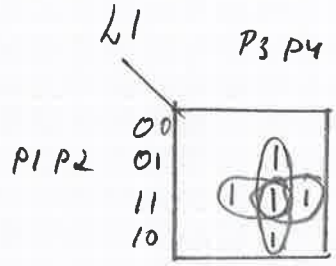
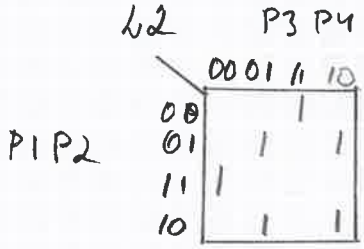
P1 P2 P3 P4	L1 L2 L3
0 0 0 0	0 0 1
0 0 0 1	0 0 1
0 0 1 0	0 0 1
0 0 1 1	0 1 0
0 1 0 0	0 0 1
0 1 0 1	0 1 0
0 1 1 0	0 1 0
0 1 1 1	1 0 0
1 0 0 0	0 0 1
1 0 0 1	0 1 0
1 0 1 0	0 1 0
1 0 1 1	1 0 0
1 1 0 0	0 1 0
1 1 0 1	1 0 0
1 1 1 0	1 0 0
1 1 1 1	1 0 0



$L1 = P1 \cdot P2 \cdot P3 + P2 \cdot P3 \cdot P4 + P1 \cdot P2 \cdot P4 + P1 \cdot P2 \cdot P3$



$P1 \cdot P4 (P2 \oplus P3)$

$L2 = \overline{P1} \cdot \overline{P2} \cdot P3 \cdot P4 + \overline{P1} \cdot P2 \cdot P3 \cdot P4 + \overline{P1} \cdot P2 \cdot P3 \cdot \overline{P4} + P1 \cdot P2 \cdot \overline{P3} \cdot \overline{P4} +$

$\overline{P1} \cdot P2 \cdot (P3 \oplus P4)$

$+ P1 \cdot \overline{P2} \cdot \overline{P3} \cdot P4 + P1 \cdot \overline{P2} \cdot P3 \cdot \overline{P4}$ $(P1 \oplus P2) \cdot (P3 \oplus P4)$

$P1 \cdot \overline{P2} (P3 \oplus P4)$

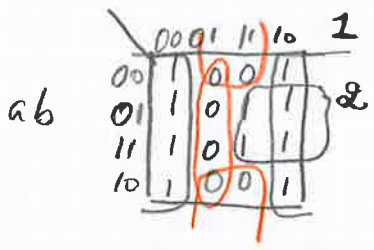
$L2 = (P1 \oplus P4) \cdot (P2 \oplus P3) + (P1 \oplus P2) \cdot (P3 \oplus P4) + (P1 \oplus P3)(P2 \oplus P4)$

Ur tabell fås $\Rightarrow$ $L3 = \overline{L1 + L2}$ (9p)

2) Ur schema:

$f = \bar{a} \cdot \bar{c} \cdot \bar{d} + b \cdot \bar{c} \cdot \bar{d} + a \cdot \bar{c} \cdot \bar{d} + b \cdot c \cdot d + \bar{a} \cdot c \cdot \bar{d} + a \cdot c \cdot \bar{d}$

f cd



$f = \bar{d} + b \cdot c$

$\bar{f} = \overline{\bar{c} \cdot d + \bar{b} \cdot d}$ (4p)

$f = \overline{\bar{c} \cdot d} + \overline{\bar{b} \cdot d}$

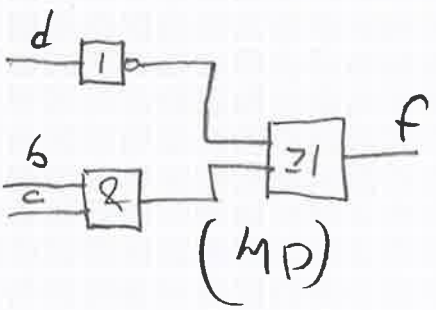
$f = \overline{\bar{c} \cdot d} \cdot \overline{\bar{b} \cdot d}$

$f = (c + \bar{d}) \cdot (b + \bar{d})$

Vänd!

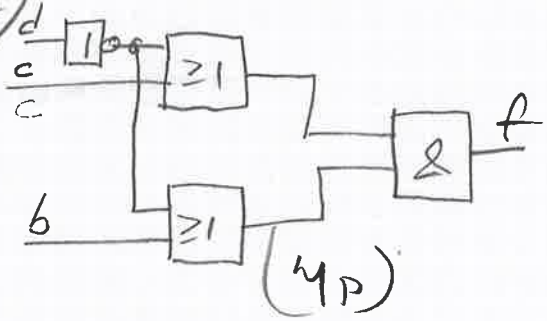
2 forts

a)



(MP)

b)



(MP)

3)

92

	Σ
1	0
2	0
4	1 92
8	1 88
16	1 80
32	0 64
64	1 64
128	0

a) $92_{10} = 01\,01\,1100_2$ (1 | 3 | 4; 5 | C)

b) $92 = 5C_{16}$

c) $92 = 134_8$

d) $^2 01011100_2 = 10100011$ (1)

10100100

e) $80 = 64+16 = 01010000_2$

$80 - 92 = 01010000 + 10100100$

$$\begin{array}{r} 01010000 \\ +\ 10100100 \\ \hline 11110100 \end{array}$$ neg!

$^2 11110100 = 00001011$ (1)

$+\ 00001100 = 12_{10}$

Svar: -12

f)

$C = 10 - 13 = -3$

0	00000000	0
-1	11111111	255
-2	11111110	254
-3	11111101	253

Svar: 253

8p

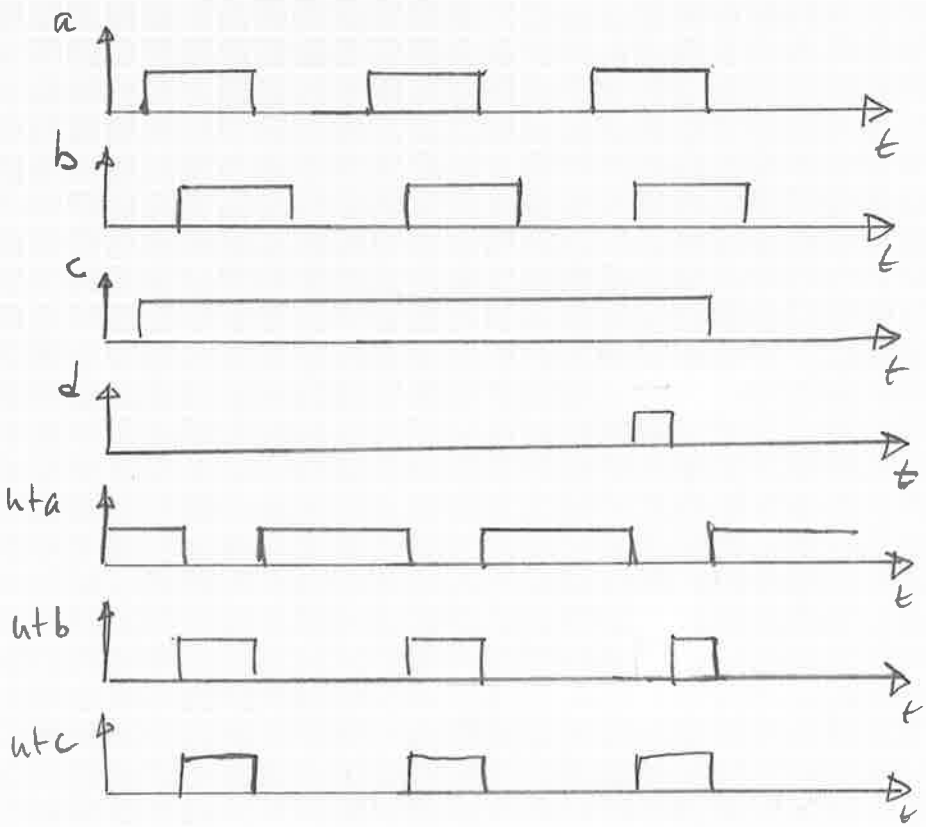
4)

$$uta = \overline{a \cdot b + c \cdot d}$$

$$utb = a \cdot b \oplus c \cdot d$$

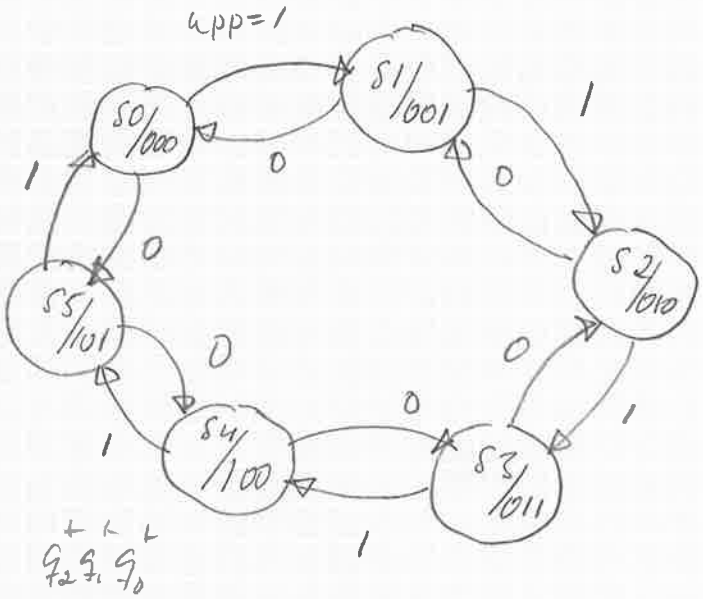
$$utc = \overline{\overline{a \cdot b} \cdot \overline{c \cdot d}} = a \cdot b + c \cdot d$$

a	b	c	d	a·b	c·d	uta	utb	utc
0	0	0	0	0	0	1	0	0
0	0	0	1	0	0	1	0	0
0	0	1	0	0	0	1	0	0
0	0	1	1	0	1	0	1	1
0	1	0	0	0	0	1	0	0
0	1	0	1	0	0	1	0	0
0	1	1	0	0	0	1	0	0
0	1	1	1	0	1	0	1	1
1	0	0	0	0	0	1	0	0
1	0	0	1	0	0	1	0	0
1	0	1	0	0	0	1	0	0
1	0	1	1	0	1	0	1	1
1	1	0	0	1	0	0	1	1
1	1	0	1	1	0	0	1	1
1	1	1	0	1	0	0	1	1
1	1	1	1	1	1	0	0	1



(6p)

5)



q_2 q_1 q_0	upp=0	upp=1
0 0 0	1 0 1	0 0 1
0 0 1	0 0 0	0 1 0
0 1 0	0 0 1	0 1 1
0 1 1	0 1 0	1 0 0
1 0 0	0 1 1	1 0 1
1 0 1	1 0 0	0 0 0

q_2^+ q_0 upp

$q_2 q_1$ \ upp q_0	00	01	11	10
00	1	0	0	0
01	0	0	1	0
11	–	–	–	–
10	0	1	0	1

q_1^+ q_0 upp

$q_2 q_1$ \ upp q_0	00	01	11	10
00	0	0	1	0
01	0	1	0	1
11	–	–	–	–
10	0	0	0	0

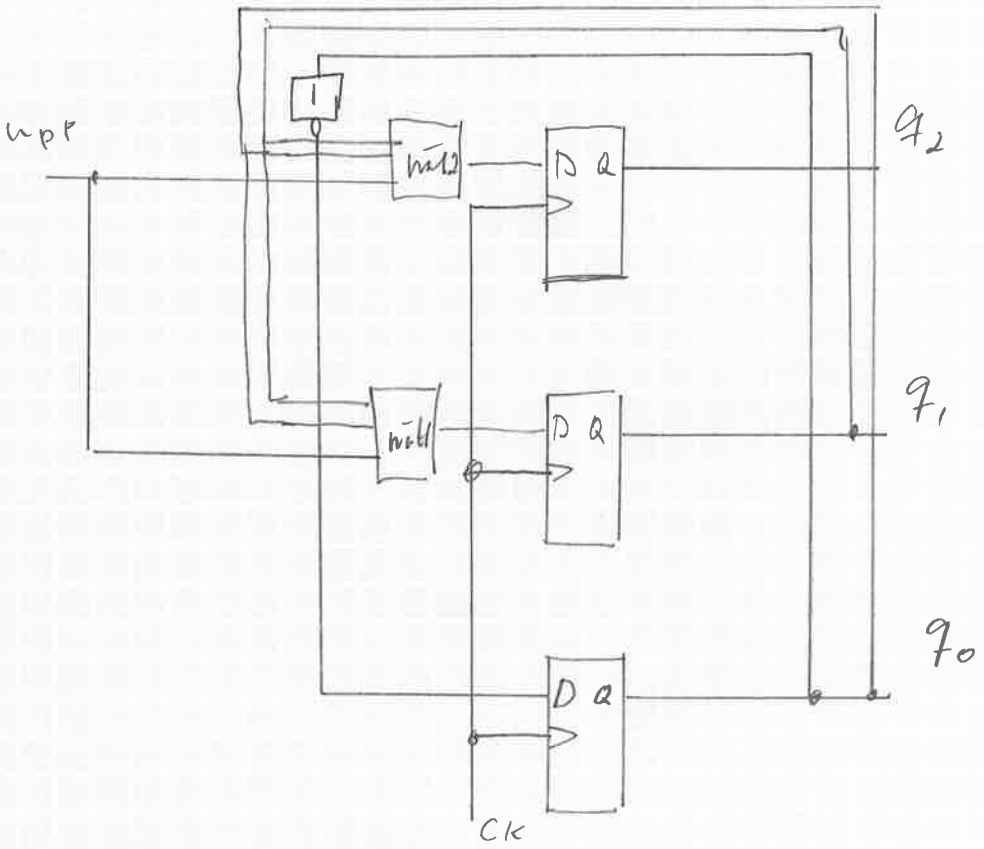
q_0^+ q_0 upp

$q_2 q_1$ \ upp q_0	00	01	11	10
00	1	1	0	0
01	1	1	0	0
11	–	–	–	–
10	1	1	0	0

$$q_0^+ = \overline{q_0}$$

$$q_1^+ = q_1 \cdot \overline{q_0} \cdot upp + q_1 \cdot \overline{q_0} \cdot upp + \overline{q_2} \cdot \overline{q_1} \cdot q_0 upp + q_2 \cdot \overline{q_1} \cdot \overline{q_0} \cdot upp$$

$$q_2^+ = q_2 \cdot \overline{q_0} \cdot upp + q_1 \cdot q_0 \cdot upp + q_2 \cdot q_0 \cdot \overline{upp} + \overline{q_2} \cdot \overline{q_1} \cdot \overline{q_0} \cdot upp$$

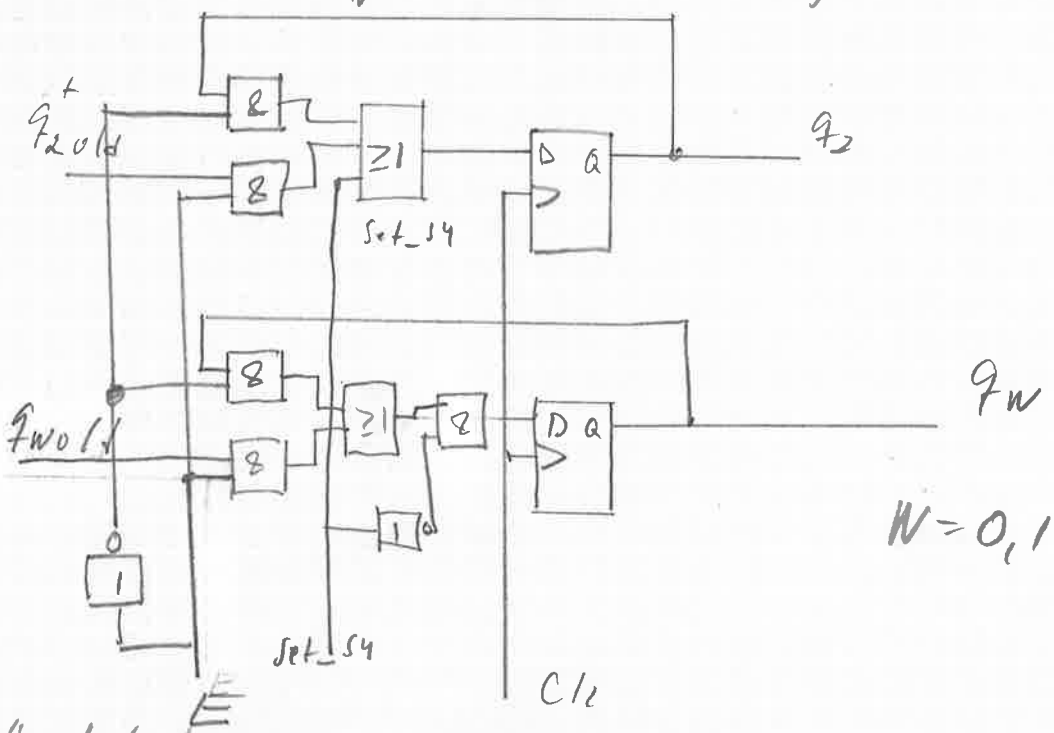


nät 1 och nät 2 är de som har erhållits ur Karnaughdiagrammen

6)

Vippa 2: $q_2^+ = E \cdot q_{2old}^+ + \overline{E} \cdot q_2 + \overline{set\_S4}$

Vippa 0 och 1: $q_N^+ = \left(E \cdot q_{Nold}^+ + \overline{E} \cdot q_N\right) \cdot \overline{set\_S4}$



$N = 0, 1$

8) Se läroboken.

```
1   -- enkel modula 6 räknare med upp/nedfunktion
2   library IEEE;
3   use IEEE.std_logic_1164.all;
4
5   entity mod6 is
6     port(upp, clock, enable, set_S4:in std_logic;
7         q: out std_logic_vector(2 downto 0));
8   end entity mod6;
9
10  architecture beteende of mod6 is
11  type state_type is (s0,s1,s2,s3 ,s4, s5);
12  signal present_state, next_state:state_type;
13  begin
14     process(present_state, upp)
15     begin
16         case present_state is
17              when s0 =>  if upp='1' then
18                          next_state<=s1;
19                     else
20                          next_state<=s5;
21                     end if;
22            when s1  => if upp='1' then
23                          next_state<=s2;
24                     else
25                          next_state<=s0;
26                     end if;
27            when s2  => if upp='1' then
28                          next_state<=s3;
29                     else
30                          next_state<=s1;
31                     end if;
32            when s3  => if upp='1' then
33                          next_state<=s4;
34                     else
35                          next_state<=s2;
36                     end if;
37            when s4  => if upp='1'  then
38                          next_state<=s5;
39                     else
40                          next_state<=s3;
41                     end if;
42            when s5  => if upp='1' then
43                          next_state<=s0;
44                     else
45                          next_state<=s4;
46                     end if;
47        end case;
48     end process;
49
50     process(present_state)
51     begin
52        case (present_state) is
53           when s0  => q<="000";
54           when s1  => q<="001";
55           when s2  => q<="010";
56           when s3  => q<="011";
```

uppg. 7 forts.

```
57            when s4  => q<="100";
58            when s5  => q<="101";
59         end case;
60      end process;
61
62      process(clock,enable, set_S4)
63      begin
64         if rising_edge(clock) then
65            if set_S4='1' then
66               present_state<=s4;
67            elsif
68               enable='0' then
69               present_state<=present_state;
70            else
71               present_state<=next_state;
72            end if;
73         end if;
74      end process;
75   end architecture beteende;
76
77
```