1.

a) Kännetecknet på att det å ena sidan är en **Graykod** är att kodorden för två närliggande siffror bara skiljer sig åt i en position och

kännetecknet på att det är en **Excess-3 kod** är att kodordet för siffran 0 ej är 0000 utan förskjutet tre steg så att det i Graykoden blir 0010.

b) Antalet element (heltal) i området 0 t o m 9999 är 10^4. För att koda dem binärt krävs n_2 binära siffror där det gäller att $2^{n_2} \geq 10^4$, dvs $n_2 \cdot \log 2 \geq 4$.

Man kan genom huvudräkning bestämma n_2 på följande sätt. En digitaltekniker vet att $2^{10} = 1024$ och att $8 \cdot 1024 < 10^4 < 16 \cdot 1024$. Således krävs att

$$2^{n_2} = 16 \cdot 1024 = 2^4 \cdot 2^{10} = 2^{14}$$

dvs $n_2 = 14$

c) Följande söks: $(1011\ 0011.1100\ 1101)_{2421} = (d_1d_0,d_{-1}d_{-2})_{10}$

2421-talet består av binärkodade decimala siffror. Avkodning sker på följande sätt

$d_1 = 1{\cdot}2 + 0{\cdot}4 + 1{\cdot}2 + 1{\cdot}1 = 5$

$d_0 = 0{\cdot}2 + 0{\cdot}4 + 1{\cdot}2 + 1{\cdot}1 = 3$

$d_{-1} = 1{\cdot}2 + 1{\cdot}4 + 0{\cdot}2 + 0{\cdot}1 = 6$

$d_{-2} = 1{\cdot}2 + 1{\cdot}4 + 0{\cdot}2 + 1{\cdot}1 = 7$

Talet är $(d_1d_0,d_{-1}d_{-2})_{10} = (53,67)_{10}$

d) Subtraktionen A-T med talen W och V så görs på följande sätt, se Bilaga 2:

```
          1    1
   11001010    W
  +11110111    V1k
  ----------
   11000010
```

Således är W ett negativt tal, vars belopp får genom 2-komplementering
$W = (-\ 00110110)_2 = -54$

och

V ett positivt tal, vars belopp fås genom 1-komplementering av V_{1k}
$V = (+00001000)_2 = +8$

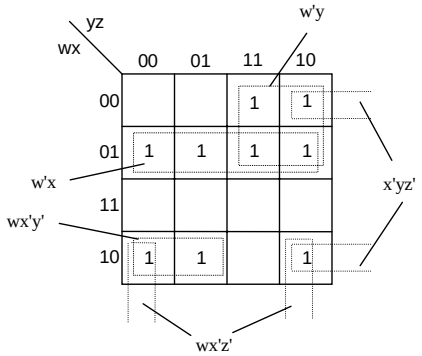
2. **Funktionsbeskrivning**

via funktionstabell

	w	x	y	z	f
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	1
5	0	1	0	1	1
6	0	1	1	0	1
7	0	1	1	1	1
8	1	0	0	0	1
9	1	0	0	1	1
10	1	0	1	0	1
11	1	0	1	1	0
12	1	1	0	0	0
13	1	1	0	1	0
14	1	1	1	0	0
15	1	1	1	1	0

Lösning/förenkling

via Karnaughdiagram



Ur Karnaughdiagrammet ställs följande uttryck upp:

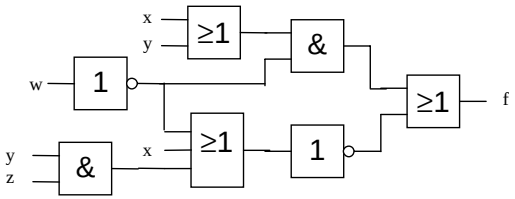
$$f = w'x + w'y + wx'y' + x'yz' = w'(x+y) + w(x+y)' + y(x+z)' \quad (1)$$

eller

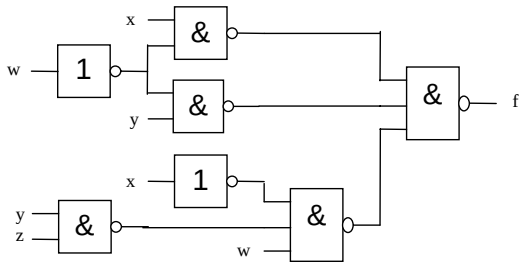
$$f = w'x + w'y + wx'y' + wx'z' = w'(x+y) + wx'(yz)'$$
$$= w'(x+y) + (w' + x)'(yz)' = w'(x+y) + (w' + x + yz)' \quad (2)$$

Realisering

- med inverterare, OCH- och ELLER-grindar realiseras nätet enklast med uttrycket (2)



- med NAND-grindar realiseras nätet enklast också med uttrycket (2)



3. 1-till-8 avkodare är ju en mintermsavkodare. f_0 och f_1 indikerar om vissa mintermer är "1".
Detta kan sammanfattas i funktionstabell

term	x y z	f_0	f_1
0	0 0 0	0	0
1	0 0 1	1	0
2	0 1 0	1	0
3	0 1 1	0	1
4	1 0 0	1	0
5	1 0 1	0	1
6	1 1 0	0	1
7	1 1 1	1	1

a) Ur figuren och tabellen erhålles $f_0 = x'y'z + x'yz' + xy'z' + xyz$

$f_1 = x'yz + xy'z + xyz' + xyz$

b) Funktionen ser man bäst i funktionstabellen. Där ser man att f_0 och f_1 är samma funktioner som summasiffran s och minnessiffran c_{ut} hos en heladderare där y och z är inkommande talsiffror samt x inkommande minnessiffra c_{in}.

---

4. **a)** Synkront sekvensnät

b) Tillstånds- och utsignalstabellen

Ur kopplingen kan vi teckna de Booleska uttrycken för q_1^+ , q_0^+ och u:

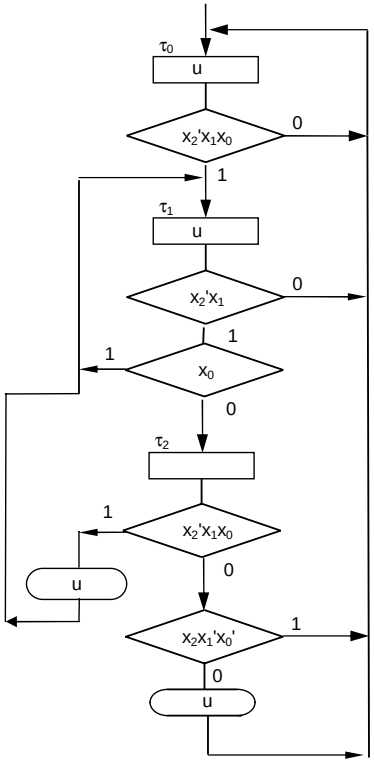
$q_1^+ = x_2'x_1x_0'q_0$

$q_0^+ = x_2'x_1x_0$

$u = (x_2x_1'x_0'q_1)'$

Tillstånd		Insign	Nästa tillst	Utsign
	q_1q_0	$x_2x_1x_0$	$q_1^+q_0^+$	u
τ_0	0 0	0 1 0 0 1 1 1 0 0 övr	0 0 0 1 0 0 0 0	1 1 1 1
τ_1	0 1	0 1 0 0 1 1 1 0 0 övr	1 0 0 1 0 0 0 0	1 1 1 1
τ_2	1 0	0 1 0 0 1 1 1 0 0 övr	0 0 0 1 0 0 0 0	1 1 0 1
τ_3	1 1	0 1 0 0 1 1 1 0 0 övr	1 0 0 1 0 0 0 0	1 1 0 1

c) Funktionsbeskrivning i ASM-plan



5. a)

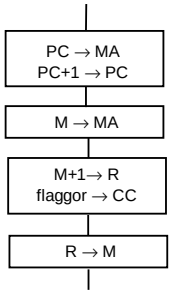
Klock-cykel	OE_A	OE_B	OE_R	OE_{CC}	LD_A	LD_B	LD_T	LD_R	LD_{CC}	g_2	g_1 g_0	ALU-funktion	U bin	RTN-beskr
1								1				0000	00000000	00H → R
2			1					1			01	1000	00000001	R+1 → R
3			1					1			01	1011	00000011	2R + 1 → R
4			1					1				1011	00000110	2R → R
5			1				1					-	-	R → T
6		1						1	1			0111	$b_7b_6b_5b_4b_3b_2'b_1'b_0$	A ⊕ T → R flaggor → CC
7			1			1						-	-	R → A

b) Operationen innebär att bitarna nr 1 och 2 av B-registret inverteras, med motsvarande flaggsättning.

6. a) Till höger ges den rätta ASM-planen. Teknologernas ASM-plan har två fel.

- De har missat att **MA** i andra tillståndet pekar på den byte som **innehåller adressen** till operanden som skall skiftas. MA pekar således inte på operanden.
- De har också missat att ladda flaggregistret med de flaggvärden som inkrementeringen resulterar i.

b) Teknologerna beskriver en instruktion som ej påverkar vare sig processorns interna tillstånd eller dataarean i minnet utan den inkrementerar ("förstör") den i instruktionen ingående adressen.



7. a)

```
LDX   #$D2E0     8E  D2  E0
CMPD  #0         10  83  00  00
BEQ   L1         27  06
ADDD  ,X         E3  84
STD   10,X       ED  0A
```

b)

```
LDX   #$A1C0     omedelbar (immediate)
LDD   ,X         register indirekt
BEQ   L1         PC-relativ
ASLB             inherent
STD   10,X       register-relativ
```

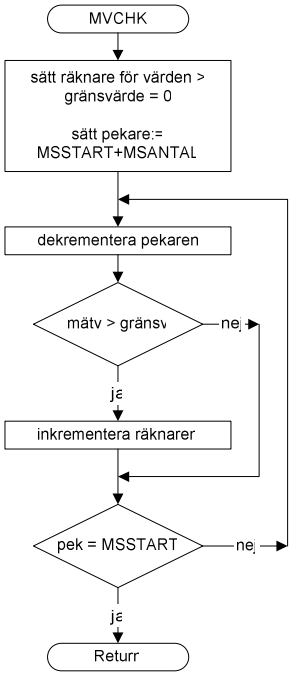
c)

```
      LDX   #$A1C0     *
      LDD   ,X         *hämta 16-bitars värde w
      CMPD  #0         *
      BEQ   L1         *om w=0 gå till L1
      ASLB             *multiplicera w med 2
      ROLA             *
      ASLB             *multiplicera 2w med 2
      ROLA             *
      ADDD  ,X         *addera w till 4w, dvs bilda 5w
L1    STD   10,X       *lagra 5w på 10 adresser längre fram
```

dvs instruktionssekvensen hämtar ett operandvärde w och multiplicerar det med 5.

d) Man löper då risken att få ”overflow” om w är för stort

8. Maskin**oberoende** flödesplan



Assemblerprogram

```
MVCHK   PSHS    B,CC    *spara antal mätvärden i serien
                        *och CC
        CLR     ,-S     *räknare för värden > GRVRDE
                        *på TOS nollställs

L0      DECB            *förbered pekaravstånd
        CMPA    B,X     *jämför gränsvärdet med mätvärde
                        *i serien, bakifrån, stegvis
        BHS     L1      *om mätvärdet större än
gränsvärdet

        INC     ,S      *öka räknare på TOS med 1

L1      TSTB            *undersök pekaravståndet
        BNE     L0      *om > 0, jämför med nästa
mätvärde

        PULS    A       *antal mätvärden > GRVRDE till A
        PULS    B,CC    *återställ i B antal mätvärden i
serien
*samt återställ CC
        RTS
```