1)

A)

BCD

BCD står for binary coded decimal og er en måte å notere desimale tall på. Det vil si altså hvert enkelt tall. I det desimale tallet 43, vil 4 være representert med en binær kode, det samme vil 3 være. De desimale tallene (0-9) representeres med en binær kode på fire bit. Gjennom å bruke posisjonssystemet og de ulike vektene tallene har kan man også lett gjøre om fra desimalt til BCD-kode. 8421 BCD-kode er en slik type BCD-kode. 8421 angir vektene til de ulike tallene, altså $2^3, 2^2, 2^1, 2^0$

BCD-kode har sin store fordel når for eksempel digitale termometere, digitale klokker og liknende skal vise desimale tall. Omgjøringen gjør det lett og vise riktig tall. For summering og slikt er det ikke like effektivt som ren binær kode, og bruksområde stopper derfor ofte der.

ASCII

ASCII står for American Standard Code for Information Interchange. ASCII er et universalt kode-system som ligger I bunn av de fleste datamaskiner og digitale systemer. ASCII inneholder 128 symboler/bokstaver som blir representert av en 7-bit binær kode, men man kan se på det som 8-bit kode for MSB-bitet alltid er 0. Sammen med den binære representasjonen finnes det i ASCII også en desimal og hexadesimal representasjon av symbolet. De første 32 kodene i ASCII-systemet representerer for øvrig ulike «kommandoer» som ikke vises visuelt. Dette kan for eksempel være mellomrom og liknende. ASCII er dermed et kodesystem som brukes i nesten alle datamaskiner til å vise tall, bokstaver og tegn.

Unicode

Unicode kan ses på som en «stor» utvidelse ASCII. Unicode bruker de samme kodene for de første 128-tegnene som ASCII. Utover dette gir Unicode mulighetene til å representere alle tegn som brukes i skriftlige språk verden over. Det gjør at Unicode består av cirka 100 000 forskjellige tegn. Systemet brukes derfor i datamaskiner/ systemer som benytter seg av tekst på flere ulike språk, spesial-tegn og matematiske symboler. Hadde det ikke vært for Unicode hadde vi for eksempel ikke hatt en måte å representere «våre» spesialtegn (æ,ø,å) på.

Paritetsbit

Bruk av paritetsbit er en metode som benyttes for å detektere feil i overføring av data (informasjon). Antall enere i en binær kode kan være enten et partall eller et oddetall. Kalles ofte for «like paritet» eller «ulike paritet». Ved bruk av denne metoden legger man til et bit enten i starten eller slutten av koden. Hvis man benytter like paritet skal bitet settes til å være 1(slik at vi får et partall), omvendt skal det settes til 0 (hvis man benytter ulik paritet og antall enere allerede er et partall). Avsender og mottaker av data-en må være enige om hvilken metode som skal benyttes.

Feil detekteres ved at man ikke mottar avtalt antall enere. Hvis man for eksempel har avtalt at man benytter like paritetsbit, vil man forvente et partall antall enere i koden, skjer det da en feil hvor for eksempel en 0 i koden blir til en 1-er vil man motta en ulikt antall 1-ere og man vil få en feil.

Checksum

Bruk av checksum er en annen metode som brukes for å detektere feil i overføring av data. Denne brukes for å finne en og to-bits feil. Den kan også finne fler enn det. Metoden går ut på å legge til et gitt nummer med «check»-bits. Ofte kalt checksum, og derav navnet. Disse ekstra check-bitene blir lagt til på slutten av koden. Prosessen består av følgende steg

-Man velger en gitt kode, og det er viktig at den er forstått av både sender og mottaker

-Deretter legger man til et antall 0-er som tilsvarer antall bits i koden man valgte i trinnet over

-Data-bitene inkludert den valgte koden deles ved hjelp av modulo 2.

-Hvis resten er 0 blir det sendt, hvis det ikke er 0 blir endringer gjort slik at resten er 0.

- På det mottakerens side blir bitene som ble lagt til delt på den valgte koden fra første steg, og hvis resten er 0 blir det konkludert med at det ikke finnes feil. Er resten ulik 0 vil det ha oppstått en feil i koden.

B)

1.

$(A + BC)\overline{A} + ABC$

Ganger først inn i parantesen (Den distributive loven)

$A\,\overline{A} + \overline{A}BC + ABC$

Bruker regel 8

$0 + \overline{A}BC + ABC$

Bruker regel 1

$\overline{A}BC + ABC$

Bruker den kommutative loven

$ABC + \overline{A}BC$

Bruker lov 11

$BC + BC$

Bruker lov 5

$\underline{\underline{= BC}}$

2.

$\overline{A}B + A\overline{B} + A$

Bruker den kommutative loven

$A + \overline{A}B + A\overline{B}$

Bruker lov 11

$A + \ B + A\overline{B}$

Bruker den kommutative loven

$A + B + \overline{B}A$

Bruker lov 11

$A + B + A$

Bruker den kommutative loven

$A + A + B$

Bruker lov 5

$\underline{\underline{= A + B}}$

3.

$A\overline{C} + AB\overline{C}$

Faktorerer ut $\boldsymbol{A\overline{C}}$

$A\overline{C}(B + 1)$

Bruker regel 2 på parentesen

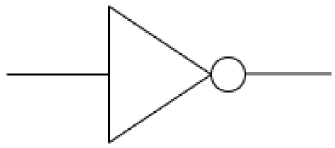
$A\overline{C} * 1$

$= A\overline{C}$

C)

Inverter

Symbol

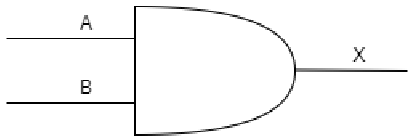


Sannhetstabell

INNPUT	OUTPUT
0	1
1	0

AND

Symbol:

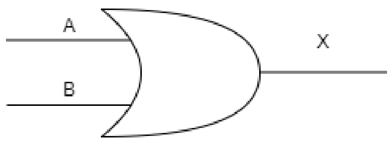


Sannhetstabell:

INNPUT	INNPUT	OUTPUT
0	0	0
0	1	0
1	0	0
1	1	1

OR

Symbol:



Sannhetstabell:

INNPUT	INNPUT	OUTPUT
0	0	0
0	1	1
1	0	1
1	1	1

D)

NAND

NAND-porten er en reversert AND-port. Altså not AND/ ikke og. Denne gir da 0 (lav) ut kun når begge inngangene er 1 (høye). NAND-porten er ofte brukt i kretser fordi de kan sammen brukes til å utføre AND, OR og Inverterer-logikken.

NOR

NOR-porten er en reversert OR-port. (Not OR/ ikke eller). Denne porten produserer en 0 (lav) når en av inputene er 1 (høy). Bare når alle inputs er 0 (lav) gir porten ut en 1 (høy). På samme måte som NAND-porten ovenfor er dette en universal port som ofte brukes i ulike kretser. Det fordi de kan settes sammen og utføre samme logikk som AND, OR og invertere.

E)

$\overline{XY} = \overline{X} + \overline{Y}$

X	Y	$\overline{X}$	$\overline{Y}$	XY	$\overline{XY}$	$\overline{X} + \overline{Y}$
0	0	1	1	0	1	1
0	1	1	0	0	1	1
1	0	0	1	0	1	1
1	1	0	0	1	0	0

Ser av de siste to kolonnene at venstre side og høyre side av utrykket av samme sannhetsverdi for alle innganger, regelen må da være riktig.

F)

Sannhetstabell: XOR

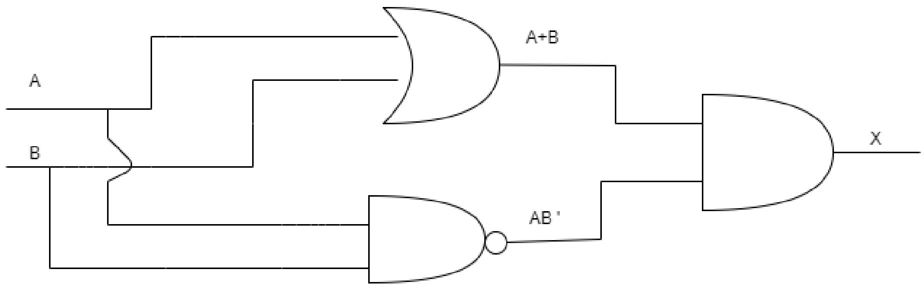
A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

Sannhetstabell: $X = (A + B)(\overline{AB})$

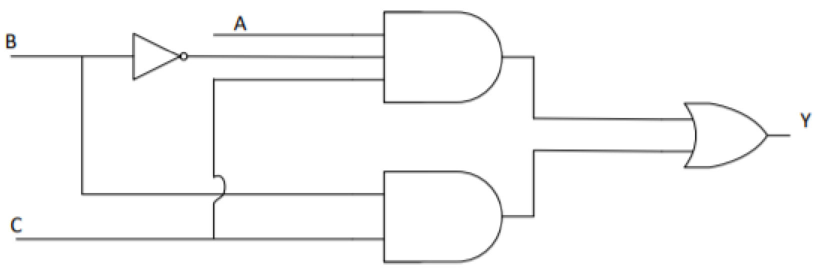
A	B	$A + B$	AB	$\overline{AB}$	$(A + B)(\overline{AB})$
0	0	0	0	1	0
0	1	1	0	1	1
1	0	1	0	1	1
1	1	1	1	0	0

Ser av sannhetstabellen at utganger (X) tilsvarer sannhetsverdiene til en XOR-port. Utrykket utfører derfor XOR.

Kretstegning:



G)



Det logiske utrykket for kretsen er:

$$Y = \left(A\overline{B}C\right) + (BC)$$

Bruker et Karnaugh Diagram, og utvider først utrykket slik at et blir på standard SOP-form:

$$Y = \left(A\overline{B}C\right) + (BC) * \left(A + \overline{A}\right)$$

$$Y = \left(A\overline{B}C\right) + (ABC) + (\overline{A}BC)$$

Leser direkte av hvert ledd i utrykket for å finne når det er sant. Utrykket er sant i disse tilfellene:

A	B	C	Y
0	1	1	1
1	0	1	1
1	1	1	1

Legger dette så inn i et Karnaugh- Diagram

C \ AB	00	01	11	10
0	0	0	0	0
1	0	1	1	1

Grupperer enerene i to grupper slik (grupper av to som overlapper hverandre)

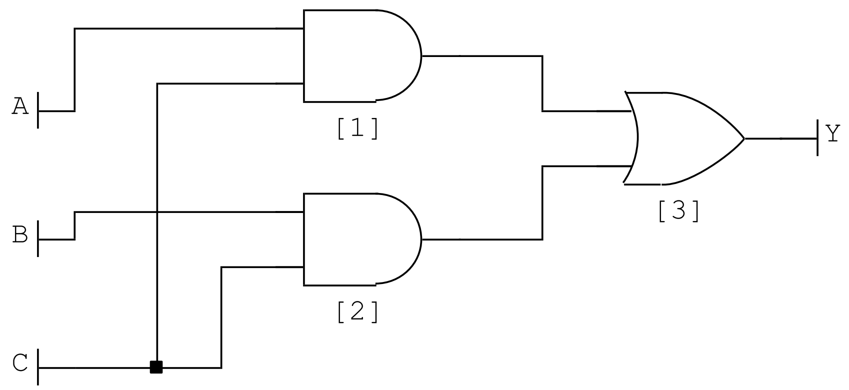
Den første gruppen gir: BC

Den andre gruppen gir: AC

Det forenklede utrykket er da:

$Y = AC + BC$

Kretstegning:



OPPGAVE 2

A)

SOP-form: $ABC + CDE + \overline{B}C\overline{D}$

POS-form: $(A + B + C)\left(A + B + \overline{D}\right)$

B)

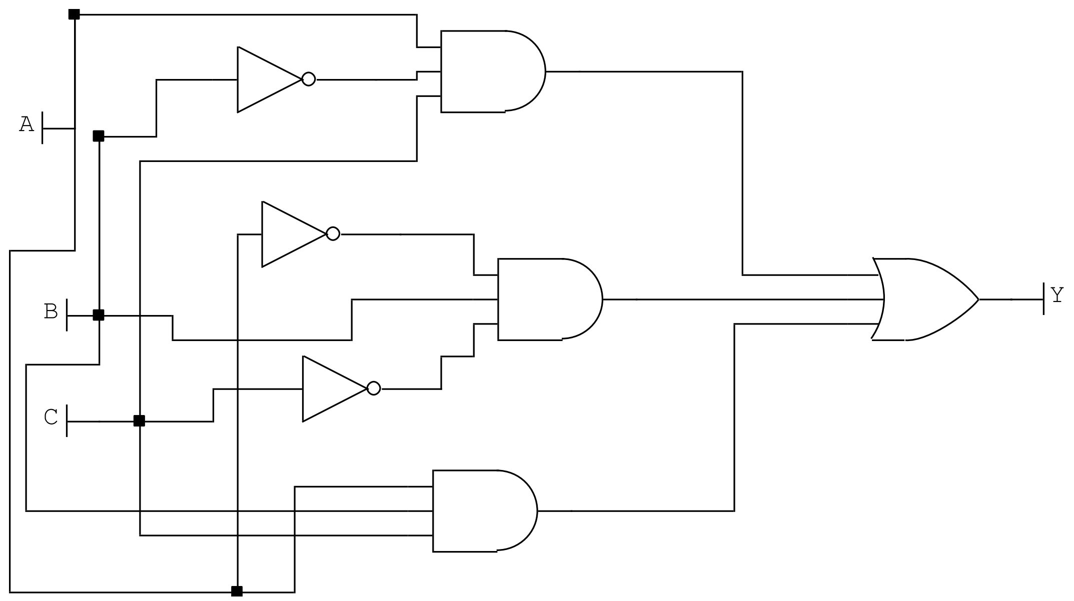
$X = A\overline{B}C + \overline{A}B\overline{C} + ABC$

Sannhetstabell:

A	B	C	X
1	0	1	1
0	1	0	1
1	1	1	1

Utrykket er falskt (0) I alle andre tilfeller

Kretstegning:



C)

$Q = \overline{X} + Y\overline{Z} + X\overline{Y}Z$

Gjør først om til standard SOP-form

$Q = \overline{X}(Y + \overline{Y})(Z + \overline{Z}) + Y\overline{Z}(X + \overline{X}) + X\overline{Y}Z$

$Q = \overline{X}Y(Z + \overline{Z}) + \overline{XY}(Z + \overline{Z}) + XY\overline{Z} + \overline{X}Y\overline{Z} + X\overline{Y}Z$

$Q = \overline{X}YZ + \overline{X}Y\overline{Z} + \overline{XY}Z + \overline{XYZ} + XY\overline{Z} + \overline{X}Y\overline{Z} + X\overline{Y}Z$

Tar bort like ledd:

$Q = \overline{X}YZ + \overline{X}Y\overline{Z} + \overline{XY}Z + \overline{XYZ} + XY\overline{Z} + X\overline{Y}Z$

Kan deretter lese av når hvert av leddene er sanne

Sannhetstabell:

X	Y	Z	Q
0	1	1	1
0	1	0	1
0	0	1	1
0	0	0	1
1	1	0	1
1	0	1	1

Utrykket er falskt i alle andre tilfeller

D)

$$\mathrm{Y} = \mathrm{AB\bar{C}\bar{D}} + \mathrm{AB\bar{C}D} + \mathrm{ABCD} + \mathrm{ABC\bar{D}} + \mathrm{A\bar{B}CD} + \mathrm{A\bar{B}C\bar{D}}$$

Faktoriserer to og to ledd:

$$Y = AB\overline{C}\left(D+\overline{D}\right) + ABC\left(D+\overline{D}\right) + A\overline{B}C\left(D+\overline{D}\right)$$

Bruker regel 6 på alle parenteser:

$$Y = AB\overline{C}\,(1) + ABC(1) + A\overline{B}C(1)$$

$$Y = AB\overline{C} + ABC + A\overline{B}C$$

Faktoriserer ut A av alle ledd:

$$Y = A\left(B\overline{C} + BC + \overline{B}C\right)$$

Faktoriserer ut C av de to siste leddene:

$$Y = A(B\overline{C} + C(\overline{B} + B))$$

Bruker regel 6 på innerste parentes:

$$Y = A(B\overline{C} + C(1))$$

$$Y = A(B\overline{C} + C)$$

Bruker den kommutative loven:

$$Y = A(\overline{C}B + C)$$

Bruker regel 11:

$$Y = A\,(B + C)$$

$$\underline{\underline{Y = AB + AC}}$$

Sannhetstabell

A	B	C	D	Y
1	1	0	0	1
1	1	0	1	1
1	1	1	1	1
1	1	1	0	1
1	0	1	1	1
1	0	1	0	1

Setter dette inn i en Karnaugh-diagram

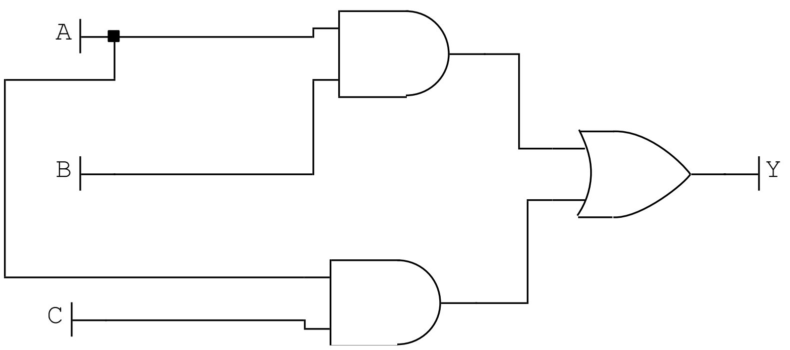
CD \ AB	00	01	11	10
00	0	0	1	0
01	0	0	1	0
11	0	0	1	1
10	0	0	1	1

Gruppen med fire 1-ere: Ser at C og D sine verdier ikke har noen innvirkning på verdien. Gir: $\boldsymbol{AB}$

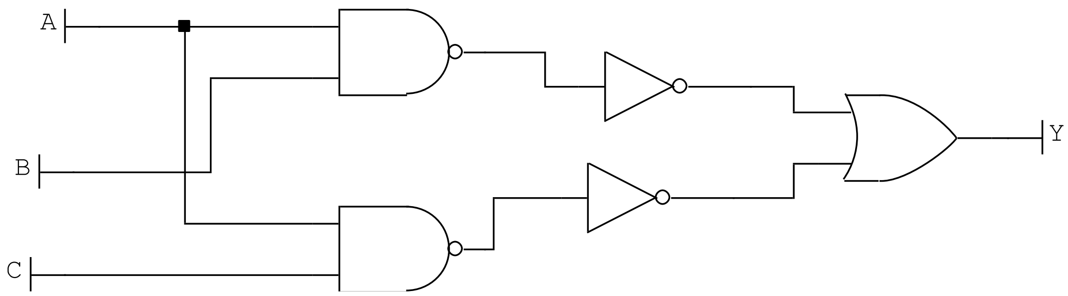
Gruppen med to 1-ere: A og C er konstante, D endrer seg og B er 0. Gir: $\boldsymbol{AC}$

$\boldsymbol{Y = AB + AC}$

Kretstegning



Kretstegning med NAND-porter



E)

$$Y = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}B\bar{C}\bar{D} + \bar{A}C\bar{D} + AB\bar{C}\bar{D} + A\bar{B}\bar{C}\bar{D} + AC\bar{D}$$

Omgjør først utrykket til standard SOP-form.

$$Y = \overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}B\overline{C}\overline{D} + \overline{A}C\overline{D}\left(B + \overline{B}\right) + AB\overline{C}\overline{D} + A\overline{B}\overline{C}\overline{D} + AC\overline{D}(B + \overline{B})$$

$$Y = \overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}B\overline{C}\overline{D} + \overline{A}BC\overline{D} + \overline{A}\overline{B}C\overline{D} + AB\overline{C}\overline{D} + A\overline{B}\overline{C}\overline{D} + ABC\overline{D} + A\overline{B}C\overline{D}$$

Sannhetstabell

A	B	C	D	Y
0	0	0	0	1
0	1	0	0	1
0	1	1	0	1
0	0	1	0	1
1	1	0	0	1
1	0	0	0	1
1	1	1	0	1
1	0	1	0	1

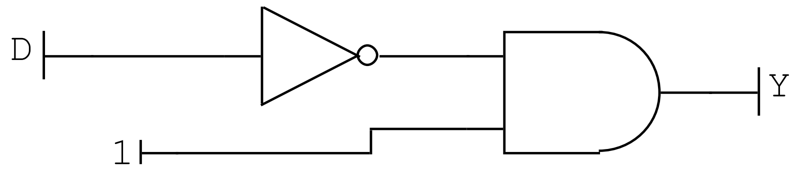
Utrykket er falskt i alle andre tilfeller

Karnaugh-Diagram:

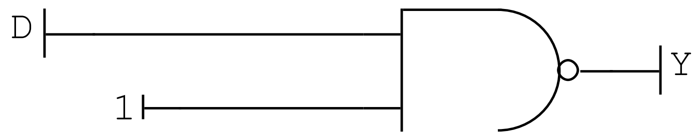
CD \ AB	00	01	11	10
00	1	1	1	1
01	0	0	0	0
11	0	0	0	0
10	1	1	1	1

$Y = \overline{D}$

Kretstegning:



Kretstegning med NAND-porter:



F)

Segment **f** brukes i tallene : 0,4, 5,6,8,9

Segment **g** brukes i tallene: 2,3,4,5,6,8,9

$0 = 0000 = \overline{A}\overline{B}\overline{C}\overline{D}$

$2 = 0010 = \overline{A}\overline{B}C\overline{D}$

$3 = 0011 = \overline{A}\overline{B}CD$

$4 = 0100 = \overline{A}B\overline{C}\overline{D}$

$5 = 0101 = \overline{A}B\overline{C}D$

$6 = 0110 = \overline{A}BC\overline{D}$

$8 = 1000 = A\overline{B}\overline{C}\,\overline{D}$

$9 = 1001 = A\overline{B}\overline{C}D$

Segment f:

$$\boldsymbol{f = \overline{A}\,\overline{B}\,\overline{C}\,\overline{D} + \overline{A}\,B\,\overline{C}\,\overline{D} + \overline{A}\,B\,\overline{C}\,D + \overline{A}\,B\,C\,\overline{D} + A\,\overline{B}\,\overline{C}\,\overline{D} + A\,\overline{B}\,\overline{C}\,D}$$

Sannhetstabell:

A	B	C	D	Y
0	0	0	0	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
1	0	0	0	1
1	0	0	1	1

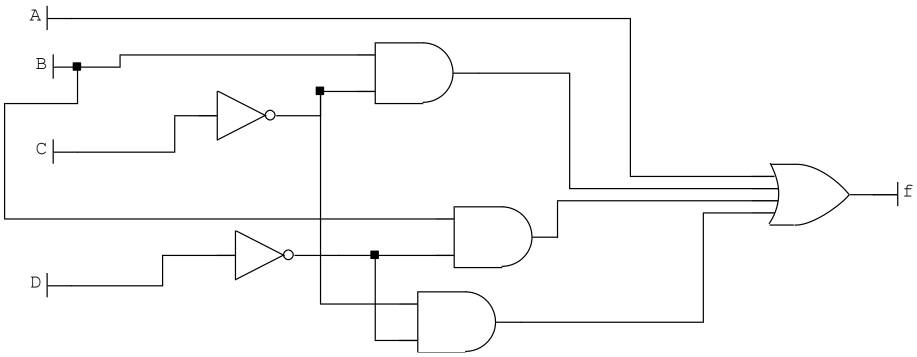
Karnaugh-Diagram (med dont-care)

CD / AB	00	01	11	10
00	1	0	0	0
01	1	1	0	1
11	X	X	X	X
10	1	1	X	X

Gir utrykket:

$$f = A + \overline{CD} + B\overline{C} + B\overline{D}$$

Kretstegning:



Segment g:

$$g = \overline{A}\,\overline{B}C\,\overline{D} + \overline{A}\;\overline{B}\,C\,D + \overline{A}\,B\,\overline{C}\,\overline{D} + \overline{A}\,B\,\overline{C}\,D + \overline{A}\,B\,C\,\overline{D} + A\,\overline{B}\,\overline{C}\;\overline{D} + A\,\overline{B}\,\overline{C}\,D$$

A	B	C	D	Y
0	0	1	0	1
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	0	1
1	0	0	1	1

Karnaugh-Diagram (med dont-care)

CD / AB	00	01	11	10
00	0	0	1	1
01	1	1	0	1
11	X	X	X	X
10	1	1	X	X

$$Y = A + C\overline{D} + B\overline{C} + \overline{B}C$$

Kretstegning: