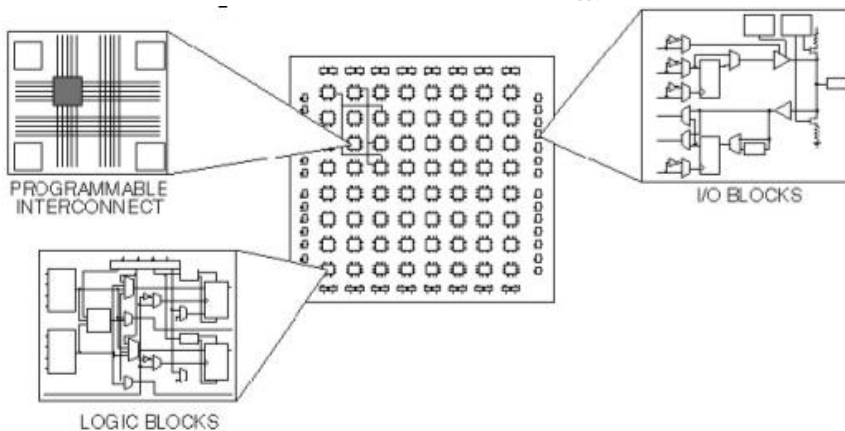
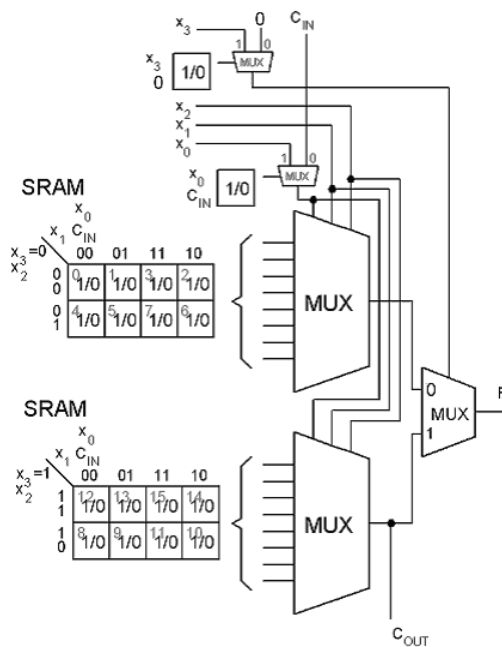


# Digital Design IE1204/5



## Övningshäfte



Sammanställt av *William Sandqvist* [william@kth.se](mailto:william@kth.se)

ICT/Elektroniksystem



## Talsystem och koder

### 1.1

Nedanstående decimala tal (med basen 10) är givna. Ange motsvarande binära tal.

- a) 9 b) 12 c) 71 d) 503

### 1.2

Omvandla nedanstående binärtal till decimaltal.

- a)  $101101001_2$  b)  $110100.010_2$

### 1.3

Omvandla nedanstående binärtal (bas=2) till motsvarande oktala tal (bas=8) och hexadecimala tal (bas=16).

- a)  $011101_2$  b)  $10001011_2$  c)  $100110101_2$  d)  $1101111010010001_2$  e)  $101001.001_2$

### 1.4

Omvandla nedanstående hexadecimala tal (bas=16) till motsvarande oktala tal (bas=8).

- a)  $94D_{16}$  b)  $9E.7A_{16}$

### 1.5

Omvandla det oktala (bas=8) talet  $4515_8$  till motsvarande hexadecimala tal (bas=16).

### 1.6

Skriv det hexadecimala (bas=16) talet  $BAC_{16}$  på decimal form (bas=10).

### 1.7

Vad karakteriserar Gray-koder, och hur kan de konstrueras?

### 1.8

Skriv följande tal "med tecken" med två-komplementsnotation,  $x = (x_6, x_5, x_4, x_3, x_2, x_1, x_0)$ .

- a) -23 b) -1 c) 38 d) -64

### 1.9

Skriv följande tal "med tecken" med ett-komplementsnotation,  $x = (x_6, x_5, x_4, x_3, x_2, x_1, x_0)$ .

- a) -23 b) -1 c) 38 d) -0

## Digital aritmetik

### 2.1

Addera för hand följande par binära tal.

- a)  $110 + 010$  b)  $1110 + 1001$  c)  $110011.01 + 111.1$  d)  $0.1101 + 0.1110$

### 2.2

Addera eller subtrahera (addition med motsvarande negativa tal) nedanstående tal. Talen skall representeras som binära 4-bitstal (Nibble) på två-komplementform.

- a)  $1 + 2$  b)  $4 - 1$  c)  $7 - 8$  d)  $-3 - 5$

### 2.3

Multiplitera för hand följande par av teckenlösa binära tal.

- a)  $110 \cdot 010$  b)  $1110 \cdot 1001$  c)  $110011.01 \cdot 111.1$  d)  $0.1101 \cdot 0.1110$

### 2.4

Dividera för hand följande par av teckenlösa binära tal.

- a)  $110/010$  b)  $1110/1001$

## 2.5

IEEE-754 standarden för lagring av 32-bitars flyttal.

Antag att ett 32-bitars flyttal lagras i ett register som:  $40C80000_{16}$  vilket reellt decimaltal är det?

## (2.6)

Flyttalsformatets principer blir mer överskådliga om man av pedagogiska skäl ”skalar ned” det till 4 bitars registerstorlek (Nibble). Däremot skulle ett 4-bitarsformat vara praktiskt oanvändbart.

Antag följande fyrabitars flyttalsformat:  $[b_3b_2b_1b_0] = (-1^{b_3}) \cdot (1.b_0) \cdot (2^{b_2b_1-1})$

Tecknet uttrycks med biten  $b_3$ , binalerna representeras av en bit  $b_0$ , och exponenten har två bitar  $b_2b_1$  uttryckta som excess-1.

- Räkna upp de tal som kan representeras med full precision. Markera dem på tallinjen.
- Hur stort är det största kvantiseringsfelet.
- Kan talet 0 representeras? Om inte, föreslå en förändring av formatet så att 0 kan representeras.

## (2.7)

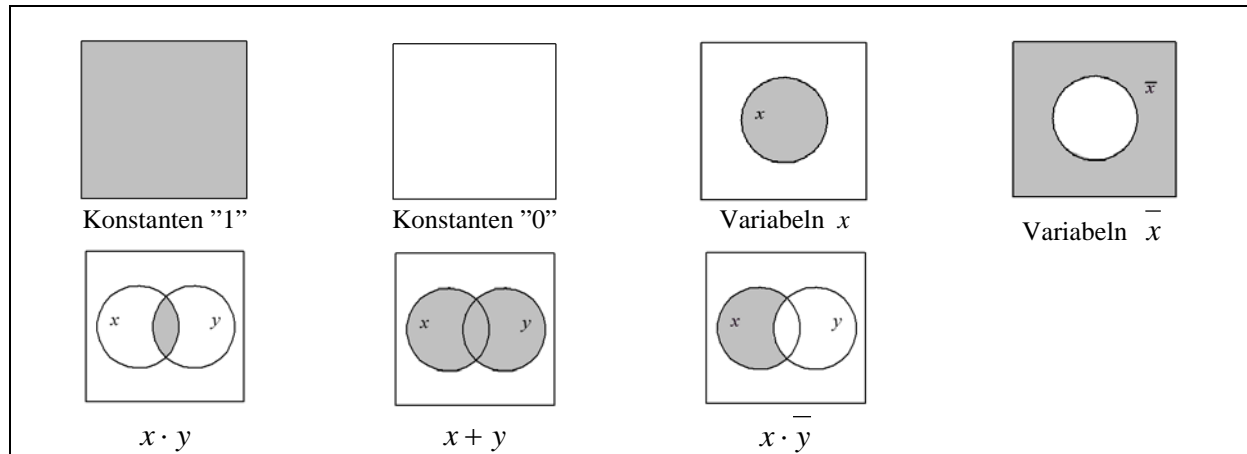
För att undersöka addition och multiplikation av flyttal antar vi nu av pedagogiska skäl ett 6-bitarsformat. (Detta är fortfarande för få bitar för att vara praktiskt användbart).

$[b_5b_4b_3b_2b_1b_0] = (-1^{b_5}) \cdot (1.b_2b_1b_0) \cdot (2^{b_4b_3-1})$

- Vilka av följande tal kan representeras i detta format? 0,25 0,8125 -1,375 4,25 7.5
- Addera talen  $(b_5b_4b_3b_2b_1b_0)$  001111 och 010010. Vad krävs för att undvika precisionsförlust?
- Multiplitera de föregående talen med varandra.

# Mängdräkning och kubteori

## Venn-diagram representation



### 3.1

Bevisa den distributiva lagen med hjälp av Venn-diagram.

$$x + y \cdot z = (x + y) \cdot (x + z)$$

### 3.2

Bevisa De Morgans lag med hjälp av Venn-diagram.

$$\overline{x \cdot y} = \overline{x} + \overline{y}$$

### 3.3

a) Rita ett Venn-diagram för tre variabler och markera var sanningstabellens alla mintermer är placerade.

b) Minimera funktionen med hjälp av Venn-diagram.

$$f = \overline{x_2} \overline{x_1} \overline{x_0} + \overline{x_2} \overline{x_1} x_0 + \overline{x_2} x_1 \overline{x_0} + \overline{x_2} x_1 x_0 + x_2 \overline{x_1} \overline{x_0}$$

## Kub representation

### 3.4

a) Representera följande funktion av tre variabler som en 3-dimensionell kub med Gray-kodade hörn.

$$f(x_2, x_1, x_0) = \sum m(0, 2, 3, 4, 6)$$

b) Använd kuben för att förenkla funktionen.

# Boolesk algebra och grindar

## Booles algebra

|  |  |   |                           |  |                       |  |                         |  |
|--|--|---|---------------------------|--|-----------------------|--|-------------------------|--|
| $A \cdot A = A \quad A \cdot 0 = 0 \quad A + 0 = A$<br>$A + A = A \quad A \cdot 1 = A \quad A + 1 = 1$ |  | <table border="1"> <tr> <td><b>Absorbtionslagarna</b></td> <td> <math>A + A \cdot B = A</math><br/> <math>A \cdot (A + B) = A</math> </td> </tr> <tr> <td><b>Koncensuslagen</b></td> <td> <math>A \cdot B + \bar{A} \cdot C =</math><br/> <math>A \cdot B + \bar{A} \cdot C + B \cdot C</math> </td> </tr> <tr> <td><b>de Morgans lagar</b></td> <td> <math>\overline{(A + B)} = \bar{A} \cdot \bar{B}</math><br/> <math>\overline{(A \cdot B)} = \bar{A} + \bar{B}</math> </td> </tr> </table> | <b>Absorbtionslagarna</b> | $A + A \cdot B = A$<br>$A \cdot (A + B) = A$ | <b>Koncensuslagen</b> | $A \cdot B + \bar{A} \cdot C =$<br>$A \cdot B + \bar{A} \cdot C + B \cdot C$ | <b>de Morgans lagar</b> | $\overline{(A + B)} = \bar{A} \cdot \bar{B}$<br>$\overline{(A \cdot B)} = \bar{A} + \bar{B}$ |
| <b>Absorbtionslagarna</b>  | $A + A \cdot B = A$<br>$A \cdot (A + B) = A$   |   |                           |  |                       |  |                         |  |
| <b>Koncensuslagen</b>  | $A \cdot B + \bar{A} \cdot C =$<br>$A \cdot B + \bar{A} \cdot C + B \cdot C$                 |   |                           |  |                       |  |                         |  |
| <b>de Morgans lagar</b>  | $\overline{(A + B)} = \bar{A} \cdot \bar{B}$<br>$\overline{(A \cdot B)} = \bar{A} + \bar{B}$ |   |                           |  |                       |  |                         |  |
| <b>Distributiva lagarna</b>  | $A \cdot (B + C) = A \cdot B + A \cdot C$<br>$A + (B \cdot C) = (A + B) \cdot (A + C)$       |   |                           |  |                       |  |                         |  |
| <b>Kommutativa lagarna</b>   | $A \cdot B = B \cdot A$<br>$A + B = B + A$   |   |                           |  |                       |  |                         |  |
| <b>Associativa lagarna</b>   | $(A \cdot B) \cdot C = A \cdot (B \cdot C)$<br>$(A + B) + C = A + (B + C)$                   |   |                           |  |                       |  |                         |  |

### 4.1

Använd räknelagarna i den booleska algebran för att förenkla följande logiska uttryck:

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

### 4.2

Bevisa algebraiskt att följande samband är giltiga.

- $\overline{(x_3 x_2 + 1 + x_3 x_2)} x_1 + x_2 x_1 + x_2 + x_1 = 1$
- $\overline{x_3 x_2 x_1} + \overline{(x_3 + x_1)} = \overline{x_3 x_1}$
- $\overline{(x_2 + x_1)} x_2 x_1 + x_3 = \overline{x_2 x_1} + x_3$
- $\overline{x_2 + x_1} + \overline{x_2 x_1} + x_3 = \overline{x_2 x_1} + x_3$

### 4.3

Förenkla nedanstående tre uttryck så långt som möjligt.

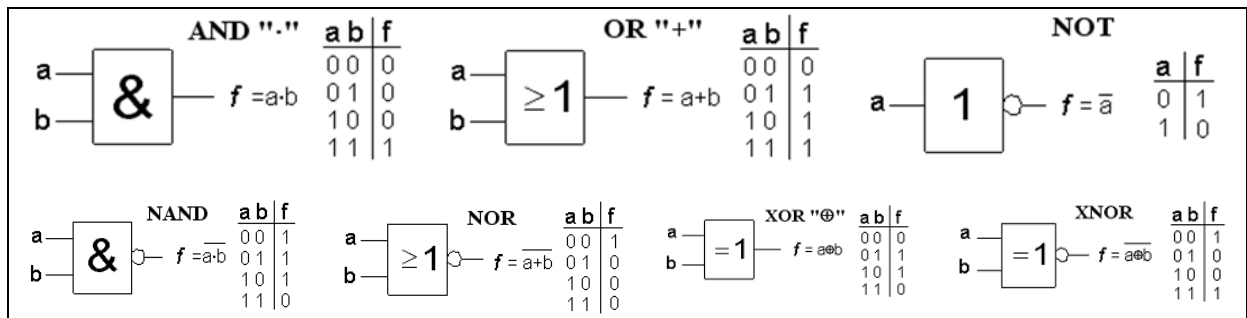
- $(x + y)(\bar{x} + z)$
- $(x + \bar{y} + xy)(x + \bar{y})xy$
- $x(1 + \bar{xy}) + \bar{x}$

### 4.4

Förenkla nedanstående uttryck så långt som möjligt.

$$\overline{(a + b + \bar{c})(a + \bar{b} + \bar{c})(\bar{a} + \bar{bc} + \bar{bc})}$$

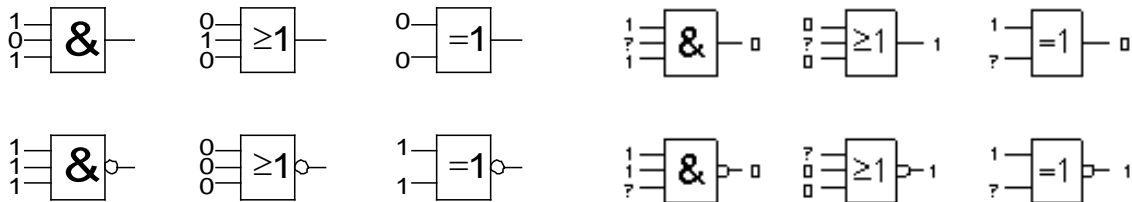
# Grindar



## 4.5

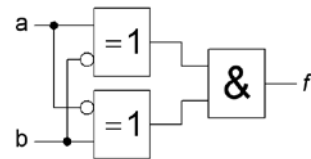
a) Ange **utsignalen** 1/0 för följande sex grindtyper när insignalerna är de som angivits i figuren.

b) Ange **insignalen** 1/0 för följande sex grindtyper när utsignalerna är de som angivits i figuren.



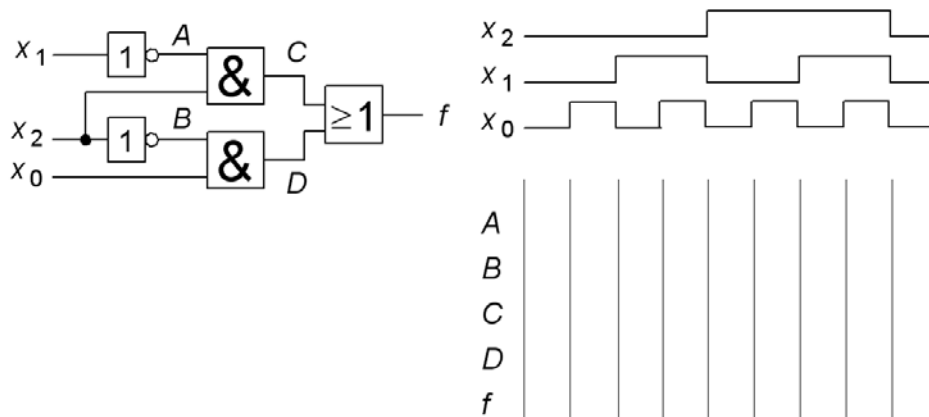
## 4.6

Förenkla  $f(a,b)$  som realiseras av figurens grindnät, så långt som möjligt, och ange funktionens namn.



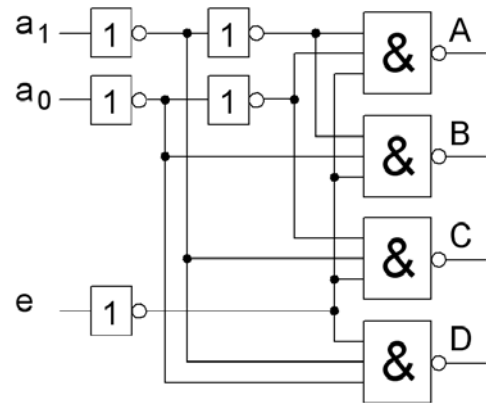
## 4.7

a) Rita tidsdiagram över signalerna  $A, B, C, D, f$ . Insignalerna till  $x_0, x_1,$  och  $x_2$  har frekvensförhållandet 4: 2:1 för att "svepa" igenom sanningstabellens *alla* kombinationer i "rätt" ordning.  
 b) Skriv upp sanningstabellen för funktionen  $f$ .



### 4.8

Ange de logiska uttrycken för A, B, C och D.



### 4.9

Förenkla det sammansatta uttrycken nedan så långt som möjligt.

a)  $x_2 \oplus x_1 \oplus x_1 x_2$     b)  $x_2 x_1 \oplus (x_2 + x_1)$

### 4.10

Visa att

a)  $\overline{x_2 \oplus x_1} = \overline{x_2} \oplus x_1 = x_2 \oplus \overline{x_1}$     b)  $x_2 \oplus x_1 = \overline{x_2} \oplus \overline{x_1}$

### 4.11

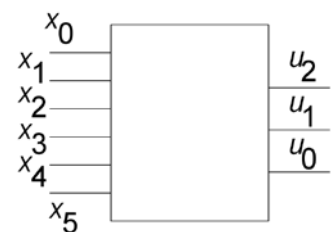
Här visas de Europeiska grindsymbolerna. Amerikas dominans inom halvledarområdet gör att man även måste känna till de amerikanska symbolerna. Namnge grindarna och rita de motsvarande amerikanska grindsymbolerna.

|  |  |  |  |  |  |  |
|--|--|--|--|--|--|--|
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

### 4.12

Ett kombinatoriskt nät med sex insignaler  $x_5, x_4, x_3, x_2, x_1, x_0$  och tre utsignaler  $u_2, u_1, u_0$ , beskrivs med text på följande sätt:

- $u_0 = 1$  om och endast om "antingen både  $x_0$  och  $x_2$  är 0 eller  $x_4$  och  $x_5$  är olika"
- $u_1 = 1$  om och endast om " $x_0$  och  $x_1$  är lika och  $x_5$  är inversen av  $x_2$ "
- $u_2 = 0$  om och endast om " $x_0$  är 1 och någon av  $x_1 \dots x_5$  är 0"



Beskriv nätet med Boolesk algebra och operationerna AND OR NOT XOR i stället.



## Sanningstabellen, SP och PS -form, fullständig logik

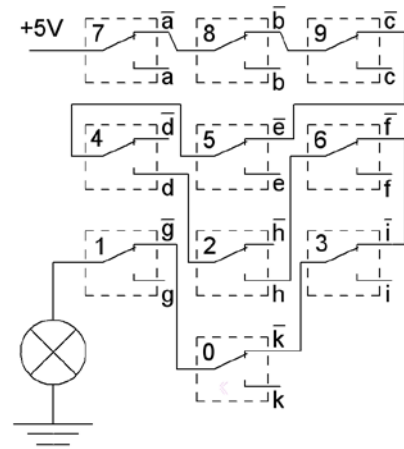
### 5.1

I figuren visas ett enkelt "kodlås" med 10 st växlingskontakter. För en viss kombination av samtidigt nedtryckta kontakter lyser lampan.

a) Vilken kombination?

b) Ange den logiska funktionen för lysande lampa. Variablernas beteckningar står i figuren ( $a \dots k$ ).

$f =$



### 5.2

En logisk funktion har följande sanningstabell:

| $a$ | $b$ | $c$ | $f$ |
|-----|-----|-----|-----|
| 0   | 0   | 0   | 1   |
| 0   | 0   | 1   | 0   |
| 0   | 1   | 0   | 0   |
| 0   | 1   | 1   | 0   |
| 1   | 0   | 0   | 1   |
| 1   | 0   | 1   | 1   |
| 1   | 1   | 0   | 0   |
| 1   | 1   | 1   | 1   |

Ange funktionen på PS-normalform (produkter av summatermer):

$f(a, b, c) =$

Ange funktionen på SP-normalform (summa av produkttermer):

$f(a, b, c) =$

### 5.3

En minimerad funktion är angiven på SP form (Summa av Produkter).

Ange samma funktion med mintermer som SP, respektive med maxtermer som PS (Produkt av Summor).

$$f(x, y, z) = x\bar{y} + y\bar{z} + \bar{x}z$$

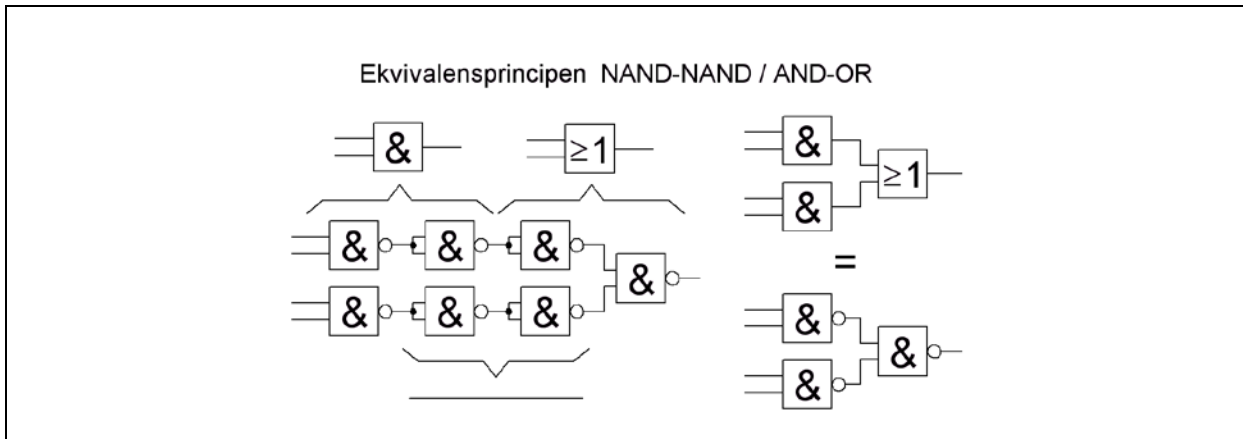
### 5.4

En funktion är angiven med en blandning av produkter och delsummor.

Ange samma funktion som SP (Summa av Produkter) med mintermer, respektive PS (Produkt av Summor) med maxtermer.

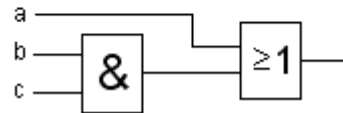
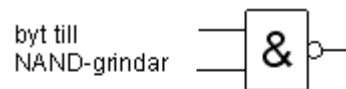
$$f(x, y, z) = (x + \bar{y})(xyz + \bar{y}(x + z)) + xy\bar{z}(x + \bar{y})$$

## Ekvivalens AND-OR / NAND-NAND och OR-AND / NOR-NOR



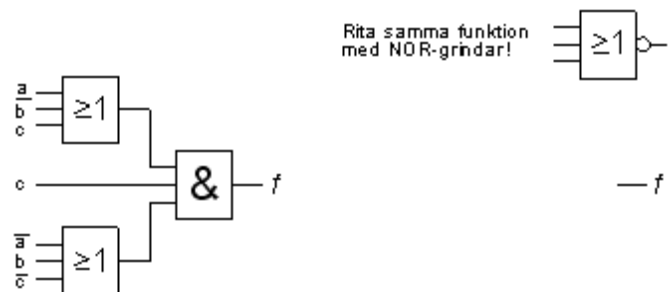
### 5.5

Rita vidstående AND/OR-nät som ett NAND/NAND-nät.



### 5.6

Rita vidstående OR/AND-nät som ett NOR/NOR nät.



### 5.7

a) Skriv upp sanningstabellen för en krets med fyra ingångar som definierar jämn paritet; dvs. kretsens utgång är "1" när ett jämnt antal av ingångarna samtidigt är "1".

b) Implementera denna funktion med så få NOR-grindar som möjligt.

# Karnaughdiagrammet

## 6.1

Gör bästa möjliga hoptagningar i Karnaughdiagrammet. Ange den minimerade funktionen på SP-form.

$f =$

|   |   |    |    |    |    |
|---|---|----|----|----|----|
|   |   | cd |    |    |    |
|   |   | 00 | 01 | 11 | 10 |
| a | b | 0  | 1  | 1  | 0  |
|   |   | 0  | 0  | 1  | 0  |
| 1 | 1 | 0  | 1  | 1  | 0  |
|   |   | 1  | 0  | 0  | 1  |

## 6.2

Gör bästa möjliga hoptagningar i Karnaughdiagrammet. Ange den minimerade funktionen på SP-form.

$f =$

|   |   |    |    |    |    |
|---|---|----|----|----|----|
|   |   | cd |    |    |    |
|   |   | 00 | 01 | 11 | 10 |
| a | b | 0  | 1  | 0  | 0  |
|   |   | 0  | 0  | 0  | 0  |
| 1 | 1 | 0  | 1  | 1  | 1  |
|   |   | 1  | 1  | 0  | 0  |

## 6.3

Placera ut följande funktion i Karnaughdiagrammet.

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

Försök att göra bättre hoptagningar. Ange därefter den "minimerade" funktionen:

$f =$

|   |   |    |    |    |    |
|---|---|----|----|----|----|
|   |   | cd |    |    |    |
|   |   | 00 | 01 | 11 | 10 |
| a | b | 0  |    |    |    |
|   |   | 0  |    |    |    |
| 1 | 1 | 0  |    |    |    |
|   |   | 1  |    |    |    |

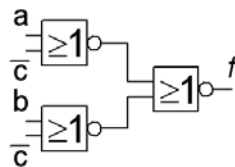
## 6.4

Överst i figuren till höger finns ett NOR-NOR nät.

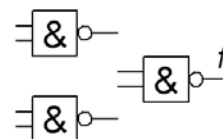
Analysera detta nät och för in sanningstabellen i Karnaughdiagrammet.

Gör hoptagningar i Karnaughdiagrammet och realisera funktionen med NAND-grindar nederst i figuren i stället.

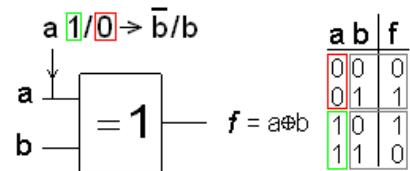
Variablerna  $a$   $b$  och  $c$  finns tillgängliga både normala och inverterade.



|   |   |    |    |    |    |
|---|---|----|----|----|----|
|   |   | bc |    |    |    |
|   |   | 00 | 01 | 11 | 10 |
| a | 0 | 0  | 1  | 3  | 2  |
|   | 1 | 4  | 5  | 7  | 6  |



PLD-kretsar har ofta en XOR-grind på utgången så att man vid behov skall kunna invertera funktionen. Man kan då välja mellan att ta ihop 0:or eller 1:or efter vad som är fördelaktigast.



### 6.5

En funktion med fyra variabler definieras med mintermer på SP-form. Använd Karnaughdiagram för att minimera funktionen. Minimera även funktionens invers.

$$f(x_3, x_2, x_1, x_0) = \sum m(0, 2, 4, 8, 10, 12) \quad f = ? \quad \bar{f} = ?$$

### 6.6

En funktion med fyra variabler definieras med maxtermer på PS-form. Använd Karnaughdiagram för att minimera funktionen. Minimera även funktionens invers.

$$f(x_3, x_2, x_1, x_0) = \prod M(0, 1, 4, 5, 10, 11, 14, 15) \quad f = ? \quad \bar{f} = ?$$

### 6.7

En funktion med fyra variabler definieras med mintermer på SP-form. Använd Karnaughdiagram för att minimera funktionen. Minimera även funktionens invers.

$$f(x_3, x_2, x_1, x_0) = \sum m(0, 2, 3, 4, 6, 7, 8, 9, 10, 12, 13, 14) \quad f = ? \quad \bar{f} = ?$$

### 6.8

Ibland kan problemställningen vara sådan att vissa ingångskombinationer är "omöjliga" och därför *inte* kan inträffa. Sådana mintermer (eller maxtermer) betecknar man med d ("don't care") och använder dom som ettor eller nollor allt efter vad som passar bäst för att få så stora hoptagningar som möjligt.

$$f(x_3, x_2, x_1, x_0) = \sum m(3, 5, 7, 11) + d(6, 15) \quad f = ? \quad \bar{f} = ?$$

### 6.9

$$f(x_3, x_2, x_1, x_0) = \sum m(1, 4, 5) + d(2, 3, 6, 7, 8, 9, 12, 13) \quad f = ? \quad \bar{f} = ?$$

### 6.10

En funktion med **fem** variabler definieras

$$f(x_4, x_3, x_2, x_1, x_0) = \sum m(9, 11, 12, 13, 14, 15, 16, 18, 24, 25, 26, 27)$$

se ifylld sanningstabell.

Använd Karnaughdiagram-metoden för att minimera funktionen. Minimera även funktionens invers.

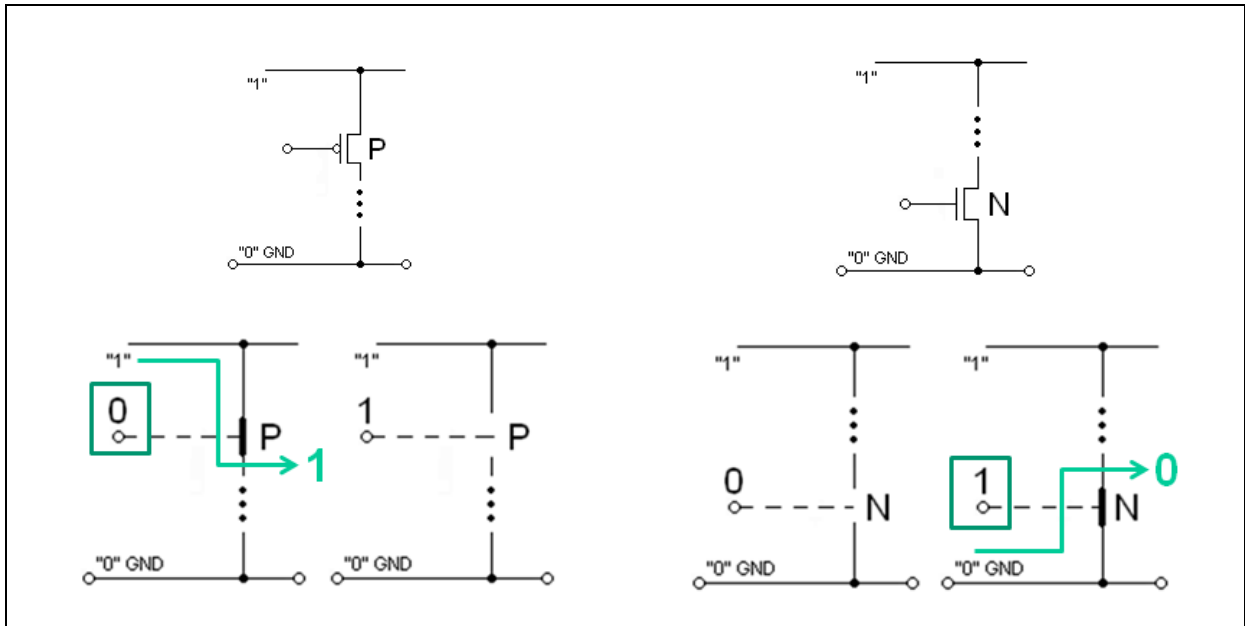
$$f(x_4, x_3, x_2, x_1, x_0) \quad f = ? \quad \bar{f} = ?$$

|    | $x_4$ | $x_3$ | $x_2$ | $x_1$ | $x_0$ | $f$ |    | $x_4$ | $x_3$ | $x_2$ | $x_1$ | $x_0$ | $f$ |
|----|-------|-------|-------|-------|-------|-----|----|-------|-------|-------|-------|-------|-----|
| 0  | 0     | 0     | 0     | 0     | 0     | 0   | 16 | 1     | 0     | 0     | 0     | 0     | 1   |
| 1  | 0     | 0     | 0     | 0     | 1     | 0   | 17 | 1     | 0     | 0     | 0     | 1     | 0   |
| 2  | 0     | 0     | 0     | 1     | 0     | 0   | 18 | 1     | 0     | 0     | 1     | 0     | 1   |
| 3  | 0     | 0     | 0     | 1     | 1     | 0   | 19 | 1     | 0     | 0     | 1     | 1     | 0   |
| 4  | 0     | 0     | 1     | 0     | 0     | 0   | 20 | 1     | 0     | 1     | 0     | 0     | 0   |
| 5  | 0     | 0     | 1     | 0     | 1     | 0   | 21 | 1     | 0     | 1     | 0     | 1     | 0   |
| 6  | 0     | 0     | 1     | 1     | 0     | 0   | 22 | 1     | 0     | 1     | 1     | 0     | 0   |
| 7  | 0     | 0     | 1     | 1     | 1     | 0   | 23 | 1     | 0     | 1     | 1     | 1     | 0   |
| 8  | 0     | 1     | 0     | 0     | 0     | 0   | 24 | 1     | 1     | 0     | 0     | 0     | 1   |
| 9  | 0     | 1     | 0     | 0     | 1     | 1   | 25 | 1     | 1     | 0     | 0     | 1     | 1   |
| 10 | 0     | 1     | 0     | 1     | 0     | 0   | 26 | 1     | 1     | 0     | 1     | 0     | 1   |
| 11 | 0     | 1     | 0     | 1     | 1     | 1   | 27 | 1     | 1     | 0     | 1     | 1     | 1   |
| 12 | 0     | 1     | 1     | 0     | 0     | 1   | 28 | 1     | 1     | 1     | 0     | 0     | 0   |
| 13 | 0     | 1     | 1     | 0     | 1     | 1   | 29 | 1     | 1     | 1     | 0     | 1     | 0   |
| 14 | 0     | 1     | 1     | 1     | 0     | 1   | 30 | 1     | 1     | 1     | 1     | 0     | 0   |
| 15 | 0     | 1     | 1     | 1     | 1     | 1   | 31 | 1     | 1     | 1     | 1     | 1     | 0   |

|       |           |             |    |    |    |
|-------|-----------|-------------|----|----|----|
|       |           | $\bar{x}_4$ |    |    |    |
|       | $x_1 x_0$ | 00          | 01 | 11 | 10 |
| $x_3$ | 0         | 0           | 1  | 3  | 2  |
| $x_2$ | 0         | 4           | 5  | 7  | 6  |
|       | 1         | 12          | 13 | 15 | 14 |
|       | 1         | 8           | 9  | 11 | 10 |
|       | 0         |             |    |    |    |

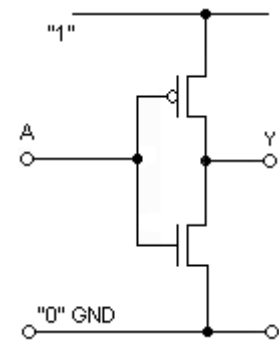
|       |           |       |    |    |    |
|-------|-----------|-------|----|----|----|
|       |           | $x_4$ |    |    |    |
|       | $x_1 x_0$ | 00    | 01 | 11 | 10 |
| $x_3$ | 0         | 16    | 17 | 19 | 18 |
| $x_2$ | 0         | 20    | 21 | 23 | 22 |
|       | 1         | 28    | 29 | 31 | 30 |
|       | 1         | 24    | 25 | 27 | 26 |
|       | 0         |       |    |    |    |

# MOS-transistorn och digitala kretsar



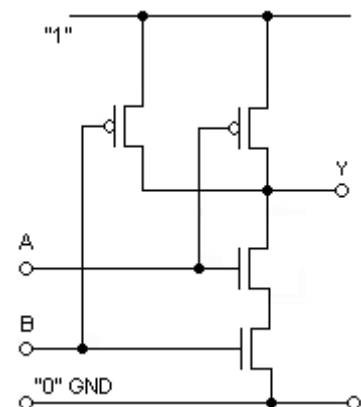
## 7.1

Identifiera transistorernas beteenden och skriv upp sanningstabellen för  $Y(A)$ .  
Vilken logisk funktion är det?



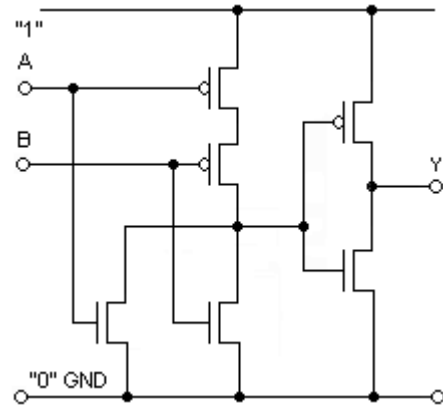
## 7.2

Identifiera transistorernas beteenden och skriv upp sanningstabellen för  $Y(A,B)$ .  
Vilken logisk funktion är det?



### 7.3

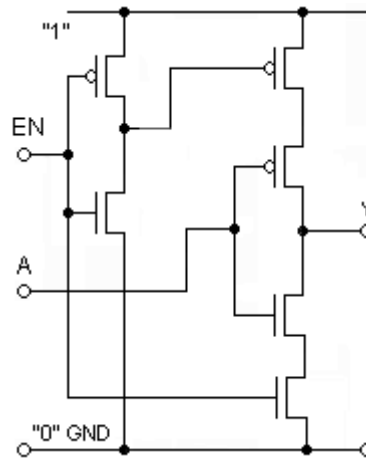
Identifiera transistorernas beteenden och skriv upp sannings-tabellen för  $Y(A,B)$ .  
Vilken logisk funktion är det?



### 7.4

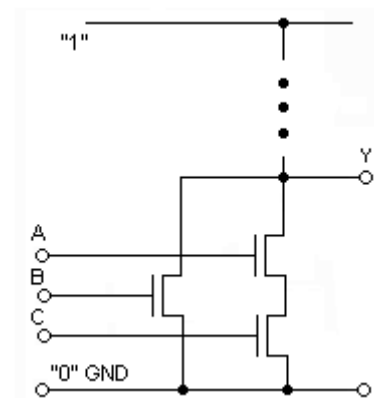
Studera kretsen och beskriv funktionen. Vilken roll spelar signalen EN? Vilket samband gäller mellan Y och A?  $Y(B)$ .

Hur många "tillstånd" kan utgången ha?



### 7.5

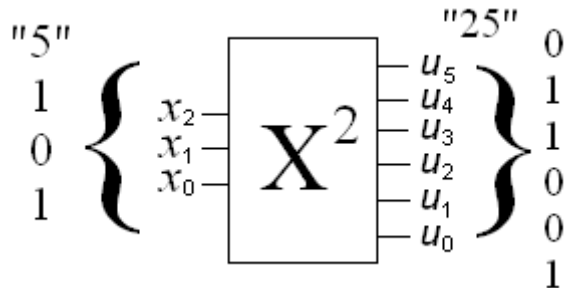
Figuren visar ena halvan av en CMOS krets. Rita dit den andra halvan, som innehåller PMOS transistorerna. Ange den logiska funktionen  $Y(A,B,C)$ .



## Kombinatoriska krets nät

### 8.1.

Tag fram de Boolska uttrycken på minimerad SP-form för ett kombinatoriskt nät som omvandlar ett trebitars binärkodat tal  $X$  ( $x_2, x_1, x_0$ ) till ett binärkodat sexbitstal  $U$  ( $u_5, u_4, u_3, u_2, u_1, u_0$ ) som är lika med kvadraten på talet, dvs.  $U = X^2$ . Använd Karnaughdiagram.

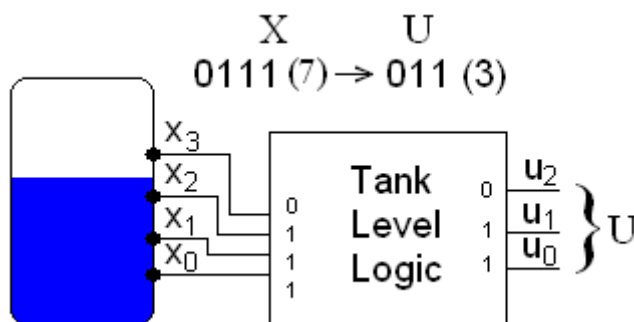


### 8.2

Ett övervakningssystem till en vattentank består av fyra nivågivare  $x_3, x_2, x_1, x_0$ . Signalerna från dessa bildar ett binärt fyr-bitstal  $X$ . Ett logiknät "Tank Level Logic" omkodar  $X$  till ett trebitars tal  $U$  ( $u_2, u_1, u_0$ ) som presenterar nivån som ett binärt tal mellan 0 och 4.

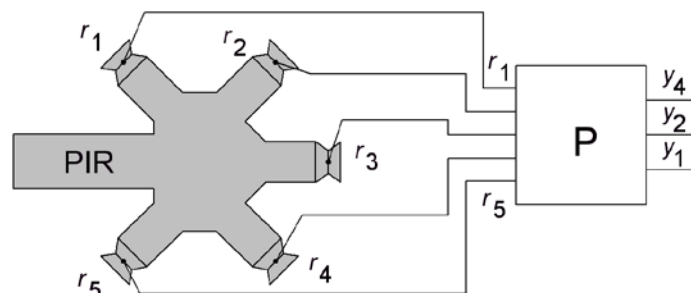
Konstruera logiknätet. Ange de Boolska uttrycken på minimerad SP-form. Utnyttja att det är många av insignalskombinationerna som *aldrig* kan uppkomma! Ingångsvariablerna finns tillgängliga i både inverterad och oinverterad form från nivågivarna.

Använd ekvivalensprincipen för att ta fram logiknätet med enbart NAND-grindar.



### 8.3

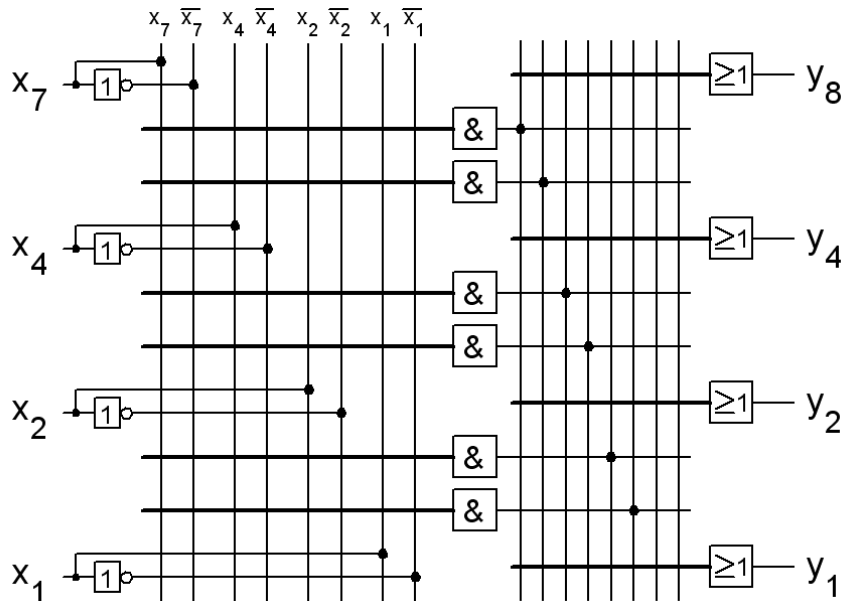
En PIR på en flygplats har fem anslutningsramper (GATES). Ramperna numreras 1...5. Vid varje ramp finns en givare som ger en utsignal  $r_i = 0$  om ett flygplan är anslutet till rampen, annars 1. Ett kombinatoriskt nät,  $P$ , hjälper flygtrafikledaren att dirigera anländande flygplan till lediga ramper. Nätet har insignalerna  $r_1, r_2, r_3, r_4, r_5$  och utsignalerna  $y_4, y_2, y_1$ . Kombinationen av utsignalerna  $y_4, y_2, y_1$  skall i binärkod ge numret på den ramp med högst ordningsnummer som är ledig. Om *ingen* ramp är ledig används numret  $(y_4, y_2, y_1) = (1, 1, 1)$ . Minimera varje utgång för sig.





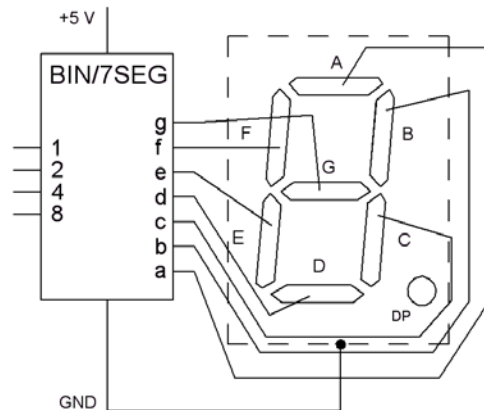
### 8.4

De decimala siffrorna 0 till 9 kan kodas i så kallad 7421-kod. Det är en viktad binärkod med positionsvikterna 7, 4, 2, och 1. Där två kombinationer av de viktade talen kan ge samma värde väljes det med det minsta antalet ettor. (7421-Koden har egenskapen att den kodar siffrorna 0 till 9 med minimalt antal ettor, totalt 14 st).  
 Konstruera en krets som översätter från 7421-koden till den mer vanliga BCD-koden (8421-kod).  
 Använd en PLD-krets av AND-OR typ. Både AND-planet och OR-planet kan programmeras var för sig.  
 Rita kryss i figuren nedan för att visa vilka programmerade förbindelser som skall göras. Grindarnas ingångar har ritats med "förenklat" rit sätt".



### 8.5

En 7-segmentavkodare avkodar ett binärt 4-bitstal till motsvarande segmentbild för siffrorna 0...9 (eller hexadecimalt 0 ... F).  
 Ställ upp sanningstabellen, och ange ett minimerat logiskt samband för tex. segmentet "G".



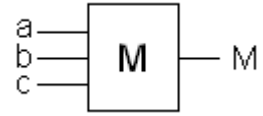
### 8.6

Visa hur en 4-1 multiplexor kan användas som en funktionsgenerator och tex. generera OR-funktionen.

### 8.7

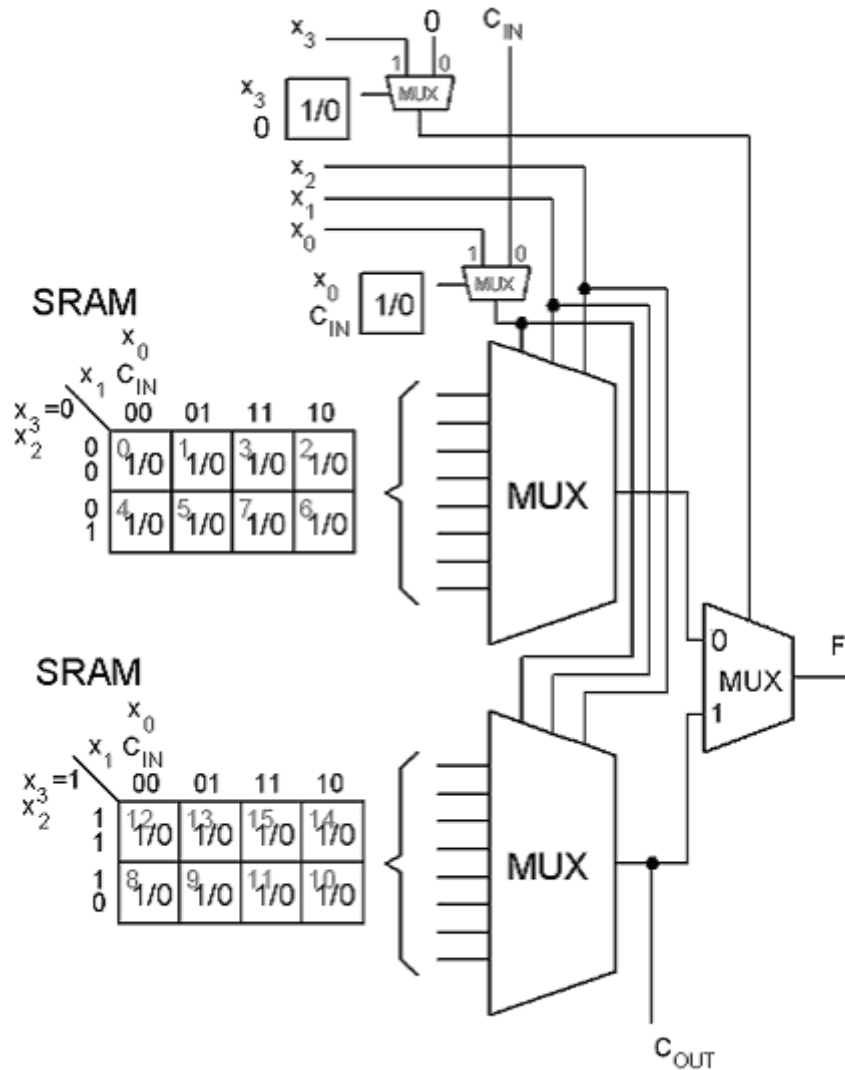
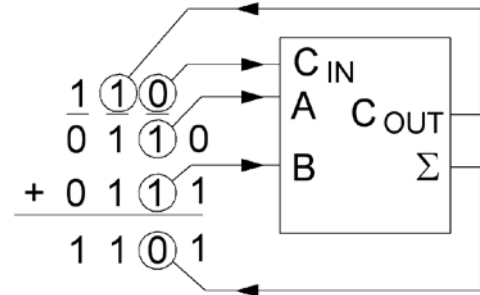
En majoritetsgrind antar på utgången samma värde som en *majoritet* av ingångarna. Grinden kan tex. användas i feltolerant logik, eller till bildbehandlingskretsar.

- Ställ upp grindens sanningstabell och minimera funktionen med Karnaughdiagram. Realisera funktionen med AND-OR grindar.
- Realisera majoritetsgrinden med en 8:1 MUX.
- Använd Shannon dekomposition och realisera majoritetsgrinden med en 2:1 MUX och grindar.
- Realisera majoritetsgrinden med bara 2:1 MUXar.



### 8.8

Ställ upp heladderarens sanningstabell. Visa hur en heladderare realiseras i en FPGA-krets. Logikelementen i en FPGA har möjlighet att kaskadkoppla  $C_{OUT}$  och  $C_{IN}$  mellan "grannarna". Visa innehållet i SRAM-cellerna (LUT, LookUp Table).

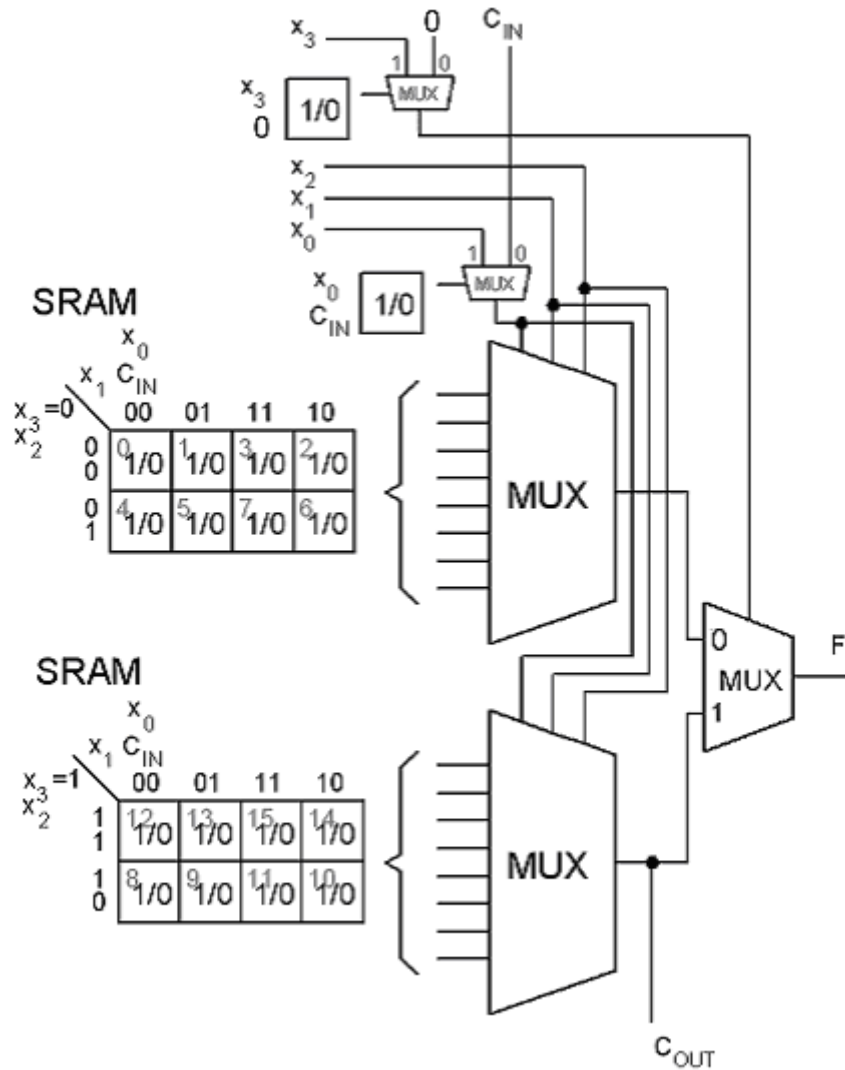
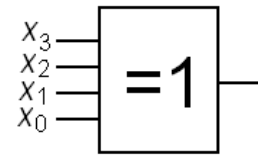


### 8.9

Visa hur en 4 ingångars exorgrind (XOR) realiseras i en FPGA-krets. Visa innehållet i SRAM-cellerna (LUT, LookUp Table).

XOR

|       |           |    |    |    |    |
|-------|-----------|----|----|----|----|
|       | $x_1 x_0$ | 00 | 01 | 11 | 10 |
| $x_3$ | 0         | 0  | 1  | 3  | 2  |
| $x_2$ | 0         | 0  | 1  | 3  | 2  |
|       | 0         | 4  | 5  | 7  | 6  |
|       | 1         | 12 | 13 | 15 | 14 |
|       | 1         | 8  | 9  | 11 | 10 |



### 8.10

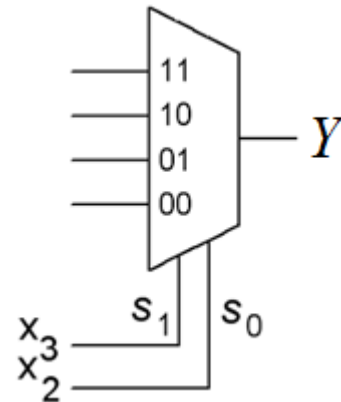
Den Booleska funktionen  $Y$  av fyra variabler  $x_3, x_2, x_1, x_0$  definieras av sanningstabellen.

a) Använd Karnaughdiagrammet för att konstruera ett minimalt nät för funktionen (utnyttja ”–” som don't care). Använd valfria grindar.

b) Realisera funktionen  $Y$  med en 4:1 multiplexor och (valfri) grindlogik. Använd  $x_3$  och  $x_2$  som multiplexorns dataväljarsignaler.

| $x_3x_2x_1x_0$ | $Y$  | $x_3x_2x_1x_0$ | $Y$ |      |   |
|----------------|------|----------------|-----|------|---|
| 0              | 0000 | –              | 8   | 1000 | – |
| 1              | 0001 | –              | 9   | 1001 | – |
| 2              | 0010 | 1              | 10  | 1010 | 1 |
| 3              | 0011 | 0              | 11  | 1011 | 0 |
| 4              | 0100 | 0              | 12  | 1100 | 0 |
| 5              | 0101 | 1              | 13  | 1101 | 1 |
| 6              | 0110 | 0              | 14  | 1110 | 1 |
| 7              | 0111 | 1              | 15  | 1111 | 0 |

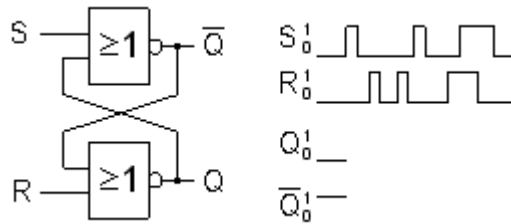
|       |   | $x_1x_0$ |    |    |    |
|-------|---|----------|----|----|----|
|       |   | 00       | 01 | 11 | 10 |
| $x_3$ | 0 | 0        | 1  | 3  | 2  |
|       | 1 | 4        | 5  | 7  | 6  |
| $x_2$ | 0 | 0        | 1  | 3  | 2  |
|       | 1 | 4        | 5  | 7  | 6  |



## Sekvenskretsar, låskretsar och klockade vippor

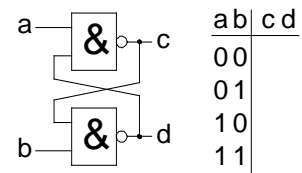
### 9.1

Kompletera tidsdiagrammet för utsignalerna  $Q$  och  $\bar{Q}$ . Avståndet mellan pulserna är *mycket längre* än grindfördröjningen.  
(Ledning, vad är låsande insignal för NOR-grindar)



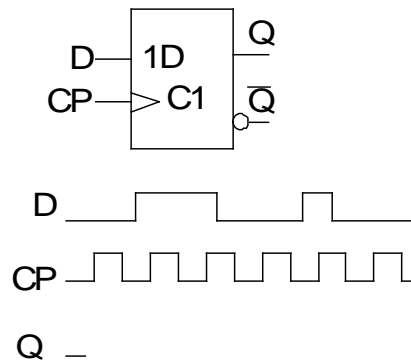
### 9.2

Du känner säkert igen låskretsen till höger. Här har de vanliga beteckningarna bytts ut mot **a b c d**. Fyll i sanningstabellen.



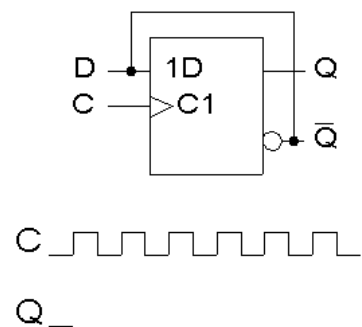
### 9.3

Rita tidsdiagrammet för utsignalen  $Q$ , för D-vippan.



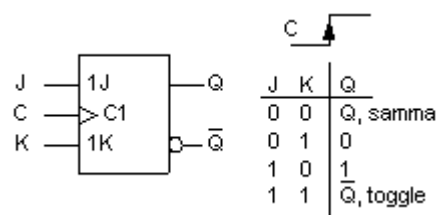
### 9.4

Rita in  $Q$  i detta tidsdiagram.



### 9.5

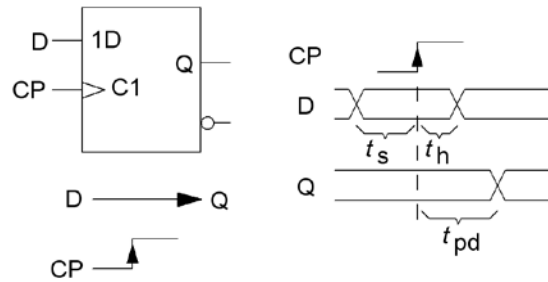
JK-vippan var en äldre typ av "universalvippa". Visa hur den kan användas som T-vippa och som D-vippa.



### Flip-Flop Timing Parameters.

Vippan laddas med data vid klockpulsens positiva flank, men datat måste vara stabilt tiden  $t_s$  före klockpulsens flank och även tiden  $t_h$  efter. Datat återfinns på utgången efter tiden  $t_{pd}$ . ( $t_{pd}$  kan vara olika för 0→1 respektive 1→0).

Om dessa tider inte respekteras blir vippans funktion osäker.

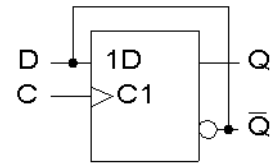


### 9.6

Vilken är den högsta klockfrekvens man kan använda till kretsen i figuren utan att riskera felfunktion?

Antag att

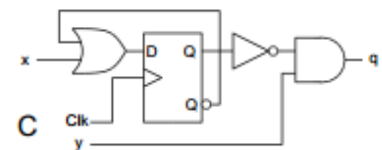
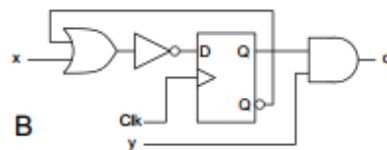
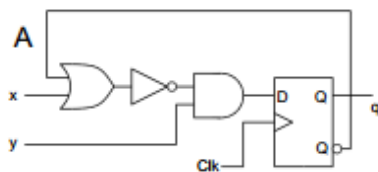
$$t_s = 20 \text{ ns} \quad t_h = 5 \text{ ns} \quad t_{pd} = 30 \text{ ns}$$



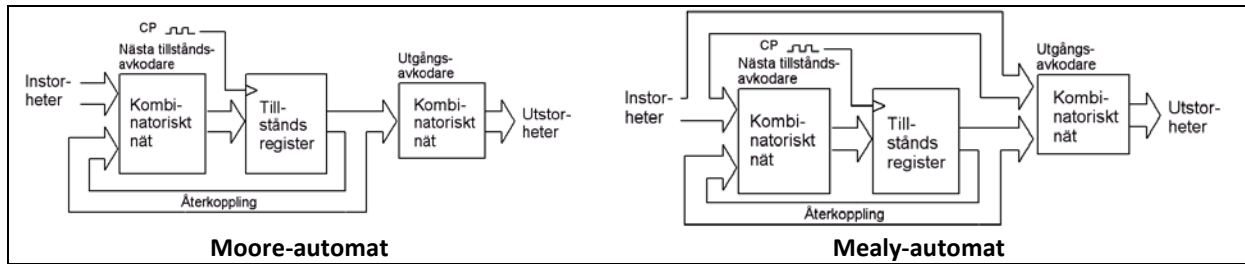
### 9.7

Figuren visar tre olika tillståndsmaskiner. Ange vilken tillståndsmaskin (A, B eller C) som kan operera vid högst klockfrekvens. Markera den kritiska vägen (den väg som begränsar klockfrekvensen) i denna figur samt beräkna periodtiden för klocksignalen Clk.

$$t_{AND} = 0,4 \text{ ns}, t_{OR} = 0,4 \text{ ns}, t_{NOT} = 0,1 \text{ ns}, t_{setup} = 0,3 \text{ ns}, t_{dq} = 0,4 \text{ ns}$$

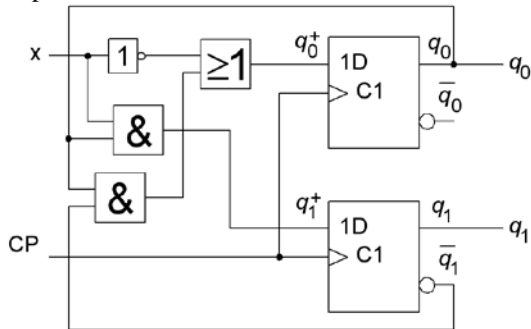


# Sekvensnät, automater



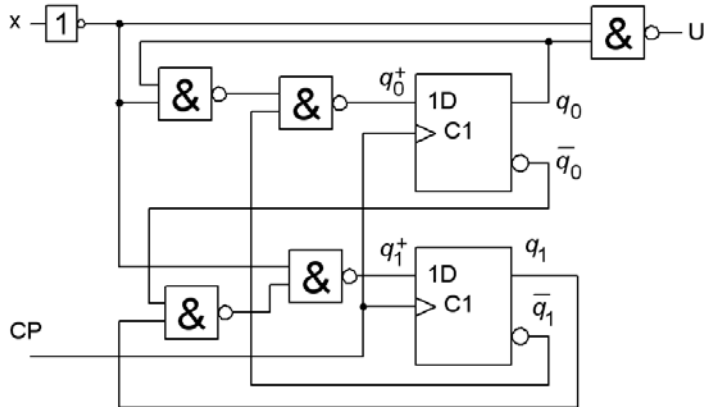
## 10.1

Bestäm tillståndsdigram och tillståndstabell för sekvenskretsen. Vilken av modellerna Mealy eller Moore passar in på kretsen?



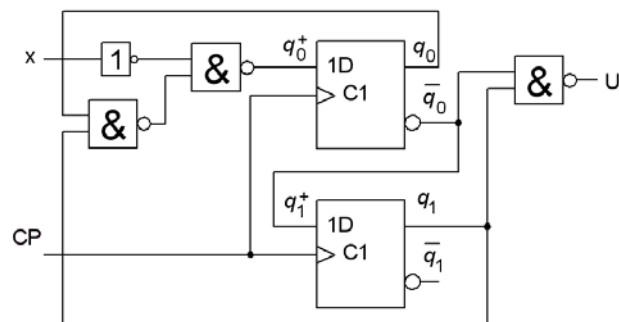
## 10.2

Bestäm tillståndsdigram och tillståndstabell för sekvenskretsen. Vilken av modellerna Mealy eller Moore passar in på kretsen?



## 10.3

Bestäm tillståndsdigram och tillståndstabell för sekvenskretsen. Vilken av modellerna Mealy eller Moore passar in på kretsen?



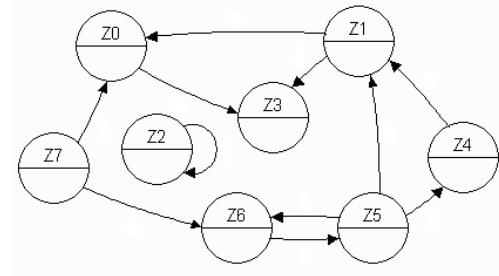
### 10.4

Finns det några stopptillstånd, förlusttillstånd eller isolerade tillstånd i tillståndsdiagrammet till höger?

Stopptillstånd:

Förlusttillstånd:

Isolerade tillstånd:



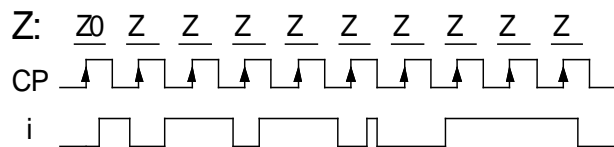
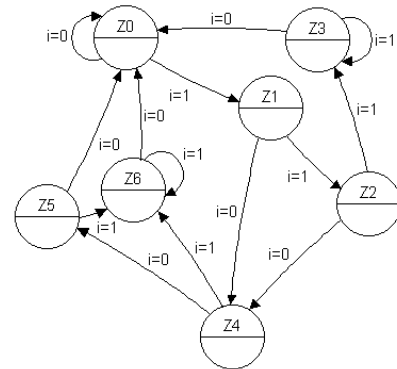
### 10.5

Till höger finns ett tillståndsdiagram för en Moore-automat. (Den ska upptäcka dubbeltryckning).

En apa råkar få tag i tryckknappen för ingångssignalen  $i$ , och trycker enligt tidsdiagrammet nedan.

Moore-automaten har vippor som triggas av klockpulsens positiva flank. Antag att den från början står i starttillståndet  $Z_0$ .

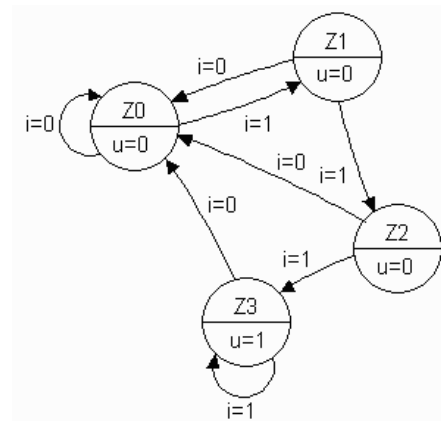
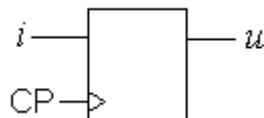
Fyll i vilka tillstånd man hamnar i.



### 10.6

Konstruera en Moore-automat som kräver att insignalen är lika med ett ( $i = 1$ ) under tre på varandra följande klockpulser för att utsignalen skall bli ett ( $u = 1$ ). Så fort insignalen blir noll under en klockpuls ( $i = 0$ ) skall kretsen återgå till att utsignalen är noll ( $u = 0$ ). Se tillståndsdiagrammet. Välj Graykod för tillståndskodningen. ( $Z_0=00$ ,  $Z_1=01$ ,  $Z_2=11$ ,  $Z_3=10$ ). använd AND-OR grindar.

( Kretsen är en säkerhetskrets som skall förhindra "falsklarm". Vi kan kalla principen för "truga kaka" efter den svenska seden att man inte kan tacka nej till en bjuden kaka om den erbjuds tre gånger i rad ... )

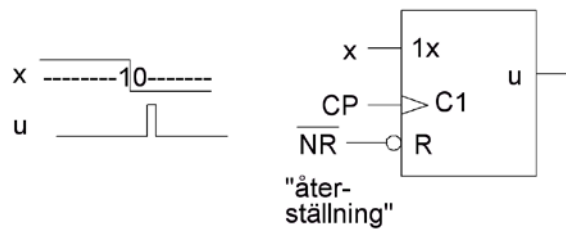




### 10.7

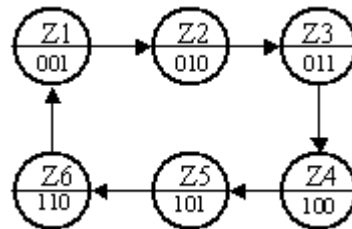
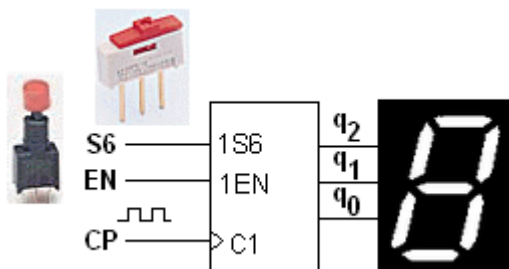
Konstruera ett sekvensnät som upptäcker när signalen  $x$  har en övergång mellan  $1 \rightarrow 0$  och då signalerar detta med att  $u = 1$  i det nästföljande klockpulsintervallet för att sedan bli 0 under resten av sekvensen.

Med en asynkron återställningspuls (NR aktiv låg) skall kretsen kunna "resettas" så att den bevakar signalen på nytt.



- Rita tillståndsdigram för en automat av Moore typ för sekvensnätet.
- Tag fram de boolska uttrycken för nästa tillståndsavkodaren och utgångsavkodaren för tre olika tillståndskodning:
  - "Binärkod"
  - "Graykod"
  - "One hot" kod
- Visa hur återställningssignalen NR ansluts till D-vippornas direktverkande PRE och CLR ingångar.

### 10.8



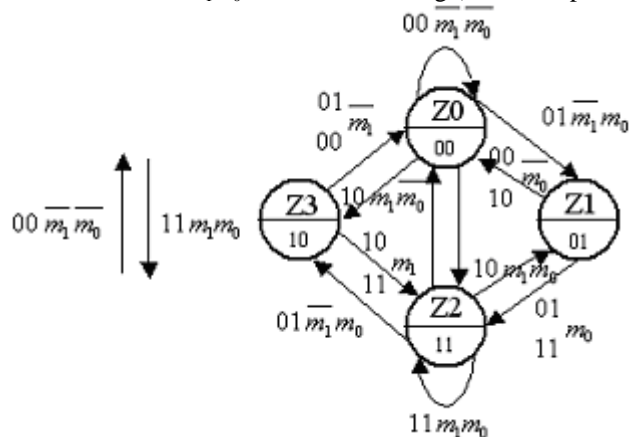
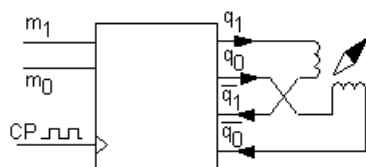
Konstruera en räknare som räknar  $\{ \dots 1, 2, 3, 4, 5, 6, 1 \dots \}$ . Räknesekvensen,  $q_2q_1q_0$ , är tänkt att visas på en 7-segmentdisplay, som ett tärningskast.

- Ange uttrycken för nästatillståndsavkodaren.
- Kompletera uttrycken med en signal EN som "fryser" tillståndet för  $EN = 0$  (släppt knapp). Räknaren skall räkna för  $EN = 1$  (nedtryckt knapp).
- Kompletera uttrycken med en signal S6 som när  $S6 = 1$  tvingar räknaren till tillståndet "6" (fusk-knappen). S6 är överordnad EN.

### 10.9

En stegmotor är en digital komponent som drivs med pulser. Stegmotorer brukar anslutas till räknare som räknar Gray-kod. Figurens räknare har dessutom en mode-ingång,  $m_1m_0$ .

- $m_1m_0 = 00 \rightarrow 0$ -ställning (fix position)
- $m_1m_0 = 01 \rightarrow$  upp-räkning (cw)
- $m_1m_0 = 10 \rightarrow$  ner-räkning (ccw)
- $m_1m_0 = 11 \rightarrow 1$ -ställning (annan fix position)

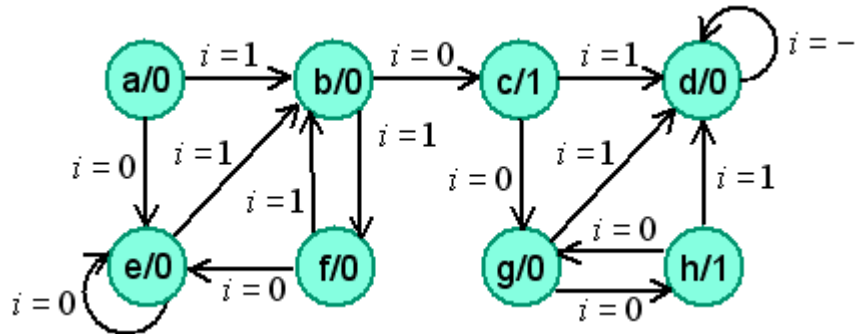


Ibland skriver man boolska vilkor i stället för siffror vid pilarna. I figuren används både vilkor och siffror. Skriv upp de minimerade uttrycken för räknarens nästatillståndsavkodare.

### 10.10

Detta tillståndsdigram gäller ett synkront sekvensnät.

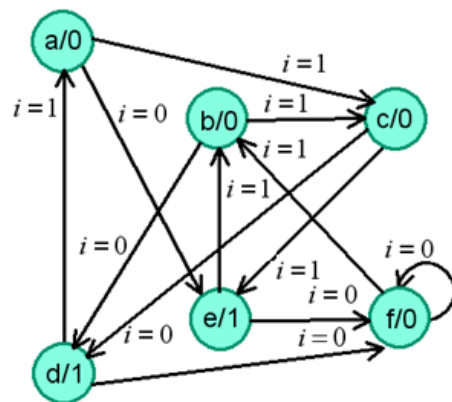
- Skriv tillståndstabell.
- Minimera antalet tillstånd.
- Skriv minimerad tillståndstabell
- Rita minimerat tillståndsdigram.



### 10.11

Detta tillståndsdigram gäller ett synkront sekvensnät.

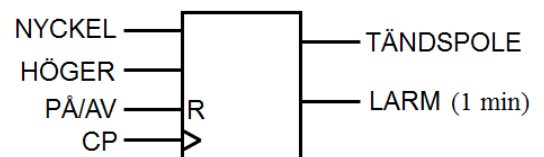
- Skriv tillståndstabell. Minimera antalet tillstånd.
- Skriv minimerad tillståndstabell och rita minimerat tillståndsdigram.



### 10.12

En teknolog bygger ett biltjuvlarm som en synkron Moore-automat. Larmet får sin ”säkerhet” av att vara hemligt och unikt. För att kunna starta bilen måste man manövrera bilens reglage i följande ordningsföljd:

- 1) Vrid på startnyckeln (tändning på)
- 2) Ställ körriktningsvisaren (blinkers) till höger
- 3) Vrid av startnyckeln (tändning av)
- 4) Ställ körriktningsvisaren i neutralläge (höger av)
- 5) Vrid på startnyckeln



Om man vid någon punkt i listan gör ”fel” hamnar man (fastnar) i larmtillståndet. Gör man allt rätt startar bilen (= fastnar man i tändspoletillståndet).

Sekvensnätet har också en ”dold” knapp som går till D-vippornas Reset, och som innebär att larmet kan kopplas PÅ/AV.

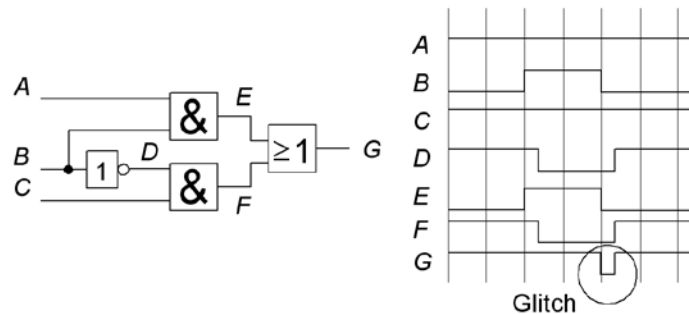
- Rita sekvensnätets tillståndsdigram.

# Asynkrona sekvensnät

## 11.1

Om signalerna passerar olika många grindsteg på vägen mot utgången kan kortvariga oönskade avvikelser från sanningstabellen uppkomma, så kallade "glitchar".

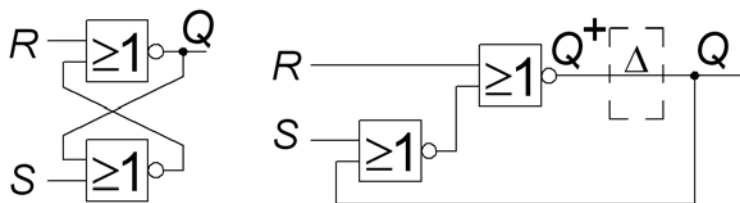
Visa i Karnaughdiagrammet hur man undviker dessa.



## 11.2

Till vänster i figuren visas en SR-låskrets som två "korskopplade" grindnät. Till höger är kretsen omritad som en "Moore"-automat. Det finns ingen klocksignal, och inget egentligt tillståndsregister. Alla grindfördröjningar som finns i nätet tänks placerade i symbolen  $\Delta$  mellan  $Q^+$  och  $Q$  som får en liknande funktion som D-vippan i ett synkront sekvensnät.

Analysera SR-kretsen på samma sätt som en Moore-automat.

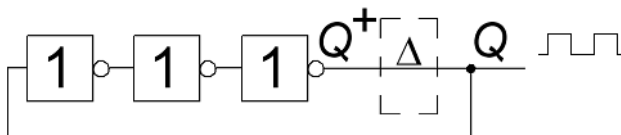


## 11.3

Visa att det blir ett instabilt nät – en oscillator – om ett udda antal inverterare återkopplas.

Antag att grindfördröjningen  $t_{pd}$  är 5 ns och att tre grindar kopplats som i figuren.

Vilket värde får oscillationsfrekvensen?



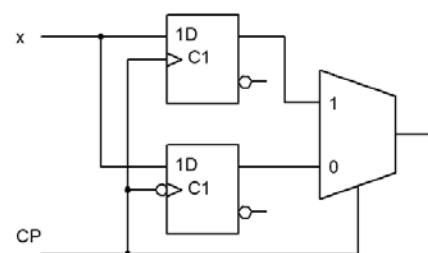
## 11.4

Analysera följande krets:

Rita ett tillståndsdigram.

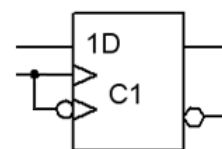
Betrakta kretsen som ett asynkront sekvensnät där klockpulsingången är en av de asynkrona ingångarna.

Vad har kretsen för funktion?



## 11.5

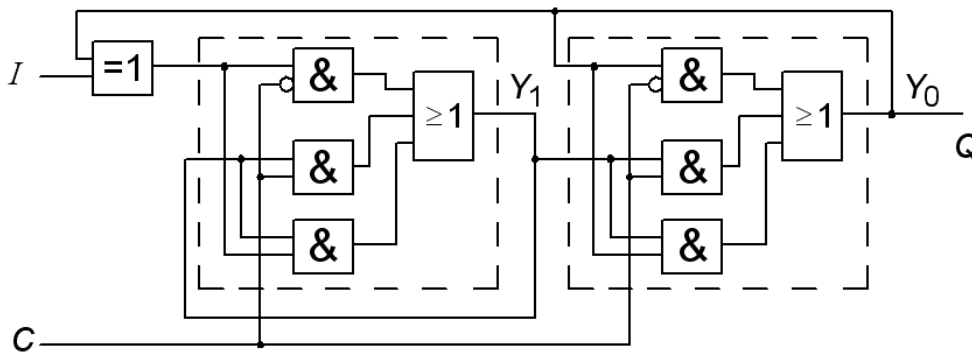
Konstruera en asynkron statemaskin som fungerar som en dubbelflankad D vippa (DETF), dvs vippan skall ändra värde både på den *positiva* och den *negativa* flanken av klockan.



- Härled FSMen.
- Ta fram flödestabellen och minimera den.
- Tilldela tillstånd (states), överför till Karnaughdiagram och härled de boolska uttrycken.
- Rita kretsen.

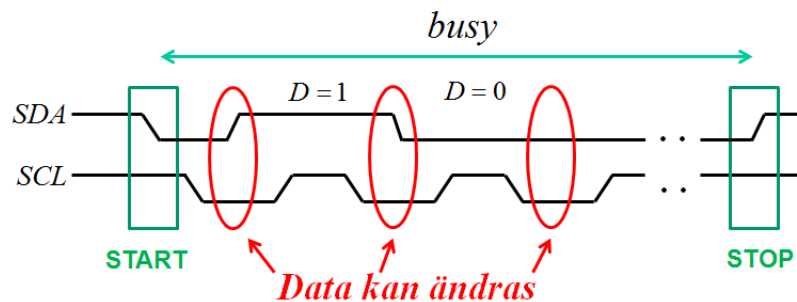
### 11.6

Analysera följande krets:



- Härled de booleska uttrycken för state variablerna  $Y_1$  och  $Y_0$ .
- Härled excitationstabellen. (Ledning: Vilken funktion finns i de två innersta looparna)
- Härled flödestabell, tilldela symboliska states och rita FSM:en.
- Identifiera funktionen hos den asynkrona kretsen. Vilken vippa motsvarar den?

### 11.7



Dataöverföring mellan olika chip inom elektronikutrustningar kan ske med den så kallade I2C-bussen. Den består av två ledningar SDA och SCL. I figuren ovan visas en principbild när ett fåtal bitar överföres, men det vanliga är att mycket större datamängder överföres.

Data  $D$  får bara ändras när  $SCL=0$ .

Positiv och negativ SDA-flank när  $SCL=1$  används som **unika** start och stopp-signaler för dataöverföringen. ( Under dataöverföringen förekommer därför *inga* sådana flanker ). (Innan stop-pulsen kan mottagaren "kvittera" motagningen – i figuren bortses från detta.)

För att kunna studera I2C-dataöverföringen vill man konstruera ett Moore-ekvivalent asynkront sekvensnät som ger utsignalen  $busy = 1$  under tiden från start-signalen fram till stopp-signalen. När ingen datakommunikation förekommer är  $busy = 0$ .



- Ställ upp en primitiv flödestabell
- Minimera flödestabellen
- Välj tillståndskod
- Ställ upp excitationstabellen (motivera att konstruktionen är fri från kritisk kapplöpning)
- Tag fram minimerade booleska uttryck (motivera hazardfrihet)

# Avkodning av minnen och I/O-kretsar

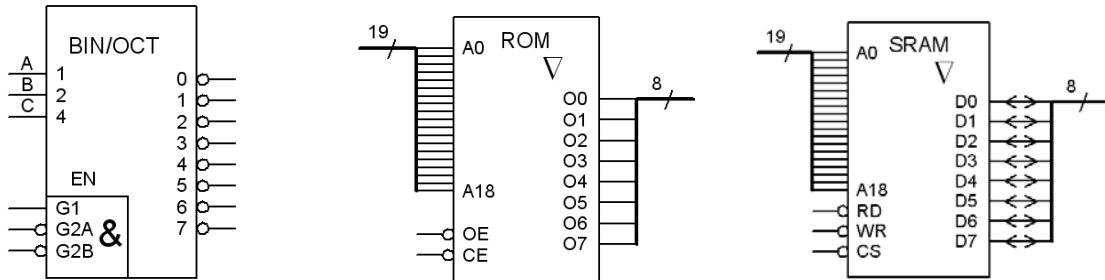
## 12.1

Ett dynamiskt RAM-minne består av ett antal 256Mbit minneskaplar som är organiserade som 32 Mx8.

a) Hur många kaplar behövs för 256Mx64?

b) Hur många kaplar behövs för 512Mx72? (Vad kan anledningen till den ”underliga” bitbredden ”72” vara?)

## 12.2



3:8-avkodare

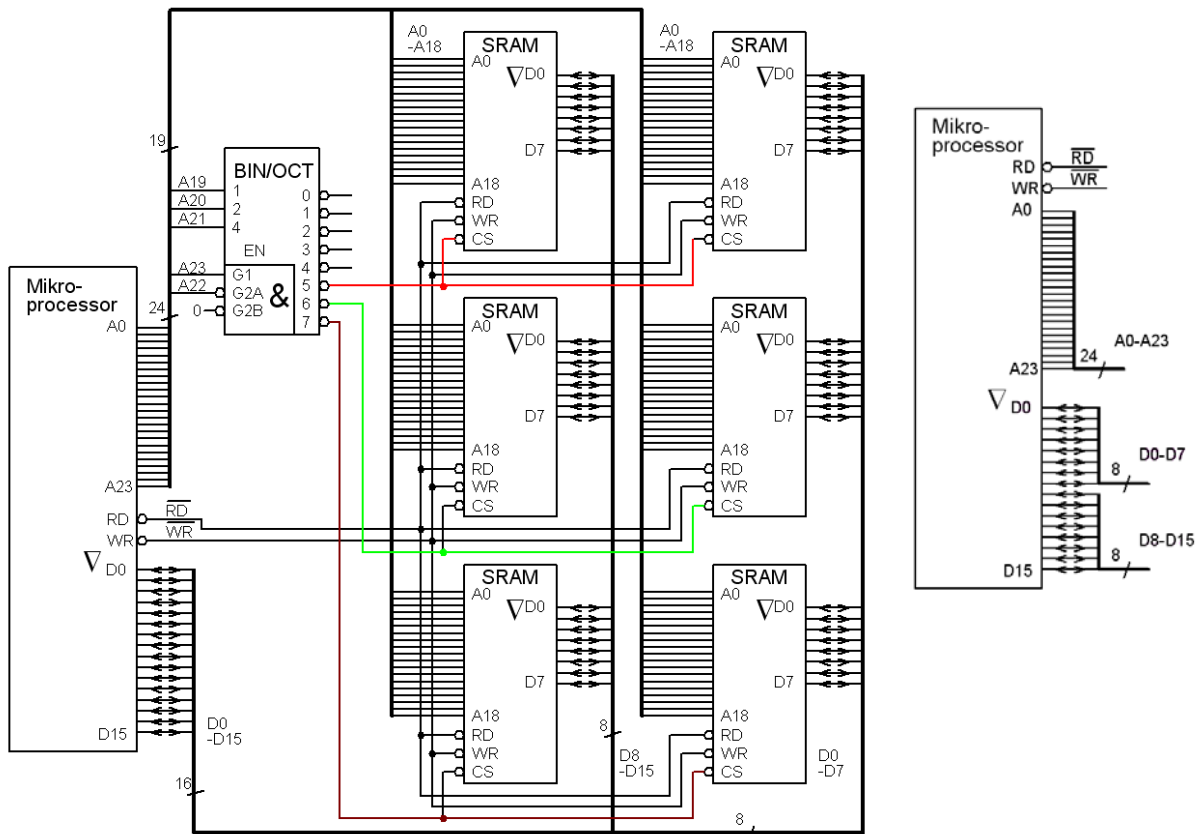
ROM 512kx8

SRAM 512kx8

En viss 16 bitars processor kan adressera 24 bitar. Minnesrymden fördelas mellan ROM, SRAM och IO-kretsar.

Adressavkodningen sker med hjälp av en 3:8-avkodare.

a) Hur stort är figurens RAM? Vilket är adressområdet uttryckt i hexadecimala siffror?



b) Hur ändrar man adressområdet till 980000 – AFFFFF ?

c) Ändra till adressområde 480000 – 5FFFFF ?



# Lösningar

## Talsystem och koder

### 1.1

Nedanstående decimala tal (med basen 10) är givna. Ange motsvarande binära tal.

- a)  $9_{10} = (1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0) = (8+1) = 1001_2$   
b)  $12_{10} = (8+4) = 1100_2$  c)  $71_{10} = (64+7 = 64+4+2+1) = 1000111_2$   
d)  $503_{10} = (512 - 9 = 511 - 8) = (111111111_2 - 1000_2) = 111110111_2$

### 1.2

Omvandla nedanstående binärtal till decimaltal.

- a)  $101101001_2 = (2^8 + 2^6 + 2^5 + 2^3 + 2^0 = 256 + 64 + 32 + 8 + 1) = 361_{10}$   
b)  $110100.010_2 = (2^5 + 2^4 + 2^2 + 2^{-2} = 32 + 16 + 4 + 0.25) = 52,25_{10}$

### 1.3

Omvandla nedanstående binärtal (bas=2) till motsvarande oktala tal (bas=8) och hexadecimala tal (bas=16).

- a)  $011101_2 = 1D_{16} = (011101_2) = 35_8$   
b)  $10001011_2 = 8B_{16} = (010001011_2) = 213_8$   
c)  $100110101_2 = 135_{16} = (100110101_2) = 465_8$   
d)  $11011110010001_2 = DE91_{16} = (001101111010010001_2) = 157221_8$   
e)  $101001.001_2 = 29.2_{16} = (101001.001_2) = 51.1_8$

### 1.4

Omvandla nedanstående hexadecimala tal (bas=16) till motsvarande oktala tal (bas=8).

- a)  $94D_{16} = (100101001101_2 = 100101001101_2) = 4515_8$   
b)  $9E.7A_{16} = (10011110.01110101_2 = 010011110.011110100_2) = 236.364_8$

### 1.5

Omvandla det oktala (bas=8) talet  $4515_8$  till motsvarande hexadecimala tal (bas=16).

$$4515_8 = (100101001101_2 = 100101001101_2) = 94D_{16}$$

### 1.6

Skriv det hexadecimala (bas=16) talet  $BAC_{16}$  på decimal form (bas=10).

$$BAC_{16} = (11 \cdot 16^2 + 10 \cdot 16^1 + 12 \cdot 16^0 = 11 \cdot 256 + 10 \cdot 16 + 12 \cdot 1) = 2988_{10}$$

### 1.7

Vad karakteriserar Gray-koder, och hur kan de konstrueras?

Graykoder har avståndet "ett" mellan kodorden. Det är aldrig mer än en bit i taget som ändras vid övergångarna från ett kodord till nästa. Om man vill konstruera en N-bitars Gray-kod kan man göra detta ur koden för N-1 bitar. Först följer N-1 bitskoden med "0" som bit N, därefter fortsätter man nästa halva med N-1 koden en gång till men med kodorden i omvänd ordning och med "1" som bit N. Detta är en "speglad binärkod".

Så här gör man 3-bits Graykod från 2-bits Graykod:

00 01 11 10 är en 2-bits Graykod. 000 001 011 010 är koden kompletterad "0" som bit 3. 10 11 01 00 är 2-bits koden i omvänd ordning. 110 111 101 100 är den omvända koden kompletterad med "1" som bit 3.

Sammantaget blir 3-bits Graykoden:

000 001 011 010 110 111 101 100

Detta är inte den enda tänkbara 3-bits Graykoden, en annan möjlig kod för tre bitar är tex.

000 010 011 001 101 111 110 100.

(I allmänhet är det den "speglade binärkoden" man menar när man talar om Graykod).

### 1.8

Skriv följande tal "med tecken" med två-komplementsnotation,  $x = (x_6, x_5, x_4, x_3, x_2, x_1, x_0)$ .

- a)  $-23 = (+23_{10} = 0010111_2 \rightarrow -23_{10} = 1101000_2 + 1_2) = 1101001_2 = 105_{10}$   
b)  $-1 = (+1_{10} = 0000001_2 \rightarrow -1_{10} = 1111110_2 + 1_2) = 1111111_2 = 127_{10}$

- c)  $+38 = (32_{10}+4_{10}+2_{10}) = 0100110_2 = 38_{10}$   
 d)  $-64 = (+64_{10} = 1000000_2$  är för stort positivt tal!  $-64_{10} = 0111111_2 + 1_2) = 1000000_2 = 64_{10}$

## 1.9

Skriv följande tal "med tecken" med ett-komplementsnotation,  $x = (x_6, x_5, x_4, x_3, x_2, x_1, x_0)$ .

- a)  $-23 = (+23_{10} = 0010111_2 \rightarrow -23_{10} = 1101000_2) = 1101000_2 = 104_{10}$   
 b)  $-1 \rightarrow 1111110_2 = 126_{10}$  c)  $38 = 0100110_2 = 38_{10}$  d)  $-0 \rightarrow 1111111_2 = 127_{10}$

## Digital aritmetik

### 2.1

Addera för hand följande par binära tal.

- a)  $110 + 010$  b)  $1110 + 1001$  c)  $11\ 0011.01 + 111.1$  d)  $0.1101 + 0.1110$

$$\begin{array}{r} \text{a) } \begin{array}{r} \frac{1}{1} \frac{1}{1} \frac{1}{0} \\ + 0 \ 1 \ 0 \\ \hline 1 \ 0 \ 0 \ 0 \end{array} \quad \text{b) } \begin{array}{r} \frac{1}{1} \frac{1}{1} \frac{1}{1} \frac{1}{0} \\ + 1 \ 0 \ 0 \ 1 \\ \hline 1 \ 0 \ 1 \ 1 \ 1 \end{array} \quad \text{c) } \begin{array}{r} \frac{1}{1} \frac{1}{1} \frac{1}{0} \frac{1}{0} \frac{1}{1} \frac{1}{1} \frac{0}{1} \frac{0}{1} \\ + \quad \quad \quad 1 \ 1 \ 1 \ 1 \\ \hline 1 \ 1 \ 1 \ 0 \ 1 \ 0 \ 1 \ 1 \end{array} \quad \text{d) } \begin{array}{r} \frac{1}{0} \frac{1}{1} \frac{1}{1} \frac{0}{1} \frac{1}{0} \frac{1}{1} \\ + 0 \ 1 \ 1 \ 1 \ 0 \\ \hline 1 \ 1 \ 0 \ 1 \ 1 \end{array} \end{array}$$

### 2.2

Addera eller subtrahera (addition med motsvarande negativa tal) nedanstående tal. Talen skall representeras som binära 4-bitstal (Nibble) på två-komplementform.

- a)  $1 + 2$  b)  $4 - 1$  c)  $7 - 8$  d)  $-3 - 5$   
 $-1_{10} = (+1_{10} = 0001_2 \rightarrow -1_{10} = 1110_2 + 1_2) = 1111_2$   
 $-8_{10} = (+8_{10} = 1000_2 \rightarrow -8_{10} = 0111_2 + 1_2) = 1000_2$   
 $-3_{10} = (+3_{10} = 0011_2 \rightarrow -3_{10} = 1100_2 + 1_2) = 1101_2$   
 $-5_{10} = (+5_{10} = 0101_2 \rightarrow -5_{10} = 1010_2 + 1_2) = 1011_2$

$$\begin{array}{r} \text{a) } \begin{array}{r} 1+2=3 \\ \begin{array}{r} 0 \ 0 \ 0 \ 1 =1 \\ + 0 \ 0 \ 1 \ 0 =2 \\ \hline 0 \ 0 \ 1 \ 1 =3 \end{array} \end{array} \quad \text{b) } \begin{array}{r} 4-1=3 \\ \begin{array}{r} \frac{1}{0} \frac{1}{1} \\ \begin{array}{r} 0 \ 1 \ 0 \ 0 =4 \\ + 1 \ 1 \ 1 \ 1 =-1 \\ \hline \cancel{0} \ 0 \ 0 \ 1 =3 \end{array} \end{array} \end{array} \quad \text{c) } \begin{array}{r} 7-8=-1 \\ \begin{array}{r} 0 \ 1 \ 1 \ 1 =7 \\ + 1 \ 0 \ 0 \ 0 =-8 \\ \hline 1 \ 1 \ 1 \ 1 =-1 \end{array} \end{array} \quad \text{d) } \begin{array}{r} -3-5=-8 \\ \begin{array}{r} \frac{1}{1} \frac{1}{1} \frac{1}{0} \frac{1}{1} =-3 \\ + 1 \ 0 \ 1 \ 1 =-5 \\ \hline \cancel{1} \ 0 \ 0 \ 0 =-8 \end{array} \end{array} \end{array}$$

### 2.3

Multiplitera för hand följande par av teckenlösa binära tal.

- a)  $110 \cdot 010$  b)  $1110 \cdot 1001$  c)  $11\ 0011.01 \cdot 111.1$  d)  $0.1101 \cdot 0.1110$

$$\begin{array}{r} \text{a) } \begin{array}{r} 110 \cdot 010 = 6 \cdot 2 = 12 \\ \begin{array}{r} 1 \ 1 \ 0 =6 \\ \times 0 \ 1 \ 0 =2 \\ \hline 0 \ 0 \ 0 \\ 1 \ 1 \ 0 \\ + 0 \ 0 \ 0 \\ \hline 0 \ 1 \ 1 \ 0 \ 0 =12 \end{array} \end{array} \quad \text{b) } \begin{array}{r} 1110 \cdot 1001 = 14 \cdot 9 = 126 \\ \begin{array}{r} 1 \ 1 \ 1 \ 0 =14 \\ \times 1 \ 0 \ 0 \ 1 =9 \\ \hline 1 \ 1 \ 1 \ 0 \\ 0 \ 0 \ 0 \ 0 \\ 0 \ 0 \ 0 \ 0 \\ 0 \ 0 \ 0 \ 0 \\ + 1 \ 1 \ 1 \ 0 \\ \hline 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 =126 \end{array} \end{array} \quad \text{c) } \begin{array}{r} 11\ 0011.01 \cdot 111.1 = 11001101 \cdot 1111 = 110000000.011 \\ \begin{array}{r} 11001101 \\ \times \quad 1111 \\ \hline 11001101 \\ 11001101 \\ 11001101 \\ + 11001101 \\ \hline 110000000011 \end{array} \end{array} \quad \text{d) } \begin{array}{r} 0.1101 \cdot 0.1110 = 0.10110110 \\ \begin{array}{r} 1101 \\ \times 1110 \\ \hline 0000 \\ 1101 \\ 1101 \\ + 1101 \\ \hline 10110110 \end{array} \end{array} \end{array}$$





## 2.7

$$[b_5 b_4 b_3 b_2 b_1 b_0] = (-1^{b_5}) \cdot (1.b_2 b_1 b_0) \cdot (2^{b_4 b_3 - 1})$$

| Significand $1.b_2 b_1 b_0$ | Exponent $b_4 b_3 - 1$ (excess1)   |
|-----------------------------|--|
| $1.000_2 = 1.000_{10}$      | $00-1 = -1 \quad 2^{00-1} = 0.5$   |
| $1.001_2 = 1.125_{10}$      | $01-1 = 0 \quad 2^{01-1} = 1$  |
| $1.010_2 = 1.25_{10}$       | $10-1 = 1 \quad 2^{10-1} = 2$  |
| $1.011_2 = 1.375_{10}$      | $11-1 = 2 \quad 2^{11-1} = 4$  |
| $1.100_2 = 1.5_{10}$        |  |
| $1.101_2 = 1.625_{10}$      | • $0.25_{10} = 0.01_2 = 1.0_2 \cdot 2^{-2}$ exponent -2 <i>not</i> representable |
| $1.110_2 = 1.75_{10}$       | • $0.8125_{10} = 0.1101_2 = 1.101_2 \cdot 2^{-1}$ representable                  |
| $1.111_2 = 1.875_{10}$      | • $-1.375_{10} = -1.011_2 = -1.011_2 \cdot 2^0$ representable                    |

•  $4.25_{10} = 100.01_2 = 1.0001_2 \cdot 2^2$  significand 1.0001 *not* representable

•  $7.5_{10} = 111.1_2 = 1.111_2 \cdot 2^2$  representable

### Flyttalsaddition

#### Algorithm:

1. Check for zeroes  $a = 001111_2 \rightarrow 0\ 01\ 111 \rightarrow +\ 01\ 1.111$
2. Align significands  $b = 010010_2 \rightarrow 0\ 10\ 010 \rightarrow +\ 10\ 1.010$
3. Add/Sub significands Align significands. **10** > **01** shift smaller
4. Normalize result number *a* to right to get *same* exponent:  
Significand 0.1111 exponent **10**

|  |   |
|--|---|
| Add significands:  | Normalize result:   |
| $\begin{array}{r} 0.1111 \\ + 1.010 \\ \hline 10.0011 \end{array}$ | $10.0011 \text{ exp } 10 \rightarrow 1.00011 \text{ exp } 11$<br>Rounding: $1.00011 \sim 1.001$ |

Result: 0 11 001

$$a = 1.875 \quad b = 2.5 \quad a+b = 4.5 \quad (4.375)$$

### Flyttals multiplikation (enklare än additionen!)

#### Simpler than addition!

- |                                    |                           |
|------------------------------------|---------------------------|
| <b>Algorithm:</b>                  | Exponents: 01 10 Bias = 1 |
| 1. Check for zeroes                | exp = $01+10-1 = 10$      |
| 2. Add exponents and subtract Bias |                           |
| 3. Multiply significands           | Multiply significands     |
| 4. Normalize                       |                           |

|   |  |
|---|--|
| Normalize result:   | $\begin{array}{r} 1.010 \\ * 1.111 \\ \hline 1\ 010 \\ 10\ 10 \\ 101\ 0 \\ + 1\ 010 \\ \hline 10.010\ 110 \end{array}$ |
| $10.010110 \text{ exp } 10 \rightarrow 1.0010110 \text{ exp } 11$ |  |
| Rounding: $1.0010110 \sim 1.001$                                  |  |

Result: 0 11 001

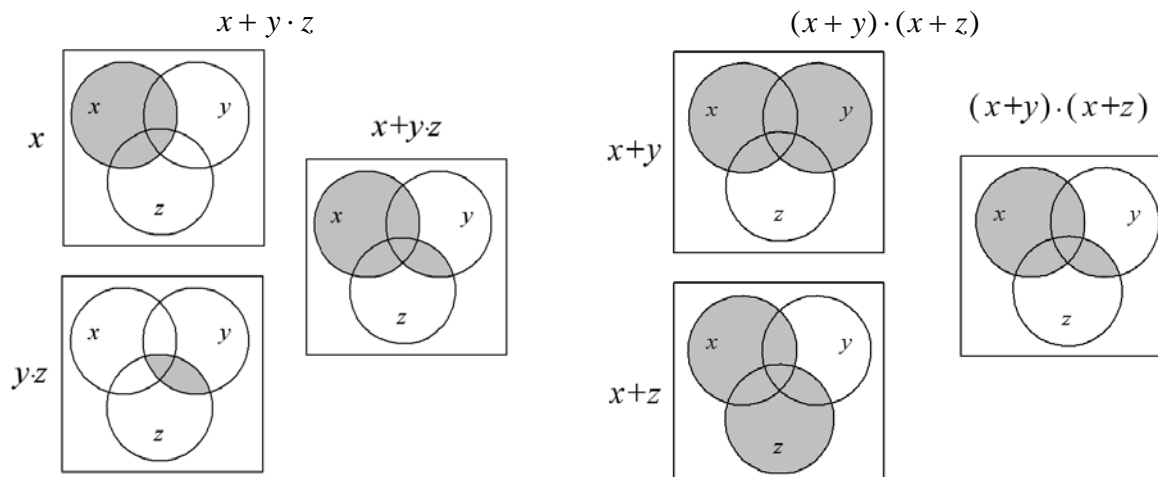
$$a = 1.875 \quad b = 2.5 \quad a*b = 4.5 \quad (4.6875)$$

# Mängdräkning och kubteori

## 3.1

Bevis av den distributiva lagen med hjälp av Venn-diagram.

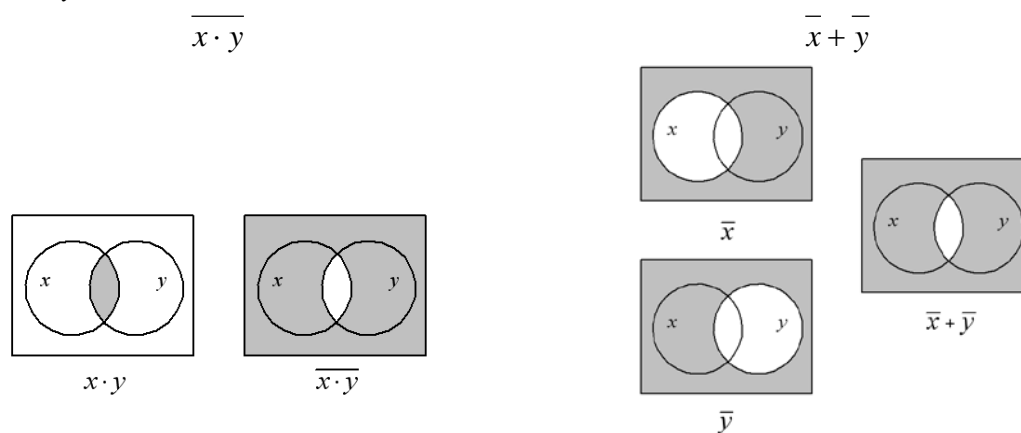
$$x + y \cdot z = (x + y) \cdot (x + z)$$



## 3.2

Bevis av De Morgans lag med hjälp av Venn-diagram.

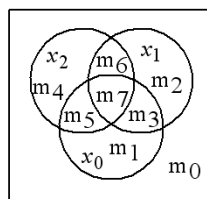
$$\overline{x \cdot y} = \bar{x} + \bar{y}$$



## 3.3

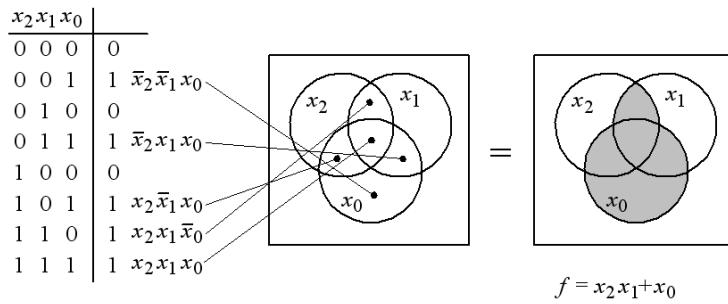
a) Mintermernas placering i ett tre variabelers Venn-diagram.

| $x_2 x_1 x_0$ |       |
|---------------|-------|
| 0 0 0         | $m_0$ |
| 0 0 1         | $m_1$ |
| 0 1 0         | $m_2$ |
| 0 1 1         | $m_3$ |
| 1 0 0         | $m_4$ |
| 1 0 1         | $m_5$ |
| 1 1 0         | $m_6$ |
| 1 1 1         | $m_7$ |



b)

$$f = \bar{x}_2 \bar{x}_1 x_0 + \bar{x}_2 x_1 x_0 + x_2 \bar{x}_1 x_0 + x_2 x_1 \bar{x}_0 + x_2 x_1 x_0$$



Venn diagram-metoden visar tydligt de boolska sambanden, men är svår att använda vid fler än tre variabler. Den är opraktisk att göra om till en datoralgoritm.

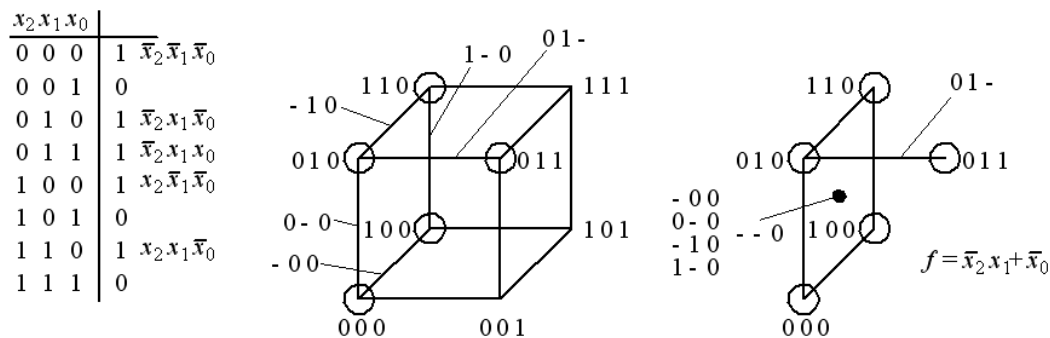
### 3.4

a) Representera följande funktion av tre variabler, som en 3-dimensionell kub med Gray-kodade hörn.

$$f(x_2, x_1, x_0) = \sum m(0, 2, 3, 4, 6) = \bar{x}_2 \bar{x}_1 x_0 + \bar{x}_2 x_1 x_0 + x_2 \bar{x}_1 x_0 + x_2 x_1 \bar{x}_0 + x_2 x_1 x_0$$

b) Använd kuben för att förenkla funktionen.

$$f = \bar{x}_2 \bar{x}_1 \bar{x}_0 + \bar{x}_2 x_1 \bar{x}_0 + \bar{x}_2 x_1 x_0 + x_2 \bar{x}_1 \bar{x}_0 + x_2 x_1 \bar{x}_0$$



Den kubiska representationen är svår att åskådliggöra för fler än 3 dimensioner, men minimerings-metoden kan enkelt definieras för valfritt antal variabler och dimensioner och sedan ligga till grund för dator-algoritmer.

## Boolsk algebra och grindar

### 4.1

- a)  $f = a \cdot \bar{c} \cdot d + a \cdot d = a \cdot d \cdot (\bar{c} + 1) = a \cdot d$
- b)  $f = a \cdot (\bar{b} + \bar{a} \cdot c + a \cdot b) = a \cdot \bar{b} + a \cdot \bar{a} \cdot c + a \cdot a \cdot b = a \cdot \bar{b} + 0 + a \cdot b = a \cdot (\bar{b} + b) = a$
- c)  $f = a + \bar{b} + \bar{a} \cdot b + \bar{c} = (1 \cdot \bar{b} + \bar{a} \cdot b) + a + \bar{c} = \{konsensus\} = (1 \cdot \bar{b} + \bar{a} \cdot b + 1 \cdot \bar{a}) + a + \bar{c} = \dots \bar{a} + a \dots = 1$
- d)  $f = (a + b \cdot \bar{c}) \cdot (\bar{a} \cdot \bar{b} + c) = a \cdot \bar{a} \cdot \bar{b} + a \cdot c + \bar{a} \cdot b \cdot \bar{b} \cdot \bar{c} + b \cdot c \cdot \bar{c} = 0 + a \cdot c + 0 + 0 = a \cdot c$
- e)  $f = (a + \bar{b}) \cdot (\bar{a} + b) \cdot (a + b) = (a \cdot \bar{a} + a \cdot b + \bar{a} \cdot \bar{b} + b \cdot \bar{b}) \cdot (a + b) = (0 + a \cdot b + \bar{a} \cdot \bar{b} + 0) \cdot (a + b) = a \cdot a \cdot b + a \cdot \bar{a} \cdot \bar{b} + a \cdot b \cdot b + \bar{a} \cdot \bar{b} \cdot b = a \cdot b + a \cdot b = a \cdot b$
- f)  $f = \bar{a} \cdot \bar{b} \cdot c + a \cdot b \cdot c + \bar{a} \cdot b \cdot c = c \cdot (\bar{a} \cdot \bar{b} + a \cdot b + \bar{a} \cdot b) = c \cdot (\bar{a} \cdot \bar{b} + b \cdot (a + \bar{a})) = c \cdot (\bar{a} \cdot \bar{b} + 1 \cdot b) = \{konsensus\} = c \cdot (\bar{a} \cdot \bar{b} + 1 \cdot b + \bar{a} \cdot 1) = c \cdot (\bar{a} \cdot \bar{b} + b + \bar{a}) = c \cdot (\bar{a} \cdot (\bar{b} + 1) + b) = c \cdot (\bar{a} + b) = \bar{a} \cdot c + b \cdot c$
- g)  $f = \bar{a} \cdot b \cdot \bar{c} + \bar{a} \cdot b \cdot \bar{d} + c \cdot d = \bar{a} \cdot b \cdot (\bar{c} + \bar{d}) + c \cdot d = \bar{a} \cdot b \cdot \overline{\overline{\overline{c+d}}} + c \cdot d = \bar{a} \cdot b \cdot \overline{\overline{c+d}} + c \cdot d = \{x + \bar{x}y = x + y\} = \bar{a} \cdot b + c \cdot d$
- h)  $f = a + (\overline{\overline{a \cdot b}}) = \{deMorgan\} = a + \bar{a} + \bar{b} = a + b$
- i)  $f = \overline{\overline{\overline{a + a \cdot b + c}}} = \{deMorgan\} = a \cdot (\bar{a} \cdot b + c) = a \cdot \bar{a} \cdot b + a \cdot c = ac$

### 4.2

Bevisa algebraiskt att följande samband är giltiga.

- a)  $(x_3 x_2 + 1 + x_3 x_2) x_1 + x_2 x_1 + x_2 + x_1 = 1$  **VL:**  $(\dots + 1)x_1 = x_1$   $x_1 + x_1 + \dots = 1$
- b)  $\overline{x_3 x_2 x_1} + \overline{(x_3 + x_1)} = \overline{x_3 x_1}$  **VL:**  $\overline{x_3 x_2 x_1} + \overline{(x_3 + x_1)} = \overline{x_3 x_2 x_1} + \overline{x_3} x_1 = \overline{x_3} x_1 (x_2 + 1) = \overline{x_3} x_1$
- c)  $\overline{(x_2 + x_1)} x_2 x_1 + x_3 = \overline{x_2 x_1} + x_3$  **VL:**  $\overline{(x_2 + x_1)} x_2 x_1 + x_3 = \overline{x_2} x_1 x_2 x_1 + x_3 = \overline{x_2} x_1 + x_3$
- d)  $x_2 + x_1 + \overline{x_2} x_1 + x_3 = \overline{x_2} x_1 + x_3$  **VL:**  $x_2 + x_1 + \overline{x_2} x_1 + x_3 = \overline{x_2} x_1 + x_2 x_1 + x_3 = \overline{x_2} x_1 + x_3$

### 4.3

Förenkla nedanstående tre uttryck så långt som möjligt.

- a)  $(x + y)(\bar{x} + z) = x\bar{x} + y\bar{x} + xz + yz = \bar{x}y + xz + yz = \{konsensus ta bort yz\} = \bar{x}y + xz$
- b)  $(x + \bar{y} + xy)(x + \bar{y})\bar{x}y = (x + xy + x\bar{y} + y)\bar{x}y = 0 + 0 + 0 + 0$
- c)  $x(1 + \bar{x}y) + \bar{x} = x(1) + \bar{x} = 1$

### 4.4

Förenkla nedanstående uttryck så långt som möjligt.

$$\begin{aligned} & \overline{(a + b + c)(a + \bar{b} + \bar{c})(a + \bar{b}c + \bar{b}c)} = \overline{(a + b + c)(a + \bar{b} + \bar{c})(a + b + c + \bar{b}c)} = \\ & \overline{(a + b + c)(a + \bar{b} + \bar{c})(a + b + c)} = \\ & \overline{(a + b + c) + (a + \bar{b} + \bar{c}) + (a + b + c)} = \overline{abc + \bar{a}bc + \bar{a}bc} = \overline{abc + \bar{a}bc + \bar{a}bc + \bar{a}bc} = \\ & \overline{ac(\bar{b} + b) + \bar{b}c(a + \bar{a})} = \overline{ac + \bar{b}c} \end{aligned}$$

#### 4.5

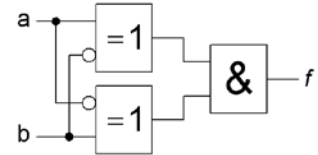


#### 4.6

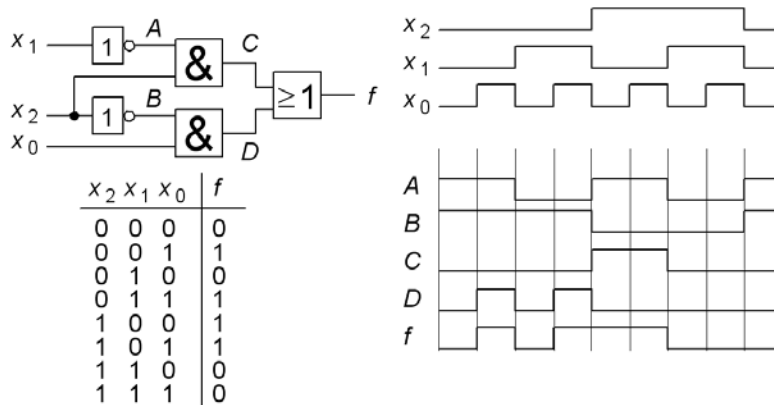
Förenkla  $f(a,b)$  som realiseras av figurens grindnät, så långt som möjligt, och ange funktionens namn.

$$(a \oplus \bar{b})(b \oplus \bar{a}) = (\overline{ab} + \overline{a\bar{b}})(\overline{ba} + \overline{b\bar{a}}) = (ab + \overline{a\bar{b}})(ab + \overline{a\bar{b}}) = ab + \overline{a\bar{b}}$$

Det blir en XNOR –funktion.



#### 4.7



#### 4.8

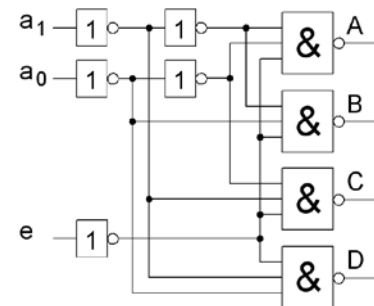
Ange de logiska uttrycken för A, B, C och D.

$$A = \overline{\overline{a_1 a_0 e}} = \overline{a_1 + a_0 + e}$$

$$B = \overline{\overline{a_1 a_0 e}} = \overline{a_1 + a_0 + e}$$

$$C = \overline{\overline{a_1 a_0 e}} = \overline{a_1 + a_0 + e}$$

$$D = \overline{\overline{a_1 a_0 e}} = \overline{a_1 + a_0 + e}$$



#### 4.9

Förenkla det sammansatta uttrycken nedan så långt som möjligt.

a)

XOR – funktionen  $a \oplus b = \overline{ab} + \overline{a\bar{b}}$

$$\begin{aligned} x_2 \oplus x_1 \oplus x_1 x_2 &= (\overline{x_2 x_1} + \overline{x_2 \bar{x}_1})x_1 x_2 + (\overline{x_2 x_1} + \overline{x_2 \bar{x}_1})x_1 \bar{x}_2 = \\ &= (\overline{x_2 x_1} + \overline{x_2 \bar{x}_1})(x_1 + \bar{x}_1) + \overline{x_2 x_1} \overline{x_2 \bar{x}_1} x_1 x_2 = (\overline{x_2 x_1} + 0 + 0 + \overline{x_2 \bar{x}_1}) + (\overline{x_2} + \bar{x}_1)(\overline{x_2} + x_1)x_1 x_2 = \\ &= +(0 + x_1 x_2 + \overline{x_2} \overline{x_2} + 0)x_1 x_2 = \overline{x_2} x_1 + \overline{x_2} x_1 + x_1 x_2 + x_1 x_2 = \\ &= x_2(\overline{x_1} + x_1) + x_1(\overline{x_2} + x_2) = x_2 + x_1 \end{aligned}$$

b)

$$\begin{aligned} x_2 x_1 \oplus (\overline{x_2} + x_1) &= x_2 x_1 (x_2 + x_1) + \overline{x_2 x_1} (\overline{x_2} + x_1) = x_2 x_1 + (\overline{x_2} + \bar{x}_1)\overline{x_2} x_1 = \\ &= x_2 x_1 + \overline{x_2} \bar{x}_1 \end{aligned}$$

#### 4.10

Visa att

a)

$$\overline{x_2 \oplus x_1} = \overline{x_2} \oplus x_1 = x_2 \oplus \overline{x_1}$$

Denna gång prövar vi att bevisa sambanden med så kallad "perfekt induktion". Det innebär att man direkt sätter in alla fyra kombinationerna av de två variablerna i de olika uttrycken. Om uttrycken har samma sanningstabell så är dom ekvivalenta. När variablerna är få, kan denna icke algebraiska metod användas.

| $x_2$ | $x_1$ | $\overline{x_2}$ | $\overline{x_1}$ | $\overline{x_2 \oplus x_1}$ | $\overline{x_2} \oplus x_1$ | $x_2 \oplus \overline{x_1}$ |
|-------|-------|------------------|------------------|-----------------------------|-----------------------------|-----------------------------|
| 0     | 0     | 1                | 1                | 1                           | 1                           | 1                           |
| 0     | 1     | 1                | 0                | 0                           | 0                           | 0                           |
| 1     | 0     | 0                | 1                | 0                           | 0                           | 0                           |
| 1     | 1     | 0                | 0                | 1                           | 1                           | 1                           |

b)

$$x_2 \oplus x_1 = \overline{x_2} \oplus \overline{x_1}$$

$$VL: x_2 \oplus x_1 = x_2 \overline{x_1} + \overline{x_2} x_1$$

$$HL: \overline{x_2} \oplus \overline{x_1} = \overline{x_2} \overline{\overline{x_1}} + \overline{\overline{x_2}} \overline{x_1} = \overline{x_2} x_1 + x_2 \overline{x_1} \quad VL = HL$$

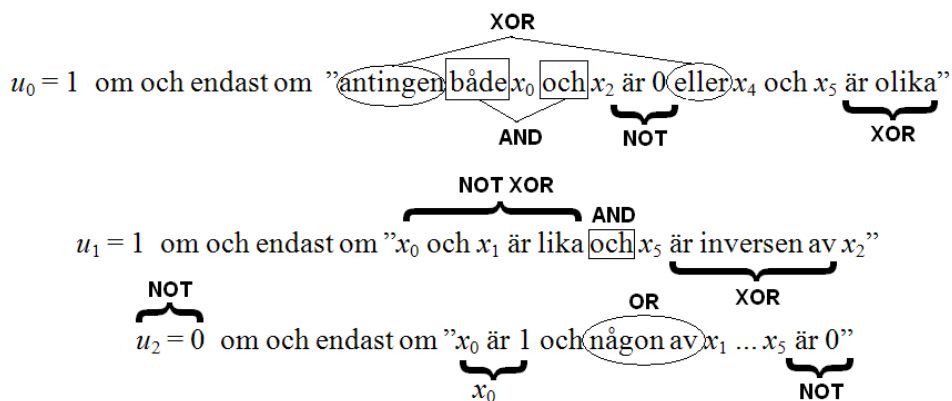
#### 4.11

Här visas de Europeiska grindsymbolerna. Rita de motsvarande amerikanska grindsymbolerna.

| AND | OR | NOT | NAND | NOR | XOR | XNOR |
|-----|----|-----|------|-----|-----|------|
|     |    |     |      |     |     |      |
|     |    |     |      |     |     |      |

#### 4.12

Från text till Booleskt uttryck.



$$u_0 = \overline{x_0} \cdot \overline{x_2} \oplus (x_4 \oplus x_5)$$

$$u_1 = x_0 \oplus x_1 \cdot (x_5 \oplus x_2)$$

$$\overline{u_2} = x_0 \cdot (\overline{x_1} + \overline{x_2} + \overline{x_3} + \overline{x_4} + \overline{x_5}) \Rightarrow u_2 = x_0 \cdot (\overline{\overline{x_1} + \overline{x_2} + \overline{x_3} + \overline{x_4} + \overline{x_5}}) = x_0 + x_1 \cdot x_2 \cdot x_3 \cdot x_4 \cdot x_5$$

# Sanningstabellen, SP och PS –form, fullständig logik

## 5.1

Kontakter avbildas alltid i opåverkat läge. För att få lampan att lysa ska man samtidigt trycka på siffrorna "4" och "2" det vill säga kontakt  $d$  och  $h$ . Observera att man *inte* får trycka ned någon annan kontakt! Den logiska funktionen (lampan lyser) blir:

$$f = \bar{a} \cdot \bar{b} \cdot \bar{c} \cdot d \cdot e \cdot \bar{f} \cdot \bar{g} \cdot h \cdot \bar{i} \cdot \bar{k}$$

Kodlåset är en avkodare, det avkodar en enda minterm i saningstabellen.

## 5.2

| $a$ | $b$ | $c$ | $f$ | $a$ | $b$ | $c$ | $f$ |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 0   | 0   | 0   | 1   | 1   | 0   | 0   | 1   |
| 0   | 0   | 1   | 0   | 1   | 0   | 1   | 1   |
| 0   | 1   | 0   | 0   | 1   | 1   | 0   | 0   |
| 0   | 1   | 1   | 0   | 1   | 1   | 1   | 1   |

Funktionen på SP-normalform:

$$f = \bar{a}\bar{b}\bar{c} + \bar{a}\bar{b}c + \bar{a}b\bar{c} + abc$$

Funktionen på PS-normalform:

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

## 5.3

$$\begin{aligned} f(x, y, z) &= \bar{x}\bar{y} + \bar{y}\bar{z} + \bar{x}z = x\bar{y}(z + \bar{z}) + (x + \bar{x})\bar{y}\bar{z} + \bar{x}(y + \bar{y})z = \\ &= x\bar{y}z + x\bar{y}\bar{z} + \bar{x}\bar{y}\bar{z} + \bar{x}yz + \bar{x}yz + \bar{x}yz \end{aligned}$$

$$\Rightarrow f(x, y, z) = \sum m(001, 010, 011, 100, 101, 110) = \sum m(1, 2, 3, 4, 5, 6)$$

$$\Rightarrow f(x, y, z) = \prod M(0, 7) = (\bar{x} + \bar{y} + \bar{z})(x + y + z)$$

## 5.4

$$\begin{aligned} f(x, y, z) &= (x + \bar{y})(xyz + \bar{y}(x + z)) + xy\bar{z}(x + \bar{y}) = (x + \bar{y})(xyz + \bar{y}(x + z)) + xy\bar{z}(x + y) = \\ &= (x + \bar{y})(xyz + \bar{y}(x + z)) + xy\bar{z} = \end{aligned}$$

$$= xyz + x\bar{y} + x\bar{y}z + xy\bar{z} + x\bar{y} + \bar{y}z + xy\bar{z} =$$

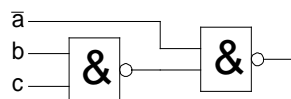
$$= xyz + x\bar{y}(z + \bar{z}) + x\bar{y}z + x\bar{y}(z + \bar{z}) + (x + \bar{x})\bar{y}z + xy\bar{z} =$$

$$= xyz + x\bar{y}z + x\bar{y}\bar{z} + x\bar{y}z + x\bar{y}z + x\bar{y}z + x\bar{y}z + \bar{x}\bar{y}z + xy\bar{z} = \bar{x}\bar{y}z + x\bar{y}\bar{z} + x\bar{y}z + xy\bar{z} + xyz$$

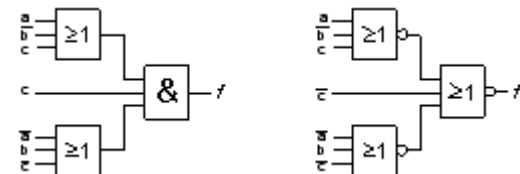
$$\Rightarrow f(x, y, z) = \sum m(001, 100, 101, 110, 111) = \sum m(1, 4, 5, 6, 7)$$

$$\Rightarrow f(x, y, z) = \prod M(0, 2, 3) = (x + y + z)(x + \bar{y} + z)(x + \bar{y} + \bar{z})$$

## 5.5



## 5.6





### 5.7

a) Paritetskrets för jämn paritet, antalet ettor skall vara ett jämnt tal (0, 2, eller 4) för "1" på utgången.

|    | d | c | b | a | J |
|----|---|---|---|---|---|
| 0  | 0 | 0 | 0 | 0 | 1 |
| 1  | 0 | 0 | 0 | 1 | 0 |
| 2  | 0 | 0 | 1 | 0 | 0 |
| 3  | 0 | 0 | 1 | 1 | 1 |
| 4  | 0 | 1 | 0 | 0 | 0 |
| 5  | 0 | 1 | 0 | 1 | 1 |
| 6  | 0 | 1 | 1 | 0 | 1 |
| 7  | 0 | 1 | 1 | 1 | 0 |
| 8  | 1 | 0 | 0 | 0 | 0 |
| 9  | 1 | 0 | 0 | 1 | 1 |
| 10 | 1 | 0 | 1 | 0 | 1 |
| 11 | 1 | 0 | 1 | 1 | 0 |
| 12 | 1 | 1 | 0 | 0 | 1 |
| 13 | 1 | 1 | 0 | 1 | 0 |
| 14 | 1 | 1 | 1 | 0 | 0 |
| 15 | 1 | 1 | 1 | 1 | 1 |

Hälften av sanningstabellens rader skall vara "1". Denna funktion går *inte* att minimera, utan alla 8 mintermerna behöver vara med i SP-formen!

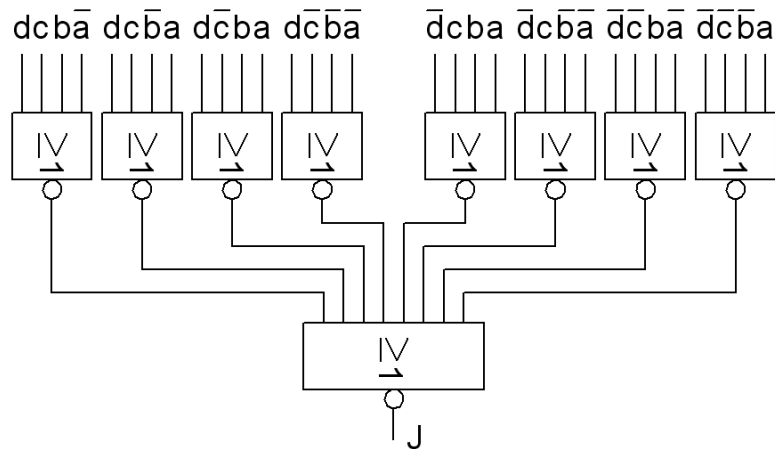
$$J = \overline{d}\overline{c}\overline{b}\overline{a} + \overline{d}\overline{c}b\overline{a} + \overline{d}c\overline{b}\overline{a} + \overline{d}c\overline{b}a + \overline{d}cb\overline{a} + \overline{d}cba + d\overline{c}\overline{b}\overline{a} + d\overline{c}b\overline{a}$$

(Den som redan nu känner till Karnaugh-diagrammet kan i ett sådant direkt se att inga "hoptagningar" låter sig göras.)

|   |   | b a  |      |      |      |
|---|---|------|------|------|------|
|   |   | 00   | 01   | 11   | 10   |
| d | 0 | 0 1  | 1 0  | 3 1  | 2 0  |
|   | 0 | 4 0  | 5 1  | 7 0  | 6 1  |
| 1 | 1 | 12 1 | 13 0 | 15 1 | 14 0 |
|   | 1 | 8 0  | 9 1  | 11 0 | 10 1 |

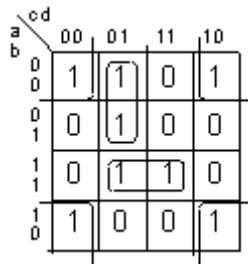
b) Med NOR-grindar lämpar sig PS-formen bättre.

$$J = (d + c + b + \overline{a})(d + c + \overline{b} + a)(d + \overline{c} + b + a)(d + \overline{c} + \overline{b} + \overline{a}) \cdot (\overline{d} + c + b + a)(\overline{d} + c + \overline{b} + \overline{a})(\overline{d} + \overline{c} + b + \overline{a})(\overline{d} + \overline{c} + \overline{b} + a)$$



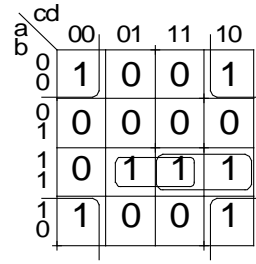
# Karnaughdiagrammet

6.1



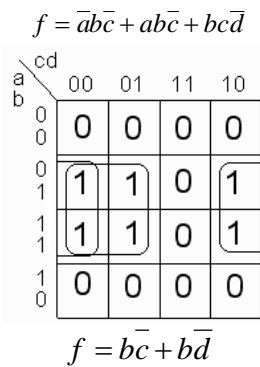
$$f = \bar{a}c\bar{d} + abd + \bar{b}\bar{d}$$

6.2

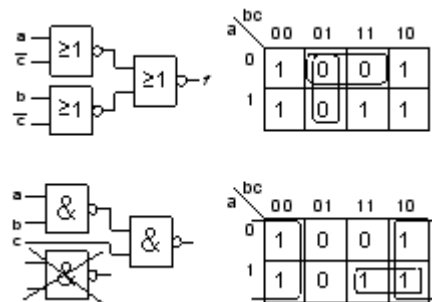


$$f = abd + abc + \bar{b}\bar{d}$$

6.3



6.4

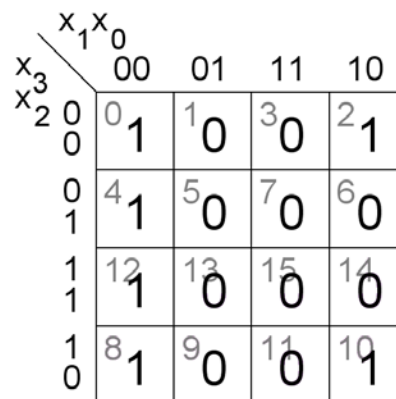


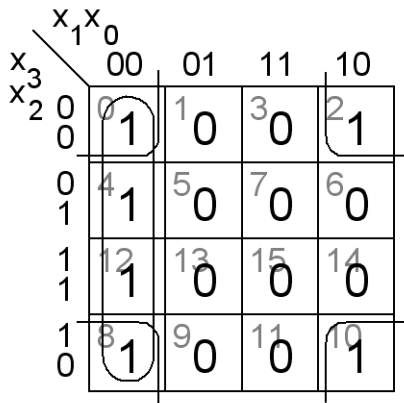
6.5

Sanningstabell och Karnaughdiagram. Den minimerade funktionen erhålls genom hoptagning av 1:or i Karnaughdiagrammet. Funktionen invers fås om 0:orna "felaktigt" tas ihop som om dom vore 1:or.

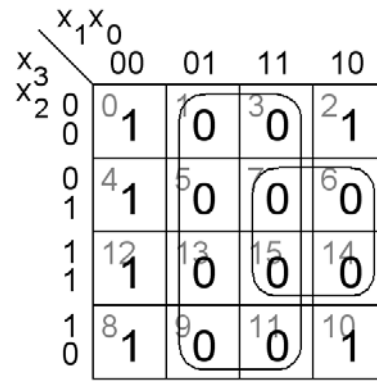
$$f(x_3, x_2, x_1, x_0) = \sum m(0, 2, 4, 8, 10, 12) \quad f = ? \quad \bar{f} = ?$$

|    | $x_3$ | $x_2$ | $x_1$ | $x_0$ | $f$ |
|----|-------|-------|-------|-------|-----|
| 0  | 0     | 0     | 0     | 0     | 1   |
| 1  | 0     | 0     | 0     | 1     | 0   |
| 2  | 0     | 0     | 1     | 0     | 1   |
| 3  | 0     | 0     | 1     | 1     | 0   |
| 4  | 0     | 1     | 0     | 0     | 1   |
| 5  | 0     | 1     | 0     | 1     | 0   |
| 6  | 0     | 1     | 1     | 0     | 0   |
| 7  | 0     | 1     | 1     | 1     | 0   |
| 8  | 1     | 0     | 0     | 0     | 1   |
| 9  | 1     | 0     | 0     | 1     | 0   |
| 10 | 1     | 0     | 1     | 0     | 1   |
| 11 | 1     | 0     | 1     | 1     | 0   |
| 12 | 1     | 1     | 0     | 0     | 1   |
| 13 | 1     | 1     | 0     | 1     | 0   |
| 14 | 1     | 1     | 1     | 0     | 0   |
| 15 | 1     | 1     | 1     | 1     | 0   |





$$f = \bar{x}_1\bar{x}_0 + \bar{x}_2\bar{x}_0$$



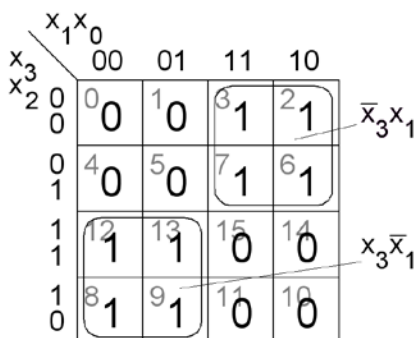
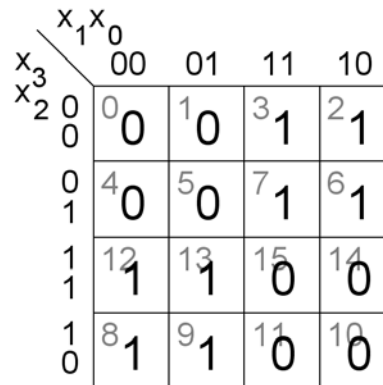
$$\bar{f} = \{0 : \text{or som } 1 : \text{or}\} = x_0 + x_2x_1$$

### 6.6

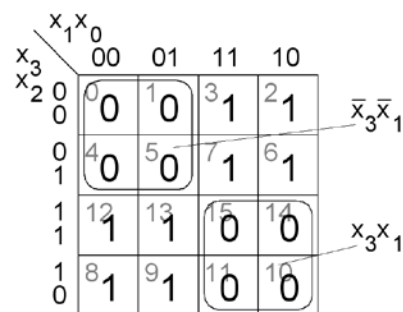
Sanningstabell och Karnaughdiagram. Den minimerade funktionen erhålls genom hoptagning av 1:or i Karnaughdiagrammet. Funktionens invers fås om 0:orna "felaktigt" tas ihop som om dom vore 1:or.

$$f(x_3, x_2, x_1, x_0) = \prod M(0, 1, 4, 5, 10, 11, 14, 15) \quad f = ? \quad \bar{f} = ?$$

|    | $x_3$ | $x_2$ | $x_1$ | $x_0$ | $f$ |
|----|-------|-------|-------|-------|-----|
| 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     | 0   |
| 5  | 0     | 1     | 0     | 1     | 0   |
| 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     | 0   |
| 11 | 1     | 0     | 1     | 1     | 0   |
| 12 | 1     | 1     | 0     | 0     | 1   |
| 13 | 1     | 1     | 0     | 1     | 1   |
| 14 | 1     | 1     | 1     | 0     | 0   |
| 15 | 1     | 1     | 1     | 1     | 0   |



$$f = x_3x_1 + \bar{x}_3x_1 = x_3 \oplus x_1$$



$$\bar{f} = \{0 : \text{or som } 1 : \text{or}\} = \bar{x}_3\bar{x}_1 + x_3x_1 = x_3 \oplus x_1$$

### 6.7

Sanningstabell och Karnaughdiagram. Den minimerade funktionen erhålls genom hoptagning av 1:or i Karnaughdiagrammet. Funktionen invers fås om 0:orna "felaktigt" tas ihop som om dom vore 1:or.

$$f(x_3, x_2, x_1, x_0) = \sum m(0, 2, 3, 4, 6, 7, 8, 9, 10, 12, 13, 14) \quad f = ? \quad \bar{f} = ?$$

|    | $x_3$ | $x_2$ | $x_1$ | $x_0$ | $f$ |
|----|-------|-------|-------|-------|-----|
| 0  | 0     | 0     | 0     | 0     | 1   |
| 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     | 0   |
| 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     | 1   |
| 13 | 1     | 1     | 0     | 1     | 1   |
| 14 | 1     | 1     | 1     | 0     | 1   |
| 15 | 1     | 1     | 1     | 1     | 0   |

|       |   | $x_1x_0$ |         |         |         |
|-------|---|----------|---------|---------|---------|
|       |   | 00       | 01      | 11      | 10      |
| $x_3$ | 0 | 0<br>1   | 1<br>0  | 3<br>1  | 2<br>1  |
|       | 1 | 4<br>1   | 5<br>0  | 7<br>1  | 6<br>1  |
| $x_2$ | 0 | 12<br>1  | 13<br>1 | 15<br>0 | 14<br>1 |
|       | 1 | 8<br>1   | 9<br>1  | 11<br>0 | 10<br>1 |

|       |   | $x_1x_0$ |         |         |         |                |
|-------|---|----------|---------|---------|---------|----------------|
|       |   | 00       | 01      | 11      | 10      |                |
| $x_3$ | 0 | 0<br>1   | 1<br>0  | 3<br>1  | 2<br>1  | $\bar{x}_3x_1$ |
|       | 1 | 4<br>1   | 5<br>0  | 7<br>1  | 6<br>1  | $\bar{x}_0$    |
| $x_2$ | 0 | 12<br>1  | 13<br>1 | 15<br>0 | 14<br>1 | $x_3\bar{x}_1$ |
|       | 1 | 8<br>1   | 9<br>1  | 11<br>0 | 10<br>1 |                |

|       |   | $x_1x_0$ |         |         |         |                         |
|-------|---|----------|---------|---------|---------|-------------------------|
|       |   | 00       | 01      | 11      | 10      |                         |
| $x_3$ | 0 | 0<br>1   | 1<br>0  | 3<br>1  | 2<br>1  | $\bar{x}_3\bar{x}_1x_0$ |
|       | 1 | 4<br>1   | 5<br>0  | 7<br>1  | 6<br>1  |                         |
| $x_2$ | 0 | 12<br>1  | 13<br>1 | 15<br>0 | 14<br>1 | $x_3x_1x_0$             |
|       | 1 | 8<br>1   | 9<br>1  | 11<br>0 | 10<br>1 |                         |

$$f = \bar{x}_0 + x_3\bar{x}_1 + x_3x_1$$

$$\bar{f} = \{0: \text{or som } 1: \text{or}\} = \bar{x}_3\bar{x}_1x_0 + x_3x_1x_0$$

6.8

$$f(x_3, x_2, x_1, x_0) = \sum m(3, 5, 7, 11) + d(6, 15) \quad f = ? \quad \bar{f} = ?$$

|       |       |          |    |    |    |
|-------|-------|----------|----|----|----|
|       |       | $x_1x_0$ |    |    |    |
|       |       | 00       | 01 | 11 | 10 |
| $x_3$ | $x_2$ | 0        | 0  | 1  | 0  |
|       |       | 0        | 0  | 1  | 0  |
| 0     | 1     | 0        | 1  | 1  | 0  |
|       | 1     | 0        | 1  | 1  | 0  |
| 1     | 1     | 0        | 0  | 1  | 0  |
|       | 1     | 0        | 0  | 1  | 0  |

$$f = x_1 x_0 + \bar{x}_3 x_2 x_0$$

|       |       |                 |    |    |    |
|-------|-------|-----------------|----|----|----|
|       |       | $x_2 \bar{x}_1$ |    |    |    |
|       |       | 00              | 01 | 11 | 10 |
| $x_3$ | $x_2$ | 0               | 0  | 1  | 0  |
|       |       | 0               | 0  | 1  | 0  |
| 0     | 1     | 0               | 1  | 1  | 0  |
|       | 1     | 0               | 1  | 1  | 0  |
| 1     | 1     | 0               | 0  | 1  | 0  |
|       | 1     | 0               | 0  | 1  | 0  |

$$\bar{f} = \bar{x}_0 + \bar{x}_2 \bar{x}_1 + x_3 \bar{x}_1$$

6.9

$$f(x_3, x_2, x_1, x_0) = \sum m(1, 4, 5) + d(2, 3, 6, 7, 8, 9, 12, 13) \quad f = ? \quad \bar{f} = ?$$

|       |       |          |    |    |    |
|-------|-------|----------|----|----|----|
|       |       | $x_1x_0$ |    |    |    |
|       |       | 00       | 01 | 11 | 10 |
| $x_3$ | $x_2$ | 0        | 0  | 1  | 0  |
|       |       | 0        | 0  | 1  | 0  |
| 0     | 1     | 0        | 1  | 1  | 0  |
|       | 1     | 0        | 1  | 1  | 0  |
| 1     | 1     | 0        | 0  | 1  | 0  |
|       | 1     | 0        | 0  | 1  | 0  |

$$f = x_2 \bar{x}_1 + \bar{x}_1 x_0$$

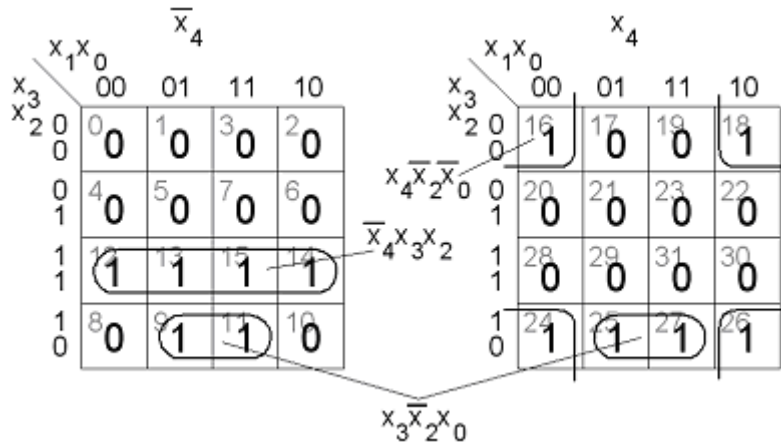
|       |       |          |    |    |    |
|-------|-------|----------|----|----|----|
|       |       | $x_1x_0$ |    |    |    |
|       |       | 00       | 01 | 11 | 10 |
| $x_3$ | $x_2$ | 0        | 0  | 1  | 0  |
|       |       | 0        | 0  | 1  | 0  |
| 0     | 1     | 0        | 1  | 1  | 0  |
|       | 1     | 0        | 1  | 1  | 0  |
| 1     | 1     | 0        | 0  | 1  | 0  |
|       | 1     | 0        | 0  | 1  | 0  |

$$\bar{f} = x_2 + \bar{x}_2 \bar{x}_0 \text{ eller } \bar{f} = x_1 + \bar{x}_2 \bar{x}_0$$

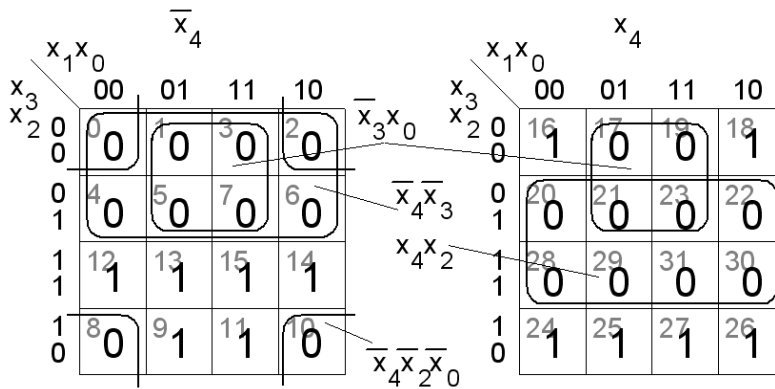
### 6.10

Karnaughdiagram för **fem** variabler. Det vänstra diagrammet är för  $\bar{x}_4$  och det högra för  $x_4$ . Om samma hoptagning kan göras i *båda* diagrammen utgår variabeln  $\bar{x}_4$  eller  $x_4$  annars tas respektive variabel med.

$$f(x_4, x_3, x_2, x_1, x_0) \quad f = ? \quad \bar{f} = ?$$



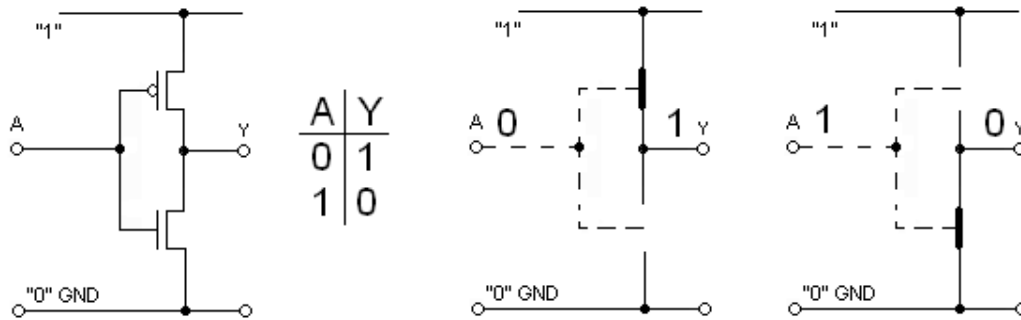
$$f = \bar{x}_4 x_3 x_2 + x_3 \bar{x}_2 x_0 + x_4 x_2 x_0$$



$$\bar{f} = \bar{x}_4 x_3 + x_3 x_0 + x_4 x_2 + x_4 x_2 x_0$$

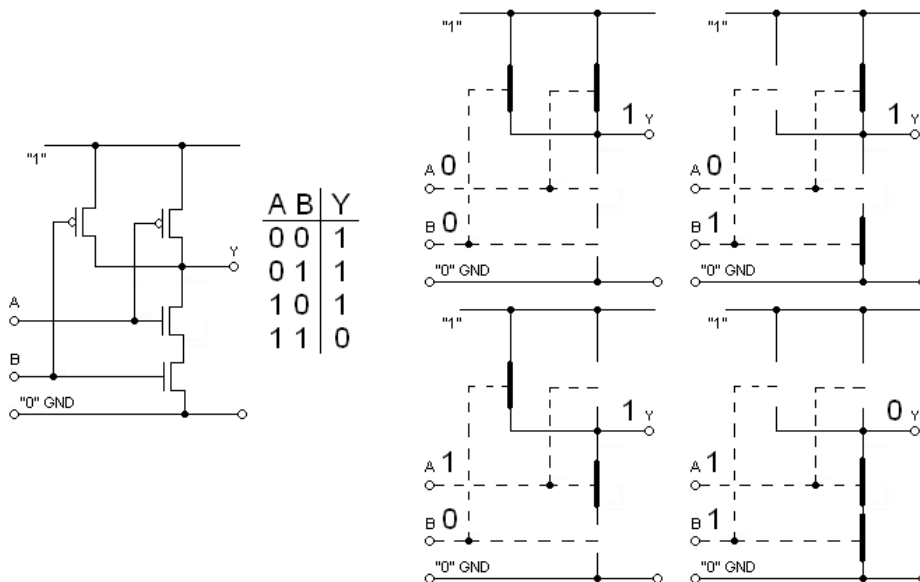
# MOS-transistorer och digitala kretsar

## 7.1



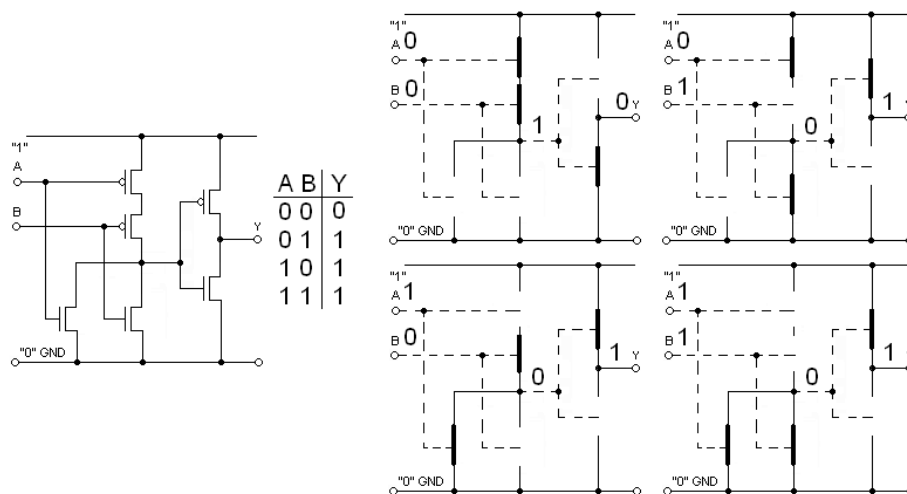
Det är en CMOS-inverterare.  $Y = \bar{A}$ .

## 7.2



Det är en CMOS-NAND-grind.  $Y = \overline{A \cdot B}$ .

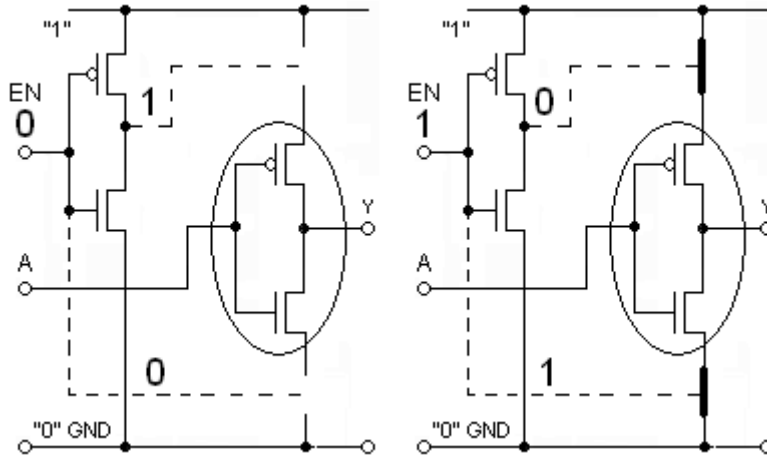
## 7.3



Det är en CMOS-OR-grind.  $Y = A + B$ .

## 7.4

Kretsen är en inverterare med THREE-state utgång. När  $EN = 1$  blir den inringade delen ”inkopplad” och sambandet mellan  $Y$  och  $A$  blir då invertering,  $Y = \bar{A}$ . När  $EN = 0$  blir den inringade delen ”urkopplad”. Utgången är då i ett tredje tillstånd, förutom ”1” och ”0” finns det således ett tillstånd ”urkopplad”. Eftersom utgången  $Y$  nu inte är inkopplad kan den då inte heller påverkas av  $A$ .

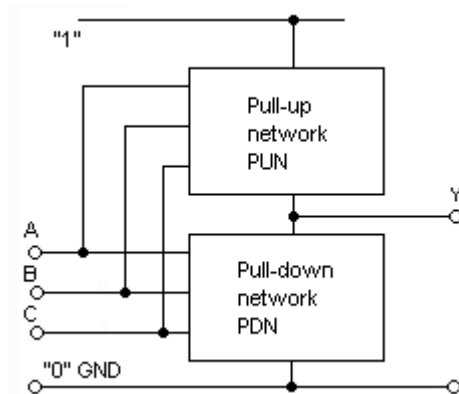


THREE-state utgångar används för att göra det möjligt att koppla samman många utgångar till en och samma ledning (**bussledning**). Flera kretsars utgångar kan utnyttja en gemensam ledning, förutsatt att bara en av kretsarna är aktiv ( $EN = 1$ ) åt gången – de övriga har  $EN = 0$  och är ”urkopplade”.

## 7.5

CMOS-kretsar består av två delnät som är varandras inverser. Pull-up-nätet, PUN, styr ”1” till utgången medan Pull-down-nätet styr ”0”. Analyserar man Pull-down-nätet så får man därför funktionen  $Y$ :s invers.

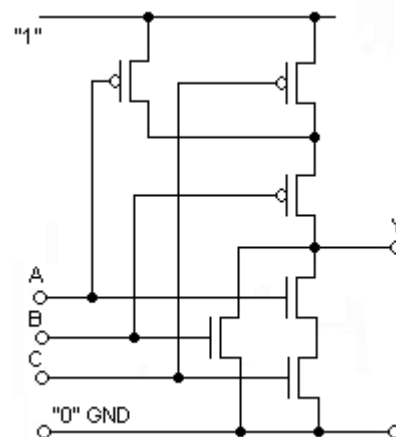
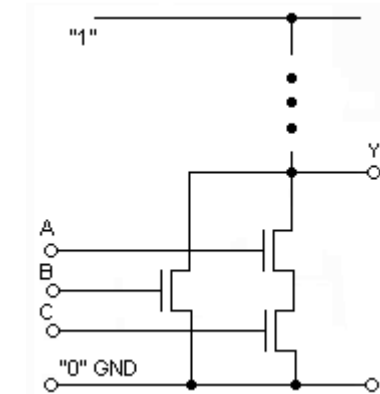
$$\bar{Y} = A \cdot C + B \Rightarrow Y = \overline{A \cdot C + B} = \overline{A \cdot C} \cdot \bar{B} = (\bar{A} + \bar{C})\bar{B}$$



$$(\bar{A} + \bar{C})\bar{B}$$

Pull-up-nätet skall således bestå av  $A$  och  $C$  i parallellkoppling (+) sedan seriekopplade ( $\cdot$ ) med  $B$ . Användandet av PMOS-transistorer inverterar variablerna  $A$ ,  $B$  och  $C$ .

$$(\bar{A} + \bar{C})\bar{B}$$



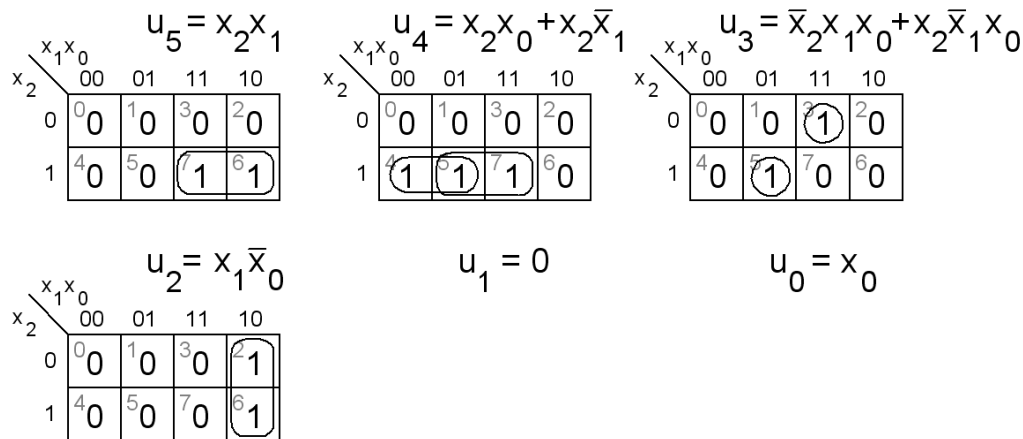
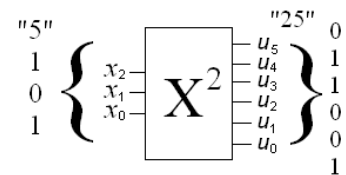


# Kombinatoriska kretsnet

## 8.1

| X | $x_2$ | $x_1$ | $x_0$ | $U = X^2$ | $u_5$ | $u_4$ | $u_3$ | $u_2$ | $u_1$ | $u_0$ |
|---|-------|-------|-------|-----------|-------|-------|-------|-------|-------|-------|
| 0 | 0     | 0     | 0     | 0         | 0     | 0     | 0     | 0     | 0     | 0     |
| 1 | 0     | 0     | 1     | 1         | 0     | 0     | 0     | 0     | 0     | 1     |
| 2 | 0     | 1     | 0     | 4         | 0     | 0     | 0     | 1     | 0     | 0     |
| 3 | 0     | 1     | 1     | 9         | 0     | 0     | 1     | 0     | 0     | 1     |
| 4 | 1     | 0     | 0     | 16        | 0     | 1     | 0     | 0     | 0     | 0     |
| 5 | 1     | 0     | 1     | 25        | 0     | 1     | 1     | 0     | 0     | 1     |
| 6 | 1     | 1     | 0     | 36        | 1     | 0     | 0     | 1     | 0     | 0     |
| 7 | 1     | 1     | 1     | 49        | 1     | 1     | 0     | 0     | 0     | 1     |

Av sanningstabellen framgår att  $u_1$  alltid är lika med 0.  $u_1$  utgången kan därför anslutas 0V (jord) så att den får konstanten 0. Man kan vidare se att  $u_0$  alltid är samma som  $x_0$ .  $u_0$  utgången kan därför förbindas direkt med  $x_0$  ingången. De övriga funktionerna tas fram med Karnaughdiagrammen.

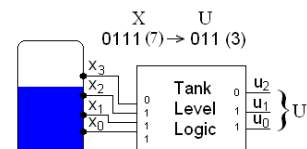


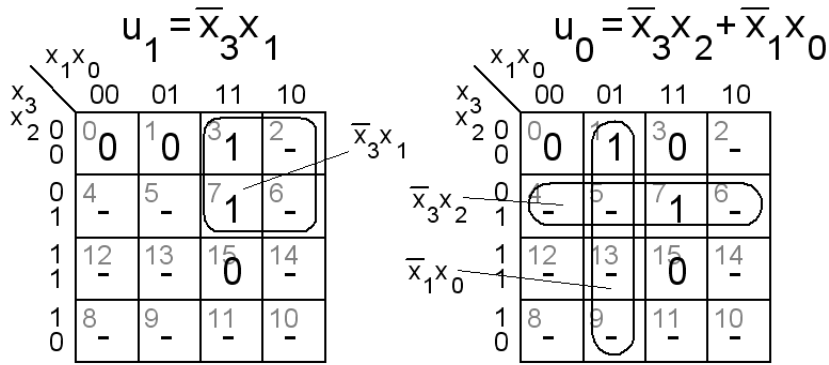
## 8.2

| X  | $x_3$ | $x_2$ | $x_1$ | $x_0$ | U | $u_2$ | $u_1$ | $u_0$ |
|----|-------|-------|-------|-------|---|-------|-------|-------|
| 0  | 0     | 0     | 0     | 0     | 0 | 0     | 0     | 0     |
| 1  | 0     | 0     | 0     | 1     | 1 | 0     | 0     | 1     |
| 3  | 0     | 0     | 1     | 1     | 2 | 0     | 1     | 0     |
| 7  | 0     | 1     | 1     | 1     | 3 | 0     | 1     | 1     |
| 15 | 1     | 1     | 1     | 1     | 4 | 1     | 0     | 0     |

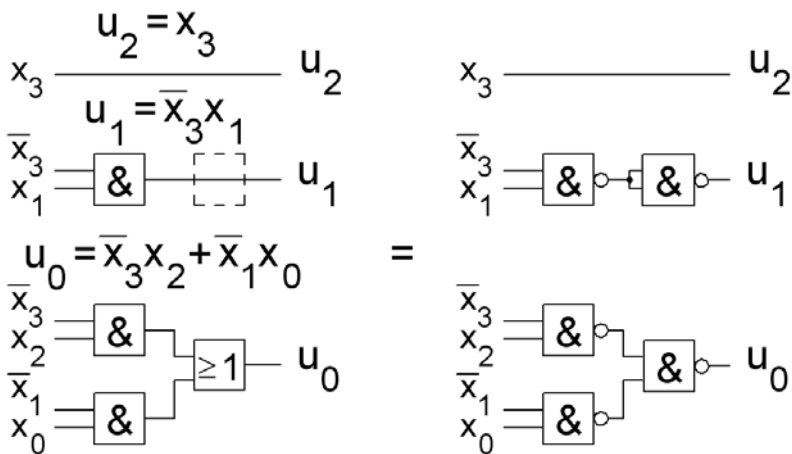
Vi kan direkt se i tabellen att  $u_2$  och  $x_3$  är lika varför  $u_2$  kan anslutas direkt till  $x_3$ .  $u_2 = x_3$ .

De övriga uttrycken fås med hjälp av deras Karnaughdiagram.





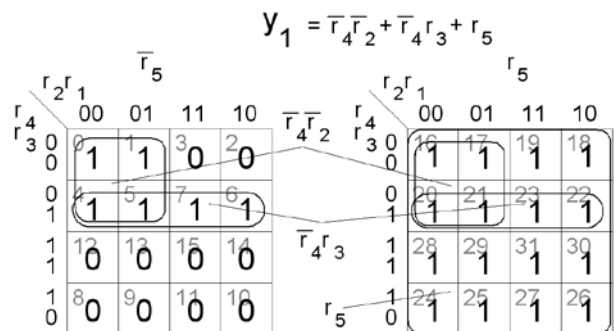
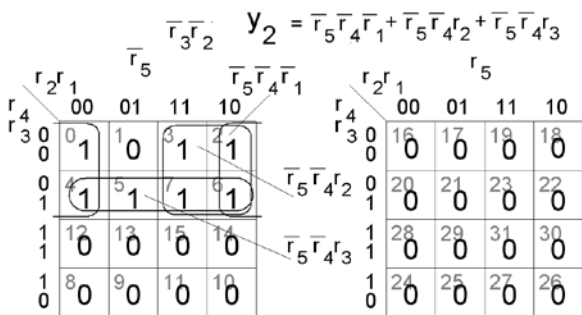
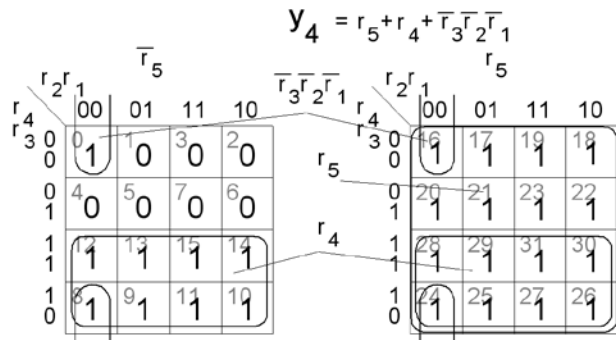
Gründnät:



### 8.3

Sanningstabell

| $r_5$ | $r_4$ | $r_3$ | $r_2$ | $r_1$ | $y_4$ | $y_2$ | $y_1$ |
|-------|-------|-------|-------|-------|-------|-------|-------|
| 1     | -     | -     | -     | -     | 1     | 0     | 1     |
| 0     | 1     | -     | -     | -     | 1     | 0     | 0     |
| 0     | 0     | 1     | -     | -     | 0     | 1     | 1     |
| 0     | 0     | 0     | 1     | -     | 0     | 1     | 0     |
| 0     | 0     | 0     | 0     | 1     | 0     | 0     | 1     |
| 0     | 0     | 0     | 0     | 0     | 1     | 1     | 1     |



### 8.4

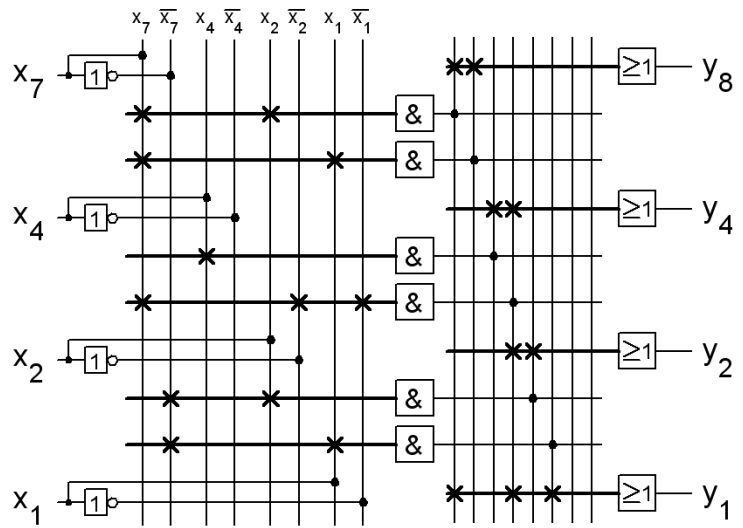
|      | 7     | 4     | 2     | 1     | 8     | 4     | 2     | 1     |
|------|-------|-------|-------|-------|-------|-------|-------|-------|
|      | $x_7$ | $x_4$ | $x_2$ | $x_1$ | $y_8$ | $y_4$ | $y_2$ | $y_1$ |
| (0)  | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
| (1)  | 0     | 0     | 0     | 1     | 1     | 0     | 0     | 1     |
| (2)  | 0     | 0     | 1     | 0     | 2     | 0     | 0     | 1     |
| (3)  | 0     | 0     | 1     | 1     | 3     | 0     | 0     | 1     |
| (4)  | 0     | 1     | 0     | 0     | 4     | 0     | 1     | 0     |
| (5)  | 0     | 1     | 0     | 1     | 5     | 0     | 1     | 0     |
| (6)  | 0     | 1     | 1     | 0     | 6     | 0     | 1     | 1     |
| (8)  | 1     | 0     | 0     | 0     | 7     | 0     | 1     | 1     |
| (9)  | 1     | 0     | 0     | 1     | 8     | 1     | 0     | 0     |
| (10) | 1     | 0     | 1     | 0     | 9     | 1     | 0     | 1     |

$$y_8 = x_7 x_2 + x_7 x_1$$

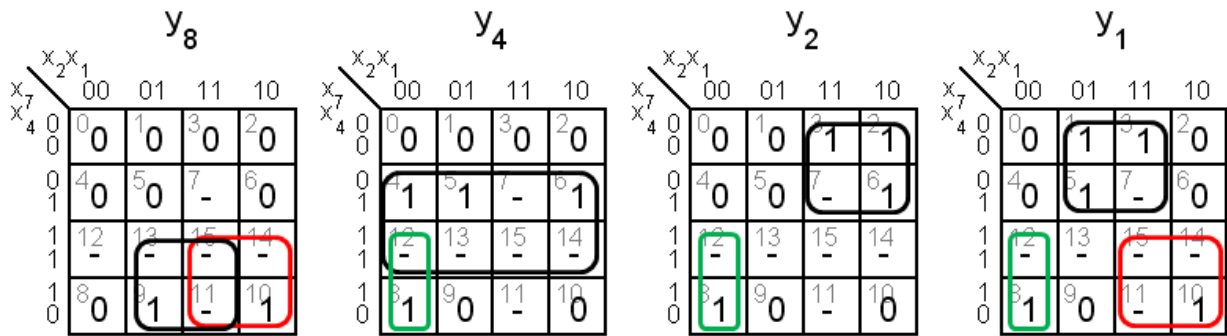
$$y_4 = x_4 + x_7 x_2 x_1$$

$$y_2 = x_7 x_2 + x_7 x_2 x_1$$

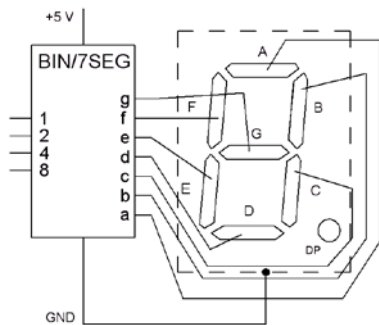
$$y_1 = x_7 x_1 + x_7 x_2 + x_7 x_2 x_1$$



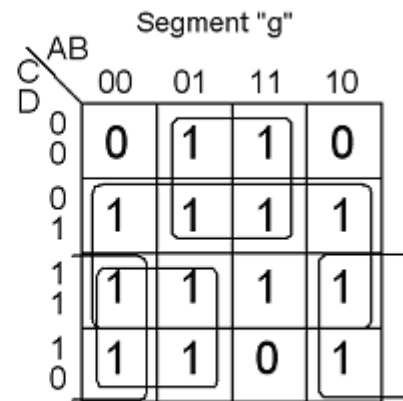
Två av AND-grindarna kan *delas* av  $y_8, y_1$  och  $y_8, y_2, y_1$ -näten.



### 8.5



| DCBA    | abcdefg       |
|---------|---------------|
| 0 0 0 0 | 1 1 1 1 1 1 0 |
| 0 0 0 1 | 0 1 1 0 0 0 0 |
| 0 0 1 0 | 1 1 0 1 1 0 1 |
| 0 0 1 1 | 1 1 1 1 0 0 1 |
| 0 1 0 0 | 0 1 1 0 0 1 1 |
| 0 1 0 1 | 1 0 1 1 0 1 1 |
| 0 1 1 0 | 1 0 1 1 1 1 1 |
| 0 1 1 1 | 1 1 1 0 0 0 0 |
| 1 0 0 0 | 1 1 1 1 1 1 1 |
| 1 0 0 1 | 1 1 1 0 0 1 1 |
| 1 0 1 0 | 1 1 1 0 1 1 1 |
| 1 0 1 1 | 0 0 1 1 1 1 1 |
| 1 1 0 0 | 0 0 0 1 1 0 1 |
| 1 1 0 1 | 0 1 1 1 1 0 1 |
| 1 1 1 0 | 1 0 0 1 1 1 1 |
| 1 1 1 1 | 1 0 0 0 1 1 1 |

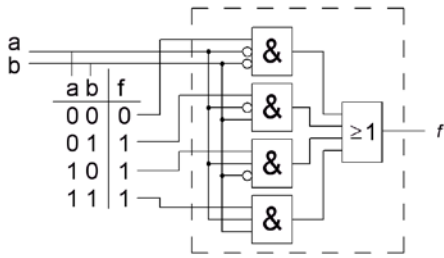


$$g = D + B\bar{C} + \bar{B}C + \bar{A}C$$

### 8.6

En 4-1 multiplexor som funktionsgenerator för OR-funktionen.

Multiplexor som funktionsgenerator



### 8.7

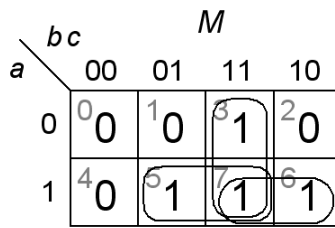
|   | a | b | c | M |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 2 | 0 | 1 | 0 | 0 |
| 3 | 0 | 1 | 1 | 1 |
| 4 | 1 | 0 | 0 | 0 |
| 5 | 1 | 0 | 1 | 1 |
| 6 | 1 | 1 | 0 | 1 |
| 7 | 1 | 1 | 1 | 1 |

$\bar{a}bc$

$a\bar{b}c$

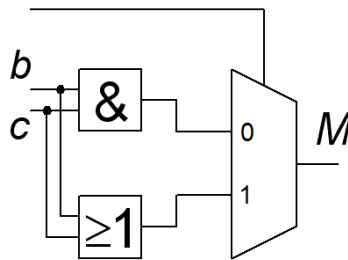
$ab\bar{c}$

$abc$



$$M = bc + ac + ab$$

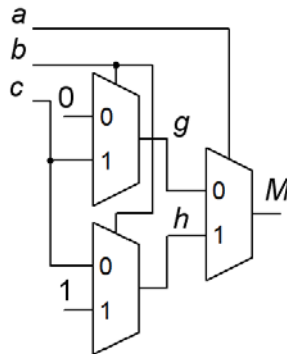
$$M = \bar{a}bc + a\bar{b}c + ab\bar{c} + abc = \bar{a}(bc) + a(\bar{b}c + b\bar{c} + bc) = \bar{a}(bc) + a(b+c)$$



$$M = \bar{a}(bc) + a(b+c) \quad g = bc \quad h = b+c$$

$$g = \bar{b}(0) + b(c) = bc$$

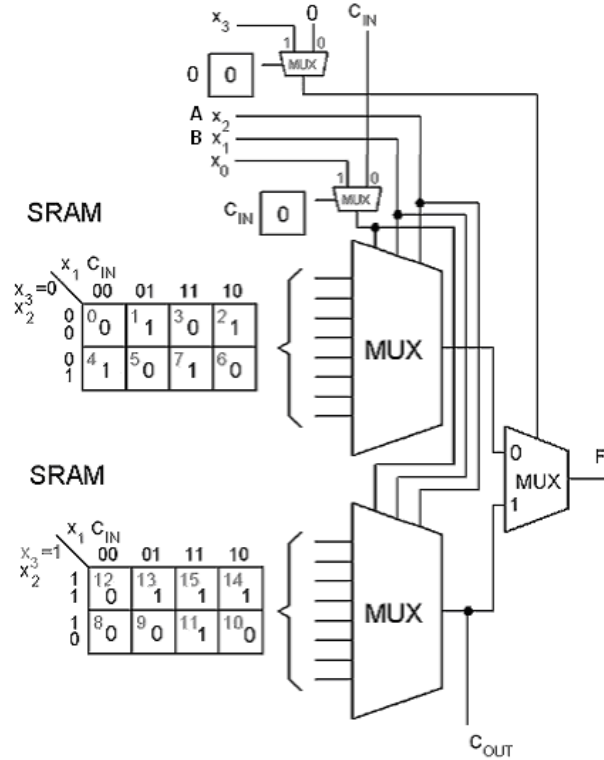
$$h = \bar{b}(c) + b(1) = \bar{b}c + b(\bar{c} + c) = \bar{b}c + b\bar{c} + bc = c(b + \bar{b}) + b(c + \bar{c}) = b+c$$



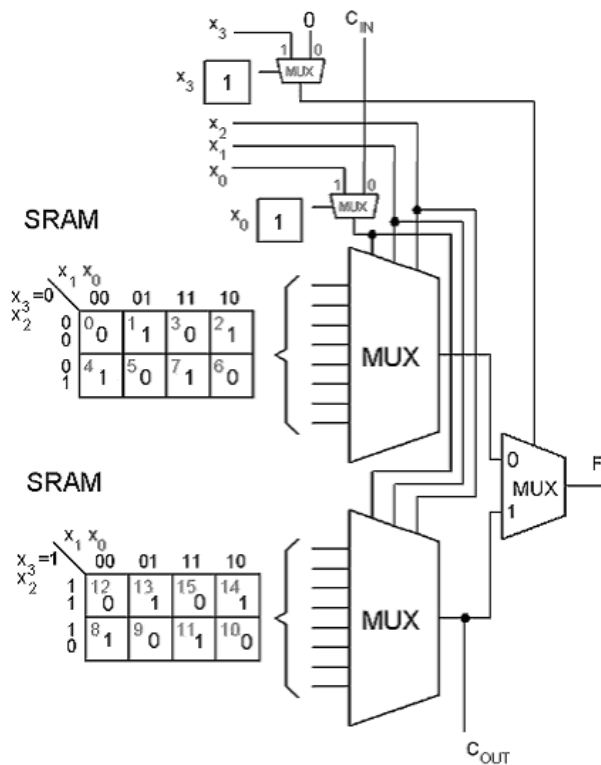
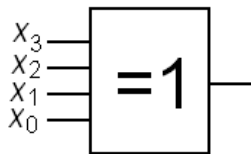
### 8.8

För att göra en heladderare behöver vi använda den övre MUXen till summafunktionen, och den nedre MUXen, som alltid är ansluten till  $C_{OUT}$ , används till Carry-funktionen. I stället för  $x_0$  väljer vi  $C_{IN}$ . För att den övre MUXen skall anslutas till logikelementets utgång ska utgångsmuxen styras med 0 i stället för med  $x_3$ .

| $x_2$ | $x_1$ | $x_0$    | $\Sigma$ | $C_{OUT}$ |
|-------|-------|----------|----------|-----------|
| A     | B     | $C_{IN}$ | $\Sigma$ | $C_{OUT}$ |
| 0     | 0     | 0        | 0        | 0         |
| 0     | 0     | 1        | 1        | 0         |
| 0     | 1     | 0        | 1        | 0         |
| 0     | 1     | 1        | 0        | 1         |
| 1     | 0     | 0        | 1        | 0         |
| 1     | 0     | 1        | 0        | 1         |
| 1     | 1     | 0        | 0        | 1         |
| 1     | 1     | 1        | 1        | 1         |

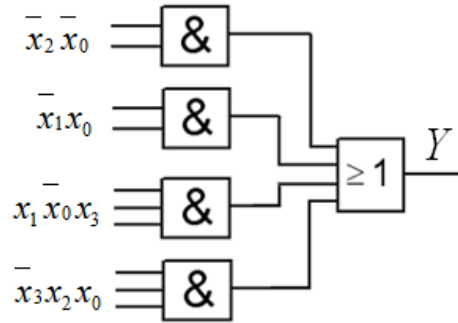


### 8.9



8.10

|       |       |          |    |     |    |    |   |
|-------|-------|----------|----|-----|----|----|---|
|       |       | $x_1x_0$ |    | $Y$ |    |    |   |
|       |       | 00       | 01 |     | 11 | 10 |   |
| $x_3$ | $x_2$ | 0        | 0  | 1   | 0  | 0  | 1 |
|       |       | 0        | -  | 3   | 0  | 2  | 1 |
| 0     | 0     | 4        | 0  | 5   | 1  | 6  | 0 |
|       | 1     | 0        | 1  | 1   | 0  | 0  |   |
| 1     | 1     | 12       | 0  | 13  | 1  | 14 | 0 |
|       | 1     | 0        | 1  | 0   | 1  | 1  |   |
| 1     | 0     | 8        | -  | 9   | -  | 10 | 0 |
|       | 1     | 0        | 1  | 0   | 1  | 1  |   |



$$Y = \bar{x}_2 \bar{x}_0 + \bar{x}_1 x_0 + x_1 x_0 x_3 + x_3 x_2 x_0$$

|       |   |       |   |
|-------|---|-------|---|
|       |   | $x_0$ |   |
|       |   | 0     | 1 |
| $x_1$ | 0 | -     | - |
|       | 1 | 1     | 0 |

$$x_3x_2(0,0) \Rightarrow Y = \bar{x}_0$$

|       |   |       |   |
|-------|---|-------|---|
|       |   | $x_0$ |   |
|       |   | 0     | 1 |
| $x_1$ | 0 | 0     | 1 |
|       | 1 | 0     | 1 |

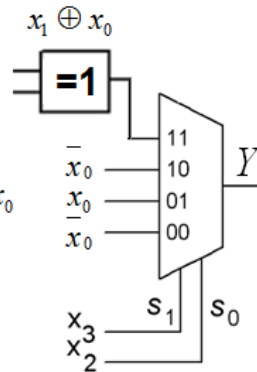
$$x_3x_2(0,1) \Rightarrow Y = x_0$$

|       |   |       |   |
|-------|---|-------|---|
|       |   | $x_0$ |   |
|       |   | 0     | 1 |
| $x_1$ | 0 | 0     | 1 |
|       | 1 | 1     | 0 |

$$x_3x_2(1,1) \Rightarrow Y = x_1 \oplus x_0$$

|       |   |       |   |
|-------|---|-------|---|
|       |   | $x_0$ |   |
|       |   | 0     | 1 |
| $x_1$ | 0 | -     | - |
|       | 1 | 1     | 0 |

$$x_3x_2(1,0) \Rightarrow Y = \bar{x}_0$$



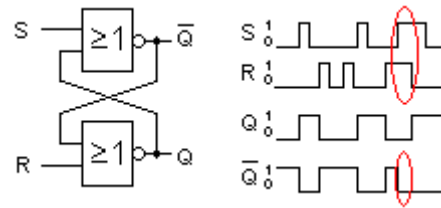
Alternativt kan XOR-grinden även användas till MUX-ingångarna 00 och 10.

# Sekvenskretsar, låskretsar och klockade vippor

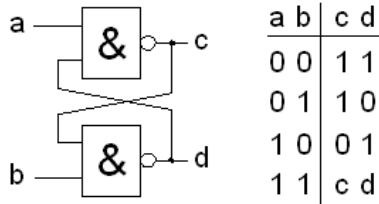
## 9.1

Figuren visar en SR-latch, men mot slutet av insignals-sekvensen förekommer den "förbjudna" insignalskombinationen  $S = 1, R = 1$ . Utgångarna blir inte varandras inverser för denna kombination.

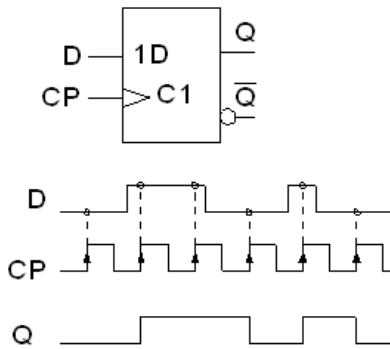
För NOR-grindar är 1 läsande insignal, därför blir  $\bar{Q} = 0$  så länge  $S$  är kvar på 1.



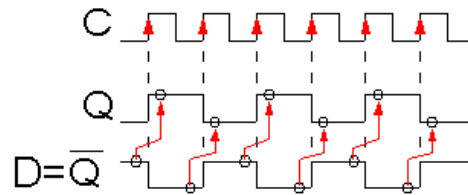
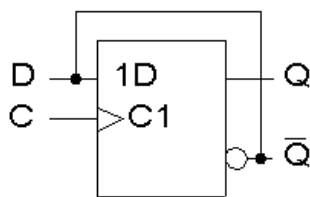
## 9.2



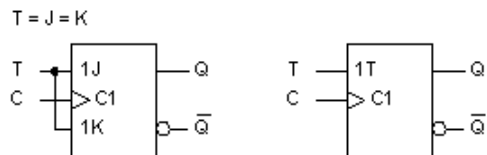
## 9.3



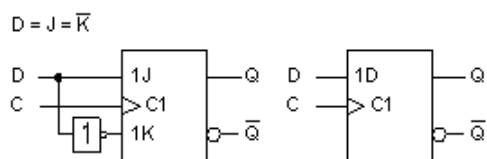
## 9.4



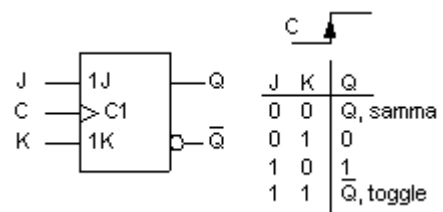
## 9.5



| T | Q         |
|---|-----------|
| 0 | Q, samma  |
| 1 | Q, toggle |



| D | Q    |
|---|------|
| 0 | D, 0 |
| 1 | D, 1 |

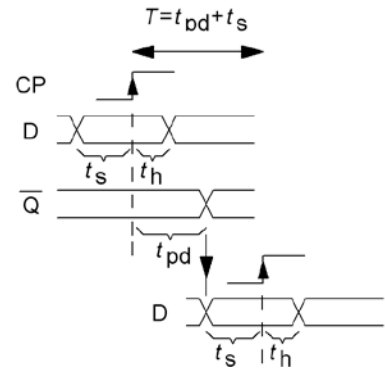
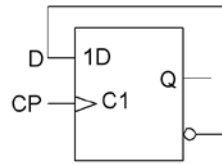


JK-vippan kan användas som T-vippa eller som D-vippa. (När vippor kopplas ihop med varandra finns det inverterade utgångar att tillgå, då behövs inte inverteraren för att göra JK-vippan till D-vippa.)

### 9.6

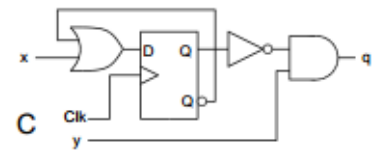
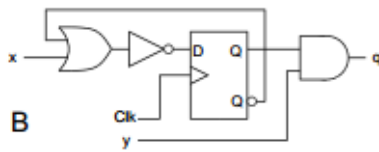
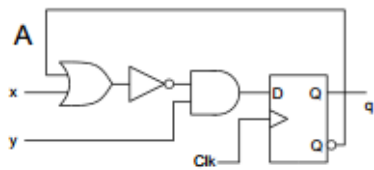
$$T = t_{pd} + t_s$$

$$f = \frac{1}{T} = \frac{1}{t_{pd} + t_s} = \frac{1}{(20 + 30)[\text{ns}]} = 20 \text{ MHz}$$

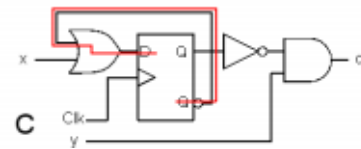


### 9.7

$$t_{\text{AND}} = 0,4 \text{ ns}, t_{\text{OR}} = 0,4 \text{ ns}, t_{\text{NOT}} = 0,1 \text{ ns}, t_{\text{setup}} = 0,3 \text{ ns}, t_{\text{dq}} = 0,4 \text{ ns}$$



$$T = T_{\text{OR}} + T_{\text{setup}} + T_{\text{dq}} = 0,4 + 0,3 + 0,4 = 1,1 \text{ ns}$$





# Sekvensnät, automater

## 10.1

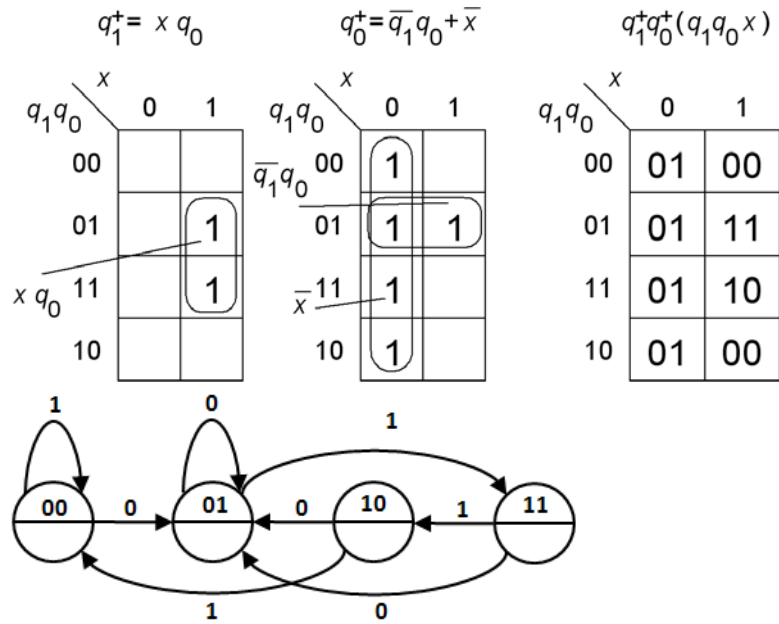
Ur kretsschemat kan följande samband ställas upp:

$$q_1 \quad q_0$$

$$q_1^+ = x \cdot q_0$$

$$q_0^+ = \bar{x} + \bar{q}_1 \cdot q_0$$

Ingen utgångsavkodare finns utan vippornas tillstånd används direkt. Moore-modellen skall användas.



## 10.2

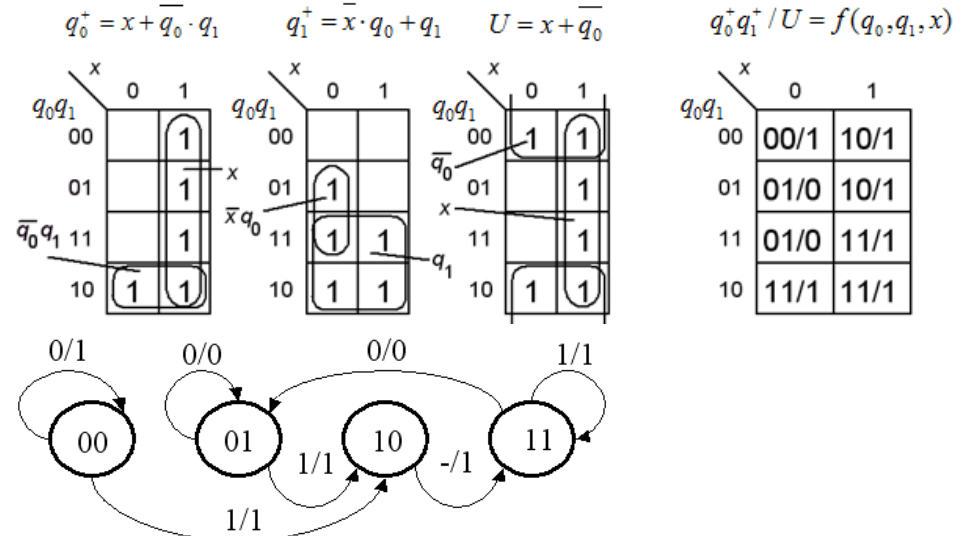
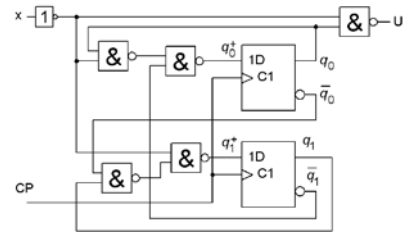
Ur kretsschemat kan följande samband ställas upp:

$$U = \bar{x} \cdot q_0 = x + \bar{q}_0$$

$$q_0^+ = \bar{q}_1 \cdot (q_0 \cdot x) = x + \bar{q}_0 \cdot q_1$$

$$q_1^+ = (q_1 \cdot q_0) \cdot x = \bar{x} \cdot q_0 + q_1$$

Eftersom  $U$  beror direkt av  $x$  så måste Mealy-modellen användas.



### 10.3

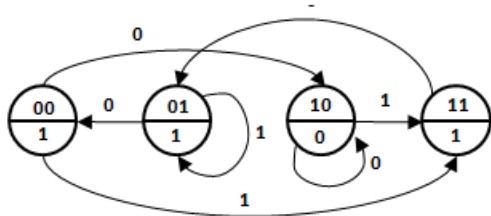
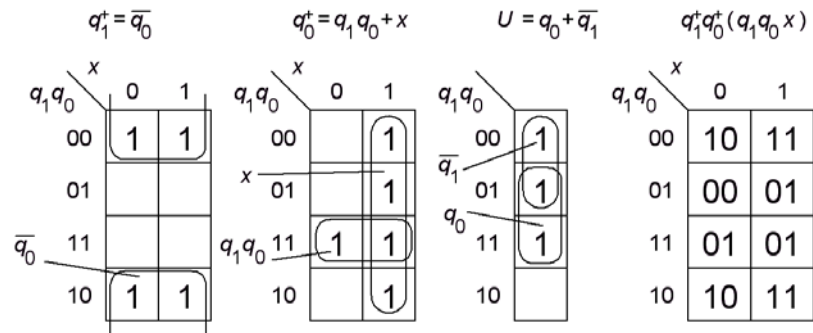
Ur kretsschemat kan följande samband ställas upp:

$$U = \overline{q_0} \cdot q_1 = q_0 + \overline{q_1}$$

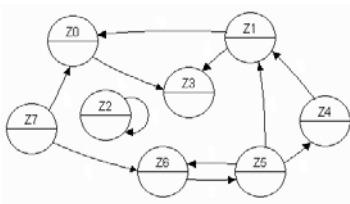
$$q_1^+ = \overline{q_0}$$

$$q_0^+ = x \cdot q_0 \cdot q_1 = q_1 q_0 + x$$

Eftersom  $U$  bara beror på tillståndet och är oberoende av  $x$  så måste Moore-modellen användas.

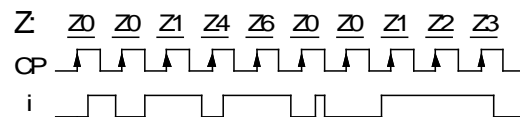
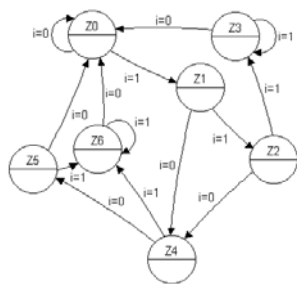


### 10.4.



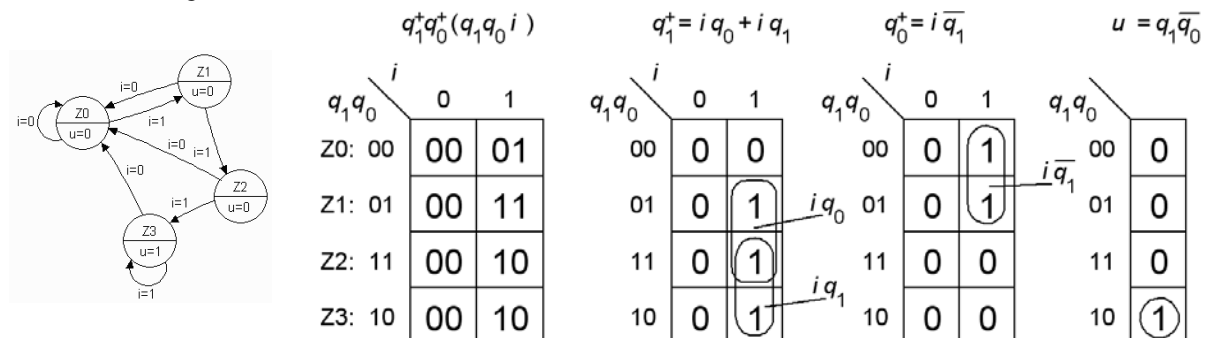
Stopptillstånd: Z3  
 Förlusttillstånd: Z7  
 Isolerade tillstånd: Z2

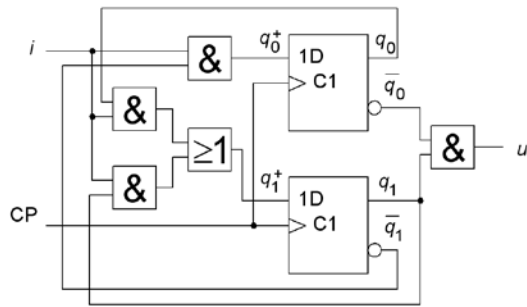
### 10.5



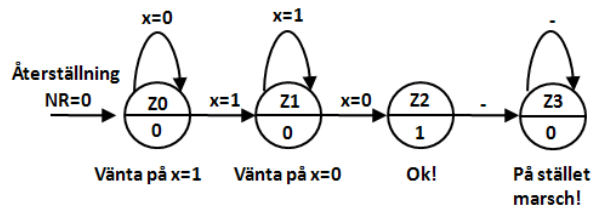
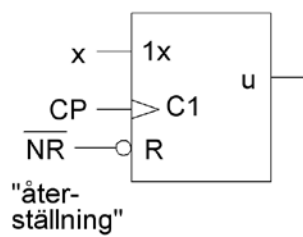
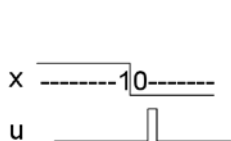
### 10.6

Från tillståndsdigram till kodad tillståndstabell.





### 10.7



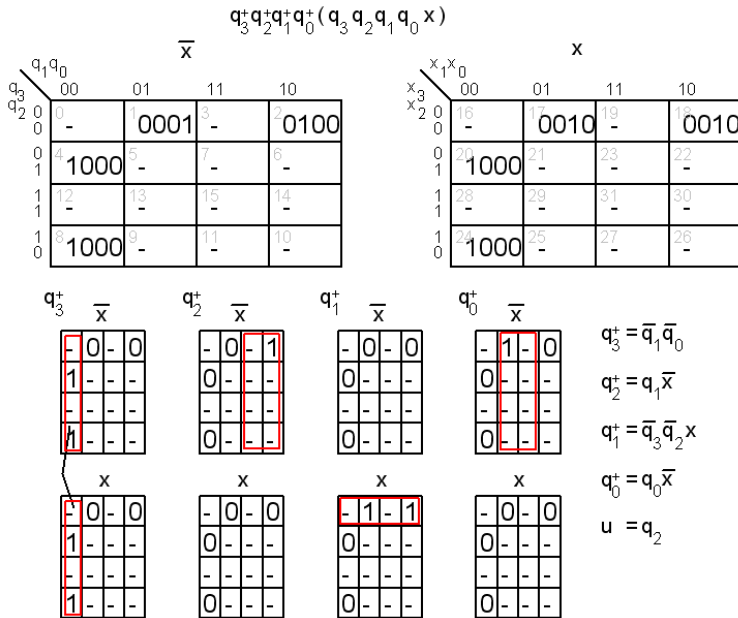
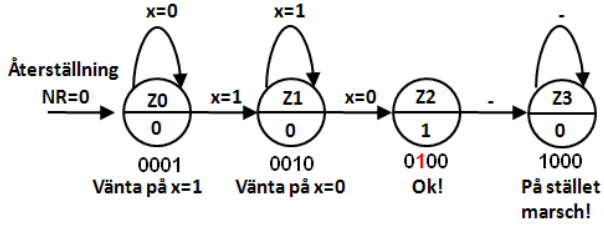
Tillståndskodning Binär:

| Bin       | $q_1^+ q_0^+ (q_1 q_0 x)$ | $q_1^+ = q_0 \bar{x} + q_1$ | $q_0^+ = q_1 + x$ | $U = q_1 \bar{q}_0$ |
|-----------|---------------------------|-----------------------------|-------------------|---------------------|
| $q_1 q_0$ |                           |                             |                   |                     |
| $x$       | 0 1                       | 0 1                         | 0 1               |                     |
| Z0: 00    | 00 01                     | 0 0                         | 0 1               | 0                   |
| Z1: 01    | 10 01                     | 1 0                         | 0 1               | 0                   |
| Z3: 11    | 11 11                     | 1 1                         | 1 1               | 0                   |
| Z2: 10    | 11 11                     | 1 1                         | 1 1               | 1                   |

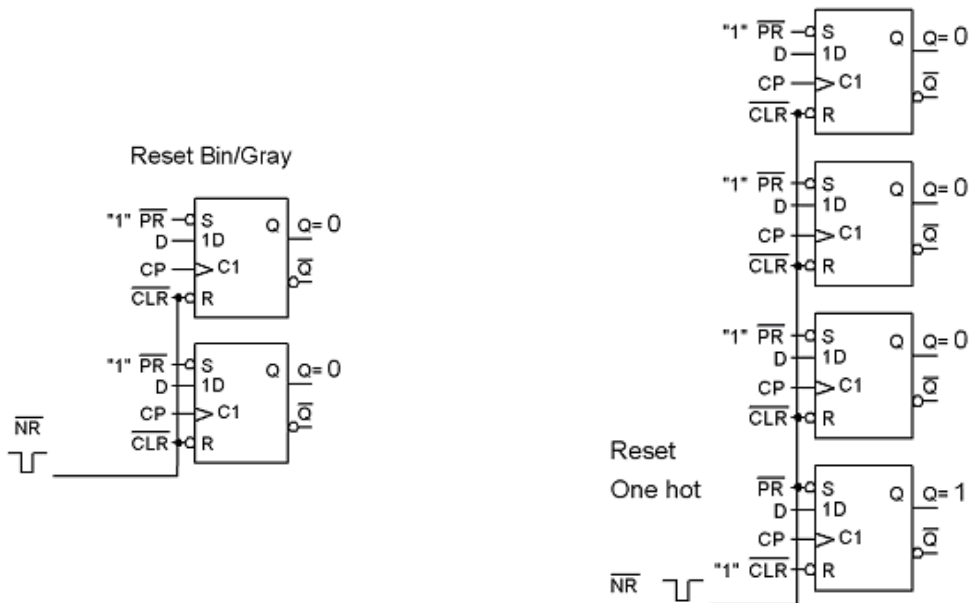
Tillståndskodning Gray:

| Gray      | $q_1^+ q_0^+ (q_1 q_0 x)$ | $q_1^+ = q_0 \bar{x} + q_1$ | $q_0^+ = \bar{q}_1 q_0 + \bar{q}_1 x$ | $U = q_1 q_0$ |
|-----------|---------------------------|-----------------------------|---------------------------------------|---------------|
| $q_1 q_0$ |                           |                             |                                       |               |
| $x$       | 0 1                       | 0 1                         | 0 1                                   |               |
| Z0: 00    | 00 01                     | 0 0                         | 0 1                                   | 0             |
| Z1: 01    | 11 01                     | 1 0                         | 1 1                                   | 0             |
| Z2: 11    | 10 10                     | 1 1                         | 0 0                                   | 1             |
| Z3: 10    | 10 10                     | 1 1                         | 0 0                                   | 0             |

Tillståndskodning One hot:

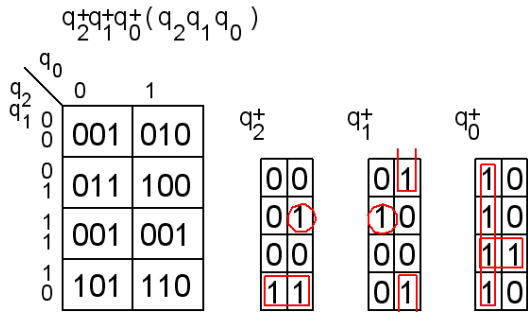
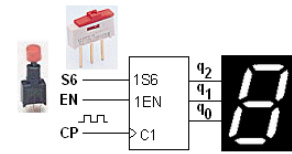


Så här gör man återställning till "00" med CLR ingångarna för Bin/Gray tillståndskodningen, respektive till "0001" med PRE/CLR ingångarna för "one hot" tillståndskodningen.



### 10.8

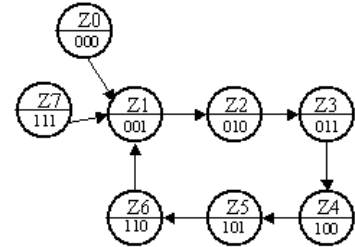
För sex tillstånd krävs det tre vippor. Då finns det 8 tillstånd totalt, två tillstånd som inte ingår i sekvensen. Vi specificerar för säkerhets skull vad som skall ske med dom tillstånden, så att räknaren inte skall riskera att "fastna".



$$q_2^+ = q_2 \bar{q}_1 + \bar{q}_2 q_1 q_0$$

$$q_1^+ = \bar{q}_1 q_0 + \bar{q}_2 q_1 \bar{q}_0$$

$$q_0^+ = \bar{q}_0 + q_2 q_1$$



Omskrivning med EN ( $EN=0 \rightarrow$  på stället marsch) :

$$(q_2^+)' = EN \cdot (q_2^+) + \bar{EN} \cdot (q_2)$$

$$(q_1^+)' = EN \cdot (q_1^+) + \bar{EN} \cdot (q_1)$$

$$(q_0^+)' = EN \cdot (q_0^+) + \bar{EN} \cdot (q_0)$$

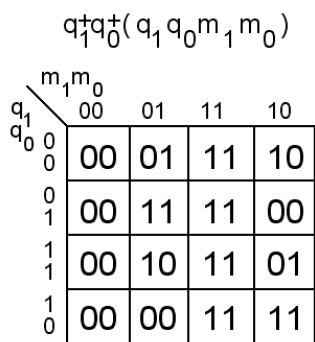
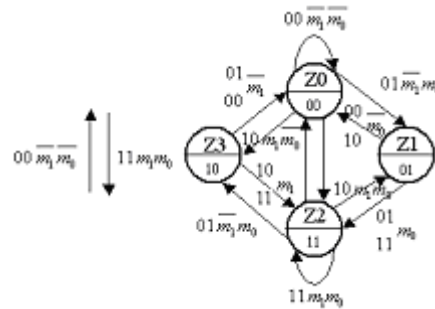
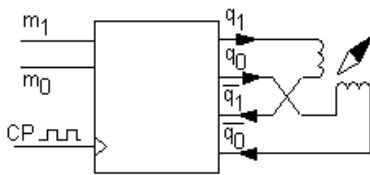
Omskrivning med S6 ( $S6=1 \rightarrow$  nästa tillstånd 110) :

$$(q_2^+)'' = (q_2^+)' + S6$$

$$(q_1^+)'' = (q_1^+)' + S6$$

$$(q_0^+)'' = (q_0^+)' \cdot \bar{S6}$$

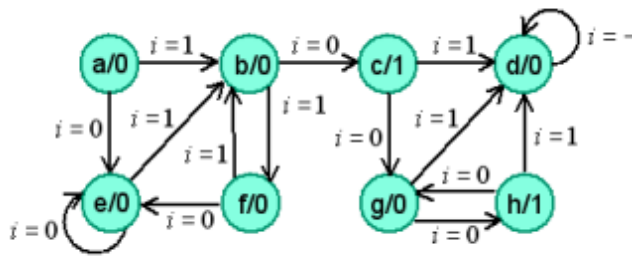
### 10.9



$$q_1^+ = q_0 m_0 + \bar{q}_0 m_1$$

$$q_0^+ = q_1 m_1 + \bar{q}_1 m_0$$

10.10



Skriv tillståndstabell.

Två tillstånd kan *inte* vara ekvivalenta om utsignalerna är olika eller om efterföljande tillstånds utsignaler är olika.



now next out

| $i =$ | 0   | 1   | $z$ |
|-------|-----|-----|-----|
| $a$   | $e$ | $b$ | 0   |
| $b$   | $c$ | $f$ | 0   |
| $c$   | $g$ | $d$ | 1   |
| $d$   | $d$ | $d$ | 0   |
| $e$   | $e$ | $b$ | 0   |
| $f$   | $e$ | $b$ | 0   |
| $g$   | $h$ | $d$ | 0   |
| $h$   | $g$ | $d$ | 1   |

Grupper med samma utsignal:

$$P_1 = (a, b, d, e, f, g)(c, h)$$

Undersök efterföljande tillstånd:

|  |  |
|--|--|
| $a_{i=0} \rightarrow (a, b, d, e, f, g)$ | $a_{i=1} \rightarrow (a, b, d, e, f, g)$ |
| $b_{i=0} \rightarrow (c, h)$             | $b_{i=1} \rightarrow (a, b, d, e, f, g)$ |
| $d_{i=0} \rightarrow (a, b, d, e, f, g)$ | $d_{i=1} \rightarrow (a, b, d, e, f, g)$ |
| $e_{i=0} \rightarrow (a, b, d, e, f, g)$ | $e_{i=1} \rightarrow (a, b, d, e, f, g)$ |
| $f_{i=0} \rightarrow (a, b, d, e, f, g)$ | $f_{i=1} \rightarrow (a, b, d, e, f, g)$ |
| $g_{i=0} \rightarrow (c, h)$             | $g_{i=1} \rightarrow (a, b, d, e, f, g)$ |

$(b, g)$  bildar egen grupp.

$$P_2 = (a, d, e, f)(b, g)(c, h)$$

$$P_2 = (a, d, e, f)(b, g)(c, h)$$

Undersök efterföljande tillstånd:

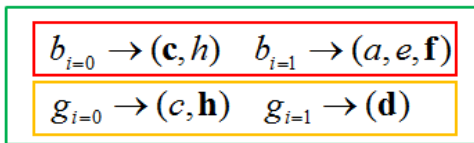
|                                    |                                    |
|------------------------------------|------------------------------------|
| $a_{i=0} \rightarrow (a, d, e, f)$ | $a_{i=1} \rightarrow (b, g)$       |
| $d_{i=0} \rightarrow (a, d, e, f)$ | $d_{i=1} \rightarrow (a, d, e, f)$ |
| $e_{i=0} \rightarrow (a, d, e, f)$ | $e_{i=1} \rightarrow (b, g)$       |
| $f_{i=0} \rightarrow (a, d, e, f)$ | $f_{i=1} \rightarrow (b, g)$       |

$(d)$  bildar en egen grupp.

$$P_3 = (a, e, f)(b, g)(d)(c, h)$$

$$P_3 = (a, e, f)(\mathbf{b, g})(d)(c, h)$$

Undersök efterföljande tillstånd:

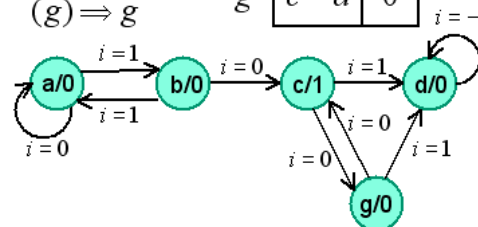


(b) (g) bildar egna grupper.

$$P_4 = (a, e, f)(\mathbf{b})(d)(\mathbf{g})(c, h)$$

Byte av beteckningar

|         | now | next | out |
|---------|-----|------|-----|
| $i = 0$ | a   | b    | 0   |
| $i = 1$ | b   | c    | a   |
| $i = z$ | c   | g    | d   |
|         | d   | d    | 0   |
|         | g   | c    | d   |



### 10.11

Grupper med samma utsignal:

$$P_1 = (a, b, c, f)(d, e)$$

Undersök efterföljande tillstånd:

| i: | next |   | out |
|----|------|---|-----|
|    | 0    | 1 | z   |
| a  | e    | c | 0   |
| b  | d    | c | 0   |
| c  | d    | e | 0   |
| d  | f    | a | 1   |
| e  | f    | b | 1   |
| f  | f    | b | 0   |

$$a_{i=0} \rightarrow (d, e) \quad a_{i=1} \rightarrow (a, b, c, f)$$

$$b_{i=0} \rightarrow (d, e) \quad b_{i=1} \rightarrow (a, b, c, f)$$

$$c_{i=0} \rightarrow (d, e) \quad c_{i=1} \rightarrow (d, e)$$

$$f_{i=0} \rightarrow (a, b, c, f) \quad f_{i=1} \rightarrow (a, b, c, f)$$

$$d_{i=0} \rightarrow (a, b, c, f) \quad d_{i=1} \rightarrow (a, b, c, f)$$

$$e_{i=0} \rightarrow (a, b, c, f) \quad e_{i=1} \rightarrow (a, b, c, f)$$

Grupper med också samma efterföljare:

$$P_2 = (a, b)(c)(f)(d, e)$$

$$P_3 = P_2$$

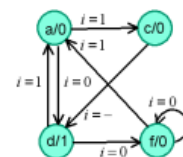
$$(a, b) \rightarrow a$$

$$(c) \rightarrow c$$

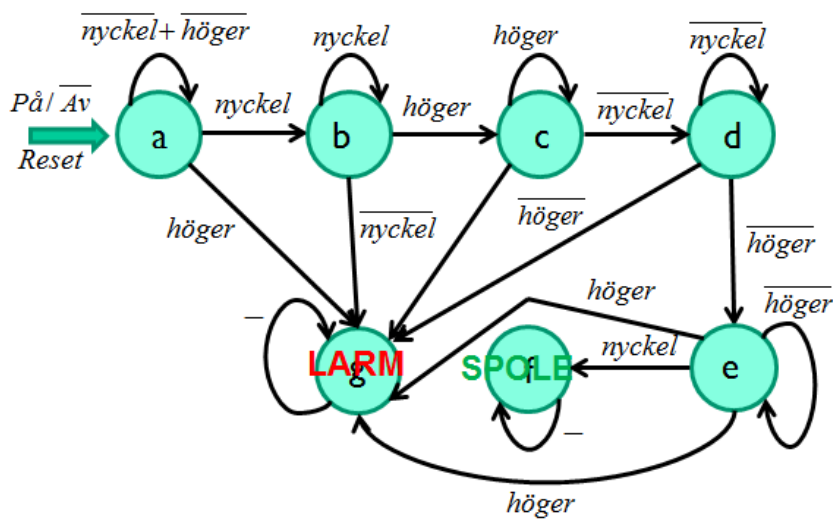
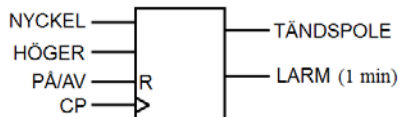
$$(f) \rightarrow f$$

$$(d, e) \rightarrow d$$

|   | next | out |   |
|---|------|-----|---|
|   | 0    | 1   | z |
| a | d    | c   | 0 |
| c | d    | d   | 0 |
| d | f    | a   | 1 |
| f | f    | a   | 0 |

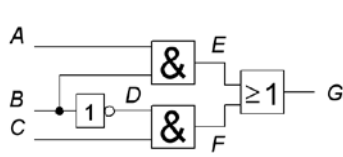


### 10.12



# Asynkrona sekvensnät

## 11.1



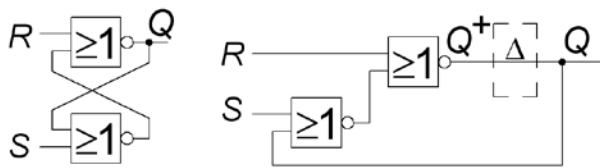
|   |    |    |    |    |
|---|----|----|----|----|
|   | BC |    | G  |    |
| A | 00 | 01 | 11 | 10 |
| 0 | 0  | 1  | 3  | 2  |
| 1 | 4  | 5  | 7  | 6  |

|   |    |    |    |    |
|---|----|----|----|----|
|   | BC |    | G  |    |
| A | 00 | 01 | 11 | 10 |
| 0 | 0  | 1  | 3  | 2  |
| 1 | 4  | 5  | 7  | 6  |

$G = \overline{BC} + AB$  {Hazardfritt}  $G = \overline{BC} + AB + AC$

## 11.2

Till vänster i figuren visas en SR-låskrets som två "korskopplade" grindnät. Till höger är kretsen omritad som en Moore-automat.

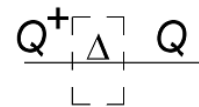


|   |    |    |                |    |
|---|----|----|----------------|----|
|   | SR |    | Q <sup>+</sup> |    |
| Q | 00 | 01 | 11             | 10 |
| 0 | 0  | 1  | 3              | 2  |
| 1 | 4  | 5  | 7              | 6  |

$Q^+ = \overline{\overline{R+S+Q}} = \overline{R \cdot \overline{(S+Q)}} = \overline{R \cdot (S+Q)} = S\overline{R} + \overline{R}Q$

Den kodade tillståndstabellen brukar kallas för **excitationstabell** när man arbetar med asynkrona tillståndsmaskiner.

| Nuvarande tillstånd Q | Nästa tillstånd Q <sup>+</sup> |    |    |    |
|-----------------------|--------------------------------|----|----|----|
|                       | Insignaler SR                  |    |    |    |
|                       | 00                             | 01 | 11 | 10 |
| 0                     | 0                              | 0  | 0  | 1  |
| 1                     | 1                              | 0  | 0  | 1  |

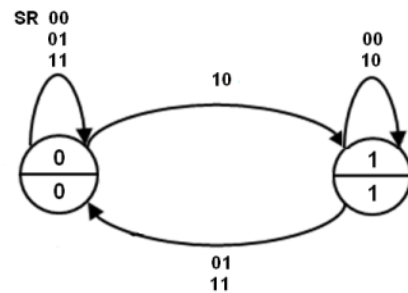


För varje insignal (kolumn) måste det finnas åtminstone något tillstånd där  $Q = Q^+$ . Sådana tillstånd är *stabila* och de brukar markeras genom att ringas in.

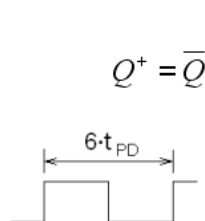
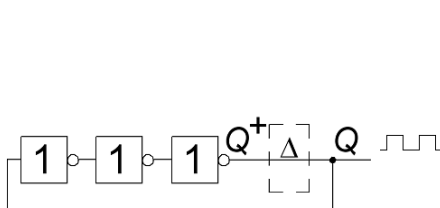
Tillståndsdigrammet följer ur excitationstabellen.

Den okodade tillståndstabellen brukar kallas för **flödestabell** när det gäller asynkrona sekvensnät.

| Nuvarande tillstånd Q | Nästa tillstånd Q <sup>+</sup> |    |    |    |
|-----------------------|--------------------------------|----|----|----|
|                       | Insignaler SR                  |    |    |    |
|                       | 00                             | 01 | 11 | 10 |
| A                     | A                              | A  | A  | B  |
| B                     | B                              | A  | A  | B  |



## 11.3



$Q^+ = \overline{Q}$

|   |                |
|---|----------------|
| Q | Q <sup>+</sup> |
| 0 | 1              |
| 1 | 0              |

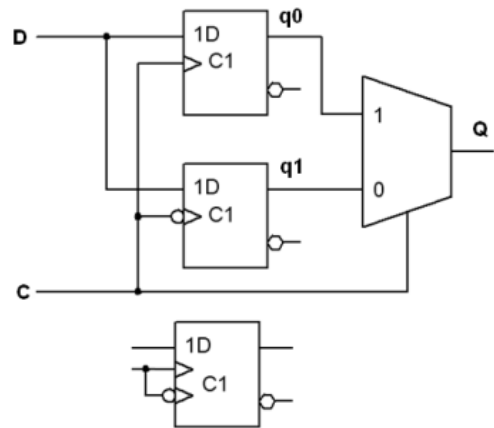
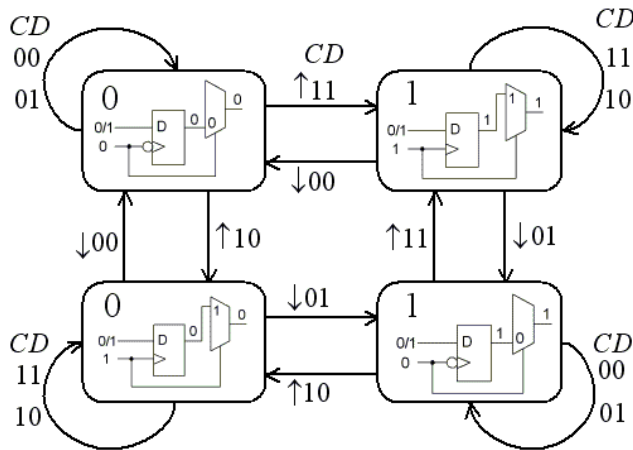
*Inga stabila tillstånd!*

$T = 6 \cdot t_{PD} \Rightarrow f = \frac{1}{6 \cdot t_{PD}}$

Sifferexempel:  $t_{pd} = 5 \cdot 10^{-9} \Rightarrow f = \frac{1}{6 \cdot 5 \cdot 10^{-9}} = 33 \text{ MHz}$



### 11.4



- Vid positiv flank  $\uparrow$  går **C** från 0 till 1 och **C=1** kopplar den övre **q0** vippan till utgången.
  - Vid negativ flank  $\downarrow$  går **C** från 1 till 0 och **C=0** kopplar den undre **q1** vippan till utgången.
- Resultatet blir en **D**-vippa som som reagerar på klockans *båda* flanker.

### 11.5

Det finns åtta kombinationer som kan inträffa för de fyra ingångsvärdeskombinationerna (CD) och de två utgångsvärdeskombinationerna O:

Möjliga input/output kombinationer

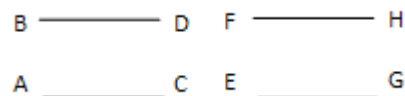
| Present state |     | Next state |        | Kommentar                                 |
|---------------|-----|------------|--------|---|
| State tag     | CDO | (CDO)+     | (CDO)- |   |
| A             | 000 | 010        | 100    | Utgången får D's värde när C ändrar värde |
| B             | 001 | 011        | 100    |   |
| C             | 010 | 000        | 111    | Ingen ändring av O när D ändrar värde     |
| D             | 011 | 001        | 111    |   |
| E             | 100 | 000        | 110    |   |
| F             | 101 | 000        | 111    |   |
| G             | 110 | 011        | 100    |   |
| H             | 111 | 011        | 101    |   |

Flödestabell (stabile states markerade som bold font)

| Present state | Next State (CD) |          |          |          | Output |
|---------------|-----------------|----------|----------|----------|--------|
|               | 00              | 01       | 11       | 10       |        |
| A             | <b>A</b>        | C        | -        | E        | 0      |
| B             | <b>B</b>        | D        | -        | E        | 1      |
| C             | A               | <b>C</b> | H        | -        | 0      |
| D             | B               | <b>D</b> | H        | -        | 1      |
| E             | A               | -        | G        | <b>E</b> | 0      |
| F             | A               | -        | H        | <b>F</b> | 1      |
| G             | -               | D        | <b>G</b> | E        | 0      |
| H             | -               | D        | <b>H</b> | F        | 1      |

Vi ser direkt att ingen minimering kan göras genom att formera ekvivalensklasser eftersom alla åtta tillstånd har olika utgångar där de har stabila tillstånd och där de har don't cares i tabellen.

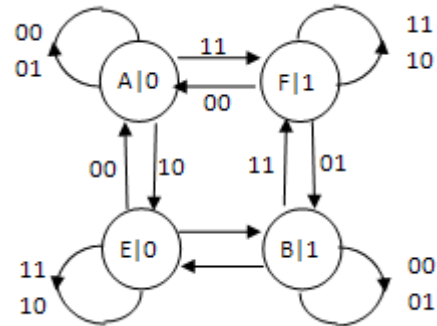
Merger-diagrammet blir som följer:



Den minimerade flödestabellen blir sålunda

Flow Table (stable states marked as bold)

| Present state | Next State (CD) |          |          |          | Output |
|---------------|-----------------|----------|----------|----------|--------|
|               | 00              | 01       | 11       | 10       |        |
| A             | <b>A</b>        | <b>A</b> | F        | E        | 0      |
| B             | <b>B</b>        | <b>B</b> | F        | E        | 1      |
| E             | A               | B        | <b>E</b> | <b>E</b> | 0      |
| F             | A               | B        | <b>F</b> | <b>F</b> | 1      |



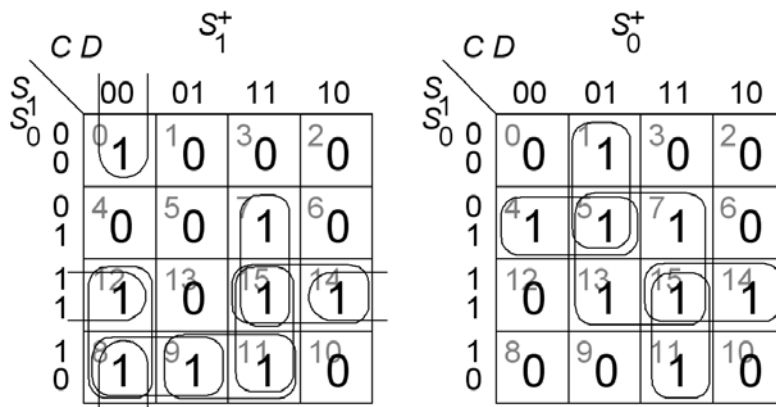
c) Tilldela states, gör Karnaugh-minimering och ta fram booleska tilldelningar. De möjliga state tilldelningarna är (E=00, B=01, A=10, F=11) plus deras rotationer och spegellösningar:

| Possible state assignments |    |    |    |
|----------------------------|----|----|----|
| A                          | F  | B  | E  |
| 00                         | 01 | 11 | 10 |
| 01                         | 11 | 10 | 00 |
| 11                         | 10 | 00 | 01 |
| 10                         | 00 | 01 | 11 |
| 11                         | 01 | 00 | 10 |
| 01                         | 00 | 10 | 11 |
| 00                         | 10 | 11 | 01 |
| 10                         | 11 | 01 | 00 |

Den resulterande flödestabellen blir då

| Present state | Next State (CD) |    |    |    | Output |
|---------------|-----------------|----|----|----|--------|
|               | (CD)            |    |    |    |        |
|               | 00              | 01 | 11 | 10 |        |
| 10            | 10              | 10 | 11 | 00 | 0      |
| 01            | 01              | 01 | 11 | 00 | 1      |
| 00            | 10              | 01 | 00 | 00 | 0      |
| 11            | 10              | 01 | 11 | 11 | 1      |

Och de motsvarande Karnaugh-diagrammen och booleska uttrycken blir:



$$S_1^+ = CD(S_1 + S_0) + \overline{C}\overline{D}(S_1 + \overline{S_0}) + S_1 S_0 (C + \overline{D})$$

$$S_0^+ = S_0 D + \overline{S_1} S_0 \overline{C} + \overline{S_1} \overline{C} D + S_1 C D + S_1 S_0 C$$

$$O = S_0$$

### 11.6

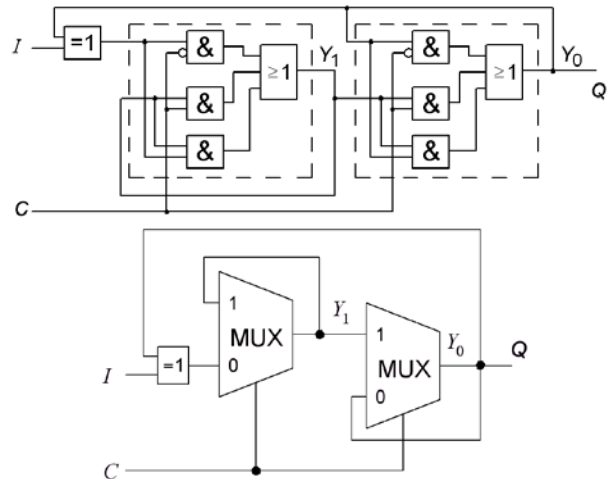
a) Härled de boolska uttrycken för tillståndsvariablerna. Svar:

$$Y_0^+ = Y_0 Y_1 + Y_0 \bar{C} + Y_1 C$$

$$Y_1^+ = Y_1 (Y_0 \oplus I) + (Y_0 \oplus I) \bar{C} + \bar{Y}_1 C$$

b) Härled exitationstabellen. (Ledning! Vilken funktion finns i de två innersta looparna)

De två inre looparna är hazard-fria MUX:ar!



| Pres. State | Next State |               |           |           | Q |
|-------------|------------|---------------|-----------|-----------|---|
|             | IC=        |               |           |           |   |
| $Y_1 Y_0$   | 00         | 01            | 11        | 10        |   |
| 00          | <b>00</b>  | <b>00</b>     | <b>00</b> | 10        | 0 |
| 01          | 11         | <del>00</del> | 00        | <b>01</b> | 1 |
| 11          | <b>11</b>  | <b>11</b>     | <b>11</b> | 01        | 1 |
| 10          | 00         | <del>11</del> | 11        | <b>10</b> | 0 |

Stable states marked in **Bold**

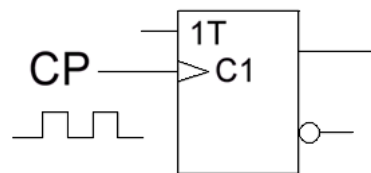
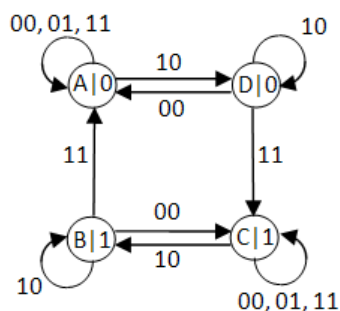
Impossible transitions marked as ~~strikethrough~~

Härled flödestabell, tilldela symboliska states och rita FSM:en.

| Pres. State | Next State |              |          |          | Q |
|-------------|------------|--------------|----------|----------|---|
|             | IC=        |              |          |          |   |
| $Y_1 Y_0$   | 00         | 01           | 11       | 10       |   |
| A           | <b>A</b>   | <b>A</b>     | <b>A</b> | D        | 0 |
| B           | C          | <b>A</b>     | <b>A</b> | <b>B</b> | 1 |
| C           | <b>C</b>   | <b>C</b>     | <b>C</b> | B        | 1 |
| D           | A          | <del>C</del> | C        | <b>D</b> | 0 |

Stable states marked in **Bold**

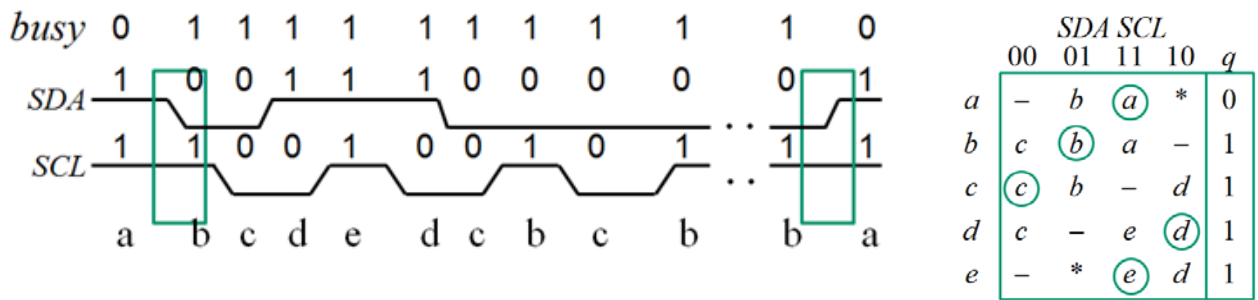
Impossible transitions marked as ~~strikethrough~~



Identifiera funktionen hos den asynkrona kretsen. Vilken vippa motsvarar den?

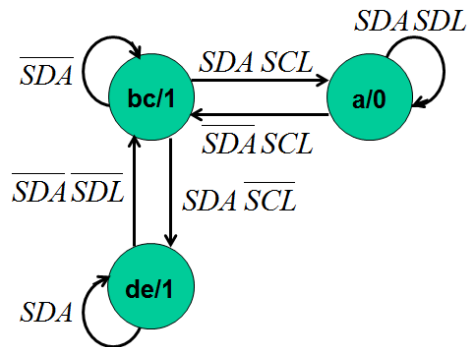
- Positiv flank-triggad T-vippa.

### 11.7



Följ tidsdiagrammet och inför nytt tillstånd vid varje händelse som inte varit med tidigare. I tillstånd *a* ”väntar” vi på startflanken (*b*), då är insignalen 10 omöjlig (markerad med \*). Protokollet förbjuder ändring av data SDA när SCL är **hög**. Därför är insignal 01 omöjlig i tillstånd *e* (markerad med \*). Detta ger två extra don't care positioner i tabellen. Man ser sedan direkt vilka tillstånd som kan slås ihop.

|    | SDA SCL |    |    |    |   |
|----|---------|----|----|----|---|
|    | 00      | 01 | 11 | 10 | q |
| a  | -       | bc | a  | -  | 0 |
| bc | bc      | bc | a  | de | 1 |
| de | bc      | -  | de | de | 1 |



Som tillståndskod kan Gray-kod användas. *a* 00, *bc* 01, *de* 11, och *x* 10. *x* kan användas som don't care bortsett från 10.

| $q_1 q_0$ | SDA SCL   |           |           |           | <i>busy</i> |                     |
|-----------|-----------|-----------|-----------|-----------|-------------|---------------------|
|           | 00        | 01        | 11        | 10        |             |                     |
| 00        | -         | 01        | 00        | -         | 0           |                     |
| 01        | 01        | 01        | 00        | 11        | 1           | <i>busy</i> = $q_0$ |
| 11        | 01        | -         | 11        | 11        | 1           |                     |
| 10        | $\neq 10$ | $\neq 10$ | $\neq 10$ | $\neq 10$ | -           |                     |

$$q_1^+ q_0^+ = SDA(\overline{SCL} + q_1)$$

| $q_1 q_0$ | SDA SCL |    |    |    |
|-----------|---------|----|----|----|
|           | 00      | 01 | 11 | 10 |
| 00        | -       | 0  | 0  | -  |
| 01        | 0       | 0  | 0  | 1  |
| 11        | 0       | -  | 1  | 1  |
| 10        | -       | -  | -  | -  |

$$q_1^+ = SDA(\overline{SCL} + q_1)$$

| $q_1 q_0$ | SDA SCL |    |    |    |
|-----------|---------|----|----|----|
|           | 00      | 01 | 11 | 10 |
| 00        | -       | 1  | 0  | -  |
| 01        | 1       | 1  | 0  | 1  |
| 11        | 1       | -  | 1  | 1  |
| 10        | -       | -  | -  | -  |

$$q_0^+ = (\overline{SDA} + \overline{SCL} + q_1)$$

Näten bildar sammanhängande områden i Karnaughdiagrammen och är därför hazardfria (om näten har två nivåer). Att realisera med valfria grindar.

# Avkodning av minnen och I/O-kretsar

## 12.1

Ett dynamiskt RAM-minne består av ett antal 256Mbit minneskapslar som är organiserade som 32 M×8.

a) Hur många kapslar behövs för 256M×64?

**Minne**  $N = 256M$   $M = 64$  bitar. **Kapsel**  $p = 32M$   $q = 8$  bitar.

Antal kolumner  $k = M/q = 64/8 = 8$ .

Antalet rader  $r = N/p = 256M/32M = 8$ .

Antal kapslar  $K = r \times k = 8 \times 8 = 64$ .

b) Hur många kapslar behövs för 512M×72? (Vad kan anledningen till den "underliga" bitbredden "72" vara?)

**Minne**  $N = 512M$   $M = 72$  bitar. **Kapsel**  $p = 32M$   $q = 8$  bitar.

Antal kolumner  $k = M/q = 72/8 = 9$ .

Antalet rader  $r = N/p = 512M/32M = 16$ .

Antal kapslar  $K = r \times k = 16 \times 9 = 144$ .

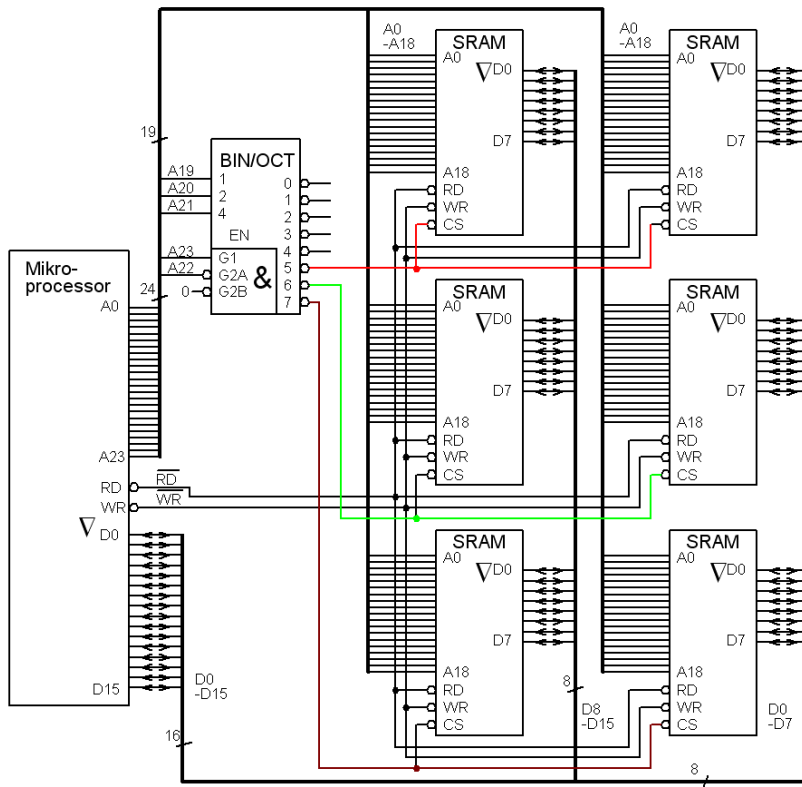
Den "ovanliga" bitbredden 72 (= 64 + 8). De 8 extra bitarna används för att korrigera enkelfel, och för att kunna upptäcka dubbelfel.

## 12.2

En viss 16 bitars processor kan adressera 24 bitar. Minnesrymden fördelas mellan ROM, SRAM och IO-kretsar.

Adressavkodningen sker med hjälp av en 3:8-avkodare.

a) Hur stort är figurens RAM? Vilket är adressområdet uttryckt i hexadecimala siffror?



• **Minneskapsel:**

$p = 512k$   $q = 8$  bitar

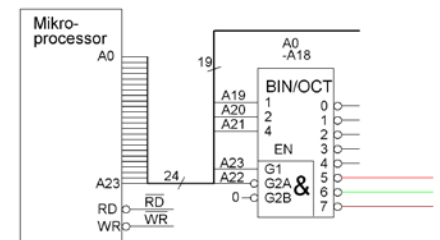
• **Minne:**

$r = 3$   $k = 2$   $K = 2 \times 3 = 6$

$M = k \times q = 2 \times 8 = 16$  bitar

$N = p \times r = 512k \times 3 = 1,5M$

Adressområde:

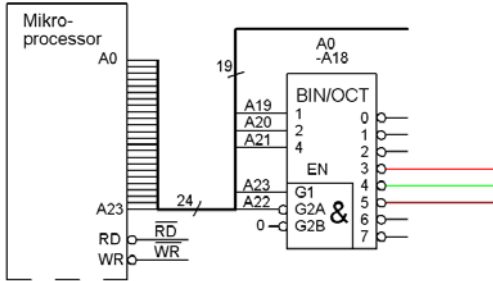


|             |    |    |    |     |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|-------------|----|----|----|-----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| Computer:   | 23 | 22 | 21 | 20  | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Decoder:    | 1  | 0  | 0  | ... | 7  |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Mem start:  | 1  | 0  | 1  | 0   | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| • Begin hex | A  |    |    |     | 8  |    |    |    | 0  |    |    |    | 0  |    |   |   | 0 |   |   |   |   |   |   |   |
| Mem end:    | 1  | 0  | 1  | 1   | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| • End hex   | B  |    |    |     | F  |    |    |    | F  |    |    |    | F  |    |   |   | F |   |   |   |   |   |   |   |

b) Hur ändrar man adressområdet till 980000 – AFFFFFF ?

980000  
 1001|1000|0000|0000|0000|0000|  
 AFFFFFF  
 1010|1111|1111|1111|1111|1111|

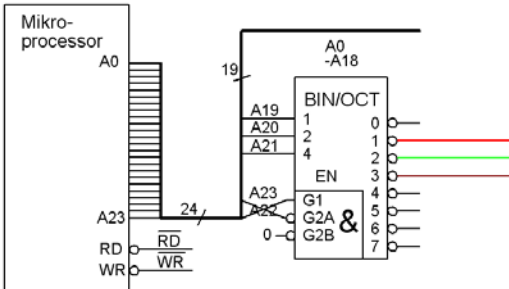
"10|011" → "3"  
 "10|101" → "5"



c) Ändra till adressområde 480000 – 5FFFFFF ?

480000  
 0100|1000|0000|0000|0000|0000|  
 5FFFFFF  
 0101|1111|1111|1111|1111|1111|

"01|001" → "1"  
 "01|011" → "3"



Vi byter plats på A23 och A22 !

d) ROM-minnet är 2M×16 bitar och att adressområdet är 000000 ... och framåt. ROM Chip är 512k×8.

- Hur många kapslar behövs?
- Hur skall avkodaren anslutas?
- Hur skall minneskretsarna anslutas?
- Ange adressområdena för avkodarens utgångar med hexadecimala siffror.

**Minne:**

$N = 2^M$  (4·512k) ordlängden  $M = 16$  bitar

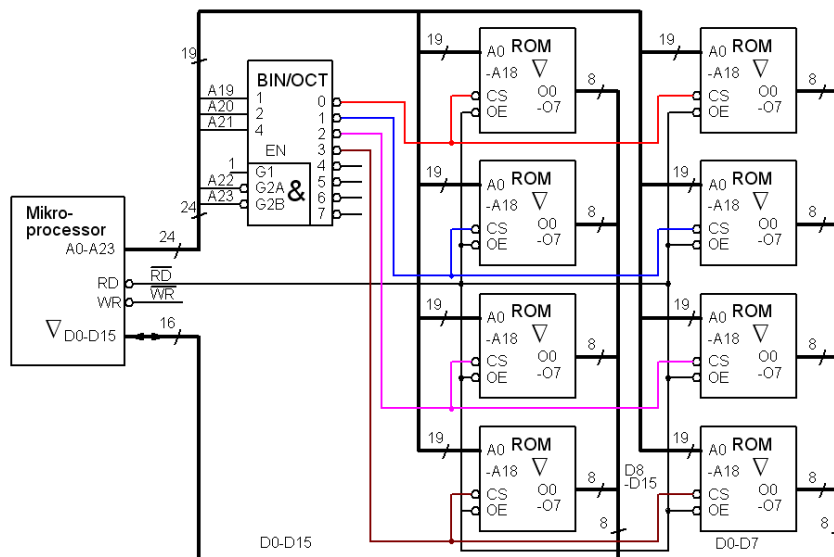
**Minneskapsel:**

$p = 512$  k ordlängden  $q = 8$  bitar

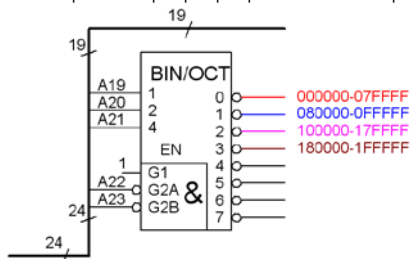
• Antalet kapselrader  $r \leq N/p = 4 \cdot 512k / 512k = 4$

• Antalet kapselkolumner  $k \geq M/q = 16/8 = 2$

• Antalet kapslar  $K = r \times k = 4 \times 2 = 8$



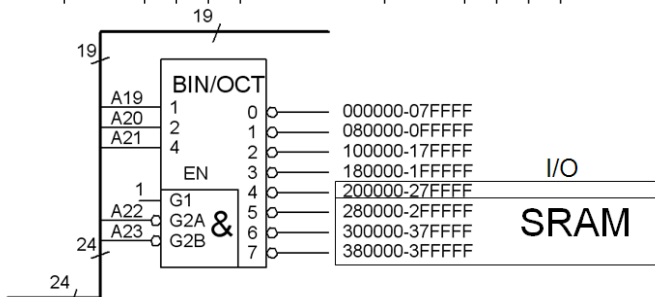
| 00ab | cmmm | mmmm | mmmm | mmmm | mmmm |   |      |      |   |   |   |   |
|------|------|------|------|------|------|---|------|------|---|---|---|---|
| 0000 | 0000 | 0    | 0    | 0    | 0    | - | 0000 | 0111 | F | F | F | F |
| 0000 | 1000 | 0    | 0    | 0    | 0    | - | 0000 | 1111 | F | F | F | F |
| 0001 | 0000 | 0    | 0    | 0    | 0    | - | 0001 | 0111 | F | F | F | F |
| 0001 | 1000 | 0    | 0    | 0    | 0    | - | 0001 | 1111 | F | F | F | F |



Totalt ROM 000000 – 1FFFFFF

e) Vilket adressområde blir ledigt för SRAM och IO-kretsar?

| 00ab | cmmm | mmmm | mmmm | mmmm | mmmm |   |      |      |   |   |   |   |
|------|------|------|------|------|------|---|------|------|---|---|---|---|
| 0010 | 0000 | 0    | 0    | 0    | 0    | - | 0010 | 0111 | F | F | F | F |
| 0010 | 1000 | 0    | 0    | 0    | 0    | - | 0010 | 1111 | F | F | F | F |
| 0011 | 0000 | 0    | 0    | 0    | 0    | - | 0011 | 0111 | F | F | F | F |
| 0011 | 1000 | 0    | 0    | 0    | 0    | - | 0011 | 1111 | F | F | F | F |



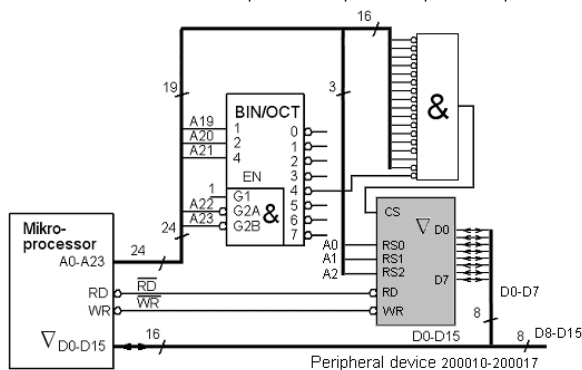
Möjliga SRAM+I/O adresser 200000 – 3FFFFFF

### 12.3

a) Anslut en 8 registers minnesmappad periferienhet (I/O) till en CPU. CPU:n har 16 bitars databuss (vi använder bara 8), och en 24 bitars adressbuss. Använd en 3:8-avkodare och vid behov grindar. Periferienheten skall kopplas in så att den får adresserna 0x200010 ... 0x200017. (Jämför med föregående uppgift).

|          |   |      |  |       |  |      |  |      |  |      |  |       |
|----------|---|------|--|-------|--|------|--|------|--|------|--|-------|
| 0x200010 | = | 0010 |  | 0.000 |  | 0000 |  | 0000 |  | 0001 |  | 0.000 |
| 0x200011 | = | 0010 |  | 0.000 |  | 0000 |  | 0000 |  | 0001 |  | 0.001 |
| 0x200012 | = | 0010 |  | 0.000 |  | 0000 |  | 0000 |  | 0001 |  | 0.010 |
| 0x200013 | = | 0010 |  | 0.000 |  | 0000 |  | 0000 |  | 0001 |  | 0.011 |
| 0x200014 | = | 0010 |  | 0.000 |  | 0000 |  | 0000 |  | 0001 |  | 0.100 |
| 0x200015 | = | 0010 |  | 0.000 |  | 0000 |  | 0000 |  | 0001 |  | 0.101 |
| 0x200016 | = | 0010 |  | 0.000 |  | 0000 |  | 0000 |  | 0001 |  | 0.110 |
| 0x200017 | = | 0010 |  | 0.000 |  | 0000 |  | 0000 |  | 0001 |  | 0.111 |

I/O adresser, 200000 – 27FFFF finns enligt tidigare uppgift på 3:8-avkodarens utgång "4". Den avkodar A23... A19, periferienheten avkodar själv A2 ... A0, resten får avkodas med en och-grind.



b) Vad är ofullständig avkodning?

För fullständig avkodning använde vi en &-grind med 17 ingångar! I bland gör man en **ofullständig avkodning**. Man struntar då i att ta med alla adress-signalerna och kan därmed använda en grind med färre ingångar. I/O-enhetens adressering blir mångtydig, den kan adresseras med många olika adresser, men den som skriver programkoden bestämmer ju själv vilka adresser det är som används. Huvudsaken är att man ser till att I/O-enhetens adresser *inte kolliderar* med någon annan enhets adresser.

