UNIVERSITETET I OSLO

Det matematisk-naturvitenskapelige fakultet

Eksamen i	IN 240 — Digital Systemkonstruksjon
Eksamensdag:	6. desember 2001
Tid for eksamen:	9.00–15.00
Oppgavesettet er på 4 sider.	
Vedlegg:	Ingen
Tillatte hjelpemidler:	Ingen

Kontroller at oppgavesettet er komplett før
du begynner å besvare spørsmålene.

Oppgave 1

1-a (vekt 5 %)

En krets har inngangene a,b,c,d og en utgang gitt av funksjonen:

F=abc+abc'+ab'cd'+a'b'd'

Forenkle funksjonen ved å bruke Karnaugh-diagram og implementer den med et minimalt antall 2-input NAND-porter. Anta at inngangssignalene finnes i både invertert og ikke-invertert utgave.

1-b (vekt 5 %)

Hvilke hovedgrupper finnes av moderne programmerbare logiske kretser (FPLD)? Hva er hovedforskjellene mellom disse?

1-c (vekt 5 %)

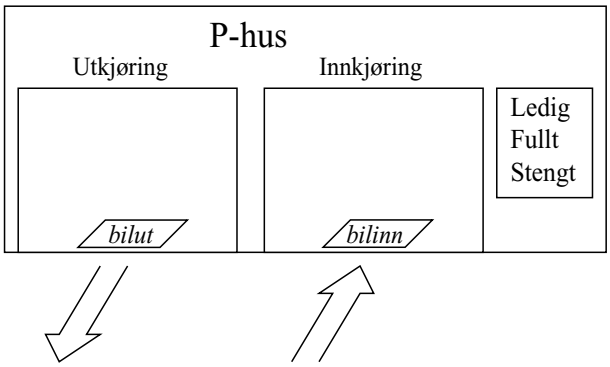
Hva menes med uttrykket *minterm*? Uttrykk følgende funksjon ved hjelp av mintermer:

F=ab+a'c'

(Fortsettes på side 2.)

Oppgave 2 Display til P-hus

Du skal i denne oppgaven designe og implementere et styringssystem til et display som er plassert ved innkjøringen til et parkeringshus (P-hus):



P-huset har plass til 17 biler. Styringssystemet skal holde oversikt over antall biler som til enhver tid er i huset. Dette er basert på en sensor kalt *bilinn* som registrerer at en bil kjører inn og en annen sensor kalt *bilut* som registrerer at en bil kjører ut. Når personalet ønsker at P-huset skal være stengt, settes en bryter (kalt *aapent*) i AV-stilling. Styringssystemet skal ha *en* digital utgang for hvert av stikkordene som kan tennes på displayet. Disse er igjen koblet selve displayet. Styringssystemet skal alltid ha aktiv *en* (og kun en) av disse tre utgangene under de gitte forutsetningene:

- **Ledig:** Antall biler i huset er mindre enn 17.
- **Fullt:** Antall biler i huset er flere eller lik 17.
- **Stengt:** Bryteren *aapent* står i AV-stilling. Dørene til P-huset er også nå lukket.

Ved alle disse tre tilfellene må styringssystemet lagre antall biler i huset. Sensorene som registrerer biler sender ut en "1"-puls med varighet på en klokkeperiode for hver bil som passerer. En kan anta at ikke begge sensorene er aktive samtidig. En kan videre anta at signalet fra bryteren er fri for støy. Spesifiser eventuelt egne forutsetninger du gjør utover oppgaveteksten.

2-a (vekt 15%)

Tegn et ASM-diagram for en tilstandsmaskin som sjekker sensorene og bryteren og gir respons på utgangene i henhold til beskrivelsen over.

(Fortsettes på side 3.)

2-b (vekt 15%)

Skriv et VHDL program i "behavioral style" for systemet du beskrev i punkt 2-a. Benytt følgende entitet for systemet:

```
ENTITY p_hus_kontroll IS
PORT
(
  clk          : IN BIT;  -- Klokke
  bilinn       : IN BIT;  -- Sensor som detekterer innkjørende bil
  bilut        : IN BIT;  -- Sensor som detekterer utkjørende bil
  aapent       : IN BIT;  -- Bryter som gir ut ``1'' når den er PÅ

  ledig        : OUT BIT; -- Utgang som går til ``Ledig'' på display
  fullt        : OUT BIT; -- Utgang som går til ``Fullt'' på display
  stengt       : OUT BIT  -- Utgang som går til ``Stengt'' på display
);
END p_hus_kontroll;
```

Oppgave 3 Multiplekser

3-a (vekt 5%)

Tegn skjema (med AND, OR og NOT porter) for en multiplekser med 4 bits inngang og en utgang. Forklar virkemåten.

3-b (vekt 10%)

Benytt multiplekseren du tegnet i forrige punkt til å realisere funksjonen:

F=abc + ab'c'+ a'b'c'

Oppgave 4 Teller

4-a (vekt 5%)

Hva er forskjellen på en *ripple* teller og en *synkron* teller? Hva er den største ulempen med en ripple teller i forhold til en synkron teller?

4-b (vekt 10%)

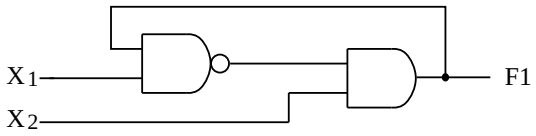
Design er 3 bits synkron teller ved å benytte D-vipper. Bruk så få komponenter som mulig. Tegn skjema for kretsen du designer.

(Fortsettes på side 4.)

Oppgave 5 Asynkron logikk

5-a (vekt 5%)

Gitt følgende krets:



Tegn transisjonstabell for kretsen. Marker stabile tilstander i tabellen. Hva kommer ut på utgangen F1 når inngangene settes til $X_1 = X_2 =$ "1" ?

5-b (vekt 5%)

Er det fare for kappløp i kretsen i punkt 5-a? Begrunn svaret.

Oppgave 6 A/D og D/A-konvertering

6-a (vekt 5%)

Tegn blokkskjema for en suksessiv-approksimasjon A/D-konverterer og forklar virkemåten.

6-b (vekt 5%)

Hva er hovedfordelen med en suksessiv-approksimasjon A/D-konverterer i forhold til en tellende (counting) A/D-konverterer? Finnes det raskere metoder for A/D-konvertering? Begrunn svarene.

6-c (vekt 5%)

En 3-bits D/A-konverter har en utgangsspenning på 0.2V når kun minst signifikante bit er "1". Hva er den maksimalt oppnåelige spenningen ut?

EKSAMEN IN 240 6. des. 2001

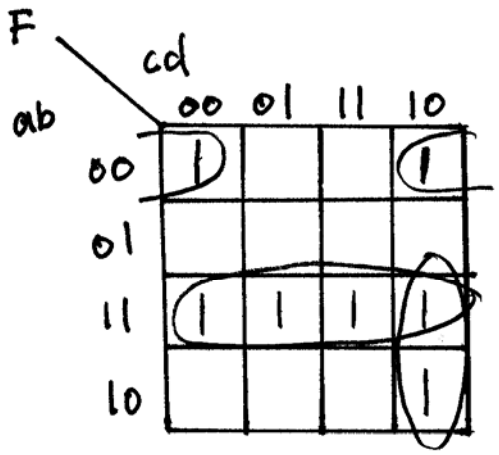
Relevante oppgaver i INF 1400 H04

Oppgave 1

a) $F = abc + abc' + ab'cd' + a'b'd'$

$= abc(d+d') + abc'(d+d') + ab'cd' + a'b'd'(c+c')$

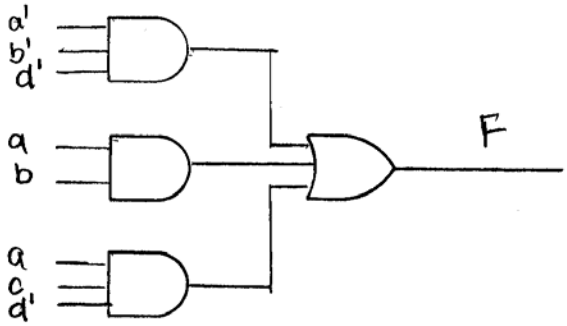
$= abcd + abcd' + abc'd + abc'd' + ab'cd' + a'b'cd' + a'b'c'd'$



Forenklet,

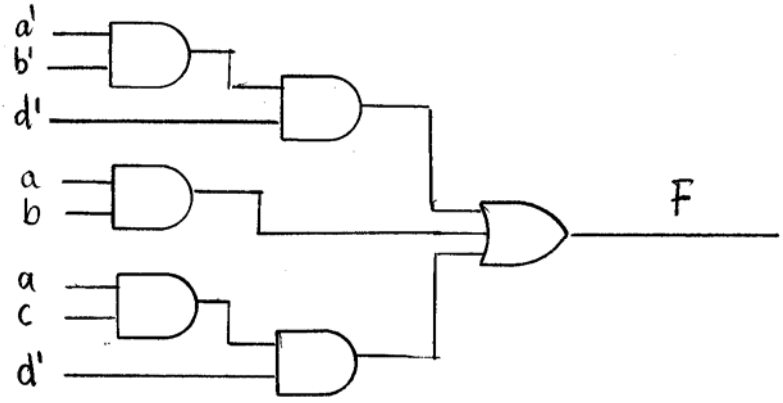
$F = a'b'd' + ab + acd'$

Implementering av F i forenklet utgave (slik den fremkommer fra Karnaugh-diagrammet).

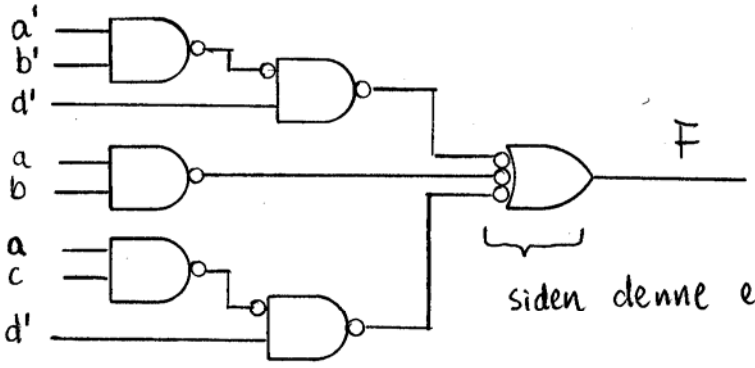


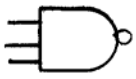
Eksamen IN240 6. des. 2001,
Fortsettelse oppg. 1a)

Overgang til 2-ports AND:
(for 1-ste og 2-dre nivå)

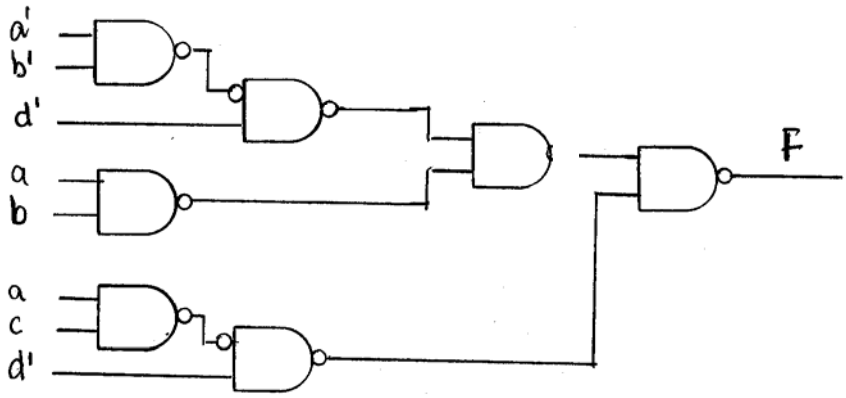


Overgang til 2-ports NAND:



siden denne er lik 

så blir kretsen slik:
(med 2-ports NAND)

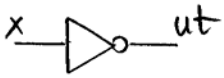


Eksamen IN240 6. des. 2001

Fortsettelse oppg. 1a)

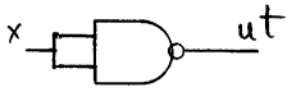
Spørsmål:
kan en avslutte oppgaven slik, eller må en endre inverterte inngangen i 2-nivå med NAND-porter

Siden:



x	ut
0	1
1	0

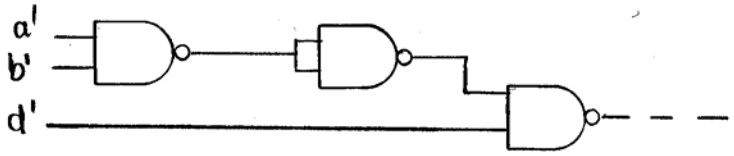
er lik



x	ut
0	1
1	0

kan kretsen's inverterte inngangen tegnes slik:

eks. for a'b'd' 1 til 2 nivå:



Ja, fint!

IN240 – 2001

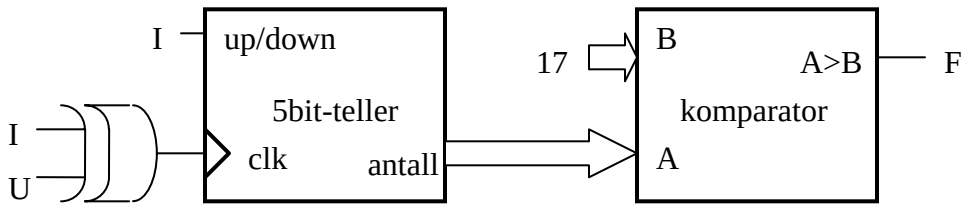
Antar vi har en 5bits synkron binærteller med ”up/down” inngang. Dvs. telleren kan telle både opp og ned avhengig av verdien inn på ”up/down” inngangen. Antar ”up/down”=1 får telleren til å telle opp på positiv klokkeflanke, og ”up/down”=0 får den til å telle ned på positiv klokkeflanke. Denne telleren skal vi ikke designe for hånd, kun beskrive i VHDL (oppgave 2b).

Klokkesignalet til telleren kommer fra bilInn / bilUt sensorene. For å få med en evt. situasjon der både bil kommer inn og ut samtidig, XORer vi sensorsignalene slik at telleren ikke teller da. Tellersignalet komparerer vi fortløpende med 17 (se figur) . Komparator designer vi i VHDL (oppgave 2b).

Signaler:

I – bil registrert inn
U – bil registrert ut
F – antall biler >17 / fullt (antar max 31biler i huset)
A – aapen bryter (”1” - åpent)

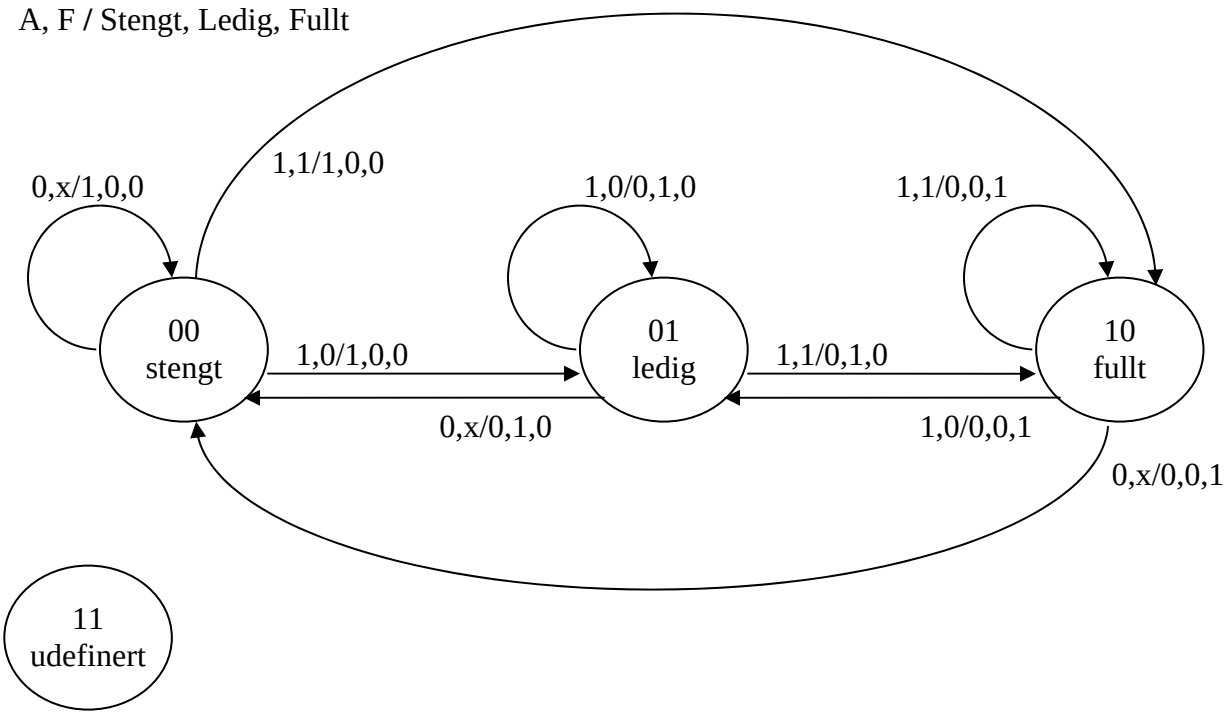
Dette systemet står og går kontinuerlig og produserer inngangssignaler til tilstandsmaskinen vi nå skal designe.



Definerer 3 tilstander:

00 – stengt
01 – ledig
10 – fullt

Tegner tilstandsdiagram



Tilstandstabell:

Antar at tilstandsmaskinen i prinsippet kan starte opp i tilstand ”11” (udefinert). Tar først sjansen på å sette inn X’er i denne tilstanden (må etterpå sjekke om tilstanden blir isolert)

Q1	Q2	A	F	D1	D2	Stengt	Ledig	Fullt
0	0	0	0	0	0	1	0	0
0	0	0	1	0	0	1	0	0
0	0	1	0	0	1	1	0	0
0	0	1	1	1	0	1	0	0
0	1	0	0	0	0	0	1	0
0	1	0	1	0	0	0	1	0
0	1	1	0	0	1	0	1	0
0	1	1	1	1	0	0	1	0
1	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0	1
1	0	1	0	0	1	0	0	1
1	0	1	1	1	0	0	0	1
1	1	0	0	x	x	x	x	x
1	1	0	1	x	x	x	x	x
1	1	1	0	x	x	x	x	x
1	1	1	1	x	x	x	x	x

Fra tabell:

D1 = Q1’ Q2’ A F + Q1 Q2’ A F + Q1 Q2’ A F
D2 = Q1’ Q2’ A F’ + Q1’ Q2 A F’ + Q1 Q2’ A F’
Stengt = Q1’ Q2’ / Ledig = Q1’ Q2 / Fullt = Q1 Q2’

Setter inn i karnaugh

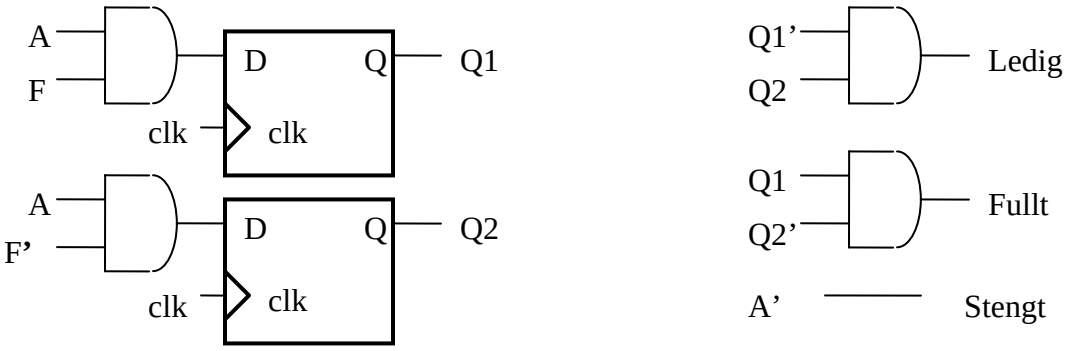
D1	AF				
		00	01	11	10
	00			1	
Q1Q2	01			1	
	11	x	x	x	x
	10			1	

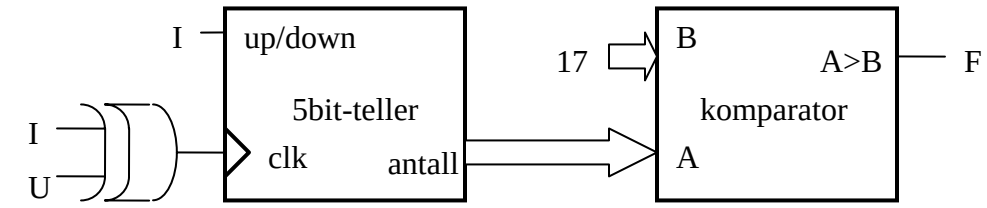
D2	AF				
		00	01	11	10
	00				1
Q1Q2	01				1
	11	x	x	x	x
	10				1

Leser ut

D1 = AF
D2 = AF’

Stengt = Q1’ Q2’ = A’
Ledig = Q1’ Q2
Fullt = Q1 Q2’





Udefinert tilstand
Sjekker tilstand ”11”: ser først på D1. D1=AF, dvs. når det ikke er både åpent og fullt samtidig vil D1 være 0. Dette betyr at i situasjoner korresponderende til linje 13,14,15 i tilstandstabellen vil Q1 gå til 0 og vi går til tilstand ”00” eller ”01”. Ser på D2. D2=AF’, dvs. når det ikke er både åpent og ikke fullt samtidig vil Q2 gå til 0, og vi er i tilstand ”00” eller ”10”. Dette skjer for linje 13,14,16. Dermed vil alle situasjoner for denne tilstanden lede til definerte tilstander og vi kan beholde logikken som den er.