

# UNIVERSITETET I OSLO

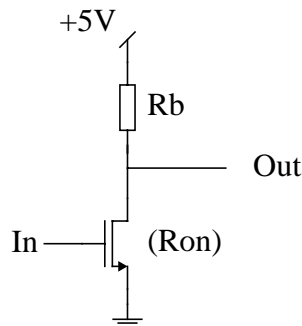
## Det matematisk-naturvitenskapelige fakultet

Eksamen i                    IN 240 — Digital systemkonstruksjon  
Eksamensdag:            12. desember 1995  
Tid for eksamen:        9.00 – 15.00  
Oppgavesettet er på 7 sider.  
Vedlegg:                    Ingen  
Tillatte hjelpemidler:  Ingen

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

### 1 (vekt 10%)

En enkel NMOS inverter kan tegnes som i figuren nedenfor.



Vi gjør en statisk betraktning og ønsker en støymargin på 1V. Vi forenkler og antar at NMOS-transistoren har en “på”- motstand ( $R_{on}$ ) når inngangen er “høy”. Videre antar vi at det ikke trekkes noe strøm eksternt på utgangen (ingen ohmsk belastning). Når inngangen er “lav”, kan vi anta at  $R_{on}$  er uendelig stor.

(Fortsettes side 2.)

**1-a**

Hvis  $R_b = 1$  kohm, hvilken verdi må  $R_{on}$  ikke overskride for at støymarginkravet skal overholdes.

**1-b**

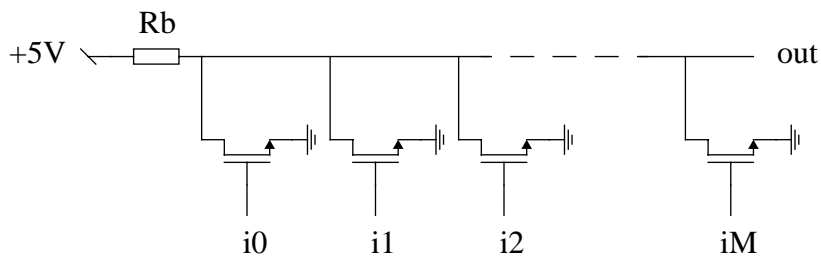
Eksemplet i a) tok for seg støymargin ved “lav” utgang. Er støymargin noe problem i tilfellet “høy” utgang? Hvorfor/hvorfor ikke?

**1-c**

Forklar kort hvordan man kan betrakte problemet på en annen måte enn å anta at NMOS-transistoren har en “på”- motstand ved “høy” inngang.

**1-d**

Strukturen som er vist i forrige figur kan utvides som vist nedenfor.



Denne koblingen kan brukes som en mye brukt primitiv (multi-input). Hvilken?

**1-e**

Ta utgangspunkt i figuren ovenfor.

Dersom en antar at kun en av inngangene ( $i_0 - i_M$ ) er “på” av gangen, kan konseptet bli brukt som en 1-bits ROM. Lag en 3-bits ROM ved hjelp av dette prinsippet. ROM'en skal ha 8 innganger og skal programmeres med følgende 3-bits ord:

(Fortsettes side 3.)

	W2	W1	W0
i0	1	0	1
i1	1	1	0
i2	0	1	1
i3	0	0	1
i4	1	0	0
i5	1	1	1
i6	1	0	0
i7	0	1	0

## 1-f

Den samme ROM'en med et bestemt innhold kan i prinsippet brukes til noe annet. Hva?

## 2

I mange parkeringshus får man et lite kort med magnetstripe når man kjører bilen inn. På kortet er tidspunktet for innkjøringen registrert. Før man henter bilen igjen, må kortet puttes i en betalingsautomat som leser av magnetstripes og beregner hvor mye man skylder for parkeringstiden. Etter at beløpet er betalt, blir betalingen registrert på magnetstripes og man får kortet tilbake. Når man kjører ut av parkeringshuset, må man putte kortet i en ny automat som kontrollerer parkeringsavgiften og åpner utkjøringsbommen dersom korrekt avgift er betalt. Det er mulig å trykke på en knapp for å få igjen selve kortet for de som måtte trenge kortet som kvittering.

Denne oppgaven går ut på å konstruere en tilstandsmaskin og tilhørende logikk for å styre automaten ved utkjøringsbommen. Nedenfor kommer først en beskrivelse av systemet slik det skal fungere, deretter en beskrivelse av grenseflaten mot øvrig elektronikk i systemet, og til slutt noen krav til gjennomføringen av to deloppgaver.

Kunden putter kortet i en kortleser som finner ut om korrekt parkeringstid er betalt. Dersom parkeringstiden ikke er betalt, returneres kortet til kunden. Dersom parkeringstiden er betalt, åpnes bommen. Når bilen har kjørt forbi bommen, lukkes den. I tiden fra bommen er åpen til bilen har kjørt forbi, kan kunden trykke på en knapp for å få parkeringskortet tilbake. Dersom det ikke skjer, kaster automaten parkeringskortet i en innvendig boks når bommen lukkes.

Tilstandsmaskinen som kontrollerer systemet, har grenseflater mot tre elektroniske enheter: bomåpneren, utkjøringsføleren og kortleseren. Disse periferenhetene er ferdig konstruerte og omfattes ikke av denne oppgaven. I konstruksjonen av tilstandsmaskinen må du imidlertid ta hensyn til kommuni-

(Fortsettes side 4.)

kasjonsgrensesnittet mot periferenhetene. Ingen av signalene har tri-state. Kommunikasjonen med bomåpneren består bare av ett signal, og bommen holdes åpen så lenge linjen har logisk verdi 1. Utkjøringsføleren gir fra seg ett signal med logisk verdi 1 så lenge bilen passerer bomåpningen. Kommunikasjonen med kortleseren er vanskeligere. Kortleseren gir fra seg 3 signaler: kort er satt i leseren, korrekt avgift er betalt, og returknappen er trykket på. De to første signalene settes til logisk verdi 1 samtidig dersom korrekt avgift er betalt, og de holdes aktive så lenge kortet er i leseren. Dersom betalingen ikke er korrekt, er signalet for korrekt avgift logisk 0. Signalet fra returknappen er logisk 1 så lenge noen holder knappen inne. Kortleseren mottar 2 signaler fra tilstandsmaskinen: returner kortet, og kast kortet. Disse signalene må ha en varighet på minst 70 nanosekunder for at kortleseren skal utføre kommandoen. Mekanikken i kortleseren bruker naturligvis noe tid på å fjerne kortet, og det vil derfor også gå noe tid før signalene om at kort står i leseren og at avgift eventuelt er betalt, blir logisk 0.

Oppgaven har to deloppgaver som griper inn i hverandre. Sett deg inn i begge deloppgavene før du begynner på løsningen. Forklar omhyggelig ideene og prinsippene bak løsningene dine. Dersom du finner det vanskelig å løse deler av oppgavene, kan du selv definere betingelser som du finner nødvendige for å komme videre. Forklar disse nøye.

## **2-a (vekt 15%)**

I beskrivelsen av grenseflaten mellom tilstandsmaskinen og de tre periferenhetene bomåpneren, utkjøringsføleren og kortleseren framgår det at kommunikasjonen er asynkron (det er ingen felles klokke) og at impulsenes varighet kan være svært varierende. Dette kan både skape problem for og stille krav til tilstandsmaskinens oppførsel. Diskuter problemets grad og natur for hvert enkelt av signalene som tilstandsmaskinen skal motta og avgi. Forklar hvordan du vil løse problemene, konstruer logikken og tegn kretsskjema.

## **2-b (vekt 35%)**

Konstruer tilstandsmaskinen som utfører den beskrevne funksjonen. Maskinen skal være av Mealy-type med positivt kant-triggete JK flip-flop. Systemklokken har en frekvens på 10 MHz. Lag ASM-diagram og tilstandstabell med utganger. Utled og forenklingene for J- og K-inngangene. Forenklingene for systemets utganger (kan puffes dersom ligningene blir kompliserte). Tegn kretsskjema. Dersom den kombinatoriske logikken i deler av systemet blir komplisert, kan du symbolisere den med bokser slik at kretsskjemaet blir oversiktlig. Skriv i så tilfelle den tilhørende Booleske ligningen på hver boks.

*(Fortsettes side 5.)*

### 3 (vekt 30%)

#### 3-a

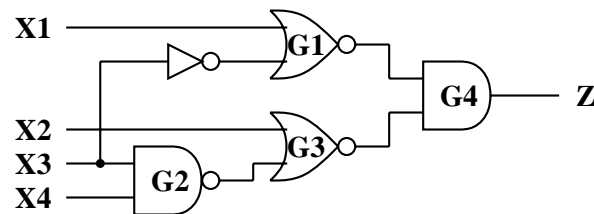
Hva er “exhaustive” test i forbindelse med testing av kombinatoriske kretser. Gi et eksempel. Hva begrenser bruk av “exhaustive” testing?

#### 3-b

Hvordan beregnes feildekkingsgrad?

#### 3-c

Ta utgangspunkt i kretsen under.



Bestem ved hjelp av “kritisk-vei analyse” hvilke av testvektorene  $X_1X_2X_3X_4 = \{0111, 1111, 1101, 1011\}$  som detekterer feilen  $X_1$  s-a-0. Vis framgangsmåten steg for steg.

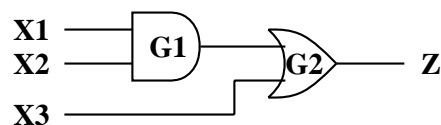
#### 3-d

Ta utgangspunkt i kretsen vist ovenfor.

Bruk metoden “boolsk differanse” til å beregne testvektorer for hver av feilene  $X_3$  s-a-0 og utgangen av  $G_2$  s-a-1. Vis framgangsmåten.

#### 3-e

Ta utgangspunkt i figuren under.



Sett opp komplett feilmatrix for kretsen.

(Fortsettes side 6.)

**3-f**

Ta utgangspunkt i figuren over.

Identifiser alle grupper av ekvivalente feil, alle dominerende og dominerte feil, samt alle essensielle testvektorer. Finn et minimalt sett med testvektorer som detekterer alle enkle låst-til feil (SSF).

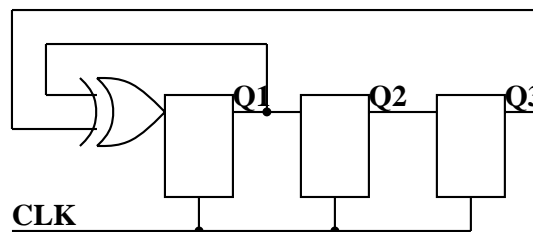
**3-g**

Ta utgangspunkt i figuren over og SCOAP-algoritmen.

Finn kombinatorisk 0- og 1-kontrollerbarhet samt kombinatorisk observerbarhet for samtlige noder i kretsen.

**3-h**

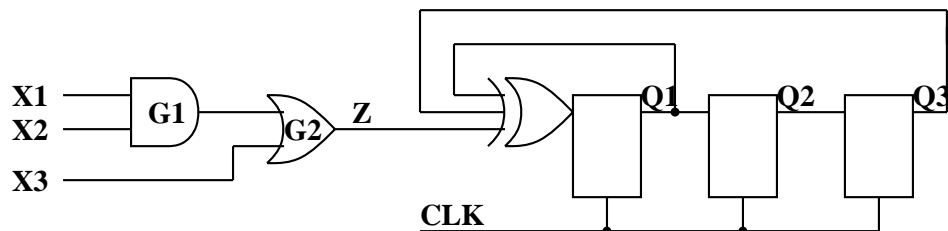
Ta utgangspunkt i kretsen under.



Bestem sekvensen og om registeret er av typen maksimal sekvenslengde LFSR.

**3-i**

Med utgangspunkt i deloppgave 3-e og 3-h, anta at et LFSR benyttes som kompakterer som vist i figuren under.



(Fortsettes side 7.)

Anta videre at den kombinatoriske kretsen tilføres en exhaustive testsekvens  $X_1X_2X_3 = \{000, 001, \dots, 110, 111\}$  og at responsen som framtrer på  $Z$  kompakteres i registeret. Avgjør om det finnes enkle låst-til-feil (SSF) i den kombinatoriske kretsen som ikke vil detekteres i resulterende signatur i kompakteringsregisteret under det gitte inngangspåtrykk.

## 4 (vekt 10%)

### 4-a

Utled det boolske uttrykket for *CARRY* for en full-adder. Reduser det mest mulig. Lønner det seg å bruke Karnaugh-diagram?

### 4-b

Vis at *SUM* kan uttrykkes ved  $A \oplus B \oplus C$  i en full-adder.

### 4-c

Beskriv karakteristiske fordeler og ulemper for:

I Ripple Adder

II Pipelined Adder

III Carry-Lookahead Adder

### 4-d

I Carry-Lookahead Adder brukes de boolske funksjonene  $P$  og  $G$ . Hva står disse benevnelsene for og hva uttrykkes de ved (som funksjon av de to delleddene  $A$  og  $B$ ).

### 4-e

$C_2$  i en Carry-Lookahead Adder kan uttrykkes som:

$$C_2 = G_2 + P_2G_1 + P_2P_1G_0 + P_2P_1P_0C_1$$

$C_1$  er her første Carry inn til adderen.

Faktoriser dette uttrykket på en praktisk måte og lag funksjonen ved hjelp av NMOS logikk a la figuren i oppgave 1. Hint: Her må du både stable transistorer oppå hverandre og sette de i parallell.