

Boolsk algebra med digitaltekniskt fokus

Föreläsning 2

Digitalteknik, TSEA22

Mattias Krysanter

Institutionen för systemteknik

lakttagelser från föreläsning 1

a)



$y = (a \cdot a)'$

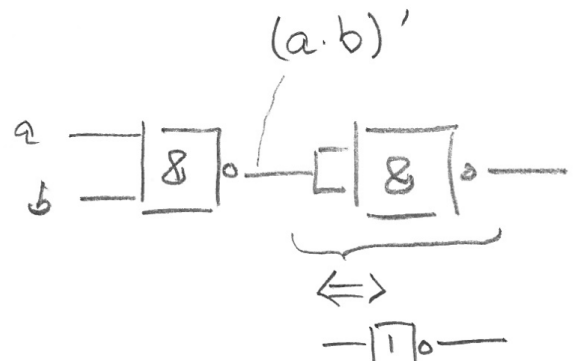
a	a · a	y = (a · a)'
0	0	1
1	1	0

$$a \cdot a = a$$

$$y = a'$$

Kretsens funktion är ekvivalent med en inverterare.

b)



$y = ((a \cdot b)')' = a \cdot b$

dubbel invertering tar ut varandra, Bodesk algebra

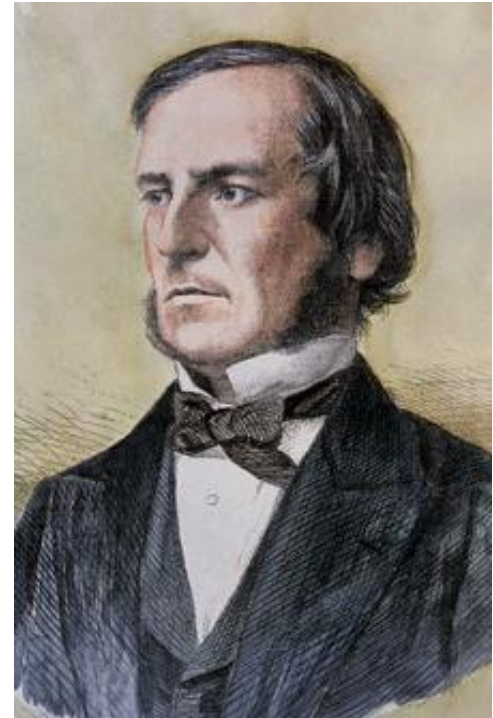
$$((a)')' = a$$

Boolesk algebra

Matematisk grund för

- Logik
- Används för analys och syntes av digitala kretsar

George Boole var en engelsk matematiker som skapade grunderna till Boolesk algebra



George Boole 1815-1864

Axiom

- Variabelvärden: 0 (falskt), 1 (sann)
- Operationer: eller, och, icke

OR (ELLER)

$$0 + 0 = 0 \quad (\text{A1})$$

$$0 + 1 = 1 \quad (\text{A5})$$

$$1 + 0 = 1 \quad (\text{A5})$$

$$1 + 1 = 1 \quad (\text{A3})$$

AND (OCH)

$$0 \cdot 0 = 0 \quad (\text{A4})$$

$$0 \cdot 1 = 0 \quad (\text{A6})$$

$$1 \cdot 0 = 0 \quad (\text{A6})$$

$$1 \cdot 1 = 1 \quad (\text{A2})$$

NOT (ICKE)

$$0' = 1 \quad (\text{A7})$$

$$1' = 0 \quad (\text{A8})$$

OBS: axiomen definierar sanningstabellerna för motsvarande grindar. Betrakta t ex OR-grinden:

a	b	y	Axiom
0	0	0	(A1)
0	1	1	(A5)
1	0	1	(A5)
1	1	1	(A3)

Räknelagar för en variabel

- Räknelagar kan härledas från axiomen. Dom ska man kunna använda utan att slå upp.
- Anta att x är en variabel som kan anta värdet 0 eller 1. Då gäller:

OR (ELLER)

$$\begin{aligned} x + x &= x & (L1) \\ x + x' &= 1 & (L3) \end{aligned}$$

$$\begin{aligned} x + 1 &= 1 & (L5) \\ x + 0 &= x & (L7) \end{aligned}$$

AND (OCH)

$$\begin{aligned} x \cdot x &= x & (L2) \\ x \cdot x' &= 0 & (L4) \end{aligned}$$

$$\begin{aligned} x \cdot 1 &= x & (L8) \\ x \cdot 0 &= 0 & (L6) \end{aligned}$$

NOT (ICKE)

$$(x')' = x \quad (L9)$$

- Gå igenom lagarna och förvissa er om att ni förstår dem.

Räknelagar för flera variabler

Associativa lagar

$$x + (y + z) = (x + y) + z \quad (\text{L10})$$

$$x(yz) = (xy)z \quad (\text{L11})$$

Absorptionslagar

$$x + xy = x \quad (\text{L16})$$

$$x(x + y) = x \quad (\text{L17})$$

Kommutativa lagar

$$x + y = y + x \quad (\text{L12})$$

$$xy = yx \quad (\text{L13})$$

Consensuslagar

$$xy + x'z = xy + x'z + yz \quad (\text{L18})$$

$$(x + y)(x' + z) = (x + y)(x' + z)(y + z) \quad (\text{L19})$$

Distributiva lagar

$$x(y + z) = xy + xz \quad (\text{L14})$$

$$x + yz = (x + y)(x + z) \quad (\text{L15})$$

De Morgans lagar

$$(x + y)' = x'y' \quad (\text{L20})$$

$$(xy)' = x' + y' \quad (\text{L21})$$

- (L10)-(L14) gäller på samma sätt som för reella tal.
- (L15) anger att även addition är distributivt i Boolesk algebra.
- (L16)-(L21) saknar motsvarighet för reella tal och är värda mer uppmärksamhet.

Axiom och räknelagar för XOR

Axiom	Räknelagar för en variabel	Räknelagar för flera variabler
$0 \oplus 0 = 0$ (E1)	$x \oplus 0 = x$ (E4)	$x \oplus (y \oplus z) = (x \oplus y) \oplus z$ (E6)
$0 \oplus 1 = 1$ (E2)	$x \oplus x = 0$ (E5)	$x \oplus y = y \oplus x$ (E7)
$1 \oplus 0 = 1$ (E2)		$x(y \oplus z) = xy \oplus xz$ (E8)
$1 \oplus 1 = 0$ (E3)		$x \oplus y = x \oplus z \Leftrightarrow y = z$ (E9)

Koppling mellan XOR och operationer i Boolesk algebra

XOR uttryckt i Boolesk algebra

$$x \oplus y = x' y + x y' \quad (\text{E10})$$

$$(x \oplus y)' = xy + x' y' \quad (\text{E11})$$

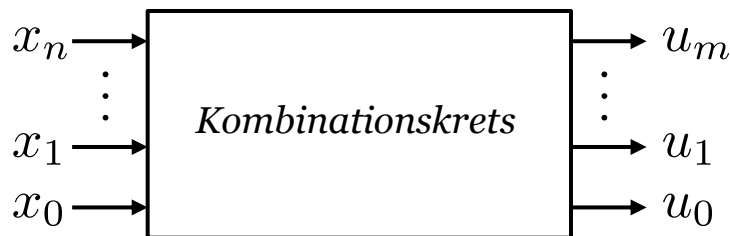
OR och NOT uttryckt i XOR

$$x + y = x \oplus y \oplus xy \quad (\text{E12})$$

$$x' = x \oplus 1 \quad (\text{E13})$$

Kombinationskretsar

En krets där utsignalvärdet vid en godtycklig tidpunkt endast beror av insignalvärdet vid samma tidpunkt.



$$u = f(x)$$

där

$$x = (x_n, x_{n-1}, \dots, x_1, x_0)$$

$$u = (u_m, u_{m-1}, \dots, u_1, u_0)$$

$$u_i, x_j \in \{0, 1\}$$

Syntes av kombinationskretsar

Ex: OR-funktionen

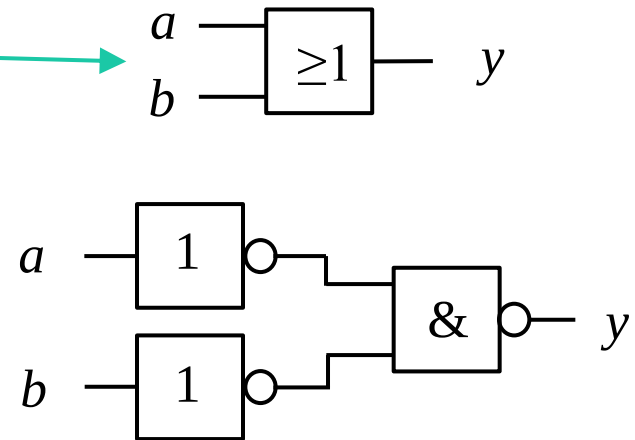
Funktionstabell

a	b	y
0	0	0
0	1	1
1	0	1
1	1	1

Booleska uttryck

$$\begin{aligned}
 y &= a + b = \\
 &= (a + b)'' = \\
 &= (a'b')'
 \end{aligned}$$

Realisering med grindnät



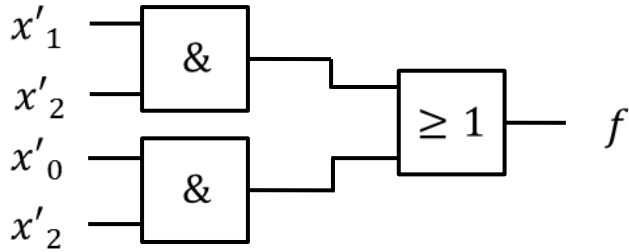
- Det finns flera olika uttryck som beskriver samma funktion.
- Uttrycken går att översätta till grindnät och vice versa.
- En funktion kan realiseras med grindnät av olika komplexitet och olika grindtyper.

Syntesfrågeställningar

- Hur ställer vi upp Booleska uttryck för en funktion specificerad av en funktionstabell?
- Hur hittar vi ett uttryck som svarar mot ett billigt grindnät?
 - Billig: få ingångar, få grindar, liten kiselarea
 - Bivillkor:
 - får använda en viss typ av grindar
 - får plats på en given programmerbar krets (PLD)
 - maximal fördröjning från insignal till utsignal, begränsar grinddjupet (maximalt antal grindar som en signal får passera)

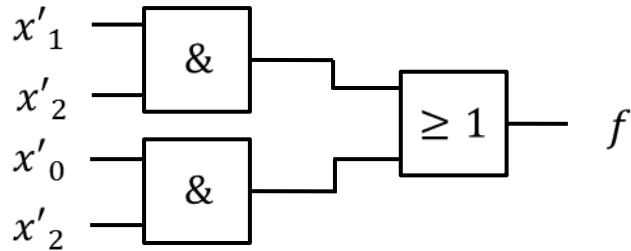
Grindnätstyper från SP-form

AND-OR nät: $f = x'_2 x'_1 + x'_2 x'_0$

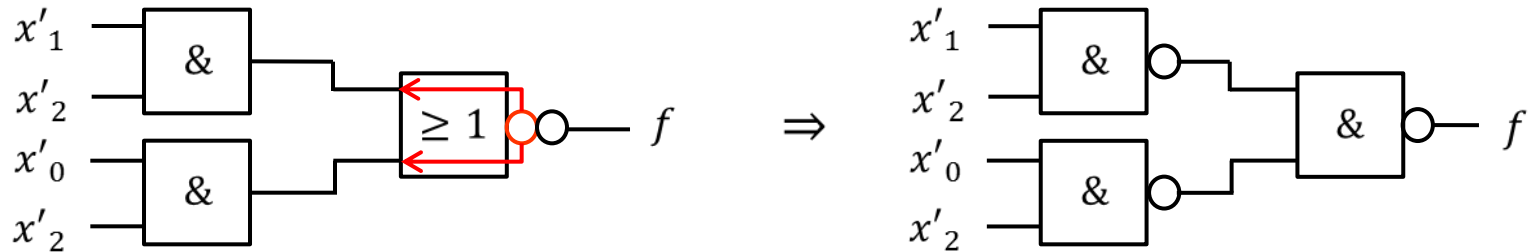


Grindnätstyper från SP-form

AND-OR nät: $f = x_2'x_1' + x_2'x_0'$

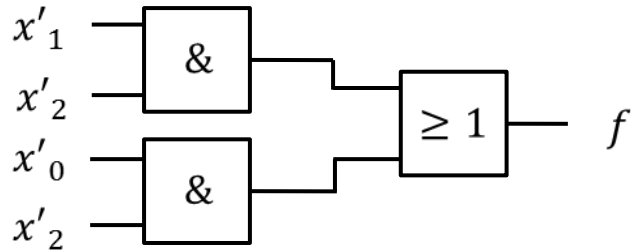


NAND-NAND nät: $f = (x_2'x_1' + x_2'x_0')'' = ((x_2'x_1')'(x_2'x_0')')'$

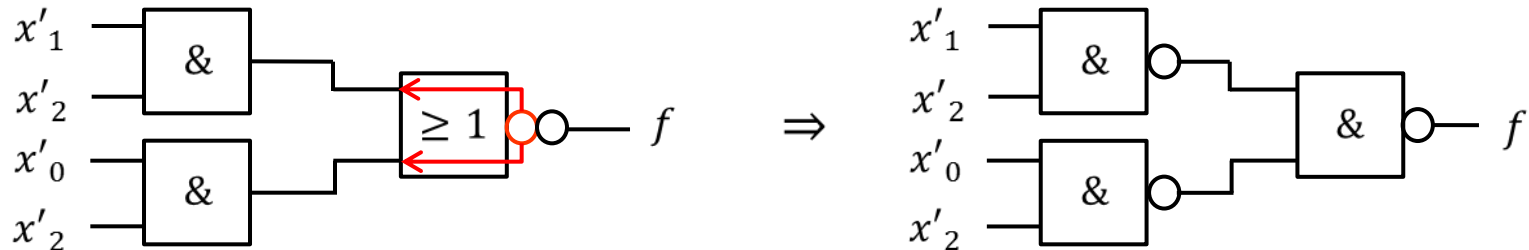


Grindnätstyper från SP-form

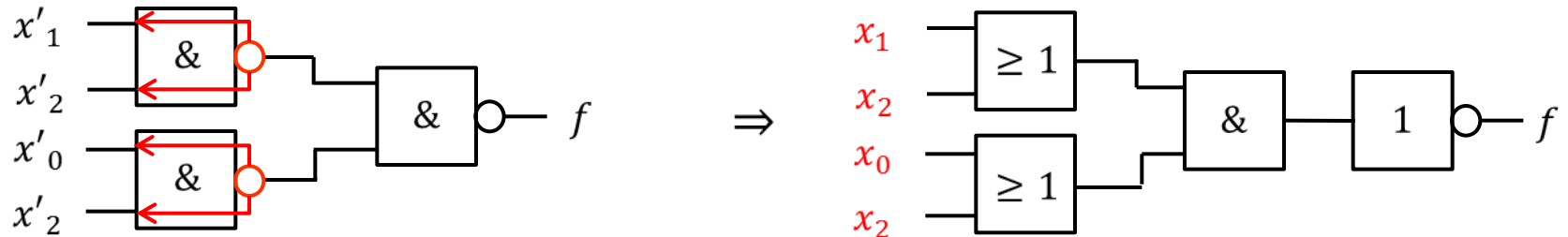
AND-OR nät: $f = x_2'x_1' + x_2'x_0'$



NAND-NAND nät: $f = (x_2'x_1' + x_2'x_0')'' = ((x_2'x_1')'(x_2'x_0')')'$

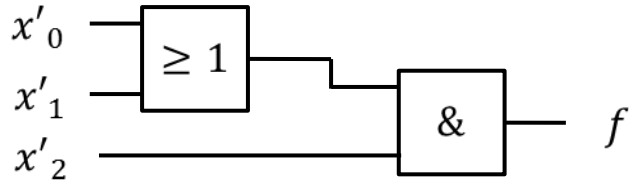


OR-AND-NOT nät: $f = ((x_2'x_1')'(x_2'x_0')')' = ((x_2 + x_1)(x_2 + x_0))'$



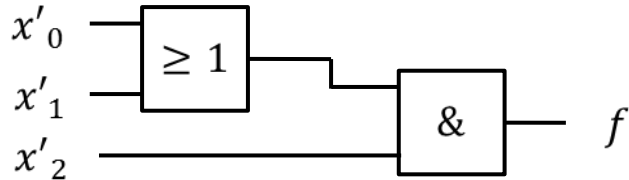
Grindnätstyper för PS-form

OR-AND nät: $f = x'_2(x'_1 + x'_0)$

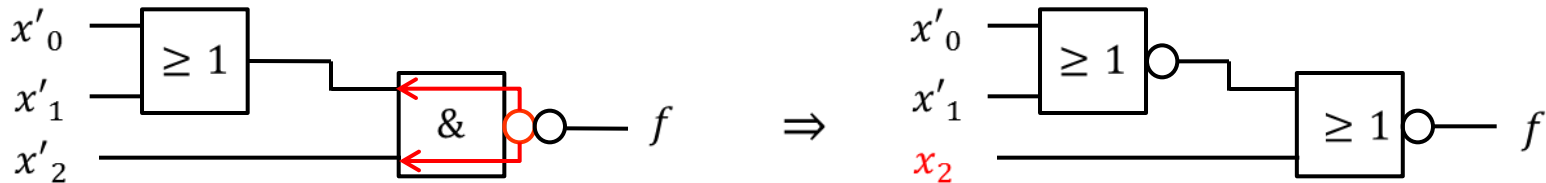


Grindnätstyper för PS-form

OR-AND nät: $f = x_2'(x_1' + x_0')$

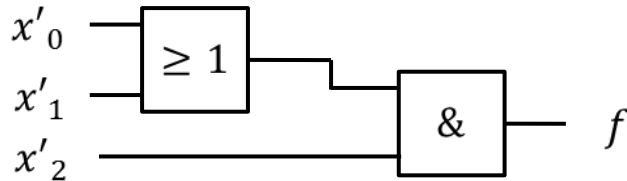


NOR-NOR nät: $f = (x_2'(x_1' + x_0'))'' = (x_2 + (x_1' + x_0')')'$

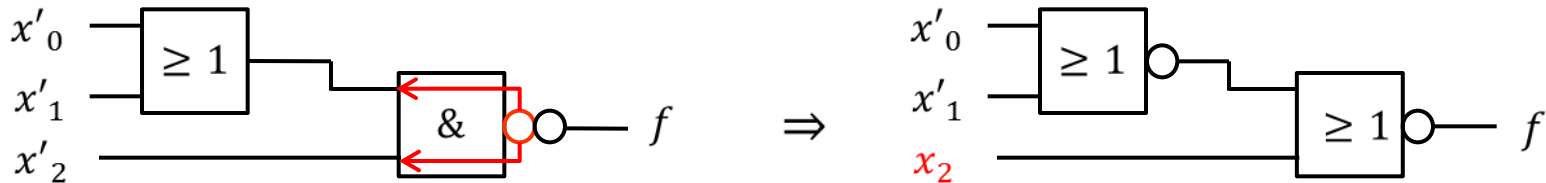


Grindnätstyper för PS-form

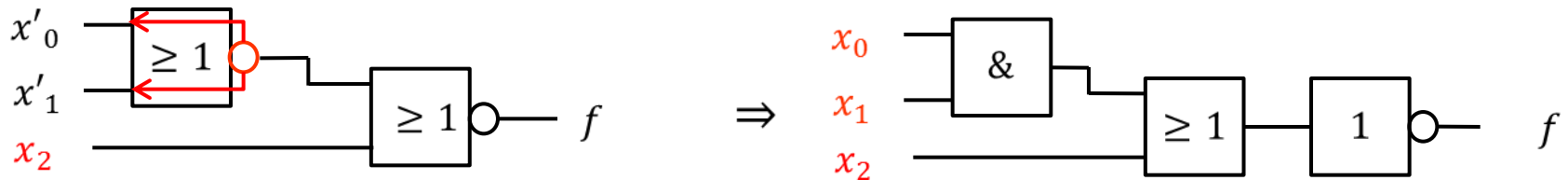
OR-AND nät: $f = x_2'(x_1' + x_0')$



NOR-NOR nät: $f = (x_2'(x_1' + x_0'))'' = (x_2 + (x_1' + x_0'))'$



AND-OR-NOT nät: $f = (x_2 + (x_1' + x_0'))' = (x_2 + x_1x_0)'$



Digitalteknik

Mattias Krysander

www.liu.se