till priset av  
ökat grind-djup  
(fördröjning),  
men effekten av  
inre kapacitanser  
hade blivit värre.



$$a \oplus b \oplus c \oplus d = (a \oplus b) \oplus (c \oplus d)$$

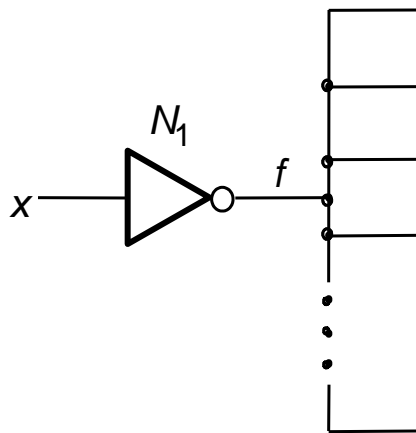


$$\overline{a \oplus b \oplus c \oplus d} = \overline{(a \oplus b) \oplus (c \oplus d)}$$

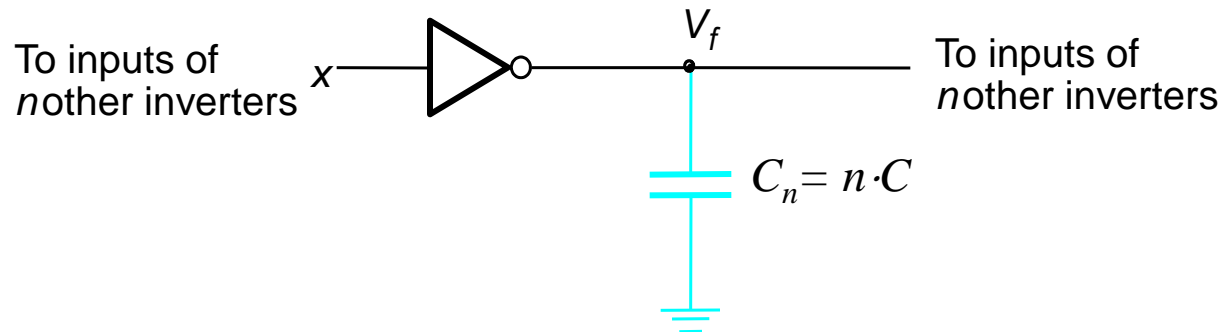
***Kan Du bevisa  
dessa likheter?***

# Fan-out

- Antalet grindar som en grind driver betecknas som fan-out
- Alla grindar som drivs ökar den kapacitativa lasten



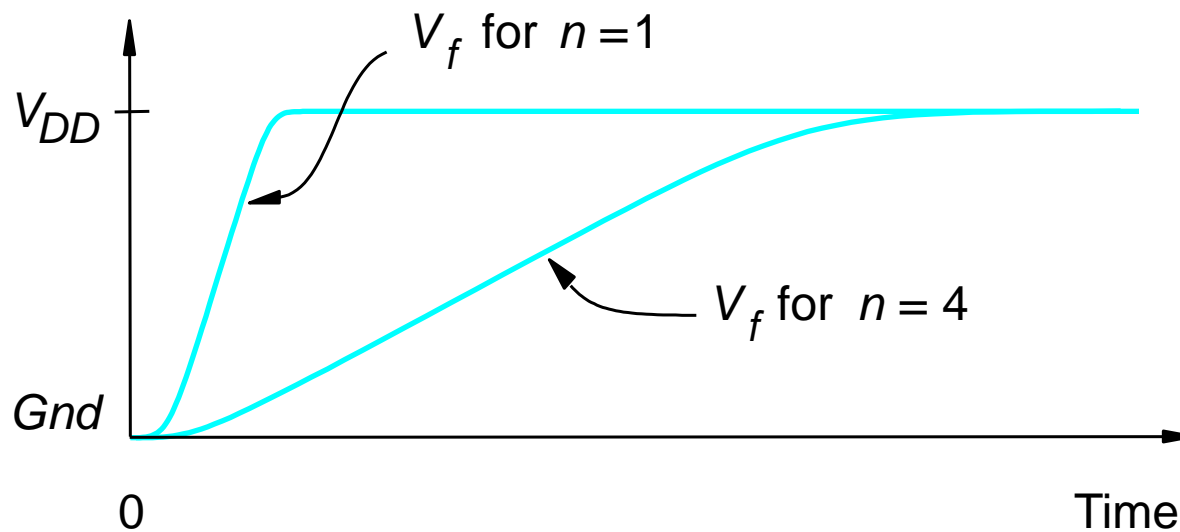
(a) Inverter that drives  $n$  other inverters



(b) Equivalent circuit for timing purposes

# Fan-out

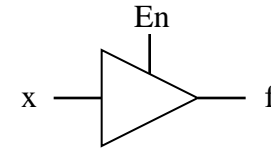
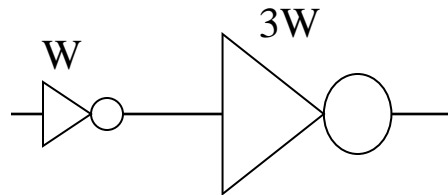
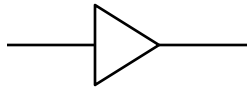
- Födröjningen för olika fan-outs



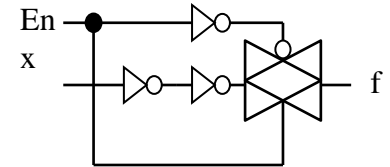
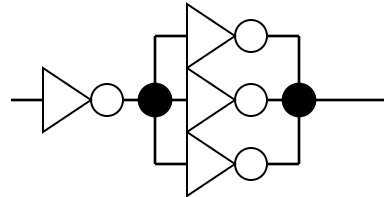
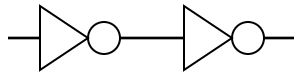
# Buffer

- En buffer är en krets som implementerar funktionen  $f(x) = x$  ( det vill säga ut = in )
- Idén med bufferten är att ökar drivförmågan av kapacitativa laster
  - För att öka drivförmågan så använder man större transistorer
  - Buffrar kan dimensioneras så att de kan driva större strömmar

# Hög Fan-out – använd buffer



x	En	f
0	0	Z
0	1	0
1	0	Z
1	1	1



**Non-inverting Buffer**

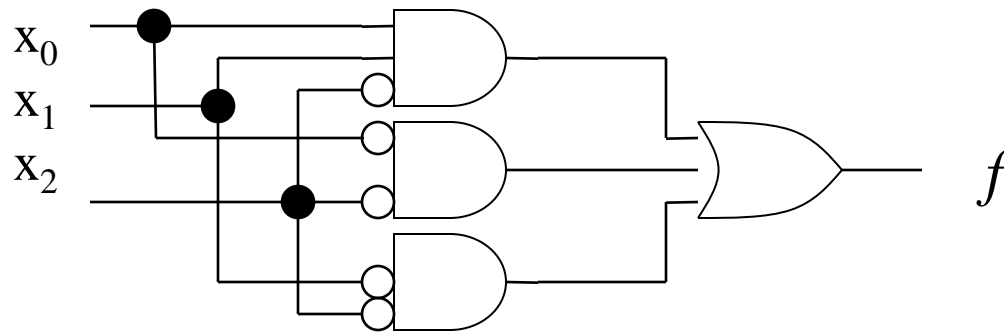
**High-Fan-Out Buffer**

**Tri-state Buffer**



# Critical path (den längsta vägen)

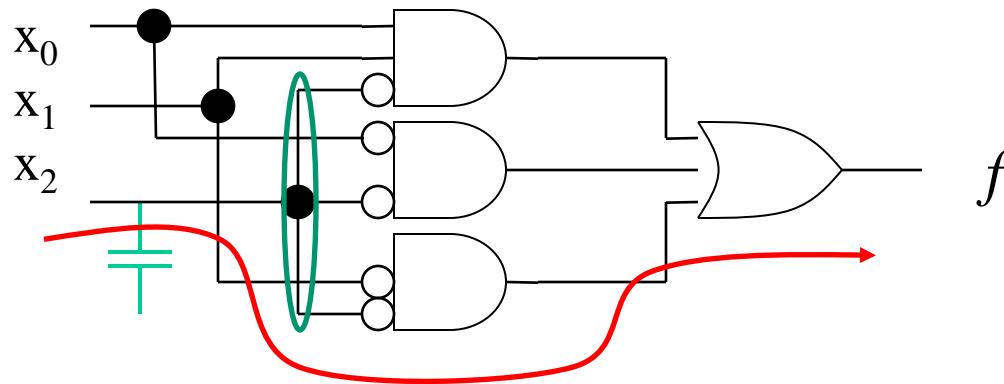
$$f = x_0 x_1 \bar{x}_2 + \bar{x}_0 \bar{x}_2 + \bar{x}_1 \bar{x}_2$$



**Vilken väg till utgången tar längst tid?  $x_0 x_1 x_2$  ?**

# ”Critical path”

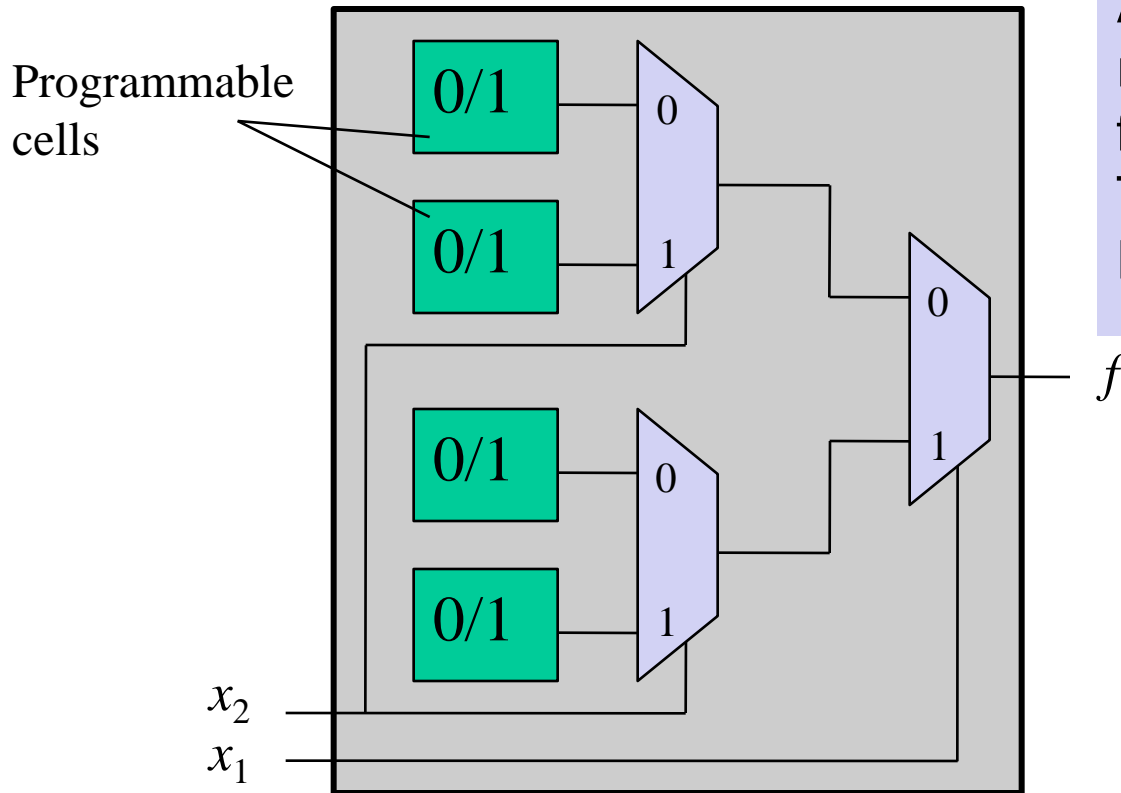
$$f = x_0 x_1 \overline{x_2} + \overline{x_0} \overline{x_2} + \overline{x_1} \overline{x_2}$$



$x_0 x_1 x_2$  passerar alla var sin NOT , AND, och OR, på vägen mot utgången  $f$ , men  $x_2$  belastas av *tre* ingångar,  $x_0$  och  $x_1$  bara av *två*. ”Critical path” blir  $x_2$  !



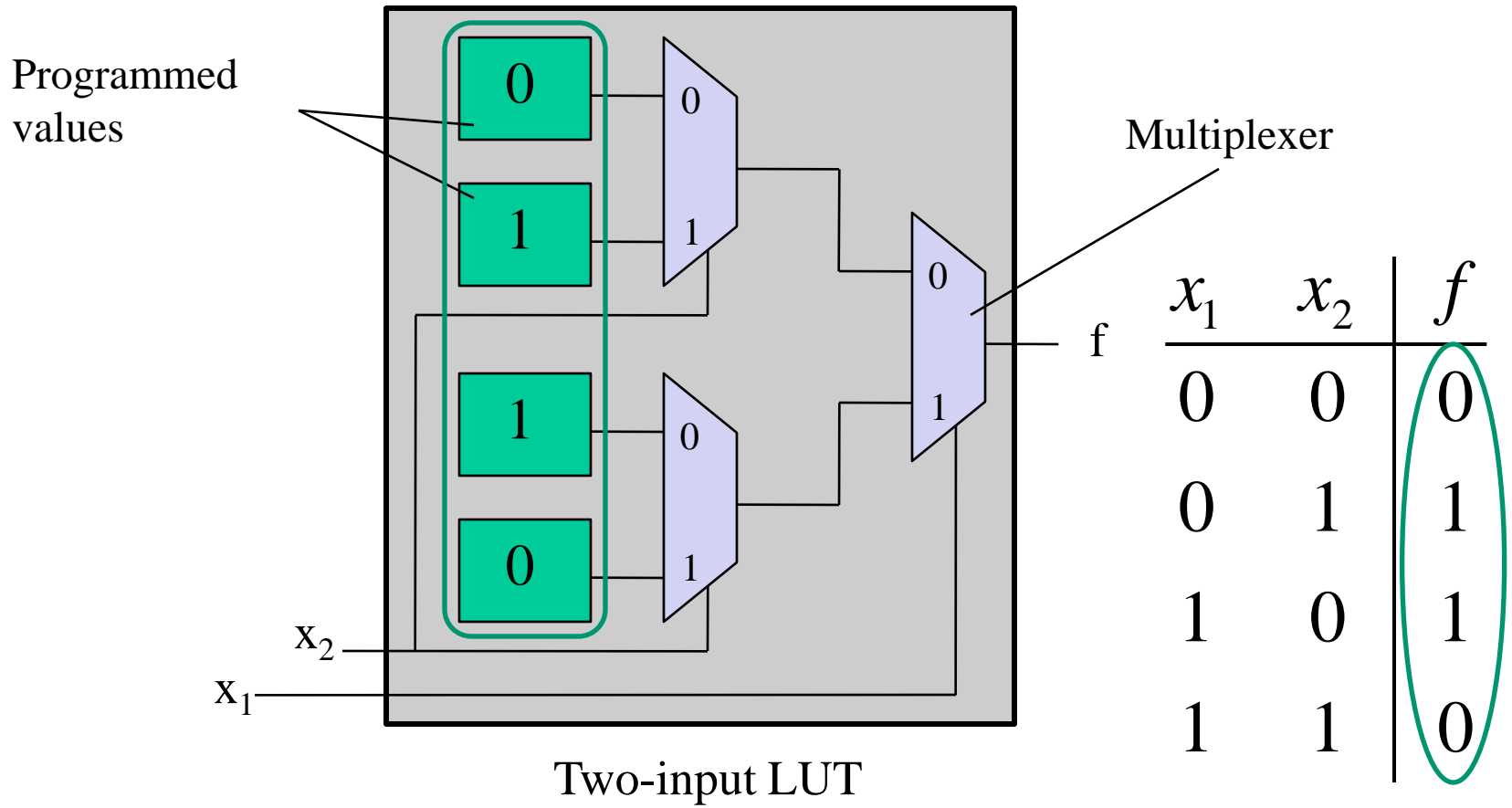
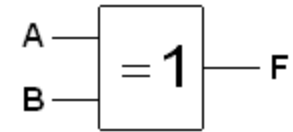
# Look-up-tables (LUT)



Two-input LUT

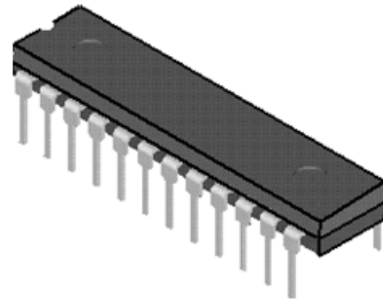
A LUT with  $n$  inputs can realize all combinational functions with  $n$  inputs  
The usual size in an FPGA is  $n=4$

# Ex. XOR-funktion

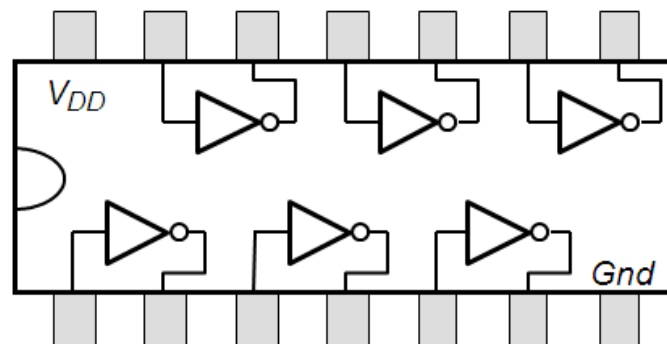




# 7400-series standard chips



(a) Dual-inline package

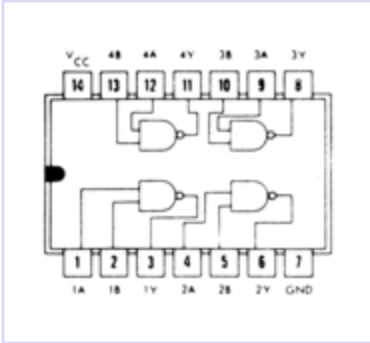


(b) Structure of 7404 chip

# Standardkretsarna används mest som reservdelar

**ELFA**  
Allt mellan antenn och jord

**Logikkrets DIL-14, 74HC00N**  
Aktiva komponenter > Digitala kretsar / Utvecklingsverktyg / Kristaller > 74-logik



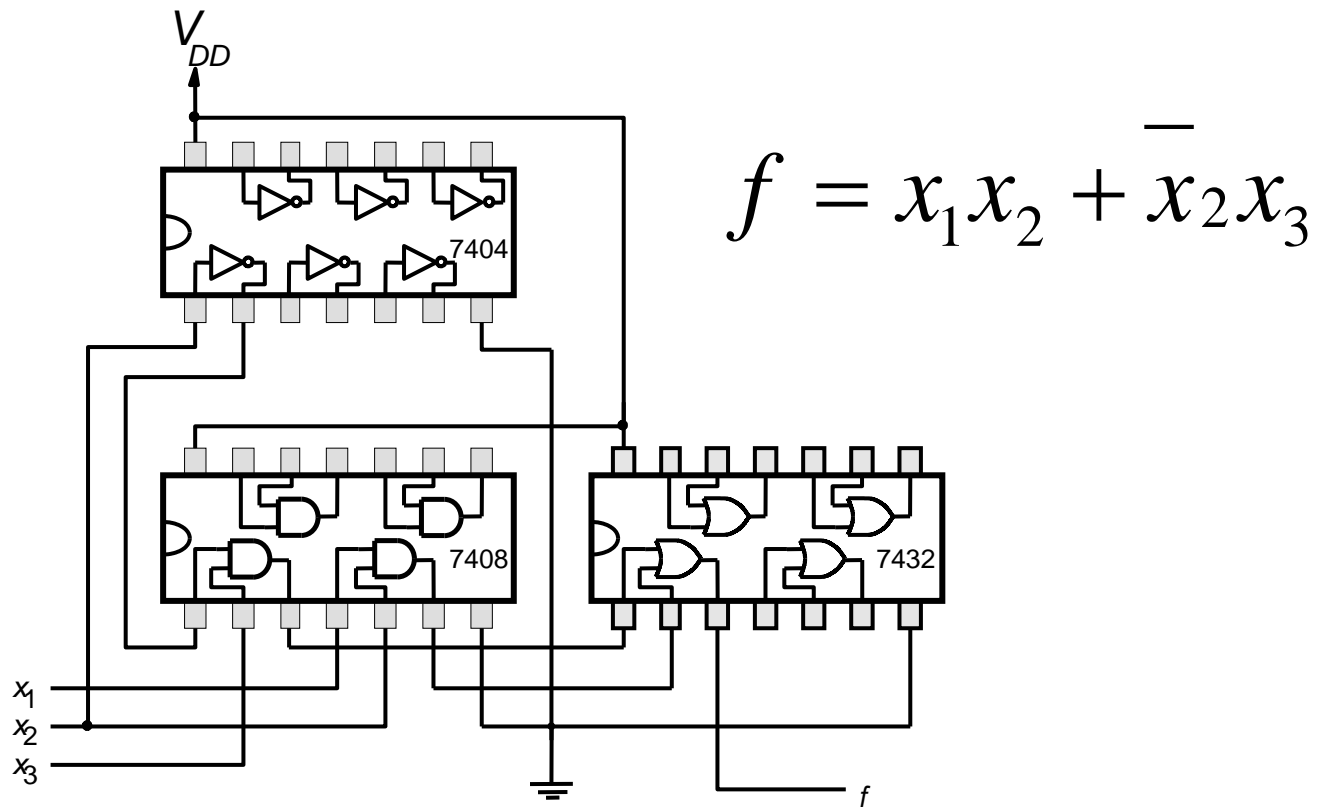
Kvantitet	Artikelnr.	Pris/styck
1	73-500-10	1- 5.77
<input type="button" value="Köp"/>		25- 3.35
		100- 2.42
<input checked="" type="checkbox"/> Levereras normalt samma dag		
<input checked="" type="checkbox"/> Uppfyller ROHS direktivet		
<b>Saldo</b>		
Centrallager Veddesta		<input type="button" value="Saldokontroll"/>
Tillgängligt lagersaldo		1175
Vi reserverar oss för mellanförsäring och lagersaldofel.		

*Inte så dyra!*

**Men många fler än skolorna behöver kretsarna. Det finns många kvar i lager ...**



# Implementering av en logisk funktion

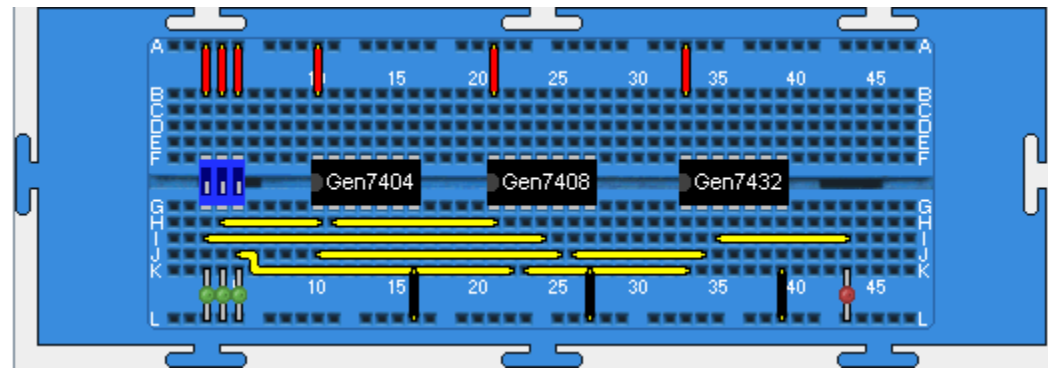
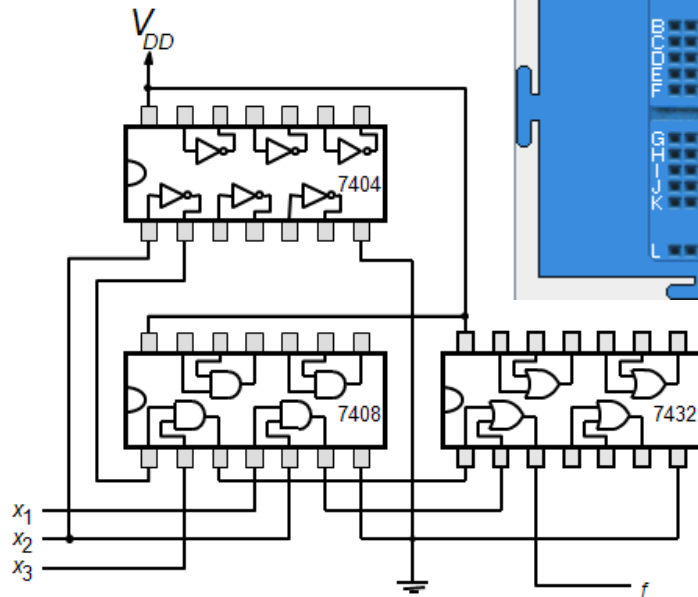


# Hur testar man logiska funktioner?

**Man kan koppla upp funktionen och kontrollmäta!**

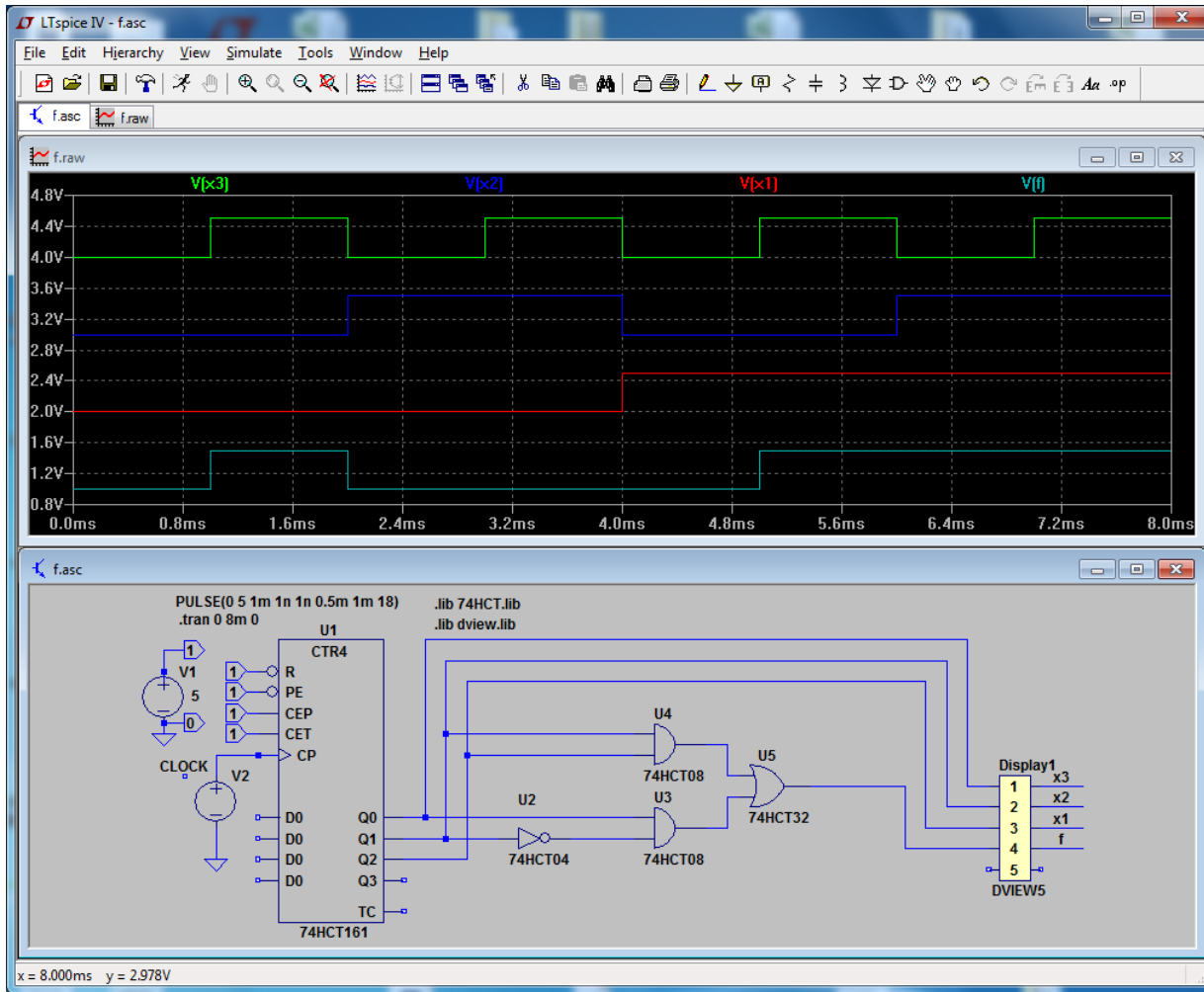
På kopplingsdäck:

Kapsel layout:



$$f = x_1x_2 + x_2x_3$$

# Hur testar man logiska funktioner?



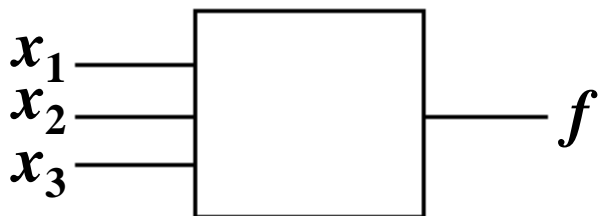
**Inför  
laborationerna  
simulerar vi  
funktionerna med  
LTSpice!**

$$f = x_1 x_2 + \overline{x_2} x_3$$

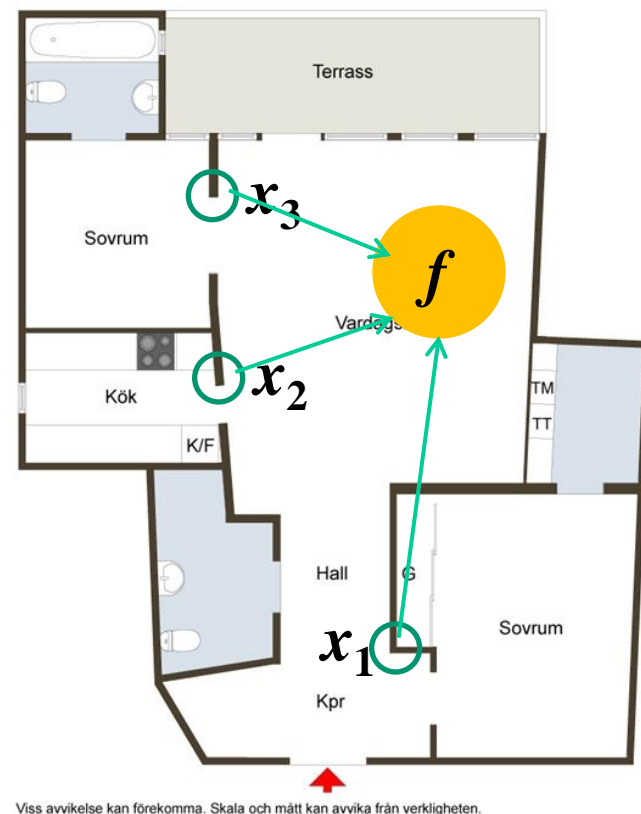
# Kommer Du ihåg? Trevägs ljuskontroll

## Brown/Vranesic: 2.8.1

Antag att vi behöver kunna tända/släcka vardagsrummet från tre olika ställen.



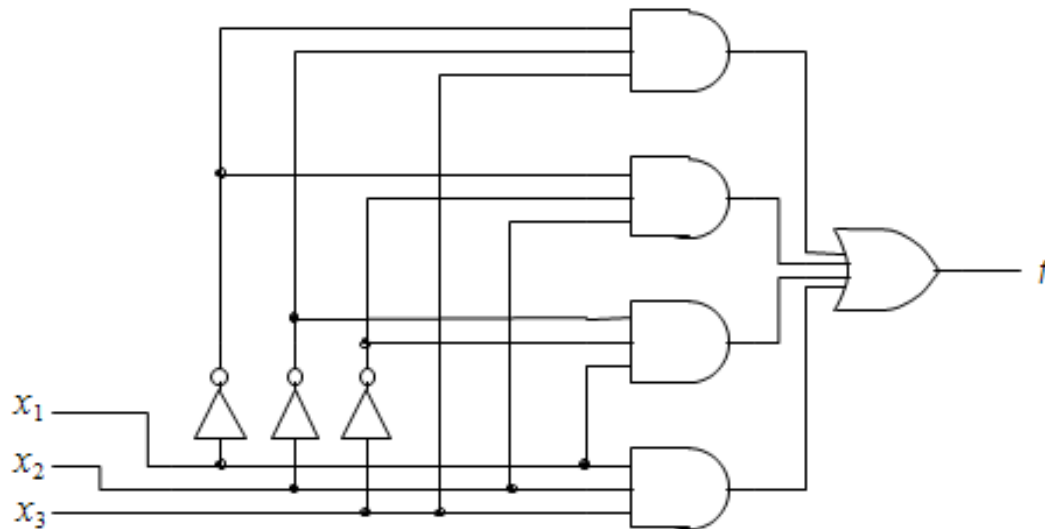
$x_1$	$x_2$	$x_3$	$f$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1



# Trevägs ljuskontroll



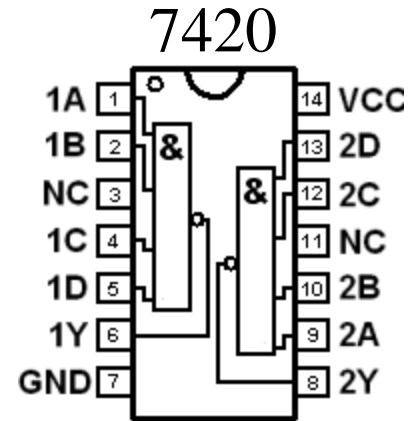
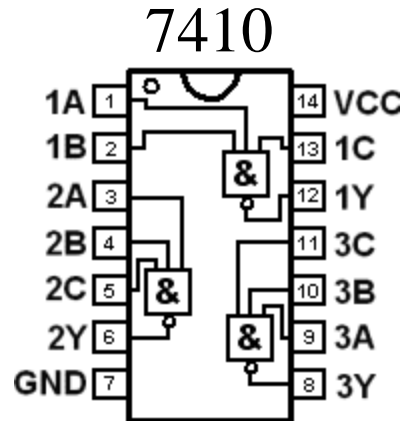
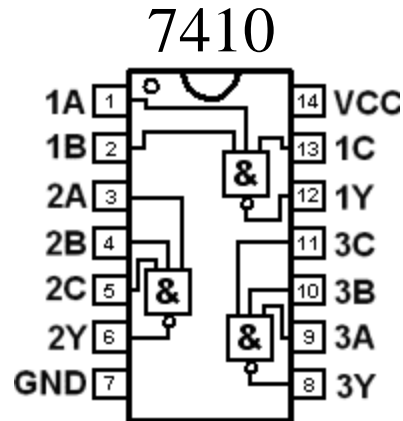
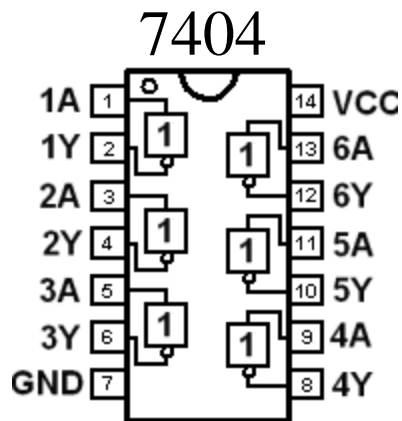
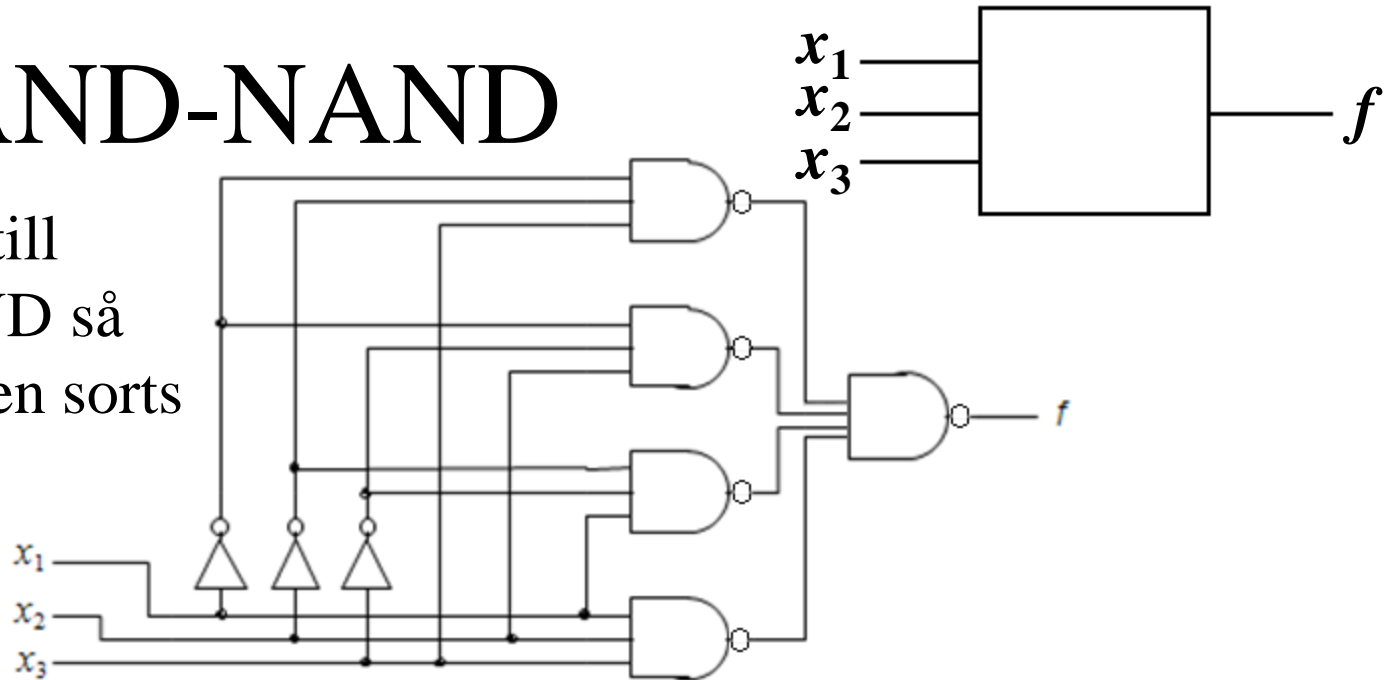
$$f = \sum m(1,2,4,7) = \bar{x}_1\bar{x}_2x_3 + \bar{x}_1x_2\bar{x}_3 + x_1\bar{x}_2\bar{x}_3 + x_1x_2x_3$$



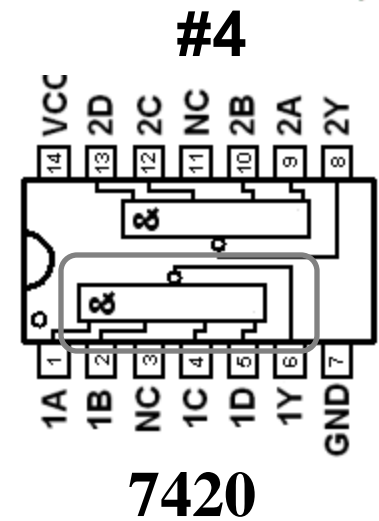
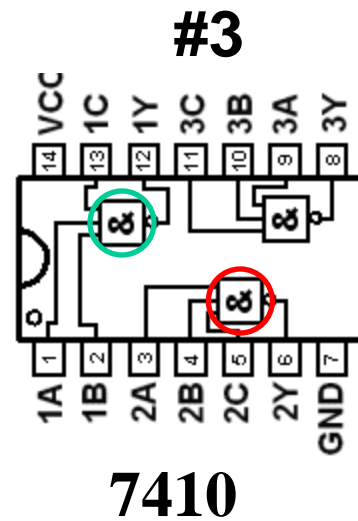
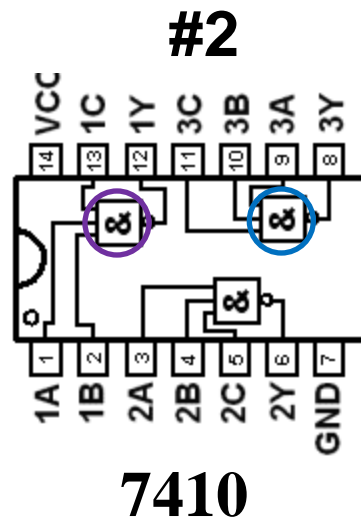
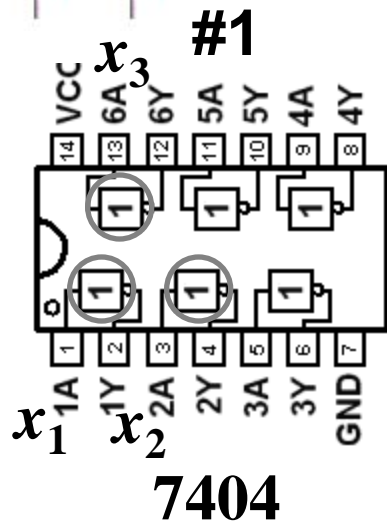
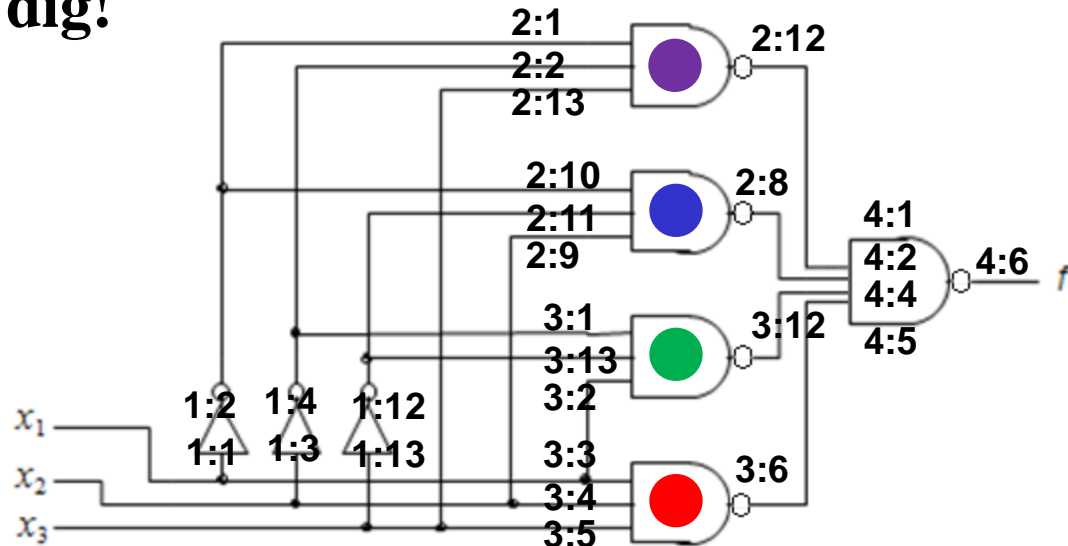
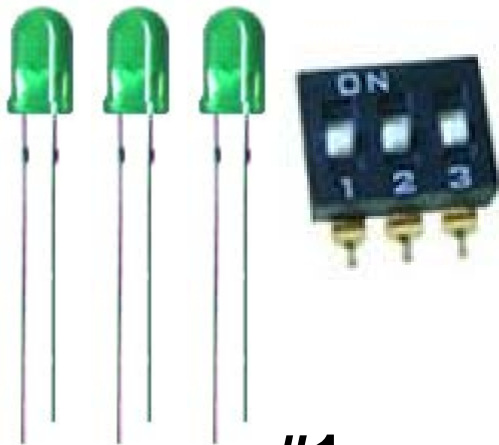
(a) Sum-of-products realization

# NAND-NAND

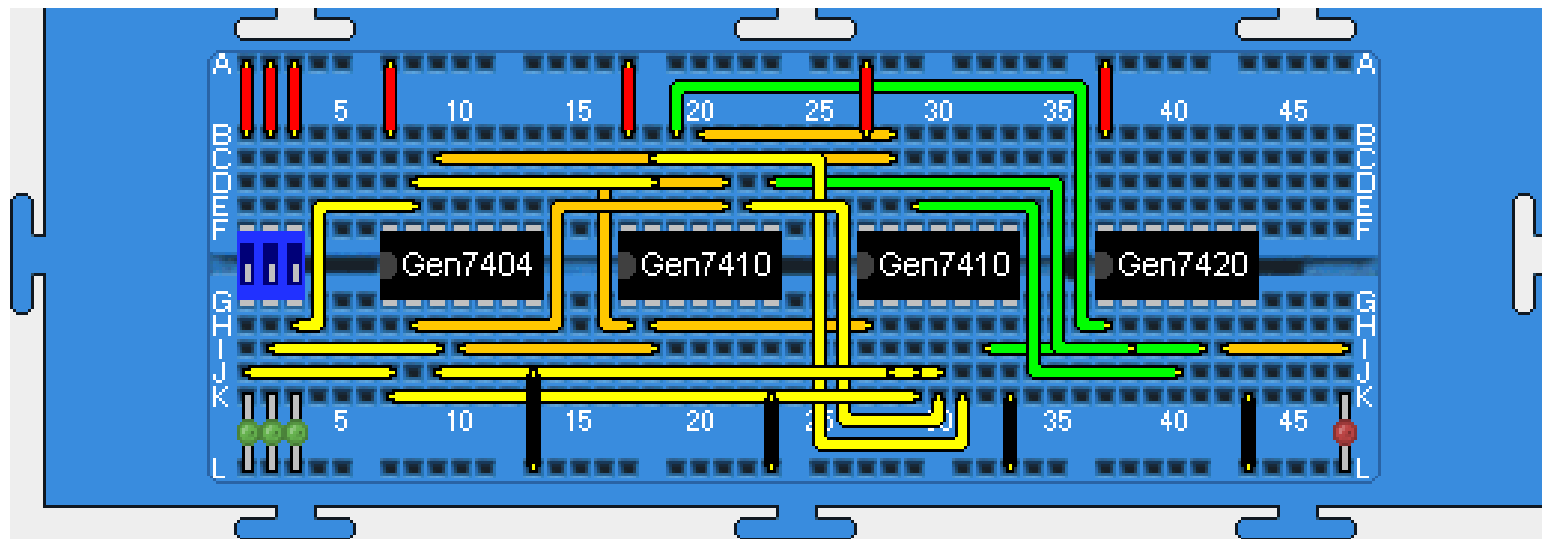
Om vi byter till  
NAND-NAND så  
behövs bara en sorts  
grindar.



**Du måste skriva dit pinn-nummer i schemat – annars kommer Du att villa bort dig!**

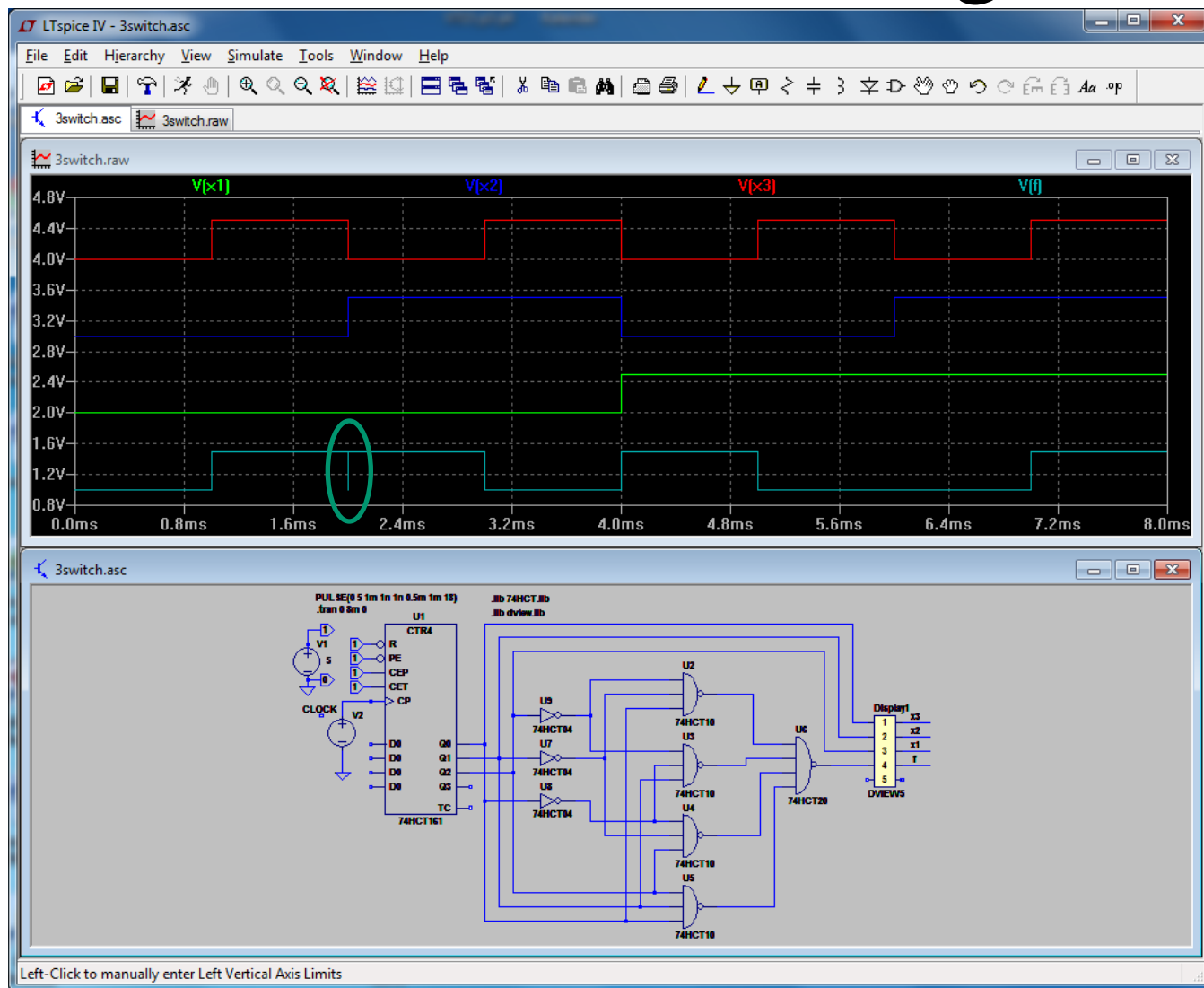


# På kopplingsdäck





# Simulera sanningstabellen



$x_1$	$x_2$	$x_3$	$f$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

*Något som Du vill fråga om?*



# Sammanfattning

- Logiska grindar kan implementeras med CMOS-teknologin
- CMOS-kretsar har en fördröjning
- **CMOS-kretsar förbrukar relativt lite effekt**



**1 Kall luft** sugas in på övervåningen. Där blandas den med uppvärmd luft från serverhallen så att den får rätt temperatur.

**2** Luften går sedan vidare genom ett luftfilter och vidare genom en vattendimma för att få rätt luftfuktighet.

**4** På baksidan av server-racken samlas den varma luften in och sugs tillbaka till luftintaget där den blandas med den kalla insugsluften.

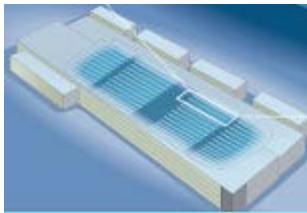
**Servrarna** har designats för att vara så strömsnåla och lättkylda som möjligt och har till exempel inget yttre hölje. De placeras i treradiga rack med 30 servrar i varje rad. Till varje rack finns en batterireserv.

**3** Luften slussas sedan ner i själva serverhallen där den blåser förbilservarna.

# Facebooks första serverhall i Luleå. Argumenten är kylan och ett elnät som inte haft avbrott sedan 1979.

# Facebooks serverhall i Luleå

Driften av de tusentals servrarna slukar enorma mängder energi. Fullt utbyggd kräver anläggningen **120 MW**, mer än SSAB:s stålverk!



# Hur skulle världen vara utan CMOS?

