

# UNIVERSITETET I OSLO

## Det matematisk-naturvitenskapelige fakultet

Eksamen i	IN 240 — Digital Systemkonstruksjon
Eksamensdag:	15. desember 1993
Tid for eksamen:	9.00 – 15.00
Oppgavesettet er på	5 sider.
Vedlegg:	Ingen
Tillatte hjelpemidler:	Ingen trykte eller skrevne hjelpemidler er tillatt

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

**Gjør dine egne presiseringer der du mener at dette eventuelt er nødvendig. Skriv isåfall tydelig hvilke forutsetninger du gjør.**

### 1 Boolsk algebra (vekt 34%)

#### 1-a

Forenkle følgende boolske uttrykk:

$$AB'CD + ABC'D' + ABC'D + ABCD' + ABCD + AB'CD'$$

#### 1-b

Karakteristisk likning for en T-flip-flop er  $Q(t+1) = TQ' + T'Q$ . Sett opp karakteristisk tabell for denne flip-flop'en.

#### 1-c

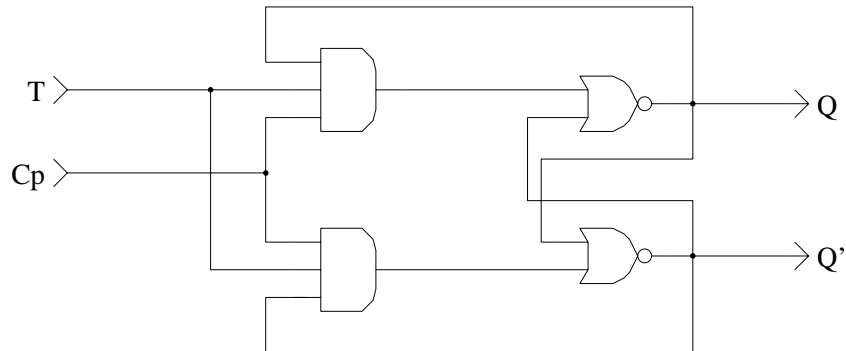
Design en 4-bits synkron BCD (Binary Coded Decimal)-teller med T flip-flop'er. Sett opp tilstandstabell og tegn logisk skjema. Bruk de ubenyttede tilstandene som *don't-care*-tilstander.

#### 1-d

Sett opp et tilstandsdiagram som viser hva som skjer hvis telleren starter i en av de ubrukte tilstandene. Modifiser eventuelt kretsen slik at telleren fungerer korrekt.

*(Fortsettes side 2.)*

1-e



T Flip-flop

Figur 1:

Analyser T flip-flop'en i figur 1 ved å betrakte den som en asynkron krets. Sett opp transisjonstabell. Er kretsen stabil når både  $T$  og  $CP$  er 1? Forklar!

## 2 Implementasjon (Vekt 33%)

2-a

Konstruktøren har mange frihetsgrader når instruksjonssettet (IS) skal bestemmes. Nevn kort positive og negative sider ved å bruke:

- komplekse operasjoner
- ortogonalt IS
- mange registre
- kort 'immediate'

2-b

Anta at du har valgt et IS og skal undersøke to forskjellige implementasjoner M1 og M2, av det samme instruksjonssettet. Det er fire klasser instruksjoner i IS: A, B, C og D. M1 og M2 har klokkefrekvenser på henholdsvis 50 MHz og 75 MHz. CPI for hver instruksjonsklasse er i de to tilfellene:

	M1	M2
A	1	2
B	2	2
C	3	4
D	4	4

Hva betyr CPI ?

(Fortsettes side 3.)

**2-c**

Hvordan vil du definere toppytelse? Under hvilke forutsetninger oppnås topp-ytelse for M1 og M2, og hva er topp-ytelsen i de to tilfellene?

**2-d**

Antall instruksjoner som eksekveres ved et program A er fordelt likt mellom de ulike instruksjonsklassene. Hva bli CPI for hver av maskinene M1 og M2?

**2-e**

Hvor mye hurtigere er M2 enn M1?

**2-f**

Ved hvilken klokkefrekvens ville M1 ha samme ytelsen som M2?

**2-g**

Hva sier Amdahls lov?

**2-h**

Anta at i M1 forbedres instruksjonsklassen D slik at CPI endres til 2. Hva blir økningen i ytelse, speedup, for M1 når programmet A kjøres?

**2-i**

Forbedringene skyldes at det innføres en carry-skip adder. Tegn skisse og forklar prinsippet. Under hvilke forhold har man minimal og maksimal adderingstid?

**3 Pipelining (Vekt 33%)**

Legg vekt på å redusere CPI når du løser denne oppgaven. Forklar hva som eventuelt er årsaken når du må gjøre noe som øker CPI.

```

1          ADDI   R1, R0, #0          /* R1 ← R0 + 0 */
2          ADDI   R2, R0, #76
3          LW     R3, V_BASE(R0)     /* R3 ← M[R0 + V_BASE] */
4  LOOP1:  ADD    R3, R3, R1         /* R3 ← R3 + R1 */
5          LW     R4, 0(R3)
6          LW     R5, 4(R3)
7          ADD    R5, R4, R5
8          SW     0(R3), R5         /* M[R3 + 0] ← R5 */
9          ADDI   R1, R1, #8
10         SLT   R6, R1, R2         /* if ( R1 < R2 ) R6 ← 1 else R6 ← 0 */
11        BNEZ  R6, LOOP1         /* if ( R6 != 0 ) PC = LOOP1 */
12        ADDI   R1, R0, #0         /* R1 ← R0 + 0 */
13        ADDI   R2, R0, #76
14        LW     R3, V_BASE(R0)
15  LOOP2:  ADD    R3, R3, R1
16        LW     R4, 0(R3)
17        LW     R5, 4(R3)
18        SLT   R6, R5, R4
19        BEQZ  R6, TRE           /* if ( R6 == 0 ) PC = TRE */
20        XOR   R4, R4, R5         /* R4 ← R4 ^ R5 */
21        XOR   R5, R4, R5
22        XOR   R4, R4, R5
23        SW     0(R3), R4
24        SW     4(R3), R5
25  TRE:    ADDI   R1, R1, #8
26        SLT   R6, R1, R2
27        BNEZ  R6, LOOP2

```

Figur 1: En instruksjonssekvens generert for en maskin uten pipelining.

### 3-a

Gjør rede for begrepene «pipeline interlocking», «delayed load» og «delayed branch», og hvordan og hvorfor de brukes.

Sekvensen med instruksjoner på figur 1 er generert for en maskin uten pipelining. Sekvensen traverserer en array med 20 32 bit tall to ganger. V\_BASE er starten på arrayen.

Anta at det er implementert 2 pipelinede maskiner som bruker de samme instruksjonene som i figur 1. Den ene, maskin A, har «delayed load» og «delayed branch», mens den andre, maskin B, ordner alt i hardware. Bruk gjerne linjenumrene på venstre side av figuren når du refererer til instruksjoner i figuren.

### 3-b

Hva vil skje om sekvensen med instruksjoner i figur 1 blir kjørt på maskin A og maskin B uten endringer?

(Fortsettes side 5.)

**3-c**

Forklar hva som må gjøres med sekvensen med instruksjoner i figur 1 for at den skal virke på maskin A.

**3-d**

Forklar hva som må gjøres med sekvensen med instruksjoner i figur 1 for at den skal virke på maskin B.

**3-e**

Forklar hva som må gjøres med sekvensen med instruksjoner i figur 1 for at den skal virke både på maskin A og maskin B.