

CHALMERS  
UNIVERSITY OF TECHNOLOGY

## Digital- och datorteknik



Föreläsning #4

Biträdande professor Jan Jonsson  
 Institutionen för data- och informationsteknik  
 Chalmers tekniska högskola

CHALMERS  
UNIVERSITY OF TECHNOLOGY

## Boolesk algebra

**SP- och PS-form:**

Vid förra föreläsningen konstaterade vi att funktionen hos en krets kan uttryckas på två olika former:

- Som en summa av produkter, SP-form, även kallad disjunktiv form. Exempel: uttrycket  $x+yz$  är på SP-form.
- Som en produkt av summor, PS-form, även kallad konjunktiv form. Exempel: uttrycket  $(x+y)(x+z)$  är på PS-form.

SP-form och PS-form för samma funktion är alltså ekvivalenta, så endera formen kan väljas.

CHALMERS  
UNIVERSITY OF TECHNOLOGY

## Boolesk algebra

**Mintermer och maxtermer:**

Utgående från funktionstabellen för en given Boolesk funktion är det möjligt att härleda funktionens SP-form respektive PS-form:

- För SP-form skall vi identifiera de rader i tabellen som har ett funktionsvärde lika med '1'. Den unika produkten av invariabler för en sådan rad kallas för en minterm. Summan av alla mintermer kallas för SP normal form.
- För PS-form skall vi identifiera de rader i tabellen som har ett funktionsvärde lika med '0'. Den unika summan av invariabler för en sådan rad kallas för en maxterm. Produkten av alla maxtermer kallas för PS normal form.

CHALMERS  
UNIVERSITY OF TECHNOLOGY

## Boolesk algebra

**Samtliga mintermer och maxtermer för tre variabler:**

Minterm	Är 1 om		
	x	y	z
$\bar{x} \cdot \bar{y} \cdot \bar{z}$	0	0	0
$\bar{x} \cdot \bar{y} \cdot z$	0	0	1
$\bar{x} \cdot y \cdot \bar{z}$	0	1	0
$\bar{x} \cdot y \cdot z$	0	1	1
$x \cdot \bar{y} \cdot \bar{z}$	1	0	0
$x \cdot \bar{y} \cdot z$	1	0	1
$x \cdot y \cdot \bar{z}$	1	1	0
$x \cdot y \cdot z$	1	1	1

Maxterm	Är 0 om		
	x	y	z
$x + y + z$	0	0	0
$x + y + \bar{z}$	0	0	1
$x + \bar{y} + z$	0	1	0
$x + \bar{y} + \bar{z}$	0	1	1
$\bar{x} + y + z$	1	0	0
$\bar{x} + y + \bar{z}$	1	0	1
$\bar{x} + \bar{y} + z$	1	1	0
$\bar{x} + \bar{y} + \bar{z}$	1	1	1

CHALMERS  
UNIVERSITY OF TECHNOLOGY

## Boolesk algebra

Ta fram PS normal form för  
 $(x + y)(x + z)$

CHALMERS  
UNIVERSITY OF TECHNOLOGY

## Karnaughminimering

**Minimering av Booleska funktioner:**  
 Om vi inte är nöjda med den lösning som ges av funktionens SP normal form eller PS normal form, kan vi härleda ett minimalt uttryck genom Karnaughminimering.

Denna metod bygger på att man för in funktionstabellen i en matris (Karnaughdiagram) med invariablernas olika värden längs rader och kolumner.

Varje ruta i matrisen representerar funktionsvärdet för en minterm (för SP form) eller en maxterm (för PS form).

CHALMERS  
UNIVERSITY OF TECHNOLOGY

## Karnaughminimering

**Minimering av Booleska funktioner:**  
 För att ge möjlighet till effektiv eliminering av onödiga variabler skall intilliggande rader respektive kolumner representera termer ordnade enligt Gray-kod, d v s de skiljer sig åt i en bitposition.

Ringa in så stora grupper av 1:or (för SP form) eller 0:or (för PS form) som möjligt, och ta fram uttrycken för dessa.

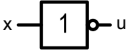
Gör Karnaughminimering på PS normal form för  
 $(x + y)(x + z)$

CHALMERS  
UNIVERSITY OF TECHNOLOGY

## Logikgrindar

**De grundläggande logikoperationerna:**


INVERTERARE  
(ICKE, NOT)



x	y	u
0	1	0
1	0	1

$u = \bar{x}$


ELLER (OR)



x	y	u
0	0	0
0	1	1
1	0	1
1	1	1

$u = (x + y)$

OCH (AND)



x	y	u
0	0	0
0	1	0
1	0	0
1	1	1

$u = (x * y)$

**CHALMERS**  
UNIVERSITY OF TECHNOLOGY

## Logikgrindar

**Logikgrindar "på riktigt":**  
 Under 1900-talet användes tre olika typer av komponenter för att realisera logikgrindar:

- Reläer
- Elektronrör
- Transistorer

Gemensamt för dessa komponenter är att de har en styrsignal som öppnar och stänger en "grind" som reglerar ett annat signalflöde.

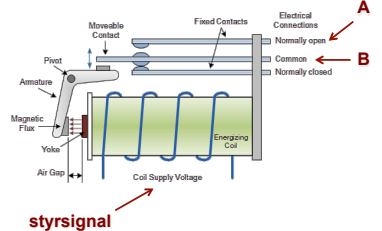

Likt en vattenventil, fast med elektronisk styrning av elektriska signaler.



**CHALMERS**  
UNIVERSITY OF TECHNOLOGY

## Logikgrindar

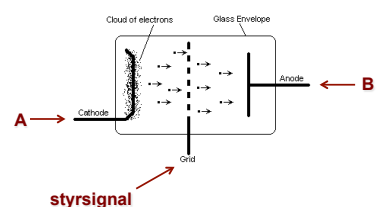

**Reläer:**  
 Styrsignal "på" ger ström genom en spole som aktiverar en elektromagnet. Denna manipulerar en strömbrytare som kan användas för att öppna och stänga förbindelsen mellan A och B.

**CHALMERS**  
UNIVERSITY OF TECHNOLOGY

## Logikgrindar

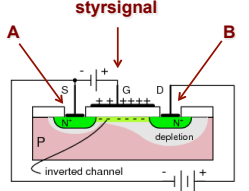
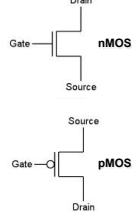
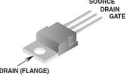
**Elektronrör:**  
 Styrsignal "på" lägger ett elektriskt fält på ett galler ("grid") som påverkar den ström som släpps igenom mellan katod och anod. Gallret kan alltså öppna och stänga förbindelsen mellan A och B.

**CHALMERS**  
UNIVERSITY OF TECHNOLOGY

## Logikgrindar

**Transistor (nMOS):**  
 Styrsignal "på" lägger ett positivt elektriskt fält på en grind ("gate") varpå en ledande kanal i kiselsubstratet öppnas upp. Grinden kan alltså öppna och stänga förbindelsen mellan A och B.

pMOS-transistorn fungerar på ett liknande sätt men har inverterad logik.

**CHALMERS**  
 UNIVERSITY OF TECHNOLOGY

## Logikgrindar

**Inverterare (ICKE, NOT) "på riktigt":**

Insignal "på" ( $x = 1$ ) lägger ett positivt elektriskt fält på den nedre transistorens (nMOS) grind varpå dess kanal öppnas och släpper igenom den logiska signalen "0" till utsignal  $u$ .

Samtidigt ges samma insignal till den övre transistoren (pMOS) vars inverterade logik gör att dess kanal stängs, och förhindrar att den logiska signalen "1" når utsignal  $u$ .

Resultat:  $x = 1$  ger utsignalen  $u = 0$

**CHALMERS**  
 UNIVERSITY OF TECHNOLOGY

## Logikgrindar

**Inverterare (ICKE, NOT) "på riktigt":**

Insignal "av" ( $x = 0$ ) tar bort det positiva elektriska fältet på den nedre transistorens (nMOS) grind varpå dess kanal stängs och förhindrar att den logiska signalen "0" når utsignal  $u$ .

Samtidigt ges samma insignal till den övre transistoren (pMOS) vars inverterade logik gör att dess kanal öppnas och släpper igenom den logiska signalen "1" till utsignal  $u$ .

Resultat:  $x = 0$  ger utsignalen  $u = 1$

**CHALMERS**  
 UNIVERSITY OF TECHNOLOGY

## Logikgrindar

**ELLER (OR) "på riktigt":**

Om båda insignalerna är "av" ( $x = 0$  och  $y = 0$ ) tas det positiva elektriska fältet bort på de två nMOS-transistorens grindar och deras kanaler stängs.

Samtidigt ges samma insignal till de två pMOS-transistorerna vars inverterade logik gör att deras kanaler öppnas. Detta gör att signalen  $w = 1$ .

Vi ser att de två transistorerna till höger utgör en inverterare, vilket innebär att utsignalen  $u = 0$

**CHALMERS**  
 UNIVERSITY OF TECHNOLOGY

## Logikgrindar

**ELLER (OR) "på riktigt":**

Om den övre insignalen slås "på" ( $x = 1$ ) läggs ett positivt elektriskt fält på den vänstra nMOS-transistorns grind varpå dess kanal öppnas upp.

Samtidigt ges samma insignal till den övre pMOS-transistorn vars inverterade logik gör att dess kanal stängs. Detta gör att signalen  $w = 0$ .

Vi ser att de två transistorerna till höger utgör en inverterare, vilket innebär att utsignalen  $u = 1$

Notera att samma resultat erhålls för  $y = 1$ .

**CHALMERS**  
UNIVERSITY OF TECHNOLOGY

## Logikgrindar

**OCH (AND) "på riktigt":**

Om båda insignalen är "på" ( $x = 1$  och  $y = 1$ ) läggs ett positivt elektriskt fält på de två nMOS-transistorernas grindar och deras kanaler öppnas.

Samtidigt ges samma insignal till de två pMOS-transistorerna vars inverterade logik gör att deras kanaler stängs.

Detta gör att signalen  $w = 0$ .

Vi ser att de två transistorerna till höger utgör en inverterare, vilket innebär att utsignalen  $u = 1$

**CHALMERS**  
UNIVERSITY OF TECHNOLOGY

## Logikgrindar

**OCH (AND) "på riktigt":**

Om den övre insignalen slås "av" ( $x = 0$ ) tas det positiva elektriskt fältet bort på den nedre nMOS-transistorns grind varpå dess kanal stängs.

Samtidigt ges samma insignal till den vänstra pMOS-transistorn vars inverterade logik gör att dess kanal öppnas.

Detta gör att signalen  $w = 1$ .

Vi ser att de två transistorerna till höger utgör en inverterare, vilket innebär att utsignalen  $u = 0$

Notera att samma resultat erhålls för  $y = 0$ .

**CHALMERS**  
UNIVERSITY OF TECHNOLOGY

## Logikgrindar

**NOR och NAND:**

Vi såg att det vid realisering av ELLER- och OCH-grinden med transistorer sitter en inverterare på grindens utgång. Detta gör att realisering av grindnät i många sammanhang kan kräva onödigt många transistorer.

**CHALMERS**  
UNIVERSITY OF TECHNOLOGY

## Logikgrindar

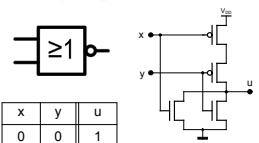
**NOR och NAND:**

Av det skälet brukar man istället använda de negerade logiska funktionerna som grundläggande byggblock. Dessa byggblock kallas för NOR- respektive NAND-grind.

**CHALMERS**  
UNIVERSITY OF TECHNOLOGY

## Logikgrindar

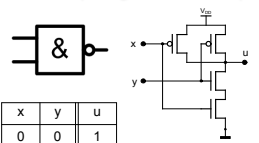
### NOR ("negated OR")



x	y	u
0	0	1
0	1	0
1	0	0
1	1	0

$u = \overline{(x + y)}$

### NAND ("negated AND")



x	y	u
0	0	1
0	1	1
1	0	1
1	1	0

$u = \overline{(x * y)}$

Vi skall senare visa hur man enkelt kan översätta grindnät på SP-form (AND/OR-logik) till NAND-logik, och grindnät på PS-form (OR/AND-logik) till NOR-logik. På så sätt får grindnätet färre transistorer, och blir dessutom snabbare.