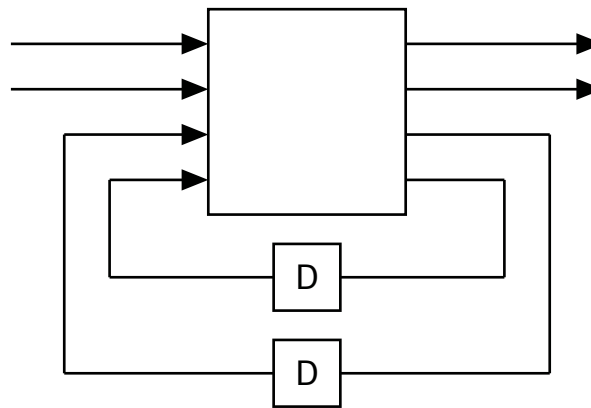
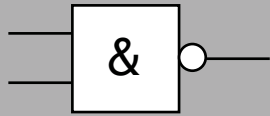


TSEA22 Digitalteknik 2021

Oscar Gustafsson

Ingemar Ragnemalm

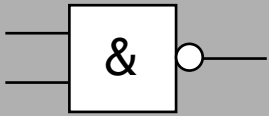




Föreläsning 3. Komb1.

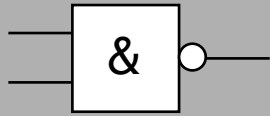
Denna föreläsning:

- Mer Booleska funktioner.
- Kombinationskretsar.
- Förenkling och realisering av Booleska funktioner, *Karnaughdiagram*.
- Ofullständigt specificerade funktioner



Förra föreläsningen:

- Boolesk algebra.
- Enklare axiom och lagar.
- Concensus, De Morgan, absorption



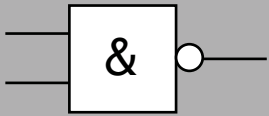
Bli annorlunda i distansläge!

Vanlig fråga: Får vi ha

- Formelsamling
- Miniräknare
- En kopplingsplatta och en drös 7400
- Min favoritundulat (nej jag skojar bara)

på tentan?

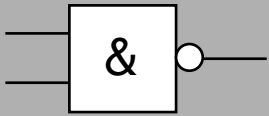
Svaret är nej, inga hjälpmedel! (Jo, papper och penna.)



Lektion 1:

Manipulation av booleska uttryck

Grindar - Boolesk algebra



Absorption

$$x + xy = x$$

$$x(x + y) = x$$

Consensus

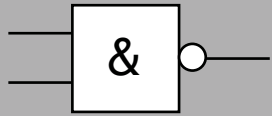
$$xy + x'z = xy + x'z + yz$$

$$(x + y)(x' + z) = (x + y)(x' + z)(y + z)$$

De Morgans lagar

$$(x + y)' = x'y'$$

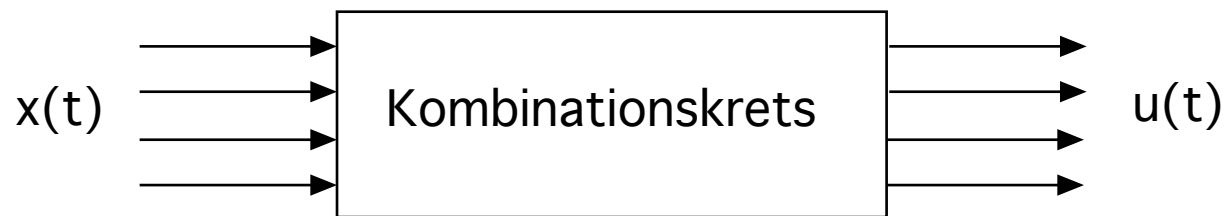
$$(xy)' = x' + y'$$



Kombinationskrets

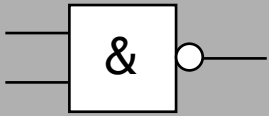
Utsignalen beror enbart av insignalen vid samma tidpunkt.

Kretsar *utan minne*.



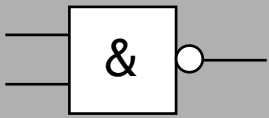
$$u(t) = f(x(t))$$

u och x består av en eller flera booleska variabler.



Några kombinationskretsar

- Multiplexer
- Demultiplexer
- Avkodare
- Komparator

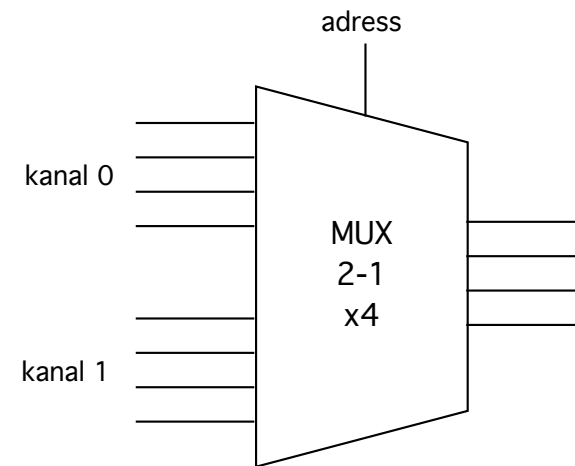
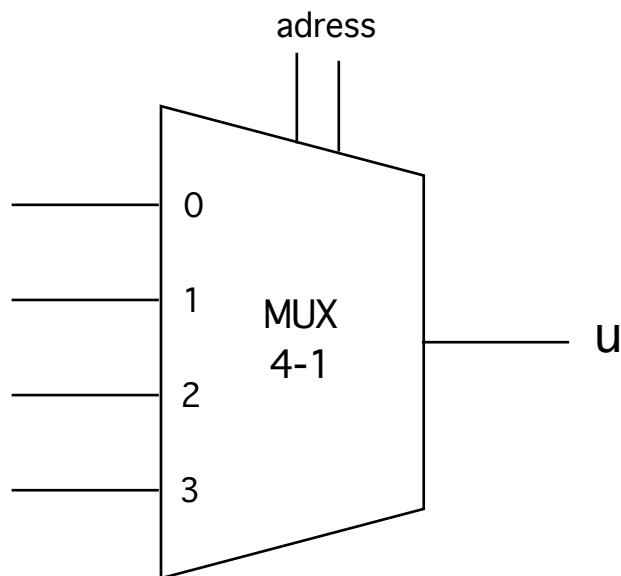


Multiplexer, MUX (dataväljare)

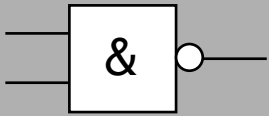
Välj mellan ett antal olika kanaler

2-1, styrs med 1 bit adress

4-1, styrs med 2 bitar adress

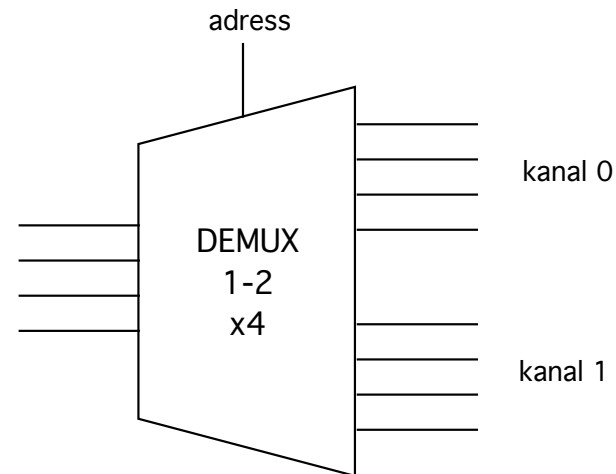
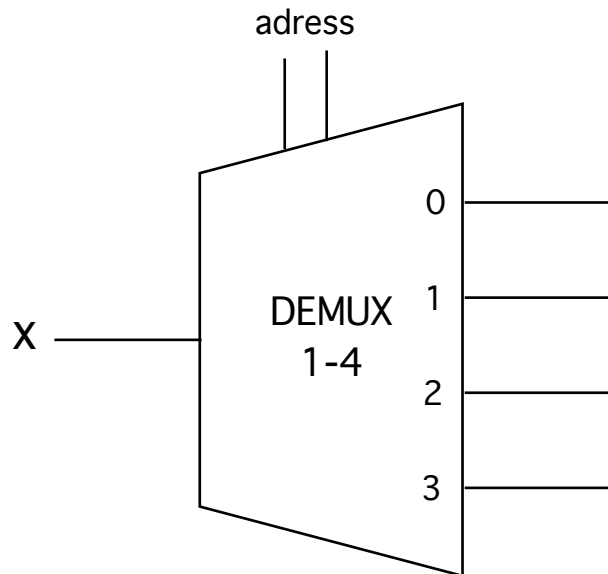


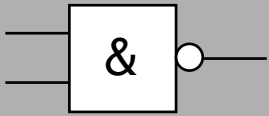
Flerkanalig



Demultiplexer, DEMUX (datafördelare)

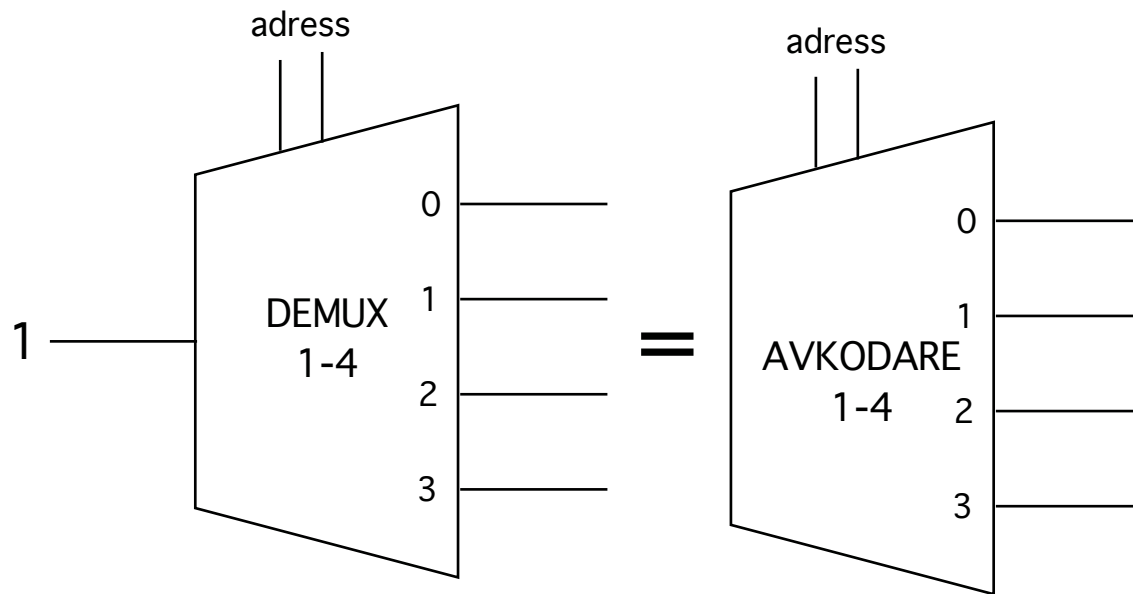
Skicka en signal till en av flera utgångar





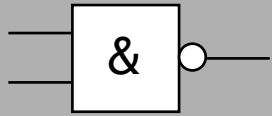
Avkodare

Skickar en 1'a på en av flera utgångar



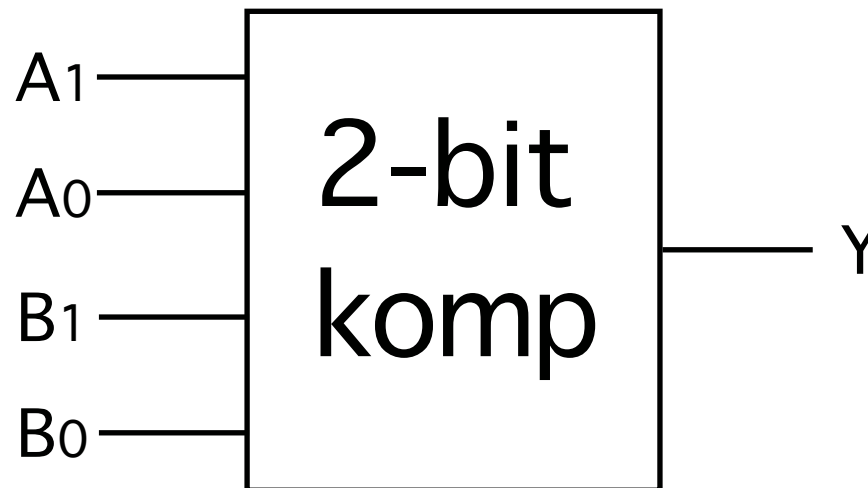
Sanningstabell

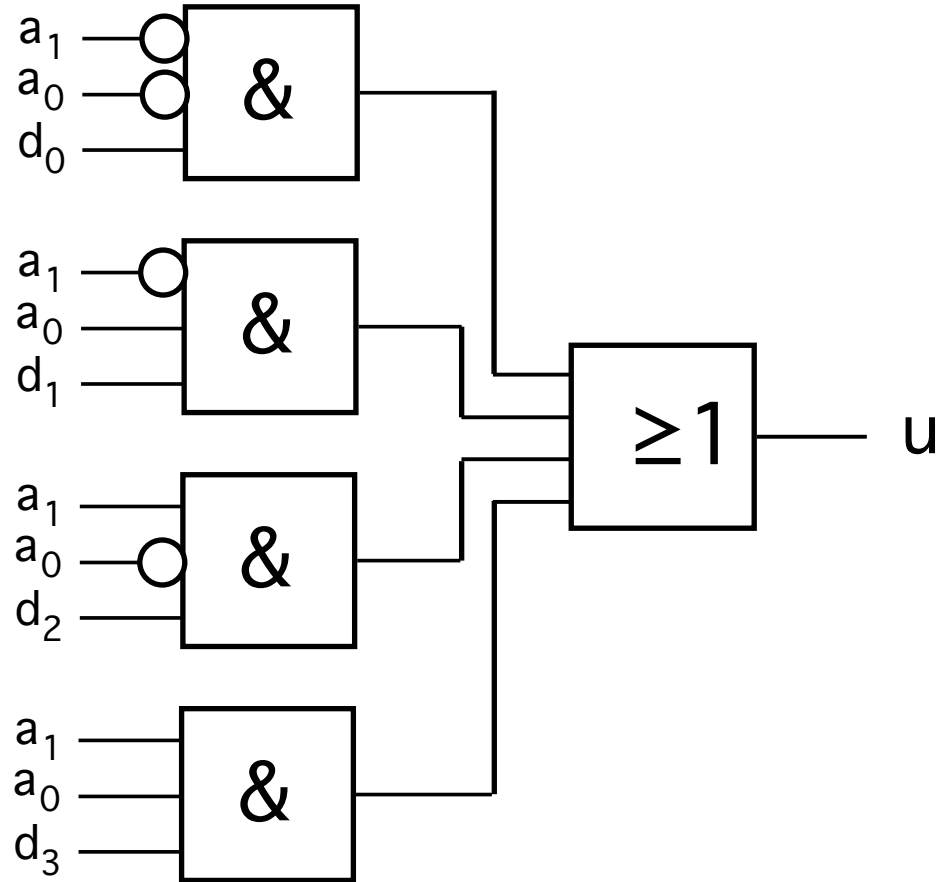
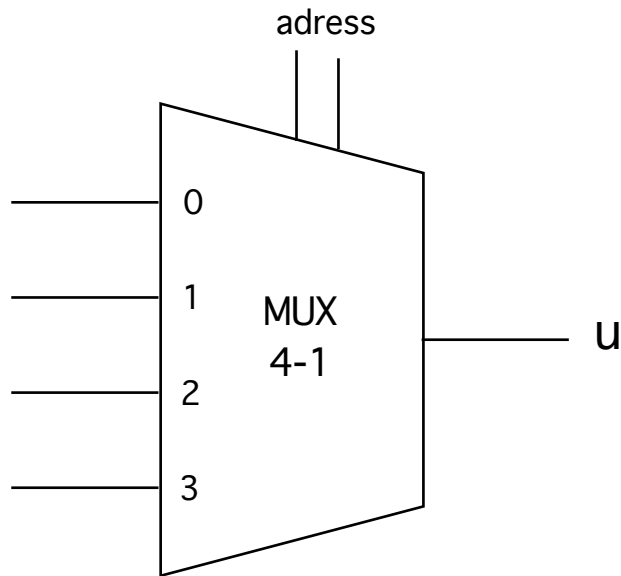
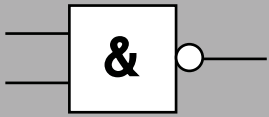
a ₁	a ₀	u ₃	u ₂	u ₁	u ₀
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0



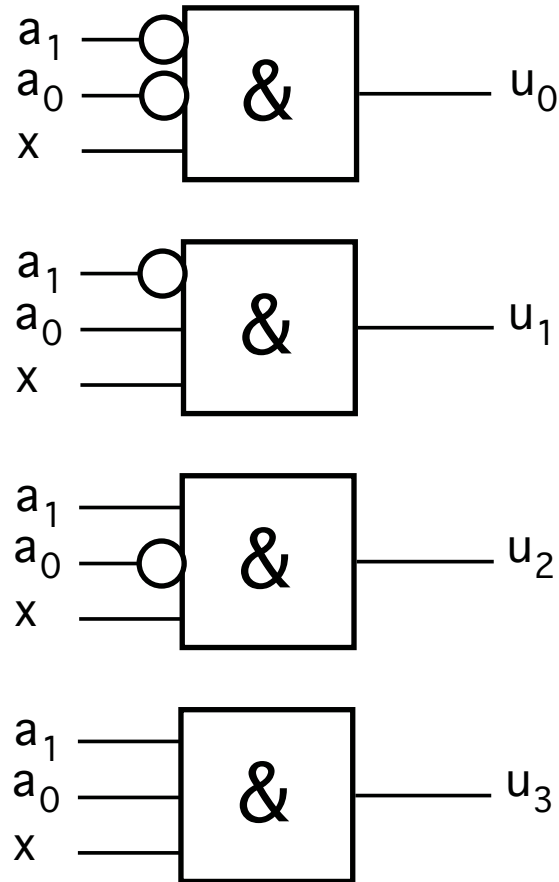
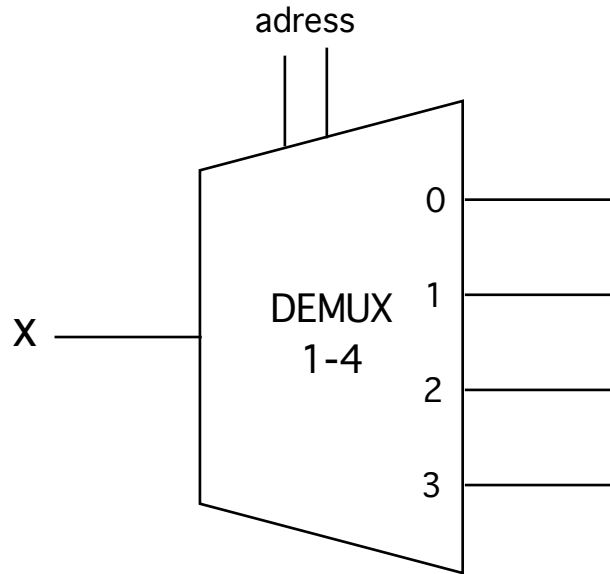
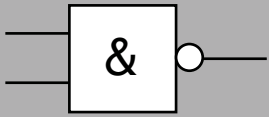
Komparator

Är två signaler eller grupper av signaler (ord) lika?

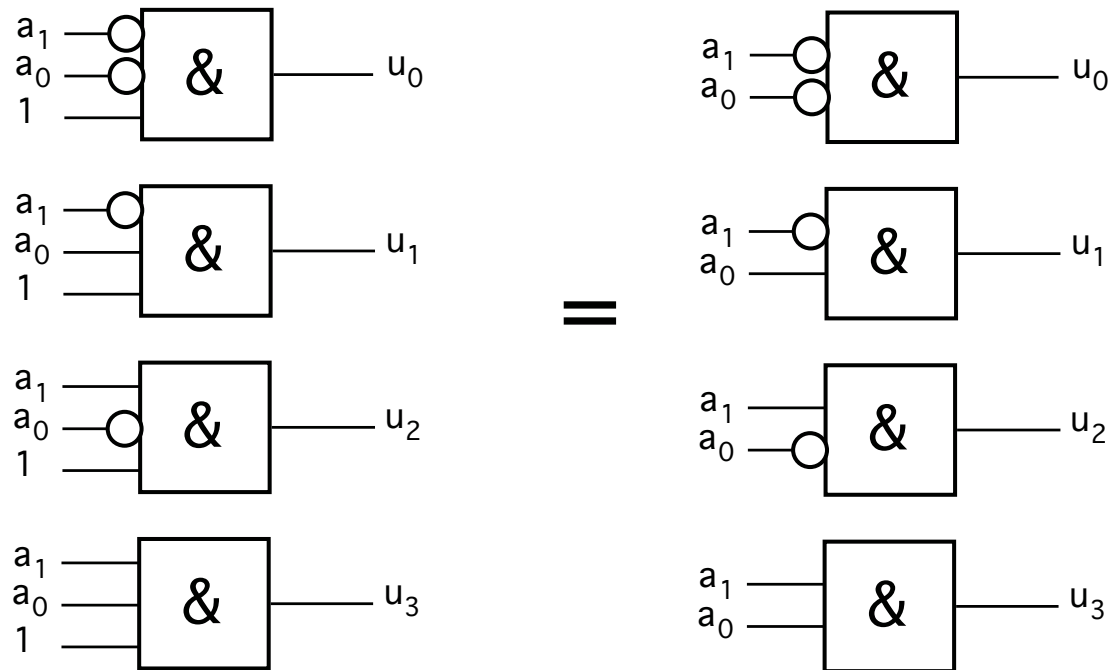
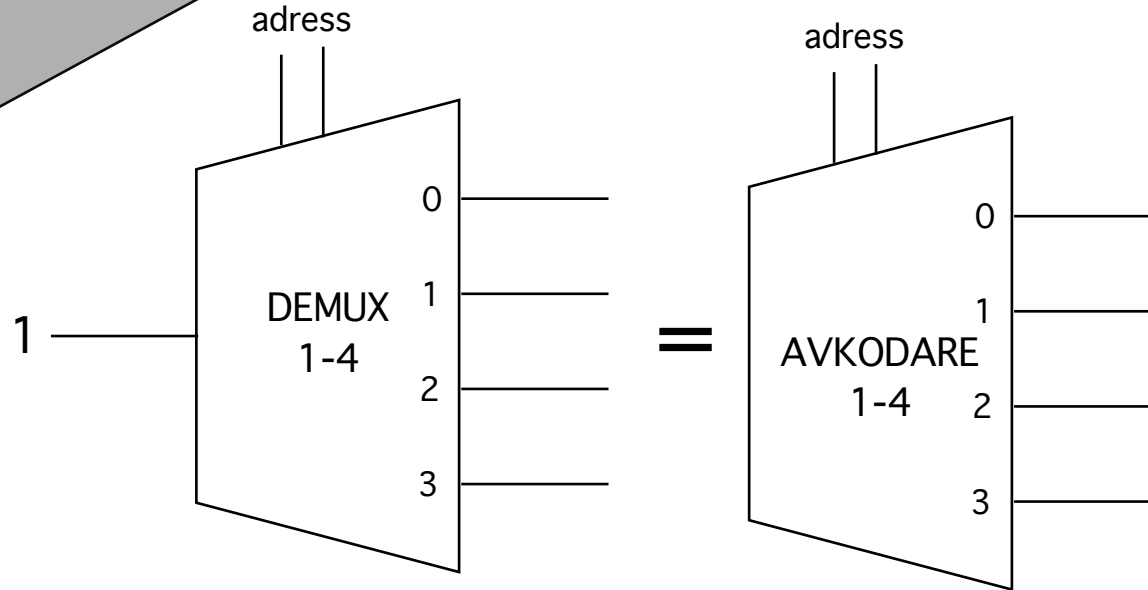
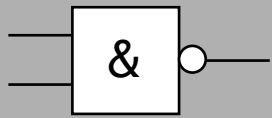




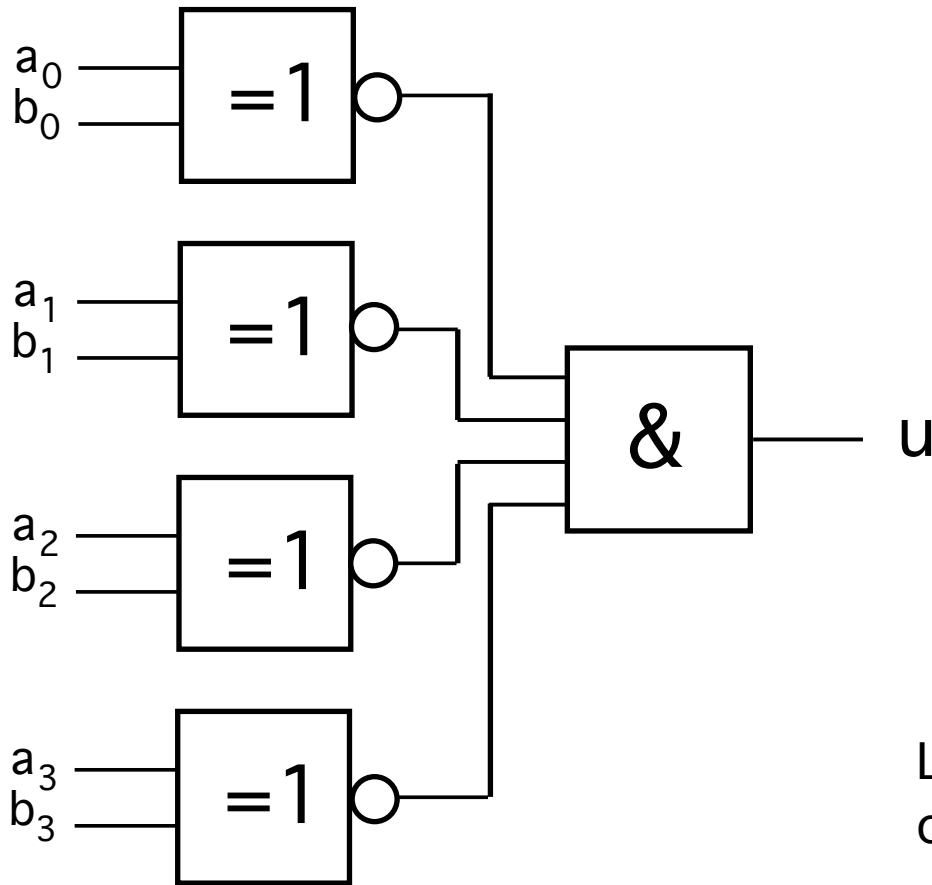
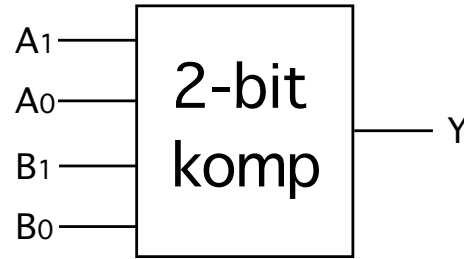
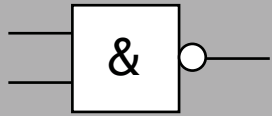
a1 och a0 bildar ett 2-bitars tal, nummer på d-ingång att "skicka vidare"



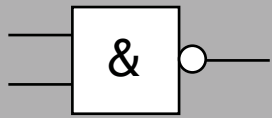
a1 och a0 bildar ett 2-bitars tal, nummer på u-utgång att "skicka ut x till"



a1 och a0 bildar ett 2-bitars tal, nummer på u-utgång att "skicka ut 1 till"

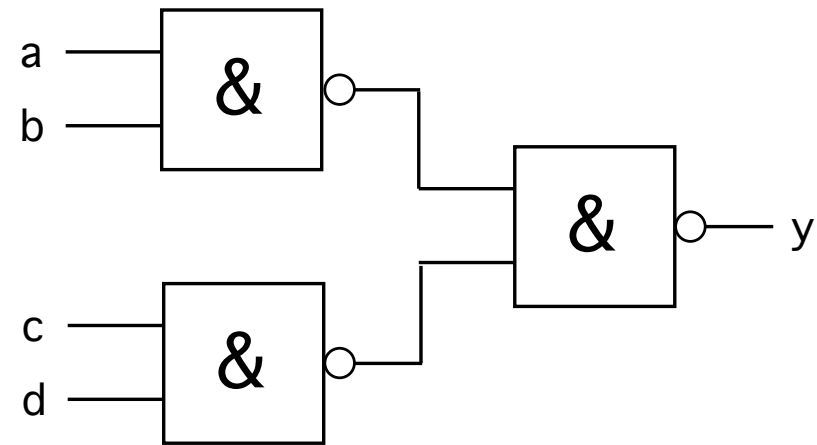
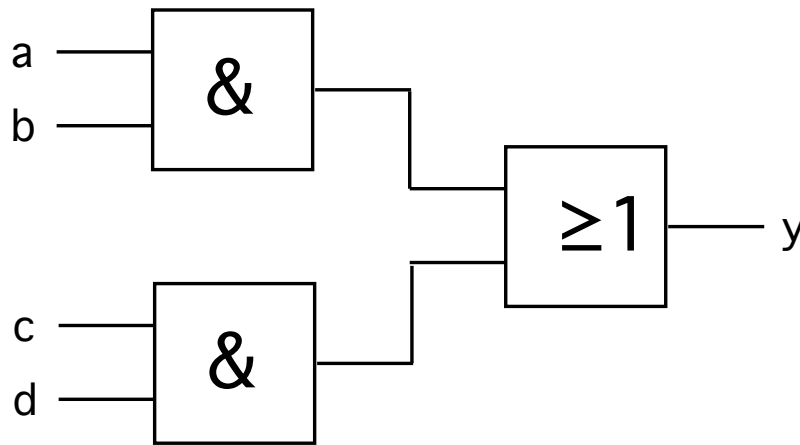


Likhet på alla par av bitar för orden/bitgrupperna a och b

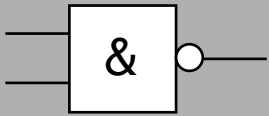


Från uttryck till grindar

$$y = ab + cd = ((ab)' \cdot (cd)')'$$



Det finns alltid mer än en lösning!



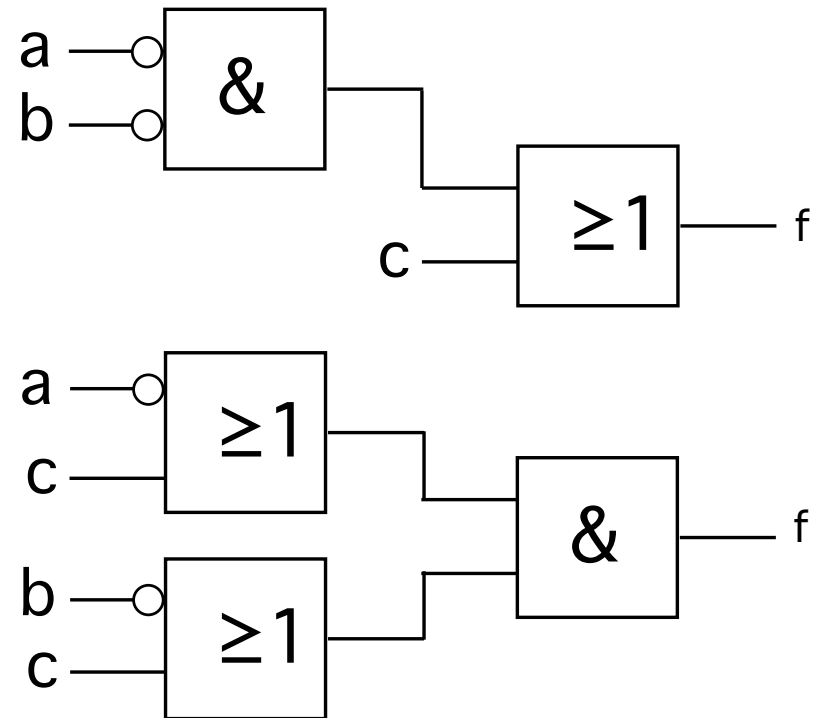
Sanningstabell

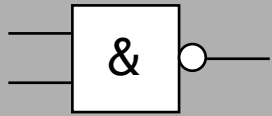
a	b	c	f
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Boolesk funktion

$$f = c + a'b' = (b'+c)(a'+c)$$

Realisering





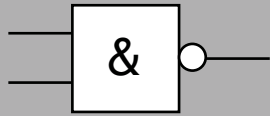
Boolesk funktion

Funktion med Booleska värden in och ut.

Definieras av sanningstabellen.

A	B	Y
0	0	y ₀
0	1	y ₁
1	0	y ₂
1	1	y ₃

Funktion av 2 variabler: 4 olika kombinationer = $2^2 = 4$ = 16 olika funktioner är möjliga.



Grundformer för Booleska funktioner: SP- och PS-form

SP = Summa av Produkter

$$y = ab + cd$$

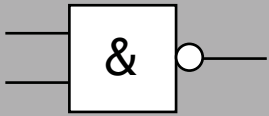
Union av mindre delar

PS = Produkt av Summor

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

Snitt av större delar

Alla funktioner kan uttryckas i båda formerna.

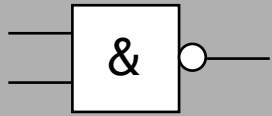


Exemplet tidigare:

SP = Summa av Produkter
 $c + a'b'$

PS = Produkt av Summor
 $(b'+c)(a'+c)$

Grundnätstyper för SP och PS

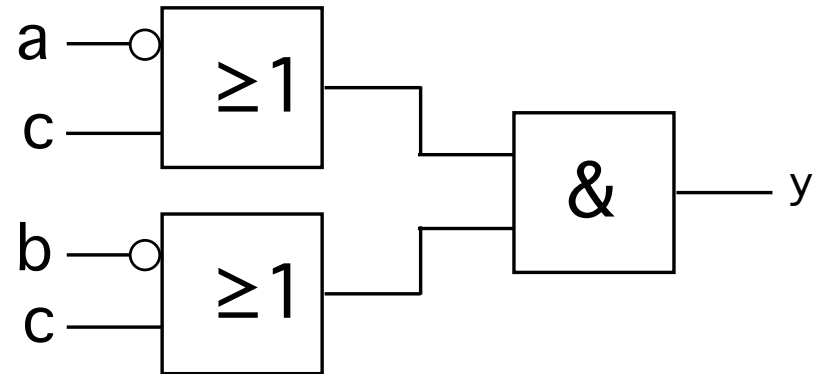
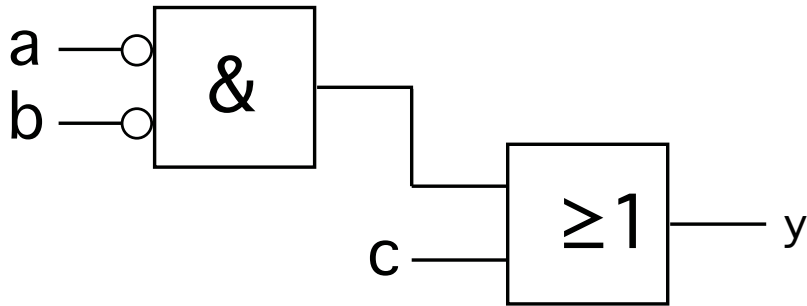


SP
 $c + a'b'$

PS
 $(b'+c)(a'+c)$

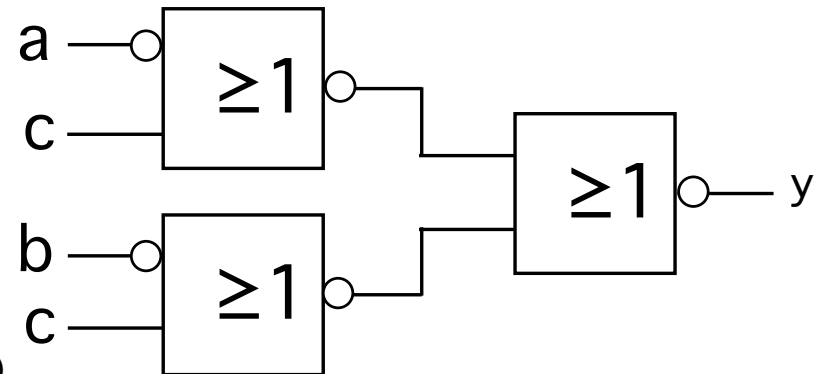
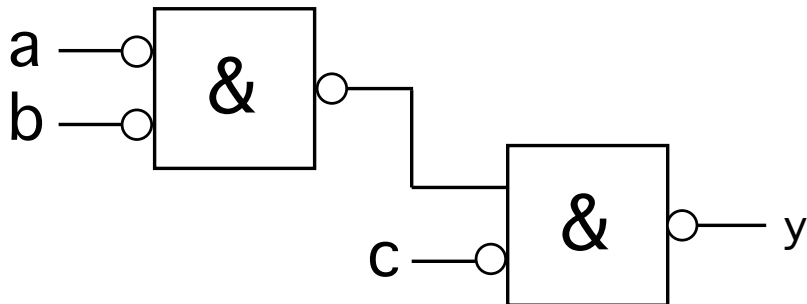
AND-OR

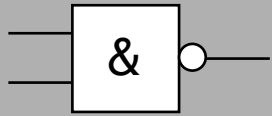
OR-AND



NAND-NAND

NOR-NOR



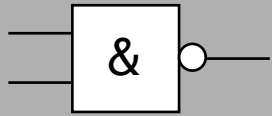


mintermer och maxtermer

minterm beskriver en etta

maxterm beskriver allt utanför en nolla

i	a b c	f	minterm m_i	maxterm M_i
0	0 0 0	1	$a'b'c'$	$a+b+c$
1	0 0 1	1	$a'b'c$	$a+b+c'$
2	0 1 0	0	$a'bc'$	$a+b'+c$
3	0 1 1	1	$a'bc$	$a+b'+c'$
4	1 0 0	0	$ab'c'$	$a'+b+c$
5	1 0 1	1	$ab'c$	$a'+b+c'$
6	1 1 0	0	abc'	$a'+b'+c$
7	1 1 1	1	abc	$a'+b'+c'$

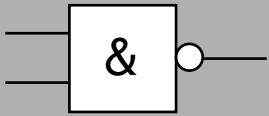


mintermer = tag med en etta

$$\begin{aligned} f(a,b,c) &= \\ &= a'b'c' + a'b'c + a'bc + ab'c + abc = \\ &= c + a'b' \end{aligned}$$

maxtermer = tag bort en nolla

$$\begin{aligned} f(a,b,c) &= \\ &= (a+b'+c)(a'+b+c)(a'+b'+c) = \\ &= (b'+c)(a'+c) \end{aligned}$$



Normalform: Σ och Π

Uttryck för SP- och PS-form

Ekvivalent med sanningtabell

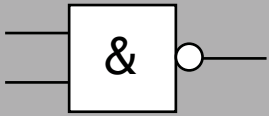
SP-normalform listar ettor.

$$f(a,b,c) = \Sigma(0,1,3,5,7)$$

PS-normalform listar nollor

$$f(a,b,c) = \Pi(2,4,6)$$

i	a b c	f
0	0 0 0	1
1	0 0 1	1
2	0 1 0	0
3	0 1 1	1
4	1 0 0	0
5	1 0 1	1
6	1 1 0	0
7	1 1 1	1

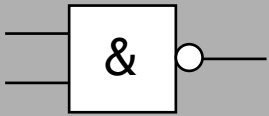


Karnaughdiagram

Ett verktyg för optimering av Booleska uttryck och därmed digitala kretsar.

Förenkling av Booleska funktioner görs visuellt och enkel enligt givna regler.

Effektivt upp till ca 6 variabler. Bäst på 3 eller 4.



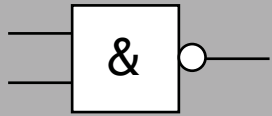
Karnaughdiagram för tre variabler

		bc			
		00	01	11	10
a	0	0	1	3	2
	1	4	5	7	6

Värden på bc i
omkastad ordning!

Här har vi bara en
variabel och skriver
båda utfallen.

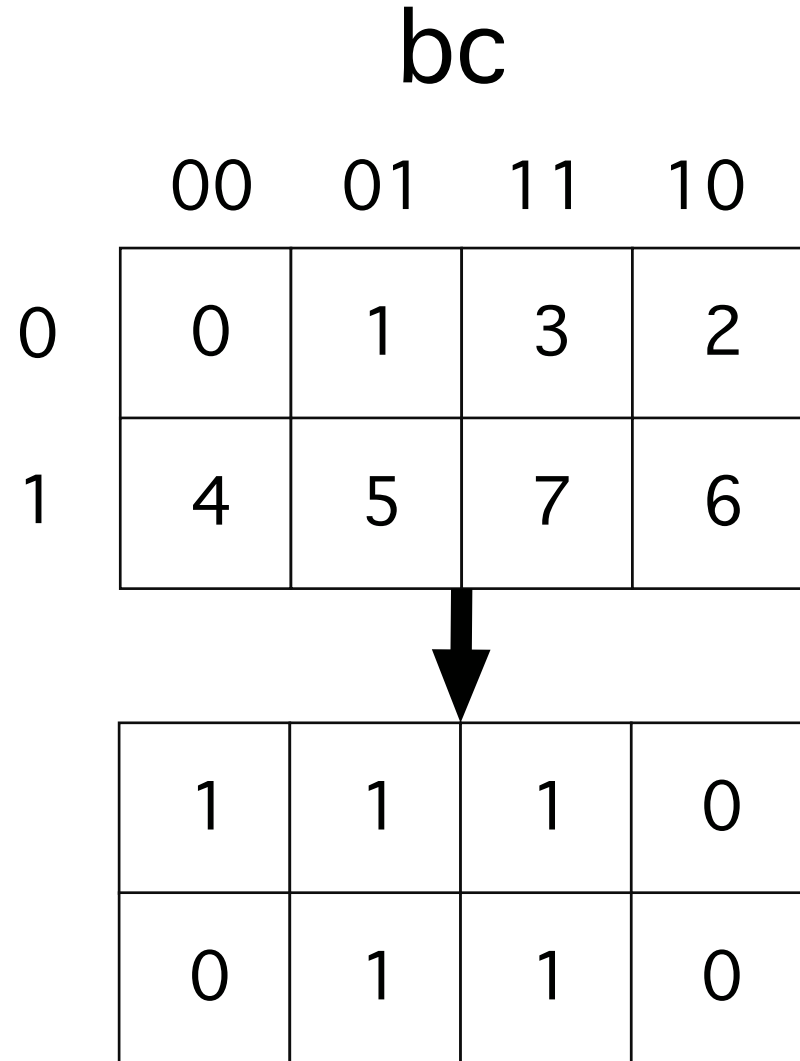
Inte graykod men
ett liknande trick:
Bara en siffra
ändras i taget!

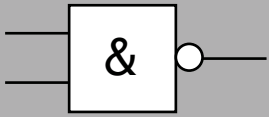


Sanningstabell till Karnaughdiagram

i	a b c	f
0	0 0 0	1
1	0 0 1	1
2	0 1 0	0
3	0 1 1	1
4	1 0 0	0
5	1 0 1	1
6	1 1 0	0
7	1 1 1	1

a



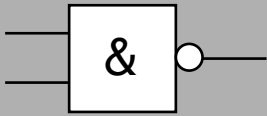


Optimering i Karnaughdiagram

Närliggande ettor = kan slås ihop!

		bc			
		00	01	11	10
a	0	1	1	1	0
	1	0	1	1	0

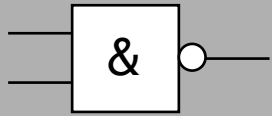
$$ab'c + abc = (b' + b)ac = ac$$



Metod

- Ringa alla ettor, inte nollor
- Så stora ringar som möjligt. Stora ringar = enklare uttryck.
- Alla ringar måste ha höjd/bredd 1, 2 eller 4. Aldrig 3! (2-potenser.)

Detta ger *SP-form*



a

a = 1 för ringen → 1

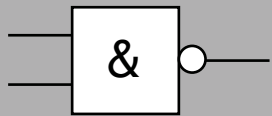
		bc			
		00	01	11	10
a	0	1	1	1	0
	1	0	1	1	0

c = 1 över hela ringen

Ringar till uttryck

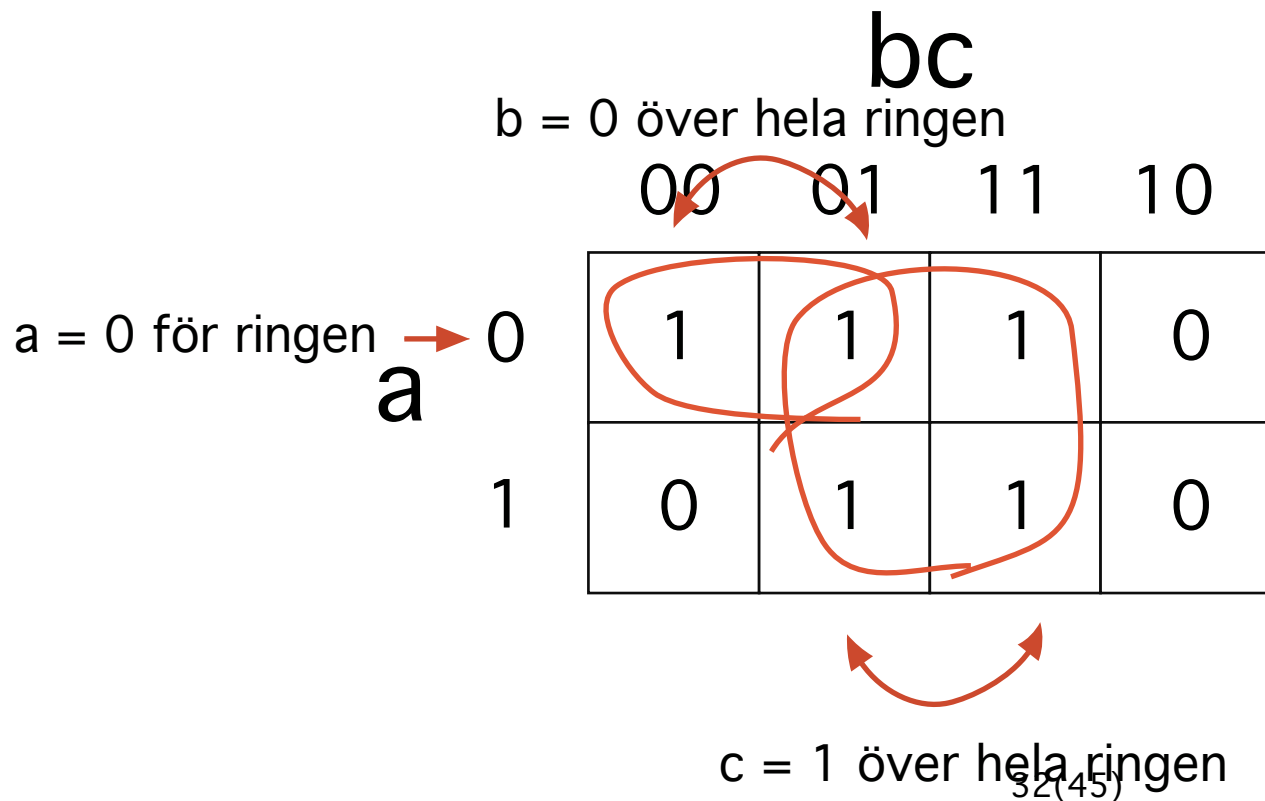
- En ring motvarar de variabler som *inte* varierar inom ringen!
- Stor ring = färre variabler som inte varierar!
- Liten ring = fler variabler som inte varierar!

ringen = ac

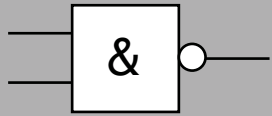


Komplett lösning

- Alla ettor
- Så stora ettor som möjligt



Funktion:
 $f = a'b' + c$



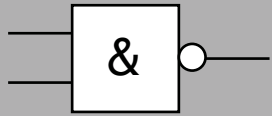
4 variabler

cd

ab

	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

Ordningstal =
binära talet
abcd



5 variabler

de

a = 0 00 01 11 10

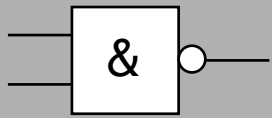
bc	00	0	1	3	2
	01	4	5	7	6
	11	12	13	15	14
	10	8	9	11	10

de

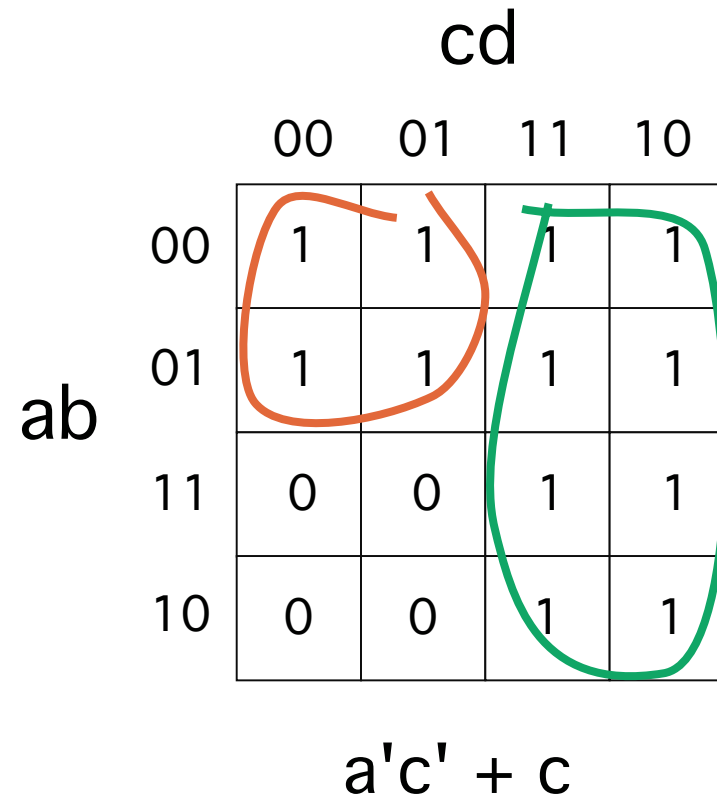
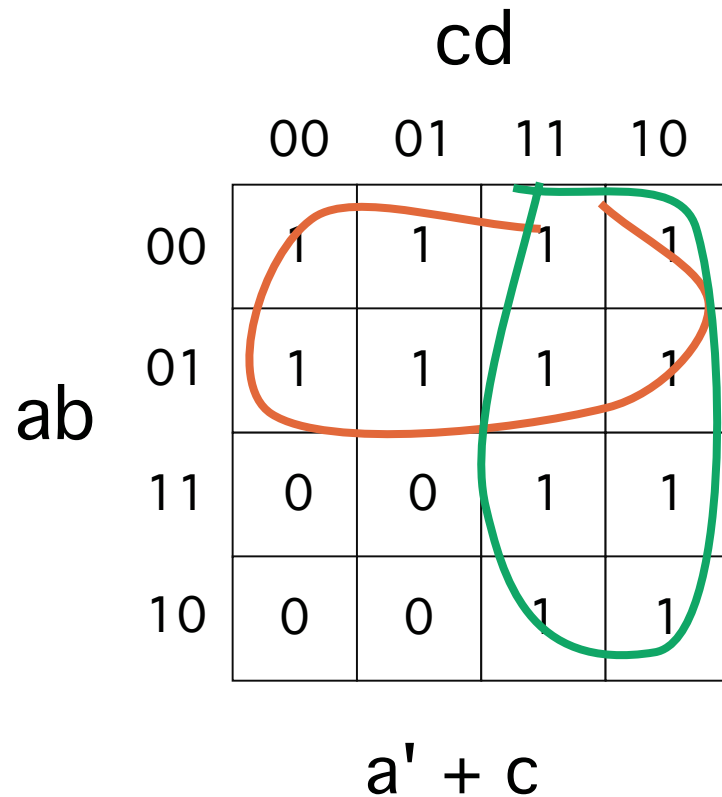
a = 1 00 01 11 10

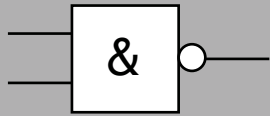
bc	00	16	17	19	18
	01	20	21	23	22
	11	28	29	31	30
	10	24	25	27	26

Ringa "på höjden" genom de två lagren



"Jamen jag har ju redan ringat in den!"



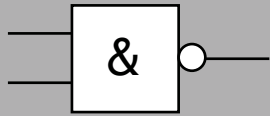


Blanda storlekar efter behov

		cd			
		00	01	11	10
ab	00	0	0	0	1
	01	0	1	1	0
	11	1	1	1	1
	10	1	0	0	0

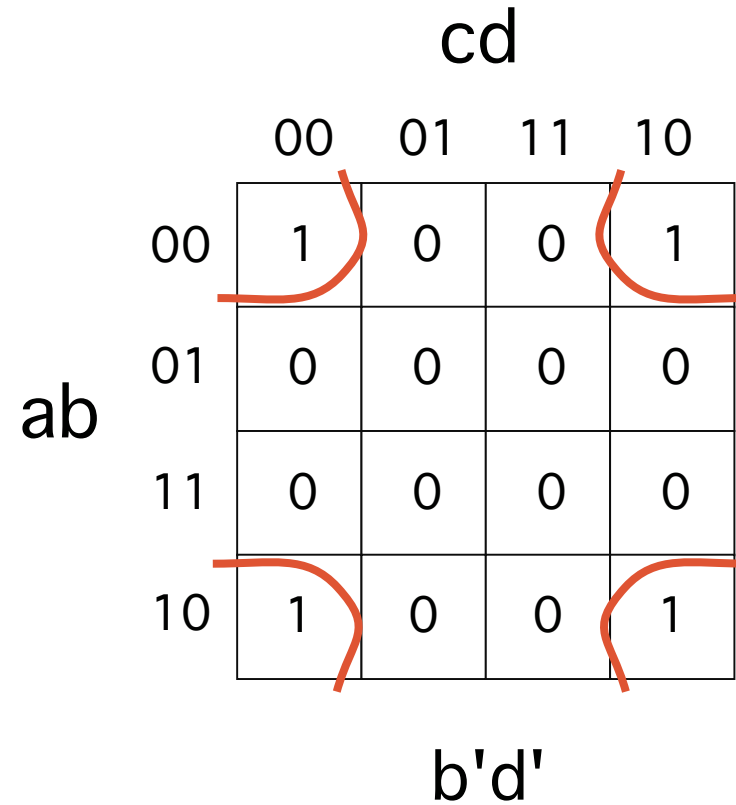
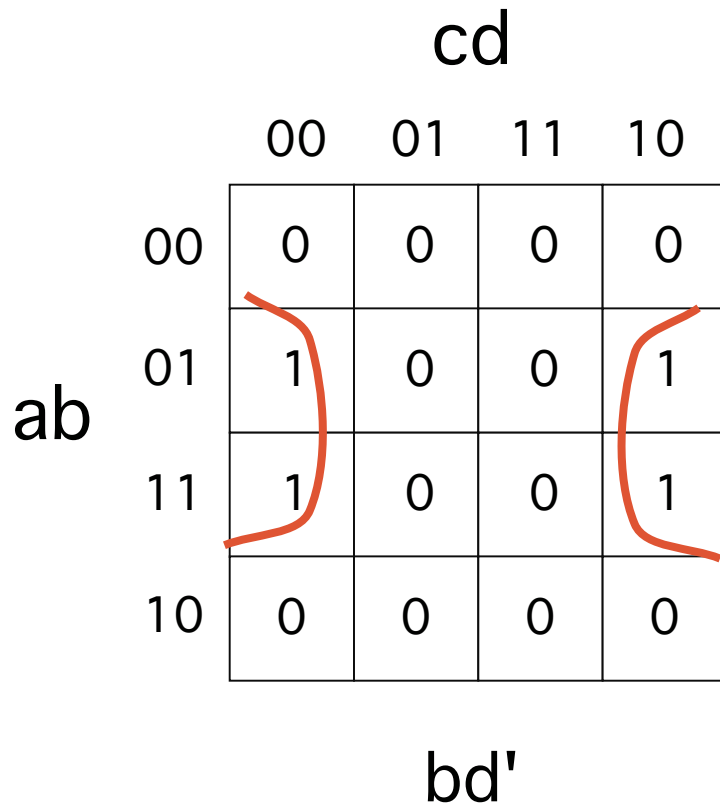
$$ab + c'd'a + bd + a'b'cd'$$

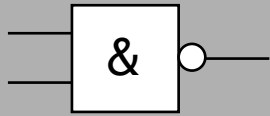
så länge de har höjd/bredd 1, 2, 4!



Wrap-around

Du får alltid ringa runt kanten!



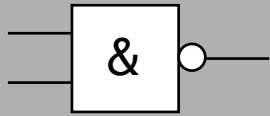


Primimplikator

Produkterm från en *maximal ring!*

Väsentlig primimplikator

En ensam etta som inte täcks av någon annan. Delar av uttrycket där vi inte har något val!



SP-form: Ringa ettor.

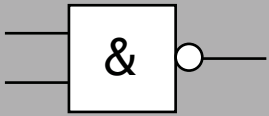
Ger summa av produkter.

PS-form: Ringa nollor!

Ger summa av produkter för alla nollor! Invertera,

De Morgan: Produkt av summor!

Alt: Finn "icke-ringen" som eller-funktion direkt.



Concensus

bc

	00	01	11	10
0	0	0	1	1
1	0	1	1	0

$$ac + a'b = ac + a'b + bc$$

Absorption

ab

00	01	11	10
0	0	1	1

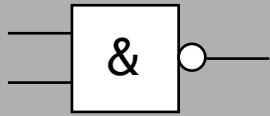
$$a + ab = a$$

De Morgan

ab

00	01	11	10
0	1	1	1

$$a + b = (a'b')'$$

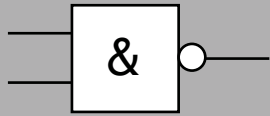


Ofullständigt specificerade funktioner "Don't care"



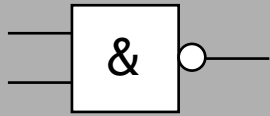
Peter Gibbons: "The thing is, Bob, it's not that I'm lazy, it's that I just don't care."

(Office Space)



KARNAUGH MED STRECK!

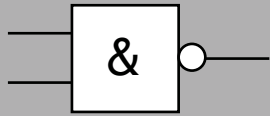
	00	01	11	10
00	0	0	1	-
01	1	0	-	1
11	1	-	1	1
10	-	0	-	-



"Don't care"

Indata som inte skall inträffa. Utdata är godtycklig, odefinierad.

Regel: Största möjliga ringar. Ringa "-" om det ger en förbättring.



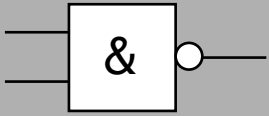
Σ och Π med don't care

SP-normalform:

$$f(a,b,c) = \Sigma(3,4,6,12,14,15) + d(2,7,8,10,11,13)$$

PS-normalform:

$$f(a,b,c) = \Pi(0,1,5,9) + d(2,7,8,10,11,13)$$



NÄSTA FÖRELÄSNING

Labb 1

Adderare

Carryaccelerator