



NTNU

Det skapende universitet

Side 1 av 14

INSTITUTT FOR ELEKTRONIKK OG TELEKOMMUNIKASJON

EKSAMEN I FAG TFE4101 KRETS- OG DIGITALTEKNIKK

Faglig kontakt: Peter Svensson (1–3.5) / Kjetil Svarstad (3.6–4)

Tlf.: 995 72 470 / 458 54 333

Eksamensdato: Mandag 25. mai 2016

Eksamenstid (fra - til): 0900-1300

Hjelpemidler: D–Ingen trykte eller håndskrevne hjelpemidler er tillatt.
Bestemt, enkel kalkulator tillatt.

Annen informasjon: Maksimalt antall poeng for hver oppgave er gitt i parentes.
Maksimalt antall poeng oppnåelig totalt: 100.

Sensur faller innen 14. juni 2016.

Målform: Bokmål

Antall nummererte sider: 14

Antall unummererte sider i vedlegg: 0

Kontrollert av:

Dato

Sign

Tom side

Oppgave 1 (15%)

- a)
- b)
- c)

Oppgave 2 (15 %)

- a)
- b)
- c)

Oppgave 3 (40%)

Nedenfor er gitt 10 spørsmål i form av 3 påstander eller svaralternativer A, B eller C. Bare en av påstandene er riktig. Kryss av for riktig svar A, B eller C i tabellen bak i oppgavesettet.

OBS! Tabellsiden må leveres inn som en del av besvarelsen!

Riktig svar gir 4 poeng, manglede svar gir 0 poeng, og galt svar gir -2 poeng. Flere svar på samme spørsmål regnes som manglende svar og gir 0 poeng. Ved feil utfyllt svar, fyll den feilsvarte ruten helt, og sett kryss i riktig rute.

3-1

3-2

3-3

3-4

3-5

3-6 Gitt tre heltall $A = 366_{(7)}$, $B = 366_{(8)}$, og $C = F0_{(16)}$. Vi betrakter nå disse tre tallene som 8-bit tall uten fortegn, og definerer oss tre nye tall $X = A \& B$, $Y = A | C$, og $Z = B \oplus C$. Her er '&' bitvis AND-operasjon, '|' bitvis OR-operasjon, og ' \oplus ' bitvis XOR-operasjon. Med bitvis menes at operasjonen utføres mellom hvert bit i hver av de to operandene slik at feks. $0101 \& 1100 = 0100$. Finn X, Y og Z, og bestem hvilken av følgende påstander som *ikke* er riktig:

LF:

$A = 366_{(7)} = 195_{(10)} = 11000011_{(2)}$, $B = 366_{(8)} = 246_{(10)} = 11110110_{(2)}$,
 $C = F0_{(16)} = 240_{(10)} = 11110000_{(2)}$. Da blir $X = 11110000_{(2)} = 194_{(10)}$,
 $Y = 11110011_{(2)} = 243_{(10)}$ og $Z = 00000110_{(2)} = 6_{(10)}$. Vi konstaterer at
 $X + Z = 200$ og at både X og Z er partall, men at bare Z er mindre enn
 $2000_{(4)} = 128_{(10)}$.

A. Summen av to av tallene X , Y og Z er $200_{(10)}$.

B. To av tallene X , Y og Z er mindre enn $2000_{(4)}$.

C. To av tallene X , Y og Z er partall.

3-7 Gitt tre 4-bit 2's-komplement tall $A = 1010_2$, $B = 0011_2$, og $C = 1101_2$. Hvilke av de tre følgende påstandene er *ikke* riktig:

LF:

Dette kan løses på flere måter. La oss ta den enkle og konvertere tallene til desimal-representasjon: $-A = (1010)' + 1 = 0101 + 1 = 0110 = 6$, $A = -6$,
 $B = 3$ og $-C = (1101)' + 1 = 0010 + 1 = 0011 = 3$, $C = -3$. Dette betyr at
 A er mindre enn C , $A + C = -9$ som gir overflyt, og $A + B = -3$ som ikke gir
overflyt.

A. A er større enn C .

B. Operasjonen $A + C$ vil gi overflyt («overflow»).

C. Operasjonen $A + B$ vil ikke gi overflyt.

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

FIGUR 1 – Karnaugh-diagram med 4 variabler

3-8 Gitt Karnaugh-diagrammet i Figur 1 som definerer funksjonen F . Finn hvilken av de tre boolske ligningene under som *ikke* er identisk med F .

LF:

Figur 1 viser de fire primimplikantene, og disse er alle essensielle. La oss inspisere alternativene under mhp. disse.

A. $F = (A\bar{B} + C\bar{D})(\bar{A} + B + \bar{C} + D)$

LF:

Denne kan vi omskrive vha. assosiasjon: $F = (A\bar{B} + C\bar{D})(\bar{A} + B + \bar{C} + D) = A\bar{B}(\bar{A} + B + \bar{C} + D) + C\bar{D}(\bar{A} + B + \bar{C} + D)$. Siden vi vet at $X\bar{X}Y = 0$, så bruker vi assosiasjon en gang til og eliminerer slike ledd og får $F = A\bar{B}\bar{C} + A\bar{B}D + \bar{A}C\bar{D} + BC\bar{D}$. Disse fire subkubene tilsvarer akkurat de fire primimplikantene i diagrammet, dermed er denne løsningen identisk.

B. $F = A\bar{B}(\bar{C} + CD) + (\bar{A} + AB)C\bar{D}$

LF:

Her benytter vi oss av absorpsjon ($X + \bar{X}Y = X + Y$) og får $F = A\bar{B}(\bar{C} + D) + (\bar{A} + B)C\bar{D} = A\bar{B}\bar{C} + A\bar{B}D + \bar{A}C\bar{D} + BC\bar{D}$, den siste ved assosiasjon av begge parenteser. Som vi ser er dette uttrykket identisk med primleddene også.

C. $F = (A + C)(\bar{A} + \bar{B})(\bar{C} + \bar{D})$

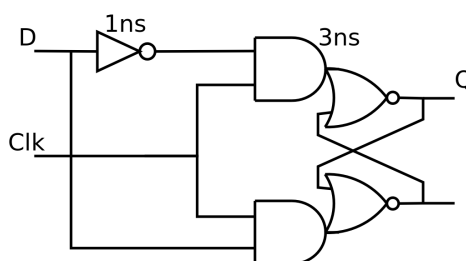
LF:

Nå aner det jo oss at dette må være det ikke identiske uttrykket, men la oss se en måte å finne det på. Enklest er å finne 0-maksternene direkte, og fylle inn i et Karnaughdiagram. Dette er vist i Figur 2 hvor det røde implikatet er første leddet, det grønne andre, og det blåe tredje. De tomme cellene er 1'ere, og som man ser er dette ikke identisk. Voilå!

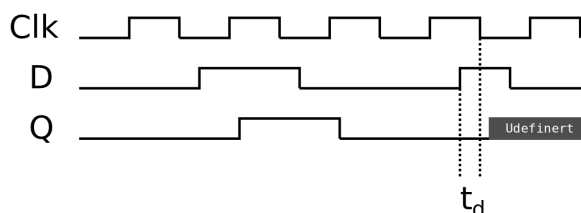
- 3-9 Figur 3 viser kretsen for en klokke D-lås («gated D-latch») der det er angitt forsinkelse gjennom inverterer (1 ns) og den kombinerte AND-NOR-porten (3 ns). Figur 4 viser hvordan vi forventer utgangssignalet Q skal bli når Clk og D er som vist i tidsdiagrammet. I tidsdiagrammet er det angitt en tidsperiode t_d mellom stigende flanke på D og fallende flanke på Clk. Hvilken av de følgende påstander er korrekt for at man skal få Q slik som vist i tidsdiagrammet:

	CD			
	00	01	11	10
AB				
00	0	0	0	
01	0	0	0	
11	0	0	0	0
10			0	

FIGUR 2 – Karnaugh-diagram med makstermer



FIGUR 3 – Krets for kloknet D-lås



FIGUR 4 – Tidsdiagram for kloknet D-lås

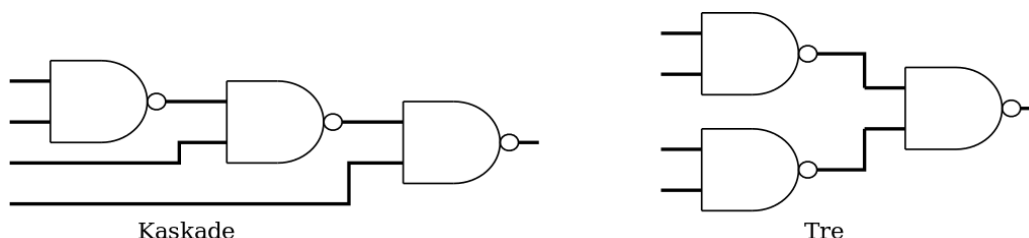
LF:

Vi vet at en RS-lås oscillerer, dvs. gir udefinert utgang hvis R og S går fra 1 til 0 samtidig. Siden en kloknet D-lås er en RS-lås med AND-port for klokking og en inverter på D for å lage (kloknet) R og D direkte for kloknet S, så betyr det at D-låsen blir ustabil hvis R og S er 1 mens klokka er 1, og at klokka går til 0. Dette kan bare oppnås i den korte periode der D går fra 0 til 1 og D' dermed går til 0 etter 1 ns pga Forsinkelsen i invertere. Det betyr at D-låsen blir ustabil hvis klokka går til 0 mindre enn 1 ns etter at D går til 1 (da både D og D' er 1). Ergo må t_d være mindre eller lik 1 ns for å gi udefinert utgang.

A. Forsinkelsen $t_d \leq 4ns$

B. Forsinkelsen $t_d \leq 1ns$

C. Forsinkelsen $t_d \leq 7ns$



FIGUR 5 – Hypotetiske NAND4-kretser

- 3-10 I ett kretsdesign har vi spesifikt bruk for en 4-inngangs NAND-port (NAND4), men slik er ikke tilgjengelig i vårt portbibliotek. Vi er derfor henvist til å realisere en slik NAND4-funksjon med 2-inngangs NAND-porter. I Figur 5 er det vist to forsøk på slik implementering, henholdsvis kalt «Kaskade» og «Tre». Hvilke av disse hypotetiske NAND4-kretsene er korrekte?

LF:

Dette kan analyseres på flere måter. En n-inngangs NAND-port skal (1) ha 0 ut når alle innganger er 1, og (2) ha 1 ut hvis en eller flere innganger er 0. (1) er OK for «Kaskade», men ikke (2) da feks. en 0 på tredje inngang fra oven gir 0 ut. For «Tre» stemmer ikke (1) engang. Ergo er ingen lik en NAND4.

Det kan også argumenteres enkelt med at vi vet at NAND-operasjonen ikke er assosiativ, men kommutativ. Siden «Kaskade» kan ordnes forskjellig (navngi innganger feks.) så må disse produsere forskjellig resultat (ikke assosiativ) for forskjellig navngiving, så holder dette ikke kravet til kommutativitet.

A. Ingen.

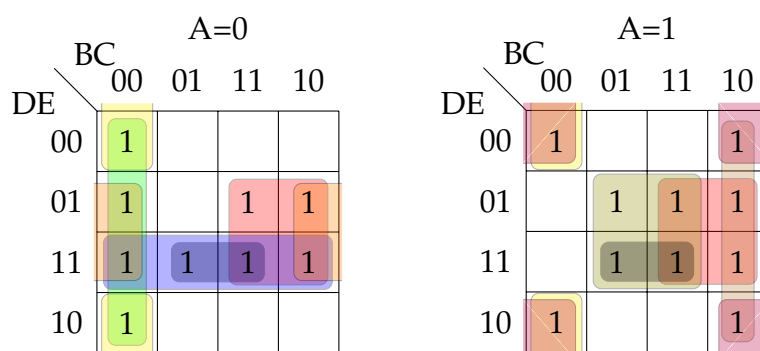
B. Begge.

C. Bare «Kaskade».

Oppgave 4 (30%)

a) (10%)

Karnaughdiagrammet i Figur 6 beskriver funksjonen F med sine 1-mintermer markert med 1 i diagrammet. Finn (i) alle primimplikanter for F, (ii) alle essensielle primimplikanter for F, og (iii) finn en ikke-redundant dekning for F.



FIGUR 6 – Karnaugh-diagram med 5 variabler for funksjonen F

LF:

Alle primimplikantene er markert i Figur 6. Det er $BE + \bar{A}\bar{B}\bar{C} + \bar{A}DE + \bar{A}\bar{C}\bar{E} + \bar{B}\bar{C}\bar{E} + A\bar{C}\bar{E} + ACE + CDE + ABC$. De essensielle er $BE + ACE$. For å få ir-redundant dekning må vi ha med tre ikke-essensielle primimplikanter, $F = BE + ACE + \bar{A}\bar{B}\bar{C} + \bar{A}DE + A\bar{C}\bar{E}$. Det finnes andre muligheter også.

b) (4%)

Du skal realisere en 2:1-multiplekser («selector») og en 4:1-multiplekser ved hjelp av bare to-inngangs NAND og NOR-porter samt inverterer INV. For den første er det ett enkelt kontrollsignal S, mens det for den andre er to bit kontroll med signalene S1 og S2. Skriv først opp Boolsk ligning for de to multiplekserne, og realiser dem så med bare INV, NAND2 og NOR2.

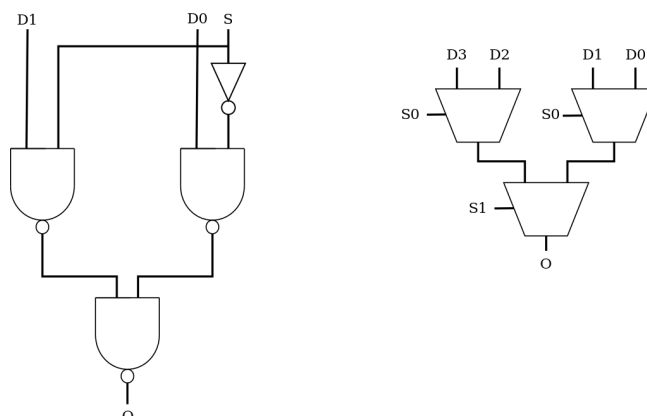
LF:

Skriver vi om funksjonen for en 2:1-mux vha. DeMorgan, så får vi: $O = \bar{S}D_0 + SD_1 = \overline{(\bar{S}D_0)} \cdot \overline{(SD_1)}$ hvor S er kontrollsignalet. Dette realiserer vi med tre NAND2-porter som vist til venstre i Figur 7. Tilhøyre vises hvordan en 4:1-mux implementeres vha. 3 2:1-muxer, først velger vi to vha. S1, og så en av de to resterende vha. S0.

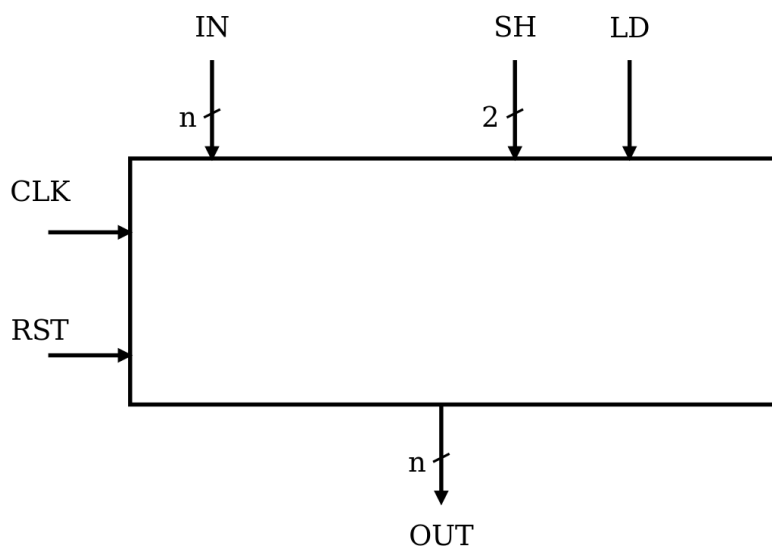
c) (12%)

Vi skal nå designe en krets for å gjøre suksessiv skifting av bit med rotering. Systemet er vist som en toppnivå modul i Figur 8, og de forskjellige signalene er som følger:

- Modulen inneholder et n-bit skiftregister man kan anta er laget med D-vipper. Vi benevner dette med REG fra nå av.



FIGUR 7 – 2:1 og 4:1 multiplekser



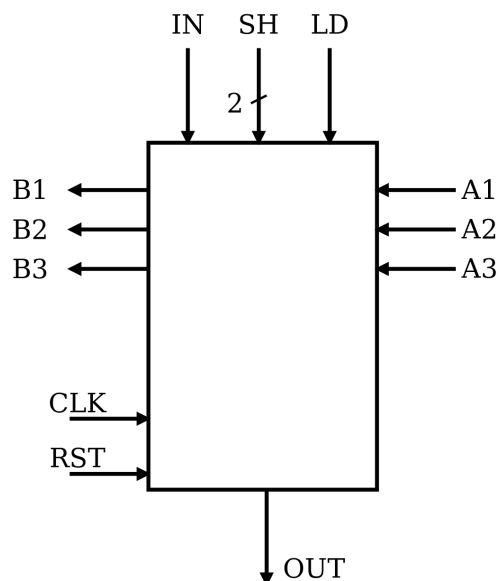
FIGUR 8 – Systemmodul for programmerbar skift-operasjon

- IN er et n -bit ord som skal kunne lastes parallelt inn i modulen, og som også er en parameter i skift-operasjonen (se under).
- SH er et 2-bit ord som bestemmer hvor mange plasser til venstre data i REG skal skiftes (med rotasjon) i skift-operasjonen.
- LD er et signal som styrer innlasting av et nytt dataord fra IN til REG i modulen.
- CLK er klokkesignal for REG.
- RST er resetsignal for REG.
- OUT er n -bit utgangssignal for det dataord som til enhver tid ligger i REG.

Funksjonen for denne skifteren med rotering er slik at for hver positive klokkeflanke skal følgende utføres:

- Når LD=1 skal IN lastes inn i REG.
- Når LD=0 og SH=00 skal verdien i REG beholdes.
- Når LD=0 og SH>00 skal REG venstreskiftes antall posisjoner angitt av SH, feks. hvis SH=11 (3), og REG=01000001, så vil etter neste klokke REG=00001010. Merk at de tre MSB'ene i dette tilfellet alle roteres rundt til LSB i REG. Samme gjelder for SH=10 og SH=01, bare for henholdsvis 2 og 1 posisjon.
- Etter klokke skal OUT ha den nye verdien i REG etter skifting.

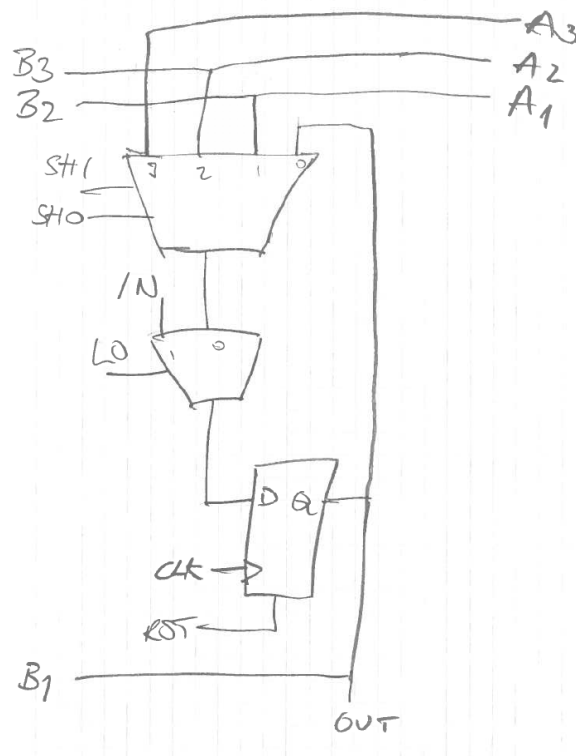
Vi ser at funksjonen er den samme for hvert bit i n-bit ordet i REG, og vil derfor realisere en 1-bit modul som den angitt i Figur 9. Denne modulen skal så kunne kobles opp med n instanser for å lage en n-bit systemmodul. Vi ønsker også at modulene bare skal kobles til sin nærmeste nabo med A1 til B1, A2 til B2, og A3 til B3 slik at all viderekobling av de bit som skal skiftes eller roteres går via disse signalene. Da kan man også realisere rotering ved å koble B1, B2 og B3 fra MSB-modulen inn på A1, A2 og A3 på LSB-modulen. Realiser denne en-bits skifter-modulen vha. D-vippe med klokke og reset, de logiske portene i Tabell 1 og eventuelt de multiplekserne du realiserte i forrige deloppgave.



FIGUR 9 – En-bit skifter-modul

LF:
Modulen er realisert i Figur 10.

d) (4%)



FIGUR 10 – 1-bit rotasjons-modul

Gitt en 4-bits versjon av systemet fra forrige deloppgave så skal du nå finne kritisk sti. Beregn så forsinkelse på kritisk sti gitt forsinkelsene i Tabell 1 og eventuelt realiseringen av multiplekserne. D-vippene har intern forsinkelse $T_{D \rightarrow Q} = 4,4 \text{ nS}$, oppsett-tid $T_{Setup} = 2,2 \text{ nS}$ og holdetid $T_{Hold} = 1,8 \text{ nS}$. Hva blir maksimal klokkefrekvens for systemet i dette tilfellet? Har antall bit noe å si for maksimal klokkefrekvens?

Funksjon	Navn	Innganger	Forsinkelse i nS
Bit-komplement	INV	1	1,2
OG-funksjon	AND2	2	3,6
Komplementert OG-funksjon	NAND2	2	2,4
ELLER-funksjon	OR2	2	3,6
Komplementert ELLER-funksjon	NOR2	2	2,4
Eksklusiv ELLER-funksjon	XOR2	2	4,6

TABELL 1 – Tilgjengelige logiske porter

LF:

Modulen realisert i forrige deloppgave har kritisk sti som er uavhengig av hvor mange plasser man skifter siden signalet som skal inn på D bare går gjennom en 4:1-mux og en 2:1-mux fra hvilken som helst Q i og med at A-signalene sendes direkte videre til modulen til venstre (skiftet en posisjon). Da blir forsinkelse i kritisk sti lik 6 NAND2-porter, $t_{PD} = 6 \cdot 2,4nS = 14,4nS$. Minimal klokkeperiode blir da $t_{CLK,min} = t_{PD} + T_{D \rightarrow Q} + T_{Setup} = 21,0nS$ siden klokka må ta hensyn til at D må være klar T_{Setup} før neste klokke. Maks klokkefrekvens blir da $f_{CLK,max} = 1000MHz/21,0 = 47,6MHz$.

Kritisk sti er som sagt ikke avhengig av hvor mange posisjoner som skiftes, og derfor heller **ikke** avhengig av hvor mange bit det er i skift/rotasjons-modulen.

Oppgitt formel - kretsdel

$$x(t) = x(t_{slutt}) + [x(t_0) - x(t_{slutt})] e^{-(t-t_0)/\tau}; \quad t \geq t_0$$

Vedlegg 1
HUSK Å LEVERE
DETTE ARKET SOM EN
DEL AV BESVARELSEN

Kandidatnr: _____
Emnekode: _____
Side: _____ / _____

SPØRSMÅL NR.	A	B	C
3-1			
3-2			
3-3			
3-4			
3-5			
3-6			
3-7			
3-8			
3-9			
3-10			