

UNIVERSITETET I OSLO

Det matematisk-naturvitenskapelige fakultet

Eksamen i: INF1400 – Digital teknologi
Eksamensdag: 5. desember 2005
Tid for eksamen: 9-12
Vedlegg: Ingen
Tillatte hjelpemidler: Alle trykte og skriftlige samt kalkulator
Oppgavesettet er på 2 sider

Eksamen med løsningsforslag. Løsningsforslaget er for noen deloppgaver bare en skisse, og har man laget er mer utfyllende forklaring er det bra.

Oppgave 1 – Karnaughdiagram (vekt 15%)

1a Bruk Karnaughdiagram for å forenkle følgende funksjon

$$Y = A'B'C'D' + A'B'CD' + ABCD'$$

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

$$Y = A'B'D' + ABCD'$$

1b Anta følgende: hvis både A=1 og B=0 og C=1 samtidig, kan ikke D være lik 0. Vis hvordan denne antagelsen kan gi et enklere uttrykk for Y.

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

Inngangskombinasjonen (A=1, B=0, C=1, D=0) kan nå ikke inntreffe, og hva Y blir for denne inngangskombinasjonen betyr da ingenting - don't care X. I utgangspunktet skulle Y være 0 for denne inngangskombinasjonen i følge ligningen i 1a, men siden denne

inngangskombinasjonen ikke vil inntreffe velger vi å sette **X til 1**, dvs. vi lar Y blir 1 for denne inngangskombinasjonen for å få større (blå) gruppe. Y kan da skrives som

$$Y = A'B'D' + ACD'$$

Oppgave 2 – Boolsk algebra (vekt 14%)

2a Hvis hvordan man kan forenkle følgende uttrykk ved å bruke den distributive loven

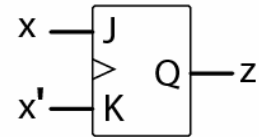
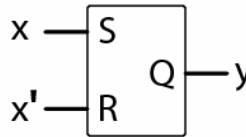
$$\begin{aligned} F &= A + (B \cdot A') \\ F &= (A + B)(A + A') \\ F &= (A + B)(1) \\ F &= A + B \end{aligned}$$

2b Bruk DeMorgan's teorem for å forenkle følgende uttrykk (redukere antall inverteringer)

$$\begin{aligned} G &= (A' + (B'C'))' \\ G &= A''(B'C')' \\ G &= A(B'' + C'') \\ G &= A(B + C) \end{aligned}$$

Oppgave 3 – Sekvensiell logikk (vekt 14%)

I figurene til høyre er det vist to komponenter. Beskriv forskjeller / likheter i virkemåte. Hva gjør de to kretsene når x varierer (anta at nødvendige aktive klokkesignaler er koblet til).



SR latch – y blir satt hvis $x=1$ ($x'=0$)
y blir resatt hvis $x=0$ ($x'=1$)

dette skjer i det øyeblikket x forandres (+ en liten portforsinkelse)

- Kretsen slipper i gjennom x direkte (+ en liten portforsinkelse)

JK flipflop – z blir satt hvis $x=1$ ($x'=0$) (likt med forrige)

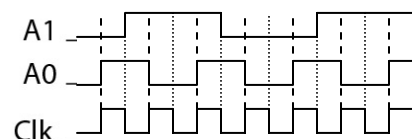
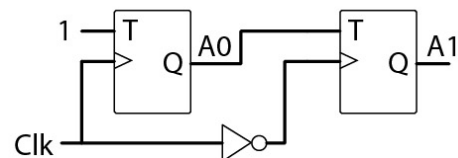
z blir resatt hvis $x=0$ ($x'=1$) (likt med forrige)

men dette skjer kun i det øyeblikket klokkesignalet går HØY (+ en liten portforsinkelse)

- Kretsen slipper i gjennom x på positiv klokkeflanke (D flip-flop)

Oppgave 4 – Sekvensielle systemer (vekt 14%)

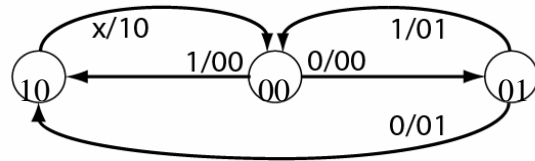
I figuren til høyre er det vist en sekvensiell krets. Skisser hvordan signalene A0 og A1 forandrer verdi i



løper av 6 klokke (Clk) perioder. Anta at A0 og A1 begge er 0 i utgangspunktet

Oppgave 5 – Tilstandsmaskin (vekt 16%)

5a Til høyre er det vist et tilstandsdiagram for en tilstandsmaskin. Finn ut hvor mange innganger/utganger denne maskinen har, velg tilstandskoder og tegn opp en tilstandstabell basert på diagrammet. Hva betyr "x" i figuren?



Anta at systemet alltid vil starte opp i tilstanden helt til venstre i diagrammet.

1 inngang / 2 utganger, x betyr her dont'care fordi i denne tilstanden følger man denne pilen uansett verdi på inngangen. Tilstandskoder kan velges på flere måter. Enklest løsning vist under

Nåværende tilstand		Inngang Inn	Neste tilstand		Utganger	
Q1	Q0		Q1 (D1)	Q0 (D0)	Y1	Y0
0	0	0	0	1	0	0
0	0	1	1	0	0	0
0	1	0	1	0	0	1
0	1	1	0	0	0	1
1	0	0	0	0	1	0
1	0	1	0	0	1	0
1	1	0	x	x	x	x
1	1	1	x	x	x	x

Systemet har en ubrukt tilstand (11), men vil ikke havne her under oppstart

5b Tegn opp denne tilstandsmaskinen basert på D flip-flop'er og nødvendige porter.

D1	Inn	
	0	1
Q1Q0	00	1
	01	1
	11	x
	10	

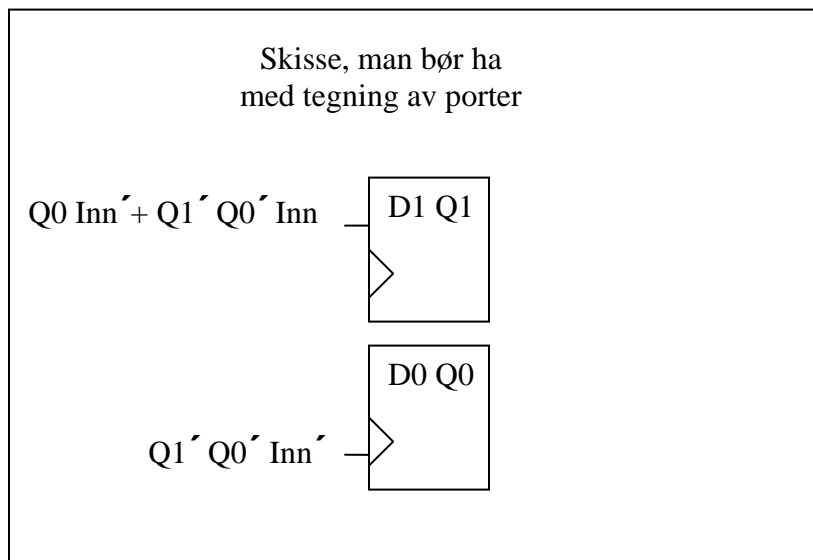
D0	Inn	
	0	1
Q1Q0	00	1
	01	
	11	x
	10	

$$D1 = Q0 \text{ Inn}' + Q1' Q0' \text{ Inn}$$

$$D0 = Q1' Q0' \text{ Inn}' , \text{ (ingen forenkling)}$$

$$Y1 = Q1$$

$$Y0 = Q0$$



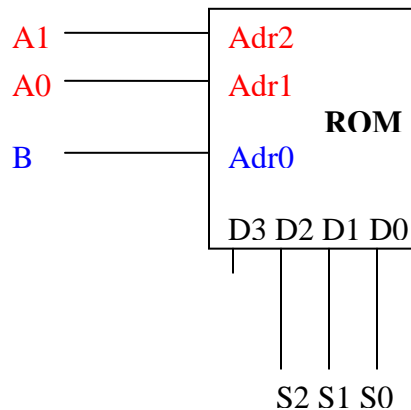
Oppgave 6 – Minne/adder (vekt 13%)

Vi ønsker å lage en binær adder som kan addere to binære tall A og B. Tall A er representert ved 2-bit, mens tall B er bare representert ved 1-bit. Adderen skal ikke ha mente (carry) inn og trenger 3-bit ut. Vi har en ROM minnekrets som kan lagre 8 stk. 4-bits ord. Adresseringen av ROM'en foregår på vanlig måte (binært). Vi vil bruke denne ROM-kretsen til å implementere hele adderen, og vi vil ikke bruke noe ekstra logikk utenom denne ROM-kretsen. Vis hvordan signalene A og B må kobles til ROM-kretsen. Spesifiser hva innholdet i ROM-kretsen (innholdet i hver minnecelle) må være for at ROM-kretsen skal summere A og B riktig.

ord nr.0	?	?	?	?
ord nr.1	?	?	?	?
ord nr.2	?	?	?	?
ord nr.3	?	?	?	?
ord nr.4	?	?	?	?
ord nr.5	?	?	?	?
ord nr.6	?	?	?	?
ord nr.7	?	?	?	?

Lar for eksempel A gå til bit2 og bit1 i adresseinngangen, lar B gå til bit0 i adresseinngangen. Sum blir bit D2, D1, D0 i datautgangen fra minne. Bit D3 har vi ikke bruk for.

Innhold i minne				
Adr / ord nr	D3	D2	D1	D0
00 0	x	0	0	0
00 1	x	0	0	1
01 0	x	0	0	1
01 1	x	0	1	0
10 0	x	0	1	0
10 1	x	0	1	1
11 0	x	0	1	1
11 1	x	1	0	0



Oppgave 7 – VHDL (vekt 14%)

7a VHDL koden til høyre beskriver en mye brukt krets. Forklar hva denne kretsen gjør.

Koden beskriver en **2-4 dekode**r

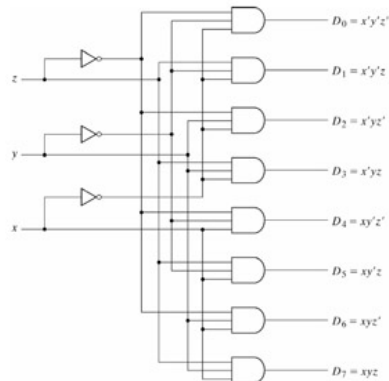
7b Skisser grafisk med en tegning av porter og ledninger hvordan denne kretsen kan kobles opp.

NB: figuren under viser en 3x8 decoder

Dekoder

Dekoder - tar inn et binært ord, gir ut alle mintermer

Eksempel: 3bit inn / 8bit ut



```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity krets is
    Port ( A : in std_logic_vector(1 downto 0);
          WL : out std_logic_vector(3 downto 0));
end krets;

architecture Behavioral of krets is
begin
    process(A)
    begin
        case A is
            when "00" =>
                WL <= "0001";
            when "01" =>
                WL <= "0010";
            when "10" =>
                WL <= "0100";
            when others =>
                WL <= "1000";
            end case;
        end process;
    end Behavioral;
```