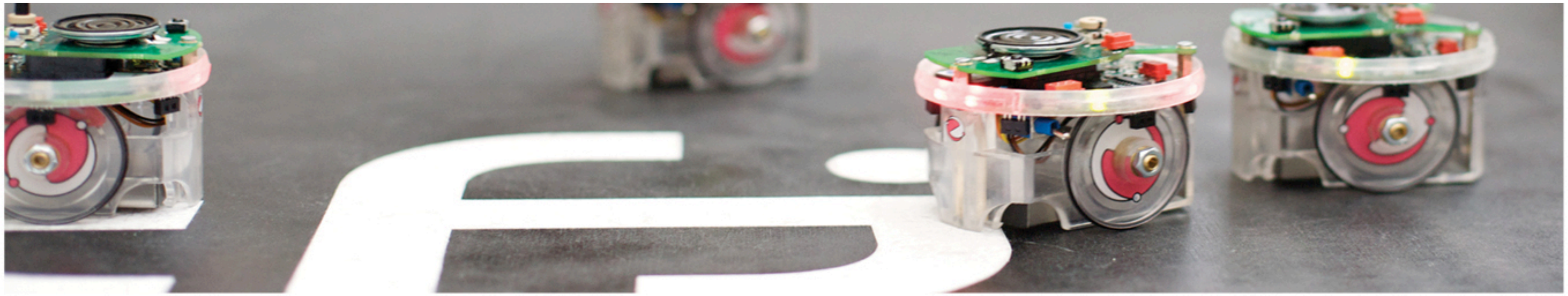
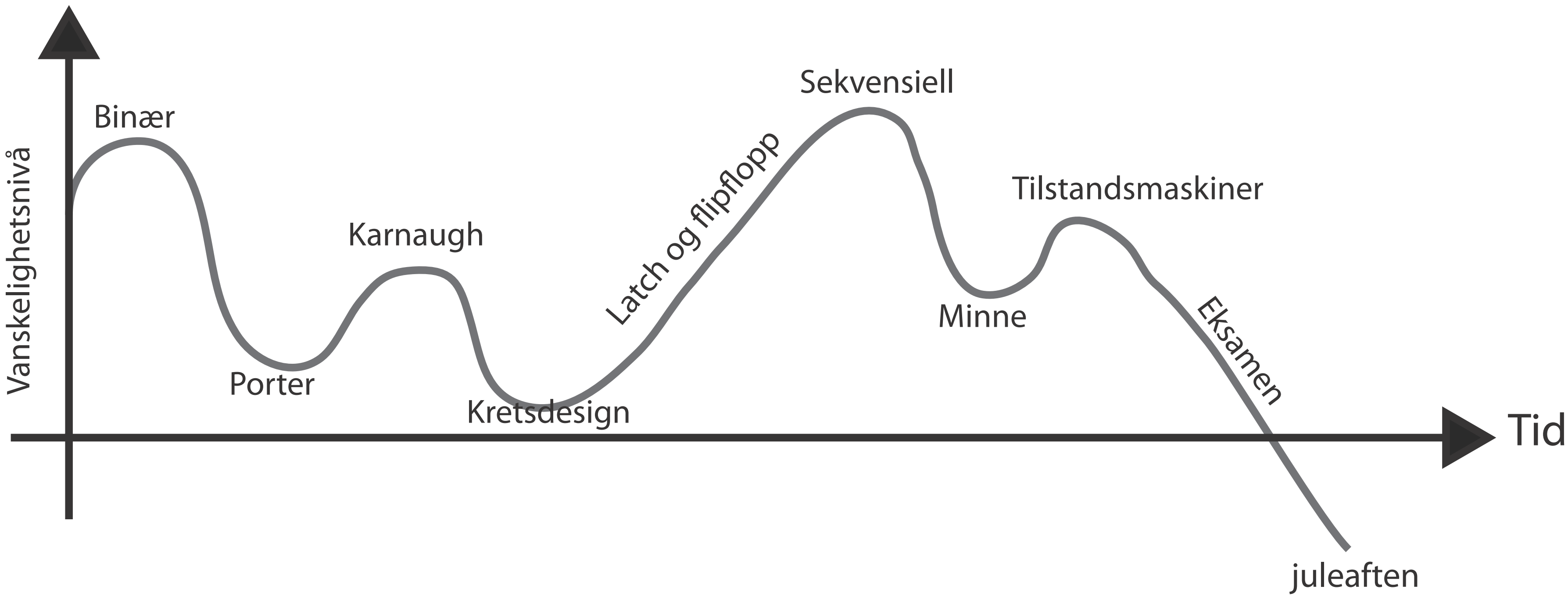


UiO : Institutt for informatikk
Det matematisk-naturvitenskapelige fakultet

INF1400

Karnaughdiagram

Hvor er vi?


Vanskelighetsnivå
Binær
Porter
Karnaugh
Kretsdesign
Latch og flipflopp
Sekvensiell
Minne
Tilstandsmaskiner
Eksamen
juleaften
Tid

Hva lærte vi forrige uke?

- Tallsystemer
- Boolsk algebra
  - Regneregler
- Porter
  - AND – OR – NOT – NAND – NOR

Test deg selv

Gitt funksjonen: F(x,y,z) = S(0,1,2,4,5,6)

(a) Sett opp sannhetsverditabellen

(b) Forenkle uttrykket ved regning

(c) Forenkle uttrykket ved Karnaughdiagram

(d) Implementer det forenklede uttrykket på portnivå

(e) Implementer det forenklede uttrykket med bare 2-inputs NAND

Hovedpunkter

- Karnaughdiagram
  - Diagram med 2-4 variable
  - Don’t care tilstander
  - Produkt av sum (leser ut ”0”ere)

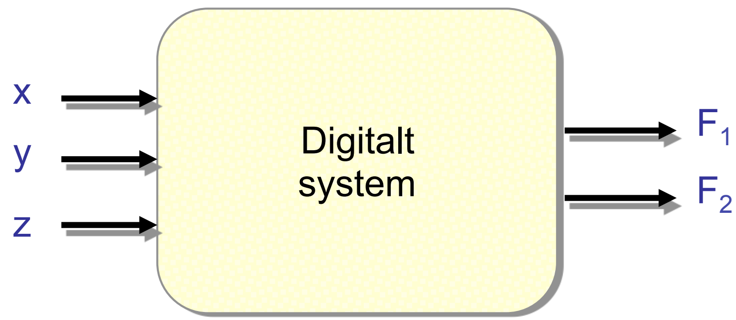
- XOR implementasjon

- NAND implementasjon ved DeMorgan

- LogiSim introduksjon

Bakgrunn, typisk problemstilling

Anta at vi designer et digitalt system med, f.eks, 3 innganger og 2 utganger



Sannhetstabell, eksempel

Starter med å sette opp en sannhetstabell

Setter så opp et uttrykk for hver utgang F_1 , F_2

Ser på utgangens "1"ere

$F_1 =$ x´y´z´+ x´y´z + xyz´+ xyz

$F_2 =$ x´y´z + xy´z + xyz

Innganger			Utganger	
x	y	z	F_1	F_2
0	0	0	1	0
0	0	1	1	1
0	1	0	0	0
0	1	1	0	0
1	0	0	0	0
1	0	1	0	1
1	1	0	1	0
1	1	1	1	1

Karnaughdiagram

Grafisk metode for forenkling av Boolske uttrykk

- Uttrykket må være representert ved sum av mintermer (m_x). Disse leser vi direkte ut av sannhetstabellen
- Metoden egner seg for funksjoner med 2-4(5) variable

Eksempel, 2 variable:

$$F = m_1 + m_3 = a'b + ab$$

Eksempel, 4 variable:

$$F = m_0 + m_1 + m_{15} = A'B'C'D' + A'B'C'D + ABCD$$

Prosedyre, 2 variable

Setter inn mintermene i diagram

- Eksempel: generell funksjon - 2 variable

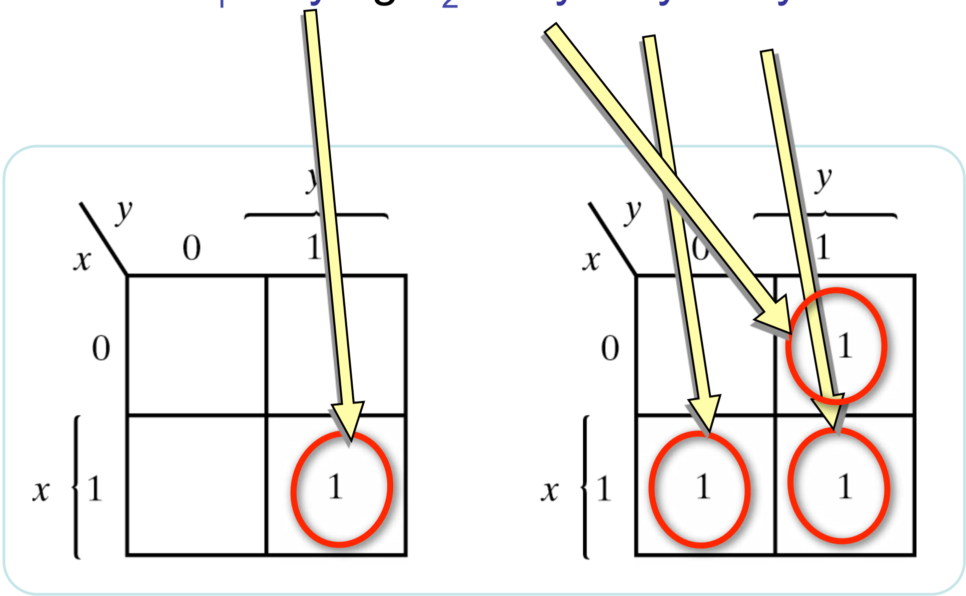
$$F = m_0 + m_1 + m_2 + m_3$$

m_0	m_1
m_2	m_3

$x \backslash y$	0	1 (y)
0	$x'y'$	$x'y$
(x) 1	xy'	xy

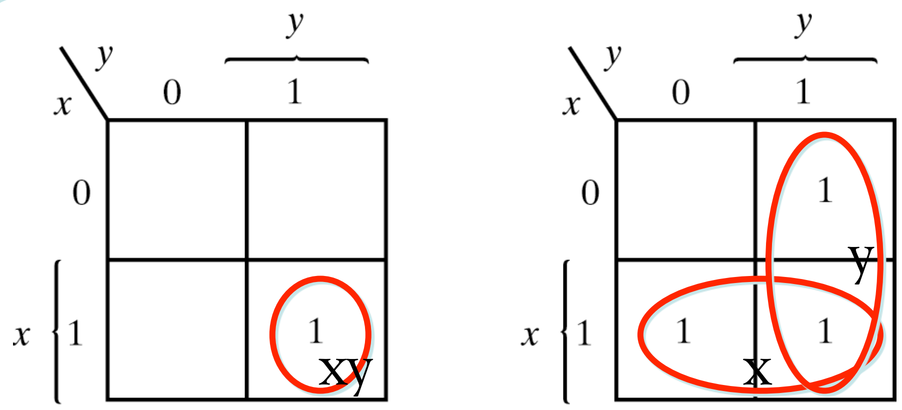
Prosedyre, 2 variable

$F_1 = xy$ og $F_2 = x'y + xy' + xy$



Prosedyre, utlesning

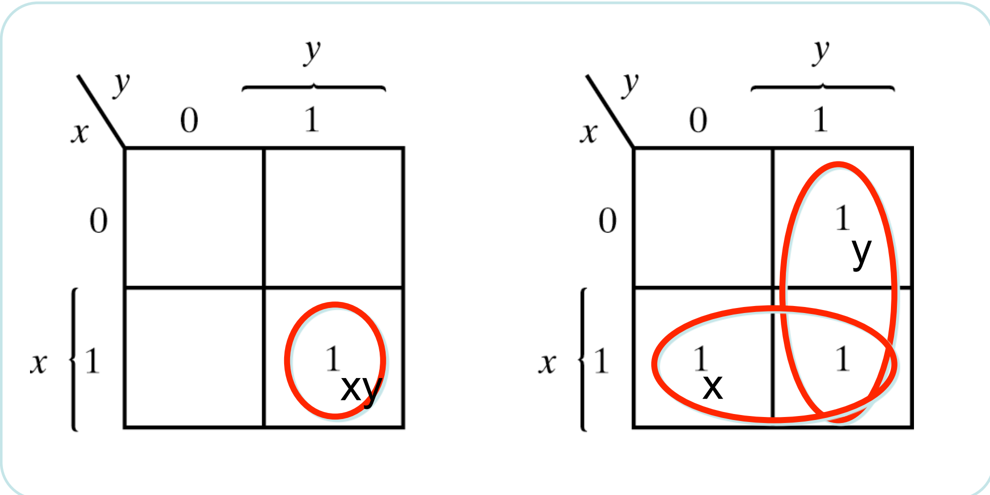
- Grupperer naboruter som inneholder ”1” slik at vi får sammenhengende rektangler, Velg så store grupper som mulig. Antall element må være en potens av 2
- Representerer gruppene ved de variablene i gruppen som ikke varierer



Prosedyre, utlesning

Funksjonene som diagrammene beskriver er nå gitt av summen av uttrykkene som representerer hver gruppe

$F_1 = xy$ $F_2 = x + y$



Karnaugh - 3 variable

Plassering av mintermer for 3-variable funksjoner:

- Mintermene plasseres slik at kun 1 variabel varierer i mellom hver vannrette/loddrette naborute

m_0	m_1	m_3	m_2
m_4	m_5	m_7	m_6

$x \backslash yz$	00	01	11	10
0	$x'y'z'$	$x'y'z$	$x'yz$	$x'yz'$
x 1	$xy'z'$	$xy'z$	xyz	xyz'

Karnaugh - 4 variable

Plassering av mintermer for 4-variable funksjoner

- Mintermene plasseres slik at kun 1 variabel varierer i mellom hver vannrette/loddrette naborute

m_0	m_1	m_3	m_2
m_4	m_5	m_7	m_6
m_{12}	m_{13}	m_{15}	m_{14}
m_8	m_9	m_{11}	m_{10}

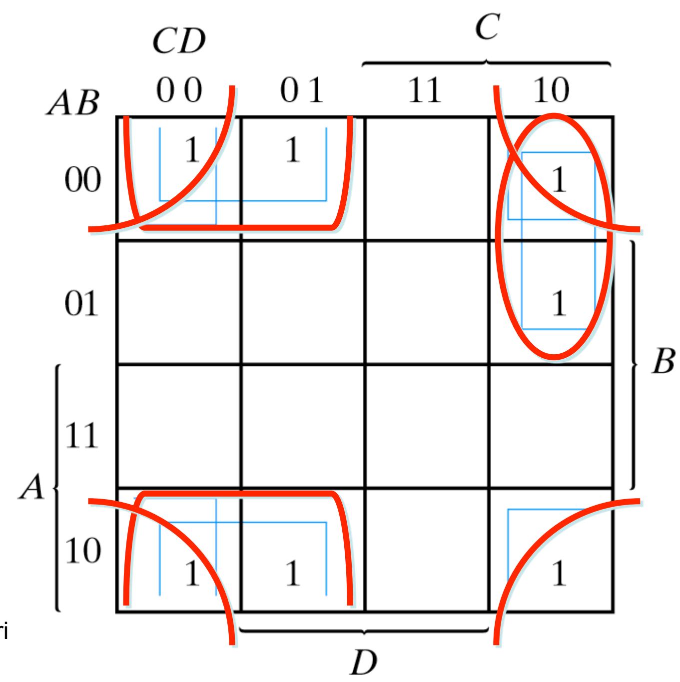
$wx \backslash yz$	00	01	11	10
00	$w'x'y'z'$	$w'x'y'z$	$w'x'yz$	$w'x'yz'$
01	$w'xy'z'$	$w'xy'z$	$w'xyz$	$w'xyz'$
11	$wxy'z'$	$wxy'z$	$wxyz$	$wxyz'$
10	$wx'y'z'$	$wx'y'z$	$wx'yz$	$wx'yz'$

Grupperingsregler for diagram med 2-4 variable

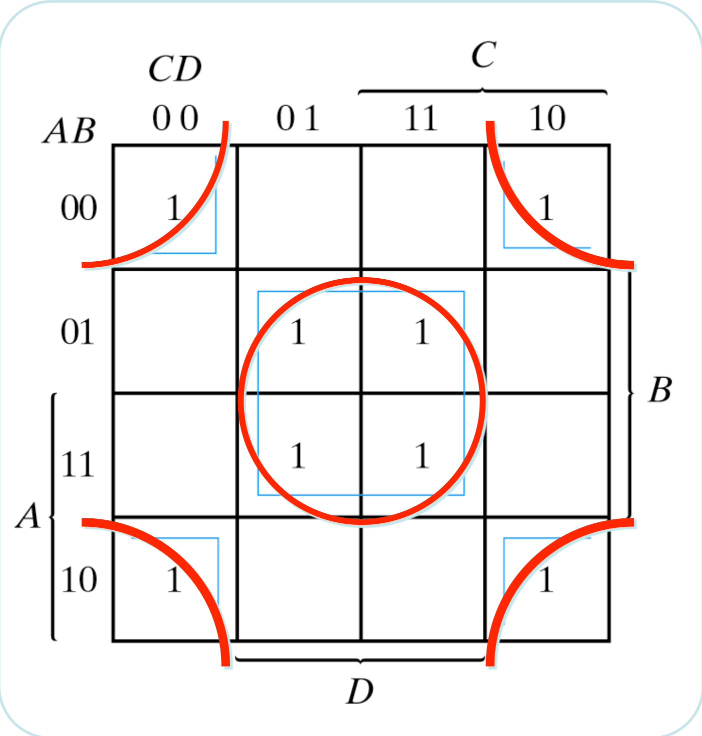
Grupperer naboruter som inneholder ”1” slik at vi får sammenhengende rektangler

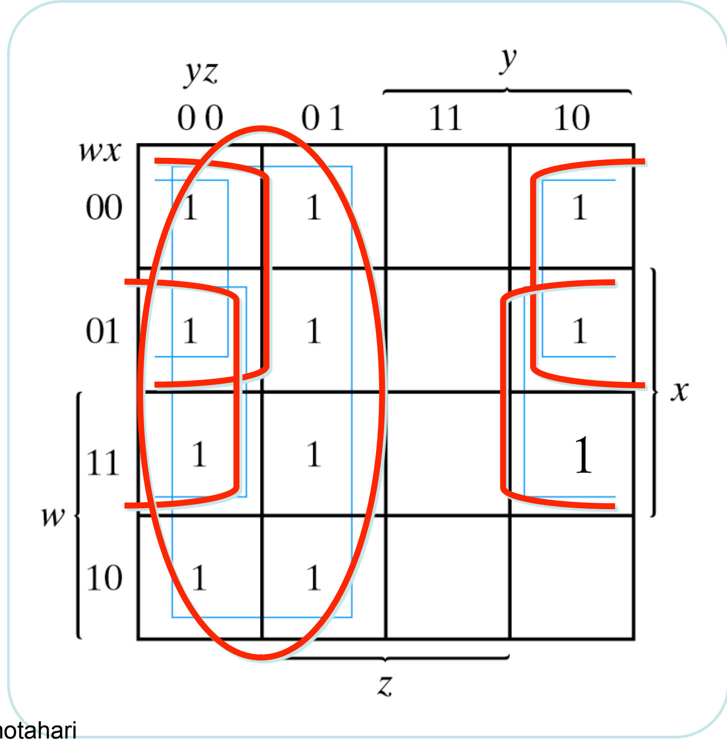
Ytterkantene av diagrammet kan også være naboruter

Eksempel

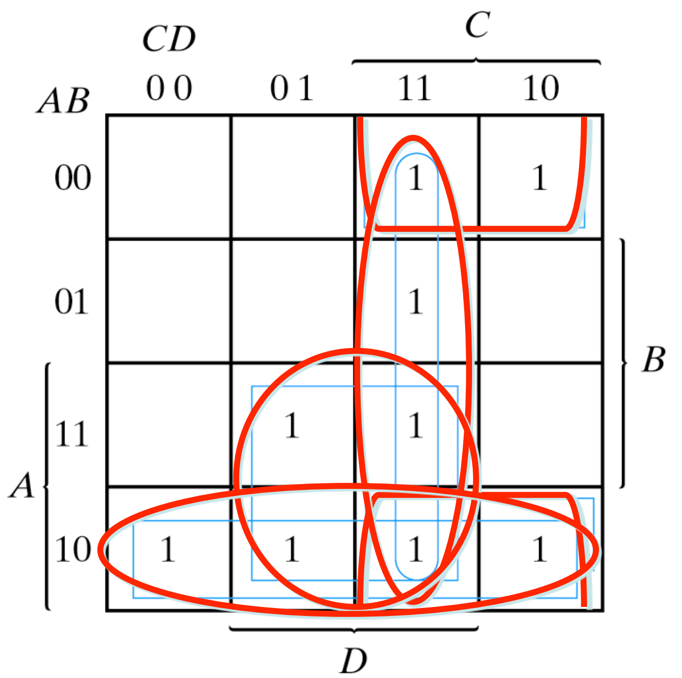


Grupperingsregler for diagram med 2-4 variable





Grupperingsregler for diagram med 2-4 variable



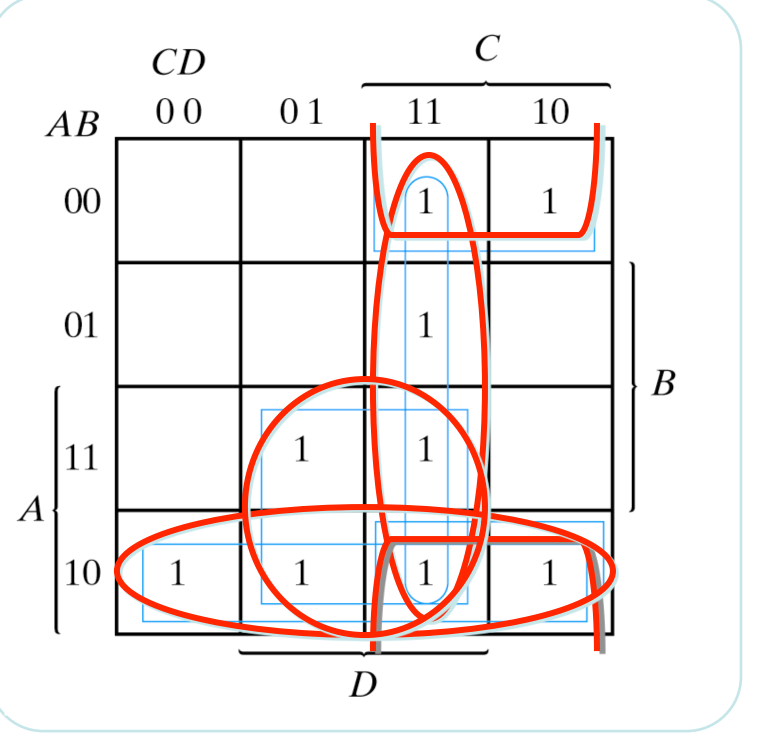
Utlesningssregler for diagram med 2-4 variable

Representerer hver gruppe ved de variablene i gruppen som ikke varierer.

Diagrammets funksjon blir summen av hvert gruppeledd:

Eksempel

$$F = AD + CD + B'C + AB'$$

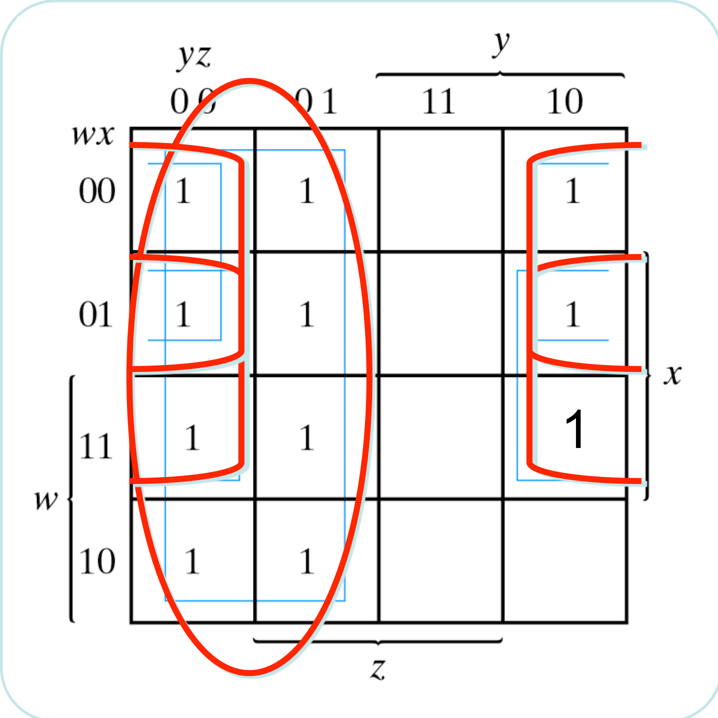


Utlesningsregler for diagram med 2-4 variable

Eksempel:

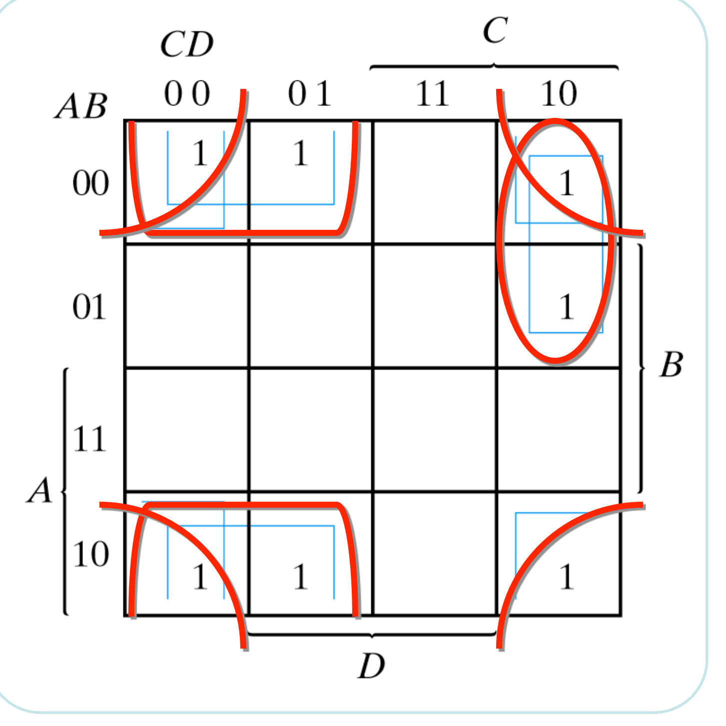
$F = y' + w'z' + xz'$

Merker oss at jo større ruter desto enklere uttrykk



Utlesningssregler for diagram med 2-4 variable

F = B´D´+ C´B´+ A´CD´



Utlesning av ”0”ere

Ved å lese ut de tomme rutene (”0”erne) fra diagrammet får man F´

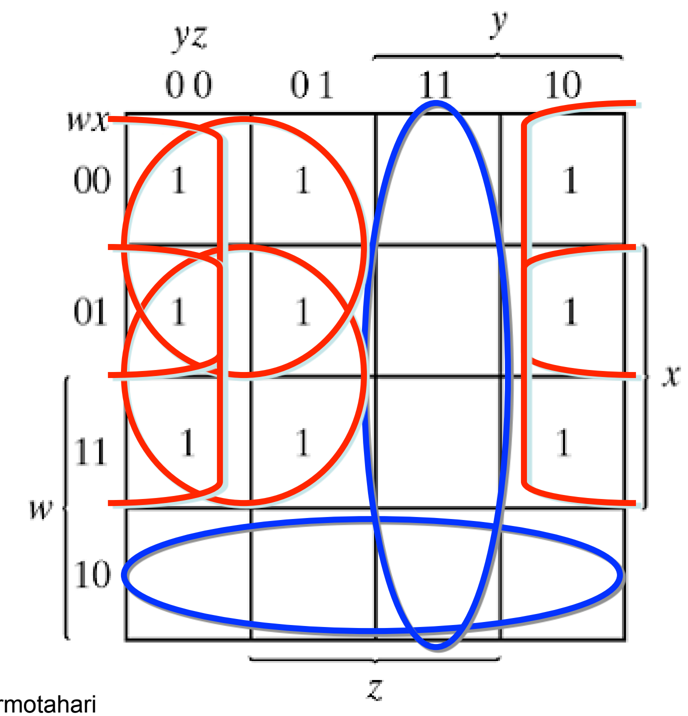
Dette kan noen ganger gi en enklere funksjon, eksempel:

F´= yz + wx´

F = (yz + wx´)´

Hadde vi lest ut ”1”ere ville vi fått

F = xy´ + w´y´+ w´z´+ xz´



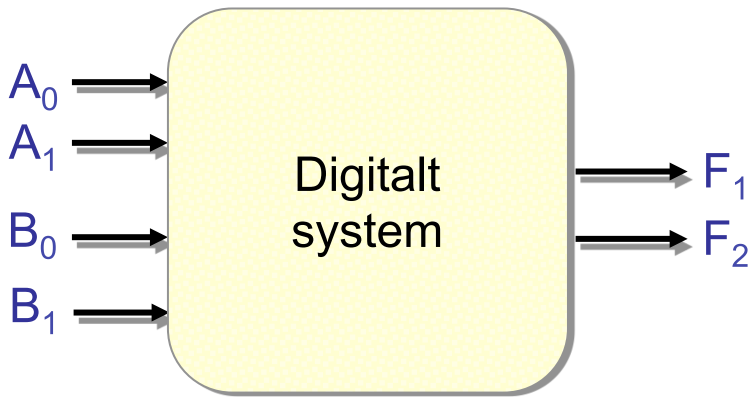
Designeksempel

Vi ønsker å designe en krets som kan sammenligne to tall A og B. Hvert tall er representert ved to bit.

Kretsen skal finne A>B samt A=B

Vi har dermed 2·2=4 innganger, og 2 utganger

Setter navn på utgangene:
F_1 for A>B og F_2 for A=B



Designeksempel

Vi trenger en oversikt over alle mulige inngangs/utgangs kombinasjoner, derfor:

- Setter opp en sannhetstabell for hver utgang (slår sammen til en dobbel tabell)
- Leser ut mintermer

$$F_1 = A_1'A_0B_1'B_0' + A_1A_0'B_1'B_0' + A_1A_0'B_1'B_0 + A_1A_0B_1'B_0' + A_1A_0B_1'B_0 + A_1A_0B_1B_0'$$

$$F_2 = A_1'A_0'B_1'B_0' + A_1'A_0B_1'B_0 + A_1A_0'B_1B_0' + A_1A_0B_1B_0$$

Innganger				Utganger	
A_1	A_0	B_1	B_0	F_1	F_2
0	0	0	0	0	1
0	0	0	1	0	0
0	0	1	0	0	0
0	0	1	1	0	0
0	1	0	0	1	0
0	1	0	1	0	1
0	1	1	0	0	0
0	1	1	1	0	0
1	0	0	0	1	0
1	0	0	1	1	0
1	0	1	0	0	1
1	0	1	1	0	0
1	1	0	0	1	0
1	1	0	1	1	0
1	1	1	0	1	0
1	1	1	1	0	1

Forenkler uttrykket for å spare porter

Setter inn i Karnaughdiagram

$$F_1 = A_1'A_0B_1'B_0' + A_1A_0'B_1'B_0' + A_1A_0'B_1'B_0 + A_1A_0B_1'B_0' + A_1A_0B_1'B_0 + A_1A_0B_1B_0'$$

$$F_2 = A_1'A_0'B_1'B_0' + A_1'A_0B_1'B_0 + A_1A_0'B_1B_0' + A_1A_0B_1B_0$$

F_1

A_1A_0 \ B_1B_0	00	01	11	10
00				
01	1			
11	1	1		1
10	1	1		

F_2

A_1A_0 \ B_1B_0	00	01	11	10
00	1			
01		1		
11			1	
10				1

Leser ut av diagrammene

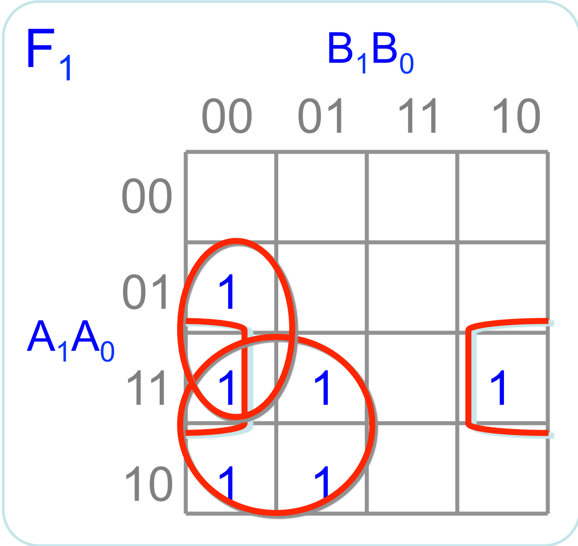
$$F_1 = A_1B_1' + A_0B_1'B_0' + A_1A_0B_0'$$

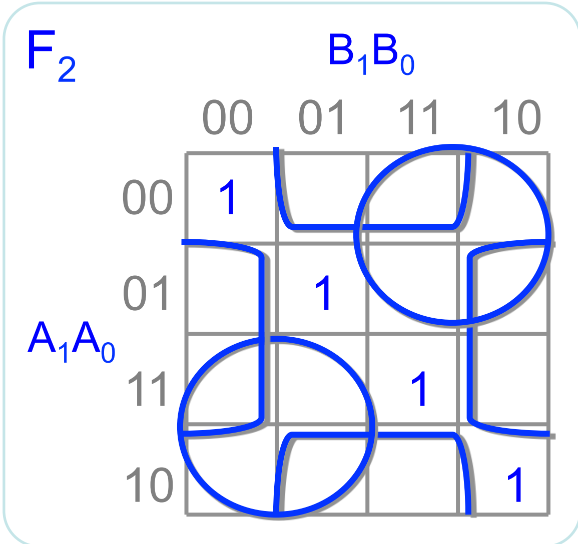
F_2: Ingen forenkling mulig ved utlesning av "1"ere, leser derfor ut "0"ere

$$F_2' = A_1B_1' + A_0B_0' + A_0'B_0 + A_1'B_1$$

Inverterer begge sider

$$F_2 = (A_1B_1' + A_0B_0' + A_0'B_0 + A_1'B_1)'$$



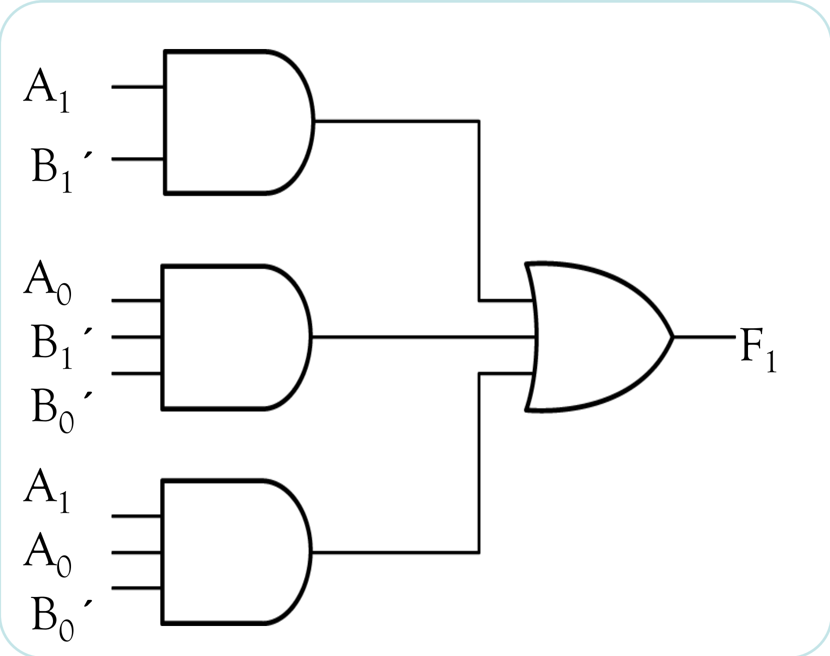


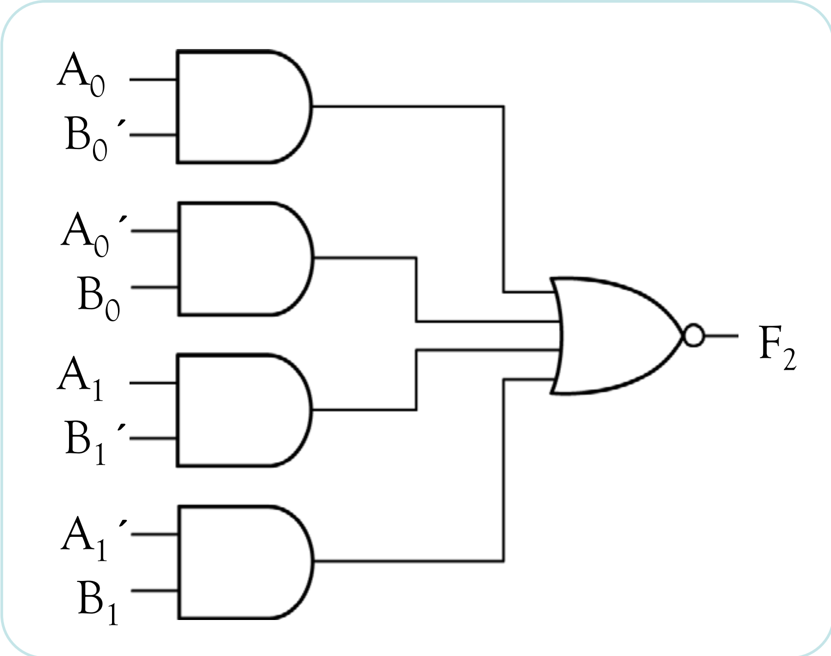
Implementerer uttrykkene

$$F_1 = A_1B_1' + A_0B_1'B_0' + A_1A_0B_0'$$

$$F_2 = (A_0B_0' + A_0'B_0 + A_1B_1' + A_1'B_1)'$$

(Hva med XOR?)





Designprosedyre:

1. Bestem innganger
2. Bestem utganger
3. Sett opp sannhetsverditabell
4. Finn mintermer
5. Sett inn i karnaughdiagram
6. Les ut av karnaughdiagram
7. Implementer uttrykkene

Algebraisk reduksjon til NAND / NOR

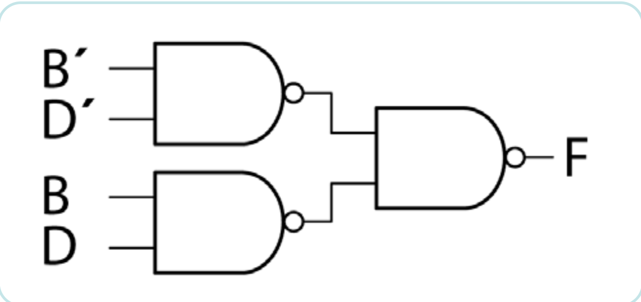
I noen tilfeller har man kun NAND og NOR kretser til rådighet

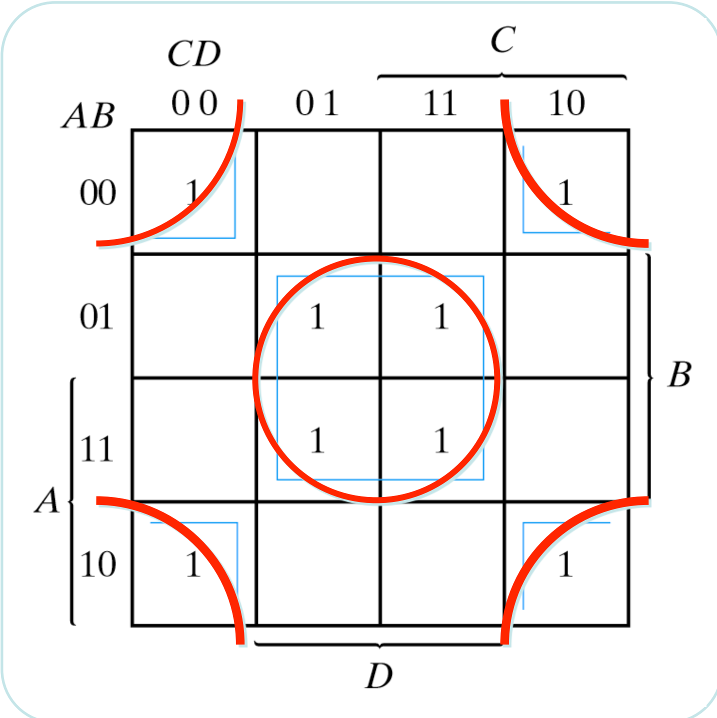
Eksempel:

$F = B'D' + BD$

DeMorgan:

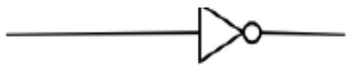
$F = ((B'D')' \cdot (BD)')'$



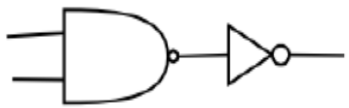


NAND konvertering / reduksjon

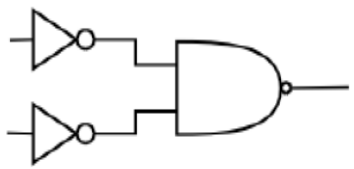
Inverter (NOT)



AND



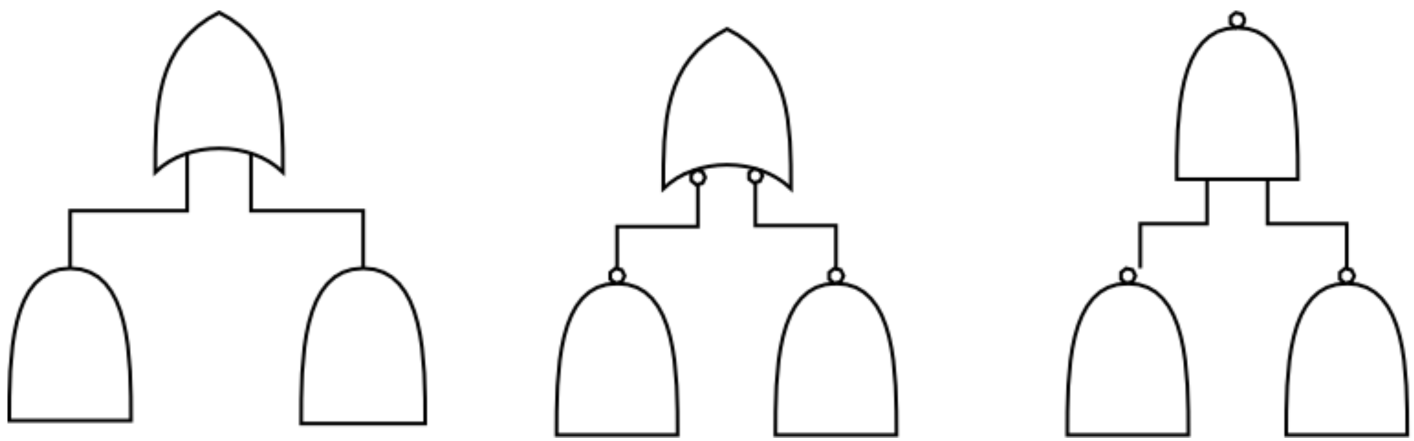
OR



All kombinatorisk logikk kan implementeres ved hjelp av NAND

NAND reduksjon

- Tegn kretsen på sum av produkt form
- Bytt ut AND med NAND og OR med invert-OR
- Bytt ut invert-ORmed NAND og sjekk alle inverteringer



Don’t care

I noen tilfeller har man inngangskombinasjoner som aldri dukker opp

I andre tilfeller bryr man seg ikke om utfallet for visse inngangskombinasjoner

Slike kombinasjoner kalles don’t care kombinasjoner og markers med ”X”

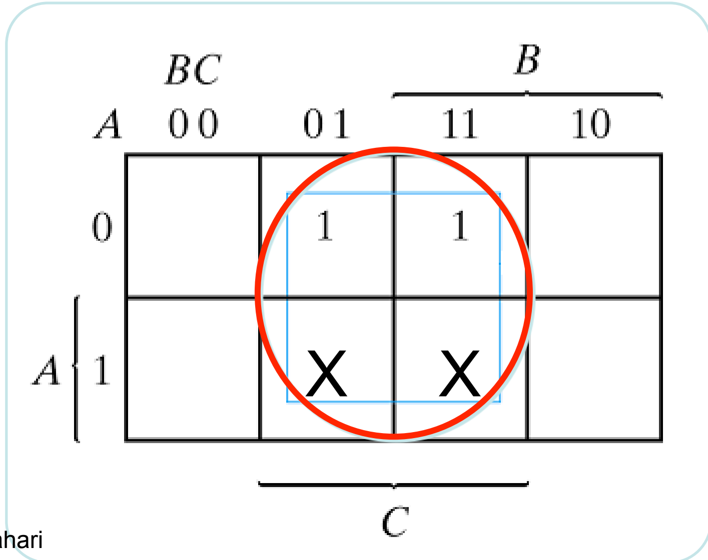
Innganger			Utganger
A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	X
1	1	0	1
1	1	1	X

Don't care

"X"er kan man sette som man vil til "0" eller "1"

I dette eksemplet velger vi "X" til "1", og får en enklere funksjon, F = C

A B C	F	
0 0 0	0	
0 0 1	1	
0 1 0	0	
0 1 1	1	
1 0 0	0	
1 0 1	X	(1)
1 1 0	0	
1 1 1	X	(1)



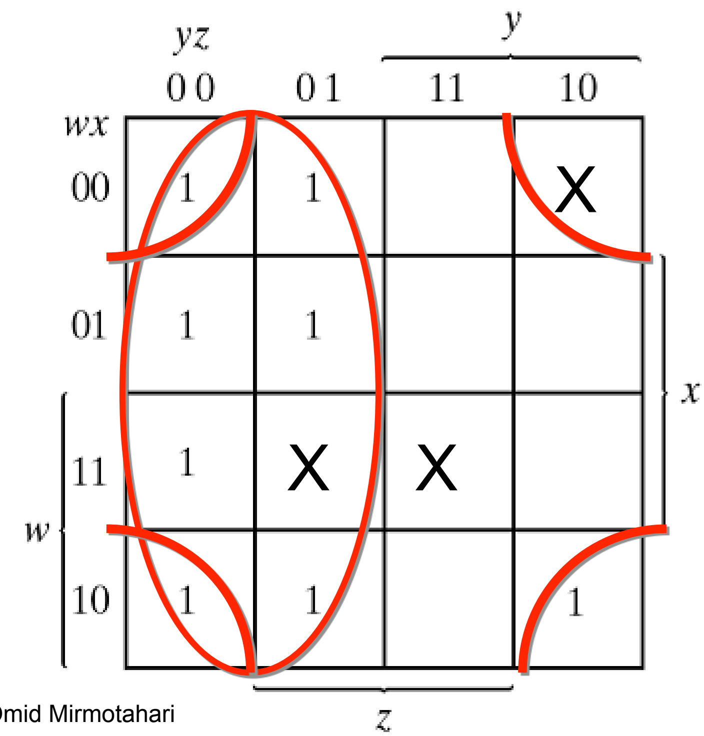
Don’t care

Eksempel:

Velger to av ”X”ene til ”1” og en ”X” til ”0”

Får to grupper:

F = y´+ x´z´



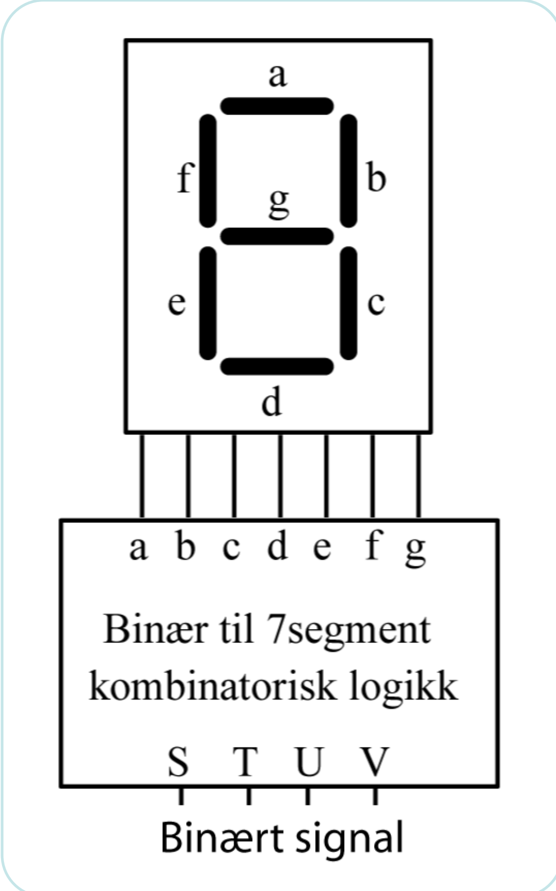
Designeksempel – Oblig nr.1

Ønsker å representere et binært tall på ett 7-segments display

5V på ledning nr. a gir lys i segment nr. a

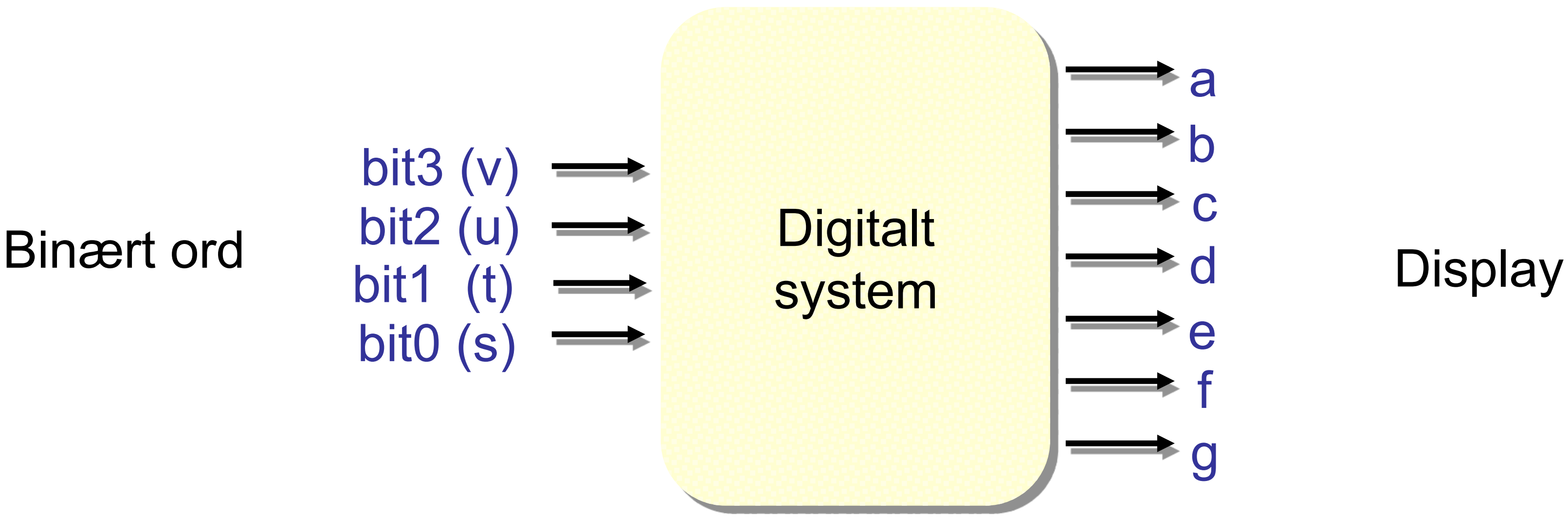
5V på ledning nr. b gir lys i segment nr. b

osv…



Oblig nr.1

Ønsker å representere et binært tall på et 7-segments display



Et 7-segmentsdisplay kan ikke visualisere tallene 10-15, for disse tallene bryr vi oss ikke om hva displayet viser
(Don’t care)

Oblig nr.1

Tips 1:

Lag en sannhetstabell for hver utgang (7stk)
(kan slås sammen i en stor tabell)

Tips 2:

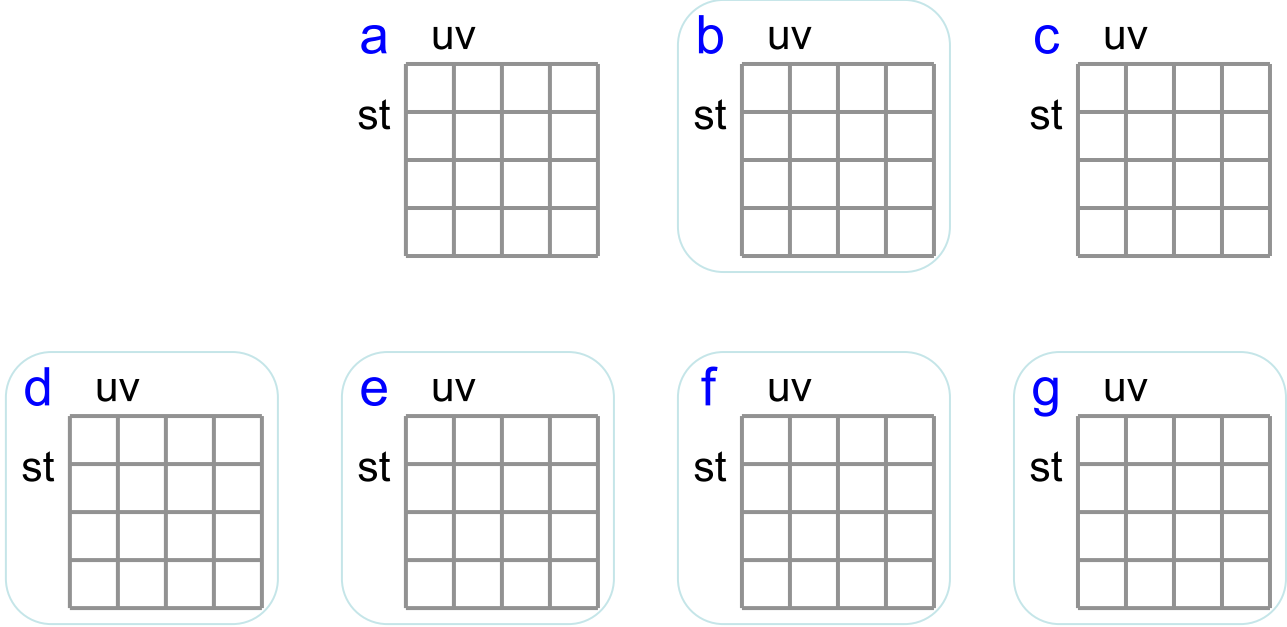
Les ut mintermer

Innganger				Utganger						
s	t	u	v	a	b	c	d	e	f	g
0	0	0	0	?	?	?	?	?	?	?
0	0	0	1	?	?	?	?	?	?	?
0	0	1	0	?	?	?	?	?	?	?
0	0	1	1	?	?	?	?	?	?	?
0	1	0	0	?	?	?	?	?	?	?
0	1	0	1	?	?	?	?	?	?	?
0	1	1	0	?	?	?	?	?	?	?
0	1	1	1	?	?	?	?	?	?	?
1	0	0	0	?	?	?	?	?	?	?
1	0	0	1	?	?	?	?	?	?	?
1	0	1	0	?	?	?	?	?	?	?
1	0	1	1	?	?	?	?	?	?	?
1	1	0	0	?	?	?	?	?	?	?
1	1	0	1	?	?	?	?	?	?	?
1	1	1	0	?	?	?	?	?	?	?
1	1	1	1	?	?	?	?	?	?	?

Oblig nr.1

Tips 3:

Bruk Karnaughdiagram for hvert utgangsuttykk (7stk)

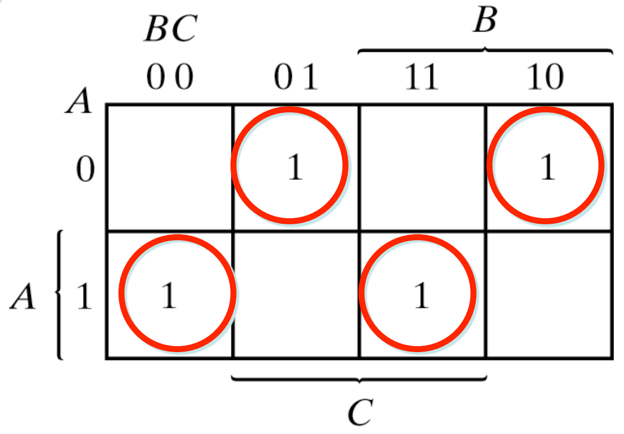


Spesialteknikker

XOR

XOR funksjonen dekker maksimalt "uheldige" "1"er plasseringer i diagrammet

Har man XOR porter til rådighet bruker man disse



(a) Odd function
$F = A \oplus B \oplus C$

I dette eksemplet kan
1 stk. 3-inputs XOR realisere F

XOR

XOR av inverterte innganger, a´, b´ osv. gir andre diagramkonfigurasjoner

XOR av 3 eller 2 innganger i et 4-variabeldiagram gir nye konfigurasjoner osv.

XNOR

cd / ab

1		1	
	1		1
1		1	
	1		1

XOR

cd / ab

	1		1
1		1	
	1		1
1		1	

cd / ab

1		1	
1		1	
	1		1
	1		1

cd / ab

		1	1
1	1		
		1	1
1	1		

cd / ab

1	1		
1	1		
		1	1
		1	1

cd / ab

		1	1
		1	1
1	1		
1	1		

XOR

XOR funksjonen kan kombineres med andre ledd

Eksempel:

$$F = A \oplus B \oplus C \oplus D + \underline{A'C'D}$$

AB \ CD	00	01	11	10
00		1		1
01	1	1	1	
11		1		1
10	1		1	

Andre algoritmer

- Karnaughdiagram er en grafisk metode for forenkling av uttrykk med 2-4 variable
  - Ikke egnet for funksjoner med flere variable
  - Ikke egnet for softwareimplementasjon

- Quine-McCluskey-algoritmen
  - Egnet for software men treg ved mange variable

- Espresso-algoritmen
  - Rask og brukes i mange designverktøy

Oppsummering

- Karnaughdiagram
  - Diagram med 2-4 variable
  - Don’t care tilstander
  - Produkt av sum (leser ut ”0”ere)
- XOR implementasjon
- NAND implementasjon ved DeMorgan