

UNIVERSITETET I OSLO

Det matematisk-naturvitenskapelige fakultet

Eksamen i:	INF1400 – Digital teknologi
Eksamensdag:	29. november 2011
Tid for eksamen:	Kl. 14.30-18.30 (4 timer)
Vedlegg:	Ingen
Tillatte hjelpemidler:	Alle trykte og skriftlige
Oppgavesettet er på 11 sider	

Kontroller at oppgavesettet er komplett før du begynner å besvare spørsmålene

Hele eksamen består av flervalgsoppgaver (multiple-choice). For alle oppgaver gjelder:

- **Oppgaven skal kun besvares med ETT kryss. Alternativt ingen kryss hvis du ikke vet svaret.**
- **Riktig avkryssing gir 4 poeng.**
- **Ingen kryss gir 0 poeng.**
- **Gal avkryssing gir -1 poeng.**
- **Flere kryss enn ett, gir 0 poeng.**
- **Hver oppgave teller likt i den endelige poengsummen**

Lykke til!

Oppgave 1

Les ut et maksimalt forenklet uttrykk fra følgende Karnaughdiagram basert på utlesning av 1 ere (produkt av sum) :

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

$B'D' + A'C + A'B + BC$

$BC + A'B' + B'CD'$

$B'C + A'B$

$ABCD$

$A'C + D' + BC + 1$

Oppgave 2

Les ut et maksimalt forenklet uttrykk fra følgende Karnaughdiagram basert på utlesning av 1 ere (produkt av sum). X – betyr dont'care.

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

$$D + C + B$$

$$A'C' + BCD' + X$$

$$ACD$$

$$C'D + A'C' + C'D' + BC$$

$$D' + C + B$$

Oppgave 3

Finns det bedre måter å finne et forenklet uttrykk fra diagrammet i oppgave 2 på?

Nei, utlesning av 1 ere basert på de grupperingsreglene vi har lært gir alltid et minimalt uttrykk

Ja, her kunne vi brukt XOR og kommet fram til $A' + D$

Ja, her kunne vi brukt XNOR og kommet fram til $A'D$

Ja, her kunne vi valgt alle X lik 1 og lest ut 0 ere (kun de 2 tomme rutene)

Nei, utlesning av 0 ere vil i dette tilfelle gi et like enkelt uttrykk

Oppgave 4

Hvilken boolsk ligning er ikke korrekt?

$$A + 1 = A$$

$$A \oplus B = AB' + A'B$$

$$A \cdot 1 = A$$

$$A + 0 = A$$

$$(A'A')' = A + A$$

Oppgave 5

Hvilket utsagn er galt

Utgangssignaler fra kombinatoriske kretser er ikke avhengig av framtidige inngangssignaler.

Utgangssignaler fra sekvensielle kretser er ikke avhengig av tidligere signaler.

Sekvensielle kretser må inneholde noe form for minne.

Sekvensielle kretser kobles ofte sammen med kombinatoriske kretser.

Kombinatorisk kretser utgjør ofte den største delen av en tilstands maskin.

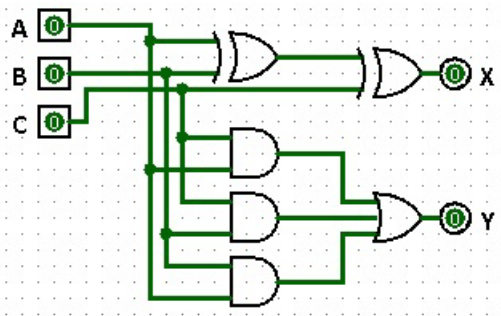
Oppgave 6

Hvilken krets må være sekvensiell

- XOR
- Multiplexer
- Decoder
- Rippelteller
- 4bits binær adder

Oppgave 7

Finn sannhetstabellen for kretsen under



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

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

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

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

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

Oppgave 8

Under ser vi en tilstandstabell for en tilstandsmaskin. "Q1 next , Q0 next" er neste tilstand, "Q1 current , Q0 current" er nåværende tilstand "Out1 , Out0" er utgangen. "Input" er inngangen og "x" betyr don't care.

Q1 current	Q0 current	Input	Q1 next	Q0 next	Out1	Out0
0	0	0	0	0	1	1
0	0	1	1	1	1	1
0	1	0	0	0	1	1
0	1	1	1	1	X	X
1	0	0	0	1	0	1
1	0	1	1	0	0	0
1	1	0	0	0	1	1
1	1	1	1	1	1	X

Reduser antall tilstander.

Q1 current	Q0 current	Input	Q1 next	Q0 next	Out1	Out0
0	1	0	1	1	1	1
0	1	1	1	0	X	0
1	1	0	0	1	1	1
1	1	1	0	1	1	X

Q1 current	Q0 current	Input	Q1 next	Q0 next	Out1	Out0
0	1	0	0	1	1	1
0	1	1	0	1	1	0
1	0	0	0	1	0	1
1	0	1	1	0	0	0

Q1 current	Q0 current	Input	Q1 next	Q0 next	Out1	Out0
0	1	0	1	1	1	1
0	1	1	1	1	X	0
1	0	0	0	1	0	1
1	0	1	1	0	0	0
1	1	0	1	1	1	1
1	1	1	1	1	1	X

Kan ikke reduseres

Q1 current	Q0 current	Input	Q1 next	Q0 next	Out1	Out0
0	1	0	0	1	1	1
0	1	1	0	1	1	1
1	0	0	0	1	0	1
1	0	1	1	0	0	0

Oppgave 9

Når man reduserer antall tilstander i en tilstandsmaskin vil de tilstandene vi fjerner alltid lede til brukte tilstander.

Riktig, på grunn av reduksjon av logikk får vi den samme oppførselen.

Galt, på grunn av reduksjon av logikk får vi ingen nye ubrukte tilstander.

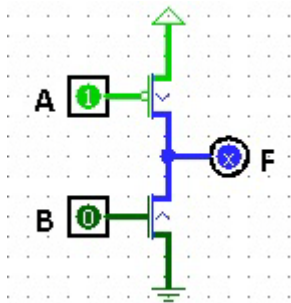
Galt, etter forenkling av logikken kan vi få tilstander som ikke leder videre til brukte tilstander.

Delvis riktig, de tilstandene vi fjerner får oppførsel identiske med de vi beholder.

Logisk sett fjerner vi egentlig ingen brukte tilstander, det er kun logikken som blir enklere.

Oppgave 10

Under ser vi en krets med 2 inngangssignaler og et utgangssignal, hvilket utsagn er galt.



Vi ser her en komplett inverter.

Vi har en problematisk krets siden tilstanden (A=0, B=1) får vi en ekvivalens til 2 lukkede brytere som kobler sammen Gnd og Vdd (kortslutning).

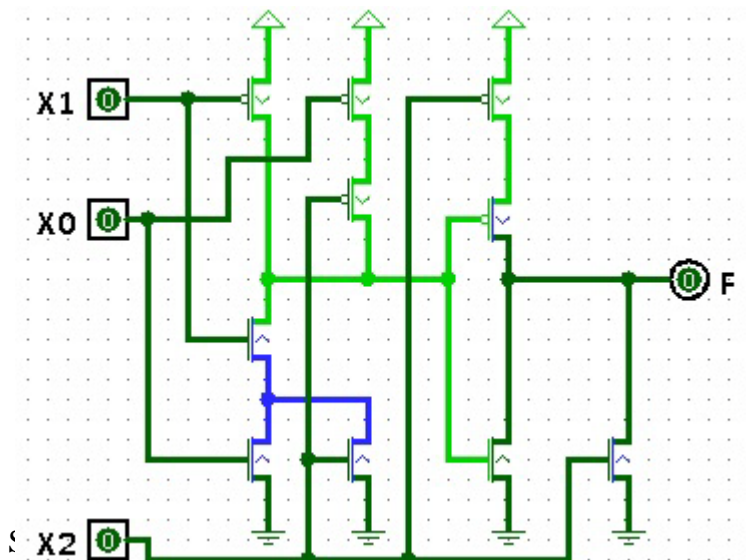
Vi har en problematisk krets siden tilstanden (A=1, B=0) får vi en ekvivalens til 2 åpne brytere som gir en udefinert spenning ut.

Denne kretsen er helt uvanlig.

Dette er en krets basert på 2 seriekoblede transistorer.

Oppgave 11

Finn det boolske uttrykket for F i kretsen under



$$F = X0$$

$$F = X0 + X1 \cdot X2'$$

$$F = X0 \cdot X1 \cdot X2'$$

$$F = X0 \cdot (X2' \oplus X1')$$

$$F = (X0 + X1)'$$

Oppgave 12

Desimal til binær konvertering. Hva er riktig?

$39_{(des)} = 100110_{(bin)}$

$39_{(des)} = 110111_{(bin)}$

$39_{(des)} = 10101_{(bin)}$

$39_{(des)} = 100111_{(bin)}$

$39_{(des)} = 10111_{(bin)}$

Oppgave 13

Hexadecimal til decimal konvertering. Hva er riktig?

$1.8_{(hex)} = 1.2_{(des)}$

$1.8_{(hex)} = 1.5_{(des)}$

$1.8_{(hex)} = 2.2_{(des)}$

$1.8_{(hex)} = 0.2_{(des)}$

$1.8_{(hex)} = -1.7_{(des)}$

Oppgave 14

Forenkle følgende uttrykk maksimalt

$a + b \cdot c + a \cdot b + c$

$a + b \cdot c + a \cdot b$

$c + a$

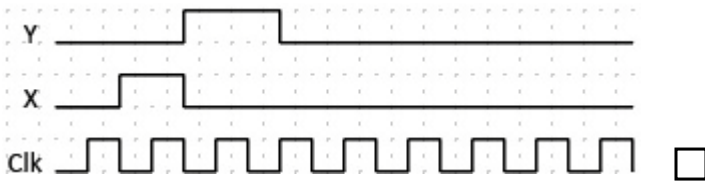
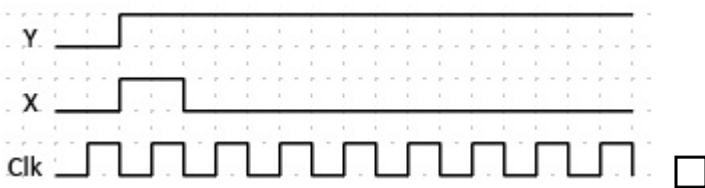
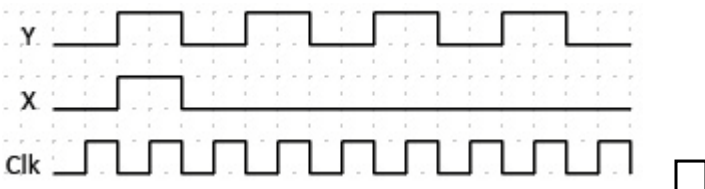
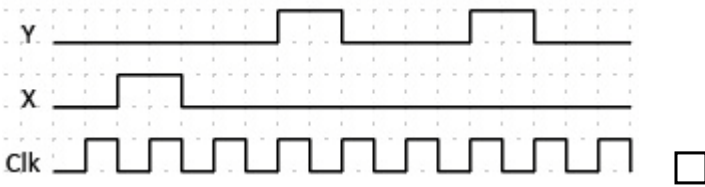
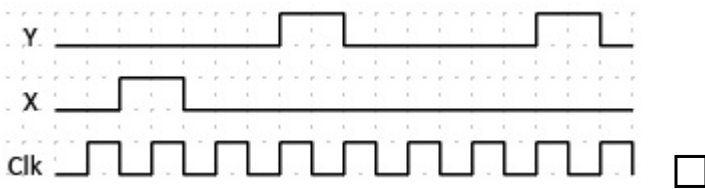
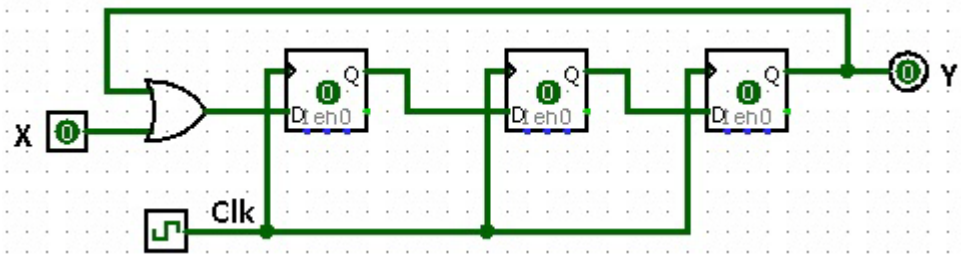
$a + b$

$c + a \cdot b'$

$b \cdot c + a \cdot b$

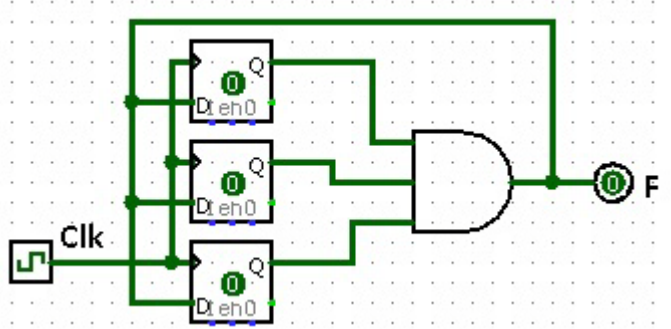
Oppgave 15

Kryss av for rett tidsforløp i kretsen under når Clk-signalet går som skissert. Anta at utgangene fra alle D flip-floppene er 0 i utgangspunktet.



Oppgave 16

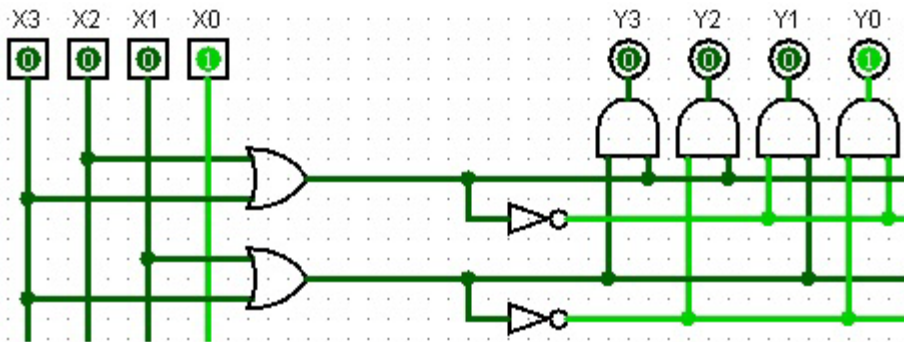
Hvilket utsagn er riktig for den sekvensielle kretsen under? Anta at utgangene fra de 2 øveste D flip-floppene er 1 i utgangspunktet, men at utgangen fra den nederste D flip-floppen er 0 i utgangspunktet.



- Kretsen er en kompakt 3-bits synkron binær teller.
- Kretsen er en kompakt 3-bits rippel teller.
- Kretsen deler klokkefrekvensen ned til halvparten.
- Kretsen er en meningsløs tullekrets.
- Kretsen deler klokkefrekvensen ned til en tredjedel.

Oppgave 17

Hvilket utsagn er riktig for den sekvensielle kretsen under?



- Kretsen består av en seriekobling av en MUX og en DEMUX.
- Kretsen består av en seriekobling av en encoder og en decoder.
- Kretsen består av en 4bits komparator med tilhørende enocder.
- Kretsen består av en 4bits komparator med tilhørende prioritets enocder.
- Kretsen består av en meget kompakt 4-bits rippeladder med tilhørende enkoder

Oppgave 18

For kretsen i oppgave 17 – hvis man setter flere inngangsverdier høye samtidig, hvilket utsagn er riktig?

Vi får summen av begge inngangene minus 1 ut

Vi får summen av inngangsverdiene ut.

Hadde vi bruket en prioritets enkoder ville i fått de samme verdiene ut.

Vi får ikke de samme verdiene ut, selv om vi brukte en prioritets enkoder.

Siden en av ledningene henger løst viker ikke kretsen helt som encoder / decoder.

Oppgave 19

Vi har det negative binære tallet 100101101 gitt på 2'er komplement form. Hvis man multipliserer det med (-1), hva får man da?

11010011

100101101

1100101101

011010010

111111111

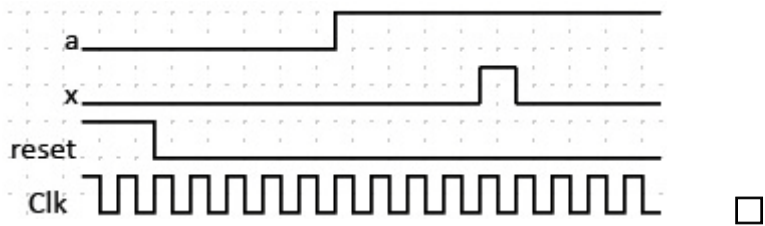
Oppgave 20

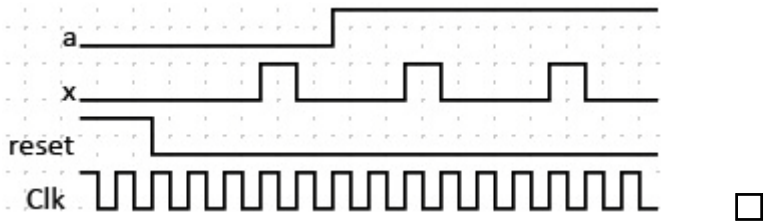
For VHDL koden under, hvilke tidsforløp vil være riktig:

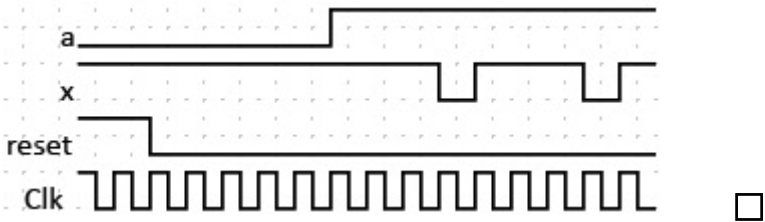
```
entity krets is
port(  a:                in std_logic;
       clock:           in std_logic;
       reset:           in std_logic;
       x:                out std_logic
);
end krets;

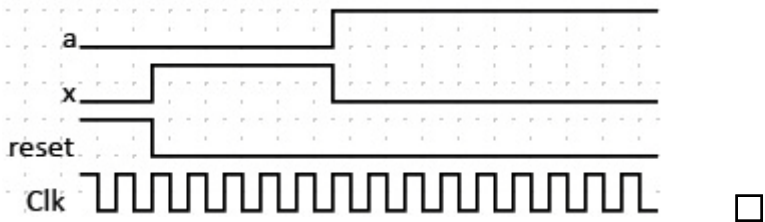
architecture F of krets is
    type V is (S0, S1, S2, S3);
    signal Sn : V;
    signal Sc : V;
begin
    process(clock, reset)
    begin
        if (reset='1') then
            Sc <= S0;
        elsif (clock'event and clock='1') then
            Sc <= Sn;
        end if;
    end process;

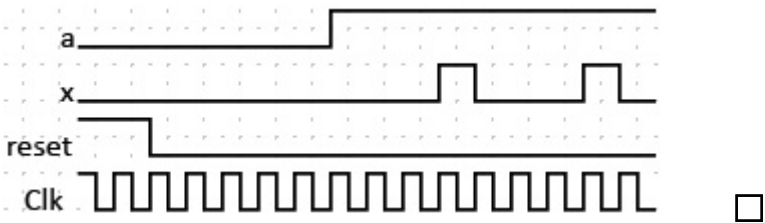
    process(Sc, a)
    begin
        case Sc is
            when S0 =>
                x <= '0';
                if a='0' then
                    Sn <= S0;
                elsif a='1' then
                    Sn <= S1;
                end if;
            when S1 =>
                x <= '0';
                if a='0' then
                    Sn <= S1;
                elsif a='1' then
                    Sn <= S2;
                end if;
            when S2 =>
                x <= '0';
                if a='0' then
                    Sn <= S2;
                elsif a='1' then
                    Sn <= S3;
                end if;
            when S3 =>
                x <= '1';
                if a='0' then
                    Sn <= S3;
                elsif a='1' then
                    Sn <= S0;
                end if;
            when others =>
                x <= '0';
                Sn <= S0;
        end case;
    end process;
end F;
```











Løsning:

1. **Kryss nr. 1**
2. **Kryss nr. 5**
3. **Kryss nr. 4/5.** Her snek det seg inn en mulig måte å tolke teksten på som medfører at vi gir riktig svar for både kryss nr.4 og kryss nr.5
4. **Kryss nr. 1**
5. **Kryss nr. 2**
6. **Kryss nr. 4**
7. **Kryss nr. 4**
8. **Kryss nr. 5** – reduserer ved å først fjerne tilstand 0 får:
Fjerner så tilstand 1

Q1 current	Q0 current	Input	Q1 next	Q0 next	Out1	Out0
0	1	0	1	1	1	1
0	1	1	1	1	X	X
1	0	0	0	1	0	1
1	0	1	1	0	0	0
1	1	0	1	1	1	1
1	1	1	1	1	1	1

9. **Kryss nr. 3.** Dette var den vanskelige nøtten. Her må man virkelig tenke seg om. *Kryss nr.1:* Tilstandsmaskinen får samme oppførsel sett fra inngang til utgang – ja, men dette faktum forhindrer ikke evt. nye ubrukte tilstandene å kunne ende opp uten overgang til brukte tilstander. *Kryss nr.2* – Siden tilstander defineres av mulige utgangsverdier på de benyttede Flip-Floppene vil man alltid ha 2^N antall tilstander der N er antall Flip-Flopper. Dermed vil man kunne få flere ubrukte tilstander når man reduserer antall brukte tilstander. *Kryss nr.4* – Ikke nødvendigvis, husk at vi forenkler logikken utover den allerede foretatte forenklingen som beholder de samme kombinatoriske inngangs/utgangs funksjonene - dermed kan man ikke garantere samme oppførsel når det gjelder hopp mellom tilstander. Hele poenget med å forenkle logikken ved reduksjon av antall tilstander er at vi kan tåle at maskinen hopper annerledes i mellom tilstandene – uten at det påvirker inngangs/utgangssignalene til selve tilstandsmaskinen. *Kryss nr.5* – Jo, når vi sier at vi reduserer antall tilstander mener vi at vi reduserer (får færre) brukte/spesifiserte tilstander.
10. **Kryss nr. 1.** En inverter er en krets med EN inngang og en utgang. Dette er en krets som ligner, men den har 2 inngangssignaler (A og B) som kan settes uavhengig. Sammenlignet med en inverter mangler denne kretsen ledningen mellom gate terminalene på N- og P-MOS transistorene. Uten denne ledningen, og med frie terminaltilkoblinger til disse to, slik som vist i figuren, får vi en meget tvilsom krets som kan kortsluttes (ødelegge selve mikrochippen) hvis man setter A=0 og B=1 samtidig. I oppgaven var det dessuten påstått at vi i figuren har en *komplett* inverter.
11. **Kryss nr. 3.** Dette kan se ut som en vanskelig oppgave, men egentlig er den lett – vi har ganske enkelt en seriekobling av de to kretsene som er vist på forelesningsnotatene med tittel ”CMOS kretser” $F = (C(A+B))'$ og ”CMOS NOR-krets” $ut = (A+B)'$.
12. **Kryss nr. 4.**
13. **Kryss nr. 2.**
14. **Kryss nr. 2.**
15. **Kryss nr. 2.**
16. **Kryss nr. 4.** Grunnet 3 inputs AND får vi her alltid ut 0.
17. **Kryss nr. 2.** Litt tankevekkende oppgave? X0 henger ”løst” – dette er riktig for en encoder.
18. **Kryss nr. 4.** Vi får alltid de samme verdiene ut hvis vi KUN har EN av inngangene høye av gangen. Selv en prioritets encoder vil her ikke kunne ”encode” mer en 4 inngangskombinasjoner, slik at ved flere ”1”ere inn samtidig får vi ikke nødvendigvis det samme ut.

19. **Kryss nr. 1.** Første null foran det resulterende positive tallet er ikke vist.
20. **Kryss nr. 5.** Her har vi en tilstandsmaskin i VHDL. Her må man være nøye når man teller klokke perioder.

Generell kommentar til eksamen 2011:

Oppgavene gitt dette året er (som indikert på forelesning) noe vanskeligere enn gitt tidligere år. Dette medfører **ikke** at gjennomsnittskarakteren dette året av denne grunn settes dårligere enn gjennomsnittskarakteren tidligere år. Det betyr heller ikke at kravet til å bestå settes høyere enn tidligere år. Fordelen med å ha en "passe" vanskelig eksamen er at den gir en størst mulig spredning i resultat. Dermed blir det lettere å differensiere mellom de forskjellige karakterene, noe som forhåpentligvis gir en mer rettferdig karaktersetting.