

Løsningsforslag

UNIVERSITETET I OSLO

Det matematisk-naturvitenskapelige fakultet

Eksamen i:	INF1400
Eksamensdag:	Fredag 3. desember
Tid for eksamen:	kl. 14:30-18:30 (4 timer).
Oppgavesettet er på ... side(r)	7 sider
Vedlegg:	ingen
Tillatte hjelpemidler:	Alle skriftelige

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

NB: Alle oppgaver skal føres inn i oppgavesettet. Bruk blyant før du eventuelt fører inn ditt endelige svar. Besvarelsene må utføres med en tydelig strek!

Oppgavene 1 til 7 er flervalgsoppgaver (multiple-choice). For disse oppgavene gjelder følgende:

- Oppgaven skal kun besvares med ETT kryss. Alternativt ingen kryss hvis du ikke vet svaret.
- Riktig avkryssing gir full uttelling.
- Ingen kryss gir 0 poeng.
- Ved gal avkryssing trekkes $\frac{1}{4}$ av full pott.
- Flere kryss enn ett, gir 0 poeng.

Oppgave 1 - Digital representasjon (5%)

Gjør om desimaltallet -113 til 8 bits binærtall på 2'ers komplement form

- 10001111 X
- 01110001
- 10001110
- 10010000
- 01101111

Oppgave 2 - Boolsk algebra (7,5%)

Forenkle følgende uttrykk maksimalt

$$Q = ((C'(AB + A'B'))' (B(AC + A'C')))')'$$

Ingen av alternativene er riktige. Riktig svar er $AB + A'C'$

- $C' + B$
- $A'B' + C$
- $BC' + A$
- $B'C + AC' + A'B$
- $AB + C'$

Oppgave 3 - Boolsk algebra (7,5%)

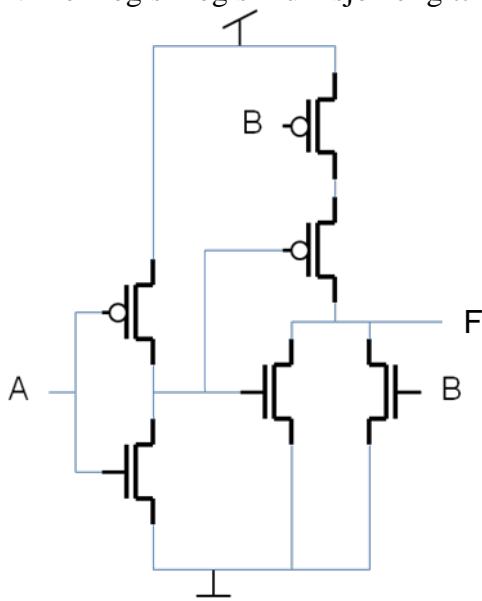
Hva er uttrykket under i minterm-form?

$$Q = BC'D' + A'B'C + ABC + ACD'$$

- $F(A,B,C,D) = \sum(2,3,4,8,10,11,15)$
- $F(A,B,C,D) = \sum(2,3,4,10,12,14,15)$ X
- $F(A,B,C,D) = \sum(2,3,4,8,10,11,14)$
- $F(A,B,C,D) = \sum(0,1,5,6,7,8,9,11,13)$
- $F(A,B,C,D) = \sum(0,1,5,6,7,9,12,13,14,15)$

Oppgave 4 - Mikroelektronikk (7,5%)

Hvilken logisk funksjon er gitt med CMOS transistorer under?



- $F = A' + B$
- $F = A'B$
- $F = A \otimes B$
- $F = (A' + B)'$ X
- $F = (AB)'$

Oppgave 5 - Karnaughdiagram (7,5%)

Forenkle uttrykket gitt i sannhetstabellen maksimalt. X er en don't care betingelse.

A	B	C	D	F(A,B,C,D)
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	X
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

- $B'D' + A'C + AC'D'$
- $A'B'C + B'C'D' + A'CD$
- $B'D' + A'CD + AC'D'$ X
- $AC' + AC'D' + B'C'D'$
- $BD + AC'D' + A'CD$

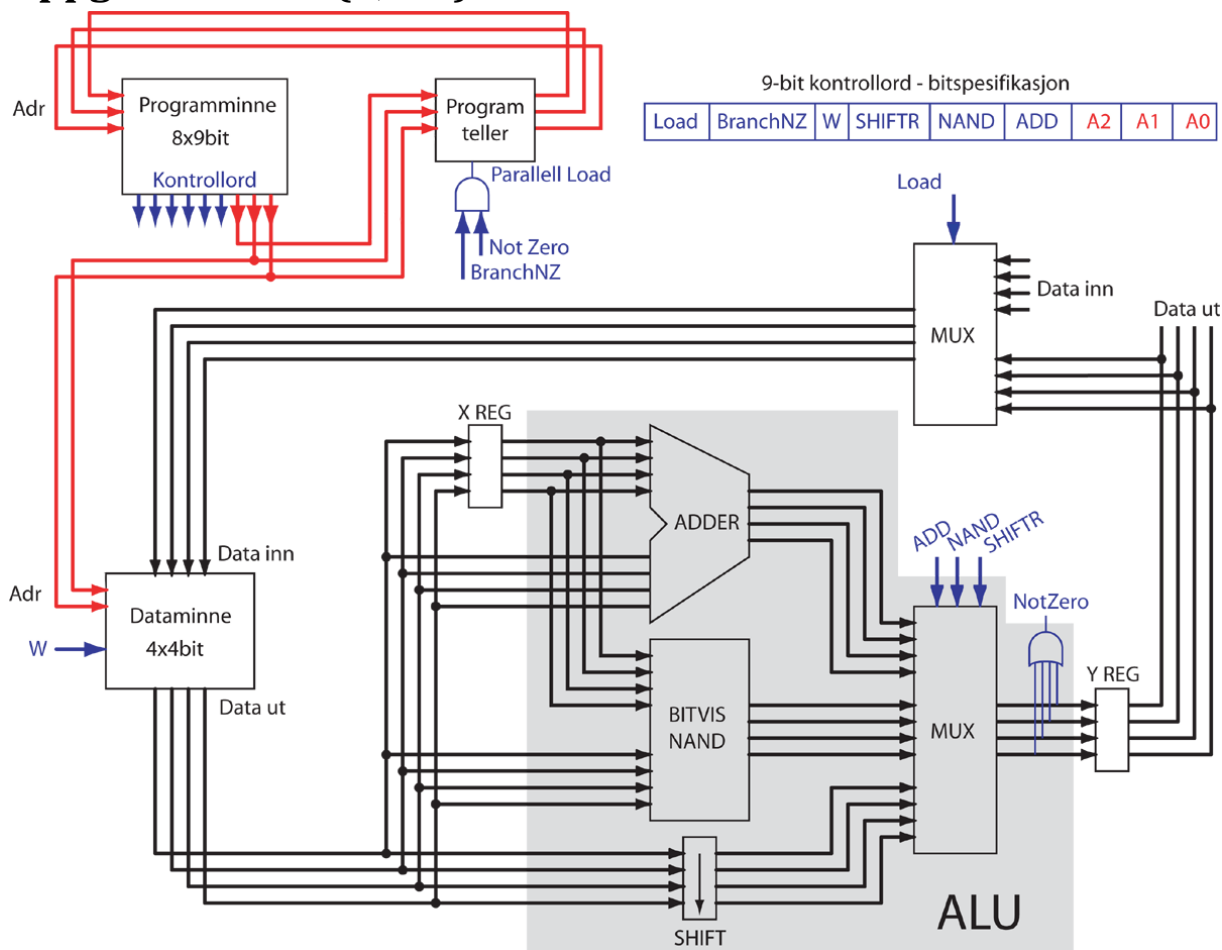
Oppgave 6 - Karnaughdiagram (7,5%)

Du skal lage en kombinatorisk krets som skal brukes som en alarm for en heis. Kretsen skal gi et høy signal ut hvis det er en feil, slik at de som har ansvaret for heisen blir informert om at det er en feil. Du har fire inngangssignaler. D er høy hvis døren til heisen er åpen, E er høy hvis heisen er i en etasje (dvs. ikke mellom to etasjer), N er høy hvis nødknappen inne i heisen har blitt trykt og O er høy hvis det er overvekt. Utgangssignalet er F som er høy hvis det er en feil.

Utgangssignalet F skal være høy hvis nødknappen har blitt trykt inn. F skal også være høy hvis døren er åpen og man er i mellom to etasjer. Det siste tilfellet F skal være høy er når det er overvekt og man ikke kan gå ut av heisen. Man kan gå ut av heisen bare når heisen er i en etasje og døren er åpen. Hvilket uttrykk under minimaliserer kretsen beskrevet ovenfor?

- $N + DE' + D'O$ X
- $NO' + D + N'E$
- $N + DE'N' + D'EO$
- $N + E'O' + D'O$
- $N + DE' + OD'E'$

Oppgave 7 - CPU (7,5%)



Ovenfor er det gitt en figur av CPU'en som ble forelest. Under er det gitt hva som er lagret i dataminne og tre programlinjer som utføres. Alle styresignalene er aktivt høye. Når signalet Load er høyt inn i MUX'en velges Data inn signalet. Hva gjør dette programmet?

Lagret i dataminne:

00: 0101
01: 1010
10: 0001
11: 1111

Program som utføres:

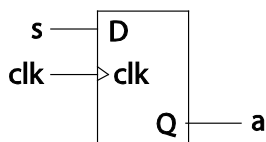
1: 000 000 001
2: 000 001 010
3: 001 000 000

- Leser inn fra IO og lagrer dette på adresse 01 og 10 i dataminnet
- Tar bitwise NAND på verdiene i på adresse 01 og 10 i dataminne og lagrer dette på adresse 00 i dataminne
- Sjekker om verdiene på adresse 01 og 10 i dataminne er like og foretar et hopp til adresse 000 i programminnet fordi de er ulike
- Legger sammen verdiene på adresse 01 og 10 i dataminnet og lagrer dette på adresse 00 i dataminnet X
- Shifter alle bit'ene i verdiene på adresse 01 og 10 i dataminnet en plass til høyre og lagrer dette på samme plass i dataminnet

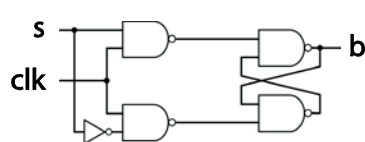
Oppgave 8 - Synkron logikk (18%)

Hver av de tre kretsene under tar inn klokkesignalet clk og inngangssignalet s . Anta at a , b og c har startverdien 0 mens m har 1. Tegn inn manglende tidsforløpet for signalene a , b , m og c . Bruk blyant før dere eventuelt fører inn med penn.

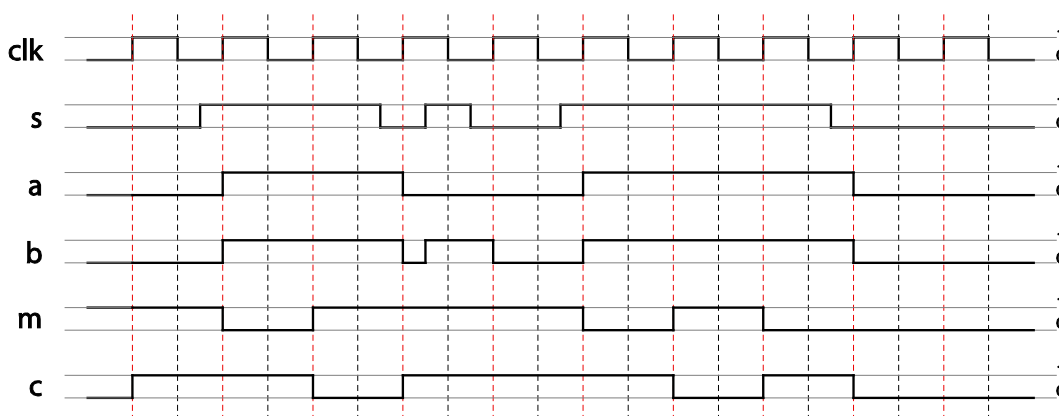
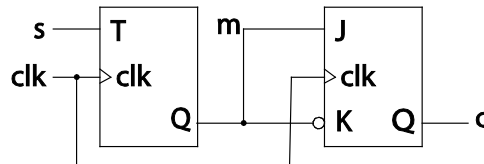
krets 1 (5%)



krets 2 (5%) D-latch



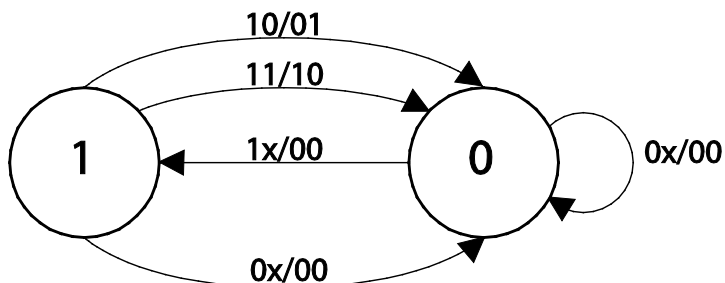
krets 3 (8%) Litt vanskelig



Oppgave 9 - Tilstandsmaskiner (16%)

a) 8%

Under ser dere et tilstandsdiagram til en tilstandsmaskin. Fyll inn tilstandstabell under slik at den gjenspeiler oppførselen til tilstandsdiagrammet. Her er ab innganger og yz utganger.

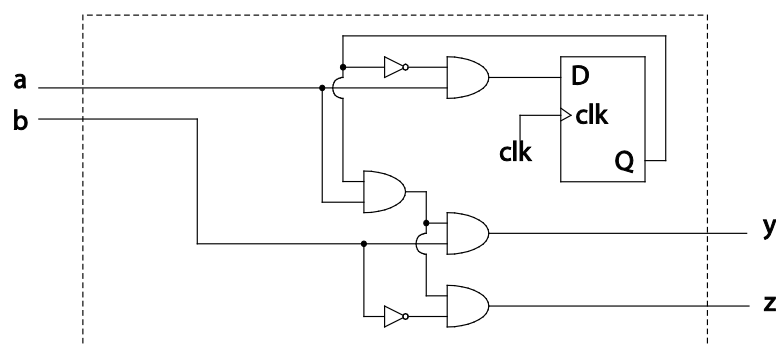
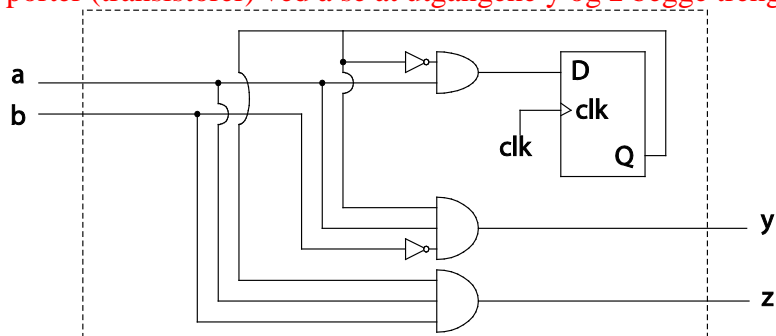


Q_t	a	b	Q_{t+1}	y	z
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	1	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	0	0	0
1	1	0	0	0	1
1	1	1	0	1	0

b) 8%

Tegn inn det logiske skjemaet som gjenspeiler oppførselen til tilstandsmaskinen over. Dere kan bruke vanlige kretser direkte og flerinnngangsporter (for eksempel 4 inngangs AND).

Kan løses på flere måter. Den første er en direkte implementasjon. Men vi kan også redusere antall porter (transistorer) ved å se at utgangene y og z begge trenger AND(Q,a).



Oppgave 10 - VHDL (16%)

a) 8%

VHDL-koden under beskriver en kjent krets. Hva kalles den? Her betyr *inout* at porten både kan brukes som inngangs- og utgangssignal.

```

1  library ieee;
2  use ieee.std_logic_1164.all;
3
4  Entity krets is port (
5      A : in STD_LOGIC;
6      B : in STD_LOGIC;
7      C : inout STD_LOGIC;
8      D : inout STD_LOGIC
9  );
10 End krets;
11
12 Architecture behavioral of krets is
13 Begin
14     Process(A,B,C,D)
15     Begin
16         C <= D nor A;
17         D <= C nor B;
18     End process;
19 End behavioral;

```

Svar: SR-latch laget av to NOR porter.

b) 8%

VHDL koden under beskriver en tilstandsmaskin. Tegn tilstandsdiagrammet som gjenpseiler VHDL-koden. Husk å påføre inn- og utgangsverdier.

```

1  library IEEE;
2  use IEEE.std_logic_1164.all;
3
4  entity fsm is
5      port ( clk, reset, a : IN std_logic;
6              outp : OUT std_logic);
7  end entity;
8  architecture behaviour of fsm is
9      type state_type is (s1,s2,s3,s4);
10     signal state: state_type ;
11 begin
12     process (clk,reset)
13     begin
14         if (reset ='1') then
15             state <=s1; outp<='1';
16         elsif (clk='1' and clk'event) then
17             case state is
18                 when s1 =>  if a='1' then state <= s2;
19                             else          state <= s3;
20                             end if;
21                             outp <= '1';
22                 when s2 => state <= s4; outp <= '1';
23                 when s3 => state <= s4; outp <= '0';
24                 when s4 => state <= s1; outp <= '0';
25             end case;
26         end if;
27     end process;
28 end behaviour;

```

