4 – kombinatorisk logikk, løsning

1) Legg sammen følgende binærtall uten å konvertere til desimaltall:

a. 1101 + 1001 = 10110
b. 0011 + 1111 = 10010
c. 11010101 + 001011 = 11100000
d. 1110100 + 0001011 = 1111111
e. 8E + 4B (kan evt. konvertere fra heksadesimal til binær) = CA = 11011010

2) Gjør om følgende tall til 2'er komplement(oppg. E må konverteres til binær først):

a. $(10011)_2$ => $(01101)_2$
b. $(01101011)_2$ => $(10010101)_2$
c. $(10010101)_2$ => $(01101011)_2$
d. Samme som c.
e. $(57)_{10}$ => $(000111)_2$

3) Utfør følgende subtraksjoner uten å konvertere til desimaltall (hint: bruk 2'er komplement):

a. 0111111 – 0111010 = 0000101
b. 0100101 – 0010111 = 0001110
c. 010 – 000101 = 111101 = -000011
d. 000001111 – 010010011 = 101111100 = -010000100
e. 000001000 – 000000011 = 000000101
f. 000001100 – 011110111 = 100010101 = -011101011
g. 011100111 – 000010011 = 011010100
h. 010001000 – 011100010 = 110100110 = -001011010

4) Hva er det største positive tallet vi kan representere på 2'er komplement-form hvis vi har et 10-bits tall?
Svar: 511, vi har 10 bit tilgjengelig, og dermed blir det største tallet vi kan representere 0111111111 = 511

5) Hva er det minste negative tallet vi kan representere 2'er komplement-form hvis vi har et 13-bits tall?
Svar: --4096, vi har 13 bit tilgjengelig, og dermed blir det minste tallet vi kan representere 1000000000000 = -4096 (2'er komplement)

6) Multipliser følgende binærtall uten å konvertere til desimaltall:

a. 110101 og 111110 = 110011010110
b. 001011 og 110001 = 1000011011
c. 0000000010 og 0000111111 = 00001111110

7) For følgende binærtall, 1011 og 101, gjør følgende operasjoner:

a. Legg sammen uten å konvertere til desimaltall = 10000
b. Multipliser uten å konvertere til desimaltall = 110111

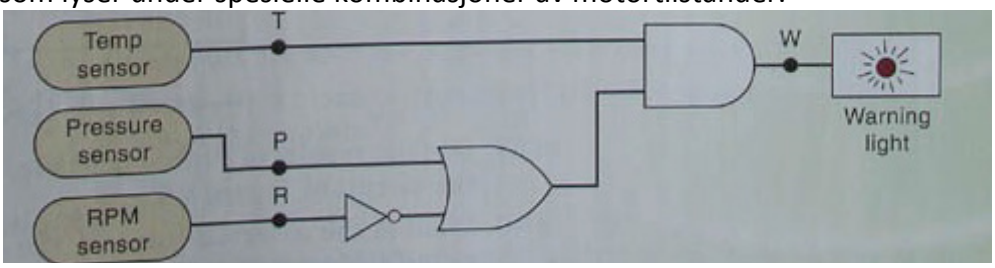
8) Et jetfly bruker et system for å overvåke rpm, trykk og temperaturverdier i motorene ved hjelp av sensorer som opererer som følger:

RPM sensor utgang = 0 bare når fart < 4800 rpm

Pressure sensor utgang = 0 bare når pressure < 220 psi

Temp sensor utgang = 0 bare når temperature < 200 F

Figuren under viser den logiske kretsen som kontrollerer en varsellampe i cockpiten, som lyser under spesielle kombinasjoner av motortilstander.

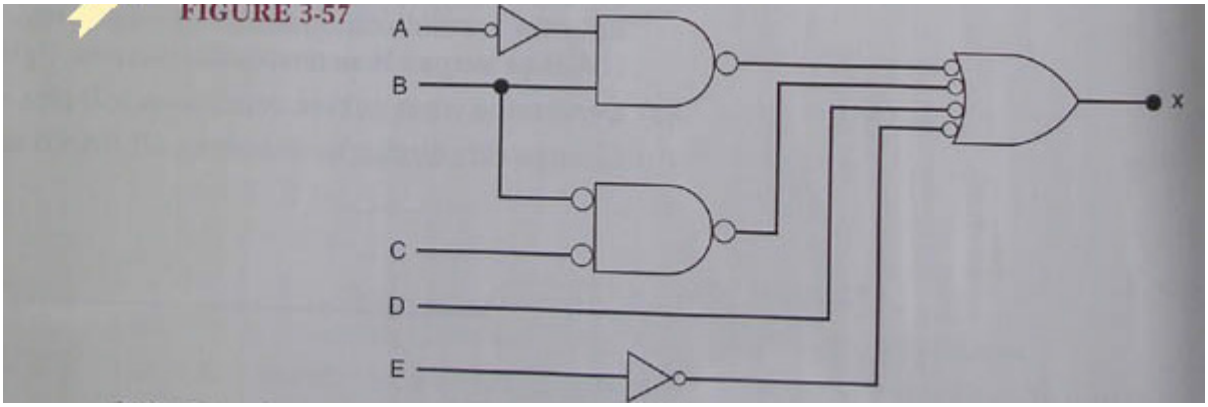


a. Avgjør hvilke motortilstander som gir en advarsel til piloten.

Piloten får advarsel når Temperaturen er over 200F, og Pressure er over 220psi og/eller farten til motorene er under 4800RPM.

b. Sett opp funksjonsuttrykket for W.
W = T(P + R')

9) For figuren under:



a. bestem hvilke inngangsverdier som gjør utgangen aktiv(x = true).
   Utgangen er aktiv når minst en av linjene under er sanne:
   - E = 1
   - D = 0
   - C = 0 og B = 0
   - A = 0 og B = 1

b. Sett opp funksjonsuttrykket for x.
   X = E + D’ + B’C’ + A’B

10) En kombinatorisk krets skal ha tre innganger, x, y og z, og tre utganger, A, B og C. Når inngangen er 0, 1, 2 og 3, skal utgangen være en større enn inngangen, når inngangen er 4, 5, 6 og 7, skal utgangen være en mindre enn inngangen.

a. Sett opp en sannhetstabell for kretsen.

x	y	z	A	B	C
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	0	1	1
1	0	1	1	0	0
1	1	0	1	0	1
1	1	1	1	1	0

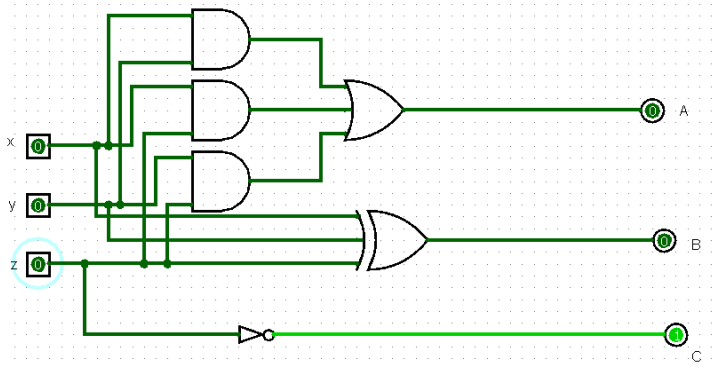
b. Sett opp funksjonsuttrykkene for de 3 utgangene.

A = xy + yz+ xz

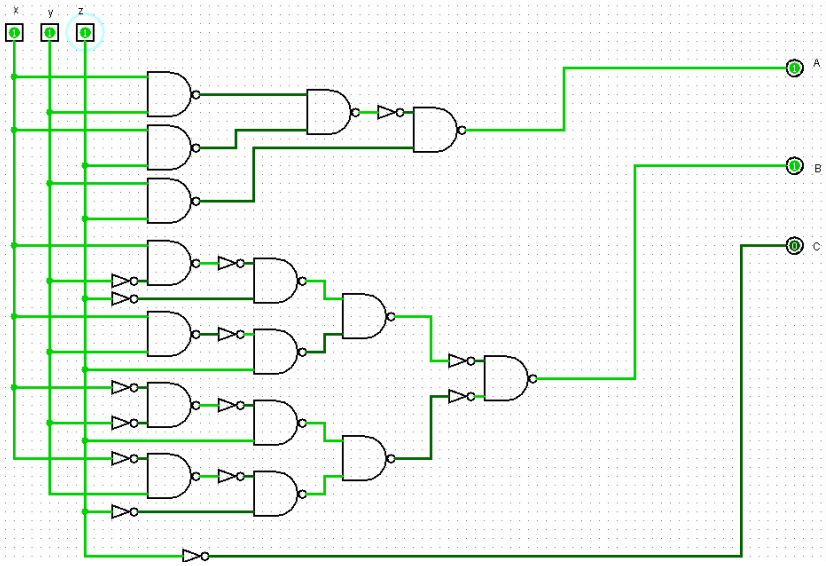
B = x XOR y XOR z

C = z’

c. Tegn opp kretsen med valgfrie porter.



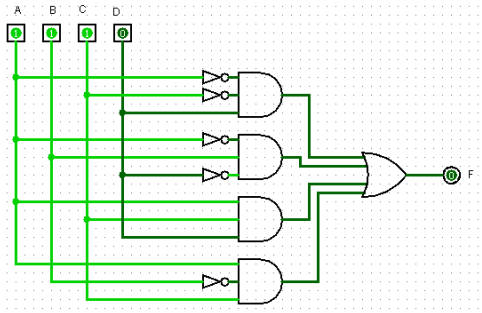
d. Tegn opp kretsen med kun 2-input NAND porter.



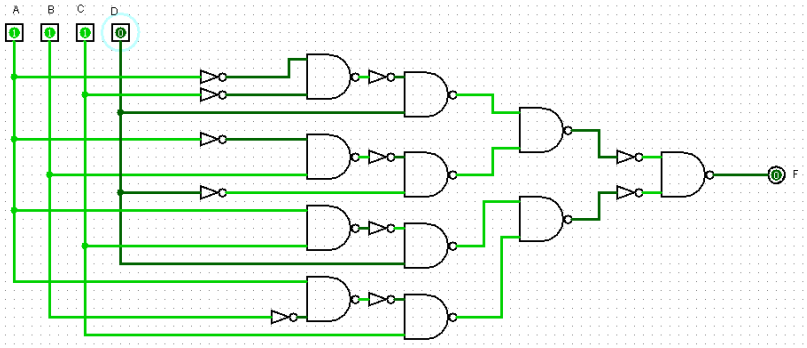
11) For funksjonsuttrykket F(A, B, C, D) = Σ(1, 4, 5, 6, 10, 11, 15):

a. Sett opp et karnaugh-diagram og forenkle funksjonen.
F = A'C'D + A'BD' + ACD + AB'C

b. Implementer kretsen med logiske porter.



c. Implementer kretsen med kun 2-inputs NAND-porter.



12) Du skal lage en krets som kan multiplisere to 2-bits tall. Kretsen har to 2-bits innganger x(x1x0) og y(y1y0), og en 4-bits utgang p(p3p2p1p0).

a. Sett opp sannhetstabellen for utgangene.

X1	X0	Y1	Y0	P3	P2	P1	P0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	1	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	1	1	0
1	1	1	1	1	0	0	1

b. Forenkle funksjonene for utgangene ved hjelp av fire karnaugh-diagrammer.

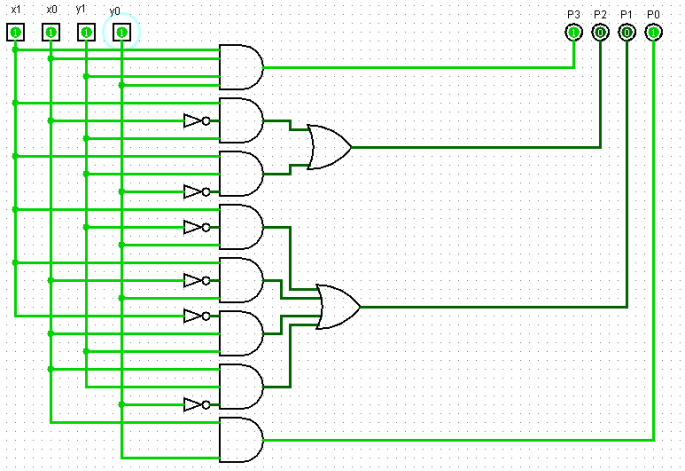
P3 = x1x0y1y0

P2 = x1x0'y1 + x1y1y0'

P1 = x1y1'y0 + x1x0'y0 + x1'x0y1 + x0y1y0'

P0 = x0y0

c. Tegn opp kretsen.



13) Du har en kombinatorisk krets som legger til 1 til et 4-bits binærtall. Det betyr at hvis inngangen er 1101, skal utgangen være 1110. For denne kretsen:

a. Lag sannhetstabellen for kretsen.

A	B	C	D	P3	P2	P1	P0
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	1	1
0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	1
1	0	0	1	1	0	1	0
1	0	1	0	1	0	1	1
1	0	1	1	1	1	0	0
1	1	0	0	1	1	0	1
1	1	0	1	1	1	1	0
1	1	1	0	1	1	1	1
1	1	1	1	0	0	0	0

b. Sett opp funksjonsuttrykkene til kretsen ved å bruke regneregler for å forenkle.

   P3 = AB’ + AC’ + AD’ + A’BCD

   P2 = BC’ + BD’ + B’CD

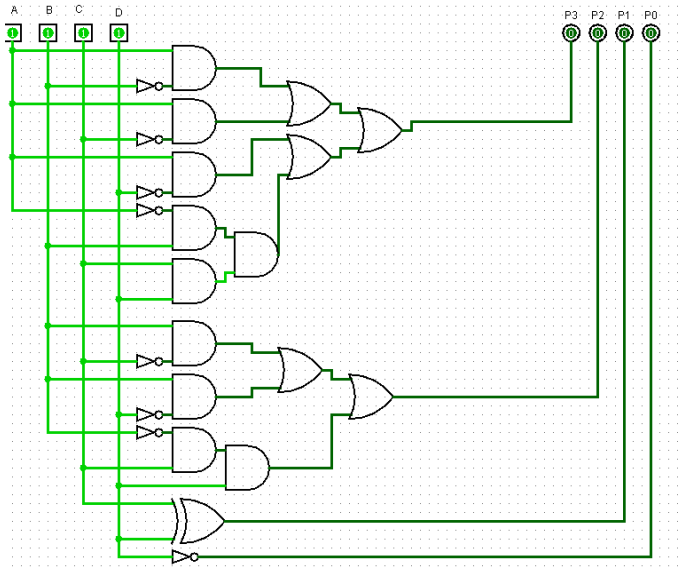
   P1 = C’D + CD’

   P0 = D’

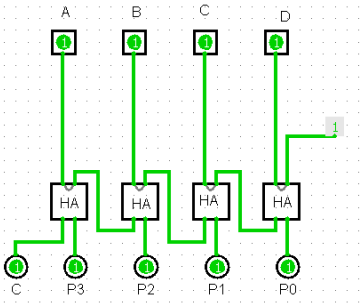
c. Sett opp funksjonsuttrykkene til kretsen ved å bruke fire 4-verdi karnaugh-diagram.

   Samme svar som 13b.

d. Tegn opp kretsen ved hjelp av 2-inngangs porter.

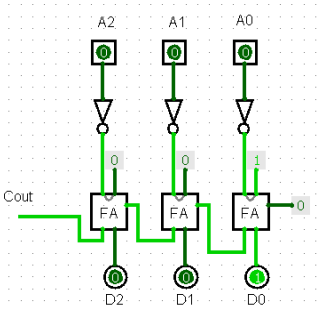


e. Tegn opp kretsen kun ved hjelp av 4 halvaddere.

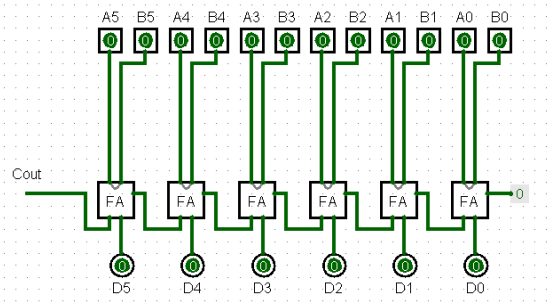


14) Vanskelig) Under er tabellen for en 3-bits 2'er-komplement konvertering. Implementer den ved hjelp av fulladdere.

A_2	A_1	A_0	D_2	D_1	D_0
0	0	0	0	0	0
0	0	1	1	1	1
0	1	0	1	1	0
0	1	1	1	0	1
1	0	0	1	0	0
1	0	1	0	1	1
1	1	0	0	1	0
1	1	1	0	0	1



15) Tegn en krets som kan legge sammen to 6-bits tall. (Hint: bruk fulladdere)



16) Hvorfor gir ”Carry Lookahead” en økt hastighet i forhold til ”Carry Propagate”

Fordi ”Carry Propagate”-kretser er laget slik at hver Full Adder må vente på svaret fra forrige Full Adder for å regne ut riktig svar (avhengig av en annen krets’ output). En liten forsinkelse per Full Adder blir fort en god del forsinkelse når man har mange Full Addere etter hverandre.