Qarqet Kompjuterike Kombinuese

Book · September 2003

1 author:

Agni Dika
University of Prishtina

Universiteti i Prishtinës
Fakulteti Elektroteknik

Dr. Agni H. DIKA

Qarqet Kompjuterike
Kombinuese
1

Parathënie

Libri Qarqet Kompjuterike u dedikohet studentëve të Fakultetit Elektroteknik të Universitetit të Prishtinës, si dhe studentëve të Fakultetit të Shkencave dhe Teknologjive të Komunikimit në Universitetin e Europës Juglindore në Tetovë. Por, libri njëkohësisht mund t'u shërbejë edhe atyre që punojnë në lëmin e kompjuterikës, ose duan të mësojnë më tepër për analizën dhe sintezën e qarqeve të ndryshme kompjuterike.

Gjatë zhvillimit të njësive mësimore që përfshihen në libër, me qëllim të përvehtësimit sa më të lehtë të materies, janë marrë shembuj të shumtë praktikë, duke e reduktuar në minimum pjesën tekstuale.

Mbështetur në materien që ka të bëjë me qarqet kompjuterike, për arsye teknike, libri është ndarë në dy pjesë. Në pjesën e parë, të cilën e keni në dorë, përfshihen *qarqet kompjuterike kombinuese*. Kurse pjesa e dytë e librit ka të bëjë me analizën dhe sintezën e *qarqeve sekuenciale sinkrone dhe asinkrone*. Kjo pjesë e librit edhepse ishte në fazën përfundimtare për shtypje, gjatë luftës në Kosovë, u zhduk bashkë me kompjuterët në të cilët ruhej materiali përkatës.

Për kontakt me autorin, lidhur me materialin e përfshirë në libër, lexuesi mund ta shfrytëzoi adresën elektronike **agnidika@yahoo.com**.

Autori

Përmbajtja

Sistemet numerike

1

Sistemet numerike paraqesin grumbuj të rregulluar simbolesh (shifrash), mbi të cilët janë definuar katër operacione elementare: *mbledhja* (**+**), *zbritja* (**-**), *shumëzimi* (**·**) dhe *pjesëtimi* (**/**).

Numri i shifrave të ndryshme të cilat përdoren gjatë shkruarjes së numrave në një sistem numerik, e paraqet *bazën e sitemit numerik*. Kështu, baza e sistemit decimal të numrave është **10**, sepse numrat në këtë sistem numerik shkruhen duke shfrytëzuar **10** shifra të ndryshme:

0 1 2 3 4 5 6 7 8 9

Çdo numër **X.Y** në sistemin numerik me bazë **B** mund të shkruhet si numër decimal **N**, përmes kompleksionit me **(m+n)** elemente, kështu:

$$N = \sum_{i=1}^{m} x_i \cdot B^{m-i} + \sum_{j=1}^{n} y_j \cdot B^{-j} \qquad (1.1)$$

ku janë:

$$X = x_1 x_2 ... x_m$$
$$Y = y_1 y_2 ... y_n$$
$$x_i, y_j = \{0, 1, ..., B-1\}$$

Shprehja **(1.1)** vlen vetëm për *sistemet numerike me peshë*, te të cilët çdo pozitë e shifrave brenda numrit ka një peshë të caktuar. Të tillë janë: *sistemi decimal*, *sistemi binar*, *sistemi oktal* ose *sistemi heksadecimal* i numrave, të cilët do të përmenden në vijim. Kështu, p.sh., numri 235 në sistemin decimal të numrave lexohet "dyqind e tridhjet e pesë" sepse me të nënkuptohet vlera:

$$2 \cdot 10^2 + 3 \cdot 10^1 + 5 \cdot 10^0$$

përkatësisht:

$$2 \cdot 100 + 3 \cdot 10 + 5 \cdot 1$$

ose:

$$200 + 30 + 5$$

Sistemi numerik romak është *sistem numerik pa peshë* meqë pozitat e shifrave brenda numrave nuk kanë një peshë të caktuar.

Shembull	Paraqitja e numrave të sistemit decimal të numrave: *a.* **376** *b.* **0.4957** *c.* **3486.52** përmes kompleksioneve përkatëse, në po atë sistem numerik, duke pasur parasysh shprehjen **(1.1)**.

a.

$$\begin{aligned} N &= \sum_{i=1}^{3} x_i \cdot 10^{3-i} \\ &= 3\cdot 10^2 + 7\cdot 10^1 + 6\cdot 10^0 \end{aligned}$$

b.

$$\begin{aligned} N &= \sum_{j=1}^{4} y_j \cdot 10^{-j} \\ &= 4\cdot 10^{-1} + 9\cdot 10^{-2} + 5\cdot 10^{-3} + 7\cdot 10^{-4} \end{aligned}$$

c.

$$\begin{aligned} N &= \sum_{i=1}^{4} x_i \cdot 10^{4-i} + \sum_{j=1}^{2} y_j \cdot 10^{-j} \\ &= 3\cdot 10^3 + 4\cdot 10^2 + 8\cdot 10^1 + 6\cdot 10^0 + 5\cdot 10^{-1} + 2\cdot 10^{-2} \end{aligned}$$

Numri i kompleksioneve të ndryshme në sistemin numerik me bazë **B**, të cilët mund të formohen me **(m+n)** shifra, është:

$$K = B^{m+n} \tag{1.2}$$

kurse numri më i madh i mundshëm me **(m+n)** shifra është:

$$R = B^m - \frac{1}{B^n} \tag{1.3}$$

ku janë:

m - numri i shifrave para pikës dhjetore.
n - numri i shifrave pas pikës dhjetore.

Shembull	Numri i kompleksioneve të ndryshme **K** dhe numri maksimal **R** i cili mund të shkruhet me **(m+n)** shifra, në sistemin decimal të

numrave, nëse:

a. **m=4** **n=2**
b. **m=5** **n=0**
c. **m=0** **n=3**

a.

$$K = 10^{4+2} = 1000000$$

$$R = 10^{4} - \frac{1}{10^{2}} = 9999.99$$

b.

$$K = 10^{5+0} = 100000$$

$$R = 10^{5} - \frac{1}{10^{0}} = 99999$$

c.

$$K = 10^{0+3} = 1000$$

$$R = 10^{0} - \frac{1}{10^{3}} = 0.999$$

Sistemi binar i numrave

Sistemi numerik tek i cili numrat shkruhen duke përdorur vetëm shifrat **0** dhe **1** quhet *sistem binar i numrave*, prandaj edhe baza e këtij sistemi numerik është **B=2**.

Shndërrimi decimal - binar

Gjatë kalimit prej sistemit decimal në sistemin binar të numrave mund të paraqiten katër raste karakteristike, të cilat jepen në vijim.

Numrat e plotë

Ekuivalenti binar i një numri decimal të plotë fitohet *duke pjesëtuar numrin suksesivisht me* **2**, sa është baza **B** e këtij sistemi numerik. Gjatë çdo pjesëtimi, mbetja përshkruhet në një kolonë, kurse pjesëtimi vazhdon derisa numri që pjesëtohet nuk bëhet zero. Nëse vargu i shifrave binare, i cili fitohet si rezultat i mbetjeve gjatë pjesëtimit suksesiv të numrit decimal, përshkruhet nga fundi, paraqet numrin binar të kërkuar.

Shembull

Ekuivalentët binarë të numrave decimalë:

a. **79**
b. **355**
c. **4394**

a.

```
79:2=39  mbetet 1
39:2=19  mbetet 1
19:2=9   mbetet 1
 9:2=4   mbetet 1
 4:2=2   mbetet 0
 2:2=1   mbetet 0
 1:2=0   mbetet 1
```

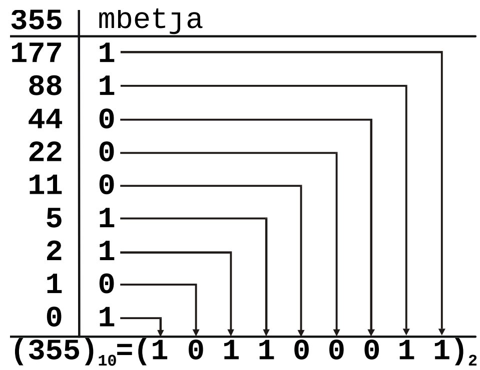
$(79)_{10}=(1001111)_2$

Ky shndërrim më shkurtë mund të paraqitet kështu:

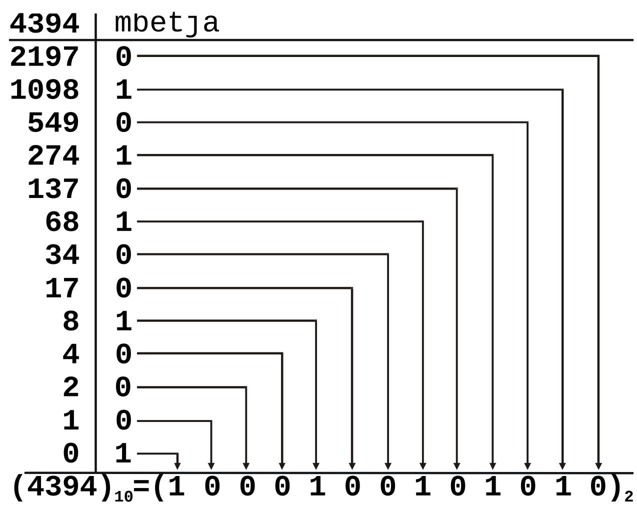
```
79 | mbetja
39 | 1
19 | 1
 9 | 1
 4 | 1
 2 | 0
 1 | 0
 0 | 1
```

$(79)_{10}=(1\ 0\ 0\ 1\ 1\ 1\ 1)_2$

b.



c.



Numrat vetëm me pjesën pas pikës dhjetore

Shndërrimi i numrave decimalë, të cilët e kanë vetëm pjesën pas pikës dhjetore, në numra të sistemit binar bëhet *duke shumëzuar numrin suksesivisht me* **2**. Gjatë çdo shumëzimi, shifra para pikës dhjetore (përfshirë edhe shifrën **0**) përshkruhet në një kolonë të veçantë si tepricë, kurse pjesa pas pikës dhjetore shumëzohet përsëri me **2**. Procesi i shumëzimit vazhdon derisa pjesa pas pikës dhjetore nuk bëhet zero. Në fund, vargu i shifrave binare që fitohet duke e përshkruar prej lart kolonën e tepricave, pasi para saj të shtohet shifra zero me pikë, paraqet ekuivalentin binar të numrit decimal të dhënë.

Shembull Ekuivalentët binarë të numrave decimalë:

a. **0.859375**
b. **0.8125**
c. **0.9609375**

a.

$0.859375 \cdot 2=1.718750=0.718750$	tepron	1
$0.718750 \cdot 2=1.437500=0.437500$	tepron	1
$0.437500 \cdot 2=0.875000$	tepron	0
$0.875000 \cdot 2=1.750000=0.750000$	tepron	1
$0.750000 \cdot 2=1.500000=0.500000$	tepron	1
$0.500000 \cdot 2=1.000000=0.000000$	tepron	1

$$(0.859375)_{10}=(0.1\ 1\ 0\ 1\ 1\ 1)_2$$

b.

$0.8125 \cdot 2=1.6250=0.6250$	tepron	1
$0.6250 \cdot 2=1.2500=0.2500$	tepron	1
$0.2500 \cdot 2=0.5000$	tepron	0
$0.5000 \cdot 2=1.0000=0.0000$	tepron	1

$$(0.8125)_{10}=(0.1\ 1\ 0\ 1)_2$$

c.

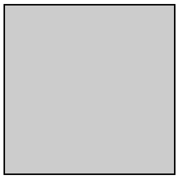
$0.9609375 \cdot 2=1.9218750=0.9218750$	tepron	1
$0.9218750 \cdot 2=1.8437500=0.8437500$	tepron	1
$0.8437500 \cdot 2=1.6875000=0.6875000$	tepron	1
$0.6875000 \cdot 2=1.3750000=0.3750000$	tepron	1
$0.3750000 \cdot 2=0.7500000$	tepron	0
$0.7500000 \cdot 2=1.5000000=0.5000000$	tepron	1
$0.5000000 \cdot 2=1.0000000=0.0000000$	tepron	1

$$(0.9609375)_{10}=(0.1\ 1\ 1\ 1\ 0\ 1\ 1)_2$$

Numrat me pjesën para dhe pas pikës dhjetore

Shndërrimi i numrave decimalë në numra të sistemit binar, të cilët e kanë pjesën para dhe pas pikës dhjetore, bëhet *duke gjetur veç ekuivalentët binarë për pjesën para dhe veç për pjesën pas pikës dhjetore*. Në fund, me bashkimin e dy pjesëve të fituara në një numër të vetëm, duke përshkruar pjesën e gjetur para dhe pjesën pas pikës - përkatësisht, fitohet numri binar i kërkuar.

Shembull Ekuivalentët binarë të numrave decimalë:



a. **125.875**
b. **500.4375**
c. **6.9375**

a.

125	mbetja
62	1
31	0
15	1
7	1
3	1
1	1
0	1

$(125)_{10}=(1\ 1\ 1\ 1\ 1\ 0\ 1)_2$

$0.875 \cdot 2=1.750=0.750$ tepron 1
$0.750 \cdot 2=1.500=0.500$ tepron 1
$0.500 \cdot 2=1.000=0.000$ tepron 1

$(0.875)_{10}=(0.1\ 1\ 1)_2$

$$(125.875)_{10}=(1111101.111)_2$$

b.

500	mbetja
250	0
125	0
62	1
31	0
15	1
7	1
3	1
1	1
0	1

$(500)_{10}=(1\ 1\ 1\ 1\ 1\ 0\ 1\ 0\ 0)_2$

```
0.4375 · 2=0.8750          tepron  0
0.8750 · 2=1.7500=0.7500   tepron  1
0.7500 · 2=1.5000=0.5000   tepron  1
0.5000 · 2=1.0000=0.0000   tepron  1
```

$(0.4375)_{10}=(0.0\ 1\ 1\ 1)_2$

$$(500.4375)_{10}=(111110100.0111)_2$$

c.

6	mbetja
3	0
1	1
0	1

$(6)_{10}=(1\ 1\ 0)_2$

```
0.9375 · 2=1.8750=0.8750   tepron  1
0.8750 · 2=1.7500=0.7500   tepron  1
0.7500 · 2=1.5000=0.5000   tepron  1
0.5000 · 2=1.0000=0.0000   tepron  1
```

$(0.9375)_{10}=(0.1\ 1\ 1\ 1)_2$

$$(6.9375)_{10}=(110.1111)_2$$

Shndërrimi i përafërt

Gjatë gjetjes së ekuivalentit binar të pjesës së numrit pas pikës dhjetore mund të ndodhë të fitohet varg i pafund shifrash binare. Në këto raste, ekuivalenti binar do të jetë i përafërt. Procesi i shndërrimit ndërpritet në një numër të caktuar shifrash pas pikës, gjë që varet nga saktësia e përcaktuar që më parë.

Shembull

Ekuivalentët binarë të numrave decimalë:

a. **0.05**
b. **31.6**
c. **924.358**

duke marrë pas pikës vetëm **5** shifra binare.

a.

$0.05 \cdot 2 = 0.10$ tepron 0
$0.10 \cdot 2 = 0.20$ tepron 0
$0.20 \cdot 2 = 0.40$ tepron 0
$0.40 \cdot 2 = 0.80$ tepron 0
$0.80 \cdot 2 = 1.60 = 0.60$ tepron 1
....

$(0.05)_{10} = (0.0\ 0\ 0\ 0\ 1)_2$

b.

31	mbetja
15	1
7	1
3	1
1	1
0	1

$(31)_{10} = (1\ 1\ 1\ 1\ 1)_2$

$0.6 \cdot 2 = 1.2 = 0.2$ tepron 1
$0.2 \cdot 2 = 0.4$ tepron 0
$0.4 \cdot 2 = 0.8$ tepron 0
$0.8 \cdot 2 = 1.6 = 0.6$ tepron 1
$0.6 \cdot 2 = 1.2 = 0.2$ tepron 1
....

$(0.6)_{10} = (0.1\ 0\ 0\ 1\ 1)_2$

$$(31.6)_{10} = (11111.10011)_2$$

c.

924	mbetja
462	0
231	0
115	1
57	1
28	1
14	0
7	0
3	1
1	1
0	1

$(924)_{10} = (1\ 1\ 1\ 0\ 0\ 1\ 1\ 1\ 0\ 0)_2$

```
0.358 · 2=0.716           tepron 0
0.716 · 2=1.432=0.432     tepron 1
0.432 · 2=0.864           tepron 0
0.864 · 2=1.728=0.728     tepron 1
0.728 · 2=1.456=0.456     tepron 1
...............................
```

$$(0.358)_{10}=(0.0\ 1\ 0\ 1\ 1)_2$$

$$(924.358)_{10} = (1110011100.01011)_2$$

Për llogaritjen e numrit të kompleksioneve **K** dhe të numrit maksimal të mundshëm **R** edhe te sitemi binar i numrave përdoren shprehjet **(1.2)** dhe **(1.3)**, por, për dallim nga sistemi decimal i numrave, këtu baza është **B=2**.

Shembull

Numri i kompleksioneve të ndryshme **K** dhe vlera maksimale e mundshme **R**, për numrat binarë me **m**-shifra para pikës dhe **n**-shifra pas pikës, nëse:

a. **m=4 n=0**
b. **m=0 n=3**
c. **m=5 n=2**

a.

$$K = 2^{4+0} = 16$$

Kompleksionet, përkatësisht numrat e ndryshëm të mundshëm, janë:

```
0000
0001
0010
....
1110
1111
```

$$R = 2^4 - \frac{1}{2^0} = 15$$

përkatësisht $(1111)_2$.

b.

$$K = 2^{0+3} = 8$$

Kompleksionet përkatëse:

```
0.000
0.001
0.010
.....
0.110
0.111
```

$$R = 2^0 - \frac{1}{2^3} = 0.875$$, përkatësisht $(0.111)_2$

c.

$$K = 2^{5+2} = 128$$

Kompleksionet përkatëse:

```
   0.00
   0.01
   0.10
........
11111.01
11111.10
11111.11
```

$$R = 2^5 - \frac{1}{2^2} = 31.75$$, përkatësisht $(11111.11)_2$

Shndërrimi binar - decimal

Për gjetjen e ekuivalentëve decimalë të numrave binarë mund të përdoret shprehja **(1.1)**, gjatë së cilës baza e sitemit numerik duhet të merret **B=2**.

Shembull

Shndërrimi i numrave binarë:

a. **11011**
b. **0.111011**
c. **1110.11**

në numra të sistemit decimal të numrave.

a.

$$X=11011$$

$$N=\sum_{i=1}^{5} x_i \cdot 2^{5-i}$$

$$=1\cdot 2^4+1\cdot 2^3+0\cdot 2^2+1\cdot 2^1+1\cdot 2^0=27$$

b.

$$Y=1\ 1\ 1\ 0\ 1\ 1$$

$$N=\sum_{j=1}^{6} Y_j \cdot 2^{-j}$$

$$=1\cdot 2^{-1}+1\cdot 2^{-2}+1\cdot 2^{-3}+0\cdot 2^{-4}+1\cdot 2^{-5}+1\cdot 2^{-6}$$

$$=0.921875$$

c.

$$X=1\ 1\ 1\ 0$$

$$Y=1\ 1$$

$$N=\sum_{i=1}^{4} x_i \cdot 2^{4-i}+\sum_{j=1}^{2} y_j \cdot 2^{-j}$$

$$=1\cdot 2^3+1\cdot 2^2+1\cdot 2^1+0\cdot 2^0+1\cdot 2^{-1}+1\cdot 2^{-2}$$

$$=14.75$$

Aritmetika binare

Sikurse në sitemin decimal, edhe në sistemin binar të numrave mund të kryhen katër operacionet elementare aritmetikore, duke i përdorur rregullat që janë shpjeguar në vijim.

Mbledhja

Tabela e mbledhjes së numrave në sistemin binar të numrave duket kështu:

0+0=0
0+1=1
1+0=1
1+1=0 dhe bartja **1**

Shembull Mbledhja e numrave binarë:

a.

```
  1110000
+ 111011
```

b.

```
  111011
+ 110111
```

c.

```
 1011.110
+1010.101
```

d.

```
 10110101
 10001101
+11000111
```

Me qëllim që të shihet më mirë procedura e bartjes, në pjesën vijuese do të përdoret shifra decimale **2**, në vend të "dhjetëshes" binare **10**, sa është edhe ekuivalenti decimal përkatës.

a.

```
111      Bartja
 1110000
+ 111011
10101011
```

b.

```
111111  Bartja
 111011
+110111
1110010
```

c.

```
1 111     Bartja
 1011.110
+1010.101
10110.011
```

d.

```
 21111211  Bartja
  10110101
  10001101
 +11000111
1000001001
```

Në shembullin *d* mbledhja e numrave të dhënë mund të kryhet edhe duke i gjetur shumat parciale, gjatë së cilës fillimisht mblidhen dy numrat e parë dhe pastaj shumës i shtohet edhe numri i tretë:

```
1 1111 1 Bartja
 10110101
+10001101
101000010
```

```
111   11  Bartja
 101000010
+ 11000111
1000001001
```

Zbritja

Gjatë procesit të zbritjes së numrave binarë, si edhe te sistemi decimal i numrave, paraqitet nevoja e huazimit. Tabela sipas së cilës kryhet zbritja në sistemin binar të numrave është:

0-0=0
0-1=1 dhe **1** hua
1-0=1
1-1=0

Shembull

Zbritja e numrave binarë:

a.

```
 1110000
-111011
```

b.

```
 1111011
-101111
```

c.

```
  10101011.10
- 1110000.11
```

Me qëllim që të shihet më mirë procesi i huazimit, fillimisht është treguar në tërësi huazimi i nevojshëm dhe pastaj në fund kryhet zbritja e numrave. Edhe këtu, për ta ndjekur më qartë procedurën e huazimit, është përdorur shifra decimale **2**, në vend të "*dhjetëshes*" së sistemit binar.

a.

```
    2222          2          2      Huaja
   01112         02         02      Mbetja
 1110000      1101112    1021112
- 111011     - 111011   - 111011

                           221112
                          -111011
                           110101
```

b.

```
     2          2      Huaja
    02         02      Mbetja
 1111011    1110211
-101111    -101111
 -------    -------

            1102211
           -101111
            -------
            1001100
```

c.

```
         2           2        2             22         Huaja
        02         0 2       02            012         Mbetja
10101011.10  10101011.02  10101010.22  10021010.22
-1110000.11  -1110000.11  -1110000.11  -1110000.11
-----------  -----------  -----------  -----------

                                      01221010.22
                                      -1110000.11
                                      -----------
                                      00111010.11
```

Nëse kërkohet zbritja e numrit binar **B** nga numri binar **A**, kur **A<B**, mund të veprohet si edhe te sistemi decimal i numrave, përkatësisht zbritja të kryhet kështu:

A-B=-(B-A)

Shembull	Zbritja e numrave binarë:

a.

```
   101
– 11010
 -------
```

b.

```
  1001
– 1110
 ------
```

c.

```
  1101.10
– 1110.11
 ---------
```

a.

```
                 2       2
                02      02
   101       11010   11002
-11010       - 101   - 101

                     10202      101
                     - 101   -11010
                     10101   -10101
```

b.

```
                2
               02
  1001       1110      1102      1001
 -1110      -1001     -1001     -1110
                       0101     -0101
```

c.

```
                2
               02
 1101.10     1110.11      1102.11      1101.10
-1110.11    -1101.10     -1101.10     -1110.11
                          0001.01     -0001.01
```

Shumëzimi

Operacioni i shumëzimit kryhet plotësisht njëlloj si edhe në sistemin binar të numrave, duke pasur parasysh tabelën vijuese:

$$\mathbf{0 \cdot 0 = 0}$$
$$\mathbf{0 \cdot 1 = 0}$$
$$\mathbf{1 \cdot 0 = 0}$$
$$\mathbf{1 \cdot 1 = 1}$$

Shembull

Shumëzimi i numrave binarë:

a. *b.* *c.*

```
   1111     10101011.11     0.1101
 x 1101     x     0.011     x 0.11
```

a.

```
    1111
  x 1101
    1111
   0000
  1111
 1111
11000011
```

b.

```
   10101011.11
   x     0.011
    1010101111
   1010101111
  0000000000
 0000000000
01000000.01101
```

c.

```
  0.1101
   x 0.11
   01101
  01101
 00000
0.100111
```

Problemi i vetëm që paraqitet gjatë procesit të shumëzimit të dy numrave është ai i mbledhjes së më shum numrave binarë, gjatë së cilës bartja është numër binar disashifrorë. Ky problem tejkalohet, nëse zbatohet parimi i dhënë më sipër për mbledhje parciale, sipas të cilit së pari mblidhen dy numrat e parë, pastaj shumës i shtohet numri i tretë dhe kështu me radhë derisa të mblidhen të gjithë numrat tjerë.

Pjesëtimi

Gjatë pjesëtimit të dy numrave binarë, më lehtë është që pjesa e shifrave që pjesëtohen dhe pjesëtuesi të konvertohen në numra të sistemit decimal. Tabela e pjesëtimit që zbatohet në sistemin binar të numrave duket kështu:

```
0:0=?
0:1=0
1:0=?
1:1=1
```

ku me **?** janë shënuar rezultatet kur pjesëtimi me zero është i palogjikshëm.

Shembull

Pjesëtimi i numrave binarë:

a. **1111:11**

b. **10100.1:10**

c. **10000:11**

a.

b.

c.

```
1111:11=101
11
==1
  0
  11
  11
  ==
```

```
10100.1:10=1010.01
10
==1
  0
  10
  10
  ==0
    0
    =1
     0
     10
     10
     ==
```

```
10000:11=101.01...
 11
==10
  00
  100
   11
  ==10
    00
    100
     11
    ==1
```

Siç shihet edhe nga shembujt e mësipërm, pjesëtimi direkt i numrave binarë nuk është i thjeshtë. Në praktikë, numrat binarë pjesëtohen edhe *përmes zbritjes së përsëritur të pjesëtuesit nga i pjesëtuari*, duke filluar prej shifrave me peshë më të madhe. Nëse zbritja është e mundshme, në një kolonë të veçantë shënohet shifra binare **1**, përndryshe shënohet shifra binare **0** pa e kryer zbritjen. Zbritja vijuese kryhet duke zhvendosur pjesëtuesin që zbritet për një pozicion djathtas.

Shembull	Pjesëtimi i numrave binarë: *a.* **1111:11** *b.* **1100111:101** *c.* **10100.1:10** duke e përdorur metodën e pjesëtimit përmes zbritjes sukcesive.

a.

b.

```
1111      1
11
----
0011      0
 11
----
0011      1
  11
----
0000
1111 : 11 = 1 0 1
```

```
1100111       1
101
-------
0010111       0
 101
-------
0010111       1
  101
-------
0000011       0
   101
-------
0000011       0
    101
-------
00000110      1
     101
--------
00000001  mbetja
1100111 : 101 = 1 0 1 0 0.1...
```

c.

```
10100.1     1
10
-------
00100.1     0
 10
-------
00100.1     1
  10
-------
00000.1     0
   10
-------
00000.1     0
    1 0
-------
00000.1     1
      10
--------
00000.00
10100.1 : 10 =  1 0 1 0.0 1
```

Gjatë kryerjes së operacioneve të ndryshme aritmetikore, mund të bëhet edhe prova për vërtetimin e rezultatit të fituar, ngjashëm me atë se si veprohet edhe në sistemin decimal të numrave. Por, prova këtu mund të bëhet edhe duke i shndërruar në numra të sitemit decimal numrat që marrin pjesë në operacion si dhe rezultatin që fitohet.

Sistemi heksadecimal i numrave

Në sistemin heksadecimal, numrat shkruhen duke përdorur **16** shifra të ndryshme:

0 1 2 3 4 5 6 7 8 9 A B C D E F

ku, në vend të numrave:

10 11 12 13 14 15

janë përdorur (përkatësisht) shkronjat:

A B C D E F

Meqë në sistemin numerik heksadecimal shfrytëzohen **16** shifra të ndryshme, baza e këtij sistemi numerik është **B=16**.

Shndërrimi decimal - heksadecimal

Në mënyrë analoge me sistemin binar të numrave, gjatë shndërrimit të numrave të sistemit decimal në numra të sistemit heksadecimal, paraqiten katër raste karakteristike. Rruga që ndiqet në këto katër raste është plotësisht e njëjtë me atë që u dha gjatë shndërrimit në sistemin binar të numrave, por te sistemi heksadecimal ndryshon vetëm baza.

Shembull

Shndërrimi i numrave të sistemit decimal:

a. **927143**
b. **0.37109375**
c. **154.8125**
d. **35.47392**

në numra të sistemit heksadecimal.

a.

927143:16=57946 mbetet **7**
57946:16=3621 mbetet **A**
3621:16=226 mbetet **5**
226:16=14 mbetet **2**
14:16=0 mbetet **E**

$$(927143)_{10}=(E\ 2\ 5\ A\ 7)_{16}$$

b.

0.37109375 · 16=5.93750000=0.93750000 tepron **5**
0.93750000 · 16=15.00000000=0.00000000 tepron **F**

$$(0.93750000)_{10}=(0.5\ F)_{16}$$

c.

154:16=9 mbetet **A**
9:16=0 mbetet **9**

$$(154)_{10}=(9\ A)_{16}$$

0.8125 · 16=13.0000=0.0000 tepron **D**

$$(0.8125)_{10}=(0.D)_{16}$$

$$(154.8125)_{10} = (9A.D)_{16}$$

d.

35:16=2 mbetet **3**
2:16=0 mbetet **2**

$$(35)_{10}=(2\ 3)_{16}$$

0.47392 · 16=7.58272=0.58272 tepron **7**
0.58272 · 16=9.32352=0.32352 tepron **9**
0.32352 · 16=5.17632=0.17632 tepron **5**
0.17632 · 16=2.82112=0.82112 tepron **2**
0.82112 · 16=13.13792=0.13792 tepron **D**
....

$$(0.47392)_{10}=(0.7\ 9\ 5\ 2\ D...)_{16}$$

$$(35.47392)_{10} = (23.7952D...)_{16}$$

Edhe te sistemi heksadecimal i numrave, duke përdorur shprehjet **(1.2)** dhe **(1.3)**, mund të gjendet numri i kompleksioneve të ndryshme **(K)** dhe numri maksimal i mundshëm **(R)**, kur numri ka **m**-shifra para pikës dhe **n**-shifra pas pikës.

Shembull	Gjetja e numrit të kompleksioneve dhe e numrit maksimal të mundshëm, në sistemin heksadecimal të numrave, nëse numri i shifrave para dhe pas pikës është: *a.* **m=3 n=0** *b.* **m=0 n=4** *c.* **m=2 n=1**

a.

$$K = 16^{3+0} = 4096$$

$$N_{max} = 16^{3} - \frac{1}{16^{0}} = 4095 \qquad (N_{max})_{16} = (FFF)_{16}$$

b.

$$K = 16^{0+4} = 65536$$

$$N_{max} = 16^{0} - \frac{1}{16^{4}} = 0.9999847 \qquad (N_{max})_{16} = (0.FFFF)_{16}$$

c.

$$K = 16^{2+1} = 4096$$

$$N_{max} = 16^{2} - \frac{1}{16^{1}} = 255.9375 \qquad (N_{max})_{16} = (FF.F)_{16}$$

Shndërrimi heksadecimal - decimal

Sikurse gjatë shndërrimit të numrave binarë në numra të sistemit decimal, edhe gjatë shndërrimit të numrave heksadecimalë mund të përdoret shprehja **(1.1)**. Por, këtu baza e sistemit numerik duhet të merret **B=16**.

Shembull	Ekuiualentët decimalë të numrave në sistemin heksadecimal: *a.* **1DA5** *b.* **63247** *c.* **0.35** *d.* **3AB.F1**

a.

$$X = 1DA5$$

$$N = \sum_{i=1}^{4} x_i \cdot 16^{4-i}$$

$$= 1 \cdot 16^3 + D \cdot 16^2 + A \cdot 16^1 + 5 \cdot 16^0 = 7589$$

b.

$$X = 63247$$

$$N = \sum_{i=1}^{5} x_i \cdot 16^{5-i}$$

$$= 6 \cdot 16^4 + 3 \cdot 16^3 + 2 \cdot 16^2 + 4 \cdot 16^1 + 7 \cdot 16^0 = 406087$$

c.

$$Y = 35$$

$$N = \sum_{j=1}^{2} y_j \cdot 16^{-j}$$

$$= 3 \cdot 16^{-1} + 5 \cdot 16^{-2} = 0.2070312$$

d.

$$X = 3AB$$

$$Y = F1$$

$$N = \sum_{i=1}^{3} x_i \cdot 16^{3-i} + \sum_{j=1}^{2} y_j \cdot 16^{-j}$$

$$= 3 \cdot 16^2 + A \cdot 16^1 + B \cdot 16^0 + F \cdot 16^{-1} + 1 \cdot 16^{-2}$$

$$= 939.94141$$

Aritmetika heksadecimale

Në sistemin heksadecimal të numrave operacionet aritmetikore kryhen ngjashëm me sistemin decimal, ose me sistemin binar të numrave. Por, dallimi qëndron në atë se si "*dhjetëshe*" këtu merret baza **B=16** e sistemit heksadecimal. Në vazhdim do të shpjegohen operacionet elementare të mbledhjes (**+**) dhe zbritjes (**-**) në këtë sistem numerik, të cilët përdoren edhe në praktikë.

Mbledhja

Meqë në praktikë jemi mësuar me mbledhjen në sistemin decimal të numrave, është më lehtë që mbledhja e shifrave heksadecimale të kryhet duke pasur parasysh ekuivalentët decimalë të tyre.

Shembull

Mbledhja e numrave heksadecimalë:

a.	*b.*	*c.*	*d.*
1DDC + 159	2FD71 + A3542	D2.534 + A.12F	1AB.2F + 345.12

a.

```
  11  Bartja
 1DDC
+ 159
 1F35
```

sepse:

```
                1              1
   C            D              D
  +9           +5             +1
```

$(21)_{10}=(15)_{16}$ $(19)_{10}=(13)_{16}$ $(15)_{10}=(F)_{16}$

b.

```
  11    Bartja
  2FD71
 +A3542
  D32B3
```

sepse:

```
                                  1         1
 1       7       D                F         2
+2      +4      +5               +3        +A
```

$(3)_{10}=(3)_{16}$ $(11)_{10}=(B)_{16}$ $(18)_{10}=(12)_{16}$ $(19)_{10}=(13)_{16}$ $(13)_{10}=(D)_{16}$

c.

```
      1  Bartja
 D2.534
+ A.12F
 DC.663
```

sepse:

```
           1
  4        3         5         2
 +F        2         1        +A
(19)10=(13)16  (6)10=(6)16  (6)10=(6)16  (12)10=(C)16
```

d.

```
  1  1  Bartja
 1AB.2F
+345.12
 4F0.41
```

sepse:

```
            1                          1
 F          2          B               A          1
+2         +1         +5              +4         +3
(17)10=(11)16  (4)10=(4)16  (16)10=(10)16  (15)10=(F)16  (4)10=(4)16
```

Zbritja

Gjatë kryerjes së operacionit të zbritjes paraqitet problemi i huazimit, edhe në sistemin heksadecimal të numrave. Me qëllim që të shpjegohet procedura e huazimit, për "*dhjetëshen*" e sistemit heksadecimal të numrave, përkatësisht për bazën **16**, këtu do të përdoret shkronja **G**.

Shembull

Zbritja e numrave heksadecimalë:

a.

```
 1CD
- 4E
```

b.

```
 53A.2B
-74.1C
```

Sikurse gjatë zbritjes në sistemin binar, edhe këtu së pari është dhënë e tërë procedura e huazimit dhe pastaj është kryer zbritja.

a.

```
        G            Huaja
       BG          G Mbetja
      1CD        1BD
      -4E        -4E
                 ---
                 17F
```

sepse:

$$[(G + D) - E]_{16} = [(16 + 13) - 14]_{10} = (15)_{10} = (F)_{16}$$
$$[B - 4]_{16} = [11 - 4]_{10} = (7)_{10} = (7)_{16}$$

b.

```
       G       G                  Huaja
      1G      4G    G      G    G Mbetja
  53A.2B  53A.1B      43A.1B
  -74.1C  -74.1C      -74.1C
```

sepse:

$$[(G + B) - C]_{16} = [(16 + 11) - 12]_{10} = (15)_{10} = (F)_{16}$$
$$(A - 4)_{16} = (10 - 4)_{10} = (6)_{10} = (6)_{16}$$
$$[(G + 3) - 7]_{16} = [(16 + 3) - 7]_{10} = (12)_{10} = (C)_{16}$$

Me qëllim që të vërtetohet saktësia e rezultateve të fituara, mund të bëhet prova përmes operacionit komplementar, përkatësisht duke zbritur mbledhësin nga rezultati i cili fitohet gjatë operacionit të mbledhjes, ose duke mbledhur rezultatin me zbritësin - gjatë operacionit të zbritjes.

Sistemi oktal i numrave

Për shkruarjen e numrave në sistemin oktal përdoren **8** shifra të ndryshme:

0 1 2 3 4 5 6 7

prandaj thuhet se baza e këtij sistemi numerik është **B=8**.

Shndërrimi decimal - oktal

Rruga që ndiqet gjatë shndërrimit të numrave decimalë në numra të sistemit oktalë është e njëjtë me atë që u dha për sistemin binar dhe heksadecimal, por këtu shumëzohet ose pjesëtohet me bazën **B=8**.

Shembull Ekuivalentët oktalë të numrave në sistemin decimal:

a. **1981**
b. **0.65625**
c. **742.25**
d. **365.24**

a.

1981:8=247 mbetet **5**
247:8=30 mbetet **7**
30:8=3 mbetet **6**
3:8=0 mbetet **3**

$(1981)_{10}=(3\ 6\ 7\ 5)_8$

b.

0.65625 · 8=5.25=0.25 tepron **5**
0.25000 · 8=2.00=0.00 tepron **2**

$(0.65625)_{10}=(0.5\ 2)_8$

c.

742:8=92 mbetet **6**
92:8=11 mbetet **4**
11:8=1 mbetet **3**
1:8=0 mbetet **1**

$(742)_{10}=(1\ 3\ 4\ 6)_8$

0.25 · 8=2.00=0.00 tepron **2**

$(0.25)_{10}=(0.2)_8$

$$(742.25)_{10} = (1346.2)_8$$

d.

```
365:8=45  mbetet  5
 45:8=5   mbetet  5
  5:8=0   mbetet  5
              (365)10=(5 5 5)8

0.24 · 8=1.92=0.92  tepron  1
0.92 · 8=7.36=0.36  tepron  7
0.36 · 8=2.88=0.88  tepron  2
0.88 · 8=7.04=0.04  tepron  7
0.04 · 8=0.32       tepron  0
0.32 · 8=2.56=0.56  tepron  2
............................
                    (0.24)10=(0.1 7 2 7 0 2...)8
```

$$(365.24)_{10} = (555.172702\ldots)_8$$

Shndërrimi oktal - decimal

Sikurse te sistemi binar dhe te sistemi oktal i numrave, këtu shndërrimi mund të bëhet duke përdorur shprehjen **(1.1)**, vetëm se baza e sistemit numerik është **8**.

Shembull	Ekuivalentët decimalë për numrat e sistemit oktal: *a.* **534** *b.* **0.62** *c.* **3724.61**

a.

$$X = 534$$

$$N = \sum_{i=1}^{3} x_i \cdot 8^{3-i}$$

$$= 5 \cdot 8^2 + 3 \cdot 8^1 + 4 \cdot 8^0 = 348$$

b.

$$\mathbf{y} = \mathbf{62}$$

$$\mathbf{N} = \sum_{j=1}^{2} \mathbf{y_j} \cdot \mathbf{8^{-j}}$$

$$= \mathbf{6} \cdot \mathbf{8^{-1}} + \mathbf{2} \cdot \mathbf{8^{-2}} = \mathbf{0.78125}$$

c.

$$\mathbf{X} = \mathbf{3724}$$

$$\mathbf{Y} = \mathbf{61}$$

$$\mathbf{N} = \sum_{i=1}^{4} \mathbf{x_i} \cdot \mathbf{8^{4-i}} + \sum_{j=1}^{2} \mathbf{y_j} \cdot \mathbf{8^{-j}}$$

$$= \mathbf{3} \cdot \mathbf{8^3} + \mathbf{7} \cdot \mathbf{8^2} + \mathbf{2} \cdot \mathbf{8^1} + \mathbf{4} \cdot \mathbf{8^0} + \mathbf{6} \cdot \mathbf{8^{-1}} + \mathbf{1} \cdot \mathbf{8^{-2}}$$

$$= \mathbf{2004.765625}$$

Aritmetika oktale

Operacionet aritmetikore në sistemin oktal të numrave, kryhen plotësisht njëlloj si edhe te sistemet e tjera numerike, gjë që do të tregohet përmes shembujve të mbledhjes së disa numrave. Këtu duhet pasur kujdes në faktin se "dhjetëshja" e sistemit oktal është baza numerike përkatëse **8**.

Shembull

Mbledhja e numrave oktalë:

a.

```
 1737
+423
```

b.

```
 135.6241
+  0.3724
```

c.

```
 624.325
+137.453
```

a.

```
1 1  Bartja
1737
+423
2362
```

sepse:

$$\begin{array}{r} 7\\ +3\\ \hline (10)_{10}=(12)_8 \end{array} \qquad \begin{array}{r} 1\\ 3\\ +2\\ \hline (6)_{10}=(6)_8 \end{array} \qquad \begin{array}{r} 7\\ +4\\ \hline (11)_{10}=(13)_8 \end{array} \qquad \begin{array}{r} 1\\ +1\\ \hline (2)_{10}=(2)_8 \end{array}$$

b.

$$\begin{array}{r} 1\ 1\quad\\ 135.6241\\ +\ 0.3724\\ \hline 136.2165 \end{array} \quad \text{Bartja}$$

sepse:

$$\begin{array}{r} 1\\ 6\\ +3\\ \hline (10)_{10}=(12)_8 \end{array} \qquad \begin{array}{r} 2\\ +7\\ \hline (9)_{10}=(11)_8 \end{array}$$

c.

$$\begin{array}{r} 11\ 11\quad\\ 624.325\\ +137.453\\ \hline 764.000 \end{array} \quad \text{Bartja}$$

$$\begin{array}{r} 5\\ +3\\ \hline (8)_{10}=(10)_8 \end{array} \qquad \begin{array}{r} 1\\ 2\\ +5\\ \hline (8)_{10}=(10)_8 \end{array} \qquad \begin{array}{r} 1\\ 3\\ +4\\ \hline (8)_{10}=(10)_8 \end{array} \qquad \begin{array}{r} 1\\ 4\\ +7\\ \hline (12)_{10}=(14)_8 \end{array}$$

Aritmetika komplementare

Për çdo numër me bazën **B**, mund të gjenden numrat komplementarë përkatës, përkatësisht **B**-komplementi dhe **(B-1)**-komplementi. Pastaj, përmes numrave komplementarë, p.sh., kryhet më lehtë operacioni i zbritjes ose edhe disa operacione logjike.

B-komplementi

Komplementi i një numri **N**, i cili ka **m**-shifra në pjesën e plotë të tij, në sistemin numerik me bazën **B**, shkurt quhet **B**-komplement dhe gjendet përmes shprehjes:

$$N_{\overline{B}} = \begin{cases} B^m - N & \text{për} \quad N \neq 0 \\ 0 & \text{për} \quad N = 0 \end{cases} \qquad (1.4)$$

Nëse pjesa e plotë e numrit është zero, atëherë numri i shifrave merret **m=0**.

Shembull

B-komplementi për numrat e sistemit decimal:

a. **N=82530**
b. **N=0.5287**
c. **N=85.459**

ku **B=10**.

a.

$$N_{\overline{10}} = 10^5 - 82530 = 17470$$

b.

$$N_{\overline{10}} = 10^0 - 0.5287 = 0.4713$$

c.

$$N_{\overline{10}} = 10^2 - 85.459 = 14.541$$

Ngjajshëm veprohet edhe gjatë gjetjes së **B**-komplementit te numrat e sistemit binar, duke pasur kujdes se baza e këtij sistemi numerik është **B=2.**

Shembull

B-komplementi për numrat e sistemit binar:

a. **N=101100**
b. **N=0.0110**
c. **N=110.10101**

a.

$$N_{\bar{2}} = (2^6)_{10} - 101100$$
$$= 1000000 - 101100 = 010100$$

b.

$$N_{\bar{2}} = (2^0)_{10} - 0.0110$$
$$= 1 - 0.0110 = 0.1010$$

c.

$$N_{\bar{2}} = (2^3)_{10} - 110.10101$$
$$= 1000 - 110.10101 = 001.01011$$

Nëse analizohen shembujt e dhënë më sipër, shihet se **B**-komplementi i një numri mund të gjendet edhe në rrugë direkte, duke shfrytëzuar rregullat vijuese:

- zerot në fund të numrit nuk ndryshohen
- shifra e parë nga fundi i numrit që është jozero, zbritet nga baza **B**
- shifrat e tjera zbriten nga vlera **B-1**.

Nëse pjesa e plotë e numrit është zero, duke e pasur parasysh shprehjen **(1.4)**, **B**-komplementi i kësaj pjese të numrit merret zero.

Shembuj

B-komplementi për numrat decimalë:

a. **N=82530**
b. **N=0.5287**
c. **N=85.459**

dhe numrat binar:

d. **N=101100**
e. **N=0.0110**
f. **N=110.10101**

i gjetur në rrugë direkte.

a.

$$N_{\overline{10}} = (9-8)\,(9-2)\,(9-5)\,(10-3)\quad 0$$
$$= \;1\quad 7\quad 4\quad 7\quad 0$$

b.

$$N_{\overline{10}} = 0.(9-5)\,(9-2)\,(9-8)\,(10-7)$$
$$= 0.\;4\quad 7\quad 1\quad 3$$

c.

$$N_{\overline{10}} = (9-8)\,(9-5).(9-4)\,(9-5)\,(10-9)$$
$$= \;1\quad 4\,.\quad 5\quad 4\quad 1$$

d.

$$N_{\overline{2}} = (1-1)\,(1-0)\,(1-1)\,(2-1)\quad 0\quad 0$$
$$= \;0\quad 1\quad 0\quad 1\quad 0\quad 0$$

e.

$$N_{\overline{2}} = 0.(1-0)\,(1-1)\,(2-1)\quad 0$$
$$= 0.\;1\quad 0\quad 1\quad 0$$

f.

$$N_{\overline{2}} = (1-1)(1-1)(1-0).(1-1)\,(1-0)\,(1-1)\,(1-0)\,(2-1)$$
$$= \;0\quad 0\quad 1\,.\quad 0\quad 1\quad 0\quad 1\quad 1$$

(B-1)-komplementi

Për numrin **N** në sistemin numerik me bazën **B**, i cili ka **m**-shifra në pjesën e plotë të tij dhe **n**-shifra në pjesën pas pikës, **(B-1)**-komplementi gjendet përmes shprehjes:

$$N_{\overline{B-1}} = \begin{cases} B^{m} - B^{-n} - N & \text{për} \quad N \neq 0 \\ 0 & \text{për} \quad N = 0 \end{cases} \qquad (1.5)$$

Nëse pjesa e plotë e numrit është zero, numri i shifrave merret **m=0**.

Shembull

(B-1)-komplementi për numrat decimalë:

a. **N=82530**
b. **N=0.5287**
c. **N=85.459**

ku baza **B=10**.

a.

$$N_{\bar{9}} = 10^5 - 10^0 - 82530$$
$$= 100000 - 1 - 82530 = 17469$$

b.

$$N_{\bar{9}} = 10^0 - 10^{-4} - 0.5287$$
$$= 1 - 0.0001 - 0.5287 = 0.4712$$

c.

$$N_{\bar{9}} = 10^2 - 10^{-3} - 85.459$$
$$= 100 - 0.001 - 85.459 = 14.540$$

(B-1)-komplementi edhe për numrat binarë gjendet përmes shprehjes **(1.5)**, duke pasur kujdes që komponentet ku paraqitet fuqizimi të konvertohen në vlera të sistemit binar të numrave.

Shembull **(B-1)-**komplementi për numrat binarë:

a. **N=101100**
b. **N=0.0110**
c. **N=110.10101**

a.

$$N_{\bar{1}} = (2^6)_{10} - (2^0)_{10} - 101100$$
$$= 1000000 - 1 - 101100 = 010011$$

b.

$$N_{\bar{1}} = (2^0)_{10} - (2^{-4})_{10} - 0.0110$$
$$= 1 - 0.0001 - 0.0110 = 0.1001$$

c.

$$N_{\bar{1}} = (2^3)_{10} - (2^{-5})_{10} - 110.10101$$
$$= 1000 - 0.00001 - 110.10101 = 001.01010$$

Nga shembujt e dhënë më sipër, si dhe te **B**-komplementi, shihet se **(B-1)-**komplementi i një numri mund të gjendet në rrugë direkte, duke zbritur çdo shifër të numrit nga vlera **B-1**. **(B-1)-**komplementi i pjesës së plotë të numrit merret zero, nëse vlera e kësaj pjese është zero.

Shembull

(B-1)-komplementi për numrat decimalë:

a. **N=82530**
b. **N=0.5287**
c. **N=85.459**

dhe për numrat binarë:

d. **N=101100**
e. **N=0.0110**
f. **N=110.10101**

i gjetur në rrugë direkte.

a.

$$N_{\bar{9}} = (9-8)(9-2)(9-5)(9-3)(9-0)$$
$$= 1 \quad 7 \quad 4 \quad 6 \quad 9$$

b.

$$N_{\bar{9}} = 0.\,(9-5)(9-2)(9-8)(9-7)$$
$$= 0.\quad 4 \quad 7 \quad 1 \quad 2$$

c.

$$N_{\bar{9}} = (9-8)(9-5).(9-4)(9-5)(9-9)$$
$$= 1 \quad 4 \;.\; 5 \quad 4 \quad 0$$

d.

$$N_{\bar{1}} = (1-1)(1-0)(1-1)(1-1)(1-0)(1-0)$$
$$= 0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1$$

e.

$$N_{\bar{1}} = 0.(1-0)(1-1)(1-1)(1-0)$$
$$= 0.\; 1 \quad 0 \quad 0 \quad 1$$

f.

$$N_{\bar{1}} = (1-1)(1-1)(1-0).(1-1)(1-0)(1-1)(1-0)(1-1)$$
$$= 0 \quad 0 \quad 1 \;.\; 0 \quad 1 \quad 0 \quad 1 \quad 0$$

Raporti mes komplementeve

Me kombinimin e shprehjeve **(1.4)** dhe **(1.5)** fitohet shprehja:

$$N_{\overline{B-1}} = B^m - B^{-n} - N$$

ose

$$N_{\overline{B-1}} = N_{\overline{B}} - B^{-n} \qquad (1.6)$$

prej nga pastaj mund të nxirret raporti:

$$N_{\overline{B}} = N_{\overline{B-1}} + B^{-n} \qquad (1.7)$$

Shembull

Gjetja e $N_{\overline{2}}$, nëse dihet $N_{\overline{1}}$:

a. **01001011**
b. **010011**
c. **0.1001**
d. **101.1101**

a.

$$\begin{aligned} N_{\overline{2}} &= N_{\overline{1}} + (2^0)_{10} \\ &= 01001011 + 1 = 01001100 \end{aligned}$$

b.

$$\begin{aligned} N_{\overline{2}} &= N_{\overline{1}} + (2^0)_{10} \\ &= 010011 + 1 = 010100 \end{aligned}$$

c.

$$\begin{aligned} N_{\overline{2}} &= N_{\overline{1}} + (2^{-4})_{10} \\ &= 0.1001 + 0.0001 = 0.1010 \end{aligned}$$

d.

$$\begin{aligned} N_{\overline{2}} &= N_{\overline{1}} + (2^{-4})_{10} \\ &= 101.1101 + 0.0001 = 101.1110 \end{aligned}$$

Gjetja direkte e 1 dhe 2-komplementit

Në rrugë më të shkurtër, **1**-komplementi i numrit binar gjendet duke i zëvendësuar shifrat **1** me **0** dhe shifrat **0** me **1**. Nëse gjatë këtij veprimi vlera e pjesës së plotë të numrit *është zero*, atëherë **1**-komplementi përkatës i kësaj pjese *merret zero*.

Shembull

1-komplementi për numrat binar:

a. **N=101100**
b. **N=0.0110**
c. **N=110.10101**

a.

$$\begin{array}{rcccccc} N = & 1 & 0 & 1 & 1 & 0 & 0 \\ & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ N_{\bar{1}} = & 0 & 1 & 0 & 0 & 1 & 1 \end{array}$$

b.

$$\begin{array}{rcccccc} N = & 0 & . & 0 & 1 & 1 & 0 \\ & & & \downarrow & \downarrow & \downarrow & \downarrow \\ N_{\bar{1}} = & 0 & . & 1 & 0 & 0 & 1 \end{array}$$

c.

$$\begin{array}{rccccccccc} N = & 1 & 1 & 0 & . & 1 & 0 & 1 & 0 & 1 \\ & \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ N_{\bar{1}} = & 0 & 0 & 1 & . & 0 & 1 & 0 & 1 & 0 \end{array}$$

Nëse dihet **1**-komplementi, lehtë mund të gjendet edhe **2**-komplementi përkatës, duke shfrytëzuar shprehjen **(1.7)**.

Shembull

2-komplementet e numrave nga shembulli i mësipërm, duke shfrytëzuar **1**-komplementet përkatëse.

a.

$$\begin{aligned} N_{\bar{2}} &= 010011 + (2^0)_{10} \\ &= 010011 + 1 = 010100 \end{aligned}$$

b.

$$\begin{aligned} N_2 &= \mathbf{0.1001} + (2^{-4})_{10} \\ &= \mathbf{0.1001} + \mathbf{0.0001} = \mathbf{0.1010} \end{aligned}$$

c.

$$\begin{aligned} N_2 &= \mathbf{001.01010} + (2^{-5})_{10} \\ &= \mathbf{001.01010} + \mathbf{0.00001} = \mathbf{001.01011} \end{aligned}$$

Për numrat e plotë, **2**-komplementi gjendet duke ia shtuar **1**-komplementit përkatës vlerën **1**, gjë që shihet edhe në shembullin e mësipërm, te numri nën *a*.

Parashenja e numrave

Pavarësisht nga sistemi numerik në të cilin gjendemi, parashenja **+** e numrave pozitivë nuk është e domosdoshme të shënohet, kurse parashenja **-** e numrave negativë shënohet patjetër. Parashenja te pajisjet digjitale paraqitet përmes shifrave binare dhe atë **0** për parashenjën **+**, kurse **1** për parashenjën **-**, p.sh. kështu:

+ 1101	paraqitet si	**0 1101**
- 1010	paraqitet si	**1 1010**

ku me shifrat binare të shkruara veç para numrave janë shënuar parashenjat e tyre.

Numrat negativ mund të paraqiten edhe përmes **2**-komplementit të numrit pozitiv përkatës. Gjatë kësaj duhet të përcaktohet saktë edhe se sa shifra shfrytëzohen për paraqitjen e vlerave komplementare të numrave. Kështu, p.sh., numri **-5** i paraqitur përmes **2**-komplementit të numrit pozitiv **5**, në **3**-bita, është:

101	numri **5**
010	**1**-komplementi
011	**2**-komplementi

Për ta vërtetuar se **2**-komplementi e paraqet numrin negativ, le ta gjejmë zbritjen:

5-5=0

përkatësisht e llogarisim shumën e vlerës pozitive dhe të vlerës negative, duke shfrytëzuar numrat **3**-bitësh:

```
  101      numri 5
+ 011      numri –5    (2-komplementi)
1 000      5-5=0
```

nga shihet se në **3**-bitë është fituar vlera **0**.

Rezultati i njëjtë do të fitohet edhe nëse llogaritja kryhet, p.sh., me numra **6**-bitësh:

```
000101     numri 5
111010     1-komplementi
111011     2-komplementi
```

ose, nëse llogaritet shuma e numrit **5** dhe e **2**-komplementit përkatës:

```
  000101     numri 5
+ 111011     numri –5   (2-komplementi)
1 000000     5-5=0
```

ku rezultati në **6**-bita përsëri është **0**, kurse biti i shtatë, majtas, nuk merret parasyshë, sepse nuk gjendet në hapësirën **6**-bitëshe.

Zbritja indirekte

Zbritja direkte e numrave binarë ndërlikohet për shkak të procesit të huazimit, kur shifra e zbritësit është më e madhe se shifra përkatëse e numrit të zbritur. Por, duke i shfrytëzuar numrat komplementarë, zbritja realizohet përmes mbledhjes. Zbritja mund të realizohet si përmes **(B-1)**-komplementit ashtu edhe përmes **B**-komplementit, gjë që shpjegohet në vazhdim.

Zbritja përmes B-komplementit

Zbritja **M-N** e dy numrave pozitivë, **M** dhe **N**, përmes **B**-komplementit, realizohet kështu:

- Barazohen numrat e shifrave te të dy numrat, duke i shtuar zero numrit me më pakë shifra;
- Mblidhet numri **M** me **B**-komplementin e numrit **N**, përkatësisht gjendet:

$$K = M + N_{\overline{B}}$$

- Nëse gjatë mbledhjes ka bartje, zbritja **M-N** fitohet duke mos e marrë parasyshë bartjen;

Nëse gjatë mbledhjes nuk ka bartje, zbritja **M-N** gjendet përmes shprehjes:

$$\mathbf{M} - \mathbf{N} = -\mathbf{K}_{\overline{\mathbf{B}}}$$

përkatësisht:

$$\mathbf{M} - \mathbf{N} = -(\mathbf{M} + \mathbf{N}_{\overline{\mathbf{B}}})_{\overline{\mathbf{B}}}$$

Shembull Zbritjet e numrave në sistemin decimal:

a.

$$\begin{array}{r} 64397 \\ \underline{-3654} \end{array}$$

b.

$$\begin{array}{r} 7238 \\ \underline{-\ 8432} \end{array}$$

duke shfrytëzuar **10**-komplementin.

a.

$$\begin{array}{rl} 64397 & \mathbf{M} \\ \underline{-03654} & \mathbf{N} \end{array}$$

$$\mathbf{N}_{\overline{10}} = 10^5 - 3654 = 96346$$

$$\begin{array}{rl} 64397 & \mathbf{M} \\ +\ 96346 & \mathbf{N}_{\overline{10}} \\ \hline \boxed{1}60743 & \mathbf{K} \end{array}$$

M-N = 60743

b.

$$\begin{array}{rl} 7238 & \mathbf{M} \\ \underline{-8432} & \mathbf{N} \end{array}$$

$$\mathbf{N}_{\overline{10}} = 10^4 - 8432 = 1568$$

$$\begin{array}{rl} 7238 & \mathbf{M} \\ +\ 1568 & \mathbf{N}_{\overline{10}} \\ \hline 8806 & \mathbf{K} \end{array}$$

$$\mathbf{K}_{\overline{10}} = 10^4 - 8806 = 1194$$

M-N = -1194

Në shembullin e zbritjes së numrave nën *b*, në të vërtetë është rasti i zbritjes **M-N**, kur **M<N**, e cila në jetën e përditshme realizohet duke pasur parasyshë shprehjen:

$$\mathbf{M\text{-}N} = \mathbf{-(N\text{-}M)}$$

Zbritja e numrave të sistemit binar të numrave përmes **2**-komplementit, realizohet plotësisht njëlloj.

Shembull Zbritja e numrave binar:

a.

$$\begin{array}{r} 1110000 \\ -\ \underline{111011} \end{array}$$

b.

$$\begin{array}{r} 1101.10 \\ -\ \underline{1110.11} \end{array}$$

përmes **2**-komplementit.

a.

$$\begin{array}{rl} 1110000 & M \\ \underline{-0111011} & \underline{N} \end{array}$$

$$\begin{array}{lccccccccc} N & = & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ & & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ N_{\bar{1}} & = & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ & & & & & & & + & 1 \\ \hline N_{\bar{2}} & = & 1 & 0 & 0 & 0 & 1 & 0 & 1 \end{array}$$

$$\begin{array}{rl} 1110000 & M \\ +\ 1000101 & N_{\bar{2}} \\ \hline 10110101 & K \end{array}$$

$$M-N=0110101$$

b.

$$\begin{array}{rl} 1101.10 & M \\ \underline{-1110.11} & \underline{N} \end{array}$$

$$\begin{array}{lcccccccc} N & = & 1 & 1 & 1 & 0 & . & 1 & 1 \\ & & \downarrow & \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow \\ N_{\bar{1}} & = & 0 & 0 & 0 & 1 & . & 0 & 0 \\ & & + & & & 0 & . & 0 & 1 \\ \hline N_{\bar{2}} & = & 0 & 0 & 0 & 1 & . & 0 & 1 \end{array}$$

$$\begin{array}{rl} 1101.10 & M \\ +\ 0001.01 & N_{\bar{2}} \\ \hline 1110.11 & K \end{array}$$

$$\begin{array}{lcccccccc} K & = & 1 & 1 & 1 & 0 & . & 1 & 1 \\ & & \downarrow & \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow \\ K_{\bar{1}} & = & 0 & 0 & 0 & 1 & . & 0 & 0 \\ & & + & & & 0 & . & 0 & 1 \\ \hline K_{\bar{2}} & = & 0 & 0 & 0 & 1 & . & 0 & 1 \end{array}$$

$$M - N = -0001.01$$

Zbritja përmes (B-1)-komplementit

Zbritja **M-N** e dy numrave pozitivë **M** dhe **N** mund të kryhet edhe përmes **(B-1)**-komplementit sipas procedurës vijuese.

- Barazohen numrat e shifrave te të dy numrat, duke i shtuar *zero* numrit me më pakë shifra;
- Mblidhet numri **M** me **(B-1)**-komplementin e numrit **N**, përkatësisht gjendet:

$$K = M + N_{\overline{B-1}}$$

- Nëse gjatë mbledhjes ka bartje, zbritja **M-N** fitohet duke ia shtuar bartjen shumës së fituar;

Nëse gjatë mbledhjes nuk ka bartje, zbritja **M-N** gjendet përmes shprehjes:

$$M - N = -K_{\overline{B-1}}$$

përkatësisht:

$$M - N = -(M + N_{\overline{B-1}})_{\overline{B-1}}$$

Shembull	Zbritja e numrave të sistemit decimal:
	a. $\begin{array}{r} 64397 \\ -\ 3654 \\ \hline \end{array}$ *b.* $\begin{array}{r} 7238 \\ -\ 8432 \\ \hline \end{array}$
	duke shfrytëzuar **9**-komplementin.

a.

$$\begin{array}{rl} 64397 & M \\ -\ 03654 & N \\ \hline \end{array}$$

$$N = 03654$$
$$N_{\overline{9}} = 10^5 - 10^0 - 03654 = 96345$$

$$\begin{array}{rl} 64397 & M \\ +\ 96345 & N_{\bar{9}} \\ \hline 1\,60742 & K \\ +1 & \\ \hline 60743 & \end{array}$$

$$M - N = 60743$$

b.

$$\begin{array}{rl} 7238 & M \\ -\ 8432 & N \\ \hline \end{array}$$

$$N = 8432$$
$$N_{\bar{9}} = 10^4 - 10^0 - 8432 = 1567$$

$$\begin{array}{rl} 7238 & M \\ +\ 1567 & N_{\bar{9}} \\ \hline 8805 & K \end{array}$$

$$K_{\bar{9}} = 10^4 - 10^0 - 8805 = 1194$$
$$M - N = -1194$$

Plotësisht njëlloj veprohet edhe gjatë zbritjes së numrave binarë përmes **1**-komplementit.

Shembull Zbritja e numrave binarë:

a.
$$\begin{array}{r} 1110000 \\ -111011 \\ \hline \end{array}$$

b.
$$\begin{array}{r} 1101.10 \\ -1110.11 \\ \hline \end{array}$$

duke shfrytëzuar **1**-komplementin.

a.

$$\begin{array}{rl} 1110000 & M \\ -\ 0111011 & N \\ \hline \end{array}$$

$$\begin{array}{cccccccc} N & = & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ & & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ N_{\bar{1}} & = & 1 & 0 & 0 & 0 & 1 & 0 & 0 \end{array}$$

$$
\begin{array}{rrl}
 & 1110000 & M \\
+ & 1000100 & N_{\bar{1}} \\
\hline
 & 10110100 & K \\
 & +\,1 & \\
\hline
 & 0110101 &
\end{array}
$$

$$M - N = 0110101$$

b.

$$
\begin{array}{rrl}
 & 1101.10 & M \\
- & 1110.11 & N \\
\hline
\end{array}
$$

$$
\begin{array}{lcccccccc}
N & = & 1 & 1 & 1 & 0 & . & 1 & 1 \\
 & & \downarrow & \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow \\
N_{\bar{1}} & = & 0 & 0 & 0 & 1 & . & 0 & 0
\end{array}
$$

$$
\begin{array}{rrl}
 & 1101.10 & M \\
+ & 0001.00 & N_{\bar{1}} \\
\hline
 & 1110.10 & K
\end{array}
$$

$$
\begin{array}{lcccccccc}
K & = & 1 & 1 & 1 & 0 & . & 1 & 0 \\
 & & \downarrow & \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow \\
K_{\bar{1}} & = & 0 & 0 & 0 & 1 & . & 0 & 1
\end{array}
$$

$$M - N = -0001.01$$

Kalimi direkt në mes të sistemeve numerike

Për kalimin prej një sistemi numerik në një sistem tjetër numerik, mund të shfrytëzohet si ndërmjetësues sistemi decimal i numrave. Kështu, p.sh., kalimi prej sistemit heksadecimal në sistemin binar të numrave mund të realizohet duke kaluar prej sistemit heksadecimal në sistemin decimal të numrave dhe pastaj prej sistemit decimal në sistemin binar. Meqë kalimet e tilla kërkojnë mjaft punë, në praktikë shfrytëzohet kalimi direkt në mes të sistemeve numerike.

Kalimi direkt prej sistemit numerik me bazën **b** në sistemin numerik me bazën **B**, për **b<B**, është i mundshëm, nëse vlen raporti:

$$B = b^k \qquad (1.8)$$

ku **k=1,2,....** Ky kalim realizohet duke ndarë numrin në sistemin numerik me bazë **b**, në grupe me nga **k**-shifra. Grupimi i shifrave fillohet prej fundit të numrit, përkatësisht duke shkuar majtas dhe djathtas pikës.

Për sistemet numerike të përmendura më parë, kushti i përcaktuar me shprehjen **(1.8)** plotësohet nga sistemi numerik *binar*, *oktal* dhe *heksadecimal*, përkatësisht kalime direkte mund të bëhen vetëm mes këtyre sistemeve numerike. Gjatë këtyre kalimeve shfrytëzohen ekuivalencat e grupeve të shifrave binare me shifrat e sistemit oktal dhe heksadecimal, të dhëna në tabelën e Fig.1.1.

Decimal	Binar	Heksadecimal	Oktal
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	8	10
9	1001	9	11
10	1010	A	12
11	1011	B	13
12	1100	C	14
13	1101	D	15
14	1110	E	16
15	1111	F	17

Fig.1.1. Ekuivalenca në sisteme numerike të ndryshme

Kalimi binar-heksadecimal

Kalimi prej sistemit binar në sistemin heksadecimal bëhet në dy hapa:

- Numri binar ndahet në grupe prej nga **4** shifra.
- Çdo grupi shifrash i gjendet ekuivalenti heksadecimal.

Te numrat e plotë ndarja fillon prej fundit të numrit, kurse te numrat me pikë binare ndarja bëhet në grupe duke shkuar prej pikës majtas dhe djathtas. Grupet kufitare që kanë më pak se **4** shifra, duhet të plotësohen me zero, duke shtuar zero para pjesës së plotë të numrit, ose në fund të pjesës jo të plotë të numrit.

Shembull

Ekuivalentët heksadecimalë të numrave binar:

a. **101110**
b. **0.1011111011**
c. **1111011.011**

a.

|0010|1110|

$(0010)_2 = (2)_{16}$

$(1110)_2 = (E)_{16}$

$$(101110)_2 = (2E)_{16}$$

b.

|0000|.|1011|1110|1100|

$(0000)_2 = (0)_{16}$

$(1011)_2 = (B)_{16}$

$(1110)_2 = (E)_{16}$

$(1100)_2 = (C)_{16}$

$$(0.1011111011)_2 = (0.BEC)_{16}$$

c.

$$|0111|1011|.|0110|$$

$$(0111)_2 = (7)_{16}$$
$$(1011)_2 = (B)_{16}$$
$$(0110)_2 = (6)_{16}$$

$$(1111011.011)_2 = (7B.6)_{16}$$

Kalimi direkt prej sistemit heksadecimal në sistemin binar të numrave bëhet duke gjetur për çdo shifër të numrit heksadecimal ekuivalentin binar katërbitësh.

Shembull Ekuivalentët binarë për numrat heksadecimalë:

a. **AB35C**
b. **0.3F4**
c. **1D1.4ABC**

a.

AB35C

$$(A)_{16} = (1010)_2$$
$$(B)_{16} = (1011)_2$$
$$(3)_{16} = (0011)_2$$
$$(5)_{16} = (0101)_2$$
$$(C)_{16} = (1100)_2$$
$$(AB35C)_{16} = (10101011001101011100)_2$$

b.

0.3F4

$$(0)_{16} = (0000)_2$$
$$(3)_{16} = (0011)_2$$
$$(F)_{16} = (1111)_2$$
$$(4)_{16} = (0100)_2$$
$$(0.3F4)_{16} = (0.001111110100)_2$$

c.

1D1.4ABC

$(1)_{16} = (0001)_2$

$(D)_{16} = (1101)_2$

$(4)_{16} = (0100)_2$

$(A)_{16} = (1010)_2$

$(B)_{16} = (1011)_2$

$(C)_{16} = (1100)_2$

$(1D1.4ABC)_{16} = (111010001.0100101010111100)_2$

Kalimi binar-oktal

Kalimi prej sistemit binar në sistemin oktal të numrave bëhet në rrugë të njëjtë si edhe kalimi prej sistemit binar në sistemin hekasdecimal të numrave, por këtu shifrat e numrit binar grupohen në grupe me nga **3** shifra.

Shembull Ekuivalentët oktalë për numrat e sistemit binar:

a. **11011101**
b. **0.1101101**
c. **111011.11011**

a.

|011|011|101|

$(011)_2 = (3)_8$

$(101)_2 = (5)_8$

$(11011101)_2 = (335)_8$

b.

|000|.|110|110|100|

$(000)_2 = (0)_8$

$(110)_2 = (6)_8$

$(100)_2 = (4)_8$

$(0.1101101)_2 = (0.664)_8$

c.

$$|111|011|.|110|110|$$

$$(111)_2 = (7)_8$$
$$(011)_2 = (3)_8$$
$$(110)_2 = (6)_8$$
$$(111011.11011)_2 = (73.66)_8$$

Kalimi prej sistemit oktal në sistemin binar të numrave bëhet duke gjetur për çdo shifër të numrit në sistemin oktal ekuivalentin binar treshifror.

Shembull

Ekuivalentët binar për numrat e sistemit octal të numrave:

a. **15247**
b. **0.3624**
c. **4726.35**

a.

15247

$$(1)_8 = (001)_2$$
$$(5)_2 = (101)_2$$
$$(2)_8 = (010)_2$$
$$(4)_8 = (100)_2$$
$$(7)_8 = (111)_2$$
$$(15247)_8 = (1101010100111)_2$$

b.

0.3624

$$(0)_8 = (000)_2$$
$$(3)_8 = (011)_2$$
$$(6)_8 = (110)_2$$
$$(2)_8 = (010)_2$$
$$(4)_8 = (100)_2$$
$$(0.3624)_8 = (0.011110010100)_2$$

c.

4726.35

$(4)_8 = (100)_2$
$(7)_8 = (111)_2$
$(2)_8 = (010)_2$
$(6)_8 = (110)_2$
$(3)_8 = (011)_2$
$(5)_8 = (101)_2$

$$(4726.35)_8 = (100111010110.011101)_2$$

Kalimi oktal-heksadecimal

Kalimi prej sistemit oktal në sistemin heksadecimal bëhet me ndërmjetësimin e sistemit binar të numrave.

Shembull	Ekuivalentët heksadecimalë për numrat e sistemit oktal:
	a. **7523**
	b. **0.3624**
	c. **1125.641**

a.

7523

$(7)_8 = (111)_2$
$(5)_8 = (101)_2$
$(2)_8 = (010)_2$
$(3)_8 = (011)_2$

$$(7523)_8 = (111101010011)_2$$

$|1111|0101|0011|$

$(1111)_2 = (F)_{16}$
$(0101)_2 = (5)_{16}$
$(0011)_2 = (3)_{16}$

$$(7523)_2 = (F53)_{16}$$

b.

0.3624

$(0)_8 = (000)_2$
$(3)_8 = (011)_2$
$(6)_8 = (110)_2$
$(2)_8 = (010)_2$
$(4)_8 = (100)_2$

$$(0.3624)_8 = (0.011110010100)_2$$

$|0000|.|0111|1001|0100|$

$(0000)_2 = (0)_{16}$
$(0111)_2 = (7)_{16}$
$(1001)_2 = (9)_{16}$
$(0100)_2 = (4)_{16}$

$$(0.3624)_8 = (0.794)_{16}$$

c.

1125.641

$(1)_8 = (001)_2$
$(2)_8 = (010)_2$
$(5)_8 = (101)_2$
$(6)_8 = (110)_2$
$(4)_8 = (100)_2$

$$(1125.641)_8 = (1001010101.110100001)_2$$

$|0010|0101|0101|.|1101|0000|1000|$

$(0010)_2 = (2)_{16}$
$(0101)_2 = (5)_{16}$
$(1101)_2 = (D)_{16}$
$(0000)_2 = (0)_{16}$
$(1000)_2 = (8)_{16}$

$$(1125.641)_8 = (255.D08)_{16}$$

Gjatë kalimit prej sistemit heksadecimal në sistemin oktal të numrave veprohet në drejtim të kundërt me atë të kalimit prej sistemit oktal në sistemin heksadecimal. Këtu, në sistemin binar të numrave kalohet duke zëvendësuar shifrat heksadecimale me grupe **4**-shifrore të ekuivalentëve binarë përkatës. Pastaj gjendet ekuivalenti oktal për numrin në sistemin binar të numrave, duke shfrytëzuar procedurën e shpjeguar më parë.

Numrat me pikë të lëvizshme

Numrat me pikë decimale, binare, heksadecimale ose oktale, të cilët u përmendën më sipër, ndryshe quhen edhe *numra me pikë fikse*. Por, në praktikë, shpeshëherë përdoren edhe *numra me pikë të lëvizshme*, siç janë, p.sh., numrat decimalë:

$$3 \cdot 10^{9}$$

$$9.175 \cdot 10^{-31}$$

Numrat binar

Në formë të përgjithshme, numrat binarë me pikë të lëvizshme shkruhen kështu:

$$N = M \cdot 2^{\pm E}$$

ku është:

M - mantisa
E - eksponenti.

Kur kemi të bëjmë me numra binar, mantisa dhe eksponenti shkruhen si numra binar.

Në praktikë, format eksponenciale të numrave jepen ashtu që mantisa të jetë *numër më i vogël se* **1**, përkatësisht vlera e saj të lëvizë në diapazonin:

$$0.5 \le (M)_{10} < 1$$

te numrat binarë mantisa merret ashtu që shifra pas pikës binare gjithnjë të jetë **1**.

Eksponenti tregon se për sa shifra duhet të lëvizë pika binare *para* **(+E)**, ose *pas* **(-E)**, për ta fituar numrin përkatës me pikë fikse. Për arsye praktike, eksponenti i numrave binar me pikë të lëvizëshme shpeshëherë jepet në formë decimale. Por, në pajisjet digjitale mantisa dhe eksponenti ruhen në formën binare.

Shembull Forma me pikë të lëvizshme e numrave me pikë fikse:

a. **111101111.01**
b. **0.0000011101**
c. **111.01101**
d. **-1011.001001**

a.

$$\begin{aligned} 111101111.01 &= 0.11110111101 \cdot 2^{(9)_{10}} \\ &= 0.11110111101 \cdot 2^{1001} \end{aligned}$$

b.

$$\begin{aligned} 0.0000011101 &= 0.11101 \cdot 2^{-(5)_{10}} \\ &= 0.11101 \cdot 2^{-101} \end{aligned}$$

c.

$$\begin{aligned} 111.01101 &= 0.11101101 \cdot 2^{(3)_{10}} \\ &= 0.11101101 \cdot 2^{11} \end{aligned}$$

d.

$$\begin{aligned} -1011.001001 &= -0.1011001001 \cdot 2^{(4)_{10}} \\ &= -0.1011001001 \cdot 2^{100} \end{aligned}$$

Aritmetika e numrave me pikë të lëvizshme

Mbi numrat binarë me pikë të lëvizshme mund të zbatohen katër operacionet aritmetikore elementare, duke pasur parasysh rregullat për operim me numra me pikë të lëvizshme në sistemin decimal të numrave.

Mbledhja

Mund të mblidhen vetëm numrat me pikë të lëvizshme të cilët kanë *eksponentë të barabartë*. Prandaj, gjatë mbledhjes së numrave të cilët kanë eksponentë të ndryshëm së pari barazohen eksponentët e tyre, *duke rritur eksponentin më të vogël*. Pastaj, mbledhja kryhet në atë mënyrë që mblidhen mantisat dhe përshkruhet eksponenti.

Shembull Mbledhja e numrave binar me pikë të lëvizshme:

a.

$$\begin{array}{r} 0.1011 \cdot 2^{101} \\ +\,0.1101 \cdot 2^{11} \\ \hline \end{array}$$

b.

$$\begin{array}{r} 0.101101 \cdot 2^{111} \\ +\,0.111011 \cdot 2^{110} \\ \hline \end{array}$$

c.

$$\begin{array}{r} 0.111011 \cdot 2^{11} \\ +\,0.100011 \cdot 2^{110} \\ \hline \end{array}$$

d.

$$\begin{array}{r} 0.1001110 \cdot 2^{11} \\ +\,0.1110001 \cdot 2^{-10} \\ \hline \end{array}$$

a.

$$\begin{array}{r} 0.1011 \cdot 2^{101} \\ +\,0.1101 \cdot 2^{11} \\ \hline \end{array} \qquad \begin{array}{r} 0.101100 \cdot 2^{101} \\ +\,0.001101 \cdot 2^{101} \\ \hline 0.111001 \cdot 2^{101} \end{array}$$

b.

$$\begin{array}{r} 0.101101 \cdot 2^{111} \\ +\,0.111011 \cdot 2^{110} \\ \hline \end{array} \qquad \begin{array}{r} 0.1011010 \cdot 2^{111} \\ +\,0.0111011 \cdot 2^{111} \\ \hline 1.0010101 \cdot 2^{111} \end{array}$$

ose $0.10010101 \cdot 2^{1000}$

c.

$$\begin{array}{r} 0.111011 \cdot 2^{11} \\ +\,0.100011 \cdot 2^{110} \\ \hline \end{array} \qquad \begin{array}{r} 0.000111011 \cdot 2^{110} \\ +\,0.100011000 \cdot 2^{110} \\ \hline 0.101010011 \cdot 2^{110} \end{array}$$

d.

$$\begin{array}{r} 0.1001110 \cdot 2^{11} \\ +\,0.1110001 \cdot 2^{-10} \\ \hline \end{array} \qquad \begin{array}{r} 0.100111000000 \cdot 2^{11} \\ +\,0.000001110001 \cdot 2^{11} \\ \hline 0.101000110001 \cdot 2^{11} \end{array}$$

Zbritja

Zbritja e numrave me pikë të lëvizshme kryhet plotësisht njëlloj si edhe mbledhja, duke zbritur në fund mantisat.

Shembull Zbritja e numrave me pikë të lëvizshme:

a.

$$\begin{array}{r} 0.11011 \cdot 2^{110} \\ -\ 0.10110 \cdot 2^{11} \\ \hline \end{array}$$

b.

$$\begin{array}{r} 0.1111011 \cdot 2^{101} \\ -\ 0.1111101 \cdot 2^{111} \\ \hline \end{array}$$

c.

$$\begin{array}{r} 0.111101 \cdot 2^{10} \\ -\ 0.110110 \cdot 2^{11} \\ \hline \end{array}$$

d.

$$\begin{array}{r} 0.1101101 \cdot 2^{10} \\ -\ 0.1111011 \cdot 2^{-10} \\ \hline \end{array}$$

a.

$$\begin{array}{r} 0.11011 \cdot 2^{110} \\ -\ 0.10110 \cdot 2^{11} \\ \hline \end{array} \qquad \begin{array}{r} 0.11011000 \cdot 2^{110} \\ -\ 0.00010110 \cdot 2^{110} \\ \hline 0.11000010 \cdot 2^{110} \end{array}$$

b.

$$\begin{array}{r} 0.1111011 \cdot 2^{101} \\ -\ 0.1111101 \cdot 2^{111} \\ \hline \end{array} \qquad \begin{array}{r} 0.001111011 \cdot 2^{111} \\ -0.111110100 \cdot 2^{111} \\ \hline -0.101111001 \cdot 2^{111} \end{array}$$

c.

$$\begin{array}{r} 0.111101 \cdot 2^{10} \\ -\ 0.110110 \cdot 2^{11} \\ \hline \end{array} \qquad \begin{array}{r} 0.0111101 \cdot 2^{11} \\ -0.1101100 \cdot 2^{11} \\ \hline 0.0101111 \cdot 2^{11} \end{array}$$

ose

$$-0.101111 \cdot 2^{10}$$

d.

$$\begin{array}{r} 0.1101101 \cdot 2^{10} \\ -\ 0.1111011 \cdot 2^{-10} \\ \hline \end{array} \qquad \begin{array}{r} 0.11011010\ 000 \cdot 2^{10} \\ -0.00001111\ 011 \cdot 2^{10} \\ \hline -0.11001010\ 101 \cdot 2^{10} \end{array}$$

Shumëzimi

Dy numra me pikë të lëvizëshme shumëzohen duke shumëzuar mantisat dhe duke mbledhur eksponentët e tyre. P.sh., nëse shumëzohen numrat:

$$A = M_a \cdot 2^{E_a}$$
$$B = M_b \cdot 2^{E_b}$$

rezultati i shumëzimit fitohet kështu:

$$\begin{aligned} C &= A \cdot B \\ &= (M_a \cdot 2^{E_a}) \cdot (M_b \cdot 2^{E_b}) \\ &= M_a \cdot M_b \cdot 2^{E_a+E_b} \end{aligned}$$

Shembull Shumëzimi i numrave me pikë të lëvizëshme:

a.

$$\begin{array}{r} 0.1011 \cdot 2^{11} \\ \times\ 0.1101 \cdot 2^{10} \\ \hline \end{array}$$

b.

$$\begin{array}{r} 0.111101 \cdot 2^{101} \\ \times\ 0.100011 \cdot 2^{10} \\ \hline \end{array}$$

a.

$$\begin{array}{r} 0.1011 \cdot 2^{11} \\ \times\ 0.1101 \cdot 2^{10} \\ \hline 01011 \\ 00000 \\ 01011 \\ 01011 \\ 00000 \\ \hline 0.10001111 \cdot 2^{11+10} \\ \text{ose} \\ 0.10001111 \cdot 2^{101} \end{array}$$

b.

$$\begin{array}{r} 0.111101 \cdot 2^{101} \\ \times\ 0.100011 \cdot 2^{10} \\ \hline 0111101 \\ 0111101 \\ 0000000 \\ 0000000 \\ 0000000 \\ 0111101 \\ 0000000 \\ \hline 0.100001010111 \cdot 2^{101+10} \\ \text{ose} \\ 0.100001010111 \cdot 2^{111} \end{array}$$

Pjesëtimi

Gjatë pjesëtimit të dy numrave me pikë të lëvizshme pjesëtohen mantisat dhe zbriten eksponentet e tyre. P.sh., nëse pjesëtohen numrat:

$$A = M_a \cdot 2^{E_a}$$
$$B = M_b \cdot 2^{E_b}$$

rezultati i pjesëtimit fitohet kështu:

$$\begin{aligned} C &= A / B \\ &= (M_a \cdot 2^{E_a}) / (M_b \cdot 2^{E_b}) \\ &= (M_a / M_b) \cdot 2^{E_a - E_b} \end{aligned}$$

Shembull Pjesëtimi i numrave me pikë të lëvizshme:

a. $\dfrac{0.101 \cdot 2^{110}}{0.1001 \cdot 2^{10}}$

b. $\dfrac{0.111011 \cdot 2^{11}}{0.10001 \cdot 2^{111}}$

a.

$$\begin{aligned} \frac{0.101 \cdot 2^{110}}{0.1001 \cdot 2^{10}} &= \frac{0.101}{0.1001} \cdot 2^{110-10} \\ &= 1.000111 \cdot 2^{100} = 0.1000111 \cdot 2^{101} \end{aligned}$$

b.

$$\begin{aligned} \frac{0.111011 \cdot 2^{11}}{0.10001 \cdot 2^{111}} &= \frac{0.111011}{0.10001} \cdot 2^{11-111} \\ &= 1.1 \cdot 2^{-100} = 0.11 \cdot 2^{-11} \end{aligned}$$

Kodet 2

Gjuha përmes së cilës njerëzit komunikojnë mes vete formohet si grumbull fjalësh, të cilat në fakt paraqesin kombinime të një numri të caktuar tingujsh - kur flasim, përkatësisht shkronjash - kur shkruajmë. Kështu, kur e themi fjalën *lapsi*, në atë rast e nënkuptojmë një mjet për shkruarje, kurse kur themi *fletore*, mendojmë në diçka për shkruarje, sepse ashtu është *marrëveshja* në gjuhën shqipe. Grumbulli i të gjitha fjalëve paraqet *gjuhën*, përkatësisht *kodin* (ang. code) për komunikim mes njerëzve të cilët e flasin atë gjuhë.

Në jetën e përditshme njeriu shfrytëzon kode të ndryshme. P.sh., gjatë kalimit nëpër udhëkryq, ku komunikacioni rregullohet përmes semaforit, shfrytëzohet kodi sipas të cilit udhëkryqi mund të kalohet nëse është e ndezur ngjyra e gjelbër, kurse në ngjyrë të kuqe nuk lejohet kalimi i udhëkryqit.

Te pajisjet digjitale, për përpunim dhe për transmetim të informatave, gjithashtu përdoren kode të ndryshme, te të cilët çdo shifre numerike, shkronje, ose simboli, i shoqërohet një kombinim i caktuar shifrash binare **1** dhe **0**. Kombinimet e tilla, të cilat krijohen për kodimin e simboleve elementare të një kodi, quhen *fjalë kodike* (ang. code word).

Në vazhdim do të përmenden disa lloj kodesh binare të cilët përdoren më shpesh.

Kodet BCD

Njeriu gjatë llogaritjeve të ndryshme në jetën e përditshme e shfrytëzon sistemin decimal të numrave. Kurse te pajisjet digjitale, gjatë përpunimit dhe transmetimit të informatave, shfrytëzohen kode binare. Për këtë arsye, janë krijuar të ashtuquajturit kode **BCD** (nga Binary Coded Decimal), te të cilët çdo shifre decimale i shoqërohet një fjalë kodike binare, përkatësisht, bëhet *kodimi binar i shifrave decimale*.

Gjatë krijimit të kodeve **BCD** tentohet që fjalët kodike të zgjidhen ashtu që kodi të jetë i përshtatshëm për numërim, llogaritje, konvertim, zbulim ose për korrigjim të gabimeve etj. Te këto kode çdo shifër decimale zëvendësohet me një grup shifrash binare, përkatësisht me *fjalë kodike katërshifrore*. Me katër shifra binare mund të krijohen gjithsej **16** fjalë kodike të ndryshme. Por, për kodimin e shifrave decimale zgjidhen vetëm **10** fjalë kodike, gjë që jep mundësi të krijimit të më shumë kodeve **BCD**.

Zgjedhja e kombinimeve të shifrave binare brenda një kodi mund të bëhet duke e pasur, ose duke mospasur, parasysh peshën e pozicioneve brenda fjalëve kodike. Në këtë mënyrë dallohen kodet **BCD** *me peshë* dhe *pa peshë*.

Kodet me peshë

Nëse shifrave binare **x**, **y**, **z** dhe **v**, brenda fjalës kodike **xyzv**, u korrespondojnë peshat **a**, **b**, **c** dhe **d**, për çdo fjalë kodike të kodit **BCD** mund të llogaritet ekuivalenti decimal përkatës kështu:

$$\mathbf{N = x \cdot a + y \cdot b + z \cdot c + v \cdot d}$$

Disa nga kodet me peshë të cilët përdoren më shpesh janë dhënë në tabelën e Fig.2.1, ku shifrave decimale u korrespondojnë fjalët kodike në kolonën e parë të tabelës.

Fjalët kodike	Kodi								
	Pastër binar	8421 (NBCD)	2421 (AIKEN)	5211 (WHITE)	4221	5421	5221	5311	84-2-1
0000	0	0	0	0	0	0	0	0	0
0001	1	1	1	1	1	1	1	1	
0010	2	2	2		2	2	2		
0011	3	3	3	2	3	3	3	2	
0100	4	4	4			4		3	4
0101	5	5		3				4	3
0110	6	6			4		4		2
0111	7	7		4	5				1
1000	8	8		5		5	5	5	8
1001	9	9		6		6	6	6	7
1010	10					7	7		6
1011	11		5	7		8	8	7	5
1100	12		6		6	9		8	
1101	13		7	8	7			9	
1110	14		8		8		9		
1111	15		9	9	9				9

Fig.2.1 Kodet* BCD *me peshë

Kodi NBCD

Si kod **BCD** i cili përdoret më shpesh është kodi **8421**, i cili ndryshe quhet edhe **NBCD** (nga Natural Binary Coded Decimal), sepse fjalët kodike të këtij kodi u korrespondojnë **10** numrave të sistemit binar natyror.

Kodimi i numrave decimalë, në kodin **NBCD**, kryhet duke zëvendësuar shifrat e veçanta brenda numrave me fjalët kodike përkatëse.

Shembull

Kodimi i numrave të sistemit decimal:

a. **5476**
b. **89.763**

në kodin **NBCD**.

a.

$$(5476)_{10}=(\underline{0101}\ \underline{0100}\ \underline{0111}\ \underline{0110})_{NBCD}$$

b.

$$(89.763)_{10}=(\underline{1000}\ \underline{1001}.\underline{0111}\ \underline{0110}\ \underline{0011})_{NBCD}$$

Ekuivalenti decimal i numrave të koduar në kodin **NBCD** gjendet duke ndarë numrin në grupe **4**-shifrore.

Shembull

Ekuivalentët decimalë të numrave të koduar në kodin **NBCD**:

a. **1000001101010001**
b. **01100010.100101110100**

a.

$$(\underline{1000}\ \underline{0011}\ \underline{0101}\ \underline{0001})_{NBCD}=(8351)_{10}$$

b.

$$(\underline{0110}\ \underline{0010}.\underline{1001}\ \underline{0111}\ \underline{0100})_{NBCD}=(62.974)_{10}$$

Peshat e shifrave brenda fjalëve kodike të këtij kodi janë:

$$2^3 \quad 2^2 \quad 2^1 \quad 2^0$$

përkatësisht:

$$8 \quad 4 \quad 2 \quad 1$$

nga edhe është krijuar emri i kodit. Kështu, p.sh., për fjalën kodike **0101**, në bazë të shprehjes së dhënë më sipër, fitohet numri:

$$N = 0 \cdot 8 + 1 \cdot 4 + 0 \cdot 2 + 1 \cdot 1 = 5$$

i cili i përgjigjet ekuivalentit decimal, nëse fjala kodike përkatëse merret si numër binar.

Mbledhja në kodin NBCD

Me qëllim që të tregohet se operacionet aritmetikore kryhen edhe mbi numrat e paraqitur përmes kodeve të ndryshme, në vazhdim është dhënë procedura e mbledhjes së numrave në kodin **NBCD**.

Fjalët kodike të cilat shfrytëzohen nga kodi **NBCD** paraqesin *fjalë kodike të lejuara*, për dallim nga *fjalët kodike të ndaluara*, të cilat nuk i përkasin këtij kodi (shih tabelën e dhënë në Fig.2.2).

Numrat decimalë	Numrat binarë	
0	0000	Fjalët kodike të lejuara
1	0001	
2	0010	
3	0011	
4	0100	
5	0101	
6	0110	
7	0111	
8	1000	
9	1001	
10	1010	Fjalët kodike të ndaluara
11	1011	
12	1100	
13	1101	
14	1110	
15	1111	

Fig.2.2. Fjalët kodike të lejuara dhe të ndaluara

Mbledhja në kodin **NBCD** kryhet në dy faza:

- Mblidhen numrat e dhënë, duke shfrytëzuar rregullat e mbledhjes binare.
- Korrigjohen fjalët kodike që fitohen pas mbledhjes, nëse ato janë:
  fjalë kodike të ndaluara, ose
  fjalë kodike me bartje.

Fjalët kodike korrigjohen duke ua shtuar numrin decimal **6** (sa është numri i fjalëve kodike të ndaluara), përkatësisht ekuivalentin binar përkatës **0110**. Pas korrigjimit të parë, vazhdon korrigjimi i fjalëve kodike të ndaluara. Por, korrigjimi i fjalëve prej të cilave ka bartje ndërpritet. Procesi i korrigjimit përfundon kur në rezultat nuk ka më fjalë kodike të ndaluara.

Shembull Mbledhja e numrave të shkruar në kodin **NBCD**:

a.

```
 0101 1000
+1001 0100
```

b.

```
 0010 0011 1000
+0011 0111 0010
```

c.

```
 0100 0111 1001 1000
+0001 0110 0111 1001
```

d.

```
 0011 0111 0110
+0100 0010 1001
```

e.

```
 1000 0011 0111
+0100 1001 0110
```

f.

```
 1001.0110 1000
+0111.1000 0111
```

a.

```
   1                 11 1 1    Bartja
 0101 1000           1110 1100
+1001 0100          +0110 0110
 1110 1100      0001 0101 0010
```

b.

```
  1   111              11 11 1 11   Bartja
 0010 0011 1000      0101 1010 1010
+0011 0111 0010      +    0110 0110
 0101 1010 1010      0110 0001 0000
```

c.

```
      1111 1111              11 11
 0100 0111 1001 1000      0101 1110 0001 0001
+0001 0110 0111 1001      +    0110 0110 0110
 0101 1110 0001 0001      0110 0100 0111 0111
```

d.

```
      11                                    11 11   Bartja
 0011 0111 0110                        0111 1001 1111
+0100 0010 1001                       +          0110
 0111 1001 1111                        0111 1010 0101
                                      +     0110
                                       1000 0000 0101
```

Për rezulatet e fituara gjatë mbledhjes në kodin **NBCD** mund të bëhet edhe prova përmes sistemit numerik decimal, duke konvertuar në këtë sistem numerik numrat që mblidhen dhe rezultatin. Kështu, p.sh., gjatë mbledhjes nën *d* te shembulli i mësipërm, kemi:

```
   0011 0111 0110      376
 + 0100 0010 1001    + 429
   1000 0000 0101      805
```

Konvertimi i numrave të shprehur në kodin **NBCD**, në numra të sistemit decimal, bëhet përmes zëvendësimit direkt të fjalëve kodike me ekuivalentët decimalë përkatës.

Kode të tjera

Përveç kodit **NBCD**, siç është treguar edhe në tabelën e Fig.2.1, në grupin e kodeve me peshë bëjnë pjesë edhe kode të tjera. Për paraqitjen e numrave decimalë në këto kode përdoret parimi i njëjtë me atë që u tha te kodi **NBCD**, përkatësishti zëvendësohen shifrat decimale me fjalët kodike përkatëse.

Shembull Paraqitja e numrit decimal **596** në të gjithë kodet **BCD** të dhënë në tabelën e Fig.2.1.

Kodi	Numri		
	5	9	6
8421	0101	1001	0110
2421	1011	1111	1100
5211	1000	1111	1001
4221	0111	1111	1100
5421	1000	1100	1001
5221	1000	1110	1001
5311	1000	1101	1001
84-2-1	1011	1111	1010

Fig.2.3 Paraqitja e numrit në kode të ndryshme

Kodi **BCD 84-2-1** dallohet nga kode të tjera, të dhëna në tabelën e Fig.2.1, sepse peshat e dy shifrave të fundit në fjalët kodike përkatëse kanë vlerë negative. Prandaj, ky kod quhet edhe *kod me peshë negative* (ang. negative weighted code) për dallim nga kode të tjera të cilat paraqesin *kode me peshë pozitive* (ang. positive weighted code).

Kodet pa peshë

Si kod **BCD** pa peshë këtu do të përmendet kodi **Excess-3**, i cili shkurt shënohet edhe **XS3**. Ky kod fitohet nga kodi **NBCD**, duke ia shtuar çdo fjale kodike ekuivalentin binar të numrit **3**, gjë që shihet edhe në tabelën e cila është dhënë në Fig.2.4.

Shifrat decimale	Fjalët kodike
0	0011
1	0100
2	0101
3	0110
4	0111
5	1000
6	1001
7	1010
8	1011
9	1100

***Fig.2.4 Kodi* Excess-3**

Kodi **XS3** është i përshtatshëm për operacione aritmetikore, gjë që mund të shihet përmes operacionit të mbledhjes, i cili kryhet në dy hapa:

- Mblidhen numrat e dhënë, duke shfrytëzuar rregullat e mbledhjes binare.
- Korrigjohen të gjitha fjalët kodike që fitohen pas mbledhjes. Për korrigjim shfrytëzohet procedura vijuese.
  Fjalëve kodike *me bartje u shtohet* numri **0011**.
  Fjalëve kodike *pa bartje u zbritet* numri **0011**.

Fjala kodike e cila fitohet si rezultat i bartjes përfundimtare nuk korrigjohet.

Shembull Mbledhja e numrave të shkruar në kodin **XS3**:

a.

```
  0011 1100 1100
+ 0100 1000 1001
```

b.

```
  1100 0110 1000
+ 1000 0011 0110
```

a.

```
 1111    1
 0011 1100 1100
+0100 1000 1001
 1000 0101 0101
-0011+0011+0011
 0101 1000 1000
```

b.

```
   1      11
     1100 0110 1000
    +1000 0011 0110
0001 0100 1001 1110
    +0011-0011-0011
0001 0111 0110 1011
```

Për rezulatet e fituara gjatë mbledhjes në kodin **XS3** mund të bëhet edhe prova përmes sistemit numerik decimal, njëlloj si edhe gjatë mbledhjes në kodin **NBCD**. Kështu, te mbledhja e mësipërme, p.sh. nën *b*, kemi:

```
     1100 0110 1000        935
   + 1000 0011 0110      + 503
0001 0111 0110 1011       1438
```

Këtu ekuivalentët decimalë të numrave të shkruar në kodin **XS3** gjenden duke zëvendësuar fjalët kodike me shifrat decimale përkatëse.

Kodet ciklike

Kodet te të cilat fjalët kodike të njëpasnjëshme dallohen mes vete vetëm në një shifër, duke formuar njëkohësisht një tërësi të mbyllur, quhen *kode ciklike* (ang. cyclic code). Kodet e tilla shfrytëzohen kryesisht gjatë konvertimit të sinjaleve analoge në digjitale, sepse zbulohen lehtë gabimet eventuale gjatë konvertimit.

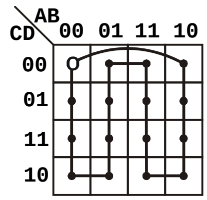
Si kod ciklik që përdoret më shpesh është *kodi i Gray-it* (ang. Gray code), i dhënë në tabelën e Fig.2.5.

Shifrat decimale	Fjalët kodike
0	0000
1	0001
2	0011
3	0010
4	0110
5	0111
6	0101
7	0100
8	1100
9	1101
10	1111
11	1110
12	1010
13	1011
14	1001
15	1000

Fig.2.5 Kodi i Gray-it

Nëse shikohen fjalët kodike të dhëna në tabelë do të vërehet ligjshmëria e theksuar më sipër, për dallim të fjalëve të njëpasnjëshme kodike vetëm në një shifër. Kjo ligjshmëri ruhet edhe mes fjalës kodike në fund dhe në fillim të tabelës, prandaj thuhet se kodi ciklik njëkohësisht formon edhe një tërësi të mbyllur.

Ligjshmëria e lidhjes së fjalëve kodike të njëpasnjëshme te *kodi i Gray-it* më së miri shihet nëse fjalët kodike vendosen në tabelën për kodim:



prej ku, fjalët kodike përkatëse për numrat decimalë formohen si kombinim i vlerave numerike **ABCD**, ku fusha **0000** i përgjigjet fjalës së parë kodike.

Duke pasur parasysh *kodin e Gray-it*, në praktikë përdoret disku rrotullues (Fig.2.6), i ndarë në **4 shtigje** dhe **16 sektorë**, përmes së cilit tek pajisjet rrotulluese mund të detektohet këndi i rrotullimit. Nëse hapësirat e nxiera paraqesin sipërfaqe përçuese dhe mbi disk mbështeten **4** brusha përçuese të pozicionuara në **4** shtigjet e veçanta, në daljet **A**, **B**, **C** dhe **D** do të merren vlerat logjike **1**, kur gjatë lëvizjes së diskut brushat gjenden mbi sipërfaqet përçuese.

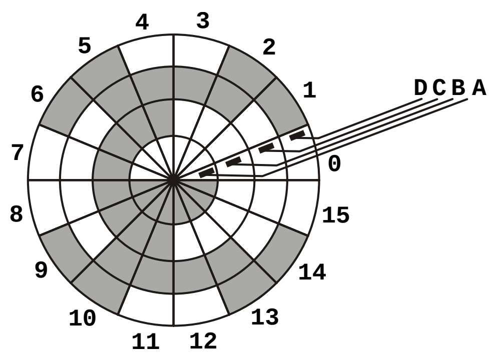


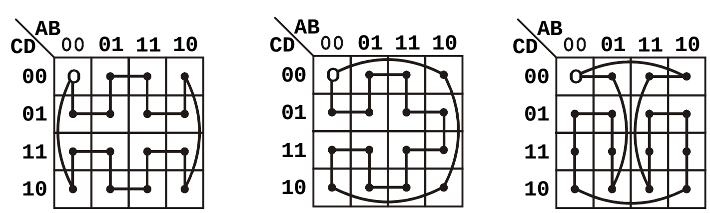
Fig.2.6 Disku i koduar sipas kodit të Gray-it

Në bazë të parimit të theksuar më sipër për diferencë mes fjalëve kodike të njëpasnjëshme në një shifër binare mund të krijohen edhe kode ciklike të tjera. Gjatë kësaj, procedura e krijimit shihet më mirë nëse shfrytëzohet tabela e fjalëve të mundshme kodike, e dhënë më sipër.

Shembull

Kodet ciklike të përcaktuara përmes tabelave për kodim:

a. *b.* *c.*



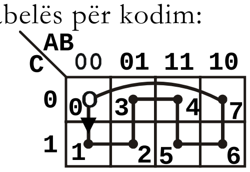
Numri decimal	Kodi ciklik		
	a	b	c
0	0000	0000	0000
1	0010	0001	0100
2	0011	0101	0110
3	0111	0100	0111
4	0110	1100	0101
5	1110	1101	0001
6	1111	1001	0011
7	1011	1011	0010
8	1010	1111	1010
9	1000	1110	1011
10	1001	0110	1001
11	1101	0111	1101
12	1100	0011	1111
13	0100	0010	1110
14	0101	1010	1100
15	0001	1000	1000

Fig.2.7 Shembuj të kodeve ciklike

Kodet ciklike nuk është e domosdoshme të formohen vetëm me fjalë kodike **4**-shifrore.

Shembull

Kodi ciklik për kodimin e shifrave të sistemit oktal të numrave, i përcaktuar përmes tabelës për kodim:



Shifrat oktale	Fjalët kodike
0	000
1	001
2	011
3	010
4	110
5	111
6	101
7	100

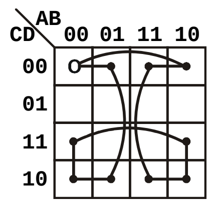
Fig.2.8 Kodi ciklik për kodimin e shifrave oktale

Kodet ciklike që u dhanë më sipër, quhen edhe *kode ciklike komplete*, sepse gjatë krijimit të tyre shfrytëzohen të gjitha fjalët kodike të mundshme. Por, në praktikë shfrytëzohen edhe *kode ciklike jokomplete*, të cilat nuk i përmbajnë të gjitha fjalët kodike të mundshme.

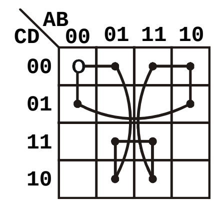
Shembull

Kodet ciklike jokomplete, të përcaktuara përmes tabelave për kodim:

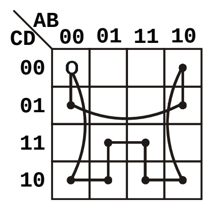
a.



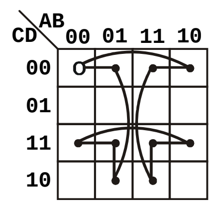
b.



c.



d.



përmes së cilave kodohen shifrat e sistemit decimal të numrave.

Shifra decimale	Kodi ciklik jokomplet			
	a	b	c	d
0	0000	0000	0000	0000
1	0100	0001	0001	0100
2	0110	1001	1001	0110
3	0010	1000	1000	0111
4	0011	1100	1010	0011
5	1011	1110	1110	1011
6	1010	1111	1111	1111
7	1110	0111	0111	1110
8	1100	0110	0110	1100
9	1000	0100	0010	1000

Fig.2.9 Kode ciklike jokomplete

Në tabelat për kodim nuk është e domosdoshme të vizatohet edhe cikli përkatës, por radha e zgjedhjes së fushave mund të përcaktohet me numra ose me shkronja.

Shembull

Kodet ciklike jokomplete të përcaktuara përmes tabelave për kodim:

a.

CD \ AB	00	01	11	10
00	8	9		
01	7	0	1	2
11	6	5	4	3
10				

b.

CD \ AB	00	01	11	10
00		1	2	3
01				4
11	8	7	6	5
10	9	0		

ku cikli i fjalëve kodike përcaktohet me radhën e shifrave decimale.

Shifra decimale	Kodi	
	a	b
0	0101	0110
1	1101	0100
2	1001	1100
3	1011	1000
4	1111	1001
5	0111	1011
6	0011	1111
7	0001	0111
8	0000	0011
9	0100	0010

Fig.2.10 Shembuj të kodeve ciklike jokomplete

Kodet optimale

Gjatë transmetimit të informatave në distancë, për ta zvogëluar kohën e transmetimit, përdoren *kodet optimale*. Në pjesën paraprake u përmendën kodet me gjatësi fikse të fjalëve kodike. Por, për krijimin e kodeve optimale imponohet nevoja që fjalët kodike brenda një kodi të kenë gjatësi të ndryshme.

Për krijimin e kodit optimal, përmes të cilit kodohet një grumbull simbolesh elementare, përdoret parimi sipas të cilit *gjatësia e fjalëve kodike varet nga frekuenca e përdorimit të informatave elementare*. Në bazë të këtij parimi, simboleve elementare që përdoren më shpesh u ndahen fjalë kodike më të shkurtra, gjë që mundëson transmetimin e inforamatave për kohë minimale.

Procedura e përcaktimit të fjalëve kodike mbështetet në *frekuencat relative*, përkatësisht *gjasat e përdorimit* të simboleve elementare që kodohen. Në literaturë përmenden metoda të ndryshme të kodimit optimal. Këtu, në vijim do të jepen dy metoda të kodimit optimal, *metoda e Shannon-Fanos* dhe *metoda e Huffman-it*.

Metoda e Shannon-Fanos

Kodimi optimal sipas kësaj metode mbështetet në procedurën vijuese:

1. Radhiten simbolet elementare, në bazë të madhësive së gjasave të përdorimit të tyre.

2. Informatat elementare ndahen në dy grupe, ashtu që shumat e gjasave të grupeve të jenë të barabarta, përkatësisht diferenca absolute e tyre të jetë minimale.

3. Grupeve u shoqërohen shifra binare të ndryshme, p.sh., grupit të sipërm shifra **0**, kurse grupit të poshtëm - shifra **1**. Ky parim ruhet deri në fund të procesit të kodimit.

Procesi i ndarjes në dy grupe (hapi i *2*) dhe i shoqërimit të shifrave binare (hapi i *3*) vazhdon edhe për grupet e riformuara, deri në copëtimin e plotë të tyre. Nga shifrat binare të cilat u shoqërohen grupeve gjatë copëtimit suksesiv të tyre, formohen fjalët kodike të simboleve elementare, përkatësisht kodi optimal i kërkuar.

Shembull Kodi optimal për **6** simbole elementare $\mathbf{x_i}$, të cilat përdoren me gjasat $\mathbf{p(x_i)}$, ashtu siç është dhënë në tabelën vijuese.

x_i	x_1	x_2	x_3	x_4	x_5	x_6
$p(x_i)$	0.20	0.15	0.10	0.35	0.08	0.12

x_i	$p(x_i)$	Shifrat e grupeve		
x_4	0.35	0	0	
x_1	0.20	0	1	
x_2	0.15	1	0	0
x_6	0.12	1	0	1
x_3	0.10	1	1	0
x_5	0.08	1	1	1

x_1	01
x_2	100
x_3	110
x_4	00
x_5	111
x_6	101

Për gjetjen e frekuencave f_i të paraqitjes së simboleve elementare x_i, të cilët kodohen, bëhen matje në një interval të caktuar kohor dhe numërohen paraqitjet e tyre. Pastaj, duke pjesëtuar këto frekuenca me numrin total F të paraqitjes së simboleve elementare, gjenden frekuencat relative, përkatësisht gjasat:

$$p(x_i)=f_i/F$$

Shembull

Kodi optimal për simbolet elementare k_i, të cilat në një interval të caktuar kohor janë paraqitur me frekuencat f_i, ashtu siç është dhënë në tabelën vijuese.

k_i	f_i
k_1	153
k_2	237
k_3	23
k_4	74
k_5	192

Numri total i simboleve elementare është:

$$F=\sum_{i=1}^{5} f_i = 153+237+23+74+192=679$$

Frekuencat relative, përkatësisht gjasat e paraqitjes së simboleve elementare, janë:

$$p(k_1) = \frac{153}{679} = 0.225$$

$$p(k_2) = \frac{237}{679} = 0.349$$

$$p(k_3) = \frac{23}{679} = 0.034$$

$$p(k_4) = \frac{74}{679} = 0.109$$

$$p(k_5) = \frac{192}{679} = 0.283$$

Gjasat e llogaritura mund të kontrollohen nëse e kemi parasysh faktin se shuma e të gjitha gjasave duhet të jetë:

$$\sum_{i=1}^{5} p(k_i) = 1$$

Grupimet dhe fjalët kodike për kodin optimal janë:

k_i	$p(k_i)$	Shifrat e grupeve		
k_2	0.349	0	0	
k_5	0.283	0	1	
k_1	0.225	1	0	
k_4	0.109	1	1	0
k_3	0.034	1	1	1

k_1	10
k_2	00
k_3	111
k_4	110
k_5	01

Nëse gjatë grupimit në dukje kemi më shumë mundësi, si optimale duhet të merret grupimi tek i cili diferenca absolute e shumave të gjasave të dy grupeve është minimale. P.sh., gjatë grupimit të parë, te shembulli i mësipërm, mund të fitohen këto dy raste:

Rasti	Shuma e grupit të sipërm S_1	Shuma e grupit të poshtëm S_2	Diferenca absolute $\lvert S_1 - S_2 \rvert$
a	$p(k_2) + p(k_5) = 0.632$	$p(k_1) + p(k_4) + p(k_3) = 0.368$	$\lvert 0.632 - 0.368 \rvert = 0.246$
b	$p(k_2) = 0.349$	$P(k_5) + p(k_1) + p(k_4) + p(k_3) = 0.651$	$\lvert 0.349 - 0.651 \rvert = 0.302$

Nga kjo shihet se diferenca më e vogël fitohet në rastin *a*, prandaj edhe si zgjidhje është marrë ky grupim.

Gjasat e paraqitjes së simboleve elementare që kodohen mund të kenë një ligjshmëri të caktuar.

Shembull

Kodi optimal për **6** simbole elementare të cilat paraqiten me gjasat:

$$p(x_i) = 2^{-i} \quad \text{për} \quad i = 1,2,3,4,5$$
$$p(x_6) = p(x_5)$$

x_i	$p(x_i)$	Shifrat e grupeve				
x_1	$\frac{1}{2}$	0				
x_2	$\frac{1}{4}$	1	0			
x_3	$\frac{1}{8}$	1	1	0		
x_4	$\frac{1}{16}$	1	1	1	0	
x_5	$\frac{1}{32}$	1	1	1	1	0
x_6	$\frac{1}{32}$	1	1	1	1	1

x_1	0
x_2	10
x_3	110
x_4	1110
x_5	11110
x_6	11111

Duke i shfrytëzuar fjalët kodike të një kodi optimal, mund të bëhet kodimi i vargut të simboleve që përpunohen ose transmetohen.

Shembull

Kodimi i vargut të informatave:

$$x_2 x_6 x_1 x_3 x_3 x_5 x_4 x_1$$

nëse shfrytëzohet kodi optimal i krijuar në shembullin paraprak.

101111101101101111011100

Nga vargu i dhënë i informatave të koduara sipas një kodi optimal mund të gjendet vargu i informatave elementare, duke u nisur prej fillimit të vargut dhe krahasuar vargjet e shifrave binare me fjalët kodike të cilat i përkasin kodit.

Shembull

Vargu i informatave elementare, për vargun e shifrave binare:

1101111001001111101110

nëse dihet se është shfrytëzuar kodi optimal i përcaktuar në shembullin e parafundit.

110 11110 0 10 0 11111 0 1110

Prej këtu mund të nxirret vargu përkatës i informatave elementare:

$X_3X_5X_1X_2X_1X_6X_1X_4$

Metoda e Huffman-it

Kodimi optimal sipas *metodës së Huffman-it* mbështetet në procedurën vijuese:

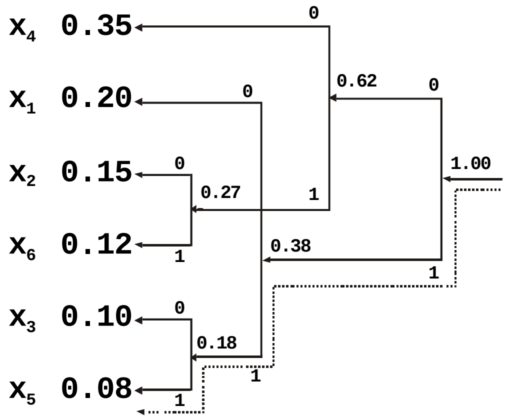
1. Radhiten simbolet elementare, në bazë të madhësive të gjasave përkatëse.
2. Grupohen dy simbolet elementare me gjasa më të vogla dhe njërit i shoqërohet shifra binare **0**, kurse tjetrit shifra binare **1**. P.sh., mund të përvetësohet parimi se shifra **0** i shoqërohet simbolit me gjasë më të madhe dhe ky parim duhet të ruhet deri në fund të procedurës së kodimit.
3. Grupimi vazhdon duke futur në proces edhe shumat e gjasave të simboleve që janë grupuar paraprakisht, derisa nuk bëhet grupimi i plotë.

Për gjetjen e fjalëve kodike, duke shkuar prej fundit të strukturës së krijuar gjatë grupimit, në drejtim të simboleve elementare, përshkruhen shifrat binare të cilat takohen gjatë rrugës. Grupet e tilla të shifrave binare paraqesin fjalët kodike të simboleve elementare që kodohen.

Shembull

Kodi optimal për **6** simbolet elementare x_i, të cilat përdoren me gjasat $p(x_i)$, të dhëna në tabelën vijuese.

x_i	x_1	x_2	x_3	x_4	x_5	x_6
$p(x_i)$	0.20	0.15	0.10	0.35	0.08	0.12



x_1	10
x_2	010
x_3	110
x_4	00
x_5	111
x_6	011

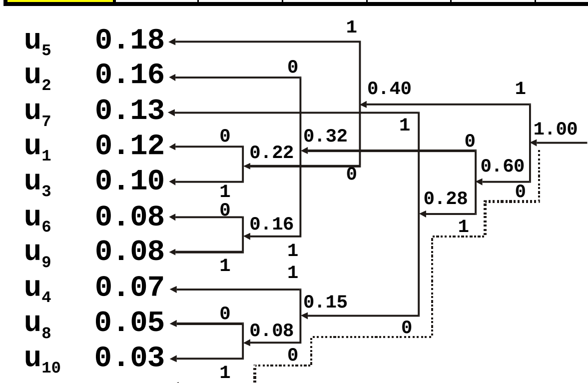
Nga shembulli i dhënë mund të shihet se radhitja fillestare e simboleve elemenentare nuk është e domosdoshme.

Nëse ndodh që dy simbole elementare të kenë gjasa të njëjta, ose shuma e gjasave të simboleve që grupohen të barazohet me ndonjë gjasë ekzistuese, gjatë grupimit nuk ka rëndësi se cila prej tyre formalisht nënkuptohet si më e madhe.

Shembull

Kodimi i simboleve elementare u_i, gjasat e paraqitjes së të cilave janë dhënë në tabelën vijuese.

u_i	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	u_9	u_{10}
$p(u_i)$	0.12	0.16	0.10	0.07	0.18	0.08	0.13	0.05	0.08	0.03



u_1	100
u_2	000
u_3	101
u_4	0101
u_5	11
u_6	0010
u_7	011
u_8	01000
u_9	0011
u_{10}	01001

Kodi optimal për alfabetin e gjuhës shqipe

Për gjetjen e kodit optimal për simbolet e alfabetit të gjuhës shqipe patjetër nevojiten gjasat e paraqitjes së tyre. Radha e paraqitjes së shkronjave të alfabetit shqip në bazë të frekuencave, përkatësisht gjasave të shfrytëzimit të tyre, pas përpunimit të teksteve të **12** autorëve, me madhësi përafërsisht të njëjtë dhe gjithsej **320014** shkronjave, është dhënë në tabelën e Fig.2.11.

Nr	Shkronja	Frekuenca	Gjasa
1	Zbrazësira	58896	0.1939531
2	E	25608	0.0843308
3	Ë	24907	0.0820223
4	T	21095	0.0694689
5	I	19202	0.0632349
6	A	18475	0.0608408
7	R	15650	0.0515377
8	N	15167	0.0499471
9	U	10128	0.0333529
10	K	9119	0.0300301
11	O	8876	0.0292299
12	M	8812	0.0290192
13	S	7692	0.0253308
14	P	6911	0.0227589
15	SH	6336	0.0208653
16	D	5697	0.0187610
17	J	5530	0.0182110
18	L	3906	0.0128630
19	V	3396	0.0111835
20	B	2983	0.0098234
21	DH	2953	0.0097246
22	H	2679	0.0088223
23	Q	2518	0.0082921
24	F	2415	0.0079529
25	G	2328	0.0076664
26	Y	1835	0.0060429
27	Z	1766	0.0058156
28	NJ	1567	0.0051603
29	GJ	1548	0.0050977
30	LL	1423	0.0046861
31	RR	1231	0.0040538
32	TH	1005	0.0033096
33	C	828	0.0027267
34	Ç	737	0.0024270
35	X	152	0.0005005
36	ZH	151	0.0004972
37	XH	139	0.0004577

Fig.2.11 Radha e paraqitjes së shkronjave të alfabetit shqip

Nga tabela e dhënë shihet se për nga frekuenca e përdorimit prijnë shkronjat **E** dhe **Ë**, kurse pas tyre vjen shkronja **T** dhe zanoret **I** e **A**. Më rrallë përdoren shkronjat në fund të tabelës: **Y**, **C**, **Ç** dhe **X**.

Duke pasur parasysh faktin se disa shkronja të alfabetit shqip shkruhen si kombinim i dy shkronjave të tjera, për kodim të simboleve të alfabetit duhet të përpilohet tabela përkatëse, me gjasat e paraqitjes së tyre. Kjo tabelë është dhënë në Fig.2.12.

Nr	Shkronja	Frekuenca	Gjasa
1	Zbrazësira	58896	0.1840419
2	E	25608	0.0800214
3	Ë	24907	0.0778309
4	T	22100	0.0690594
5	I	19202	0.0600036
6	A	18475	0.0577318
7	R	18112	0.0565975
8	N	16734	0.0522914
9	S	14028	0.0438355
10	H	13263	0.0414450
11	U	10128	0.0316486
12	K	9119	0.0284956
13	O	8876	0.0277362
14	M	8812	0.0275362
15	D	8650	0.0270300
16	J	8645	0.0270144
17	P	6911	0.0215959
18	L	6752	0.0210990
19	G	3876	0.0121119
20	V	3396	0.0106120
21	B	2983	0.0093214
22	Q	2518	0.0078684
23	F	2415	0.0075465
24	Z	1917	0.0059903
25	Y	1835	0.0057341
26	C	828	0.0025873
27	Ç	737	0.0023030
28	X	291	0.0009093

Fig.2.12 Radha e paraqitjes së simboleve të alfabetit shqip

Në bazë të gjasave të përdorimit, është bërë kodimi optimal i simboleve të alfabetit shqip, duke e shfrytëzuar *metodën e Huffman-it*, ashtu siç është treguar në Fig.2.13.

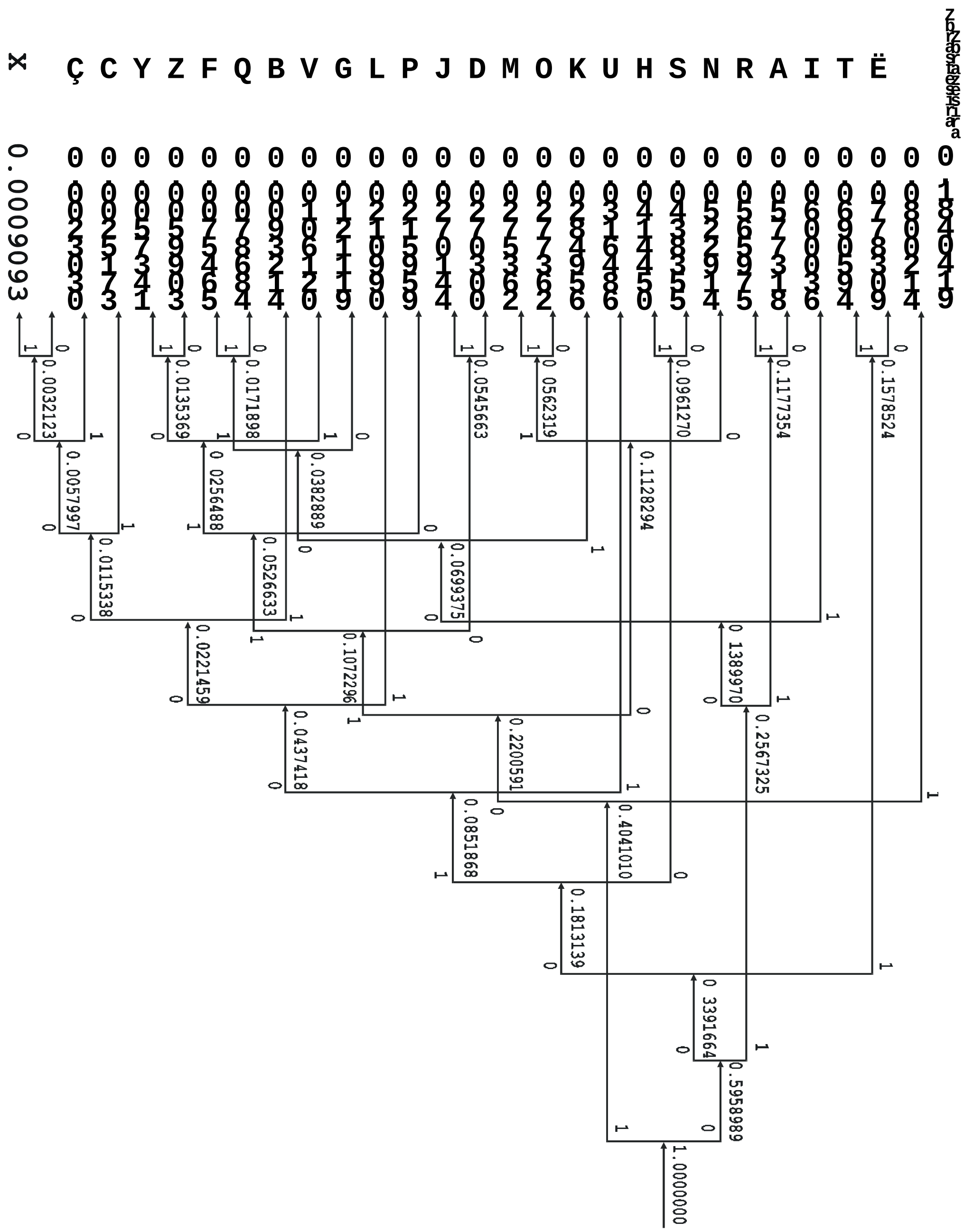


Fig.2.13 Procedura e gjetjes së kodit optimal të simboleve të alfabetit shqip

Prej këtu pastaj, duke e zbatuar parimin e shkuarjes drejt simboleve të veçanta, janë gjetur fjalët kodike, përkatësisht është krijuar kodi optimal për simbolet e alfabetit shqip, i cili është dhënë në tabelën e Fig.2.14.

Nr.	Shkronja	Kodi optimal
1	Zbrazësira	11
2	E	0010
3	Ë	0011
4	T	0101
5	I	0110
6	A	0111
7	R	1000
8	N	00000
9	S	00001
10	H	00011
11	U	01001
12	K	10010
13	O	10011
14	M	10100
15	D	10101
16	J	10110
17	P	000101
18	L	010000
19	G	101111
20	V	0001001
21	B	0100010
22	Q	0100011
23	F	1011100
24	Z	1011101
25	Y	00010001
26	C	000100001
27	Ç	0001000000
28	X	0001000001

Fig.2.14 Kodi optimal për simbolet e alfabetit shqip

Kodet siguruese

Për shkak të pengesave dhe të prishjeve në pajisje, ka gjasa të ndodhin gabime gjatë përpunimit ose transmetimit të informatave. Kodet siguruese përdoren me qëllim të sigurimit të informatave, përkatësisht zbulimit, ose edhe korrigjimit të gabimeve, gjë që varet nga rezerva kodike, ose nga distanca që ekziston në mes të fjalëve të veçanta kodike.

Në vazhdim do të flitet për kodet që mundësojnë zbulimin e gabimeve të njëfishta, ose zbulimin dhe korrigjimin e tyre.

Distanca në mes të fjalëve kodike

Në mes të fjalëve kodike $\mathbf{A = a_1 a_2 \ . \ . \ . \ a_n}$ dhe $\mathbf{B = b_1 b_2 \ldots b_n}$ të një kodi mund të llogaritet distanca kodike përmes shprehjes:

$$\begin{aligned} \mathbf{d(A,B)} &= \sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{(a_i \oplus b_i)} \\ &= \mathbf{(a_1 \oplus b_1)} + \mathbf{(a_2 \oplus b_2)} + \mathrm{K} + \mathbf{(a_n \oplus b_n)} \end{aligned}$$

ku me operatorin $\oplus$ duhet nënkuptuar:

$$\begin{aligned} \mathbf{0 \oplus 0} &= \mathbf{0} \\ \mathbf{0 \oplus 1} &= \mathbf{1} \\ \mathbf{1 \oplus 0} &= \mathbf{1} \\ \mathbf{1 \oplus 1} &= \mathbf{0} \end{aligned}$$

Shembull	Distanca në mes të fjalëve kodike të shifrave oktale **5** dhe **6**, të koduara në kodin i cili është dhënë në tabelën e Fig.2.15.

Shifra oktale	Fjala kodike
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Fig.2.15

$$\begin{aligned} d(5,6) &= d(101,110) \\ &= (1\oplus 1)+(0\oplus 1)+(1\oplus 0) \\ &= \quad 0 \quad + \quad 1 \quad + \quad 1 \quad = 2 \end{aligned}$$

Me distancën në mes të dy fjalëve kodike duhet nënkuptuar numrin e shifrave binare brenda fjalëve kodike të cilat duhet ndryshuar, për të kaluar prej njërës në fjalën tjetër. Kjo shihet edhe në shembullin e dy fjalëve kodike që u përmendën më sipër, të cilat në mes vete dallohen në dy shifra.

Distanca në mes të fjalëve kodike më së miri shihet nëse vizatohet shpërndarja hapësinore e fjalëve kodike, ku distanca është e barabartë me numrin minimal të brinjëve nëpër të cilët duhet kaluar, për të shkuar prej njërës në fjalën kodike tjetër. P.sh., për kodin e përmendur në shembullin e mësipërm, paraqitja hapësinore e fjalëve kodike duket si në Fig.2.16.

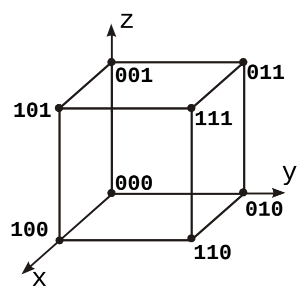


Fig.2.16 Paraqitja hapësinore e fjalëve të kodit trebitësh

Këtu vërehet se distanca në mes të fjalëve kodike **5** dhe **6** është **2**, sepse së paku duhet kaluar nëpër dy brinjë për të shkuar prej njërës në fjalën kodike tjetër.

Rezerva kodike

Kodi ka një rezervë kodike, nëse nuk shfrytëzohen të gjitha fjalët kodike të mundshme. Rezerva kodike përcaktohet përmes shprehjes:

R=Ln(m)/Ln(p)

ku janë:

m - numri i fjalëve kodike të mundëshme.
p - numri i fjalëve kodike që përdoren.

P.sh., te kodet **BCD** shfrytëzohen vetëm **10** fjalë kodike, edhe pse janë të mundshme gjithsej **16** fjalë kodike. Rezerva kodike përkatëse është:

R=Ln(16)/Ln(10)=1.20412

Kodet për zbulimin e gabimeve

Gabimet mund të zbulohen nëse ekziston një rezervë, përkatësisht nëse distanca në mes të fjalëve kodike është e mjaftueshme. P.sh., te kodi **NBCD** nuk ekziston rezervë e mjaftueshme kodike e cila do të mundësojë zbulimin e gabimeve të njëfishta. Kështu, nëse në vend të fjalës kodike **0110** gabimisht merret fjala kodike **0111**, e pamundshme është të detektohet gabimi, sepse që të dy fjalët kodike i takojnë kodit **NBCD**.

Në tabelën e Fig.2.17 janë dhënë disa kode përmes së cilave mund të zbulohen gabimet e njëfishta, sepse rezerva kodike e tyre:

$$\mathbf{R=Ln(2^5)/Ln(10)=1.5051}$$

është e mjaftueshme për t'i zbuluar ato.

Te kodet **2 prej 5** dhe **3 prej 5**, *gabimet e njëfishta* detektohen lehtë, sepse ndryshon numri i njëshave në fjalët kodike përkatëse.

Shifrat decimale	2 prej 5	3 prej 5	Kodi NBCD	
			Paritet çift	Paritet tek
0	11000	00111	0 0000	1 0000
1	00011	11100	1 0001	0 0001
2	00101	11010	1 0010	0 0010
3	00110	11001	0 0011	1 0011
4	01001	10110	1 0100	0 0100
5	01010	10101	0 0101	1 0101
6	01100	10011	0 0110	1 0110
7	10001	01110	1 0111	0 0111
8	10010	01101	1 1000	0 1000
9	10100	01011	0 1001	1 1001

Fig.2.17 Shembuj të kodeve për zbulimin e gabimeve

Në dy kolonat e fundit të tabelës është dhënë kodi **NBCD**, por tek i cili, çdo fjale kodike i është shtuar edhe një shifër për *paritet çift* (kolona e parafundit), ose *paritet tek* (kolona e fundit). Pariteti quhet çift, sepse shifrat e para të fjalëve kodike zgjidhen ashtu që numri i njësheve të bëhet, ose të mbetet, çift. Në rastin e paritetit tek shifrat e para në fjalët kodike zgjidhen ashtu që numri i njësheve të bëhet, ose të mbetet, tek.

Shembull

Zbulimi i fjalëve kodike te të cilat kanë ndodhur gabime të njëfishta, te vargu i shifrave binare:

0001100001010110010110110

nëse vargu është koduar në kodin:

a. **2 prej 5**.
b. **NBCD**, me paritet çift.

a.

00011	**00001**	**01011**	**00101**	**10110**
1	**2**	**3**	**4**	**5**

Gabime kanë ndodhur në fjalët kodike me numra rendorë **2**, **3** dhe **5**, sepse te këto fjalë numri i njësheve nuk është **2**, përkatësisht ato nuk gjenden në mesin e fjalëve kodike të dhëna në kolonën e dytë të tabelës në Fig.2.17.

b.

00011	**00001**	**01011**	**00101**	**10110**
1	**2**	**3**	**4**	**5**

Edhe këtu gabimet kanë ndodhur te fjalët kodike me numra rendor **2**, **3** dhe **5**, sepse ato nuk gjenden në mesin e fjalëve kodike të dhëna në kolonën e parafundit të tabelës. Te fjala kodike e **2**-të dhe e **3**-të, shifra e parë, e cila e tregon paritetin e njësheve, është **0**, kurse në mesin e shifrave të tjera ka numër tek njëshesh. Te fjala kodike e **5**-të, shifra për paritet është **1**, kurse te shifrat e tjera ka numër çift njëshesh.

Kodet për korrigjimin e gabimeve

Korrigjimi i gabimeve mund të bëhet nëse distanca në mes të fjalëve kodike të një kodi është e mjaftueshme. Për ta treguar këtë, p.sh., le ta marrim kodin me dy fjalë kodike trebitëshe: **000** dhe **111**, paraqitja hapësinore e së cilave është dhënë në Fig.2.18.

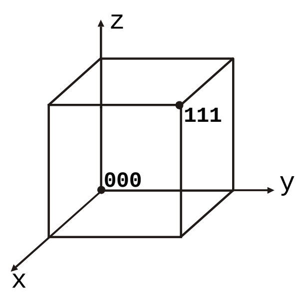


Fig.2.18 Paraqitja hapësinore e dy fjalëve kodike të kodit trebitësh

Nga paraqitja grafike shihet qartë se distanca në mes të dy fjalëve kodike është **3**. Kur ndodhë gabimi i njëfishtë te fjala kodike **000**, fitohet njëra nga tri fjalët kodike: **100**, **010** dhe **001**. Kurse, nëse gabohet fjala kodike **111**, fjalët e gabuara kodike do të jenë: **011**, **101** dhe **110**. Prej këtu dhe nga paraqitja grafike shihet qartë se të gjashtë fjalët e gabuara kodike janë të ndryshme, gjë që d.m.th. se gabimet e njëfishta mund të detektohen. Por, njëkohësisht, gabimet e tilla mund edhe të korrigjohen, meqë distanca në mes të fjalëve kodike të gabuara dhe fjalës kodike të saktë është **1**, kurse distanca nga fjala kodike tjetër e saktë është **2**.

Në vazhdim do të jepet një version i *kodit të Hamming-ut*, përmes të cilit mund të zbulohen dhe të korrigjohen gabimet e njëfishta. Kodimi sipas këtij kodi nuk qëndron në krijimin e fjalëve kodike me veti të veçanta, por në shtimin e shifrave kontrolluese brenda vargut të informatave që përpunohen, përkatësisht që transmetohen.

Fillimisht, vargu i informatave copëtohet në blloqe me nga **m**-shifra binare. Pastaj, brenda çdo blloku shtohen **k**-shifra kontrolluese, për kontrollim të paritetit, ashtu që të plotësohet kushti:

$$2^k \geq m+k+1$$

prej të cilit mund të nxirren vlerat:

k=	1	2	3	4	5	...
m≤	0	1	4	11	26	...

Procedura e kodimit në *kodin e Hamming-ut* si dhe detektimi e korrigjimi i gabimeve në vazhdim do të tregohen përmes shembujve.

Shembull

Kodimi i sigurt i vargut të shifrave decimale:

437653 **(X)**

nëse për kodimin e shifrave decimale shfrytëzohet kodi **NBCD**.

Pasi të kodohen shifrat decimale përmes kodit **NBCD**, vargu i informatave **(X)** duket kështu:

010000110111011001010011 **(Y)**

Nëse merret kushti se brenda **7** shifrave binare të cilat transmetohen, mund të ndodhë më së shumti një gabim, vargu **(Y)** duhet të copëtohet në blloqe me nga **m=4** shifra:

0100	**0011**	**0111**	**0110**	**0101**	**0011**		**(Z)**
a	**b**	**c**	**d**	**e**	**f**	Blloqet	

Meqë nga tabela e dhënë më sipër, për **m=4**, në çdo bllok duhet shtuar edhe **k=3** shifra kontrolluese, përkatësisht madhësia e blloqeve pas shtimit të shifrave kontrolluese është **m+k=7**.

Shtimi i shifrave kontrolluese përcaktohet në bazë të ligjshmërisë:

$$1,\ 2,\ 4,\ \ldots,\ 2^{k-1}$$

prej nga për **k=3** del se shifrat kontrolluese brenda çdo blloku të informatave duhet të shtohen në pozicionet **1**, **2** dhe **4**, përkatësisht blloqet do të duken kështu:

1	2	3	4	5	6	7
k_1	k_2	i_1	k_3	i_2	i_3	i_4

ku numrat e shkruar në rreshtin e parë quhen *numra pozicionalë*. Ekuivalentët binarë të numrave pozicionalë:

001 010 011 100 101 110 111

do të përdoren për ta treguar vendin e gabimit në bllok. Në bazë të pozitës së shifrave **1** (në fund, në mes ose në fillim) numrat pozicionalë mund të grupohen në tri grupe:

a.	**001**	**011**	**101**	**111**
	1	**3**	**5**	**7**
b.	**010**	**011**	**110**	**111**
	2	**3**	**6**	**7**
c.	**100**	**101**	**110**	**111**
	4	**5**	**6**	**7**

prej nga edhe është marrë që shifrat kontrolluese të përcaktohen në bazë të **3** kontrolleve të paritetit:

1-3-5-7
2-3-6-7
4-5-6-7

përkatësisht përmes shprehjeve:

$$k_1 = i_1 \oplus i_2 \oplus i_4$$
$$k_2 = i_1 \oplus i_3 \oplus i_4$$
$$k_3 = i_2 \oplus i_3 \oplus i_4$$

Procedura e shtimit të shifrave kontrolluese është dhënë përmes tabelës në Fig.2.19.

	1	2	3	4	5	6	7
	k_1	k_2	i_1	k_3	i_2	i_3	i_4
Blloku a			0		1	0	0
Kontrolli 1-3-5-7	1		0		1		0
Kontrolli 2-3-6-7		0	0			0	0
Kontrolli 4-5-6-7				1	1	0	0
Blloku a'	1	0	0	1	1	0	0

$$k_1 = 0 \oplus 1 \oplus 0 = 1$$
$$k_2 = 0 \oplus 0 \oplus 0 = 0$$
$$k_3 = 1 \oplus 0 \oplus 0 = 1$$

	1	2	3	4	5	6	7
Blloku b			0		0	1	1
Kontrolli 1-3-5-7	1		0		0		1
Kontrolli 2-3-6-7		0	0			1	1
Kontrolli 4-5-6-7				0	0	1	1
Blloku b'	1	0	0	0	0	1	1

$$k_1 = 0 \oplus 0 \oplus 1 = 1$$
$$k_2 = 0 \oplus 1 \oplus 1 = 0$$
$$k_3 = 0 \oplus 1 \oplus 1 = 0$$

	1	2	3	4	5	6	7
Blloku c			0		1	1	1
Kontrolli 1-3-5-7	0		0		1		1
Kontrolli 2-3-6-7		0	0			1	1
Kontrolli 4-5-6-7				1	1	1	1
Blloku c'	0	0	0	1	1	1	1

$$k_1 = 0 \oplus 1 \oplus 1 = 0$$
$$k_2 = 0 \oplus 1 \oplus 1 = 0$$
$$k_3 = 1 \oplus 1 \oplus 1 = 1$$

	1	2	3	4	5	6	7
Blloku d			0		1	1	0
Kontrolli 1-3-5-7	1		0		1		0
Kontrolli 2-3-6-7		1	0			1	0
Kontrolli 4-5-6-7				0	1	1	0
Blloku d'	1	1	0	0	1	1	0

$$k_1 = 0 \oplus 1 \oplus 0 = 1$$
$$k_2 = 0 \oplus 1 \oplus 0 = 1$$
$$k_3 = 1 \oplus 1 \oplus 0 = 0$$

	1	2	3	4	5	6	7
Blloku e			0		1	0	1
Kontrolli 1-3-5-7	0		0		1		1
Kontrolli 2-3-6-7		1	0			0	1
Kontrolli 4-5-6-7				0	1	0	1
Blloku e'	0	1	0	0	1	0	1

$$k_1 = 0 \oplus 1 \oplus 1 = 0$$
$$k_2 = 0 \oplus 0 \oplus 1 = 1$$
$$k_3 = 1 \oplus 0 \oplus 1 = 0$$

	1	2	3	4	5	6	7
Blloku f			0		0	1	1
Kontrolli 1-3-5-7	1		0		0		1
Kontrolli 2-3-6-7		0	0			1	1
Kontrolli 4-5-6-7				0	0	1	1
Blloku f'	1	0	0	0	0	1	1

$$k_1 = 0 \oplus 0 \oplus 1 = 1$$
$$k_2 = 0 \oplus 1 \oplus 1 = 0$$
$$k_3 = 0 \oplus 1 \oplus 1 = 0$$

Fig.2.19 Shtimit i shifrave kontrolluese sipas metodës së Hamming-ut

Pas përfundimit të procedurës së kodimit sipas *kodit të Hamming-ut*, vargu i ri do të plotësohet edhe me shifrat kontrolluese dhe përfundimisht duket kështu:

```
1001100 1000011 0001111 1100110 0100101 1000011 (Z')
   a'      b'      c'      d'      e'      f'
```

Vargu i shifrave binare që u fitua me procedurën e dhënë, brenda të cilit gjenden informatat dhe shifrat kontrolluese, është i gatshëm për transmetim. Meqë distanca në mes të fjalëve kodike është **3**, ajo mundëson zbulimin dhe korrigjimin e gabimeve të njëfishta.

Pasi të përfundojë procesi i transmetimit, blloqet e veçanta të vargut të informatave kontrollohen, duke kontrolluar paritetin përmes shprehjeve:

$$k_1 \oplus i_1 \oplus i_2 \oplus i_4 = A$$
$$k_2 \oplus i_1 \oplus i_3 \oplus i_4 = B$$
$$k_3 \oplus i_2 \oplus i_3 \oplus i_4 = C$$

Numri pozicional $g=(CBA)_2$ e përcakton vendin e gabimit brenda bllokut. Nëse gjatë kontrollimit fitohet vlera $g=(000)_2$, kjo d.m.th. se blloku është pranuar pa gabim.

Shembull Gjetja dhe korrigjimi i gabimeve brenda vargut të informatave (**Z"**) të dhëna më poshtë.

```
1011100 1000011 0001101 1101110 0100101 1000001 (Z")
   a"      b"      c"      d"      e"      f"
```

Kontrollimi i blloqeve të veçanta dhe korrigjimi i gabimeve është bërë duke shfrytëzuar tabelën e dhënë në Fig.2.20.

	1	2	3	4	5	6	7

	k_1	k_2	i_1	k_3	i_2	i_3	i_4	
Blloku a"	1	0	1	1	1	0	0	
Kontrolli 1-3-5-7	1		1		1		0	$1\oplus1\oplus1\oplus0=1$
Kontrolli 2-3-6-7		0	1			0	0	$0\oplus1\oplus0\oplus0=1$
Kontrolli 4-5-6-7				1	1	0	0	$1\oplus1\oplus0\oplus0=0$
Blloku a'	1	0	0	1	1	0	0	$G=(011)_2=(3)_{10}$
Blloku b"	1	0	0	0	0	1	1	
Kontrolli 1-3-5-7	1		0		0		1	$1\oplus0\oplus0\oplus1=0$
Kontrolli 2-3-6-7		0	0			1	1	$0\oplus0\oplus1\oplus1=0$
Kontrolli 4-5-6-7				0	0	1	1	$0\oplus0\oplus1\oplus1=0$
Blloku b'	1	0	0	0	0	1	1	$G=(000)_2=(0)_{10}$
Blloku c"	0	0	0	1	1	0	1	
Kontrolli 1-3-5-7	0		0		1		1	$0\oplus0\oplus1\oplus1=0$
Kontrolli 2-3-6-7		0	0			0	1	$0\oplus0\oplus0\oplus1=1$
Kontrolli 4-5-6-7				1	1	0	1	$1\oplus1\oplus0\oplus1=1$
Blloku c'	0	0	0	1	1	1	1	$G=(110)_2=(6)_{10}$
Blloku d"	1	1	0	1	1	1	0	
Kontrolli 1-3-5-7	1		0		1		0	$1\oplus0\oplus1\oplus0=0$
Kontrolli 2-3-6-7		1	0			1	0	$1\oplus0\oplus1\oplus0=0$
Kontrolli 4-5-6-7				1	1	1	0	$1\oplus1\oplus1\oplus0=1$
Blloku d'	1	1	0	0	1	1	0	$G=(100)_2=(4)_{10}$
Blloku e"	0	1	0	0	1	0	1	
Kontrolli 1-3-5-7	0		0		1		1	$0\oplus0\oplus1\oplus1=0$
Kontrolli 2-3-6-7		1	0			0	1	$1\oplus0\oplus0\oplus1=0$
Kontrolli 4-5-6-7				0	1	0	1	$0\oplus1\oplus0\oplus1=0$
Blloku e'	0	1	0	0	1	0	1	$G=(000)_2=(0)_{10}$
Blloku f"	1	0	0	0	0	0	1	
Kontrolli 1-3-5-7	1		0		0		1	$1\oplus0\oplus0\oplus1=0$
Kontrolli 2-3-6-7		0	0			0	1	$0\oplus0\oplus0\oplus1=1$
Kontrolli 4-5-6-7				0	0	0	1	$0\oplus0\oplus0\oplus1=1$
Blloku f'	1	0	0	0	0	1	1	$G=(110)_2=(6)_{10}$

Fig.2.20 Kontrollimi i blloqeve përmes metodës së Hamming-ut

Vlerat të cilat në tabelë janë theksuar veçanërisht, janë vlerat e korrigjuara të blloqeve **a"**, **c"**, **d"** dhe **f"**, për shkak të gabimeve. Pas korrigjimeve fitohen blloqet me vlera të sakta, përkatësisht përsëri fitohet vargu **(Z')**, nga i cili, nëse eliminohen shifrat kontrolluese, fitohet vargu i informatave të cilat transmetohen **(Z)**, përkatësisht vargu **(X)**.

Procedura e kodimit sipas *kodit të Hamming-ut* përdoret plotësisht njëlloj edhe nëse kodimi fillestar i vargut të informatave bëhet përmes kodeve te të cilat fjalët kodike nuk kanë gjatësi fikse.

Shembull Kodimi sipas *kodit të Hamming-ut*, i vargut të simboleve:

$$X_3X_5X_1X_4X_3X_6X_1X_2$$

nëse për kodimin binar të simboleve përdoret kodi optimal:

X_1	**0**
X_2	**10**
X_3	**110**
X_4	**1110**
X_5	**11110**
X_6	**11111**

Pas kodimit binar përmes kodit optimal të dhënë, vargu i informatave do të duket kështu:

110 11110 0 1110 110 11111 0 10

përkatësisht:

110111100111011011111010

Procedura e kodimit sipas *kodit të Hamming-ut*, edhe këtu fillon me copëtimin e vargut të informatave në blloqe me nga **4** shifra (nëse kodimi bëhet nën kushte të njëjta, të përmendura më lart).

1101	**1110**	**0111**	**0110**	**1111**	**1010**
a	**b**	**c**	**d**	**e**	**f**

Pas shtimit të shifrave kontrolluese përmes procedurës të shpjeguar më sipër, vargu i shifrave binare i përgatitur për transmetim do të duket kështu:

1010101 0010110 0001111 1100110 1111111 1011010

Kodet alfanumerike

Kur informatat të cilat përpunohen përmbajnë shkronja, numra dhe simbole speciale, përdoren të ashtuquajturat *kode alfanumerike* (ang. alphanumeric code), ose edhe *kode alfamerike* (ang. alphameric code). Këto kode kryesisht kanë një gjatësi prej **6** deri në **8** simbole. Në tabelën e dhënë në Fig.2.21 shihen pjesë të dy kodeve kryesore alfanumerike: **ASCII** (nga American Standard Code for Information Interchange) dhe **EBCDIC** (nga Extended Binary Coded Decimal Interchange Code). Kodi **ASCII** (në praktikë thuhet **aski**) përdoret si kod te kompjuterët personalë, kurse kodi **EBCDIC** është kod të cilin e përdor kompania kompjuterike **IBM**.

Simboli	ASCII	EBCDIC
A	100 0001	1100 0001
B	100 0010	1100 0010
C	100 0011	1100 0011
D	100 0100	1100 0100
E	100 0101	1100 0101
F	100 0110	1100 0110
G	100 0111	1100 0111
H	100 1000	1100 1000
I	100 1001	1100 1001
J	100 1010	1101 0001
K	100 1011	1101 0010
L	100 1100	1101 0011
M	100 1101	1101 0100
N	100 1110	1101 0101
O	100 1111	1101 0110
P	101 0000	1101 0111
Q	101 0001	1101 1000
R	101 0010	1101 1001
S	101 0011	1110 0010
T	101 0100	1110 0011
U	101 0101	1110 0100
V	101 0110	1110 0101
W	101 0111	1110 0110
X	101 1000	1110 0111

Simboli	ASCII	EBCDIC
Y	101 1001	1110 1000
Z	101 1010	1110 1001
Zbrazësira	010 0000	0100 0000
.	010 1110	0100 1011
(	010 1000	0100 1101
+	010 1011	0100 1110
$	010 0100	0101 1011
*	010 1010	0101 1100
)	010 1001	0101 1101
-	010 1101	0110 0000
/	010 1111	0110 0001
,	010 1100	0110 1011
'	010 0111	0111 1101
"	010 0010	0111 1111
=	011 1101	0111 1110
0	011 0000	1111 0000
1	011 0001	1111 0001
2	011 0010	1111 0010
3	011 0011	1111 0011
4	011 0100	1111 0100
5	011 0101	1111 0101
6	011 0110	1111 0110
7	011 0111	1111 0111
8	011 1000	1111 1000
9	011 1001	1111 1001

*Fig.2.21 Kodet alfanumerike **ASCII** dhe **EBCDIC***

Në praktikë, fjalëve kodike të kodit **ASCII**, kryesisht u shtohet para edhe një shifër për kontrollim të paritetit.

Shembull	Kodimi i komandës **REPEAT**, e cila është shkruar në gjuhën Pascal, në kodin: *a.* **EBCDIC** *b.* **ASCII**, duke shtuar në çdo fjalë kodike edhe një shifër për paritet çift.

a.

```
RE      1101 1001 1100 0101
PE      1101 0111 1100 0101
AT      1100 0001 1110 0011
```

b.

```
RE      1101 0010 1100 0101
PE      0101 0000 1100 0101
AT      0100 0001 1101 0100
```

3 Algjebra e Bulit

Sistemet fizike dhe sistemet logjike te të cilat mund të paraqiten vetëm dy gjendje të ndryshme përshkruhen përmes *algjebrës së Bulit* (ang. Boolean Algebra). Të tilla janë, p.sh., sistemet logjike te të cilat paraqiten gjendjet:

çdo gjë dhe *asgjë*, ose *e vërtetë* dhe *e pavërtetë*,

ose sistemet digjitale me gjendjet e tensioneve:

i lartë dhe *i ulët*, ose *ekziston* dhe *nuk ekziston* tension.

Algjebra e Bulit e ka zanafillën që nga punimet e *Aristotelit*, në kohën antike. Por, bazat e kësaj algjebre janë përcaktuar nga *George Boole* (1815-1864), në vitin 1849, me punimet e tij për *proceset e të menduarit*.

Algjebra e Bulit për herë të parë zbatohet gjatë analizës së qarqeve me ndërprerës, në punimet e *Claude Shannon*, në vitin 1938, prej nga shpesh quhet edhe *algjebër e ndërprerësve* (ang. switching algebra).

Njohuri themelore

Algjebra e Bulit definohet si grumbull prej dy elementesh **{0,1}**, mbi të cilët mund të zbatohen **3** operacione themelore: *mbledhja*, *shumëzimi* dhe *komplementimi*. Për këto tri operacione përdoren operatorët logjikë **OR**, **AND** dhe **NOT**, ose në gjuhën shqipe operatorët përkatës: **OSE**, **DHE** dhe **JO**.

Në vend të operatorëve logjikë të dhënë më sipër, për operacionin e mbledhjes dhe të shumëzimit, në praktikë, përdoren edhe operatorët **+** e **·**, kurse vlerat e komplementuara në literaturë shënohen kryesisht me *një vizë mbi vlerën*.

Operacioni i mbledhjes dhe i shumëzimit paraqesin *operacione binare*, sepse zbatohen mbi dy vlera, kurse operacioni i komplementimit quhet *operacion unar*, sepse në të merr pjesë vetëm një vlerë.

Variablat të cilat mund të *marrin vetëm dy vlera logjike*, **0** dhe **1**, njihen si *variabla të Bulit* (ang. Boolean variable), ose *variabla logjike*. Kurse funksionet të cilat formohen si kombinim i variablave dhe i operatorëve të algjebrës së *Bulit* paraqesin *funksione të Bulit* (ang. Boolean function), ose *funksione logjike*.

Postulatet

Algjebra e Bulit mbështetet në një grumbull *qëndrimesh themelore*, të cilat ndryshe quhen *postulate*. Duke i përdorur tri operacionet themelore, këto postulate shkruhen kështu:

Operacioni ***OSE***

$$0 + 0 = 0$$
$$0 + 1 = 1$$
$$1 + 0 = 1$$
$$1 + 1 = 1$$

Operacioni ***DHE***

$$0 \cdot 0 = 0$$
$$0 \cdot 1 = 0$$
$$1 \cdot 0 = 0$$
$$1 \cdot 1 = 1$$

Operacioni ***JO***

$$\overline{0} = 1$$
$$\overline{1} = 0$$

ku me vizat mbi numra duhet nënkuptuar *vlerat e komplementuara*.

Në bazë të postulateve të dhëna më sipër, duke shfrytëzuar variablën logjike **A**, e cila mund t'i marrë dy vlerat logjike të mundshme, **0** dhe **1**, si dhe tri operacioneve themelore, nxirren edhe *relacionet algjebrike* vijuese.

$$A + 0 = A \qquad A \cdot 1 = A$$
$$A + 1 = 1 \qquad A \cdot 0 = 0$$
$$A + A = A \qquad A \cdot A = A \qquad \overline{\overline{A}} = A$$
$$A + \overline{A} = 1 \qquad A \cdot \overline{A} = 0$$

Këto relacione në praktikë vërtetohen shumë thjesht, nëse në vend të variablës **A** shkruhen dy vlerat e mundshme të saj. Kështu, p.sh., për relacionin

$$A + 0 = A$$

nëse variabla **A** zëvendësohet me dy vlerat e mundshme të saj, fitohen relacionet:

$$0 + 0 = 0$$
$$1 + 0 = 1$$

prej nga shihet se vlerat pas barazimit janë të njëjta me vlerat para operatorit **+**, përkatësisht me dy vlerat e mundshme të variablës **A**.

Ligjet

Disa prej ligjeve që përdoren në algjebrën e zakonshme mund të shkruhen edhe në *algjebrën e Bulit*. Në vazhdim janë dhënë ligjet themelore të kësaj algjebre përmes *barazimeve logjike* të cilat përmbajnë dy ose tri variabla logjike.

Ligji i komutacionit

$$A + B = B + A$$
$$A \cdot B = B \cdot A$$

Ligji i asociacionit

$$A + (B + C) = (A + B) + C$$
$$A \cdot (B \cdot C) = (A \cdot B) \cdot C$$

Ligji i distribucionit

$$A \cdot (B + C) = A \cdot B + A \cdot C$$
$$A + B \cdot C = (A + B) \cdot (A + C)$$

Ligji i absorbcionit

$$A + (A \cdot B) = A$$
$$A \cdot (A + B) = A$$

Ligji i ekspansionit

$$A \cdot B + A \cdot \overline{B} = A$$
$$(A + B) \cdot (A + \overline{B}) = A$$

Ligjet e dhëna mund të vërtetohen duke pasur parasysh postulatet dhe relacionet algjebrike të cilat u dhanë më parë.

Shembull

Vërtetimi i ligjeve:

a. $A(A+B)=A$

b. $AB + A\overline{B} = A$

c. $(A + B)(A + \overline{B}) = A$

a.

$$\begin{aligned} A(A+B) &= AA + AB \\ &= A + AB \\ &= A(1+B) \\ &= A \cdot 1 \\ &= A \end{aligned}$$

b.

$$\begin{aligned} AB + A\overline{B} &= A(B+\overline{B}) \\ &= A \cdot 1 \\ &= A \end{aligned}$$

c.

$$\begin{aligned} (A+B)(A+\overline{B}) &= AA + AB + A\overline{B} + B\overline{B} \\ &= A + AB + A\overline{B} + 0 \\ &= A + AB + A\overline{B} \\ &= A(1+B) + A\overline{B} \\ &= A \cdot 1 + A\overline{B} \\ &= A + A\overline{B} \\ &= A(1+\overline{B}) \\ &= A \cdot 1 \\ &= A \end{aligned}$$

Ligjet e dhëna vlejnë edhe nëse barazimet logjike formohen duke shfrytëzuar edhe më shumë variabla.

Teoremat e De Morgan-it

Gjetja e funksioneve komplementare mbështetet në të ashtuquajturat *teorema të De Morgan-it*, të cilat, për funksionet me dy variabla, shkruhen kështu:

$$\overline{A + B} = \overline{A} \cdot \overline{B}$$

$$\overline{A \cdot B} = \overline{A} + \overline{B}$$

Shembull

Gjetja e shprehjes komplementare të shprehjes:

$$A\overline{B} + \overline{A}B$$

duke i shfrytëzuar teoremat e *De Morgan-it*.

$$
\begin{aligned}
\overline{A\overline{B} + \overline{A}B} &= \left(\overline{A\overline{B}}\right) \cdot \left(\overline{\overline{A}B}\right) \\
&= \left(\overline{A} + \overline{\overline{B}}\right) \cdot \left(\overline{\overline{A}} + \overline{B}\right) \\
&= \left(\overline{A} + B\right) \cdot \left(A + \overline{B}\right) \\
&= \overline{A}A + AB + \overline{A}\,\overline{B} + B\overline{B} \\
&= 0 + AB + \overline{A}\,\overline{B} + 0 \\
&= AB + \overline{A}\,\overline{B}
\end{aligned}
$$

Për funksionet me më shumë variabla, teoremat e *De Morgan-it* shkruhen plotësisht njëlloj, gjë që do të shihet gjatë përdorimit të tyre, në pjesën vijuese të librit.

Identitete me rëndësi

Gjatë thjeshtësimit të shprehjeve të ndryshme logjike, përveç postulateve, ligjeve dhe teoremave të dhëna më sipër, shfrytëzohet edhe një numër identitetesh, gjë që lehtëson mjaft procesin e thjeshtësimit. Disa nga këto identitete janë:

$$A\left(\overline{A} + B\right) = AB$$

$$A + \overline{A}B = A + B$$

$$(AB)(A + B) = AB$$

$$\overline{(AB)}(A + B) = A\overline{B} + \overline{A}B$$

$$\overline{(A\overline{B}+\overline{A}B)}=AB+\overline{A}\,\overline{B}$$

$$(A+B)(B+C)(A+C)=AB+BC+AC$$

$$(A+B)(\overline{A}+C)=AC+\overline{A}B$$

$$AC+AB+B\overline{C}=AC+B\overline{C}$$

$$(A+B)(B+C)(\overline{A}+C)=(A+B)(\overline{A}+C)$$

Identitetet e dhëna mund të vërtetohen duke shfrytëzuar postulatet, ligjet dhe teoremat e dhëna më sipër.

Shembull

Vërtetimi i identiteteve:

a. $A(\overline{A}+B)=AB$

b. $\overline{(AB)}(A+B)=A\overline{B}+\overline{A}B$

c. $A+\overline{A}B=A+B$

d. $(A+B)(\overline{A}+\overline{B})=A\overline{B}+\overline{A}B$

e. $(A+B)(\overline{A}+C)=\overline{A}B+AC$

f. $AB+\overline{A}C+BC=AB+\overline{A}C$

duke i shfrytëzuar postulatet, ligjet dhe teoremat e algjebrës së Bulit.

a.

$$\begin{aligned}A(\overline{A}+B)&=A\overline{A}+AB\\&=0+AB\\&=AB\end{aligned}$$

b.

$$\begin{aligned}\overline{(AB)}(A+B)&=(\overline{A}+\overline{B})(A+B)\\&=A\overline{A}+A\overline{B}+\overline{A}B+B\overline{B}\\&=0+A\overline{B}+\overline{A}B+0\\&=A\overline{B}+\overline{A}B\end{aligned}$$

c.

$$
\begin{aligned}
\mathbf{A} + \overline{\mathbf{A}} \cdot \mathbf{B} &= \mathbf{A} \cdot \mathbf{1} + \overline{\mathbf{A}} \cdot \mathbf{B} \\
&= \mathbf{A}\,(\mathbf{B} + \overline{\mathbf{B}}) + \overline{\mathbf{A}}\mathbf{B} \\
&= \mathbf{AB} + \mathbf{A}\overline{\mathbf{B}} + \overline{\mathbf{A}}\mathbf{B} \\
&= (\mathbf{AB} + \mathbf{AB})(\mathbf{A}\overline{\mathbf{B}} + \overline{\mathbf{A}}\mathbf{B}) \\
&= \mathbf{AB} + \mathbf{A}\overline{\mathbf{B}} + \mathbf{AB} + \overline{\mathbf{A}}\mathbf{B} \\
&= \mathbf{A}\,(\mathbf{B} + \overline{\mathbf{B}}) + \mathbf{B}(\mathbf{A} + \overline{\mathbf{A}}) \\
&= \mathbf{A} \cdot \mathbf{1} + \mathbf{B} \cdot \mathbf{1} \\
&= \mathbf{A} + \mathbf{B}
\end{aligned}
$$

d.

$$
\begin{aligned}
(\mathbf{A} + \mathbf{B}) \cdot (\overline{\mathbf{A}} + \overline{\mathbf{B}}) &= \mathbf{A}\overline{\mathbf{A}} + \mathbf{A}\overline{\mathbf{B}} + \overline{\mathbf{A}}\mathbf{B} + \mathbf{B}\overline{\mathbf{B}} \\
&= \mathbf{0} + \mathbf{A}\overline{\mathbf{B}} + \overline{\mathbf{A}}\mathbf{B} + \mathbf{0} \\
&= \mathbf{A}\overline{\mathbf{B}} + \overline{\mathbf{A}}\mathbf{B}
\end{aligned}
$$

e.

$$
\begin{aligned}
(\mathbf{A} + \mathbf{B})(\overline{\mathbf{A}} + \mathbf{C}) &= \mathbf{A}\overline{\mathbf{A}} + \overline{\mathbf{A}}\mathbf{B} + \mathbf{AC} + \mathbf{BC} \\
&= \mathbf{0} + \overline{\mathbf{A}}\mathbf{B} + \mathbf{AC} + \mathbf{BC} \\
&= \overline{\mathbf{A}}\mathbf{B} + \mathbf{AC} + \mathbf{BC} \\
&= \overline{\mathbf{A}}\mathbf{B} \cdot \mathbf{1} + \mathbf{AC} \cdot \mathbf{1} + \mathbf{BC} \cdot \mathbf{1} \\
&= \overline{\mathbf{A}}\mathbf{B}(\mathbf{C} + \overline{\mathbf{C}}) + \mathbf{AC}(\mathbf{B} + \overline{\mathbf{B}}) + \mathbf{BC}(\mathbf{A} + \overline{\mathbf{A}}) \\
&= \overline{\mathbf{A}}\mathbf{BC} + \overline{\mathbf{A}}\mathbf{B}\overline{\mathbf{C}} + \mathbf{ABC} + \mathbf{A}\overline{\mathbf{B}}\mathbf{C} + \mathbf{ABC} + \overline{\mathbf{A}}\mathbf{BC} \\
&= \overline{\mathbf{A}}\mathbf{BC} + \overline{\mathbf{A}}\mathbf{B}\overline{\mathbf{C}} + \mathbf{ABC} + \mathbf{A}\overline{\mathbf{B}}\mathbf{C} \\
&= \overline{\mathbf{A}}\mathbf{B}(\mathbf{C} + \overline{\mathbf{C}}) + \mathbf{AC}(\mathbf{B} + \overline{\mathbf{B}}) \\
&= \overline{\mathbf{A}}\mathbf{B} \cdot \mathbf{1} + \mathbf{AC} \cdot \mathbf{1} \\
&= \overline{\mathbf{A}}\mathbf{B} + \mathbf{AC}
\end{aligned}
$$

f.

$$
\begin{aligned}
AB + \overline{A}C + BC &= AB + \overline{A}C + 1 \cdot BC \\
&= AB + \overline{A}C + (A + \overline{A})BC \\
&= AB + \overline{A}C + ABC + \overline{A}BC \\
&= AB(1 + C) + \overline{A}C(1 + B) \\
&= AB \cdot 1 + \overline{A}C \cdot 1 \\
&= AB + \overline{A}C
\end{aligned}
$$

Principi i dualitetit

Në algjebrën e Bulit vlen *principi i dualitetit* (ang. principle of duality), sipas të cilit, nëse në shprehjet logjike, të cilat gjenden në dy anët e një barazimi, zëvendësohen:

0 me **1**
1 me **0**
+ me **·**
· me **+**

fitohet *barazimi dual* përkatës. Ekuivalenca e shprehjeve në dy anët e barazimit nuk prishet, përkatësisht barazimi ruhet.

Operacionet duale

Operacionet logjike **OSE** e **DHE** mes vete janë *operacione duale* (ang. dual operation), sepse, nëse në tabelën e cila vlen për operacionin **OSE**:

0+0=0
0+1=1
1+0=1
1+1=1

zëvendësohen:

0 me **1**
1 me **0**
+ me **·**

fitohet tabela që i përgjigjet operacionit **DHE**:

1·1=1
1·0=0
0·1=0
0·0=0

Vlen edhe e kundërta, përkatësisht nëse në tabelën e operacionit **DHE** bëhen zëvendësimet:

0 me **1**
1 me **0**
· me **+**

do të fitohet tabela e cila i përgjigjet operacionit **OSE**.

Funksionet duale

Nëse në një funksion logjik zëvendësohen:

0 me **1**
1 me **0**
+ me **·**
· me **+**

kurse variablat përshkruhen, do të fitohet *funksioni dual* (ang. dual function) përkatës. Gjatë kësaj duhet pasur kujdes të veçantë sidomos në zëvendësimin e operatorit të mbledhjes me atë të shumëzimit, duke përdorur sipas nevojës edhe kllapa.

Shembull

Funksionet duale për funksionet logjike:

a. $\mathbf{f=(A+B)(A+\overline{B})}$

b. $\mathbf{z=A(1+B)(B+C)}$

c. $\mathbf{g=(AB)+(BC)+(CA)}$

d. $\mathbf{h=(A+\overline{D})B+(C+AD)B}$

a.

$$\mathbf{f=(A+B)\cdot(A+\overline{B})}$$
$$\mathbf{f_d=(A\cdot B)+(A\cdot\overline{B})}$$

b.

$$\mathbf{z=A\cdot(1+B)\cdot(B+C)}$$
$$\mathbf{z_d=A+(0\cdot B)+(B\cdot C)}$$

c.

$$\mathbf{g=(A\cdot B)+(B\cdot C)+(C\cdot A)}$$
$$\mathbf{g_d=(A+B)\cdot(B+C)\cdot(C+A)}$$

d.

$$\mathbf{h=\{(A+\overline{D})\cdot B\}+\{(C+A\cdot D)\cdot B\}}$$
$$\mathbf{h_d=\{(A\cdot\overline{D})+B\}\cdot\{(C\cdot[A+D])+B\}}$$

Barazimet duale mund të gjenden duke i gjetur funksionet duale të shprehjeve në dy anët e barazimeve, ashtu siç u dha më sipër.

Shembull

Gjetja e barazimeve duale për barazimet:

a. $\mathbf{A+1=1}$

b. $\mathbf{(A+B+C)(A+\overline{B}+C)=A+C}$

c. $\mathbf{(A+B)(B+C)(\overline{A}+C)=(A+B)(\overline{A}+C)}$

a.

$$\mathbf{A} + \mathbf{1} = \mathbf{1}$$
$$\mathbf{A} \cdot \mathbf{0} = \mathbf{0}$$

b.

$$(\mathbf{A} + \mathbf{B} + \mathbf{C}) \cdot (\mathbf{A} + \overline{\mathbf{B}} + \mathbf{C}) = \mathbf{A} + \mathbf{C}$$
$$(\mathbf{A} \cdot \mathbf{B} \cdot \mathbf{C}) + (\mathbf{A} \cdot \overline{\mathbf{B}} \cdot \mathbf{C}) = \mathbf{A} \cdot \mathbf{C}$$

c.

$$(\mathbf{A} + \mathbf{B}) \cdot (\mathbf{B} + \mathbf{C}) \cdot (\overline{\mathbf{A}} + \mathbf{C}) = (\mathbf{A} + \mathbf{B}) \cdot (\overline{\mathbf{A}} + \mathbf{C})$$
$$(\mathbf{A} \cdot \mathbf{B}) + (\mathbf{B} \cdot \mathbf{C}) + (\overline{\mathbf{A}} \cdot \mathbf{C}) = (\mathbf{A} \cdot \mathbf{B}) + (\overline{\mathbf{A}} \cdot \mathbf{C})$$

Në këtë mënyrë mund të gjenden edhe barazimet duale të postulateve, të ligjeve, dhe të teoremave të algjebrës së Bulit.

Shembull

Vërtetimi se barazimet e dhëna te ligji i distribucionit:

$$\mathbf{A} \cdot (\mathbf{B} + \mathbf{C}) = \mathbf{A} \cdot \mathbf{B} + \mathbf{A} \cdot \mathbf{C}$$
$$\mathbf{A} + \mathbf{B} \cdot \mathbf{C} = (\mathbf{A} + \mathbf{B}) \cdot (\mathbf{A} + \mathbf{C})$$

janë barazime duale.

$$\mathbf{A} \cdot (\mathbf{B} + \mathbf{C}) = (\mathbf{A} \cdot \mathbf{B}) + (\mathbf{A} \cdot \mathbf{C})$$
$$\mathbf{A} + (\mathbf{B} \cdot \mathbf{C}) = (\mathbf{A} + \mathbf{B}) \cdot (\mathbf{A} + \mathbf{C})$$

Funksionet inverse

Me inversionin e një funksioni logjik $\mathbf{f}$ nënkuptohet gjetja e komplementit të tij $\overline{\mathbf{f}}$, për të cilët vlejnë raportet:

$$\mathbf{f} + \overline{\mathbf{f}} = \mathbf{1}$$
$$\mathbf{f} \cdot \overline{\mathbf{f}} = \mathbf{0}$$

Procesi i komplementimit mund të përshkruhet me shprehjen:

$$\overline{f}\{A, B, \ldots, 0, 1, +, \cdot\} = f\{\overline{A}, \overline{B}, \ldots, 1, 0, \cdot, +\}$$

ku shihet se variablat zëvendësohen me inversionet e tyre, vlerat **0** me **1**, vlerat **1** me **0**, operatorët **+** me **·**, operatorët **·** me **+**.

Komplementi i funksionit, përkatësisht inversioni i tij, gjendet duke shfrytëzuar funksionin dual përkatës, ose përmes teoremave të *De Morgan-it*.

Përmes funksionit dual

Nëse në funksionin dual f_d të një funksioni f, variablat zëvendësohen me vlerat e tyre kompelmentare, do të fitohet funksioni invers $\overline{f}$ përkatës.

Shembull

Funksionet inverse të funksioneve:

a. $f = AB + C(\overline{A} + B)$

b. $g = AB + \overline{B}C + A\overline{C}$

c. $v = (A + B)(C + D)$

d. $h = ABD + A(\overline{B} + \overline{C}) + \overline{B}(\overline{C} + D)$

përmes funksioneve duale përkatëse.

a.

$$f = A \cdot B + C \cdot (\overline{A} + B)$$

$$\overline{f} = (\overline{A} + \overline{B}) \cdot [\overline{C} + (A \cdot \overline{B})]$$

b.

$$g = A \cdot B + \overline{B} \cdot C + A \cdot \overline{C}$$

$$\overline{g} = (\overline{A} + \overline{B}) \cdot (B + \overline{C}) \cdot (\overline{A} + C)$$

c.

$$v = (A + B) \cdot (C + D)$$

$$\overline{v} = (\overline{A} \cdot \overline{B}) + (\overline{C} \cdot \overline{D})$$

d.

$$h = A \cdot B \cdot D + A \cdot (\overline{B} + \overline{C}) + \overline{B} \cdot (\overline{C} + D)$$

$$\overline{h} = (\overline{A} + \overline{B} + \overline{D}) \cdot [\overline{A} + (B \cdot C)] \cdot [B + (C \cdot \overline{D})]$$

Përmes teoremave të De Morgan-it

Funksionet inverse gjendet shumë më lehtë përmes teoremave të *De Morgan-it*. Gjatë kësaj duhet pasur kujdes në zbatimin e drejtë të teoremave, përkatësisht në grupimin adekuat të pjesëve të shprehjes së funksionit.

Shembull

Funksionet inverse të funksioneve nga shembulli paraprak, duke shfrytëzuar teoremat e *De Morgan-it*.

a.

$$f = AB + C(\overline{A} + B)$$

$$\overline{f} = \overline{AB + C(\overline{A} + B)}$$

$$= (\overline{AB}) \cdot [\overline{C \cdot (\overline{A} + B)}]$$

$$= (\overline{A} + \overline{B}) \cdot [\overline{C} + (\overline{\overline{A} + B})]$$

$$= (\overline{A} + \overline{B}) \cdot [\overline{C} + (\overline{\overline{A}} \cdot \overline{B})]$$

$$= (\overline{A} + \overline{B}) \cdot [\overline{C} + (A \cdot \overline{B})]$$

b.

$$g = AB + \overline{B}C + A\overline{C}$$

$$\overline{g} = \overline{AB + \overline{B}C + A\overline{C}}$$

$$= \overline{AB} \cdot \overline{\overline{B}C} \cdot \overline{A\overline{C}}$$

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

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

c.

$$v = (A + B) \cdot (C + D)$$

$$\overline{v} = \overline{(A + B) \cdot (C + D)}$$

$$= \overline{(A + B)} + \overline{(C + D)}$$

$$= (\overline{A} \cdot \overline{B}) + (\overline{C} \cdot \overline{D})$$

d.

$$\mathbf{h} = \mathbf{ABD} + \mathbf{A}(\overline{\mathbf{B}} + \overline{\mathbf{C}}) + \overline{\mathbf{B}}(\overline{\mathbf{C}} + \mathbf{D})$$

$$\overline{\mathbf{h}} = \overline{\mathbf{ABD} + \mathbf{A}(\overline{\mathbf{B}} + \overline{\mathbf{C}}) + \overline{\mathbf{B}}(\overline{\mathbf{C}} + \mathbf{D})}$$

$$= \overline{\mathbf{ABD}} \cdot [\overline{\mathbf{A}(\overline{\mathbf{B}} + \overline{\mathbf{C}})}] \cdot [\overline{\overline{\mathbf{B}}(\overline{\mathbf{C}} + \mathbf{D})}]$$

$$= (\overline{\mathbf{A}} + \overline{\mathbf{B}} + \overline{\mathbf{D}}) \cdot [\overline{\mathbf{A}} + (\overline{\overline{\mathbf{B}} + \overline{\mathbf{C}}})] \cdot [\overline{\overline{\mathbf{B}}} + (\overline{\overline{\mathbf{C}} + \mathbf{D}})]$$

$$= (\overline{\mathbf{A}} + \overline{\mathbf{B}} + \overline{\mathbf{D}}) \cdot [\overline{\mathbf{A}} + (\overline{\overline{\mathbf{B}}} \cdot \overline{\overline{\mathbf{C}}})] \cdot [\mathbf{B} + (\overline{\overline{\mathbf{C}}} \cdot \overline{\mathbf{D}})]$$

$$= (\overline{\mathbf{A}} + \overline{\mathbf{B}} + \overline{\mathbf{D}}) \cdot [\overline{\mathbf{A}} + (\mathbf{B} \cdot \mathbf{C})] \cdot [\mathbf{B} + (\mathbf{C} \cdot \overline{\mathbf{D}})]$$

Në rastin e përgjithshëm, gjatë gjetjes së funksionit invers në dy rrugë të ndryshme, si rezultat nuk do të fitohen shprehje të njëjta.

Format e paraqitjes së funksioneve logjike

Funksionet logjike kanë një numër të caktuar vlerash, sepse vlerat e variablave logjike, të cilat marrin pjesë në shprehjet përkatëse, mund të jenë vetëm **0** ose **1**. Për këtë arsye, funksionet logjike mund të paraqiten përmes qarqeve me ndërprerës, tabelave të kombinimeve, diagrameve kohore, diagrameve të Vennit, **K**-diagrameve dhe qarqeve logjike.

Qarqet me ndërprerës

Ndërprerësi elektrik, mund t'i paraqesë fizikisht variablat logjike, nëse dy pozicionet e ndërprerësit përdoren për paraqitjen e dy vlerave të mundshme të variablave. Kështu, p.sh., me pozitën e hapur të ndërprerësit mund të paraqitet vlera **0**, kurse vlerës **1** t'i përgjigjet pozita e mbyllur e ndërprerësit.

Në Fig.3.1 janë dhënë qarqet me ndërprerës për **3** funksionet logjike elementare, ku, për arsye praktike, në pozitat e ndërprerësve janë shënuar variablat përkatëse.

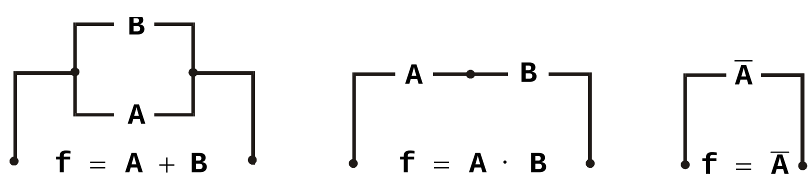


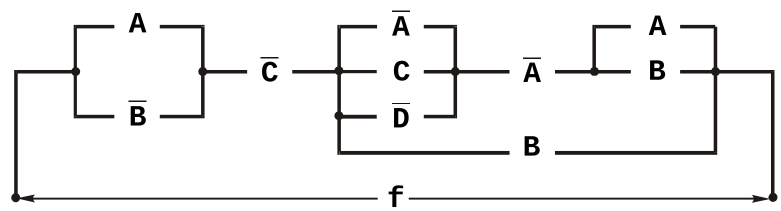
Fig.3.1 Qarqet me ndërprerës për funksionet logjike elementare

Nga qarqet e dhëna shihet se sinjali elektrik, në rrugën e tij prej hyrjes në dalje të qarkut, gjatë lidhjes paralele të degëve, mund të kalojë nëse është mbyllur ndërprerësi **A**, ose ndërprerësi **B**, ose janë të mbyllur të dy ndërprerësit. Kurse gjatë lidhjes serike sinjali mund të merret në dalje të qarkut, nëse është mbyllur ndërprerësi **A** dhe ndërprerësi **B**, përkatësisht nëse njëkohësisht janë të mbyllur të dy ndërprerësit.

Plotësisht njëlloj, përmes qarqeve me ndërprerës, mund të paraqiten edhe funksionet e ndryshme logjike, pavarësisht nga kompleksiteti i shprehjeve algjebrike përkatëse.

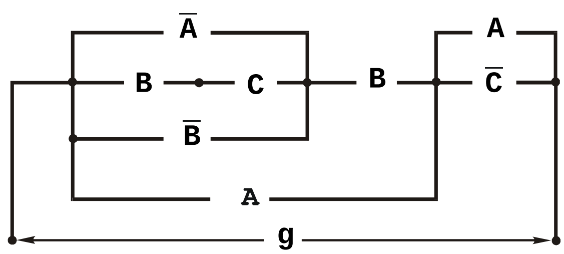
Shembull Gjetja e shprehjeve algjebrike të funksioneve logjike të cilat janë dhënë përmes qarqeve me ndërprerës, të cilët shihen më poshtë.

a.



$$\mathbf{f} = [(A + \bar{B})\cdot\bar{C}]\cdot[(\bar{A} + C + \bar{D})\cdot\bar{A}\cdot(A + B) + B]$$

b.



$$\mathbf{g} = [(\bar{A} + B\cdot C + \bar{B})\cdot B + A]\cdot[A + \bar{C}]$$

Gjatë vizatimit të qarqeve me ndërprerës, për funksionet logjike të ndryshme, sikurse edhe gjatë gjetjes së funksionit logjik nga qarku i dhënë, për çdo komponente të shprehjes së përcaktuar me operacionin e mbledhjes ose të shumëzimit vizatohet pjesa përkatëse e qarkut. Në këtë mënyrë në fund fitohet qarku me ndërprerës, i cili i përgjigjet funksionit të dhënë.

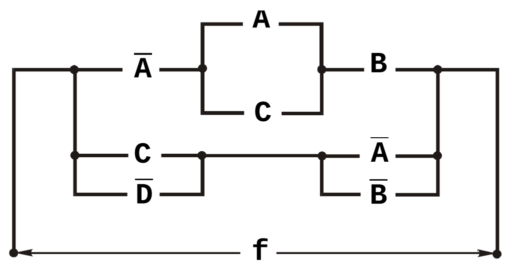
Shembull

Vizatimi i qarqeve me ndërprerës për funksionet logjike:

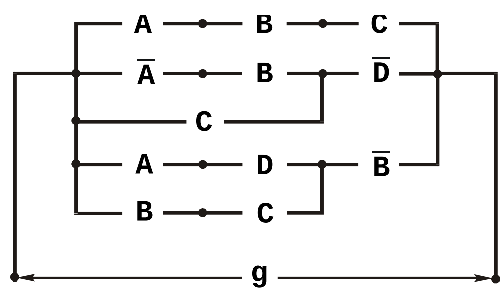
a. $\mathbf{f = (A + C) \cdot \overline{A}B + (C + \overline{D}) \cdot (\overline{A} + \overline{B})}$

b. $\mathbf{g = ABC + (\overline{A}B + C) \cdot \overline{D} + (AD + BC) \cdot \overline{B}}$

a.



b.



Tabelat e kombinimeve

Meqë vlerat e variablave të cilat marrin pjesë në shprehjet e funksioneve logjike mund të jenë **0** dhe **1**, funksionet kanë një numër të fundëm vlerash. Këto vlera fitohen nëse merren të gjitha kombinimet e mundshme të variablave. Kështu, për funksionin i cili në vete përmbanë **n**-variabla numri i kombinimeve të ndryshme të vlerave të variablave është $\mathbf{2^n}$.

Paraqitja e vlerave të funksioneve logjike për të gjitha kombinimet e mundshme të variablave brenda tyre mund të bëhet duke shfrytëzuar *tabelat e kombinimeve* (ang. table of combinations), të cilat në literaturë quhen edhe *tabela të vërtetësisë* (ang. true table). Për **3** funksionet logjike elementare tabelat e kombinimeve janë dhënë në Fig.3.2, ku dy tabelat e para kanë nga **4** kombinime, sepse në shprehjet e tyre paraqiten **2** variabla të ndryshme.

A	B	f
0	0	0
0	1	1
1	0	1
1	1	1

$f = A + B$

A	B	f
0	0	0
0	1	0
1	0	0
1	1	1

$f = A \cdot B$

A	f
0	1
1	0

$f = \overline{A}$

Fig.3.2 Tabelat e kombinimeve për funksionet logjike elementare

Në tabelat e kombinimeve, përveç kolonave për variablat e veçanta dhe funksionin logjik, rregullisht figurojnë edhe kolona plotësuese, në të cilat shënohen vlerat e komponenteve të ndyshme brenda shprehjes së funksionit logjik.

Shembull

Tabelat e kombinimeve për funksionet logjike:

a. $f = \overline{A} + B$

b. $g = A\overline{B} + A\overline{C} + \overline{A}C$

c. $h = (A + B + \overline{C})(\overline{A}B + A\overline{B})$

a.

A	B	$\overline{A}$	f
0	0	1	1
0	1	1	1
1	0	0	0
1	1	0	1

b.

A	B	C	$\overline{A}$	$\overline{B}$	$\overline{C}$	$A\overline{B}$	$A\overline{C}$	$\overline{A}C$	g
0	0	0	1	1	1	0	0	0	0
0	0	1	1	1	0	0	0	1	1
0	1	0	1	0	1	0	0	0	0
0	1	1	1	0	0	0	0	1	1
1	0	0	0	1	1	1	1	0	1
1	0	1	0	1	0	1	0	0	1
1	1	0	0	0	1	0	1	0	1
1	1	1	0	0	0	0	0	0	0

c.

A	B	C	$\bar{A}$	$\bar{B}$	$\bar{C}$	$A + B + \bar{C}$	$\bar{A}B$	$A\bar{B}$	$\bar{A}B + A\bar{B}$	h
0	0	0	1	1	1	1	0	0	0	0
0	0	1	1	1	0	0	0	0	0	0
0	1	0	1	0	1	1	1	0	1	1
0	1	1	1	0	0	1	1	0	1	1
1	0	0	0	1	1	1	0	1	1	1
1	0	1	0	1	0	1	0	1	1	1
1	1	0	0	0	1	1	0	0	0	0
1	1	1	0	0	0	1	0	0	0	0

Induksioni i plotë

Metoda sipas së cilës vërtetohet barazia e dy funksioneve logjike, duke gjetur se vlerat e tyre janë të barabarta për të gjitha kombinimet e mundshme, quhet *induksion i plotë* (ang. perfect induction).

Shembull

Vërtetimi i barazisë së funksioneve logjike:

$$f = (A + B)(B + C)(C + A)$$
$$g = AB + BC + CA$$

përmes metodës së induksionit të plotë.

A	B	C	A+B	B+C	C+A	AB	BC	CA	f	g
0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	1	1	0	0	0	0	0
0	1	0	1	1	0	0	0	0	0	0
0	1	1	1	1	1	0	1	0	1	1
1	0	0	1	0	1	0	0	0	0	0
1	0	1	1	1	1	0	0	1	1	1
1	1	0	1	1	1	1	0	0	1	1
1	1	1	1	1	1	1	1	1	1	1

Meqë vlerat në dy kolonat e fundit të tabelës për të gjitha kombinimet e mundshme janë të barabarta, vërtetohet se funksionet **f** dhe **g** janë të barabartë.

Me metodën e induksionit të plotë mund të vërtetohet edhe saktësia e ligjeve dhe e teoremave të dhëna më parë, ose edhe saktësia e barazimeve logjike të ndryshme.

Shembull

Vërtetimi i saktësisë së barazimeve:

a. $\overline{A+B}=\overline{A}\cdot\overline{B}$

b. $(A+B)(A+BC)=A+BC$

c. $A(A+B)(B+C)=A(B+C)$

duke shfrytëzuar metodën e induksionit të plotë.

a.

A	B	$\overline{A}$	$\overline{B}$	A + B	$\overline{A+B}$	$\overline{A}\cdot\overline{B}$
0	0	1	1	0	1	1
0	1	1	0	1	0	0
1	0	0	1	1	0	0
1	1	0	0	1	0	0

b.

A	B	C	A+B	BC	A+BC	(A+B)(A+BC)
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	1	0	0	0
0	1	1	1	1	1	1
1	0	0	1	0	1	1
1	0	1	1	0	1	1
1	1	0	1	0	1	1
1	1	1	1	1	1	1

c.

A	B	C	A+B	B+C	A(B+C)	A(A+B)(B+C)
0	0	0	0	0	0	0
0	0	1	0	1	0	0
0	1	0	1	1	0	0
0	1	1	1	1	0	0
1	0	0	1	0	0	0
1	0	1	1	1	1	1
1	1	0	1	1	1	1
1	1	1	1	1	1	1

Meqë në dy kolonat e fundit të tabelave të mësipërme janë fituar vlera të njëjta për të gjitha kombinimet e mundshme të vlerave të variablave, mbështetur në principin e induksionit të plotë, vërtetohet se barazimet e dhëna janë të sakta.

Mintermat dhe makstermat

Për çdo kombinim të vlerave të variablave të një funksioni mund të formohen *mintermat* (ang. minterm) dhe *makstermat* (ang. maxterm) përkatëse.

Minterma m_i, për kombinimin e i-të të vlerave të variablave, formohet si prodhim i variablave që kanë vlerën **1** dhe komplementin e variablave (*kovariablave*) me vlerën **0**. Kurse maksterma M_i, për kombinimin e i-të të vlerave të variablave formohet si shumë e variablave me vlerën **0** dhe kovariablave me vlerën **1**. Këtu, indeksi i e paraqet ekuivalentin decimal të kombinimit të vlerave të variablave sipas të cilit formohet minterma m_i, përkatësisht maksterma M_i përkatëse.

Shembull Gjetja e mintermave dhe e makstermave, për funksionet **f** dhe **g** të dhëna përmes tabelave të kombinimeve vijuese.

a.

A	B	f
0	0	1
0	1	0
1	0	0
1	1	1

b.

A	B	C	g
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

a.

i	A	B	f	Mintermat	Makstermat
0	0	0	1	$m_0 = \bar{A}\bar{B}$	$M_0 = A + B$
1	0	1	0	$m_1 = \bar{A}B$	$M_1 = A + \bar{B}$
2	1	0	0	$m_2 = A\bar{B}$	$M_2 = \bar{A} + B$
3	1	1	1	$m_3 = AB$	$M_3 = \bar{A} + \bar{B}$

b.

i	A	B	C	g	Mintermat	Makstermat
0	0	0	0	1	$m_0 = \overline{A}\overline{B}\overline{C}$	$M_0 = A + B + C$
1	0	0	1	1	$m_1 = \overline{A}\overline{B}C$	$M_1 = A + B + \overline{C}$
2	0	1	0	0	$m_2 = \overline{A}B\overline{C}$	$M_2 = A + \overline{B} + C$
3	0	1	1	1	$m_3 = \overline{A}BC$	$M_3 = A + \overline{B} + \overline{C}$
4	1	0	0	0	$m_4 = A\overline{B}\overline{C}$	$M_4 = \overline{A} + B + C$
5	1	0	1	0	$m_5 = A\overline{B}C$	$M_5 = \overline{A} + B + \overline{C}$
6	1	1	0	1	$m_6 = AB\overline{C}$	$M_6 = \overline{A} + \overline{B} + C$
7	1	1	1	1	$m_7 = ABC$	$M_7 = \overline{A} + \overline{B} + \overline{C}$

Varësisht nga vlera e funksionit (**1** ose **0**), për kombinimin e vlerave të variablave në bazë të të cilave formohen minterma dhe maksterma përkatëse ato shënohen edhe kështu:

m^0 - mintermat për të cilat funksioni ka vlerën **0**.
m^1 - mintermat për të cilat funksioni ka vlerën **1**.
M^0 - makstermat për të cilat funksioni ka vlerën **0**.
M^1 - makstermat për të cilat funksioni ka vlerën **1**.

Shprehjet algjebrike të funksioneve

Nga tabela e kombinimeve të funksionit logjik, përkatësisht duke shfrytëzuar mintermat dhe makstermat përkatëse, mund të shkruhet shprehja algjebrike e funksionit si shumë e mintermave me vlerën **1** (*forma disjunktive*), ose prodhim i makstermave me vlerën **0** (*forma konjuktive*).

Shembull

Gjetja e shprehjeve algjebrike për funksionet **f** dhe **g** nga detyra paraprake, duke shfrytëzuar mintermat dhe makstermat përkatëse.

a.

i	A	B	f
0	0	0	1
1	0	1	0
2	1	0	0
3	1	1	1

$$f(A, B) = \sum m^1(0, 3) = m_0 + m_3$$
$$= \overline{A}\overline{B} + AB$$

$$f(A, B) = \prod M^0(1, 2) = M_1 \cdot M_2$$
$$= (A + \overline{B}) \cdot (\overline{A} + B)$$

Kështu, funksioni përkatës i shprehur përmes minitermave dhe makstermave është:

$$f(A,B) = \overline{A}\,\overline{B} + A\,B$$

$$f(A,B) = (A + \overline{B})(\overline{A} + B)$$

b.

$$g(A, B, C) = \sum m^1(0,1,3,6,7) = m_0 + m_1 + m_3 + m_6 + m_7$$

i	A	B	C	g
0	0	0	0	1
1	0	0	1	1
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

$$g(A, B, C) = \prod M^0(2,4,5) = M_2 \cdot M_4 \cdot M_5 =$$

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

Këtu me simbolet Σ dhe Π duhet nënkuptuar shumën dhe prodhimin e mintermave, përkatësisht makstermave, të cilat janë shënuar brenda kllapave.

Shprehjet analitike të një funksioni me n-variabla, në formën e tyre *disjunktive* dhe *konjuktive*, mund të fitohen edhe duke i shkruar kështu:

$$f = \sum_{i=0}^{2^n-1} f_i \cdot m_i$$

$$f = \prod_{i=0}^{2^n-1} (f_i + M_i)$$

ku me f_i janë shënuar vlerat e funksioneve për kombinimet e veçanta në tabelën e kombinimeve.

Shembull Gjetja e shprehjeve algjebrike për funksionet **f** dhe **g** nga detyra paraprake përmes shprehjeve të dhëna më sipër.

a.

$$f(A, B) = 1 \cdot m_0 + 0 \cdot m_1 + 0 \cdot m_2 + 1 \cdot m_3$$

i	A	B	f
0	0	0	1
1	0	1	0
2	1	0	0
3	1	1	1

$$f(A, B) = [1 + M_0] \cdot [0 + M_1] \cdot [0 + M_2] \cdot [1 + M_3]$$

b.

$$g(A, B, C) = 1 \cdot m_0 + 1 \cdot m_1 + 0 \cdot m_2 + 1 \cdot m_3 + 0 \cdot m_4 + 0 \cdot m_5 + 1 \cdot m_6 + 1 \cdot m_7$$

i	A	B	C	g
0	0	0	0	1
1	0	0	1	1
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

$$g(A, B, C) = [1 + M_0] \cdot [1 + M_1] \cdot [0 + M_2] \cdot [1 + M_3] \cdot [0 + M_4] \cdot [0 + M_5] \cdot [1 + M_6] \cdot [1 + M_7]$$

Meqë çdo variabël ka dy vlera të mundshme, me **n**-variabla fitohen gjithsej 2^n kombinime, përkatësisht tabela e kombinimeve përkatëse do t'i ketë 2^n rreshta. Për çdo kombinim të vlerave të variablave, $x_1, x_2, ..., x_n$, funksioni përkatës $f(x_1, x_2, ..., x_n)$ ka dy vlera të mundshme (vlerën **f** dhe komplementin $\overline{f}$). Prej këtu del se mund të përpilohen gjithsej 2^k tabela të kombinimeve, ku $k=2^n$, përkatësisht mund të krijohen gjithsej 2^{2^n} funksione logjike të ndryshme. Numri i funksioneve të mundshme, për disa vlera të numrit të variablave **n**, është dhënë në tabelën e Fig.3.3.

n	2^n	2^{2^n}
0	1	2
1	2	4
2	4	16
3	8	256
4	16	65536

Fig.3.3 Numri i funksioneve me n variabla

Nga tabela shihet se për **n=0** (pa asnjë variabël) mund të formohen dy funksione logjike ($2^{2^0} = 2$) dhe ato janë:

$f_0 = 0$

$f_1 = 1$

Nëse në shprehjen e funksionit përdoret vetëm një variabël (**n=1**), mund të formohen gjithsej $2^{2^1} = 4$ funksione logjike:

$f_0 = 0$

$f_1 = A$

$f_2 = \bar{A}$

$f_3 = 1$

Kurse për **n=2** numri i funksioneve që mund të formohen është $2^{2^2} = 16$. Këto funksione bashkë me tabelën e kombinimeve janë dhënë në vijim.

A	B	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
		0	AB	$A\bar{B}$	$A\bar{B} + AB = A$	$\bar{A}B$	$\bar{A}B + AB = B$	$\bar{A}B + A\bar{B} = A \oplus B$	$\bar{A}B+A\bar{B}+AB=A+B$	$\bar{A}\bar{B}$	$\bar{A}\bar{B} + AB = A \otimes B$	$\bar{A}\bar{B} + A\bar{B} = B$	$\bar{A}\bar{B}+A\bar{B}+AB=A+\bar{B}$	$\bar{A}\bar{B} + \bar{A}B = \bar{A}$	$\bar{A}\bar{B}+\bar{A}B+AB=\bar{A}+B$	$\bar{A}\bar{B}+A\bar{B}+\bar{A}B=\bar{A}+\bar{B}$	1

Fig.3.4 Funksionet që formohen me 2 variabla

Diagramet kohore

Për paraqitjen grafike të vlerave të funksioneve për të gjitha kombinimet e mundshme të variablave që marrin pjesë në shprehjet e tyre mund të përdoren edhe *diagramet kohore* (ang. timing diagram). Te këto diagrame kombimimet e mundshme vendosen gjatë boshtit horizontal, kurse në boshtin vertikal vendosen vlerat logjike të cilat u përkasin kombinimeve të veçanta, duke marrë një përmasë të caktuar për vlerën **1**. Kështu, për **3** funksionet logjike elementare, diagramet kohore përkatëse shihen në Fig.3.5.

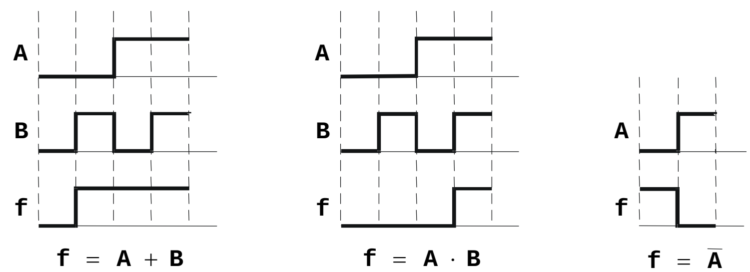


Fig.3.5 Diagramet kohore për funksionet logjike elementare

Kryesisht, amplituda e vlerës logjike **0** merret **0**, ashtu siç është paraqitur në diagramet e dhëna më sipër. Por, ajo mund të merret edhe si një vlerë më e vogël se amplituda përkatëse e vlerës logjike **1**, gjë që nuk ka ndonjë ndikim në paraqitjen e vlerave të variablave dhe funksioneve logjike.

Shembull

Paraqitja përmes diagrameve kohore e vlerave të funksioneve **f** dhe **g**, të cilat janë dhënë përmes tabelave kombinuese në vijim.

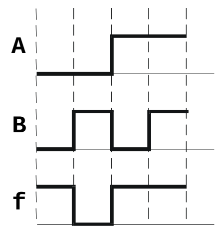
a.

A	B	f
0	0	1
0	1	0
1	0	1
1	1	1

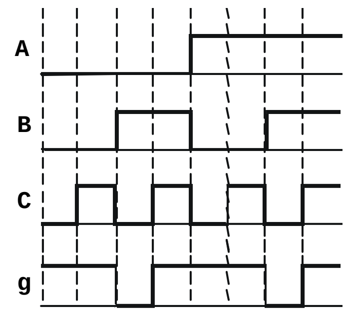
b.

A	B	C	g
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

a.



b.



Sikurse edhe gjatë paraqitjes së funksioneve përmes tabelave të kombinimeve, përveç vlerave të variablave dhe funksioneve, në diagramet kohore gjithashtu mund të vizatohen edhe vlerat e kovariablave dhe të komponenteve të veçanta brenda shprehjeve të funksioneve.

Funksionet që paraqiten përmes diagrameve kohore nuk është e domosdoshme të jepen në formë tabelare. Nëse funksionet jepen në formë algjebrike, gjatë vizatimit të diagrameve kohore operacionet e veçanta kryhen duke pasur parasysh vlerat në momentin e caktuar, në pjesët përkatëse të diagrameve kohore.

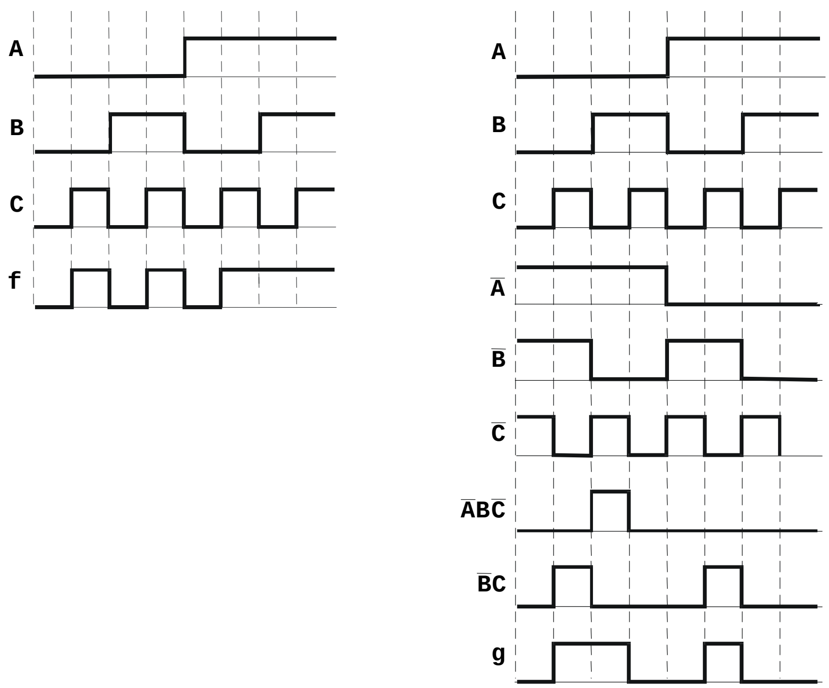
Shembull

Paraqitja përmes diagrameve kohore e vlerave të funksioneve:

a. $f = AB + C$

b. $g = \overline{A}B\overline{C} + \overline{B}C$

a. *b.*



Diagramet e Venn-it

Për paraqitjen e vlerave të funksioneve logjike përmes sipërfaqeve gjeometrike përdoren *diagramet e Venn-it* (sipas John Venn-it, i cili e ka zbatuar për herë të parë këtë teknikë të paraqitjes së funksioneve logjike, në vitin 1880).

Diagramet e *Venn-it* vizatohen brenda një drejtkëndëshi, i cili e paraqet universumin (ang. universe) **U**, përkatësisht brenda tij përfshihen të gjitha vlerat e mundshme. Vlerat e variablave paraqiten përmes rrathëve, të cilët vizatohen ashtu që të priten mes vete. Kjo më së miri shihet në Fig.3.6, ku përmes diagrameve të *Venn-it* janë paraqitur **3** funksionet logjike elementare.

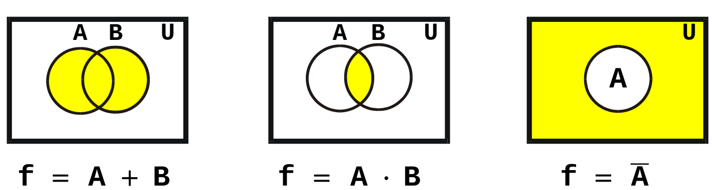


Fig.3.6 Diagramet e Venn-it për funksionet logjike elementare

Nga diagramet e dhënë në Fig.3.6 vërehet se sipërfaqja e përfshirë me rrathët e variablave **A** dhe **B** i përket mbledhjes, përkatësisht funksionit $f=A+B$. Kurse shumëzimi $f=A\cdot B$ paraqitet me sipërfaqen e cila është e përbashkët për variablat **A** dhe **B**. Nga diagrami i fundit më së miri shihet se hapësira brenda rrethit i përket variablës **A**, kurse me hapësirën jashtë rrethit paraqitet funksioni $f=\overline{A}$.

Plotësisht njëlloj mund të përdoren diagramet e *Venn-it* gjatë paraqitjes së funksioneve të ndryshme logjike. Por, me qëllim që të mos gabohet, më mirë është që *për komponentet brenda funksioneve logjike të vizatohen diagrame të veçanta të Venn-it*. Pastaj, në fund, duke mbledhur fizikisht sipërfaqet e diagrameve që u përkasin komponenteve të veçanta, fitohet diagrami i kërkuar i *Venn-it*.

Shembull

Paraqitja e funksioneve:

a. $y=\overline{A}B+A\overline{B}$

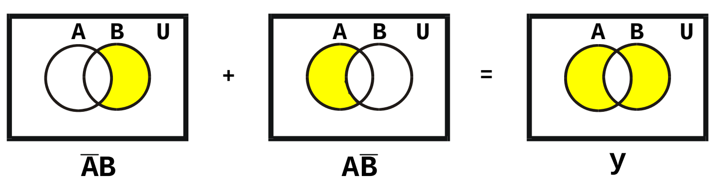
b. $z=AB+\overline{A}\,\overline{B}$

c. $v=AB+\overline{A}C+BC$

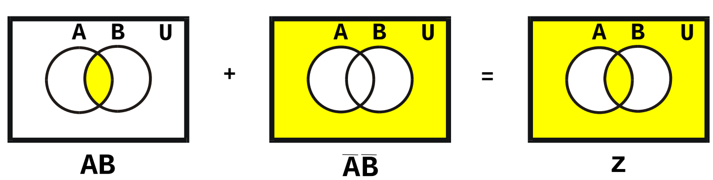
d. $u=(A+B)(\overline{A}+C)$

përmes diagrameve të *Venn-it*.

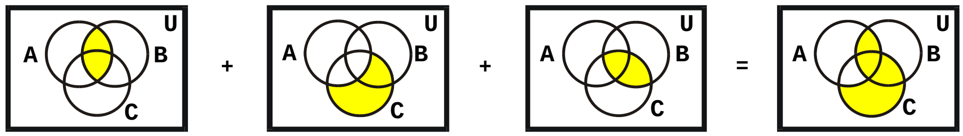
a.



b.

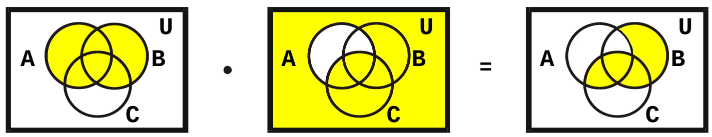


c.



AB $\quad \bar{A}C$ $\quad BC$ $\quad V$

d.



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

Diagramet e Venn-it janë të përshtatshëm për vërtetimin e ligjeve, teoremave dhe barazimeve logjike të ndyshme, sepse vlerat e funksioneve logjike në dy anët e barazimit shihen vizualisht në paraqitjet grafike përkatëse.

Shembull

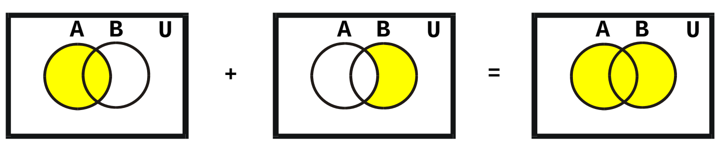
Vërtetimi i identiteteve:

a. $A + \bar{A}B = A + B$

b. $(A + B)\,(\bar{A} + C) = AC + \bar{A}B$

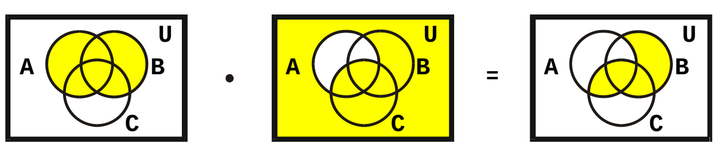
përmes diagrameve të Venn-it.

a.

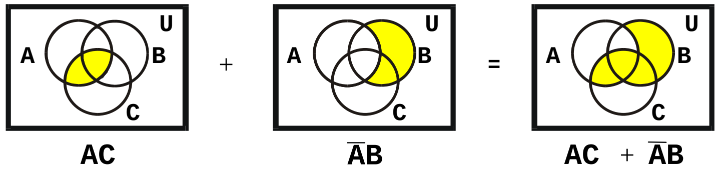


A $\quad \bar{A}B$ $\quad A + B$

b.



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



K-diagramet

Diagramet e Karnaugh-it, ose shkurt ***K**-diagramet*, paraqesin një formë të veçantë të vizatimit të tabelave të kombinimeve, ku kombinimet e vlerave të variablave fitohen me prerjen e sipërfaqeve përkatëse. Në fakt, ***K**-diagramet* janë të ngjashme me *diagramet e Venn-it*, por këtu për paraqitjen e variablave në vend të rrathëve përdoren drejtkëndësha, kurse në vend të mbushjeve me mostra të ndryshme shënohen shifrat binare **1** dhe **0**. Diagramet e tilla njihen edhe si *diagrame të Veitch-it*.

Në Fig.3.7 janë dhënë dy format e ***K***-diagrameve që përdoren më shpesh, për rastet me **1**, **2**, **3**, **4**, **5** dhe **6** variabla.

Numrat decimalë që janë shënuar brenda fushave të veçanta në **K**-diagramet e dhëna në anën e djathtë paraqesin ekuivalentët e vlerave binare të variablave. Këta numra përdoren gjatë përcjelljes direkte të vlerave nga tabelat e kombinimeve në **K**-diagrame për kombinimet e veçanta të vlerave të variablave.

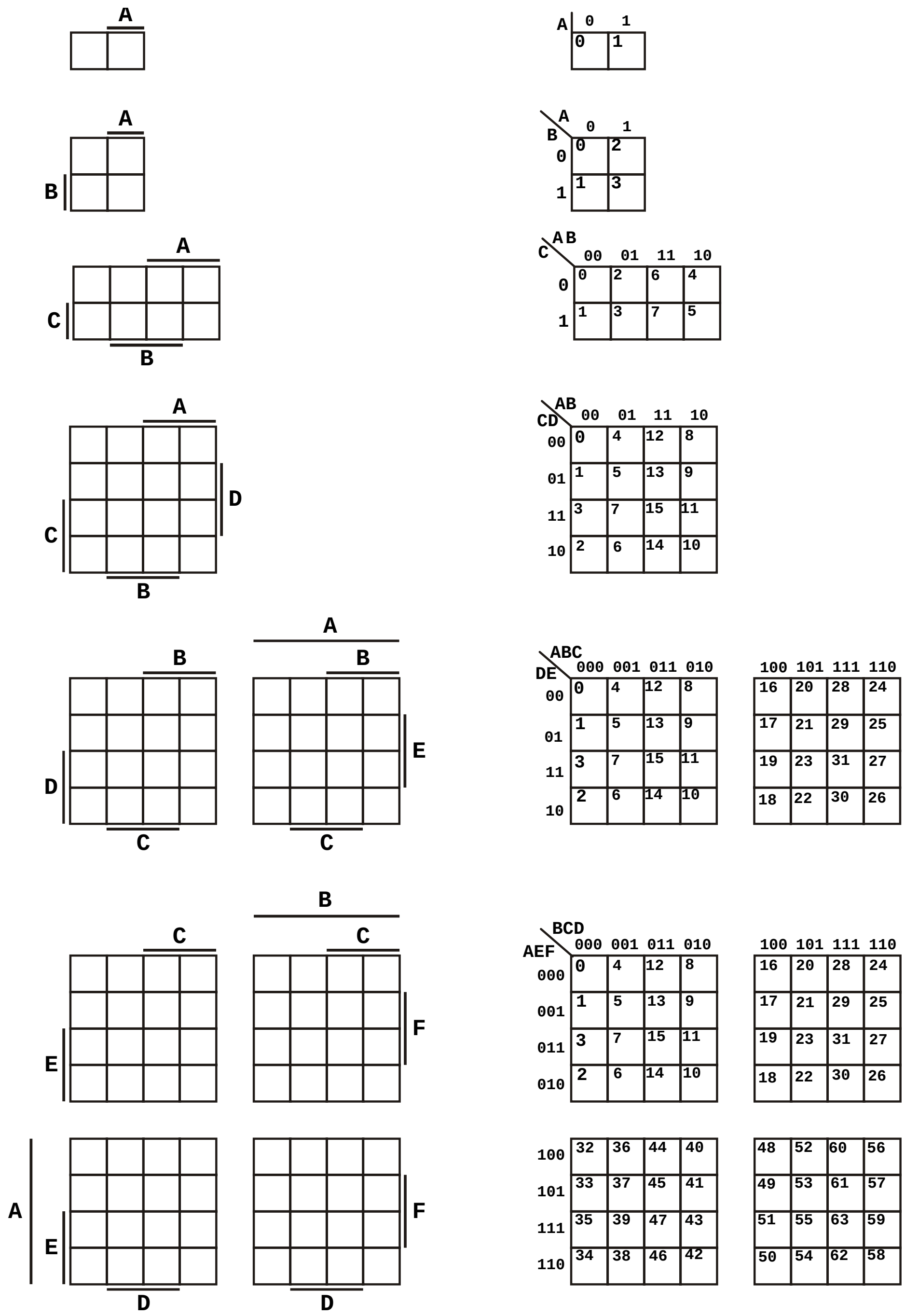


Fig.3.7 Dy format e paraqitjes së K-diagrameve

Nëse për një funksion dihet tabela e kombinimeve, mbushja e **K**-diagramit bëhet duke përcjellë vlerat e kombinimeve të veçanta në fushat përkatëse.

Shembull

Për funksionet e dhëna përmes tabelave të kombinimeve vijuese të mbushen **K**-diagramet përkatës.

a.

A	B	f
0	0	1
0	1	0
1	0	0
1	1	1

b.

A	B	C	g
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

a.

B \ A	0	1
0	1	0
1	0	1

b.

C \ AB	00	01	11	10
0	1	0	0	1
1	0	1	0	1

Nëse funksionet jepen përmes shumës së mintermave, ose prodhimit të makstermave, procedura e mbushjes së **K**-diagrameve thjeshtësohet dukshëm, duke shfrytëzuar versionet e **K**-diagrameve me fusha brenda të cilave janë shënuar numrat decimalë.

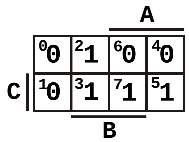
Shembull

Mbushja e **K**-diagrameve për funksionet:

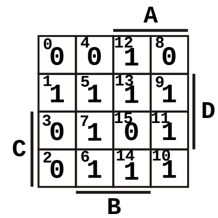
a. $f(A,B,C) = \sum m^1(2,3,5,7)$

b. $g(A,B,C,D) = \prod M^0(0,2,3,4,8,15)$

a.



b.

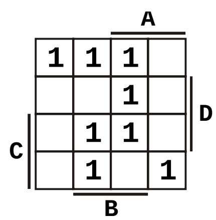


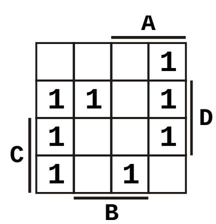
Në praktikë gjatë mbushjes së **K**-diagrameve brenda fushave të tyre shënohen vetëm vlerat **0**, ose vlerat **1**.

Nëse në **K**-diagramin e një funksioni **f** zëvendësohen vlerat **0** me **1** dhe vlerat **1** me **0**, fitohet **K**-diagrami i funksionit invers përkatës $\overline{\mathbf{f}}$.

Shembull

Gjetja e **K**-diagramit të funksionit invers për funksionin i cili është përcaktuar përmes **K**-diagramit vijues:





Nëse dihet shprehja algjebrike e një funksioni, **K**-diagrami përkatës mund të mbushet duke përpiluar së pari tabelën përkatëse të kombinimeve. Por, kjo rrugë kërkon mjaft punë dhe kujdes, sidomos kur kemi të bëjmë me funksione me më shumë variabla.

Mbushja e **K**-diagramit mund të bëhet edhe në rrugë direkte, duke mbushur fushat të cilat u përkasin komponenteve të veçanta brenda shprehjes algjebrike të funksionit.

Shembull

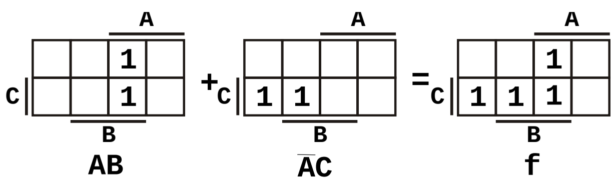
Mbushja direkte e **K**-diagrameve për funksionet e përcaktuara përmes shprehjeve algjebrike:

a. $f = AB + \overline{A}C$

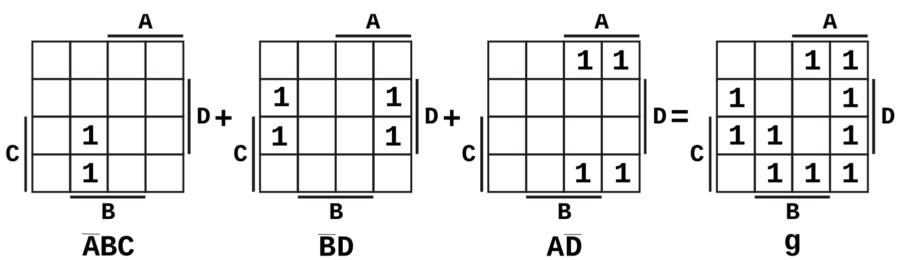
b. $g = \overline{A}BC + \overline{B}D + A\overline{D}$

c. $u = AC\overline{D} + \overline{A}\,\overline{B}\,\overline{C}\,\overline{D} + \overline{A}B$

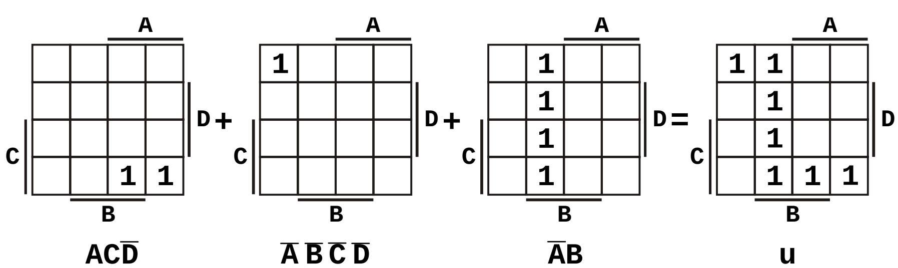
a.



b.



c.



Nga shembujt e dhënë më sipër shihet se me vlerat **1** mbushen *fushat që fitohen me prerjen e hapësirave të cilat u përkasin variablave dhe kovariablave që marrin pjesë* *në formimin e mintermës përkatëse.* P.sh., te shembulli nën **a**, komponentes **AB** i përgjigjet fusha e cila fitohet me prerjen e hapësirave të variablave **A** dhe **B**, ngjashëm siç fitohej prodhimi i këtyre variablave te diagramet e Vennit.

Nëse shprehja algjebrike e funksionit jepet si prodhim i shumave, përkatësisht nëse në shprehje paraqiten maksterma, **K**-diagrami përkatës mund të mbushet me **0**. Fushat përkatëse në këtë rast fitohen si *prerje e kovariablave të variablave që marrin pjesë në formimin e komponenteve të veçanta.*

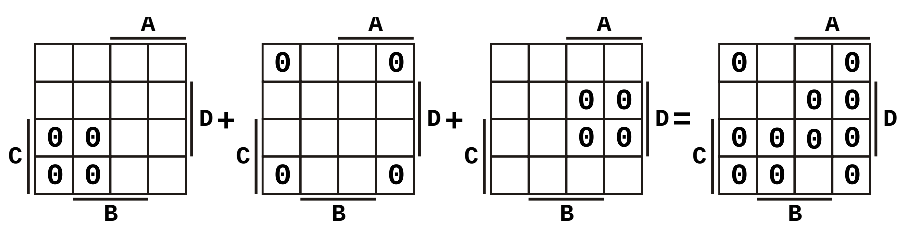
Shembull

Mbushja direkte e **K**-diagrameve për funksionet e përcaktuara përmes shprehjeve algjebrike:

a. $\mathbf{f} = (A + \overline{C})(B + D)(\overline{A} + \overline{D})$

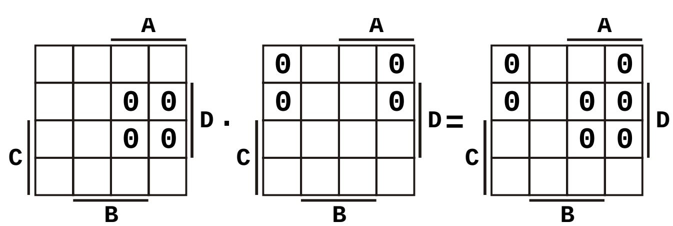
b. $\mathbf{g} = (\overline{A} + \overline{D})(B + C)$

a.



$A + \overline{C}$ $\quad$ $B + D$ $\quad$ $\overline{A} + \overline{D}$ $\quad$ $\mathbf{f}$

b.



$\overline{A} + \overline{D}$ $\quad$ $(B + C)$ $\quad$ $\mathbf{g}$

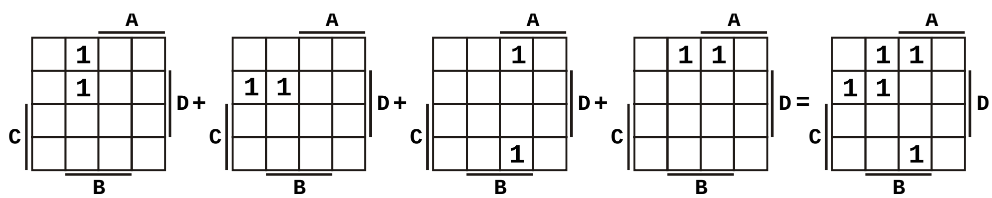
Shprehjet e funksioneve algjebrike, të cilat janë dhënë përmes prodhimit të shumave, mund të konvertohen në formën e tyre disjunktive, duke u liruar nga kllapat.

Shembull

Konvertimi i funksioneve të dhëna në shembullin paraprak në formën e tyre disjunktive dhe pastaj mbushja e **K**-diagrameve përkatës.

a.

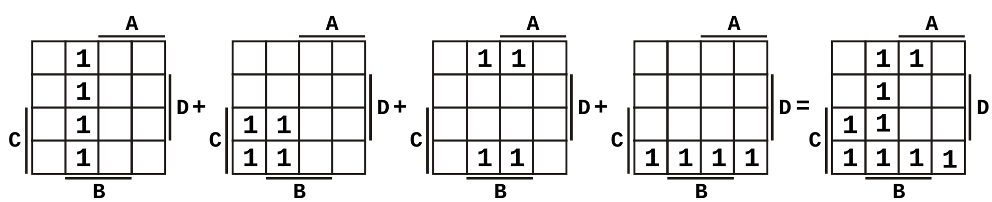
$$
\begin{aligned}
f &= (A + \bar{C})(B + D)(\bar{A} + \bar{D}) \\
&= (AB + B\bar{C} + AD + D\bar{C})(\bar{A} + \bar{D}) \\
&= A\bar{A}B + \bar{A}B\bar{C} + A\bar{A}D + \bar{A}\bar{C}D + AB\bar{D} + B\bar{C}\bar{D} + AD\bar{D} + \bar{C}D\bar{D} \\
&= \bar{A}B\bar{C} + \bar{A}\bar{C}D + AB\bar{D} + B\bar{C}\bar{D}
\end{aligned}
$$



$\bar{A}B\bar{C}$ $\quad$ $\bar{A}\ \bar{C}\ D$ $\quad$ $AB\bar{D}$ $\quad$ $B\bar{C}\ \bar{D}$ $\quad$ f

b.

$$
\begin{aligned}
g &= (\bar{A} + \bar{D})(B + C) \\
&= \bar{A}B + \bar{A}C + B\bar{D} + C\bar{D}
\end{aligned}
$$



$\bar{A}B$ $\quad$ $\bar{A}C$ $\quad$ $B\bar{D}$ $\quad$ $C\bar{D}$ $\quad$ g

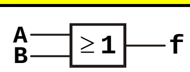
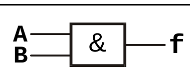
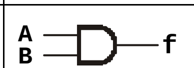
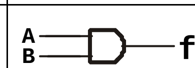
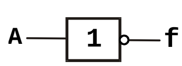
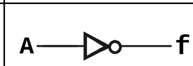
Qarqet logjike

Prodhimi i pajisjeve kompjuterike mbështetet në shfrytëzimin e qarqeve të ndyshme elektronike, të cilat me një emër njihen si *qarqe kompjuterike* (ang. computer circuits). Përshkrimi matematikor i qarqeve të tilla bëhet përmes funksioneve logjike, prandaj këto qarqe quhen edhe *qarqe logjike* (ang. logic circuits). Dy vlerat logjike të mundshme, **0** dhe **1**, te qarqet kompjuterike paraqiten përmes *dy tensioneve* ose *dy rrymave* të caktuara, përkatësisht përmes *sinjalit të lartë* dhe *sinjalit të ulët*. Nëse vlera **1** paraqitet me sinjalin e lartë, kurse vlera **0** - me sinjalin e ulët, atëherë thuhet se kemi të bëjmë me *logjikë pozitive* (ang. positive logic). Por, kur për paraqitjen e vlerës **1** përdoret sinjali i ulët, kurse për vlerën **0** - sinjali i lartë, flitet për të ashtuquajturën *logjikë negative* (ang. negative logic). Në praktikë, si tensione për paraqitjen e sinjalit të lartë dhe të ulët, përdoren, p.sh., tensionet **+5V** dhe **0V** (familja e elementeve logjike **TTL**), ose **-1.55V** dhe **-0.75V** (familja e elementeve logjike **ECL**), ose, p.sh., rrymat **20mA** dhe **0mA**. Në pjesën vijuese të librit, sa herë që flitet për vlerat logjike **1** dhe **0**, do të mendohet në tensionet **+5V** dhe **0V**.

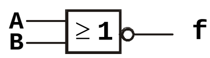
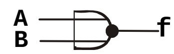
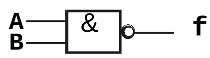
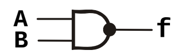
Për realizimin praktik të qarqeve logjike shfrytëzohen *elemente logjike* (ang. logic element), përmes së cilave kryhen operacione të ndryshme.

Për vizatimin e qarqeve kompjuterike elementet logjike paraqiten duke i pasur parasysh standardet përkatëse botërore. Disa nga format e paraqitjes së këtyre elementeve, në bazë të rekomandimeve të **IEC**-së (nga. International Electrotechnical Commision), standardit të ushtrisë amerikane (**MIL-STD 806B**) dhe standardit gjerman **DIN** (nga Deutsch Industrie Norm), me funksionet logjike përkatëse, janë dhënë në tabelat e Fig.3.8, të grupuara në **3** grupe të elementeve: *themelore*, *universale* dhe *speciale*.

Elementet logjike themelore

Elementi	Funksioni	Simboli		
		Rekomandimi i IEC	DIN	MIL-STD
OSE	$f = A + B$			
DHE	$f = A \cdot B$			
JO	$f = \overline{A}$			

Elementet logjike universale

Elementi	Funksioni	Simboli		
		Rekomandimi i IEC	DIN	MIL-STD
JOOSE	$f = \overline{A + B}$			
JODHE	$f = \overline{A \cdot B}$			

Elementet logjike speciale

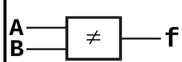
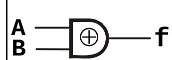
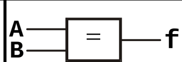
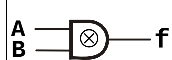
Elementi	Funksioni	Simboli		
		Rekomandimi i IEC	DIN	MIL-STD
Komparatori i jobarazisë	$f = \bar{A}B + A\bar{B}$ $= A \oplus B$			
Komparatori i barazisë	$f = AB + \bar{A}\bar{B}$ $= A \otimes B$			

Fig.3.8 Format e paraqitjes së elementeve logjike

Tabelat e kombinimeve për dy elementet logjike speciale, të dhëna në tabelën e fundit të Fig.3.8, të cilat njihen edhe si *ekskluziv*-**OSE** (**XOSE**) e *ekskluziv*-**DHE** (**XDHE**), janë:

A	B	A ⊕ B
0	0	0
0	1	1
1	0	1
1	1	0

A	B	A ⊗ B
0	0	1
0	1	0
1	0	0
1	1	1

Nga tabelat e dhëna shihet se vlejnë raportet:

$$\overline{A \oplus B} = A \otimes B$$
$$\overline{A \otimes B} = A \oplus B$$

dhe atë, si për dy ashtu edhe për më shumë variabla.

Në pjesën vijuese, për vizatimin e qarqeve logjike do të përdoren simbolet e dhëna në kolonat e fundit të tabelave, por grupi i elementeve speciale do të plotësohet edhe me disa elemente të tjera.

Shembull

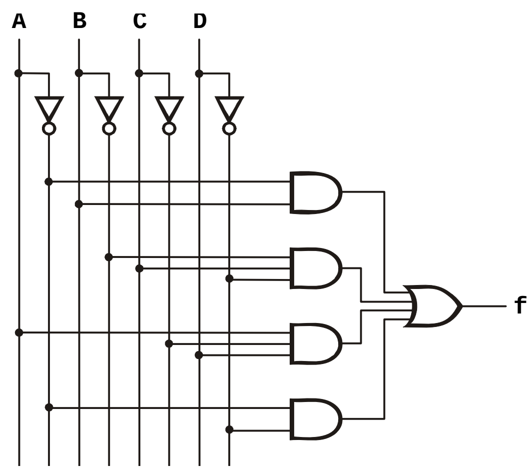
Vizatimi i qarqeve logjike të cilat përshkruhen përmes funksioneve:

a. $f = \overline{A}B + \overline{B}C\overline{D} + A\overline{C}D + \overline{A}\,\overline{D}$

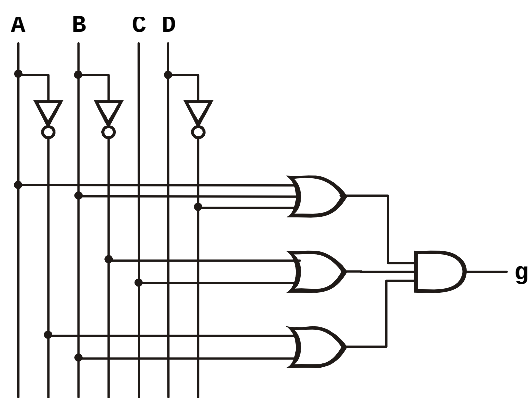
b. $g = (A + B + \overline{D})(\overline{B} + C)(\overline{A} + B)$

duke shfrytëzuar elemente logjike themelore.

a.



b.



Minimizimi i funksioneve

Gjatë realizimit praktik të qarqeve logjike kërkohen format më të thjeshta të tyre, me qëllim të kursimit në material dhe në punë. Për këtë qëllim, para realizimit të qarqeve gjenden shprehjet minimale të funksioneve logjike përmes të cilave përshkruhen, përkatësisht bëhet minimizimi i tyre.

Në përgjithësi, për minimizimin e funksioneve logjike përdoret *minimizimi algjebrik*, *minimizimi grafik* dhe *minimizimi tabelar*.

Minimizimi algjebrik

Minimizimi i funksioneve logjike në rrugë algjebrike mbështetet në shfrytëzimin e postulateve, ligjeve dhe teoremave të algjebrës së Bulit.

Shembull

Gjetja e formës minimale të funksioneve:

a. $f = A + ABC + AB + \overline{A}BC$

b. $u = (A + \overline{B})C + A\overline{B} + C$

c. $g = (A + \overline{C})(A + D)(B + \overline{C})(B + D)$

d. $v = [AB(C + D + E) + ABD](\overline{A} + \overline{B} + \overline{E})$

e. $h = (A + B + CD)(\overline{A} + B)(\overline{A} + B + E)$

duke shfrytëzuar minimizimin algjebrik.

a.

$$\begin{aligned} f &= A + ABC + AB + \overline{A}BC \\ &= A + AB + ABC + \overline{A}BC \\ &= A(1 + B) + BC(A + \overline{A}) \\ &= A + BC \end{aligned}$$

b.

$$
\begin{aligned}
u &= (A + \overline{B})C + A\overline{B} + C \\
&= AC + \overline{B}C + A\overline{B} + C \\
&= AC + A\overline{B} + (\overline{B} + 1)C \\
&= AC + A\overline{B} + C \\
&= (A + 1)C + A\overline{B} \\
&= C + A\overline{B} \\
&= A\overline{B} + C
\end{aligned}
$$

c.

$$
\begin{aligned}
g &= (A + \overline{C})(A + D)(B + \overline{C})(B + D) \\
&= [AA + A\overline{C} + AD + \overline{C}D][BB + B\overline{C} + BD + \overline{C}D] \\
&= [A + A\overline{C} + AD + \overline{C}D][B + B\overline{C} + BD + \overline{C}D] \\
&= [A(1 + \overline{C}) + AD + \overline{C}D][B(1 + \overline{C}) + BD + \overline{C}D] \\
&= [A + AD + \overline{C}D][B + BD + \overline{C}D] \\
&= [A(1 + D) + \overline{C}D] \cdot [B(1 + D) + \overline{C}D] \\
&= [A + \overline{C}D][B + \overline{C}D] \\
&= AB + B\overline{C}D + A\overline{C}D + \overline{C}\,\overline{C}DD \\
&= AB + B\overline{C}D + A\overline{C}D + \overline{C}D \\
&= AB + B\overline{C}D + (A + 1)\overline{C}D \\
&= AB + B\overline{C}D + \overline{C}D \\
&= AB + (B + 1)\overline{C}D \\
&= AB + \overline{C}D
\end{aligned}
$$

d.

$$
\begin{aligned}
v &= [AB(C + D + E) + ABD][\overline{A} + \overline{B} + \overline{E}] \\
&= [ABC + ABD + ABE + ABD][\overline{A} + \overline{B} + \overline{E}] \\
&= [ABC + ABD + ABE][\overline{A} + \overline{B} + \overline{E}] \\
&= A\overline{A}BC + A\overline{A}BD + A\overline{A}BE + AB\overline{B}C + AB\overline{B}D \\
&\quad + AB\overline{B}E + ABC\overline{E} + ABD\overline{E} + ABE\overline{E} \\
&= ABC\overline{E} + ABD\overline{E}
\end{aligned}
$$

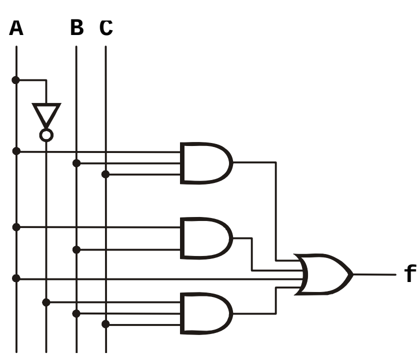
e.

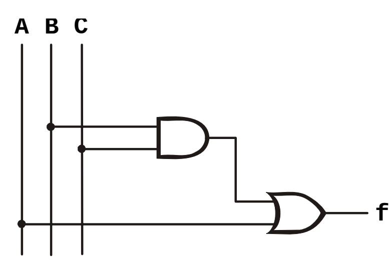
$$
\begin{aligned}
h &= (A + B + CD)(\overline{A} + B)(\overline{A} + B + E) \\
&= [A\overline{A} + \overline{A}B + \overline{A}CD + AB + BB + BCD](\overline{A} + B + E) \\
&= [\overline{A}B + \overline{A}CD + AB + B + BCD](\overline{A} + B + E) \\
&= [\overline{A}B + \overline{A}CD + (A + 1)B + BCD](\overline{A} + B + E) \\
&= [\overline{A}B + \overline{A}CD + B + BCD]\cdot(\overline{A} + B + E) \\
&= [\overline{A}B + \overline{A}CD + B(1 + CD)](\overline{A} + B + E) \\
&= [\overline{A}B + \overline{A}CD + B](\overline{A} + B + E) \\
&= \overline{A}\,\overline{A}B + \overline{A}\,\overline{A}CD + \overline{A}B + \overline{A}BB + \overline{A}BCD + BB + \overline{A}BE + \overline{A}CDE + BE \\
&= \overline{A}B + \overline{A}CD + \overline{A}B + \overline{A}B + \overline{A}BCD + B + \overline{A}BE + \overline{A}CDE + BE \\
&= \overline{A}B + \overline{A}CD + \overline{A}BCD + B + \overline{A}BE + \overline{A}CDE + BE \\
&= \overline{A}B(1 + CD) + \overline{A}CD(1 + E) + B + BE(\overline{A} + 1) \\
&= \overline{A}B + \overline{A}CD + B + BE \\
&= (\overline{A} + 1)B + \overline{A}CD + BE \\
&= B + \overline{A}CD + BE \\
&= B(1 + E) + \overline{A}CD \\
&= B + \overline{A}CD
\end{aligned}
$$

Që të shihet efekti i minimizimit të funksioneve, p.sh., le t'i vizatojmë qarqet logjike për funksionet **f** dhe **u**, të dhëna në shembullin e mësipërm, para dhe pas gjetjes së shprehjeve minimale përkatëse.

a.

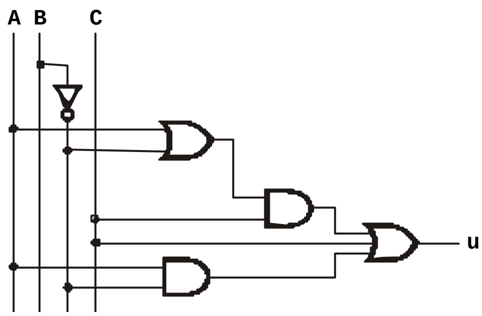
$f = A + ABC + AB + \overline{A}BC \qquad f = A + BC$

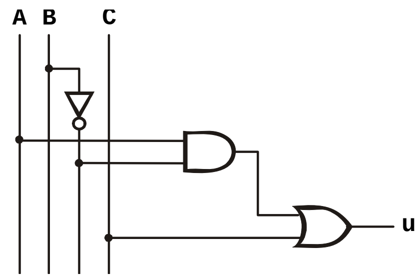




b.

$$\mathbf{u} = (\mathbf{A} + \overline{\mathbf{B}})\mathbf{C} + \mathbf{A}\overline{\mathbf{B}} + \mathbf{C} \qquad\qquad \mathbf{u} = \mathbf{A}\overline{\mathbf{B}} + \mathbf{C}$$





Përgjigjja adekuate në pyetjen se cila është forma minimale e funksionit, varet nga elementet logjike të cilat i kemi, nga çmimi i tyre, ose nga puna për realizimin e qarkut. Por, kriter esencial që në praktikë merret gjatë gjetjes së shprehjeve minimale të funksioneve është ai i *minimizimit të numrit të shkronjave që marrin pjesë në shprehje*, ku me shkronja nënkuptohen të gjitha *variablat* dhe *kovariablat*. P.sh., funksioni **f** i dhënë nën *a* në shembullin e mësipërm:

$$\mathbf{f} = \mathbf{A} + \mathbf{ABC} + \mathbf{AB} + \overline{\mathbf{A}}\mathbf{BC}$$

përmban **9** shkronja. Pas minimizimit të tij është fituar shprehja:

$$\mathbf{f} = \mathbf{A} + \mathbf{BC}$$

dukshëm më e thjeshtë, sepse përmban vetëm **3** shkronja.

Minimizimi grafik

Procesi i gjetjes së shprehjes minimale të një funksioni në rrugë grafike mbështetet në shfrytëzimin e **K**-diagramit përkatës. Gjatë kësaj ndiqet procedura vijuese.

1. Grupohen fushat me vlera **1**, duke pasur parasysh *principet e grupimit*.

2. Për çdo grup shkruhet *minterma e grupit*, si prodhim i variablave dhe i kovariablave (komplementit të variablave), të cilat gjatë gjithë grupit nuk i ndryshojnë vlerat.

3. Shkruhet shprehja minimale e funksionit si shumë e mintermave të grupeve të veçanta.

Principet e grupimit të fushave brenda **K**-diagramit janë:

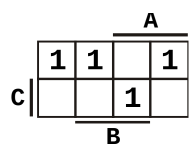
- Grupohen vetëm fushat fqinjë, kurse si fqinjë llogariten fushat të cilat tangohen mes vete me brinjë dhe jo me kënde.
- Në çdo grup mund të përfshihen $\mathbf{2^n}$ fusha, ku $\mathbf{n=0,\ 1,\ 2,\ ...}$
- Grupin mund ta formojë edhe vetëm një fushë.
- Një fushë mund të përfshihet njëkohësisht në më shumë grupe.

Si fqinjë llogariten edhe fushat në skajet e kundërta të **K**-diagramit, duke e paramenduar takimin e tyre në pakufi, ose si tangim i cili do të ndodhte nëse **K**-diagrami lakohet.

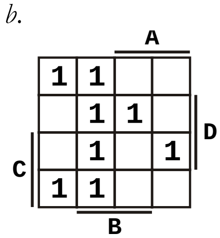
Shembull

Gjetja e shprehjeve minimale të funksioneve të dhëna përmes **K**-diagrameve:

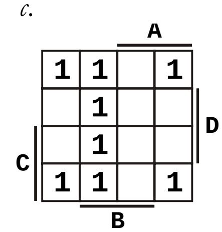
a.



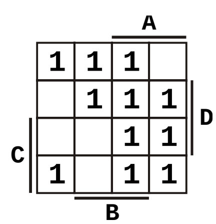
b.



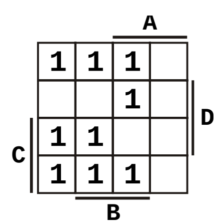
c.



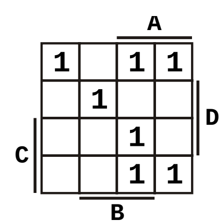
d.



e.

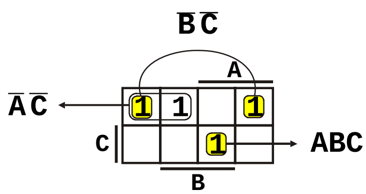


f.



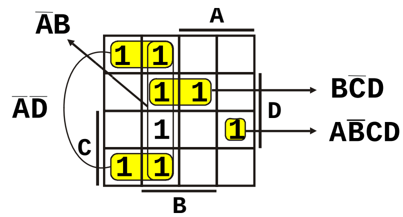
duke shfrytëzuar minimizimin grafik.

a.



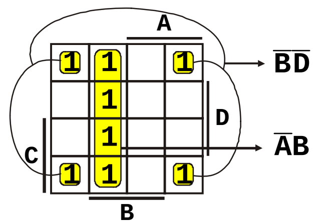
$$f = \overline{A}\,\overline{C} + \overline{B}\,\overline{C} + A\,B\,C$$

b.



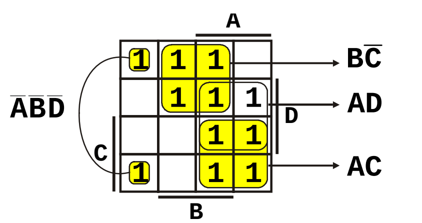
$$f = \overline{A}\,B + \overline{A}\,\overline{D} + A\overline{B}CD + B\overline{C}D$$

c.



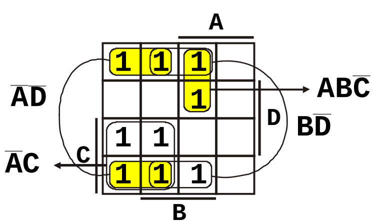
$$f = \overline{A}B + \overline{B}\,\overline{D}$$

d.



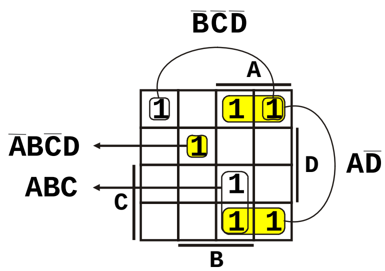
$$f = AC + AD + B\overline{C} + \overline{A}\,\overline{B}\,\overline{D}$$

e.



$$f = AB\overline{C} + \overline{A}C + \overline{A}\,\overline{D} + B\overline{D}$$

f.



$$f = A\overline{D} + ABC + \overline{A}B\overline{C}D + \overline{B}\,\overline{C}\,\overline{D}$$

Në **K**-diagram mund të grupohen edhe fushat me vlera **0**, duke zbatuar principet e njëjta të cilat vlejnë gjatë grupimit të fushave me vlera **1**. Por, në këtë rast, për çdo grup shkruhet *maksterma e grupit* si shumë e variablave dhe e kovariablave (të kundërta me ato që shënohen në **K**-diagram), të cilat gjatë gjithë grupit nuk i ndryshojnë vlerat e tyre. Në fund, shprehja minimale e funksionit fitohet përmes prodhimit të makstermave të grupeve të veçanta.

Shembull

Gjetja e shprehjeve minimale të funksioneve:

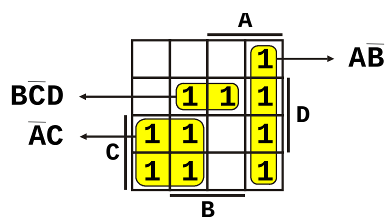
a. $f(A,B,C,D) = A\overline{B} + \overline{A}C + B\overline{C}D + A\overline{B}C$

b. $g(A,B,C) = \sum m^1(1,3,4,6)$

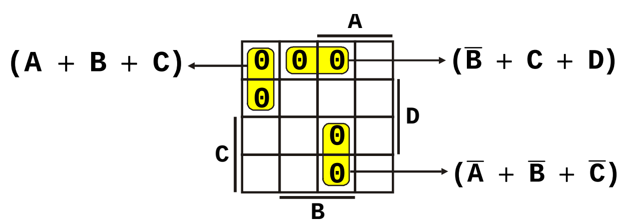
c. $h(A,B,C,D) = \prod M^0(0,2,3,7,13,15)$

në formën e tyre disjunktive dhe konjuktive, duke shfrytëzuar procedurën e minimizimit grafik.

a.

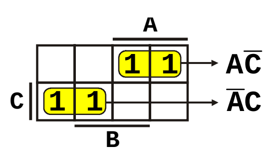


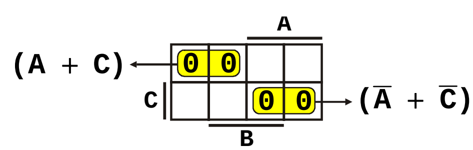
$$f = A\overline{B} + \overline{A}C + B\overline{C}D$$



$$f = (A+B+C)(\overline{A}+\overline{B}+\overline{C})(\overline{B}+C+D)$$

b.

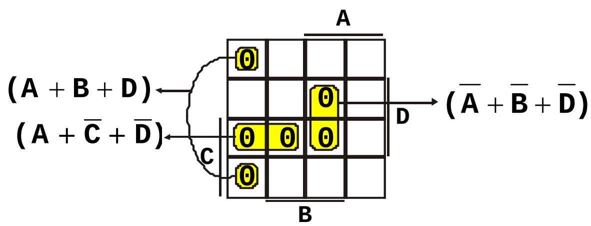




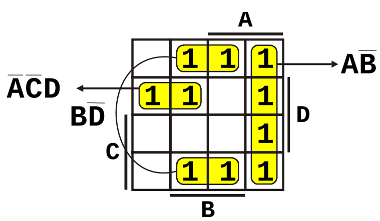
$$g = A\overline{C} + \overline{A}C$$

$$g = (A+C)(\overline{A}+\overline{C})$$

c.



$$h = (A + B + D)(\overline{A} + \overline{B} + \overline{D})(A + \overline{C} + \overline{D})$$



$$h = A\overline{B} + \overline{A}\,\overline{C}D + B\overline{D}$$

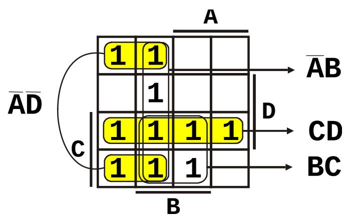
Nëse dihet **K**-diagrami i një funksioni, shprehja e funksionit invers mund të gjendet nëse përpilohet **K**-diagrami i funksionit invers (duke zëvendësuar vlerat **1** me **0** dhe vlerat **0** me **1**) dhe pastaj gjendet shprehja minimale përkatëse.

Shembull

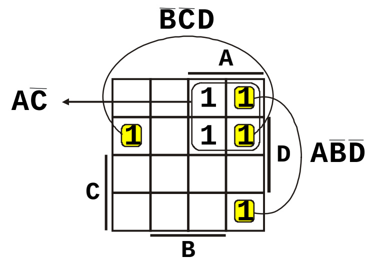
Gjetja e shprehjes së funksionit inverz në formën e saj disjunktive për funksionin i cili është dhënë përmes **K**-diagramit:

1	1	0	0
0	1	0	0
1	1	1	1
1	1	1	0

duke shfrytëzuar procedurën e minimizimit grafik.



$$f = \overline{A}B + \overline{A}\,\overline{D} + BC + CD$$



$$\overline{f} = A\overline{C} + A\overline{B}\,\overline{D} + \overline{B}\,\overline{C}D$$

Funksionet me vlera të çfarëdoshme dhe të pacaktuara

Shpesh herë, në praktikë takohen raste kur funksioni, në disa kombinime të vlerave të variablave:

- ka vlera të çfarëdoshme (**+**), përkatësisht **0** ose **1**, ose
- ka vlera të pacaktuara (**-**).

Në këto dy raste gjatë gjetjes së formës minimale të funksionit, nëse nuk është thënë ndryshe, përvetësohen lirisht vlerat **0** ose **1**, ashtu që të formohen grupe sa më të mëdha të fushave që grupohen.

Mintermat dhe makstermat e fushave të cilat u përgjigjen vlerave të çfarëdoshme (**+**) dhe vlerave të pacaktuara (**-**) në pjesën vijuese do të shënohen me: m^{+}, m^{-}, M^{+} dhe M^{-}.

Shembull

Gjetja e shprehjeve minimale të funksioneve:

a.

A	B	C	f
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	+
1	0	1	+
1	1	0	-
1	1	1	-

b.

$$f(A,B,C,D) = \sum m^{1}(2,3,4,5,9,15) + \sum m^{+}(1,6,13,14)$$

c.

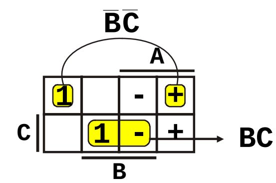
$$g(A,B,C,D) = \sum m^{1}(1,2,7,8,11,12) + \sum m^{+}(3,4,9,13) + \sum m^{-}(5,6)$$

d.

$$h(A,B,C,D) = \prod M^{0}(2,4,6,8,10) + \prod M^{+}(1,3,5,7) + \prod M^{-}(12,13,14,15)$$

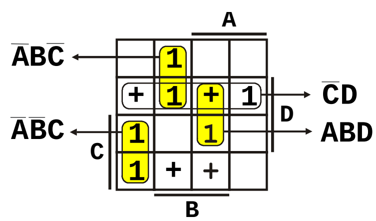
nëse vlerat e pacaktuara merren si vlera të çfarëdoshme.

a.



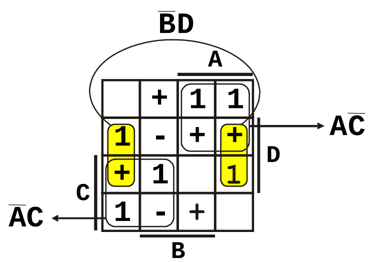
$$t = BC + \bar{B}\bar{C}$$

b.



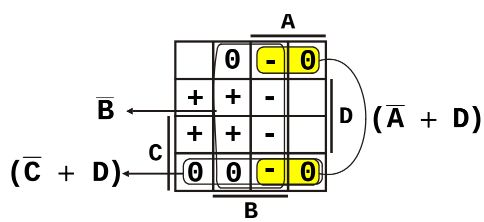
$$f = \bar{A}\bar{B}C + \bar{A}B\bar{C} + ABD + \bar{C}D$$

c.



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

d.



$$h = (\bar{A} + D) \cdot (\bar{C} + D) \cdot \bar{B}$$

Funksionet me më shumë variabla

Nëse funksionet kanë më shumë se **4** variabla gjatë minimizimit, si fqinje llogariten edhe fushat të cilat gjenden në pozicione të njëjta brenda **K**-diagrameve parciale. Kështu, së pari grupohen fushat në diagramet parciale dhe pastaj grupohen grupet e fushave të diagrameve parciale, të cilat kanë pozita plotësisht të njëjta.

Shembull

Gjetja e shprehjeve minimale të funksioneve:

a. $f(A,B,C,D,E)=\sum m^1(0,2,4,7,10,12,13,18,23,26,28,29)$

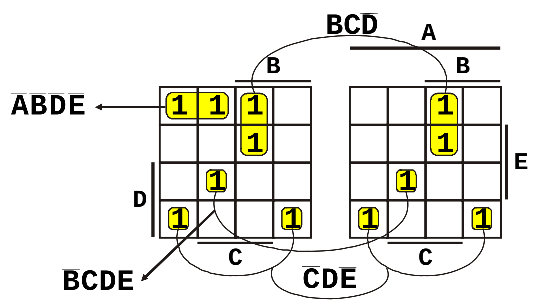
b. $g(A,B,C,D,E)=\sum m^1(1,3,4,9,11,12,13,15,17,19,22,25,27,29,30,31)$

c. $u(A,B,C,D,E)=\sum m^1(3,11,12,19,23,27,29)+\sum m^{+}(5,13,28)+\sum m^{-}(7,16)$

d. $v(A,B,C,D,E)=\sum m^1(1,3,4,9,11,12,13,15,17,19,22,25,27,29,30,31)$

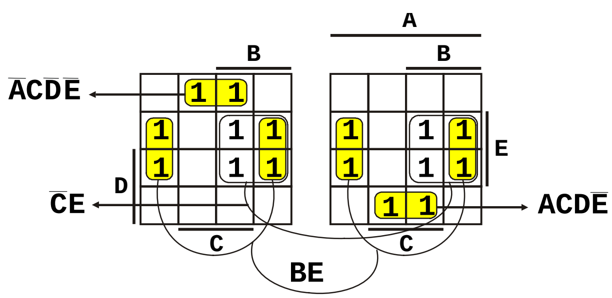
duke shfrytëzuar procedurën e minimizimit grafik.

a.



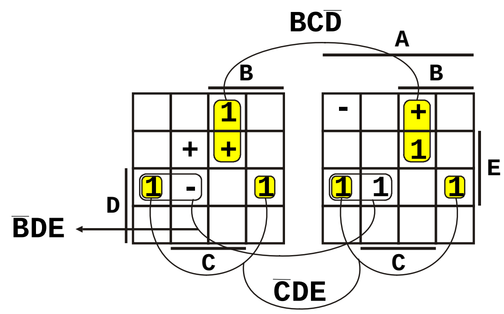
$$f=\overline{A}\,\overline{B}\,\overline{D}\,\overline{E}+BC\overline{D}+\overline{B}CDE+\overline{C}D\overline{E}$$

b.



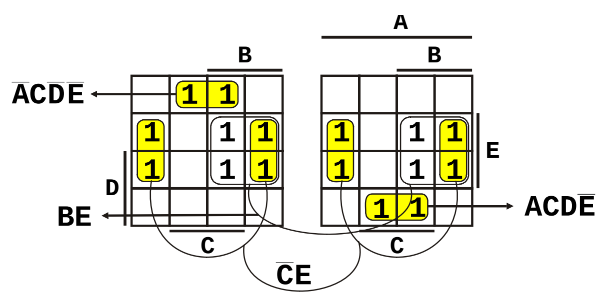
$$g=BE+\overline{C}E+ACD\overline{E}+\overline{A}\,C\,\overline{D}\,\overline{E}$$

c.



$$u = BC\bar{D} + \bar{B}DE + \bar{C}DE$$

d.



$$v = BE + \bar{C}E + ACD\bar{E} + \bar{A}C\bar{D}\bar{E}$$

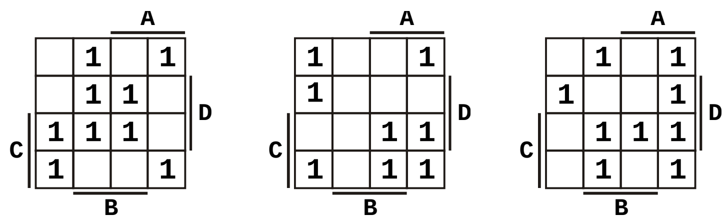
Prim - implikantët

Në **K**-diagramin e një funksioni mund të formohen më shumë grupe të fushave sesa që nevojiten për gjetjen e shprehjes minimale të tij. Mintermat të cilat u përkasin të gjitha grupeve të mundshme quhen *prim-implikantë*.

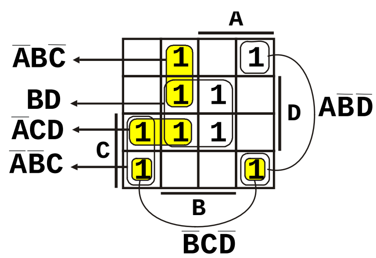
Shembull

Gjetja e prim-implikantëve të funksioneve të dhëna përmes **K**-diagrameve vijuese:

a. *b.* *c.*



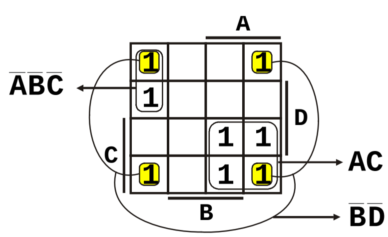
a.



Prim-implikantët

$A\bar{B}\bar{D}$	$\bar{A}CD$
$\bar{A}B\bar{C}$	$\bar{B}C\bar{D}$
BD	$\bar{A}\bar{B}C$

b.

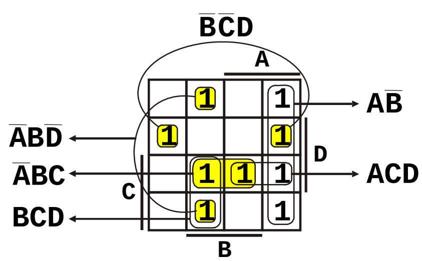


AC

$\bar{B}\bar{D}$

$\bar{A}\bar{B}\bar{C}$

c.



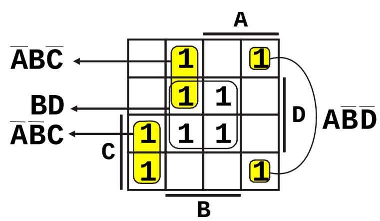
$A\bar{B}$	$\bar{A}BC$
$\bar{B}\bar{C}D$	BCD
$\bar{A}B\bar{D}$	ACD

Në shprehjen minimale të funksionit nuk marrin pjesë të gjithë prim-implikantët e mundshëm. Prim-implikantët nga të cilët formohet shprehja minimale e funksionit quhen *prim-implikantë esencialë*. Në fakt, prim-implikantë esencialë janë ata të cilët në vete përfshijnë së paku një fushë të **K**-diagramit, e cila nuk përfshihet nga prim-implikantët e tjerë.

Shembull	Gjetja e prim-implikantëve esencialë si dhe shprehjet minimale të funksioneve të cilët janë dhënë në shembullin paraprak.

a.

Prim-implikantët esencialë

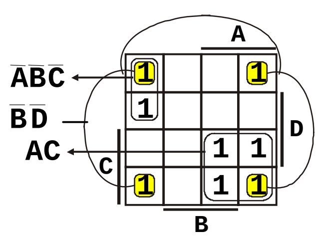


$A\overline{B}\overline{D}$ BD

$\overline{A}B\overline{C}$ $\overline{A}\overline{B}C$

$$f = A\overline{B}\overline{D} + \overline{A}B\overline{C} + BD + \overline{A}\overline{B}C$$

b.



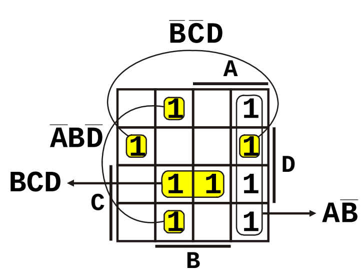
$\overline{A}\overline{B}\overline{C}$

$\overline{B}\overline{D}$

AC

$$f = \overline{A}\overline{B}\overline{C} + \overline{B}\overline{D} + AC$$

c.



$A\overline{B}$ BCD

$\overline{A}B\overline{D}$ $\overline{B}\overline{C}D$

$$f = A\overline{B} + \overline{A}B\overline{D} + BCD + \overline{B}\overline{C}D$$

Minimizimi tabelar

Përmes **K**-diagrameve mund të minimizohen funksionet me më pak variabla. Kurse shprehjet minimale të funksioneve me më shumë variabla gjenden duke e shfrytëzuar *minimizimin tabelar*, sipas metodës së *Quine-McCluskey-it*.

Shembull

Gjetja e shprehjes minimale të funksionit:

$$f(A,B,C,D) = \sum m^1(0,1,2,3,5,7,8,9,10,13,15)$$

duke shfrytëzuar procedurën e minimizimit tabelar.

I. Gjetja e prim-implikantëve

Rruga e gjetjes së prim-implikantëve rrjedh nëpër *4 hapa* karakteristikë.

1. Bëhet paraqitja binare e ekuivalentëve decimalë të mintermave:

	A	B	C	D
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
5	0	1	0	1
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
13	1	1	0	1
15	1	1	1	1

Këtu, në fakt, variablat të cilat figurojnë brenda mintermave janë zëvendësuar me shifrën binare **1**, kurse kovariablat - me shifrën binare **0**.

2. Mintermat, përkatësisht grupet e shifrave binare në tabelën e mësipërme, grupohen në bazë të numrit të shifrave **1** brenda tyre, duke filluar me grupet pa shifra **1**, ose me grupet me më pak shifra **1**.

	A	B	C	D	
0	0	0	0	0	√
1	0	0	0	1	√
2	0	0	1	0	√
8	1	0	0	0	√
5	0	1	0	1	√
9	1	0	0	1	√
10	1	0	1	0	√
7	0	1	1	1	√
13	1	1	0	1	√
15	1	1	1	1	√

3. Krahasohen mes veti mintermat e grupeve fqinjë dhe nëse dy minterma dallohen vetëm në një pozicion, ato shkrihen në një mintermë të përbashkët, duke shkruar **+** në pozicionin ku ato dallohen. Kështu, formohet tabela e re e mintermave, tek e cila figurojnë vetëm çiftet e mintermave që shkrihen mes vete. Njëkohësisht, në tabelën nga hapi paraprak mintermat që shkrihen shënohen me shenjën ✓. Pastaj, krahasimi vazhdon më tutje, duke marrë edhe simbolet **+** si elemente që gjithashtu krahasohen gjatë shkrirjes së mintermave dhe si rezultat fitohet tabela:

	A	B	C	D
0,1,8,9	+	0	0	+
0,2,8,10	+	0	+	0
0,8,1,9	+	0	0	+
0,8,2,10	+	0	+	0
1,5,9,13	+	+	0	1
1,9,5,13	+	+	0	1
5,7,13,15	+	1	+	1
5,13,7,15	+	1	+	1

Pasi në tabelë pa nevojë figurojnë minterma të njëjta, pas fshirjes së tyre tabela duket kështu:

	A	B	C	D
0,1,8,9	+	0	0	+
0,2,8,10	+	0	+	0
1,5,9,13	+	+	0	1
5,7,13,15	+	1	+	1

ku për mintermat e njëjta janë ruajtur vetëm mintermat që takohen së pari.

4. Meqë më nuk ka mundësi të shkrirjes së mintermave, procesi i krahasimit dhe i shkrirjes së tyre përfundon. Mintermat të cilat gjatë krahasimit dhe shkrirjes nuk janë shënuar me shenjën ✓, e formojn grupin e prim-implikantëve të funksionit. Kështu, pasi që nga këto minterma të fshihen elementet **+**, e në vend të shifrave binare të shkruhen variablat (për **1**) dhe kovariablat (për **0**) përkatëse, prim-implikantët e funksionit janë:

$$X = \bar{B}\,\bar{C}$$

$$Y = \bar{B}\,\bar{D}$$

$$Z = \bar{C}\,D$$

$$V = B\,D$$

II. Gjetja e prim-implikantëve esencialë

Pasi që në shprehjen minimale të funksionit duhet të figurojnë vetëm prim-implikantët esencialë, për gjetjen e tyre kalohet gjithashtu nëpër katër hapa karakteristikë.

1. Së pari formohet diagrami i prim-implikantëve me **i**-rreshta (sa ka prim-implikantë) dhe **j**-kolona (sa ka funksioni minterma m^1) kështu:

	$m_0=\bar{A}\bar{B}\bar{C}\bar{D}$	$m_1=\bar{A}\bar{B}\bar{C}D$	$m_2=\bar{A}\bar{B}C\bar{D}$	$m_5=\bar{A}B\bar{C}D$	$m_7=\bar{A}BCD$	$m_8=A\bar{B}\bar{C}\bar{D}$	$m_9=A\bar{B}\bar{C}D$	$m_{10}=A\bar{B}C\bar{D}$	$m_{13}=AB\bar{C}D$	$m_{15}=ABCD$
$X=\bar{B}\bar{C}$										
$Y=\bar{B}\bar{D}$										
$Z=\bar{C}D$										
$V=BD$										

2. Në fushat e prerjes së mintermave me prim-implikantët shënohet simboli **+**, nëse prim-implikantët i mbulojnë mintermat. Si rezultat, diagrami i prim-implikantëve duket kështu:

	$m_0=\bar{A}\bar{B}\bar{C}\bar{D}$	$m_1=\bar{A}\bar{B}\bar{C}D$	$m_2=\bar{A}\bar{B}C\bar{D}$	$m_5=\bar{A}B\bar{C}D$	$m_7=\bar{A}BCD$	$m_8=A\bar{B}\bar{C}\bar{D}$	$m_9=A\bar{B}\bar{C}D$	$m_{10}=A\bar{B}C\bar{D}$	$m_{13}=AB\bar{C}D$	$m_{15}=ABCD$
$X=\bar{B}\bar{C}$	+	+				+	+			
$Y=\bar{B}\bar{D}$	+		+			+		+		
$Z=\bar{C}D$		+		+			+		+	
$V=BD$				+	+				+	+

3. Në diagramin e fituar shihet se disa kolona në vete kanë vetëm nga një **+**, që d.m.th. se mintermat përkatëse mbulohen vetëm nga një prim-implikant. Prandaj, prim-implikantët përkatës janë esencialë dhe patjetër duhet të figurojnë në shprehjen minimale të funksionit. Të tillë janë prim-implikantët **Y** dhe **V**. Rreshtat të cilët u përkasin prim-implikantëve esencialë quhen *rreshta esencialë*. pasi që në diagramin e prim-implikantëve të fshihen kolonat e mintermave që mbulohen nga rreshtat esencialë, e këto janë kolonat **0**, **8** dhe **10** - për **Y**, si dhe kolonat **5**, **7**, **13** dhe **15** - për **V**, si rezultat fitohet diagrami i thjeshtuar:

	$m_1=\bar{A}\bar{B}\bar{C}D$	$m_9=A\bar{B}\bar{C}D$
$X=\bar{B}\bar{C}$	+	+
$Y=\bar{B}\bar{D}$		
$Z=\bar{C}D$	+	+
$V=BD$		

4. Pasi që nga prim-implikantët esencialë **Y** dhe **V** kanë ngelur të pambuluara vetëm kolonat e mintermave **1** dhe **9**, në diagramin e mësipërm shihet se këto dy kolona mund të mbulohen nga prim-implikanti **X**, ose **Z**.

Kështu, si shprehje minimale e funksionit mund të merret shprehja:

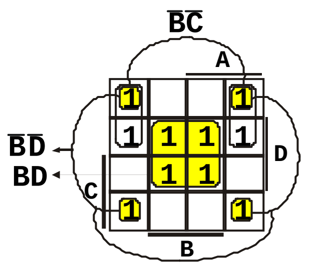
$$f = \bar{B}\,\bar{D} + B\,D + \bar{B}\,\bar{C}$$

ku dy prim-implikantëve esencialë **Y** dhe **V** u është shtuar edhe prim-implikanti **X**, ose shprehja:

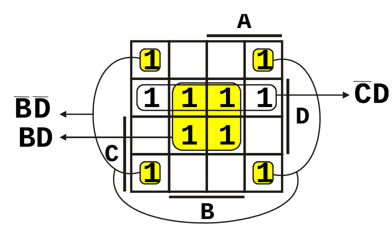
$$f = \overline{B}\,\overline{D} + B\,D + \overline{C}\,D$$

te e cila si prim-implikant esencial është marrë edhe prim-implikanti **Z**. Meqë prim-implikantët **X** dhe **Z** kanë madhësi të njëjta (kanë nga dy shkronja), të dy shprehjet e fituara janë minimale.

Për funksionin e dhënë, shprehje minimale të njëjta fitohen edhe nëse minimizimi bëhet në rrugë grafike kështu:



$$f = \overline{B}\overline{D} + BD + \overline{B}\,\overline{C}$$



$$f = \overline{B}\overline{D} + BD + \overline{C}D$$

Shembull

Gjetja e shprehjeve minimale të funksioneve:

a. $f(A, B, C, D) = \sum m^1(0, 1, 2, 5, 6, 7, 8, 9, 10, 14)$

b. $g(A, B, C, D) = \sum m^1(0, 1, 5, 7, 8, 10, 14, 15)$

duke shfrytëzuar procedurën e minimizimit tabelar.

a.

I. Gjetja e prim-implikantëve

	A	B	C	D
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
14	1	1	1	0

	A	B	C	D	
0	0	0	0	0	√
1	0	0	0	1	√
2	0	0	1	0	√
8	1	0	0	0	√
5	0	1	0	1	√
6	0	1	1	0	√
9	1	0	0	1	√
10	1	0	1	0	√
7	0	1	1	1	√
14	1	1	1	0	√

	A	B	C	D	
0, 1	0	0	0	+	√
0, 2	0	0	+	0	√
0, 8	+	0	0	0	√
1, 5	0	+	0	1	X
1, 9	+	0	0	1	√
2, 6	0	+	1	0	√
2, 10	+	0	1	0	√
8, 9	1	0	0	+	√
8, 10	1	0	+	0	√
5, 7	0	1	+	1	Y
6, 7	0	1	1	+	Z
6, 14	+	1	1	0	√
10, 14	1	+	1	0	√

	A	B	C	D
0, 1, 8, 9	+	0	0	+
0, 2, 8, 10	+	0	+	0
0, 8, 1, 9	+	0	0	+
0, 8, 2, 10	+	0	+	0
2, 6, 10, 14	+	+	1	0
2, 10, 6, 14	+	+	1	0

	A	B	C	D	
0, 1, 8, 9	+	0	0	+	V
0, 2, 8, 10	+	0	+	0	W
2, 6, 10, 14	+	+	1	0	T

Prim-implikantët:

$$X = \overline{A}\,\overline{C}D \qquad V = \overline{B}\,\overline{C}$$

$$Y = \overline{A}BD \qquad W = \overline{B}\,\overline{D}$$

$$Z = \overline{A}BC \qquad T = C\overline{D}$$

II. Gjetja e prim-implikantëve esencialë

	$m_0=\bar{A}\bar{B}\bar{C}\bar{D}$	$m_1=\bar{A}\bar{B}\bar{C}D$	$m_2=\bar{A}\bar{B}C\bar{D}$	$m_5=\bar{A}B\bar{C}D$	$m_6=\bar{A}BC\bar{D}$	$m_7=\bar{A}BCD$	$m_8=A\bar{B}\bar{C}\bar{D}$	$m_9=A\bar{B}\bar{C}D$	$m_{10}=A\bar{B}C\bar{D}$	$m_{14}=ABC\bar{D}$
$X=\bar{A}\bar{C}D$										
$Y=\bar{A}BD$				+		+				
$Z=\bar{A}BC$					+	+				
$V=\bar{B}\bar{C}$	+	+					+	+		
$W=\bar{B}\bar{D}$	+		+				+		+	
$T=C\bar{D}$			+		+				+	+

Pasi që në kolonat e mintermave m_9 dhe m_{14} ka vetëm nga një **+**, prim-implikantët e rreshtave përkatës (**V** dhe **T**) janë prim-implikantë esencialë. Rreshtat esencialë, të cilët u përkasin implikantëve **V** dhe **T**, i mbulojnë edhe mintermat:

m_0, m_1, m_8

dhe

m_2, m_6, m_{10}

Për këtë arsye, kolonat përkatëse në diagramin e prim-implikantëve mund të fshihen. Përfundimisht, diagrami duket kështu:

	$m_5=\bar{A}B\bar{C}D$	$m_7=\bar{A}BCD$
$X=\bar{A}\bar{C}D$	+	
$Y=\bar{A}BD$	+	+
$Z=\bar{A}BC$		+
$V=\bar{B}\bar{C}$		
$W=\bar{B}\bar{D}$		
$T=C\bar{D}$		

Nga diagrami shihet se mintermat m_5 dhe m_7 mund të mbulohen me prim-implikantin **Y**, prandaj edhe ai merret si prim-implikant esencial. Përfundimisht, shprehja minimale e funksionit të dhënë është:

$$f = \bar{B}\,\bar{C} + C\,\bar{D} + \bar{A}BD$$

b.

I. Gjetja e prim-implikantëve

	A	B	C	D
0	0	0	0	0
1	0	0	0	1
5	0	1	0	1
7	0	1	1	1
8	1	0	0	0
10	1	0	1	0
14	1	1	1	0
15	1	1	1	1

	A	B	C	D	
0	0	0	0	0	√
1	0	0	0	1	√
8	1	0	0	0	√
5	0	1	0	1	√
10	1	0	1	0	√
7	0	1	1	1	√
14	1	1	1	0	√
15	1	1	1	1	√

	A	B	C	D	
0, 1	0	0	0	+	p_1
0, 8	+	0	0	0	p_2
1, 5	0	+	0	1	p_3
8, 10	1	0	+	0	p_4
5, 7	0	1	+	1	p_5
10, 14	1	+	1	0	p_6
7, 15	+	1	1	1	p_7
14, 15	1	1	1	+	p_8

Prim-implikantët:

$$p_1 = \overline{A}\,\overline{B}\,\overline{C}$$
$$p_2 = \overline{B}\,\overline{C}\,\overline{D}$$
$$p_3 = \overline{A}\,\overline{C}\,D$$
$$p_4 = A\,\overline{B}\,\overline{D}$$
$$p_5 = \overline{A}\,B\,D$$
$$p_6 = A\,C\,\overline{D}$$
$$p_7 = B\,C\,D$$
$$p_8 = A\,B\,C$$

II. Gjetja e prim-implikantëve esencialë

	$m_0=\bar{A}\bar{B}\bar{C}\bar{D}$	$m_1=\bar{A}\bar{B}\bar{C}D$	$m_5=\bar{A}B\bar{C}D$	$m_7=\bar{A}BCD$	$m_8=A\bar{B}\bar{C}\bar{D}$	$m_{10}=A\bar{B}C\bar{D}$	$m_{14}=ABC\bar{D}$	$m_{15}=ABCD$
$p_1=\bar{A}B\bar{C}$	**+**	**+**						
$p_2=\bar{B}\bar{C}\bar{D}$	+				+			
$p_3=\bar{A}\bar{C}D$		+	+					
$p_4=A\bar{B}\bar{D}$					+	+		
$p_5=\bar{A}BD$			+	+				
$p_6=AC\bar{D}$						+	+	
$p_7=BCD$				+				+
$p_8=ABC$							+	+

Meqë nga tabela e dhënë nuk veçohet asnjë prim-implikant esencial, arbitrarisht mund të zgjidhet njëri prej tyre (të gjithë kanë madhësi të njëjtë dhe mbulojnë nga dy kolona). Nëse, p.sh. zgjidhet si esencial prim-implikanti i parë, e meqë ai i mbulon mintermat m_0 dhe m_1, pasi të fshihen kolonat përkatëse, tabela duket kështu:

	$m_5=\bar{A}B\bar{C}D$	$m_7=\bar{A}BCD$	$m_8=A\bar{B}\bar{C}\bar{D}$	$m_{10}=A\bar{B}C\bar{D}$	$m_{14}=ABC\bar{D}$	$m_{15}=ABCD$
$p_1=\bar{A}\bar{B}C$						
$p_2=\bar{B}\bar{C}\bar{D}$			+			
$p_3=\bar{A}\bar{C}D$	+					
$p_4=A\bar{B}\bar{D}$			**+**	**+**		
$p_5=\bar{A}BD$	+	+				
$p_6=AC\bar{D}$				+	+	
$p_7=BCD$		+				+
$p_8=ABC$					+	+

Edhe nga kjo tabelë nuk mund të veçohet asnjë prim-implikant esencial. Por, njëlloj si edhe më lart mund të zgjidhet njëri prej tyre, p.sh., p_4 (ai takohet i pari dhe njëkohësisht i mbulon dy kolona). Pastaj, njëlloj veprohet edhe në dy hapat vijues, për t'i mbuluar përfundimisht të gjitha mintermat e funksionit të dhënë.

	m_5	m_7	m_{14}	m_{15}
$p_1=\overline{A}\,\overline{B}\,\overline{C}$				
$p_2=\overline{B}\,\overline{C}\,\overline{D}$				
$p_3=\overline{A}\,\overline{C}D$	+			
$p_4=A\overline{B}\,\overline{D}$				
$p_5=\overline{A}BD$	+	+		
$p_6=AC\overline{D}$			+	
$p_7=BCD$		+		+
$p_8=ABC$			+	+

	m_{14}	m_{15}
$p_1=\overline{A}\,\overline{B}\,\overline{C}$		
$p_2=\overline{B}\,\overline{C}\,\overline{D}$		
$p_3=\overline{A}\,\overline{C}D$		
$p_4=A\overline{B}\,\overline{D}$		
$p_5=\overline{A}BD$		
$p_6=AC\overline{D}$	+	
$p_7=BCD$		+
$p_8=ABC$	+	+

Gjatë rrugës së kaluar më sipër, si prim-implikant esencial janë zgjedhur prim-implikantët: p_1, p_4, p_5 dhe p_8. Kështu, shprehja minimale e funksionit të dhënë është:

$$g = \overline{A}\,\overline{B}\,\overline{C} + A\,\overline{B}\,\overline{D} + \overline{A}\,B\,D + A\,B\,C$$

Në procesin e grupimit dhe të shkrirjes së mintermave, bashkë me mintermat m^1, mund të marrin pjesë edhe mintermat m^+.

Shembull

Gjetja e shprehjes minimale të funksionit:

$$f(A,B,C,D) = \sum m^1(2,3,7,9,11,13) + \sum m^+(1,10,15)$$

përmes metodës së minimizimit tabelar.

I. Gjetja e prim-implikantëve

	A	B	C	D
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
7	0	1	1	1
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
13	1	1	0	1
15	1	1	1	1

	A	B	C	D	
1	0	0	0	1	√
2	0	0	1	0	√
3	0	0	1	1	√
9	1	0	0	1	√
10	1	0	1	0	√
7	0	1	1	1	√
11	1	0	1	1	√
13	1	1	0	1	√
15	1	1	1	1	√

	A	B	C	D	
1, 3	0	0	1	1	√
1, 9	+	0	0	1	√
2, 3	0	0	1	+	√
2, 10	+	0	1	0	√
3, 7	0	+	1	1	√
3, 11	+	0	1	1	√
9, 11	1	0	+	1	√
9, 13	1	1	0	1	√
10, 13	1	0	1	+	√
7, 15	1	1	1	1	√
11, 15	1	1	1	1	√
13, 15	1	1	1	1	√

	A	B	C	D	
1, 3, 9, 11	+	0	+	1	X
2, 3, 10, 11	+	0	1	+	Y
3, 7, 11, 15	+	+	1	1	Z
9, 11, 13, 15	1	+	+	1	V

Prim-implikantët:

$$\mathbf{X} = \overline{\mathbf{B}}\mathbf{D} \qquad \mathbf{Z} = \mathbf{CD}$$

$$\mathbf{Y} = \overline{\mathbf{B}}\mathbf{C} \qquad \mathbf{V} = \mathbf{AD}$$

II. Gjetja e prim-implikantëve esencialë

Këtu, diagrami i prim-implikantëve formohet vetëm nga kolonat e mintermave $\mathbf{m}^1$, duke mos i marrë mintermat $\mathbf{m}^+$.

	$m_2 = \overline{A}\,\overline{B}C\overline{D}$	$m_3 = \overline{A}\,\overline{B}CD$	$m_7 = \overline{A}BCD$	$m_9 = A\overline{B}\,\overline{C}D$	$m_{11} = A\overline{B}CD$	$m_{13} = AB\overline{C}D$
$X = \overline{B}D$		+		+	+	
$Y = \overline{B}C$	+	+			+	
$Z = CD$		+	+		+	
$V = AD$				+	+	+

Në këtë rast prim-implikantët esencialë **Y**, **Z** dhe **V**, përveç mintermave $\mathbf{m}_1$, $\mathbf{m}_7$ dhe $\mathbf{m}_{13}$, njëkohësisht i mbulojnë edhe mintermat e tjera ($\mathbf{m}_3$, $\mathbf{m}_9$ dhe $\mathbf{m}_{11}$), prandaj shprehja minimale e funksionit është:

$$\mathbf{f} = \overline{\mathbf{B}}\mathbf{C} + \mathbf{CD} + \mathbf{AD}$$

Për fillimin e minimizimit tabelar, funksioni duhet të jepet si shumë e mintermave. Por, nëse funksioni është dhënë ndryshe, para fillimit të procesit të minimizimit, duhet të shndërrohet në këtë formë.

4

Qarqet kombinuese

Qarqet logjike mund të jenë *qarqe kombinuese* dhe *qarqe sekuenciale*. Qarqet te të cilat vlerat dalëse varen vetëm nga vlerat hyrëse quhen *qarqe logjike kombinuese*, ose shkurt - *qarqe kombinuese* (ang. combinational circuit). Te *qarqet sekuenciale* (ang. sequential circuit) vlerat në daljet e tyre, përveç prej vlerave hyrëse, varen edhe nga *gjendjet e elementeve memoruese* të cilat gjenden në përbërje të qarqeve.

Në çdo qark kombinues dallohen *variablat hyrëse*, *elementet logjike* dhe *variablat dalëse*. Informatat që i jepen qarkut përmes sinjaleve përkatëse në hyrjet e tij, merren në daljet e veçanta, të përpunuara përmes elementeve logjike.

Në formë të përgjithshme, qarku kombinues me **m**-hyrje dhe **n**-dalje mund të paraqitet si në Fig.4.1, ku x_1, x_2, ..., x_m janë variablat hyrëse, kurse me z_1, z_2, ..., z_n janë shënuar variablat, përkatësisht funksionet dalëse. Përmes **m**-hyrjeve qarkut i jepen gjithsej 2^m kombinime të vlerave të ndryshme, kurse përshkrimi i plotë i tij mund të bëhet përmes **n**-funksioneve logjike, aq sa ka qarku dalje. Për çdo kombinim të vlerave hyrëse në qark ekziston një dhe vetëm një kombinim i vlerave në dalje të tij.

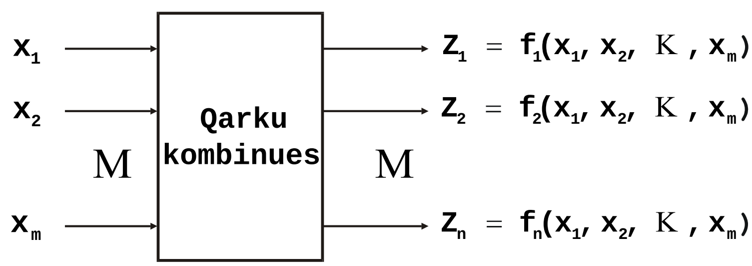


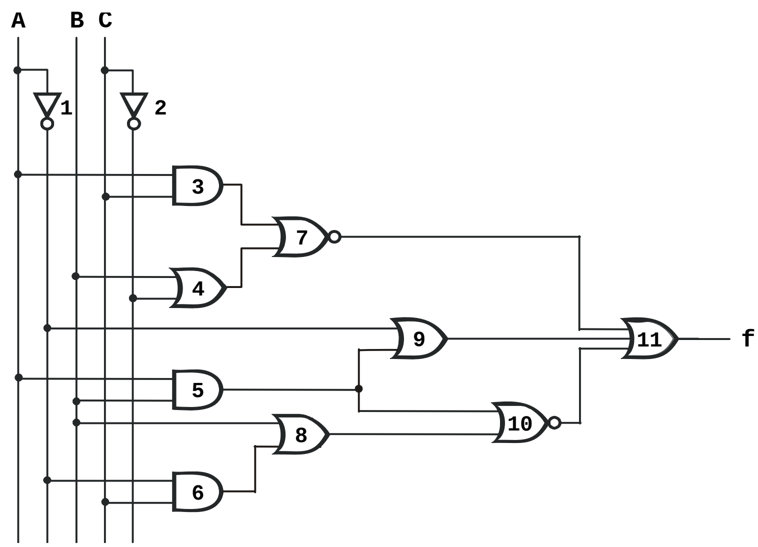
Fig.4.1 Forma e përgjithshme e qarkut kombinues

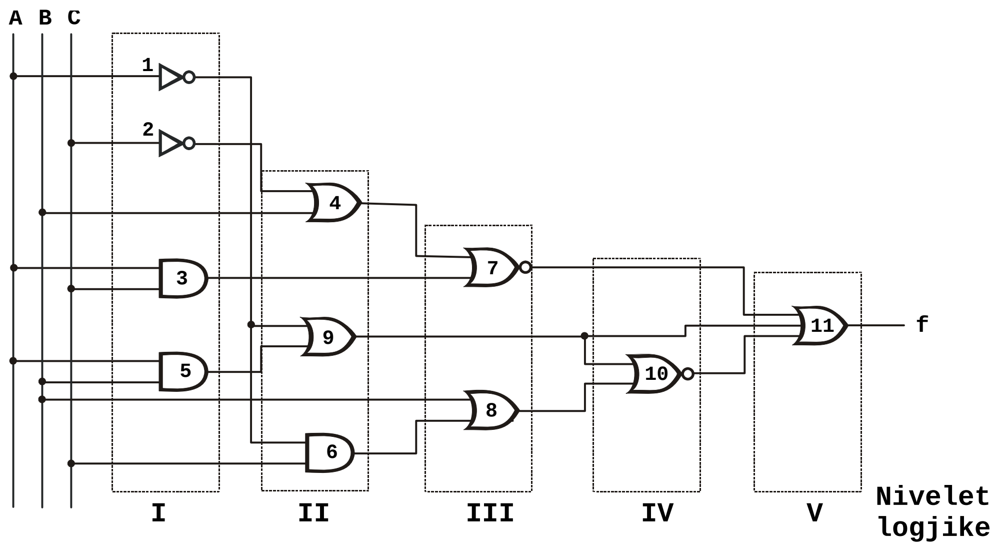
Nivelet logjike

Elementet logjike, të cilët përfshihen brenda një qarku logjik, mund të grupohen në *nivele logjike* (ang. logic level), duke shkuar prej hyrjes kah dalja e tij. *Nivelit të parë logjik* i përkasin të gjitha elementet, në hyrjet e të cilave aplikohen nga jashtë vetëm vlerat hyrëse të qarkut. Elementet logjike që kanë së paku një hyrje, nga daljet e elementeve të nivelit të parë logjik, formojnë *nivelin e dytë logjik*. Në këtë mënyrë përcaktohen edhe elementet të cilat marrin pjesë në nivelin *e tretë* logjik, ose edhe në nivelet logjike më të larta.

Shembull

Gjetja e niveleve logjike për qarkun logjik vijues.





Në praktikë tentohet që qarqet logjike të kenë sa më pakë nivele logjike, sepse zvogëlohet mundësia e pengesave, si dhe dobësimi i sinjaleve është më i vogël.

Analiza

Me analizën e një qarku kombinues nënkuptohet *procedura përmes së cilës gjenden funksionet logjike të daljeve të veçanta të qarkut*. Kjo bëhet me qëllim të zbulimit të asaj se si punon qarku i dhënë, përkatësisht verifikimit të funksioneve në bazë të së cilave ai është realizuar.

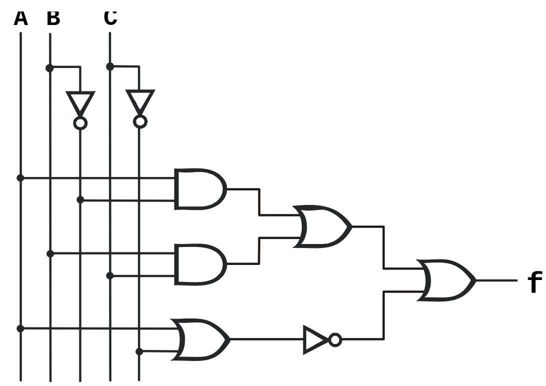
Analiza e një qarku fillon me përcaktimin e niveleve logjike, përkatësisht grupimin e elementeve logjike në këto nivele. Pastaj, duke shkuar prej hyrjeve të qarkut, në daljet e elementeve të veçanta shënohen shprehjet algjebrike të funksioneve përkatëse. Kështu arrihet deri tek elementet e nivelit të fundit, me ç'rast fitohen shprehjet e funksioneve në daljet e veçanta të qarkut dhe analiza përfundon.

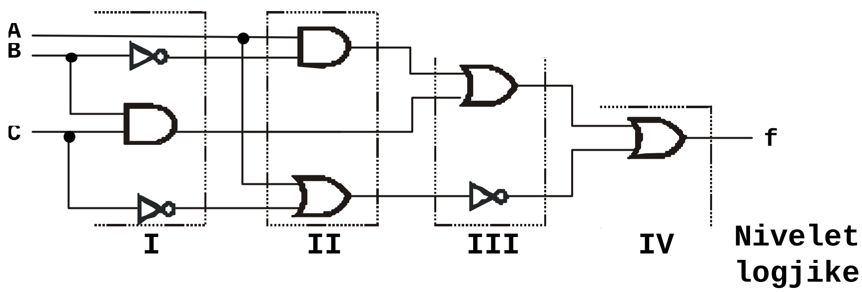
Qarqet me elemente logjike themelore

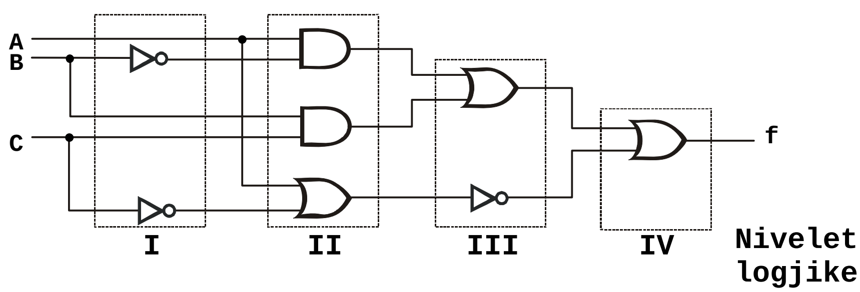
Analiza e qarqeve të cilat përmbajnë vetëm elemente logjike themelore rrjedh thjesht, pa bërë ndryshime në elementet e qarqeve, gjë që dallon nga qarqet që kanë edhe elemente logjike universale.

Shembull

Gjetja e funksionit logjik me të cilin përshkruhet qarku vijues.







$$f = A\overline{B} + BC + \overline{A + C}$$

Qarqet me elemente logjike universale

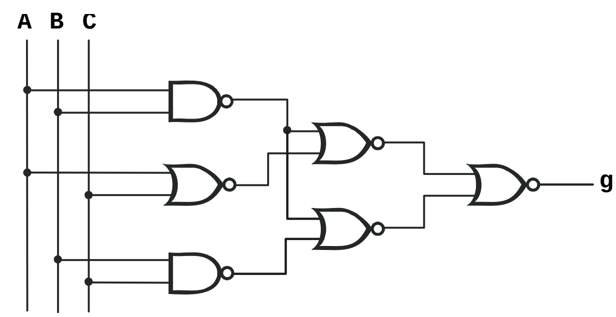
Analiza e qarqeve me elemente logjike universale mund të bëhet në rrugë direkte ose indirekte.

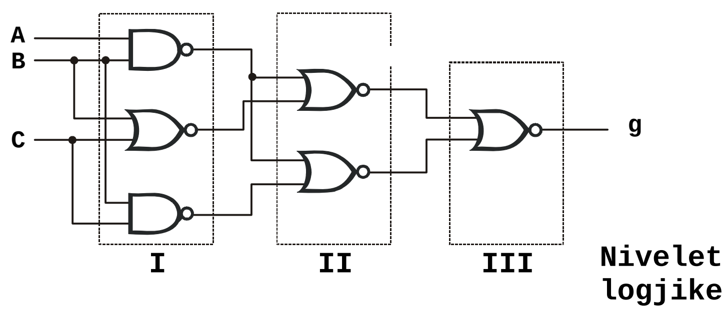
Analiza direkte

Me analizën direkte nënkuptohet rruga e dhënë më sipër gjatë analizës së qarqeve me elemente logjike themelore.

Shembull

Analiza direkte e qarkut me elemente logjike universale:

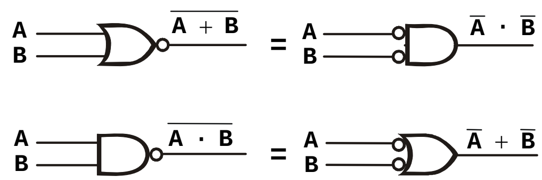




$$\mathbf{g} = \overline{\overline{AB} + \overline{A + C} + \overline{AB} + \overline{BC}}$$
$$= \overline{AB}$$

Analiza indirekte

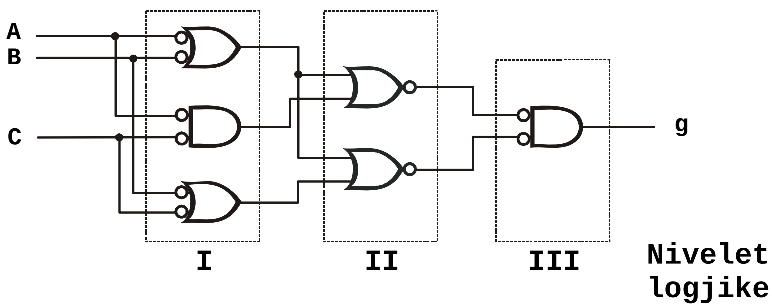
Gjatë analizës indirekte së pari bëhet zëvendësimi i elementeve logjike universale në nivelet teke, me elemente logjike speciale, të krijuara në bazë të ligjeve të *De Morganit*, ashtu siç është treguar në vijim.

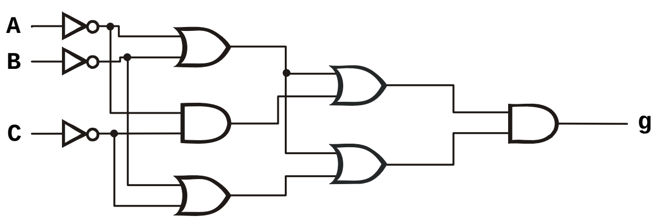


Pastaj, në qark eliminohen inversionet e dyfishta dhe para se të shkruhen shprehjet e funksioneve në daljet e elementeve të veçanta qarku vizatohet edhe njëherë përmes elementeve logjike themelore.

Shembull

Analiza indirekte e qarkut me elemente logjike universale, i cili u dha në shembullin paraprak.





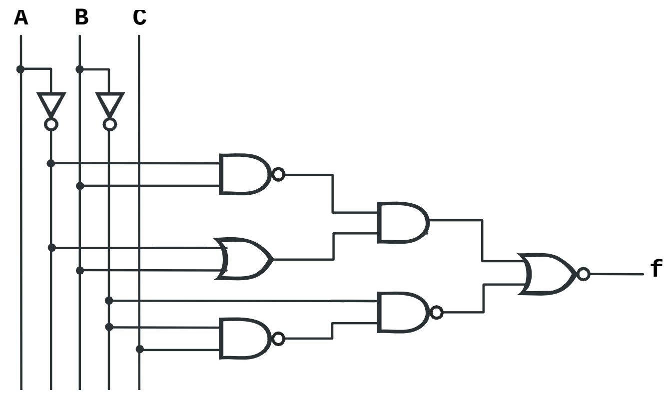
$$\begin{aligned} g &= (\overline{A} + \overline{B} + \overline{A}\overline{C}) \cdot (\overline{A} + \overline{B} + \overline{C}) \\ &= \overline{A} + \overline{B} \end{aligned}$$

Qarqet me elemente logjike të përziera

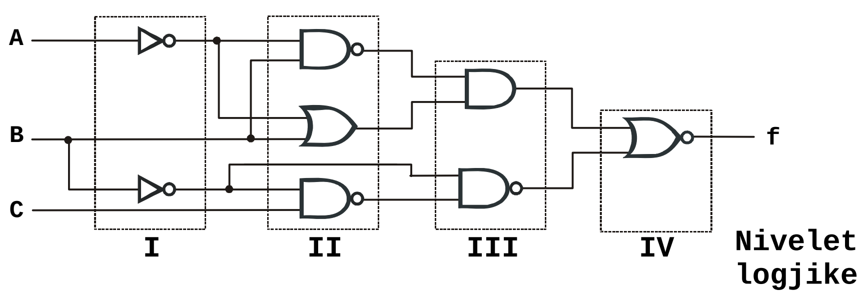
Gjatë analizës së qarqeve me elemente logjike të përziera, përkatësisht me elemente logjike themelore dhe universale, mund të përdoret njëra nga mënyrat e përmendura gjatë analizës së qarqeve me elemente logjike universale.

Shembull

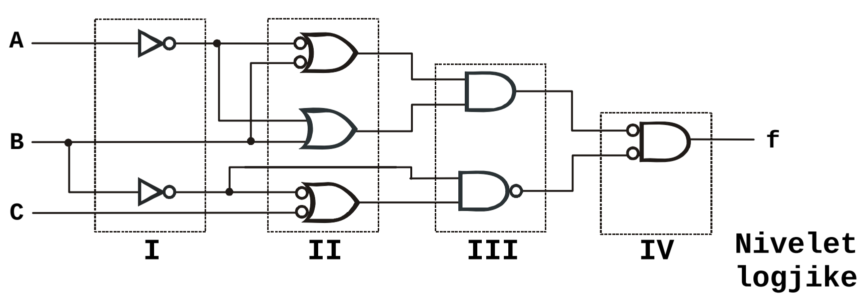
Analiza e qarkut me elemente logjike të përziera:

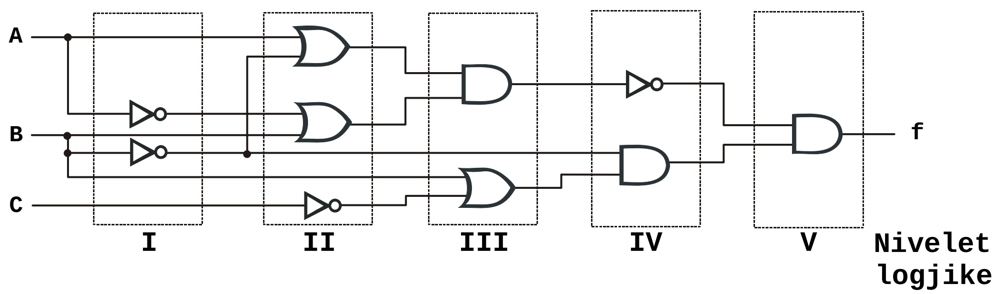


Analiza direkte



Analiza indirekte





Nëse qarku përmban edhe elemente logjike speciale, analiza e tij rrjedh ashtu siç u dha më sipër, nëse së pari elementet logjike speciale zëvendësohen me elemente logjike themelore. Kurse, nëse qarku përmban më shumë dalje, procedura e gjetjes së shprehjeve algjebrike të funksioneve dalëse nuk ndryshon aspak nga ajo që përdoret për qarqet me një dalje.

Sinteza

Me sintezën e një qarku logjik nënkuptohet procedura e projektimit të tij në bazë të funksionit logjik përkatës. Gjatë kësaj, me qëllim që realizimi i qarkut të jetë sa më i thjeshtë, përkatësisht të përdoren sa më pak elemenete logjike, gjë që i zvogëlon shpenzimet dhe punën gjatë realizimit praktik të tij, gjenden format minimale të funksioneve që e përshkruajnë qarkun.

Për realizimin e qarkut mund të përdoren vetëm elemente logjike themelore, ose vetëm elemente logjike universale, ose elemente logjike të përziera - përfshirë edhe elementet logjike speciale.

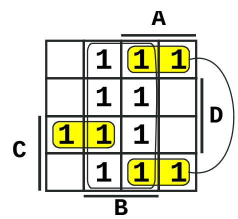
Shembull

Realizimi i qarkut logjik i cili përshkruhet përmes funksionit:

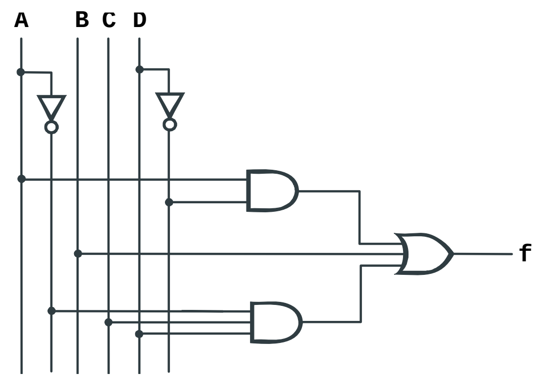
$$f = AB + BCD + A\bar{B}\bar{D} + \bar{A}CD + \bar{A}B$$

duke e gjetur shprehjen minimale të tij, në formën e tij disjunktive dhe konjuktive.

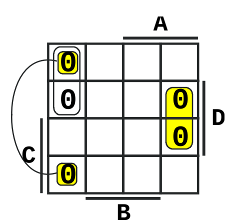
Forma disjunktive



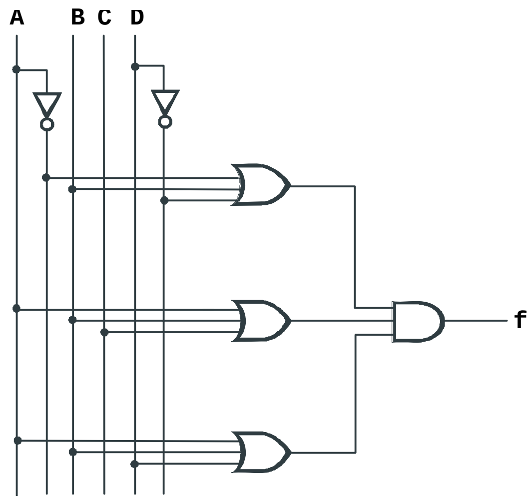
$$f = B + A\overline{D} + \overline{A}CD$$



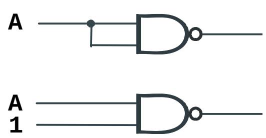
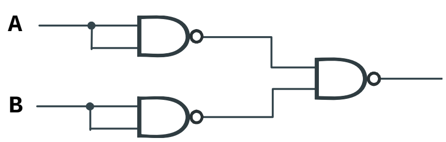
Forma konjuktive



$$f = (\overline{A} + B + \overline{D})(A + B + C)(A + B + D)$$



Operacionet logjike elementare **JO**, **OSE** e **DHE** mund të realizohen edhe duke shfrytëzuar elementet logjike universale, ashtu siç është treguar në tabelat e dhëna në Fig.4.2.

Operacioni	Realizimi përmes elementit JODHE
JO	 $\overline{A \cdot A} = \overline{A} + \overline{A} = \overline{A}$ $\overline{A \cdot 1} = \overline{A} + \overline{1} = \overline{A} + 0 = \overline{A}$
OSE	 $\overline{\overline{A} \cdot \overline{B}} = \overline{\overline{A}} + \overline{\overline{B}} = A + B$
DHE	 $\overline{\overline{AB}} = AB$

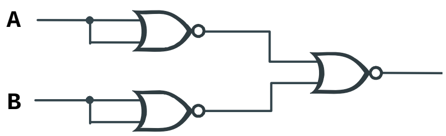
Operacioni	Realizimi përmes elementit JOOSE	
JO		$\overline{A+A} = \overline{A}\cdot\overline{A} = \overline{A}$
		$\overline{A+0} = \overline{A}\cdot\overline{0} = \overline{A}$
OSE		$\overline{\overline{A+B}} = A+B$
DHE		$\overline{\overline{A}+\overline{B}} = \overline{\overline{A}}\cdot\overline{\overline{B}} = A\cdot B$

Fig.4.2 Realizimi i funksioneve elementare përmes elementeve logjike universale

Në praktikë, me qëllim të shfrytëzimit sa më racional të qarqeve të integruara të cilat përmbajnë elemente logjike, preferohet realizimi i qarqeve logjike, duke shfrytëzuar vetëm një lloj elementesh logjike universale. Gjatë kësaj, realizimi i qarqeve fillon me komplementimin e dyfishtë të shprehjeve minimale të funksioneve, në fomën e tyre disjunktive dhe konjuktive.

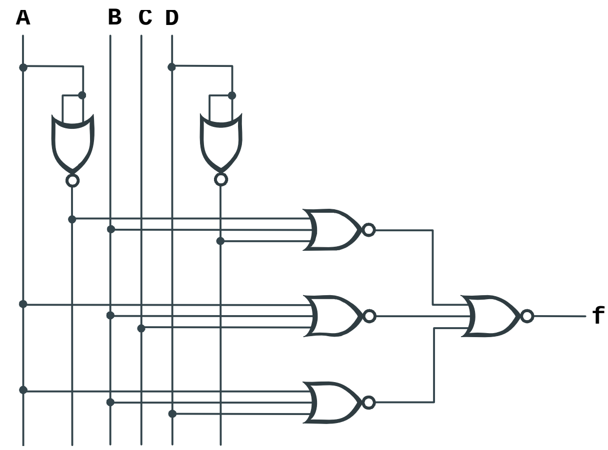
Shembull

Realizimi i qarkut logjik i cili përshkruhet përmes funksionit të dhënë në shembullin paraprak, duke shfrytëzuar format e gjetura minimale, si dhe elementet logjike universale **JOOSE** e **JODHE**.

Realizimi përmes elementeve JOOSE

$$f=\overline{\overline{(\overline{A}+B+\overline{D}))\cdot(A+B+C)\cdot(A+B+D)}}$$

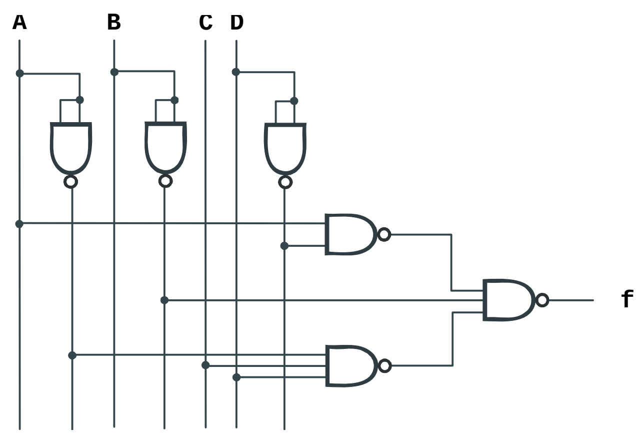
$$=\overline{\overline{(\overline{A}+B+\overline{D})}+\overline{(A+B+C)}+\overline{(A+B+D)}}$$



Realizimi përmes elementeve JODHE

$$f = \overline{\overline{B + A\bar{D} + \bar{A}CD}}$$

$$= \bar{B} \cdot \overline{A\bar{D}} \cdot \overline{\bar{A}CD}$$



Procedura e realizimit të qarqeve nuk ndryshon aspak nga ajo që u dha më sipër, nëse funksionet logjike përmes së cilave bëhet përshkrimi i tyre kanë edhe vlera arbitrare.

Shembull

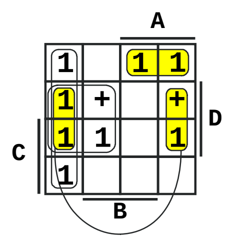
Realizimi i qarkut logjik i cili përshkruhet përmes funksionit:

$$f(A, B, C, D) = \sum m^1(0, 1, 2, 7, 8, 11, 12) + \sum m^+(3, 5, 9)$$

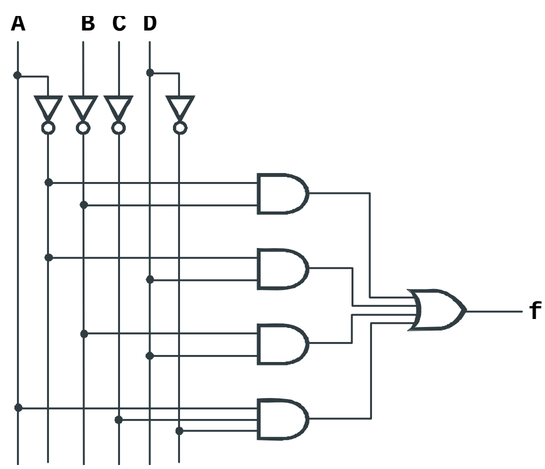
duke shfrytëzuar elemente logjike themelore dhe elemente logjike universale.

Me elemente logjike themelore

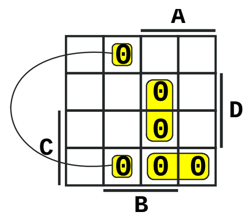
a.



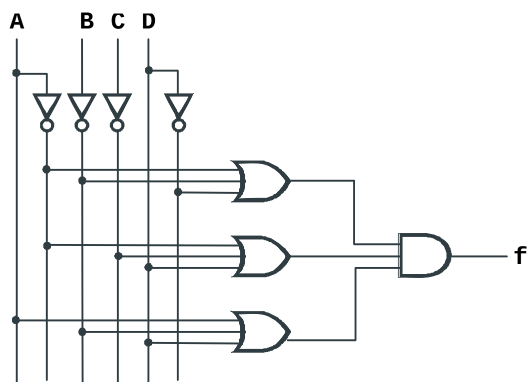
$$f = \bar{A}\,\bar{B} + \bar{A}\,D + \bar{B}\,D + A\,\bar{C}\,\bar{D}$$



b.



$$f = (\bar{A} + \bar{B} + \bar{D}) \cdot (\bar{A} + \bar{C} + D) \cdot (A + \bar{B} + D)$$

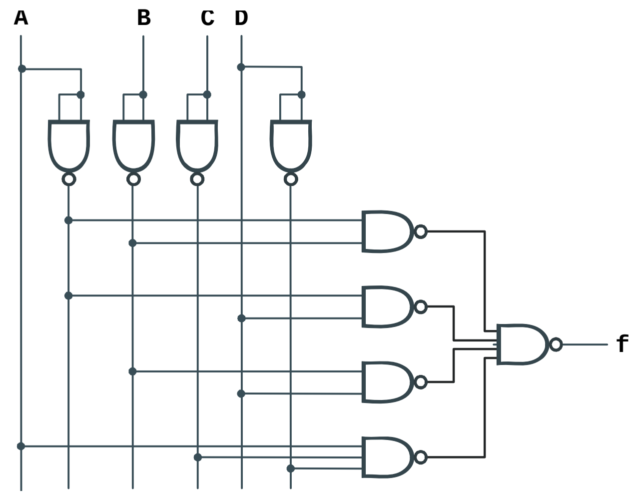


Me elemente logjike universale

a. **JODHE**

$$f = \overline{\overline{\bar{A}\,\bar{B} + \bar{A}\,D + \bar{B}\,D + A\,\bar{C}\,\bar{D}}}$$

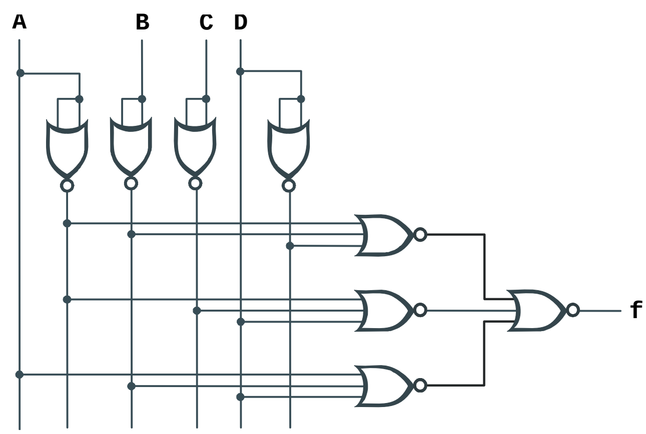
$$= \overline{\overline{\bar{A}\,\bar{B}} \cdot \overline{\bar{A}\,D} \cdot \overline{\bar{B}\,D} \cdot \overline{A\,\bar{C}\,\bar{D}}}$$



b. **JOOSE**

$$f = \overline{\overline{(\overline{A} + \overline{B} + \overline{D}) \cdot (\overline{A} + \overline{C} + D) \cdot (A + \overline{B} + D)}}$$

$$= \overline{\overline{(\overline{A} + \overline{B} + \overline{D})} + \overline{(\overline{A} + \overline{C} + D)} + \overline{(A + \overline{B} + D)}}$$



Nëse qarku i cili realizohet ka më shumë dalje, përkatësisht nëse ai përshkruhet me më shumë funksione, për secilin funksion gjendet shprehja minimale përkatëse dhe në bazë të tyre edhe vizatohet qarku përkatës.

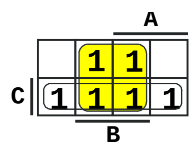
Shembull Realizimi i qarkut logjik i cili përshkruhet përmes funksioneve:

$$f = AB\overline{C} + \overline{A}C + B$$
$$g = A\overline{C} + AB + \overline{B}C$$

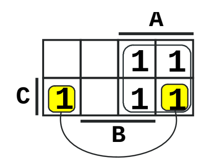
duke shfrytëzuar elemente logjike themelore dhe elemente logjike universale.

Me elemente logjike themelore

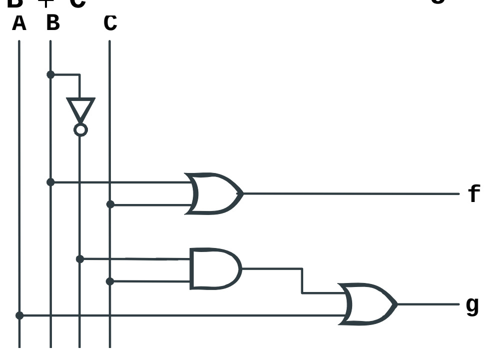
a.



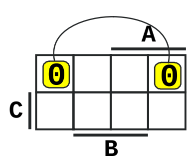
$f = B + C$



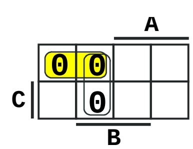
$g = A + \overline{B}C$



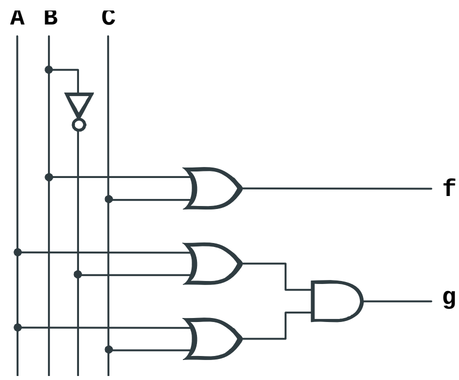
b.



$f = B + C$



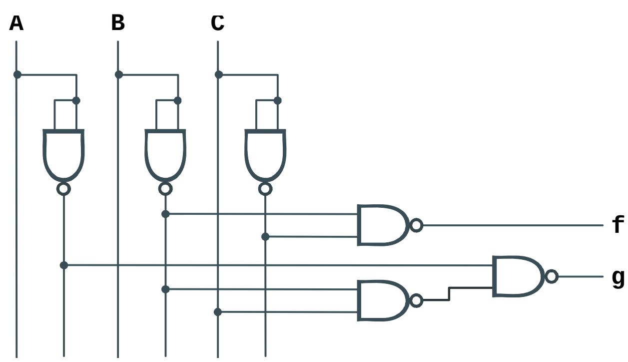
$g = (A + \overline{B})(A + C)$



Me elemente logjike universale

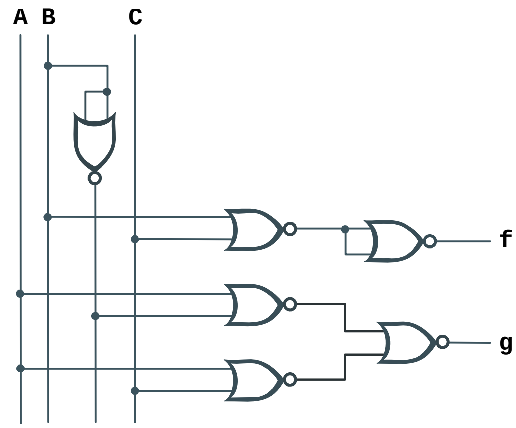
a. **JODHE**

$$
\begin{aligned}
\mathtt{f} &= \overline{\overline{\mathtt{B} + \mathtt{C}}} \\
&= \overline{\overline{\mathtt{B}} \cdot \overline{\mathtt{C}}} \\
\mathtt{g} &= \overline{\overline{\mathtt{A} + \overline{\mathtt{B}}\mathtt{C}}} \\
&= \overline{\overline{\mathtt{A}} \cdot \overline{\overline{\mathtt{B}}\mathtt{C}}}
\end{aligned}
$$



b. **JOOSE**

$$
\begin{aligned}
f &= \overline{\overline{B + C}} \\
g &= \overline{\overline{(A + \overline{B}) \cdot (A + C)}} \\
&= \overline{\overline{(A + \overline{B})} + \overline{(A + C)}}
\end{aligned}
$$



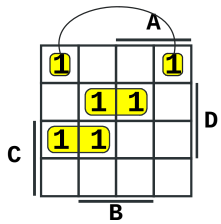
Shprehjet minimale të funksioneve logjike përmes së cilave përshkruhet qarku mund të gjenden ashtu që të kenë formë të njëjtë (disjunktive ose konjuktive), me qëllim që gjatë realizimit të qarkut të mund të shfrytëzohen komponentet e njëjta brenda funksioneve.

Shembull

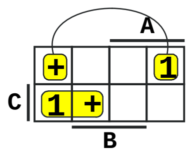
Realizimi i qarkut logjik i cili përshkruhet përmes funksioneve:

$$
\begin{aligned}
&f(A,B,C,D) = \sum m^1(0,3,5,7,8,13) \\
&g(A,B,C) = \sum m^1(1,4) + \sum m^+(0,3) \\
&h(A,B,C,D) = \prod M^0(0,1,4,8,9,10,11,12,14,15)
\end{aligned}
$$

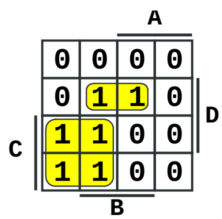
duke shfrytëzuar elemente logjike themelore.



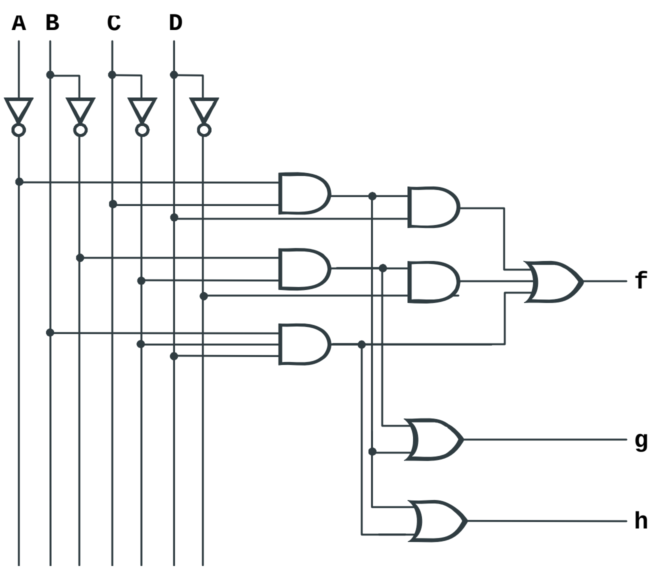
$$f = \overline{A}\,C\,D + B\,\overline{C}\,D + \overline{B}\,\overline{C}\,\overline{D}$$



$$g = \overline{A}C + \overline{B}\,\overline{C}$$



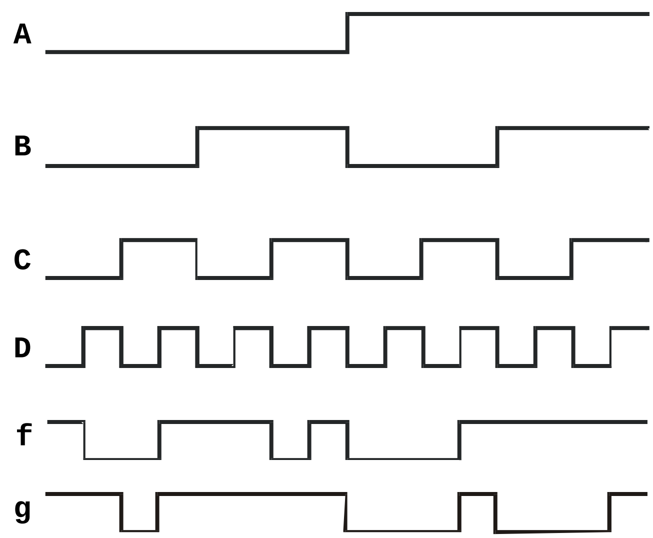
$$h = \overline{A}\,C + B\,\overline{C}\,D$$



Funksionet e qarkut që duhet të realizohet mund të jepen edhe përmes diagrameve kohore.

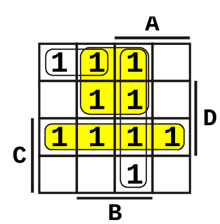
Shembull

Realizimi i qarkut logjik me katër hyrje **A**, **B**, **C** e **D** dhe dy dalje **f** e **g**, nëse qarku përshkruhet përmes diagrameve kohore vijuese:

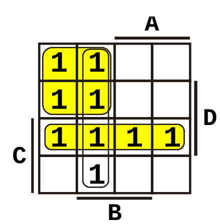


$$f = \sum m^1(0,3,4,5,7,11,12,13,14,15)$$

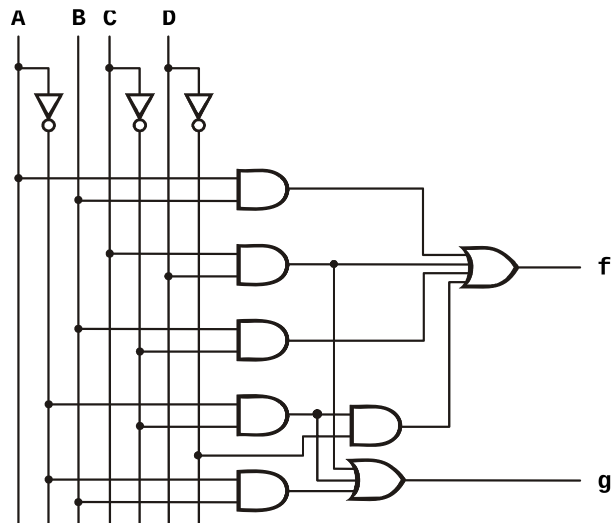
$$g = \sum m^1(0,1,3,4,5,6,7,11,15)$$



$$f = AB + CD + B\overline{C} + \overline{A}\,\overline{C}\,\overline{D}$$



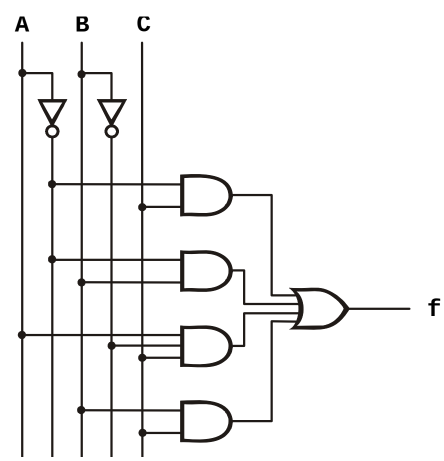
$$g = \overline{A}B + \overline{A}\,\overline{C} + CD$$



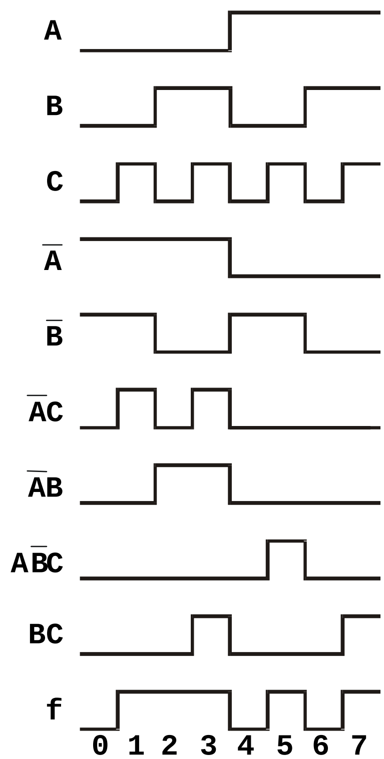
Nëse në vend të funksioneve me të cilat përshkruhet qarku logjik, jepet vizatimi i qarkut që duhet të realizohet, me qëllim të gjetjes së formës minimale të tij, duhet të minimizohen funksionet që fitohen si rezultat i analizës së qarkut.

Shembull

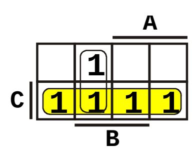
Gjetja e formës minimale të qarkut:



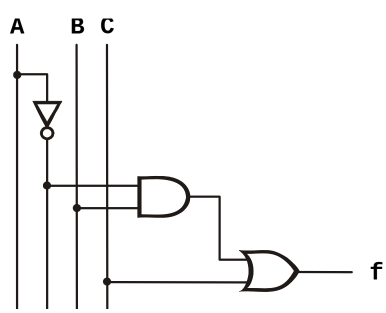
duke vizatuar fillimisht diagramet kohore në daljet e elementeve të veçanta, për të gjitha kombinimet e vlerave hyrëse.



$$f = \sum m^{1}(1, 2, 3, 5, 7)$$



$$f = \overline{A}B + C$$



Numri i hyrjeve dhe i shkronjave

Gjatë realizimit praktik të qarkut shumë ka rëndësi *numri i shkronjave* në funksionin me të cilin përshkruhet qarku, si dhe *numri i hyrjeve* në elementet që e formojnë qarkun.

Shembull

Numri i shkronjave **s** në funksionin:

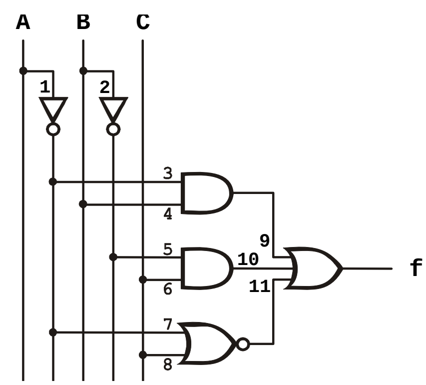
$$f = \overline{A} \cdot B + \overline{B}C + \overline{\overline{A} + C}$$

si dhe numrit i hyrjeve **h** në qarkun logjik përkatës.

Numri i shkronjave në funksionin e dhënë është **s=6** dhe llogaritet duke numëruar të gjitha shkronjat në shprehje kështu:

$$\begin{array}{cccccccccc} f & = & \overline{A} & B & + & \overline{B} & C & + & \overline{\overline{A} & + \; C} \\ & & \downarrow & \downarrow & & \downarrow & \downarrow & & \downarrow & \downarrow \\ & & 1 & 2 & & 3 & 4 & & 5 & 6 \end{array}$$

Qarku logjik i vizatuar në bazë të shprehjes së funksionit përkatës është:



prej nga shihet se numri i hyrjeve është **h=11**, ashtu siç janë shënuar edhe në qark.

Analiza dinamike

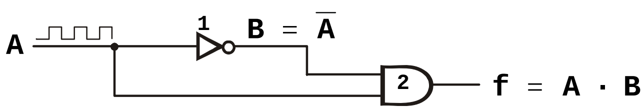
Gjatë punës reale të qarqeve logjike sinjalet në daljet e tyre nuk paraqiten njëkohësisht me ardhjen e sinjaleve hyrëse. Kjo është rezultat i pranisë së vonesave kohore të sinjaleve brenda elementeve të veçanta të qarkut. Intervali kohor që nevojitet për kalimin e sinjaleve nëpër qark quhet *vonesë e shpërndarjes* (ang. propagation delay) dhe fitohet si shumë e vonesave të shpërndarjes së sinjaleve nëpër elementet e qarkut.

Analiza e qarkut, pa i marrë parasysh vonesat e shpërndarjes së sinjaleve në qark, njihet edhe si *analizë statike*. Kurse *analiza reale e qarkut*, nën praninë e vonesave të shpërndarjes së sinjaleve në elementet e veçanta të qarkut, njihet si *analizë dinamike*.

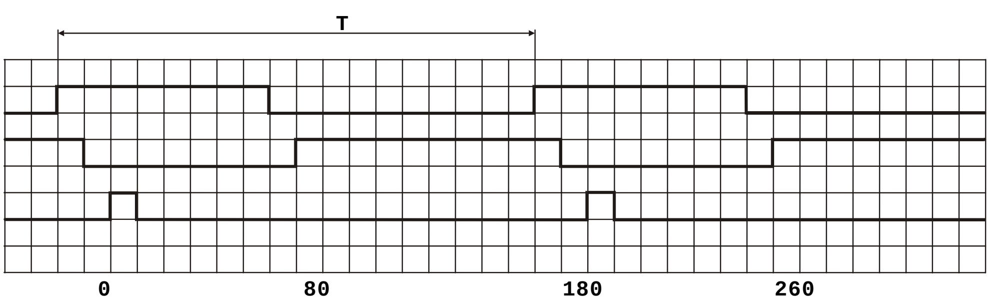
Prodhuesit e qarqeve të integruara me elemente logjike i japin edhe vlerat e vonesave të shpërndarjes së sinjaleve brenda tyre, të cilat kryesisht janë të rangut disa dhjetra *nanosekondë* (**ns**). Nëse gjatë analizës së qarqeve nuk merren parasysh këto vonesa, në punën reale të tyre mund të paraqiten anomali të padëshirueshme.

Shembull

Analiza dinamike e qarkut:



nëse në hyrje të tij aplikohen impulse me kohëzgjatje **τ=80[ns]** dhe periodë **T=180[ns]**, kurse vonesat e shpërndarjes së sinjalit brenda elementeve të veçanta janë: **t_1=10[ns]** dhe **t_2=20[ns]**.

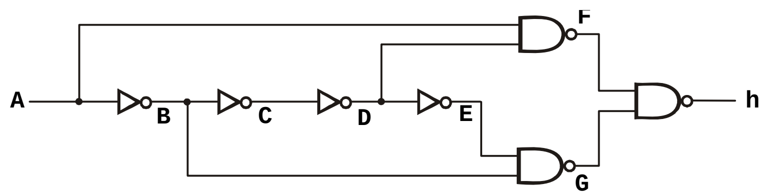


Nga diagramet e vizatuara më sipër shihet se prej vargut të impulseve hyrëse me kohëzgjatje **τ=80[ns]**, në dalje të qarkut fitohet vargu i impulseve me kohëzgjatje **10[ns]** edhe pse në bazë të shprehjes **f=AB** sinjali dalës duhet të jetë zero.

Gjatë vizatimit të diagrameve të sinjaleve në daljet e elementeve të qarkut, kur merret parasysh edhe prania e vonesës kohore, vizatimet mund të fillojnë vetëm prej momentit kur dihen vlerat e sinjaleve hyrëse në të gjitha hyrjet e elementeve.

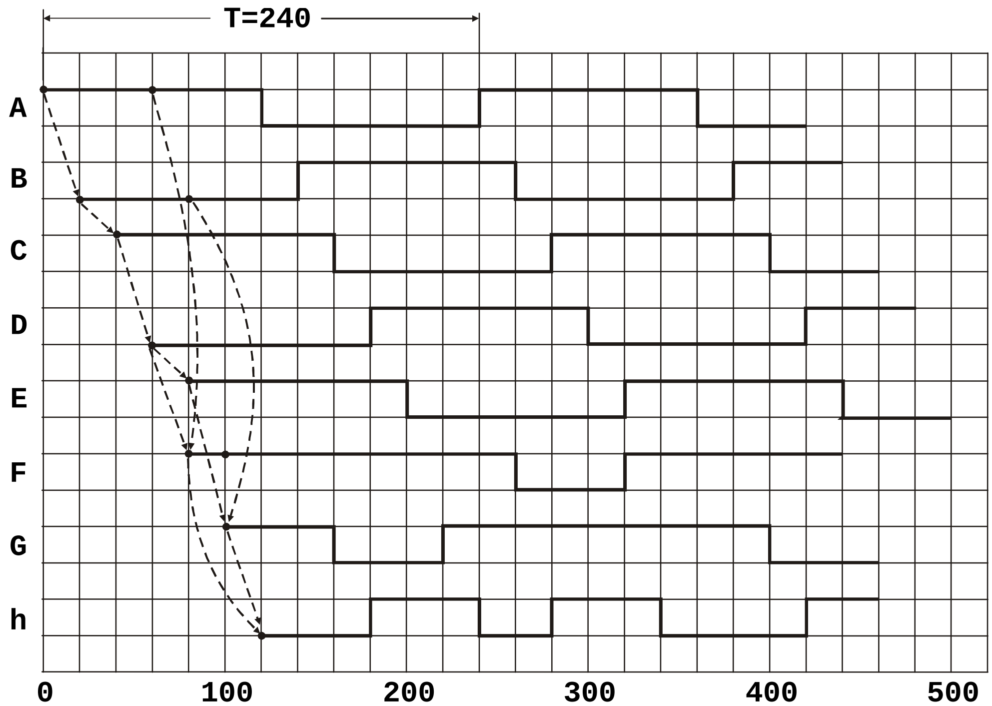
Shembull

Analiza dinamike e qarkut:



nëse në hyrjen **A** të tij aplikohen impulse me periodë **T=12τ**, ku **τ=20[ns]** janë vonesat në elementet e veçanta të qarkut.

$$B = \overline{A}$$
$$C = \overline{B}$$
$$D = \overline{D}$$
$$E = \overline{D}$$
$$F = \overline{AD}$$
$$G = \overline{BE}$$
$$h = \overline{FG}$$



Sinteza e qarqeve me më pakë nivele logjike ka një rëndësi të veçantë, sepse në këtë mënyrë zvogëlohen vonesat kohore të sinjaleve të cilat kalojnë nëpër elementet e qarqeve, përkatësisht qarqet mund të punojnë më shpejt.

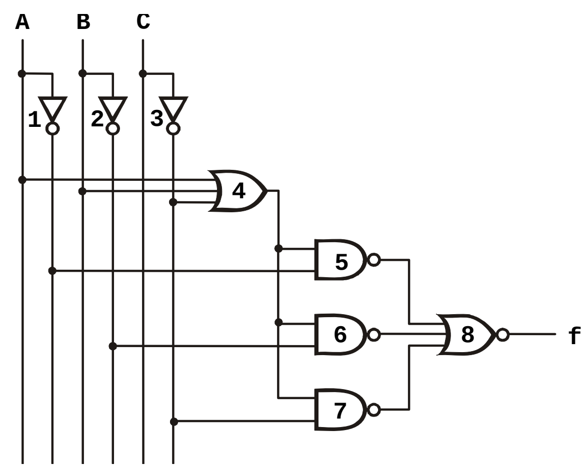
Ngarkesat e elementeve logjike

Numri i hyrjeve në një element logjik njihet si n*garkesë e hyrjeve* (ang. fan-in), kurse me *ngarkesën e daljeve* (ang. fan-out) nënkuptohet numri i elementeve të cilat mund të lidhen në daljen e një elementi logjik.

Ngarkesa e hyrjeve tek elementet logjike kufizohet nga prodhuesi i qarqeve të integruara, meqë fizkisht është e pamundshme që shfrytëzuesi ta rrisë numrin e hyrjeve (te disa elemente logjike ekziston mundësia e rritjes së numrit të hyrjeve përmes ekspanderëve të veçantë). Por, ngarkesa e daljeve të elementeve logjike, edhe pse jepet saktësisht në katalogun e prodhuesit (tipike është **8** deri në **10**), fizikisht është e pamundshme të kufizohet nga prodhuesi.

Shembull

Ngarkesa e hyrjeve dhe e daljeve te elementet logjike të qarkut vijues:



Elementi	Ngarkesa e hyrjeve	Ngarkesa e daljeve
1	1	1
2	1	1
3	1	2
4	3	3
5	2	1
6	2	1
7	2	1
8	3	0

Me qëllim të punës optimale të elementeve logjike, nga prodhuesit e qarqeve të integruara jepen rekomandimet vijuese.

Nëse hyrjet në elementet logjike të tipit **DTL** (nga Diode-Transistor Logic), ose të tipit **TTL** (nga Transistor-Transistor Logic) nuk përdoren, duhet të lidhen në tensionin furnizues pozitiv (**Vcc**) të qarkut. Tek elementet logjike të tipit **RTL** (nga Resistor-Transistor Logic) dhe **MOS** (nga Metal-Oxide Semiconductor) rekomandohet që hyrjet të cilat nuk shfrytëzohen të tokëzohen. Kurse, gjatë përdorimit të qarqeve **ECL** (nga Emitter-Coupled Logic), hyrjet e elementeve të cilat nuk shfrytëzohen, duhet të lidhen në tensionin furnizues negativ.

Për rastet kur daljet e elementeve duhet të ngarkohen më shumë sesa që është e lejueshme, ekzistojnë *elemente të fuqishme* (ang. power gate), te të cilat ngarkesa e daljes mund të jetë më e madhe. Por, ky problem në praktikë zgjidhet edhe duke i invertuar daljet e elementeve logjike dy herë, para se të tejkalohet ngarkesa e lejuar e tyre, p.sh., ashtu siç shihet në Fig.4.3.

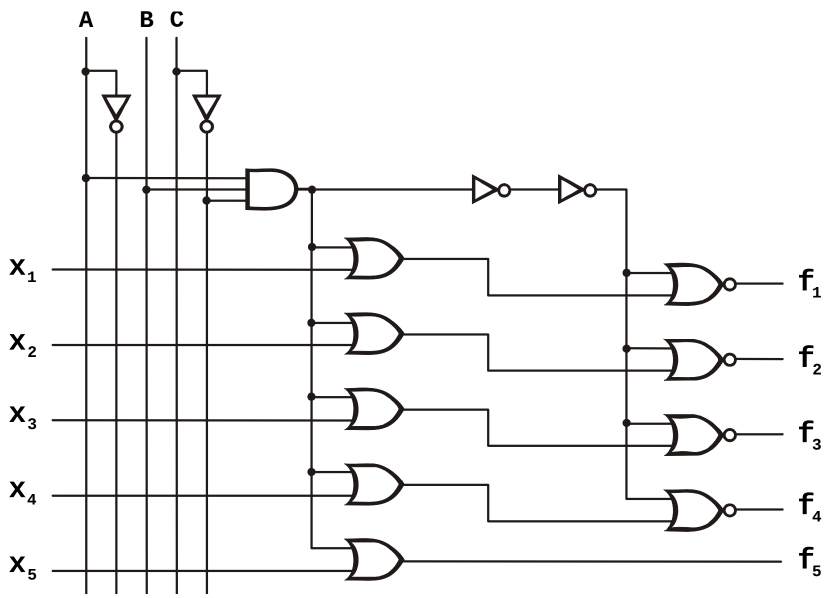


Fig.4.3 Invertimi i dyfishtë i daljeve me tejngarkesë

5 Koduesit

Në sistemet digjitale, informatat ruhen dhe përpunohen në formën e numrave ose të kodeve binare. Por, meqë informatat elementare kryesisht nuk kanë formë binare, kodimi i tyre në numra të sistemit binar, përkatësisht në ndonjë kod binar, bëhet përmes qarqeve logjike të cilat njihen si *kodues* (ang. encoder).

Koduesit e zakonshëm

Koduesit përmes së cilëve kodohen **m**-informata elementare në një kod **n**-bitësh, përmbajnë **m**-hyrje e **n**-dalje dhe skematikisht mund të paraqiten si në Fig.5.1.

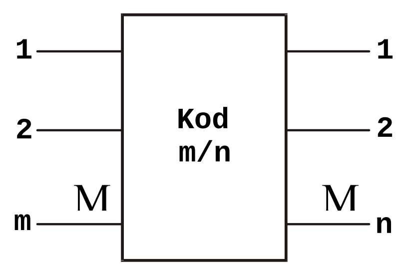


Fig.5.1 Paraqitja skematike e koduesve

Në hyrje të koduesit, në një moment të caktuar aplikohet sinjali i njërës nga **m**-informatat elementare që kodohen, kurse në dalje të tij paraqitet fjala kodike përkatëse **n**-bitëshe.

Me qëllim që të standardizohet komunikimi, në pjesën vijuese të librit, nëse nuk theksohet ndryshe, vargjet e vlerave të sinjaleve hyrëse dhe dalëse, në rastet kur ka më shumë hyrje, ose më shumë dalje, do të jepen duke i vërejtur hyrjet dhe daljet, me radhën e cila shkonë prej lart - poshtë, ose prej majtas - djathtas.

Shembull

Koduesi **4/3** përmes së cilit kodohen shifrat decimale **1**, **2**, **4** dhe **5**, me ekuivalentët binar përkatës.

Shifrat decimale				Kodi		
1	2	4	5	A	B	C
1	0	0	0	0	0	1
0	1	0	0	0	1	0
0	0	1	0	1	0	0
0	0	0	1	1	0	1

$$A = 5 + 7$$

$$B = 2$$

$$C = 1 + 5$$

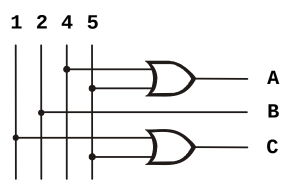


Fig.5.2 Koduesi 4/3 i shifrave decimale me ekuivalentët binar përkatës

Shembull

Koduesi përmes të cilit kodohen shifrat decimale prej **1** deri në **9**, në kodin **NBCD**.

Shifra decimale									Kodi			
1	2	3	4	5	6	7	8	9	A	B	C	D
1	0	0	0	0	0	0	0	0	0	0	0	1
0	1	0	0	0	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	0	0	0	1	1
0	0	0	1	0	0	0	0	0	0	1	0	0
0	0	0	0	1	0	0	0	0	0	1	0	1
0	0	0	0	0	1	0	0	0	0	1	1	0
0	0	0	0	0	0	1	0	0	0	1	1	1
0	0	0	0	0	0	0	1	0	1	0	0	0
0	0	0	0	0	0	0	0	1	1	0	0	1

A = 8 + 9
B = 4 + 5 + 6 + 7
C = 2 + 3 + 6 + 7
D = 1 + 3 + 5 + 7 + 9

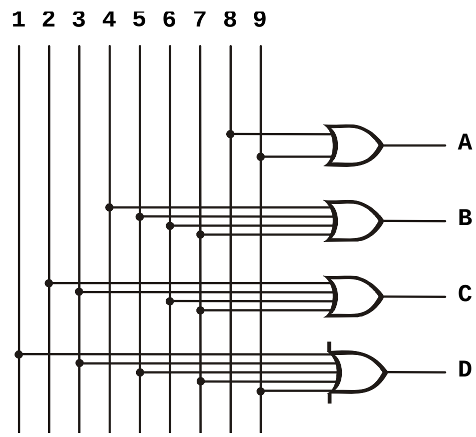


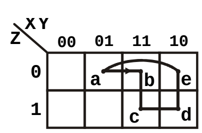
Fig.5.3 Koduesi i shifrave decimale në kodin NBCD

Funksionimi i qarkut logjik të koduesit të dhënë më sipër mund të provohet duke marrë vlera të ndryshme hyrëse. Nëse, p.sh., sinjali me vlerën **1** aplikohet në hyrjen me numër **5**, kurse në krejt hyrjet e tjera aplikohet sinjali me vlerën **0**, përkatësisht nëse në hyrjet e qarkut aplikohen vlerat **000010000**, në daljet **A**, **B**, **C** dhe **D** të qarkut fitohet fjala kodike **0101**. Me qëllim që të duket më qartë mënyra e funksionimit të koduesit, në daljet e tij mund të lidhen dioda ndriçuese, ose, siç quhen ndryshe **LED** (nga Light Emiting Diode), të cilat ndriçojnë, nëse sinjali dalës ka vlerën **1**. Kështu, p.sh., nëse në hyrje të qarkut aplikohet vargu i vlerave binare **000000100**, në dalje të tij fitohet vargu **0111**, përkatësisht ekuivalenti binar i shifrës decimale **7** dhe ndriçojnë diodat e vendosura në **3** daljet e fundit.

Pa ndonjë kufizim, informatat elementare mund të kodohen në kode të ndryshme.

Shembull

Koduesi përmes të cilit kodohen në kodin ciklik të dhënë përmes tabelës vijuese informatat elementare **a**, **b**, **c**, **d** dhe **e**, të cilat gjenerohen nga **5** terminalet e një sistemi digjital.



Informatat					Kodi		
a	b	c	d	e	X	Y	Z
1	0	0	0	0	0	1	0
0	1	0	0	0	1	1	0
0	0	1	0	0	1	1	1
0	0	0	1	0	1	0	1
0	0	0	0	1	1	0	0

$$X = b + c + d + e$$
$$Y = a + b + c$$
$$Z = c + d$$

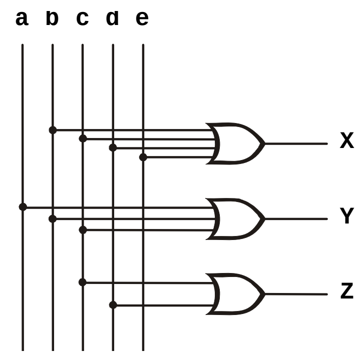


Fig.5.4 Koduesi në kodin ciklik

Që të përcaktohet saktësia e funksionimit të qarkut logjik të dhënë më sipër, si kodues në kodin ciklik të **5** sinjaleve hyrëse duhet të merren shembuj të kombinimeve të vlerave hyrëse, por ashtu që vetëm njëra prej tyre ta ketë vlerën **1**. Kështu, p.sh., nëse në hyrje aplikohet vargu i vlerave **00010**, në dalje fitohet fjala kodike **101**, e cila gjenerohet kur vetëm hyrja **d** ka vlerën **1**. Ngjashëm mund të provohet se gjenerohen edhe fjalët e tjera kodike të kodit ciklik të dhënë, nëse vlera **1** aplikohet në hyrjet e tjera të qarkut.

Gjatë kodimit të informatave elementare, në kodues mund të parashihet edhe një dalje e veçantë, përmes së cilës sinjalizohet prurja e njëkohshme në hyrjet e koduesit, të më shumë se një informate, përkatësisht dalja për zbulim të gabimit.

Shembull

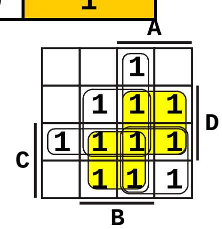
Koduesi përmes të cilit kodohen informatat elementare **A**, **B**, **C** dhe **D**, me ekuivalentët binarë të shifrave decimale **1**, **2**, **3** dhe **4** - përkatësisht. Në kodues duhet të parashihet edhe dalja e veçantë **g**, në të cilën paraqitet sinjali me vlerën **1**, nëse në hyrje të koduesit aplikohen njëkohësisht më shumë sinjale me vlerën **1**.

Hyrjet				Daljet			
Sinjalet				Kodi			Gabimi
A	B	C	D	x	y	z	g
0	0	0	0	0	0	0	0
0	0	0	1	0	0	1	0
0	0	1	0	0	1	0	0
0	0	1	1	0	0	0	1
0	1	0	0	0	1	1	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	0	1
0	1	1	1	0	0	0	1
1	0	0	0	1	0	0	0
1	0	0	1	0	0	0	1
1	0	1	0	0	0	0	1
1	0	1	1	0	0	0	1
1	1	0	0	0	0	0	1
1	1	0	1	0	0	0	1
1	1	1	0	0	0	0	1
1	1	1	1	0	0	0	1

$$x = A\,\overline{B}\,\overline{C}\,\overline{D}$$

$$y = \overline{A}\,\overline{B}\,C\,\overline{D} + \overline{A}\,B\,\overline{C}\,\overline{D}$$

$$z = \overline{A}\,\overline{B}\,\overline{C}\,D + \overline{A}\,B\,\overline{C}\,\overline{D}$$



$$g = AB + AC + AD + BC + BD + CD$$

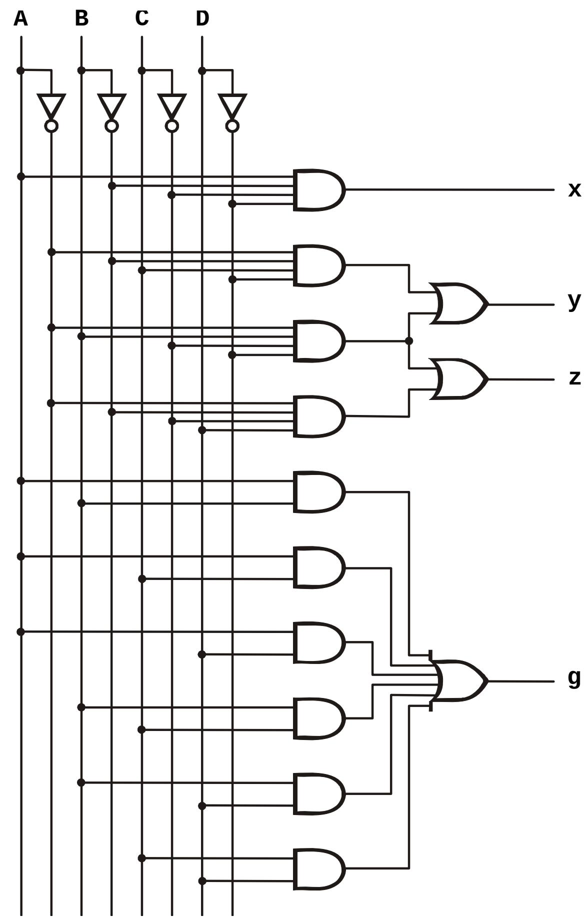


Fig.5.5 Koduesi me dalje për zbulim të gabimit

Nëse në hyrje të qarkut të mësipërm, përkatësisht në hyrjet **A**, **B**, **C** dhe **D**, p.sh., aplikohet vargu i sinjaleve **0100**, në daljet e tij **x**, **y** dhe **z** fitohet fjala kodike **011**. Njëkohësisht, vlera e sinjalit në daljen **g** është **0**, për të treguar se nuk është detektuar gabim. Por, kur njëkohësisht në dy hyrje të qarkut aplikohet sinjali **1**, përkatësisht kërkohet kodimi i njëkohshëm i dy informatave elementare, në daljen **g** të qarkut gjenerohet sinjali **1**, për të treguar se është gabuar. Kështu, p.sh., nëse vargu i sinjaleve në hyrjet e qarkut është **0110**, kurse në daljet e qarkut lidhen dioda **LED**, do të ndriçojë vetëm dioda e vendosur në daljen **g**, përkatësisht vargu i vlerave të sinjaleve dalës është **0001**.

Fjalëve kodike që fitohen në dalje të koduesit mund t'u shtohet edhe shifra për paritet, qoftë të njësheve ose të zerove. Për këtë qëllim në fakt në dalje të koduesit parashihet një dalje e veçantë.

Shembull

Koduesi përmes të cilit kodohen në kodin **ASCII** simbolet e tasteve **K**, **T**, **9**, ***** dhe **$** gjatë shtypjes së tyre në tastierën e kompjuterit, duke ua shtuar njëkohësisht edhe shifrën për paritet çift.

Simboli					K o d i							
K	T	9	*	$	P	a	b	c	D	e	f	g
1	0	0	0	0	0	1	0	0	1	0	1	1
0	1	0	0	0	1	1	0	1	0	1	0	0
0	0	1	0	0	0	0	1	1	1	0	0	1
0	0	0	1	0	1	0	1	0	1	0	1	0
0	0	0	0	1	0	0	1	0	0	1	0	0

$$\mathbf{P} = \mathbf{T} + \mathbf{*}$$

$$\mathbf{a} = \mathbf{K} + \mathbf{T}$$

$$\mathbf{b} = \mathbf{9} + \mathbf{*} + \mathbf{\$}$$

$$\mathbf{c} = \mathbf{T} + \mathbf{9}$$

$$\mathbf{d} = \mathbf{K} + \mathbf{9} + \mathbf{*}$$

$$\mathbf{e} = \mathbf{T} + \mathbf{\$}$$

$$\mathbf{f} = \mathbf{K} + \mathbf{*}$$

$$\mathbf{g} = \mathbf{K} + \mathbf{9}$$

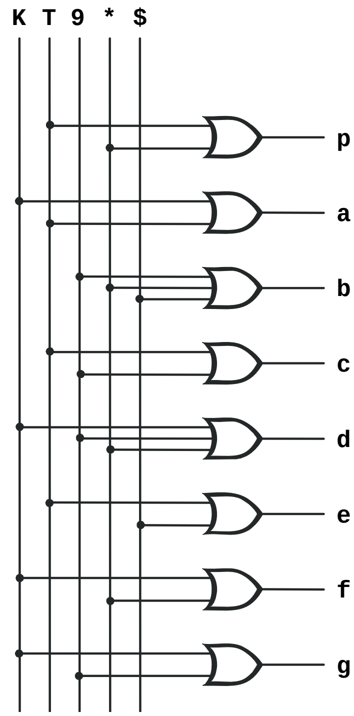


Fig.5.6 Koduesi në kodin **ASCII**

Fjalët kodike të informatave elementare këtu fitohen në daljet **a**, **b**, **c**, **d**, **e**, **f** dhe **g** të koduesit, kurse shifra për çiftësi i bashkangjitet çdo fjale kodike përmes daljes **P**. Kështu, p.sh., nëse në tastierë shtypet tasti **T**, përkatësisht sinjali me vlerën **1** aplikohet vetëm në hyrjen përkatëse, kurse në krejt hyrjet e tjera aplikohet sinjali me vlerën **0**, në dalje të qarkut do të fitohet vargu i sinjaleve **11010100**, ku vlera e parë ka të bëjë me vlerën në daljen për paritet **P**.

Koduesit me prioritet

Koduesi me prioritet (ang. priority encoder) është qark logjik i cili e *gjeneron numrin rendor të pajisjes që i jepet prioritet*, kur në një pajisje të përbashkët kërkojnë qasje njëkohësisht disa pajisje. Nëse në hyrje të koduesit me prioritet lidhen **m**-pajisje, ai kryesisht ka $n=log_2m$ dalje. Si hyrje në koduesin me prioritet, p.sh., mund të jenë linjat e terminaleve, të cilat paraqiten për qasje në kompjuterin qendror. Kur dy terminale, T_i dhe T_j, kërkojnë njëkohësisht të lidhen me kompjuterin qendror, përkatësisht në linjat përkatëse paraqiten sinjale me vlerë **1**, terminalit T_i i jepet përparësi ndaj atij T_j, nëse $i>j$. Koduesi në këtë rast e gjeneron ekuivalentin binar të numrit **i**, duke njoftuar kështu kompjuterin qendror se terminali T_i ka përparësi për qasje para terminalit T_j.

Shembull

Koduesi me prioritet përmes të cilit përcaktohet prioriteti i qasjes në kompjuterin qendror të njërit nga **5** terminalet, të cilat punojnë njëkohësisht, duke ndarë kohën e punës së kompjuterit qendror, në regjimin e punës që njihet si *time-sharing*.

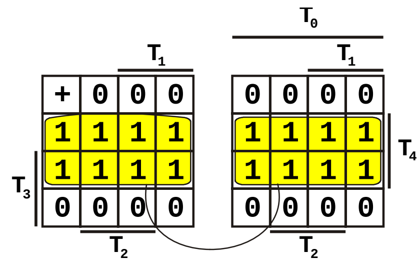
Tabela e kombinimeve për koduesin me prioritet në fjalë, në formë të shkurtuar duket kështu:

T_0	T_1	T_2	T_3	T_4	z_4	z_2	z_1
1	0	0	0	0	0	0	0
+	1	0	0	0	0	0	1
+	+	1	0	0	0	1	0
+	+	+	1	0	0	1	1
+	+	+	+	1	1	0	0

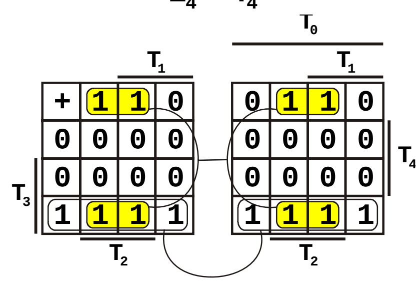
Përmes simbolit **+** në tabelë është treguar se vlerat nën diagonale nuk kanë ndikim në gjenerimin e ekuivalentit binar të numrit rendor të terminalit, kur terminali me numër rendor më të madh kërkon qasje.

Tabela komplete e kombinimeve me shprehjet e funksioneve përmes të cilave përshkruhet koduesi me prioritet si dhe qarku logjik përkatës janë dhënë në vijim.

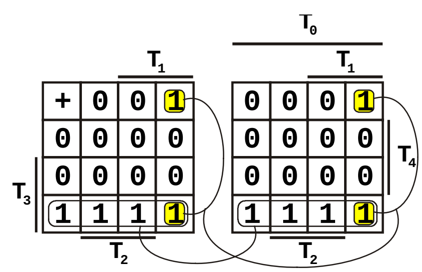
T_0	T_1	T_2	T_3	T_4	Z_4	Z_2	Z_1
0	0	0	0	0	+	+	+
0	0	0	0	1	1	0	0
0	0	0	1	0	0	1	1
0	0	0	1	1	1	0	0
0	0	1	0	0	0	1	0
0	0	1	0	1	1	0	0
0	0	1	1	0	0	1	1
0	0	1	1	1	1	0	0
0	1	0	0	0	0	0	1
0	1	0	0	1	1	0	0
0	1	0	1	0	0	1	1
0	1	0	1	1	1	0	0
0	1	1	0	0	0	1	0
0	1	1	0	1	1	0	0
0	1	1	1	0	0	1	1
0	1	1	1	1	1	0	0
1	0	0	0	0	0	0	0
1	0	0	0	1	1	0	0
1	0	0	1	0	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0	0	0	1	0
1	0	1	0	1	1	0	0
1	0	1	1	0	0	1	1
1	0	1	1	1	1	0	0
1	1	0	0	0	0	0	1
1	1	0	0	1	1	0	0
1	1	0	1	0	0	1	1
1	1	0	1	1	1	0	0
1	1	1	0	0	0	1	0
1	1	1	0	1	1	0	0
1	1	1	1	0	0	1	1
1	1	1	1	1	1	0	0



$$Z_4 = T_4$$



$$Z_2 = T_2T_4 + T_3T_4$$



$$Z_1 = T_1T_2T_4 + T_3T_4$$

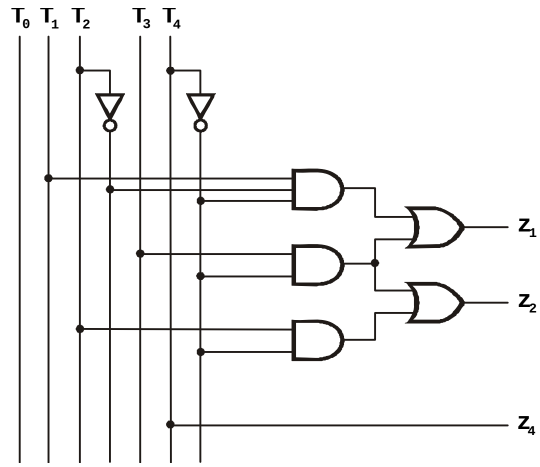


Fig.5.7 Koduesi me prioritet

Nëse, p.sh., në hyrjet T_0, T_1, T_2, T_3 dhe T_4 të koduesit me prioritet të dhënë më sipër nga terminalet e veçanta vjen vargu i sinjaleve **10110**, mund të vëretohet se në dalje të tij fitohet numri binar **011**, ekuivalenti decimal i të cilit i përgjigjet numrit rendor të terminalit T_3. Në këtë mënyrë koduesi i jep prioritet terminalit T_3, kundruall terminaleve T_0 dhe T_2, të cilat gjithashtu kanë dërguar sinjale me vlerën **1**, përkatësisht kanë kërkesë për lidhje me kompjuterin qendror.

Dekoduesit 6

Qarqet logjike të cilat kanë funksione inverse me koduesit, përkatësisht qarqet që përdoren për dekodimin e informatave të koduara, quhen *dekodues* (ang. decoder).

Dekoduesit e zakonshëm

Dekoduesit përmes së cilëve fjalët kodike **n**-bitëshe konvertohen në **m**-informata elementare, shkurt njihen si *dekodues* ***n*** *në* ***m*** dhe skematikisht mund të paraqiten si në Fig.6.1.

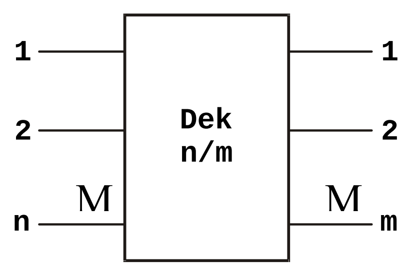


Fig.6.1 Paraqitja skematike e dekoduesit

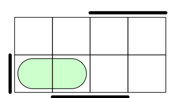
Në bazë të vlerave që vijnë në **n**-hyrjet e dekoduesit zgjidhet njëra nga $m \leq 2^n$ daljet e tij, në të cilën përcillet vlera **1**, kur njëkohësisht, në të gjitha daljet e tjera sinjali ka vlerën **0**.

Dekoduesi ka një zbatim të gjerë në qarqet digjitale komplekse. P.sh., përmes dekoduesit mund të përcaktohet adresa e lokacionit të memories ku lexohet ose ku memorohet një e dhënë etj.

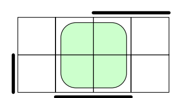
Shembull

Dekoduesi **3/4** përmes të cilit për fjalët kodike binare trebitëshe zgjidhen daljet përkatëse të ekuivalentëve decimal të numrave tek **1**, **2**, **4** dhe **5**, duke i marrë kombinimet e pashfrytëzuara si vlera arbitrare.

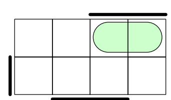
A	B	C	1	2	4	5
0	0	0	+	+	+	+
0	0	1	1	0	0	0
0	1	0	0	1	0	0
0	1	1	+	+	+	+
1	0	0	0	0	1	0
1	0	1	0	0	0	1
1	1	0	+	+	+	+
1	1	1	+	+	+	+



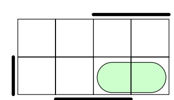
$1 = \bar{A}\,C$



$2 = B$



$4 = A\,\bar{C}$



$5 = A\,C$

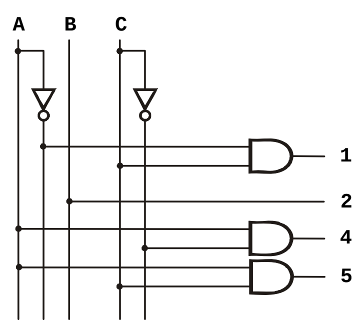


Fig.6.2 Dekoduesi 3/4 i ekvivalentëve binar të numrave decimal 1, 2, 4 dhe 5

A

+ 0 + 0

C 1 + + 0

B

A

Shembull

Dekoduesi **4/10** përmes të cilit për fjalët kodike të kodit **NBCD** zgjidhen shifrat ekuivalente decimale prej **0** deri më **9**, duke marrë kombinimet e pashfrytëzuara si vlera arbitrare.

A	B	C	D	0	1	2	3	4	5	6	7	8	9
0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	0	0
0	1	0	1	0	0	0	0	0	1	0	0	0	0
0	1	1	0	0	0	0	0	0	0	1	0	0	0
0	1	1	1	0	0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	1	0	0	0	0	0	0	0	0	0	1
1	0	1	0	+	+	+	+	+	+	+	+	+	+
1	0	1	1	+	+	+	+	+	+	+	+	+	+
1	1	0	0	+	+	+	+	+	+	+	+	+	+
1	1	0	1	+	+	+	+	+	+	+	+	+	+
1	1	1	0	+	+	+	+	+	+	+	+	+	+
1	1	1	1	+	+	+	+	+	+	+	+	+	+

Pas minimizimit përmes procedurës së shfrytëzuar në shembullin paraprak, për daljet e veçanta nga qarku fitohen shprehjet vijuese.

$$0 = \bar{A}\,\bar{B}\,\bar{C}\,\bar{D}$$
$$1 = \bar{A}\,\bar{B}\,\bar{C}\,D$$
$$2 = \bar{B}\,C\,\bar{D}$$
$$3 = \bar{B}\,C\,D$$
$$4 = B\,\bar{C}\,\bar{D}$$
$$5 = B\,\bar{C}\,D$$
$$6 = B\,C\,\bar{D}$$
$$7 = B\,C\,D$$
$$8 = A\,\bar{D}$$
$$9 = A\,D$$

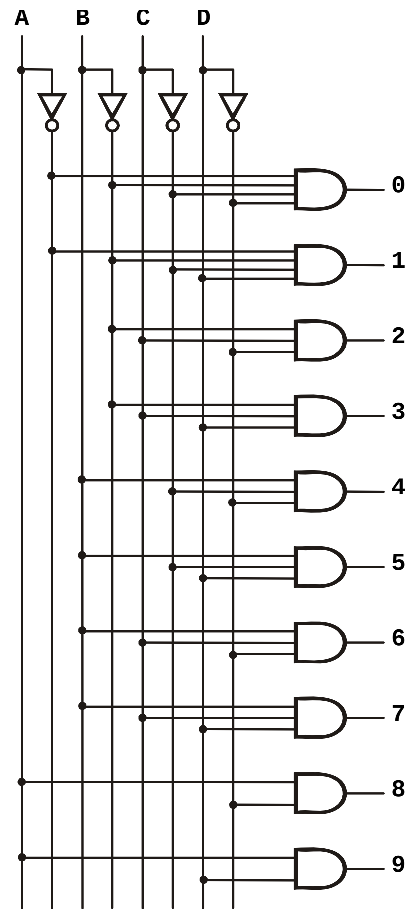


Fig.6.3 Dekoduesi NBCD - decimal

Shprehjet në bazë të të cilave është vizatuar qarku i dekoduesit të dhënë më sipër janë fituar përmes minimizimit të funksioneve me të cilat përshkruhen daljet e veçanta, përkatësisht qarku logjik përkatës. Për ta parë funksionimin e këtij qarku, le ta marrim, p.sh., fjalën kodike **0110** në hyrje të dekoduesit, e cila në kodin **NBCD** i përgjigjet shifrës decimale **6**. Mund të gjendet lehtë se vargu i vlerave të sinjaleve në dalje të qarkut është **0000001000**, përkatësisht se vetëm në daljen me numër rendor **6** vlera e sinjalit është **1**.

Nëse dekoduesi ka $\mathbf{n}$-hyrje dhe saktësisht $\mathbf{m=2^n}$ dalje, funksionet me të cilat përshkruhet qarku logjik i dekoduesit janë të barabarta me mintermat e numrave rendorë të daljeve përkatëse.

Shembull

Dekoduesi **3/8** përmes të cilit për ekuivalentët binarë të shifrave decimale prej **0** deri në **7** zgjidhen daljet të cilat u përkasin shifrave në sistemin oktal të numrave.

A	B	C	0	1	2	3	4	5	6	7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

$$0 = \bar{A}\,\bar{B}\,\bar{C}$$
$$1 = \bar{A}\,\bar{B}\,C$$
$$2 = \bar{A}\,B\,\bar{C}$$
$$3 = \bar{A}\,B\,C$$
$$4 = A\,\bar{B}\,\bar{C}$$
$$5 = A\,\bar{B}\,C$$
$$6 = A\,B\,\bar{C}$$
$$7 = A\,B\,C$$

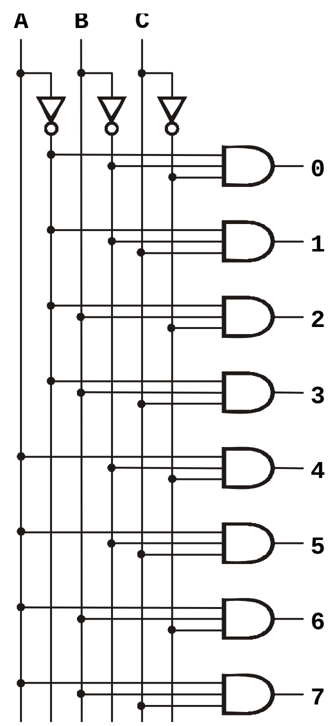


Fig.6.4 Dekoduesi binar-oktal

Për ta vërtetuar saktësinë e punës së dekoduesit të dhënë më sipër, le të marrim, p.sh., se në hyrjet **A**, **B** dhe **C** të tij aplikohet vargu i vlerave **010**. Si rezultat, vetëm në daljen me numër rendor **2** do të paraqitet sinjali me vlerën **1**, kurse sinjalet në daljet e tjera do të kenë vlerën **0**, përkatësisht vargu i sinjaleve dalëse do të jetë **00100000**.

Dekoduesi mund të përdoret edhe për gjetjen e vlerës komplementare të një numri.

Shembull

Dekoduesi **4/10** përmes të cilit gjenden **9**-komplementet e fjalëve kodike të kodit **NBCD**, nëse ata vërehen si numra të sistemit decimal.

A	B	C	D	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	1	0	0	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	0	0	0	1	0	0
0	0	1	1	0	0	0	0	0	0	1	0	0	0
0	1	0	0	0	0	0	0	0	1	0	0	0	0
0	1	0	1	0	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	0	1	0	0	0	0	0	0
0	1	1	1	0	0	1	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	0	0	0	0
1	0	0	1	1	0	0	0	0	0	0	0	0	0
1	0	1	0	+	+	+	+	+	+	+	+	+	+
1	0	1	1	+	+	+	+	+	+	+	+	+	+
1	1	0	0	+	+	+	+	+	+	+	+	+	+
1	1	0	1	+	+	+	+	+	+	+	+	+	+
1	1	1	0	+	+	+	+	+	+	+	+	+	+
1	1	1	1	+	+	+	+	+	+	+	+	+	+

$0 = AD$

$1 = A\bar{D}$

$2 = BCD$

$3 = BC\bar{D}$

$4 = B\bar{C}D$

$5 = B\bar{C}\bar{D}$

$6 = \bar{B}CD$

$7 = \bar{B}C\bar{D}$

$8 = \bar{A}\bar{B}\bar{C}D$

$9 = \bar{A}\bar{B}\bar{C}\bar{D}$

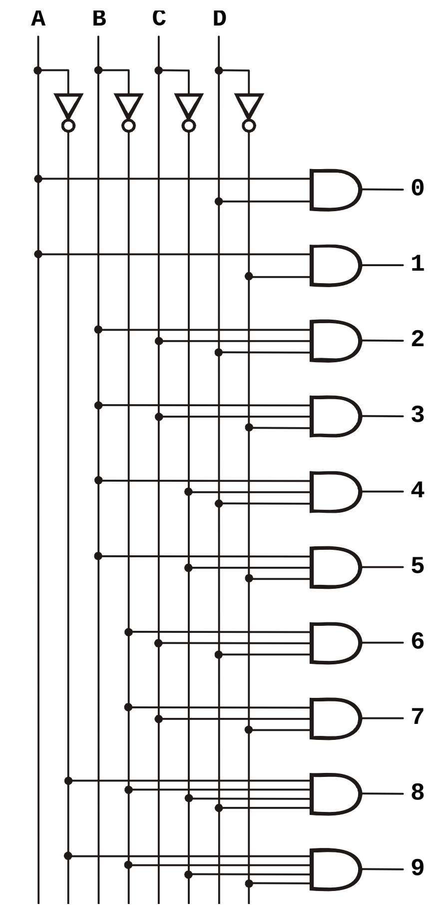


Fig.6.5 Dekoduesi NBCD - 9-komplementi

Tabela e kombinimeve e dhënë më sipër është plotësuar duke i gjetur **9**-komplementet e ekuivalentëve decimalë të fjalëve kodike të veçanta. Kështu, p.sh., për fjalën kodike **0101** ekuivalenti decimal i së cilës është numri **5**, **9**-komplementi përkatës gjendet duke shfrytëzuar shprehjen e dhënë më parë:

$$N_{\overline{9}} = 10^1 - 10^0 - 5 = 4$$

Si rrjedhim, në tabelën e kombinimeve, për fjalën kodike **0101**, sinjali me vlerën **1** merret në daljen **4**, kurse në krejt daljet e tjera merret sinjali me vlerën **0**.

Dekoduesi dynivelësh

Dekoduesit e dhënë më sipër paraqesin *dekodues njënivelësh* (ang. one-level decoder). Te dekoduesit njënivelësh me më shumë hyrje rritet numri i hyrjeve në elementet logjike brenda tyre. Me qëllim të eliminimit të këtij problemi, përdoren *dekoduesit dynivelësh* (ang. two-stage decoder), të cilët njihen edhe si *pemë dekoduese* (ang. decoding tree), ose *dekodues pemë* (ang. tree decoder).

Shembull

Dekoduesi **3/8**, përmes së cilit për ekuivalentët binarë të shifrave decimale prej **0** deri në **7** zgjidhen daljet të cilat u përkasin shifrave në sistemin oktal të numrave, i realizuar si dekodues dynivelësh.

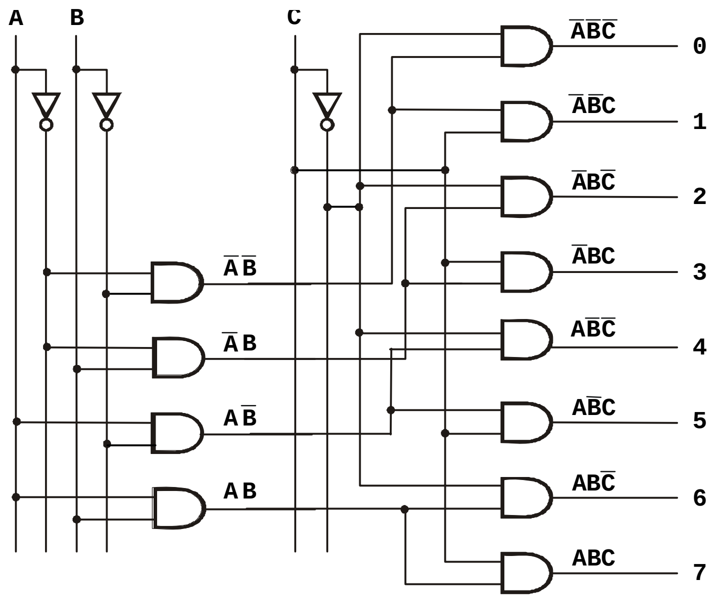


Fig.6.6 Dekoduesi dynivelësh binar-oktal

Qarku i fituar paraqet dekoduesin dynivelësh, sepse elementet logjike janë grupuar në dy nivele të veçanta. Nëse ky qark krahasohet me dekoduesin me funksion të njëjtë, i cili është dhënë më parë, shihet se këtu numri i hyrjeve në elementet e veçanta brenda qarkut është reduktuar në dy, gjë që ka rëndësi për rastet e dekoduesve me më shumë hyrje, kur ky numër është i madh.

Dekoduesi trenivelësh

Nëse në hyrje të dekoduesit paraqiten **4** variabla, ai mund të realizohet si dekodues trenivelësh, përkatësisht pema dekoduese përkatëse do të ketë degë të vendosura në më shumë nivele.

Shembull

Dekoduesi **4/16**, përmes të cilit për ekuivalentët binarë të shifrave decimale prej **0** deri në **15** zgjidhen daljet të cilat u përkasin numrave decimalë përkatës.

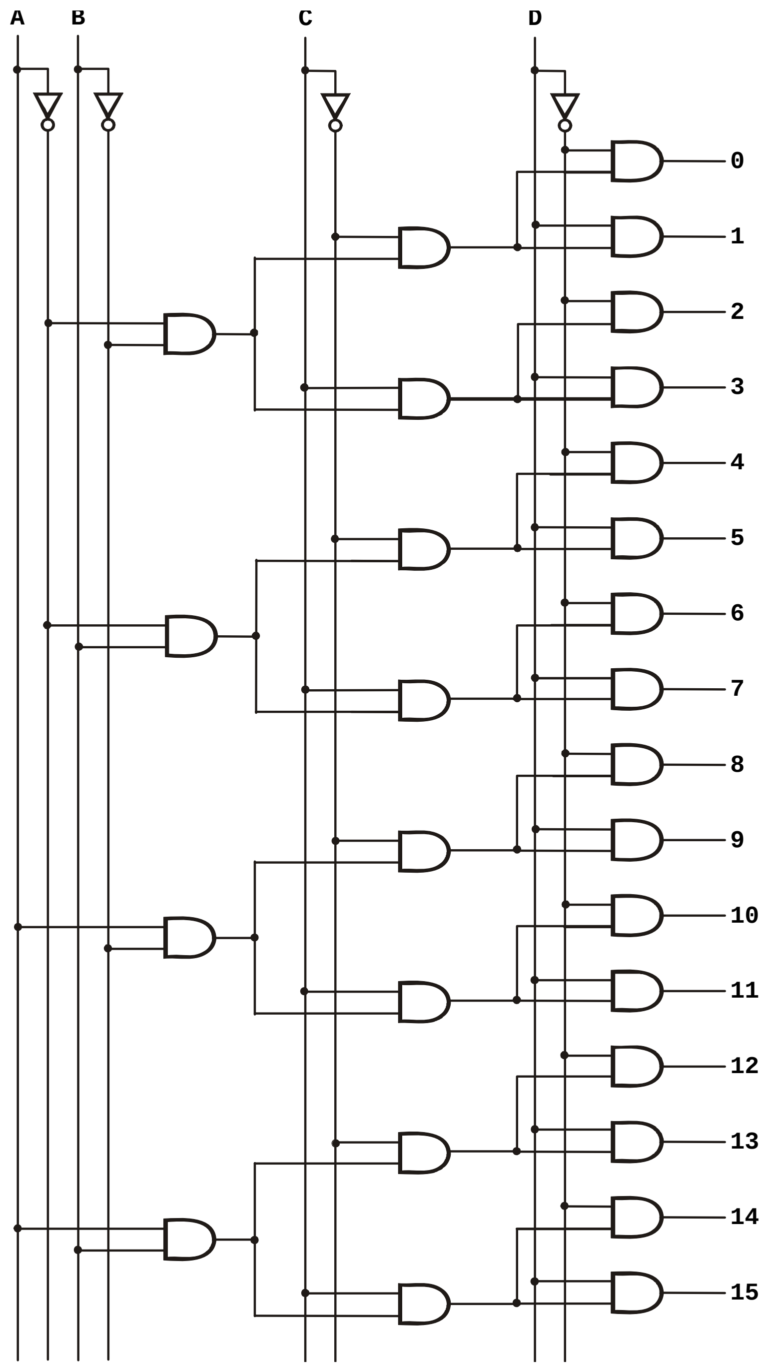


Fig.6.7 Dekoduesi trenivelësh binar-decimal

Realizimi i dekoduesve kompleks

Dekoduesit më kompleks realizohen rregullisht duke shfrytëzuar disa dekodues më të vegjël dhe elemente logjike për realizimin e ndërlidhjeve të nevojshme mes tyre.

Shembull Dekoduesi **4/16**, i realizuar përmes dy dekoduesve **2/4**, përmes të cilit gjenden ekuivalentët decimalë të numrave binarë katërshifror.

*Dekoduesi **2 në 4***

A	B	f_0	f_1	f_2	f_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

$$f_0 = \bar{A}\,\bar{B}$$
$$\overline{f_1} = \bar{A}\,B$$
$$f_2 = A\,\bar{B}$$
$$f_3 = A\,B$$

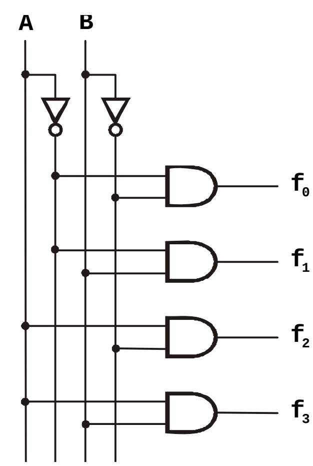


Fig.6.8 Dekoduesi 2/4

Dekoduesi **4 në 16**

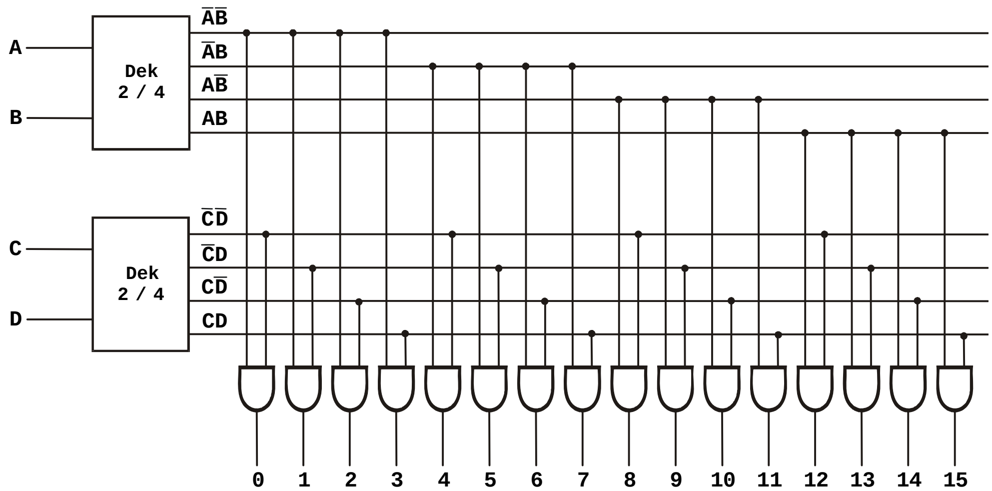


Fig.6.9 Dekoduesi 4/16 i realizuar me dekodues 2/4

Dekoduesit e tillë quhen edhe *dekodues matricorë* (ang. matrix decoder), sepse kur vizatohen, duke i vendosur dy dekoduesit elementarë, njërin në drejtim të boshtit **x** e tjetrin në drejtim të boshtit **y**, elementet për ndërlidhje mund të shpërndahen ashtu që të duken si elemente të një matrice. P.sh., dekoduesi i realizuar në shembullin e mësipërm mund të vizatohet edhe kështu:

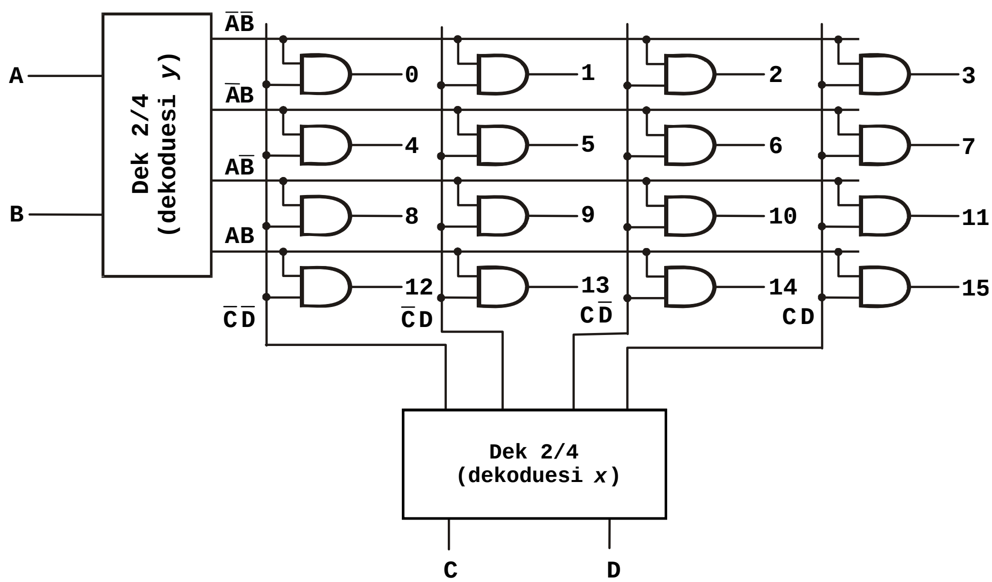


Fig.6.10 Dekoduesi matricor 4/16

Nga figura e dhënë më sipër shihet se në këtë rast kemi të bëjmë me një *dekodues matricor 4 x 4*.

Realizimi i qarqeve përmes dekoduesve

Dekoduesi me saktë **n**-hyrje dhe $\mathbf{2^n}$-dalje mund të shfrytëzohet për realizimin e funksioneve të ndryshme logjike, sepse në çdo dalje të dekoduesit të tillë fitohet një mintermë e caktuar. Në këtë mënyrë, përmes shumës së mintermave të caktuara, duke shfrytëzuar elementet logjike **OSE**, mund të gjenden vlerat e funksioneve logjike.

Shembull

Qarku logjik përmes të cilit gjendet **2**-komplementi $\mathbf{(XYZV)_2}$ i numrit hyrës $\mathbf{(ABCD)_2}$. Në dalje të qarkut është paraparë edhe dalja e veçantë **R**, në të cilën fitohet vlera e derdhjes që paraqitet gjatë komplementimt të numrit **0000**. Për realizimin e këtij qarku logjik është përdorur një dekodues **4/16** dhe elementet logjike **OSE**.

	A	B	C	D	R	X	Y	Z	V
0	0	0	0	0	1	0	0	0	0
1	0	0	0	1	0	1	1	1	1
2	0	0	1	0	0	1	1	1	0
3	0	0	1	1	0	1	1	0	1
4	0	1	0	0	0	1	1	0	0
5	0	1	0	1	0	1	0	1	1
6	0	1	1	0	0	1	0	1	0
7	0	1	1	1	0	1	0	0	1
8	1	0	0	0	0	1	0	0	0
9	1	0	0	1	0	0	1	1	1
10	1	0	1	0	0	0	1	1	0
11	1	0	1	1	0	0	1	0	1
12	1	1	0	0	0	0	1	0	0
13	1	1	0	1	0	0	0	1	1
14	1	1	1	0	0	0	0	1	0
15	1	1	1	1	0	0	0	0	1

$$R = \sum m^1(0)$$

$$X = \sum m^1(1,2,3,4,5,6,7,8)$$

$$y = \sum m^1(1,2,3,4,9,10,11,12)$$

$$Z = \sum m^1(1,2,5,6,9,10,13,14)$$

$$V = \sum m^1(1,3,5,7,9,11,13,15)$$

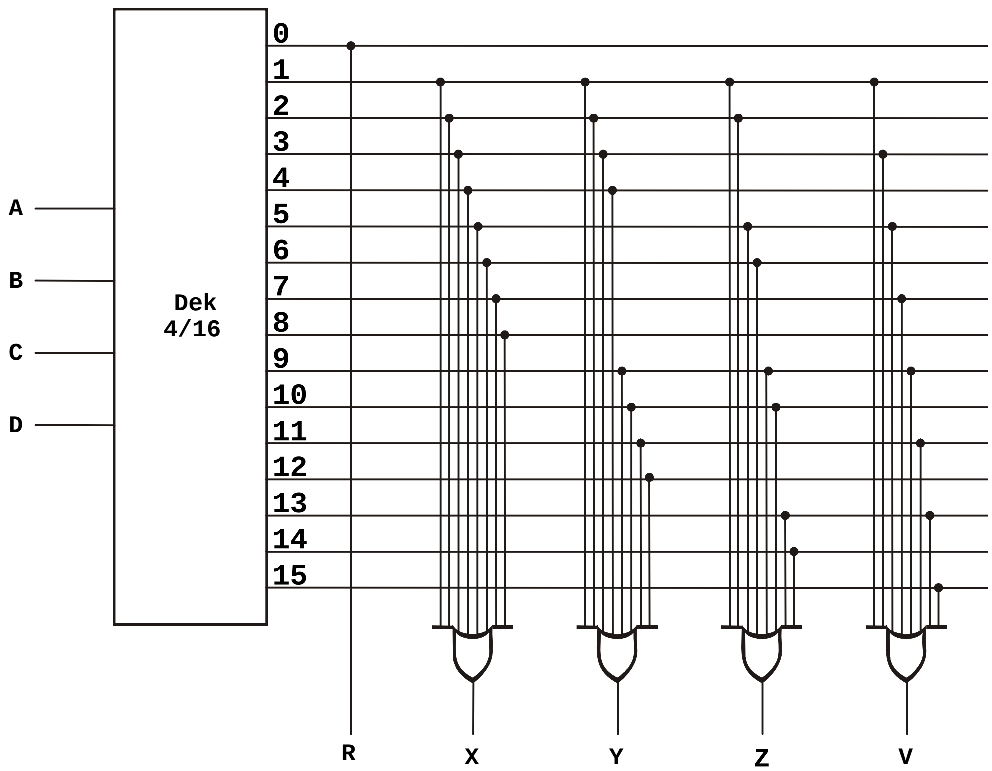


Fig.6.11 Qarku për gjetjen e 2-komplementit i realizuar me dekodues

Nga qarku logjik i dhënë më sipër shihet se hyrjet e elementeve për mbledhje janë të ngarkuara mjaft, gjë që mund të eliminohet duke përdorur struktura të lidhjes së elementeve logjike në më shumë nivele.

Në praktikë, gjatë vizatimit të qarqeve logjike të realizuara përmes dekoduesve, elementet **OSE** nuk vizatohen, por lidhjet përkatëse në dalje të dekoduesit shënohen me pika. Kështu, p.sh., qarku i dhënë në Fig.6.10 do të duket si në Fig.6.11.

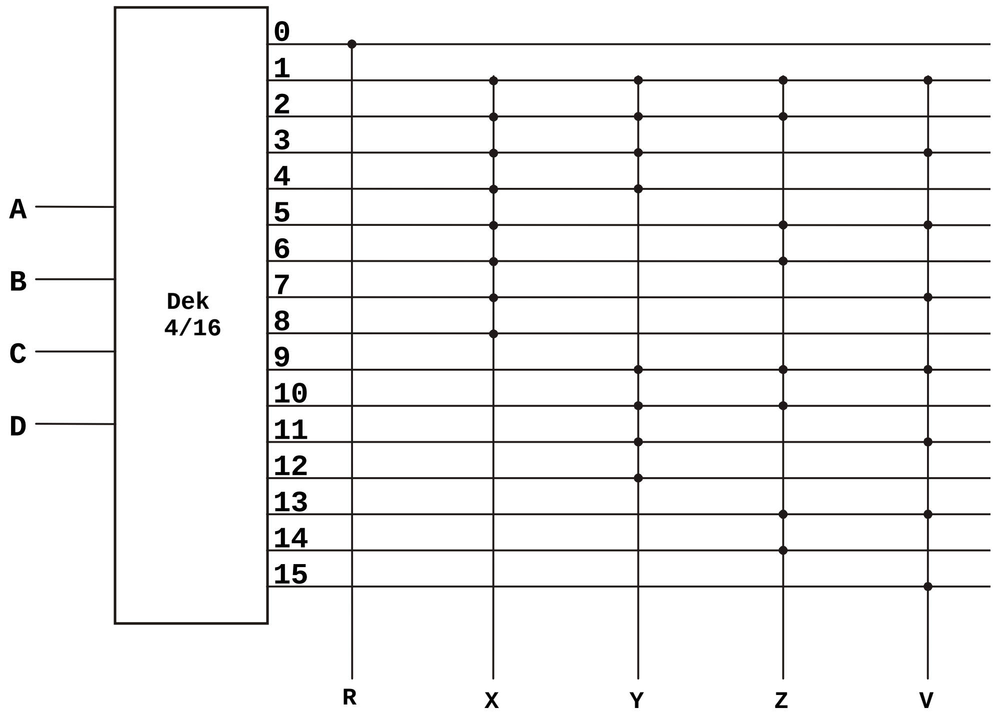


Fig.6.12 Paraqitja e thjeshtuar e qarkut me dekodues

Nëse qarku që duhet realizuar është përcaktuar përmes funksioneve logjike, për realizimin e tij përmes dekoduesit funksionet duhet të shprehen përmes shumës së mintermave të plota, përkatësisht shumës së numrave rendorë të mintermave.

Shembull

Qarku logjik i realizuar përmes një dekoduesi **4/16**, me të cilin përcaktohet funksioni:

$$f = \overline{B}\,\overline{D} + \overline{B}\,\overline{C} + \overline{A}\,C\,\overline{D}$$

Për gjetjen e mintermave të plota, të cilat marrin pjesë në shprehjen e funksionit përkatës, shfrytëzohen diagramet kohore të vizatuara për të gjitha kombinimet e mundshme të variablave hyrëse.

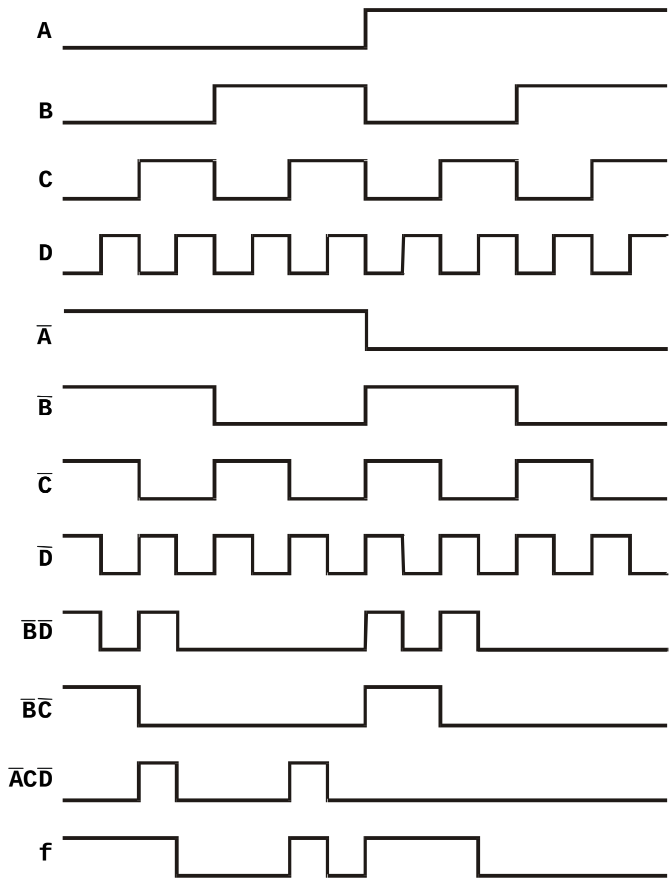


Fig.6.13 Diagramet kohore për të gjitha kombinimet e mundshme të variablave hyrëse

Nga diagramet kohore të dhënë më sipër mund të nxirret shprehja e funksionit dalës **f** si shumë e mintermave të plota me vlerë **1**:

$$f = \sum m^1(0,1,2,6,8,9,10)$$

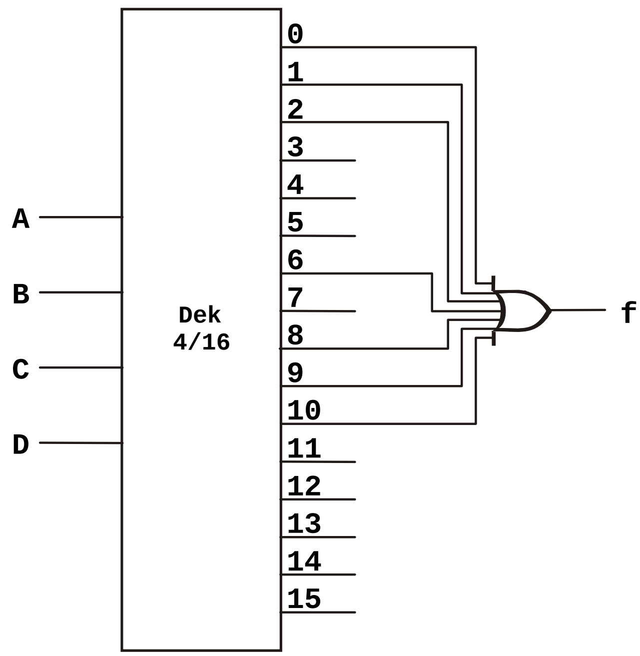


Fig.6.14 Realizimi i funksionit duke e shfrytëzuar dekoduesin

Gjatë realizimit të qarqeve logjike, përveç dekoduesve mund të shfrytëzohen edhe elemente logjike të zakonshme.

Shembull

Qarku logjik përmes të cilit gjenerohet shifra për çiftësi para numrave binarë **3**-shifror. Për realizimin e qarkut shfrytëzohet edhe një dekodues **2/4**.

A	B	C	p
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

$$P = \overline{A}\,\overline{B}\,C + \overline{A}\,B\,\overline{C} + A\,\overline{B}\,\overline{C} + A\,B\,C$$

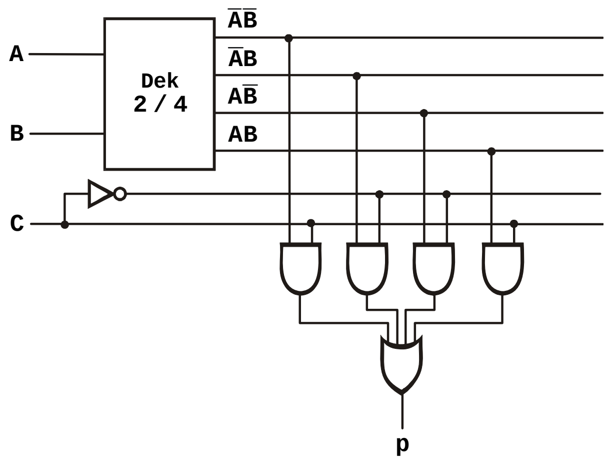


Fig.6.15 Dekoduesi në qarkun për gjenerimin e shifrave për çiftësi

7

Konvertuesit e kodeve

Sistemet e ndryshme digjitale, ose edhe njësitë e veçanta brenda tyre, për paraqitjen e informatave elementare shfrytëzojnë kode të ndryshme. Por, gjatë komunikimit mes tyre, për ta siguruar kompatibilitetin e nevojshëm, bëhet konvertimi i kodeve, përmes qarqeve të cilët njihen si *konvertues të kodeve* (ang. code converter).

Konvertuesit e zakonshëm

Konvertuesit e fjalëve kodike **m**-shifrore të një kodi, në fjalë kodike **n**-shifrore të kodit tjetër, njihen si *konvertues* **m** *në* **n** dhe skematikisht paraqiten si në Fig.7.1.

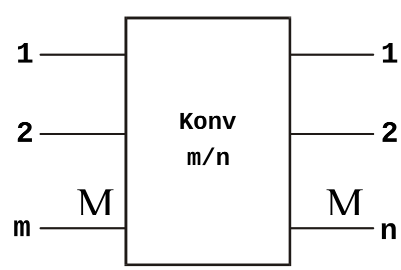
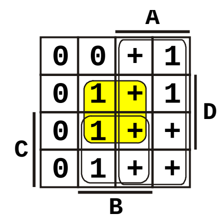


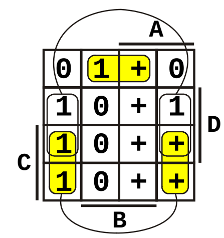
Fig.7.1 Paraqitja skematike e konvertuesit

Shembull Konvertuesi i kodit **NBCD** në kodin **Excess-3**, nëse fjalët kodike të cilat nuk përdoren merren si arbitrare.

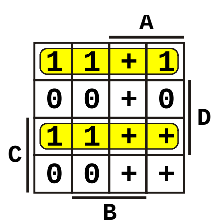
A	B	C	D	X	Y	Z	V
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0	+	+	+	+
1	0	1	1	+	+	+	+
1	1	0	0	+	+	+	+
1	1	0	1	+	+	+	+
1	1	1	0	+	+	+	+
1	1	1	1	+	+	+	+



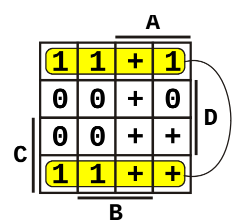
$$X = A + BD + BC$$



$$y = B\overline{C}\,\overline{D} + \overline{B}D + \overline{B}C$$



$$Z = CD + \overline{C}\,\overline{D}$$



$$V = \overline{D}$$

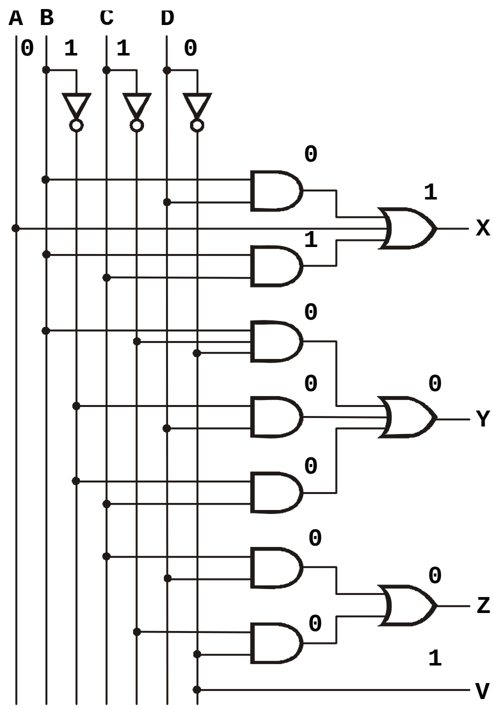
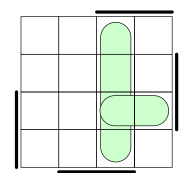


Fig.7.2 Konvertuesi **NBCD** - **Excess-3**

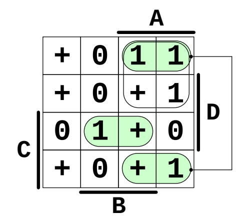
Që të jetë më e qartë mënyra e funksionimit të qarkut logjik të dhënë në Fig.7.2, le të marrim, p.sh., se në hyrje të tij vjen fjala kodike **0110**. Mund të provohet se në dalje të qarkut do të merret fjala kodike **1001**, gjë që i përgjigjet asaj që është shënuar në tabelën e kombinimeve që u dha më parë. Nga kjo shihet se për çdo kombinim **ABCD** të fjalëve kodike në hyrje të qarkut do të fitohet kombinimi përkatës **XYZV** i fjalëve kodike në dalje të qarkut. Nënkuptohet, për kombinimet e vlerave në hyrje të qarkut, të cilat nuk janë fjalë kodike të kodit **NBCD**, në dalje do të fitohen kombinime arbitrare dhe që nuk kanë rëndësi, sepse gjatë punës normale nuk mund të ndodhë që në hyrje të qarkut të paraqiten kombinime të tilla.

Shembull Konvertuesi i kodit **Excess-3** në kodin **NBCD**, i kundërt me atë që u dha në shembullin paraprak.

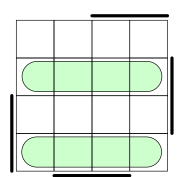
A	B	C	D	X	Y	Z	V
0	0	0	0	+	+	+	+
0	0	0	1	+	+	+	+
0	0	1	0	+	+	+	+
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	1
0	1	0	1	0	0	1	0
0	1	1	0	0	0	1	1
0	1	1	1	0	1	0	0
1	0	0	0	0	1	0	1
1	0	0	1	0	1	1	0
1	0	1	0	0	1	1	1
1	0	1	1	1	0	0	0
1	1	0	0	1	0	0	1
1	1	0	1	+	+	+	+
1	1	1	0	+	+	+	+
1	1	1	1	+	+	+	+



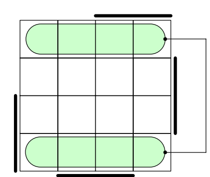
$$X = AB + ACD$$



$$Y = A\overline{C} + A\overline{D} + BCD$$



$$Z = C\overline{D} + \overline{C}D$$



$$V = \overline{D}$$

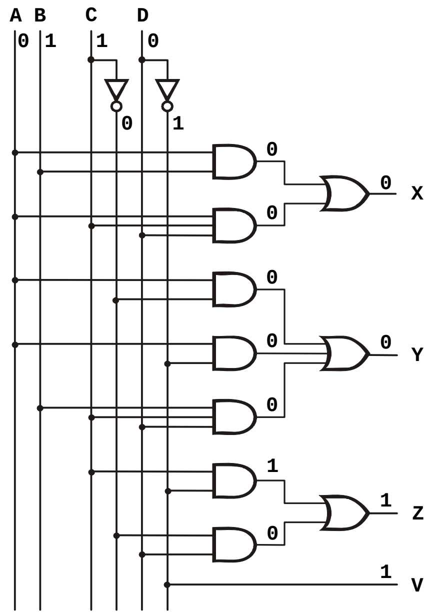


Fig.7.3 Konvertuesi ***Excess-3 - NBCD***

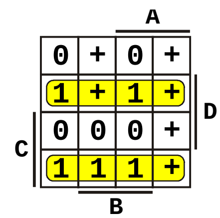
Paisjet ose komponentet e paisjeve që komunikojnë mes vete nuk është e domosdoshme të shfrytëzojnë kode me fjalë kodike të cilat kanë gjatësi të njëjtë. Për këtë arsye mund të realizohen edhe konvertues të kodeve te të cilët gjatësia e fjalëve kodike në hyrje dhe në dalje nuk është e njëjtë.

Shembull

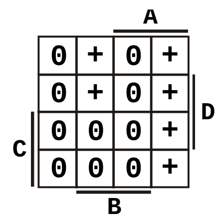
Konvertuesi i kodit **4221**, në kodin **ASCII**, tek i cili është shtuar edhe shifra për paritet **P**. Për fjalët kodike që nuk përdoren merren vlera arbitrare.

A	B	C	D	p	a	b	c	d	e	f	g
0	0	0	0	0	0	1	1	0	0	0	0
0	0	0	1	1	0	1	1	0	0	0	1
0	0	1	0	1	0	1	1	0	0	1	0
0	0	1	1	0	0	1	1	0	0	1	1
0	1	0	0	+	+	+	+	+	+	+	+
0	1	0	1	+	+	+	+	+	+	+	+
0	1	1	0	1	0	1	1	0	1	0	0
0	1	1	1	0	0	1	1	0	1	0	1
1	0	0	0	+	+	+	+	+	+	+	+
1	0	0	1	+	+	+	+	+	+	+	+

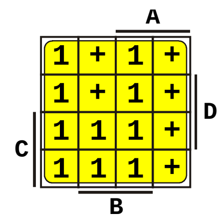
1	0	1	0	+	+	+	+	+	+	+	+
1	0	1	1	+	+	+	+	+	+	+	+
1	1	0	0	0	0	1	1	0	1	1	0
1	1	0	1	1	0	1	1	0	1	1	1
1	1	1	0	1	0	1	1	1	0	0	0
1	1	1	1	0	0	1	1	1	0	0	1



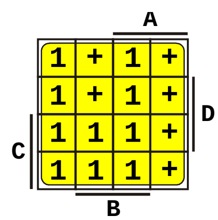
$P = C\bar{D} + \bar{C}D$



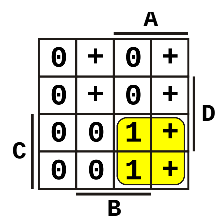
$a = 0$



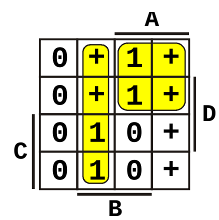
$b = 1$



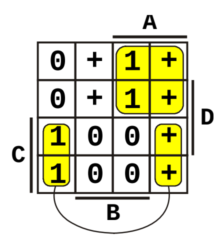
$c = 1$



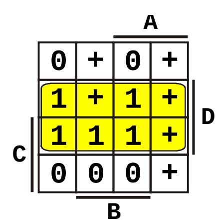
$d = AC$



$e = \bar{A}B + A\bar{C}$



$f = A\bar{C} + \bar{B}C$



$g = D$

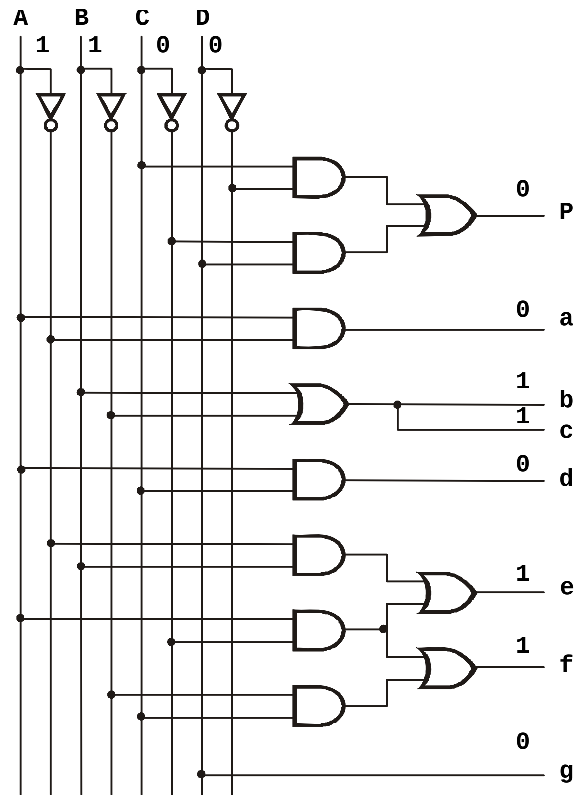


Fig.7.4 Konvertuesi 4221 - ASCII

Te qarku i dhënë më sipër funksionet **b** dhe **c**, të cilat gjatë gjithë kohës kanë vlerën **1**, janë realizuar përmes një elementi **OSE**, në hyrjet e të cilit vijnë vlerat e variablës dhe të kovariablës përkatëse, p.sh. **B** dhe $\overline{\mathbf{B}}$, sepse:

$$\mathbf{B} + \overline{\mathbf{B}} = \mathbf{1}$$

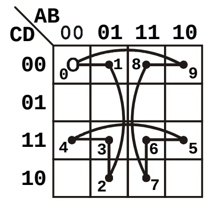
ku nuk ka rëndësi se cila prej **4** variablave merret për gjenerimin e kësaj vlere. Si vlerë **1** mund të merret direkt edhe vlera e sinjalit me të cilin paraqitet vlera logjike **1** (p.sh. tensioni **+5V**).

Funksionimi i qarkut logjik të konvertuesit që u dha më sipër mund të vërtetohet, nëse në hyrje të tij aplikohen fjalët kodike të kodit **BCD 4221**. Kështu, p.sh., për fjalën kodike **1100** në daljet **abcdefg** duhet të fitohet vargu i shifrave binare **00110110**, gjë që shihet edhe në tabelën përkatëse. Në daljen **P** të qarkut, në këtë rast fitohet shifra binare **0**, sepse numri i njësheve në fjalën kodike të kodit **ASCII** është çift. Kurse, për kombinimet të cilat nuk janë fjalë kodike të kodit **BCD 4221** në dalje të qarkut fitohen vargje të vlerave binare të cilat nuk janë të përdorshme.

Gjatë realizimit të konvertuesve të kodeve mund të përdoren elemente logjike të caktuara, siç janë, p.sh., elementet logjike universale, ose qarqet logjike të konvertuesve të kodeve mund të realizohen edhe përmes dekoduesve, gjë që më së miri shihet në shembujt vijues.

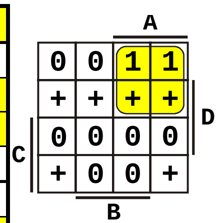
Shembull

Konvertuesi i kodit ciklik jokomplet, i cili është përcaktuar kështu:

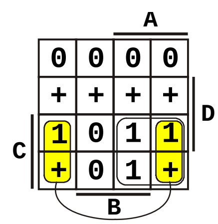


në numra binarë natyrorë prej **0** deri në **9**. Për realizimin e konvertuesit shfrytëzohen vetëm elemente logjike universale **JODHE**, kurse për kombinimet e vargjeve binare, të cilat nuk i takojnë kodit ciklik të dhënë, merren vlera arbitrare.

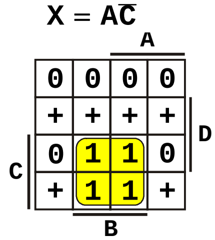
A	B	C	D	X	Y	Z	V
0	0	0	0	0	0	0	0
0	0	0	1	+	+	+	+
0	0	1	0	+	+	+	+
0	0	1	1	0	1	0	0
0	1	0	0	0	0	0	1
0	1	0	1	+	+	+	+
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	1
1	0	0	0	1	0	0	1
1	0	0	1	+	+	+	+
1	0	1	0	+	+	+	+
1	0	1	1	0	1	0	1
1	1	0	0	1	0	0	0
1	1	0	1	+	+	+	+
1	1	1	0	0	1	1	1
1	1	1	1	0	1	1	0



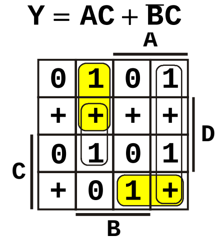
$X = A\overline{C}$



$Y = AC + \overline{B}C$



$Z = BC$



$V = A\overline{B} + AC\overline{D} + \overline{A}B\overline{C} + \overline{A}BD$

Pas konvertimeve të dyfishta, shprehjet e funksioneve dalëse në formën e tyre përfundimtare duken kështu:

$$X = \overline{\overline{A\overline{C}}}$$

$$Y = \overline{\overline{AC} \cdot \overline{\overline{B}C}}$$

$$Z = \overline{\overline{BC}}$$

$$V = \overline{\overline{A\overline{B}} \cdot \overline{AC\overline{D}} \cdot \overline{\overline{A}B\overline{C}} \cdot \overline{\overline{A}BD}}$$

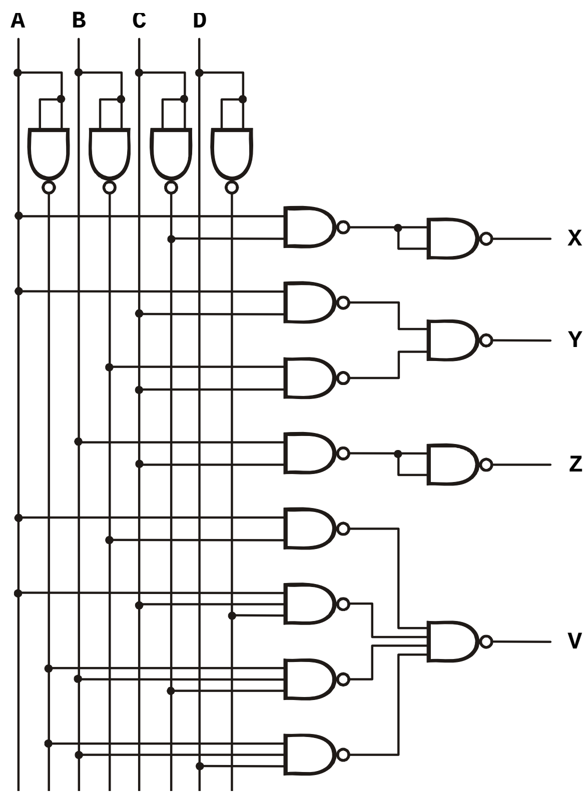


Fig.7.5 Konvertuesi i kodit ciklik në numra binar

Shembull Konvertuesi i kodit **BCD 84-2-1**, në *kodin* ***3*** *prej* ***5***, i realizuar përmes *dekoduesit* ***4*** *në* ***16***.

A	B	C	D	a	b	c	d	e
0	0	0	0	0	0	1	1	1
0	0	0	1	+	+	+	+	+
0	0	1	0	+	+	+	+	+
0	0	1	1	+	+	+	+	+
0	1	0	0	1	0	1	1	0
0	1	0	1	1	1	0	0	1
0	1	1	0	1	1	0	1	0
0	1	1	1	1	1	1	0	0
1	0	0	0	0	1	1	0	1
1	0	0	1	0	1	1	1	0
1	0	1	0	1	0	0	1	1
1	0	1	1	1	0	1	0	1
1	1	0	0	+	+	+	+	+
1	1	0	1	+	+	+	+	+
1	1	1	0	+	+	+	+	+
1	1	1	1	0	1	0	1	1

$$a = \sum m^1(4,5,6,7,10,11)$$

$$b = \sum m^1(5,6,7,8,9,15)$$

$$c = \sum m^1(0,4,7,8,9,11)$$

$$d = \sum m^1(0,4,6,9,10,15)$$

$$e = \sum m^1(0,5,8,10,11,15)$$

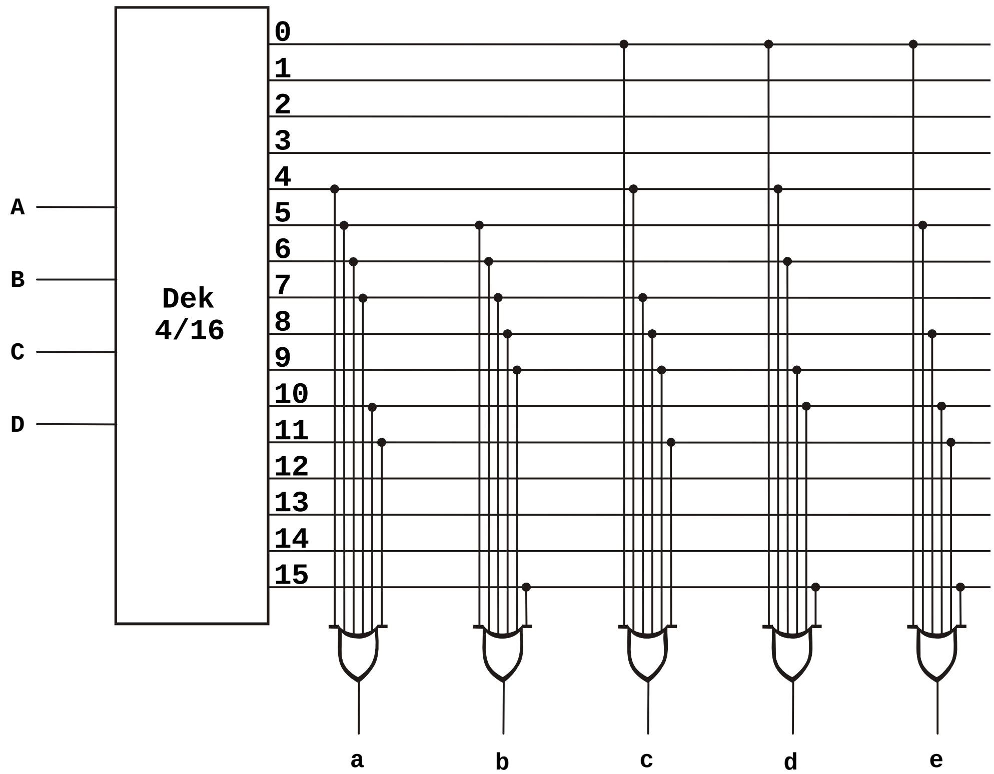


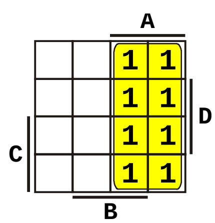
***Fig.7.6 Konvertuesi i kodit* 84-2-1 *në kodin* 3 *prej* 5**

Për ta kuptuar punën e qarkut të vizatuar, le të marrim, p.sh., se në hyrjet **ABCD** të tij aplikohet fjala kodike **1001**. Në dalje të dekoduesit, vetëm në daljen **9**, do të paraqitet sinjali me vlerë numerike **1**, kurse në të gjitha daljet e tjera sinjali ka vlerën **0**, përkatësisht vargu i sinjaleve dalëse nga dekoduesi është **0000000001000000**. Nëse shihet vizatimi i qarkut logjik, sinjali me vlerën **1** në daljen **9** të dekoduesit do të përcillet edhe në daljet **b**, **c** dhe **d** përmes lidhjeve ekzistuese të qarkut, përkatësisht vargu i sinjaleve në dalje të qarkut është **01110**.

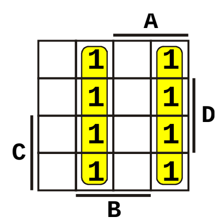
Gjatë realizimit të konvertuesve mund të përdoren edhe elemente logjike speciale, siç janë *eksluziv*-**OSE** dhe *ekskluziv*-**DHE**.

Shembull	Konvertuesi i numrave binarë **4**-shifror të kodit binar natyror, në fjalë kodike të kodit të *Gray*-it, duke shfrytëzuar vetëm elemente logjike *ekskluziv*-**OSE**.

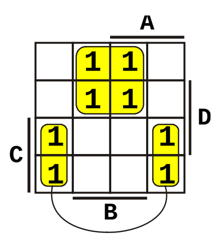
Binar				Gray			
A	B	C	D	X	Y	Z	V
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0



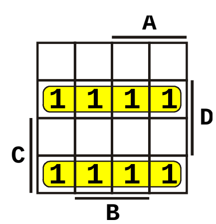
$x = A$



$$y = \overline{A}B + A\overline{B}$$
$$= A \oplus B$$



$$z = B\overline{C} + \overline{B}C$$
$$= B \oplus C$$



$$v = \overline{C}D + C\overline{D}$$
$$= C \oplus D$$

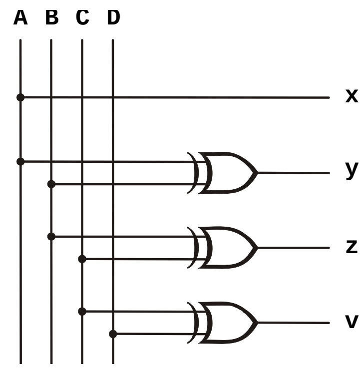
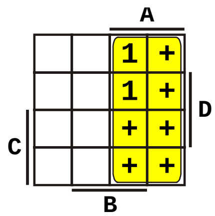


Fig.7.7 Konvertuesi i numrave natyror në kodin e* `Gray`*-it

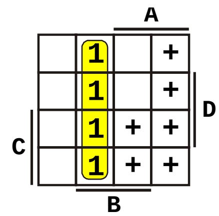
Qarqet logjike mund të thjeshtësohen edhe në drejtim të asaj që për disa prodhime logjike të njëjta të përdoret vetëm një element logjik dhe pastaj dalja e tij të përdoret sa herë që nevojitet prodhimi përkatës.

Shembull Konvertuesi i kodit të *Gray*-it, i cili është dhënë në kolonën e parë të tabelës vijuese, në kodin **NBCD**.

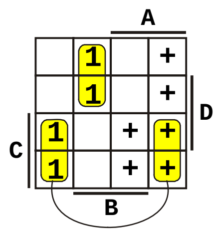
Gray				NBCD			
A	B	C	D	X	Y	Z	V
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	1	0	0	1	0
0	0	1	0	0	0	1	1
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	1
0	1	0	1	0	1	1	0
0	1	0	0	0	1	1	1
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	1



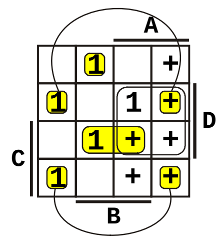
$x = A$



$y = \bar{A}B$



$Z = \bar{B}C + \bar{A}B\bar{C}$



$v = \bar{A}B\bar{C}\bar{D} + \bar{B}\bar{C}D$
$+ AD + BCD + \bar{B}C\bar{D}$

Shprehjet e funksioneve dalëse mund të shkruhen edhe kështu:

$$X = A$$
$$y = \bar{A}B$$
$$Z = \bar{B}C + Y\bar{C}$$
$$V = \bar{D}(\bar{B}C + Y\bar{C}) + D(A + \bar{B}\bar{C} + BC)$$
$$= \bar{D}Z + D(A + \bar{B}\bar{C} + BC)$$

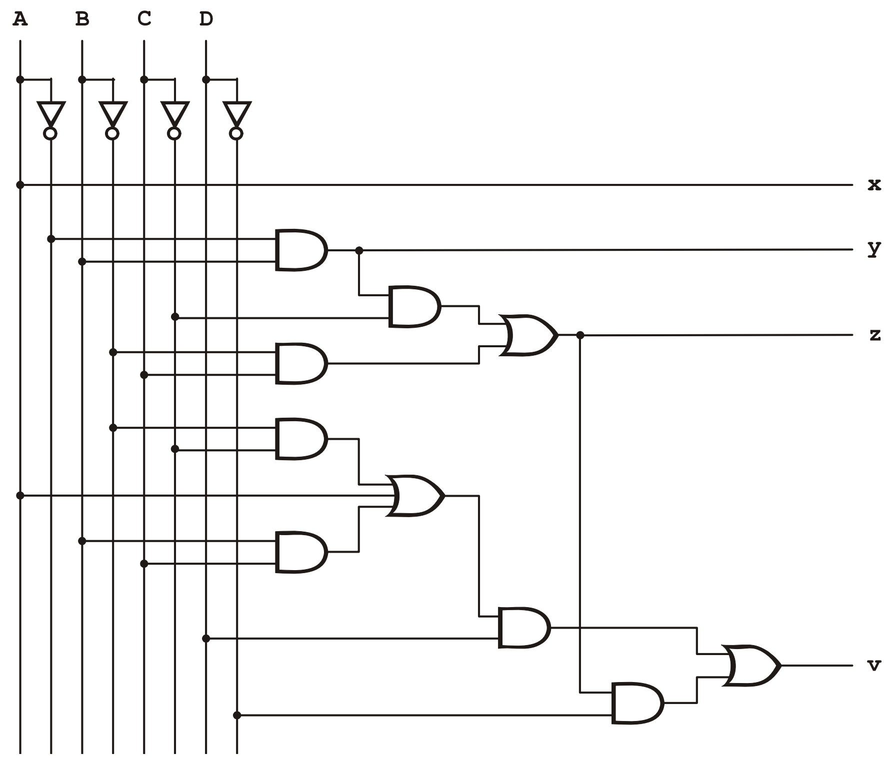


Fig.7.8 Konvertuesi **Gray** *-* ***NBCD***

Nëse për të gjitha kombinimet e mundshme të vlerave hyrëse, në një ose më shumë dalje të kovertuesit, vlerat dalëse janë **0**, për gjenerimin e kësaj vlere mund të përdoret një element logjik **DHE**, në hyrjet e të cilit lidhet njëra nga variablat dhe komplementi i saj, sepse, p.sh.:

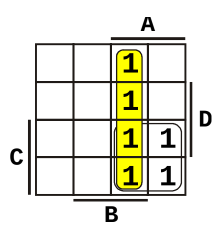
$$A \cdot \overline{A} = 0$$

Si vlerë **0** mund të merret edhe sinjali përmes të cilit paraqitet vlera logjike **0**, p.sh. tensioni **0V**.

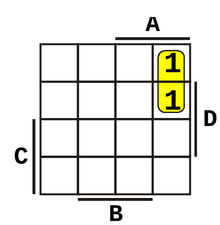
Shembull	Konvertuesi i kodit binar me fjalë kodike **4**-shifrore, në numra decimalë, të cilët shprehen përmes kodit **NBCD**. Kështu, p.sh., për numrin binar **1100** në **8** daljet e qarkut logjik fitohet vargu i shifrave binare **0001 0010**, i cili e paraqet ekuivalentin e numrit decimal **12**, të shprehur në kodin **NBCD**.

Binar					NBCD							
N	A	B	C	D	a	b	c	d	e	f	g	h
0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0	0	0	0	1
2	0	0	1	0	0	0	0	0	0	0	1	0
3	0	0	1	1	0	0	0	0	0	0	1	1
4	0	1	0	0	0	0	0	0	0	1	0	0
5	0	1	0	1	0	0	0	0	0	1	0	1
6	0	1	1	0	0	0	0	0	0	1	1	0
7	0	1	1	1	0	0	0	0	0	1	1	1
8	1	0	0	0	0	0	0	0	1	0	0	0
9	1	0	0	1	0	0	0	0	1	0	0	1
10	1	0	1	0	0	0	0	1	0	0	0	0
11	1	0	1	1	0	0	0	1	0	0	0	1
12	1	1	0	0	0	0	0	1	0	0	1	0
13	1	1	0	1	0	0	0	1	0	0	1	1
14	1	1	1	0	0	0	0	1	0	1	0	0
15	1	1	1	1	0	0	0	1	0	1	0	1

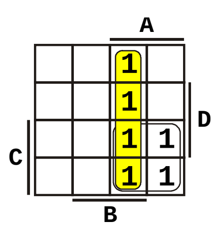
a=0
b=0
c=0



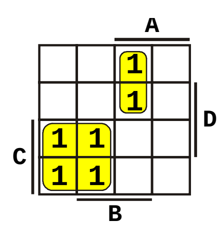
$d = AB + AC$



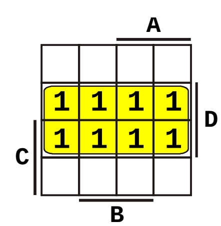
$e = A\overline{B}\overline{C}$



$f = \overline{A}B + BC$



$g = AB\overline{C} + \overline{A}C$



$h = D$

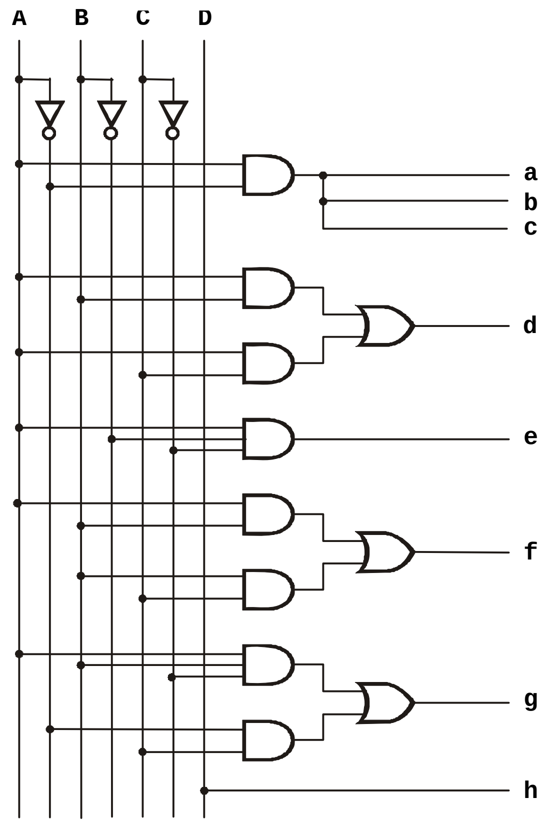


Fig.7.9 Konvertuesi i numrave binar në numra decimal të shprehur në kodin NBCD

Konvertuesit paralelë

Meqë te kodi i *Gray*-it fjalët kodike të njëpasnjëshme dallohen vetëm në një pozicion, gjatë konvertimit të tij në kodin binar dhe anasjelltas mund të përdoren qarqe logjike që paraqesin *konvertues paralelë*.

Konvertuesi binar-Gray

Duke pasur parasysh tabelën e raportit të fjalëve kodike të kodit binar natyror dhe të kodit të *Gray*-it:

NBCD				Gray			
A	B	C	D	x	y	z	v
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1

nëse fjala kodike binare shënohen me $b_4b_3b_2b_1$, fjala kodike përkatëse $g_4g_3g_2g_1$ në kodin e Gray-it mund të gjendet përmes shprehjeve:

$$g_4 = b_4 \oplus b_5$$
$$g_3 = b_3 \oplus b_4$$
$$g_2 = b_2 \oplus b_3$$
$$g_1 = b_1 \oplus b_2$$

ku b_5**=0** është shifra e **5**-të në fajlët kodike të kodit binar, sikur ato të kishin gjatësi **5**-shifrore. Kështu, p.sh., për fjalën kodike binare **0110** shifrat e fjalës kodike përkatëse **0101** në kodin e *Gray*-it gjinden kështu:

$$g_4 = 0 \oplus 0 = 0$$
$$g_3 = 1 \oplus 0 = 1$$
$$g_2 = 1 \oplus 1 = 0$$
$$g_1 = 0 \oplus 1 = 1$$

Prej këtu, mund të nxirret shprehja e përgjithshme:

$$\begin{aligned} g_k &= b_k \oplus b_{k+1} \\ &= b_k \cdot \overline{b}_{k+1} + \overline{b}_k \cdot b_{K+1} \end{aligned}$$

ku për **k=4**, vlera b_{k+1}**=0**. Duke pasur parasysh shprehjen e dhënë, qarku logjik i *konvertuesit paralel binar-Gray* do të duket si në Fig.7.10.

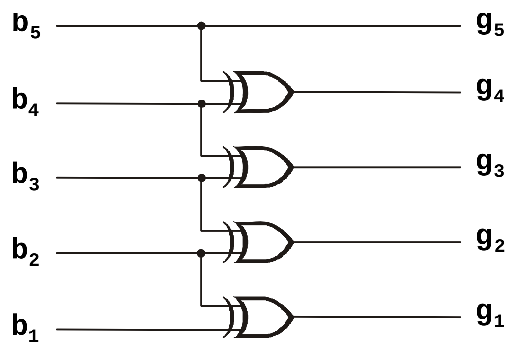


Fig.7.10 Konveruesi paralel binar - Gray

Konvertuesi Gray-binar

Raporti në mes të fjalëve kodike të kodit të *Gray*-it dhe atij binar është dhënë në tabelën vijuese.

Gray				NBCD			
A	B	C	D	X	Y	Z	v
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	1	0	0	1	0
0	0	1	0	0	0	1	1
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	1
0	1	0	1	0	1	1	0
0	1	0	0	0	1	1	1
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	1

Mbështetur në principin e konvertimit *binar-Gray* të dhënë më sipër, shifrat e fjalëve kodike gjatë konvertimit *Gray-binar* mund të gjenden përmes shprehjeve:

$$b_4 = g_4 \oplus b_5$$
$$b_3 = g_3 \oplus b_4$$
$$b_2 = g_2 \oplus b_3$$
$$b_1 = g_1 \oplus b_2$$

ku $b_5=0$, meqë fjalët kodike janë **4**-shifrore. Shprehjet e dhëna më sipër, në formë të përgjithshme mund të përcaktohen me shprehjen:

$$\begin{aligned} b_k &= g_k \oplus b_{k+1} \\ &= g_k \cdot \overline{b}_{k+1} + \overline{g}_k \cdot b_{K+1} \end{aligned}$$

ku për $k=4$, vlera $b_{k+1}=0$. Mbështetur në shprehjen e dhënë, qarku logjik i *konvertuesit paralel Gray-binar* do të duket si në Fig.7.11.

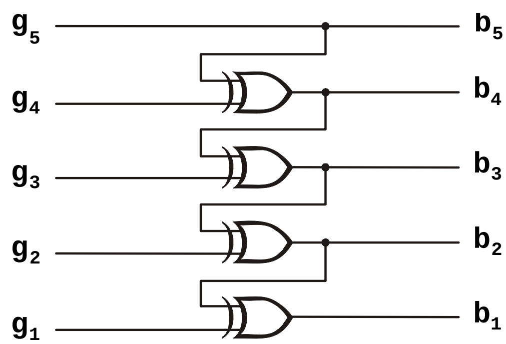


Fig.7.11 Konvertuesi paralel Gray-binar

Nëse, p.sh., në hyrje të konvertuesit zbatohet fjala kodike **0101**, në dalje të tij fitohet fjala **0110,** e cila i përket kodit binar. Shifrat e veçanta të kësaj fjale kodike mund të gjenden përmes shprehjeve të dhëna më sipër:

$$b_4 = 0 \oplus 0 = 0$$
$$b_3 = 1 \oplus 0 = 1$$
$$b_2 = 0 \oplus 1 = 1$$
$$b_1 = 1 \oplus 1 = 0$$

ose edhe direkt në daljet e konvertuesit, nëse në hyrje të tij zbatohet fjala kodike në fjalë.

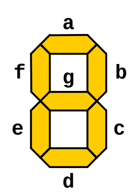
Indikatorët 8

Rezultatet që fitohen me përpunimin e informatave që vijnë në hyrjet e qarqeve logjike në formë të sinjaleve binare, në dalje të tyre mund të merren gjithashtu si sinjale binare. Por, këto rezultate mund të paraqiten edhe në një formë të dukshme, duke lidhur *dioda ndriçuese* në daljet e qarqeve, përkatësisht *dioda që emetojnë dritë* (nga Light Emiting Diode, **LED**), ose *indikatorë* të tjerë me *elemente ndriçuese*, siç janë, p.sh., indikatorët me një numër të caktuar segmentesh, pikash ose katrorësh ndriçues.

Qarqet përmes të cilëve komandohet puna e elementeve ndriçuese të indikatorëve paraqesin dekodues.

Indikatori 7-segmentësh

Në formë të përgjithshme, indikatori **7**-segmentësh duket kështu:



Në të, përmes ndriçimit të segmenteve të caktuara, mund të gjenerohen, p.sh, shifrat decimale në këtë mënyrë:

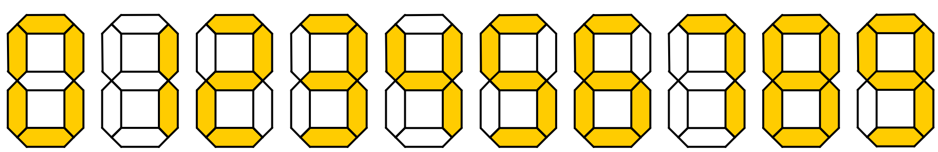
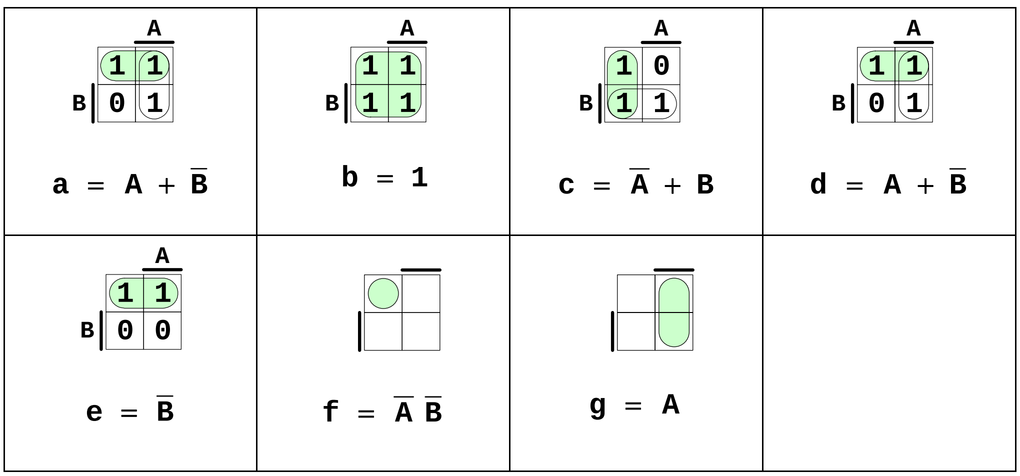


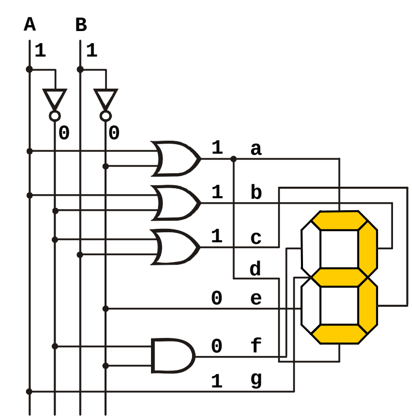
Fig.8.1 Gjenerimi i 9 shifrave decimale në indikatorin 7-segmentësh

Për komandimin e ndriçimit të segmenteve të veçantë duhet të realizohet qarku logjik përkatës.

Shembull Dekoduesi përmes të cilit në indikatorin **7**-segmentësh gjenerohen shifrat decimale prej **0** deri në **3**, nëse në hyrje të qarkut aplikohen ekuivalentët binarë përkatës.

A	B	Numri	a	b	c	d	e	f	g
0	0	0	1	1	1	1	1	1	0
0	0	1	0	1	1	0	0	0	0
0	0	2	1	1	0	1	1	0	1
0	0	3	1	1	1	1	0	0	1





A

1 0

Fig.8.2 Dekoduesi për gjenerimin e 4 shifrave decimale në indikatorin 7-segmentësh

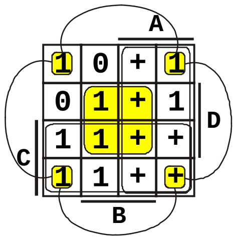
Numri **3** që është paraqitur në indikator, i përgjigjet kombinimit të vlerave hyrëse **11**, i cili është aplikuar në hyrjet **A** dhe **B** të qarkut.

Shembull

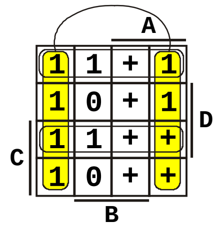
Dekoduesi përmes të cilit në indikatorin **7**-segmentësh gjenerohen shifrat decimale prej **0** deri në **9**, nëse në hyrje të qarkut aplikohen fjalët kodike përkatëse të kodit **NBCD**.

A	B	C	D	Numri	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	1	0	1	1	0	0	0	0
0	0	1	0	2	1	1	0	1	1	0	1
0	0	1	1	3	1	1	1	1	0	0	1
0	1	0	0	4	0	1	1	0	0	1	1
0	1	0	1	5	1	0	1	1	0	1	1
0	1	1	0	6	1	0	1	1	1	1	1
0	1	1	1	7	1	1	1	0	0	0	0
1	0	0	0	8	1	1	1	1	1	1	1
1	0	0	1	9	1	1	1	1	0	1	1
1	0	1	0	-	+	+	+	+	+	+	+
1	0	1	1	-	+	+	+	+	+	+	+
1	1	0	0	-	+	+	+	+	+	+	+
1	1	0	1	-	+	+	+	+	+	+	+
1	1	1	0	-	+	+	+	+	+	+	+
1	1	1	1	-	+	+	+	+	+	+	+

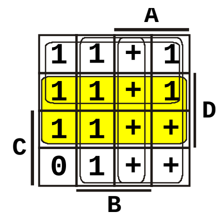
Këtu, për kombinimet që aplikohen në hyrje të cilat nuk janë fjalë kodike të kodit **NBCD**, vlerat dalëse përkatëse janë marrë si vlera arbitrare (**+**).



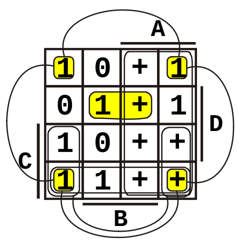
$a = A + \overline{B}\overline{D} + C + BD$



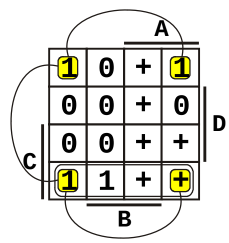
$b = \overline{B} + CD + \overline{C}\overline{D}$



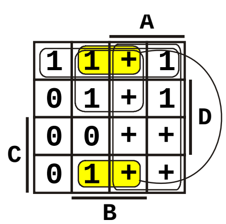
$c = B + \overline{C} + D$



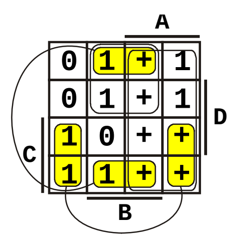
$d = A + \overline{B}\overline{D} + \overline{B}C + C\overline{D} + B\overline{C}D$



$e = \overline{B}\overline{D} + C\overline{D}$



$f = A + B\overline{C} + \overline{C}\overline{D} + B\overline{D}$



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

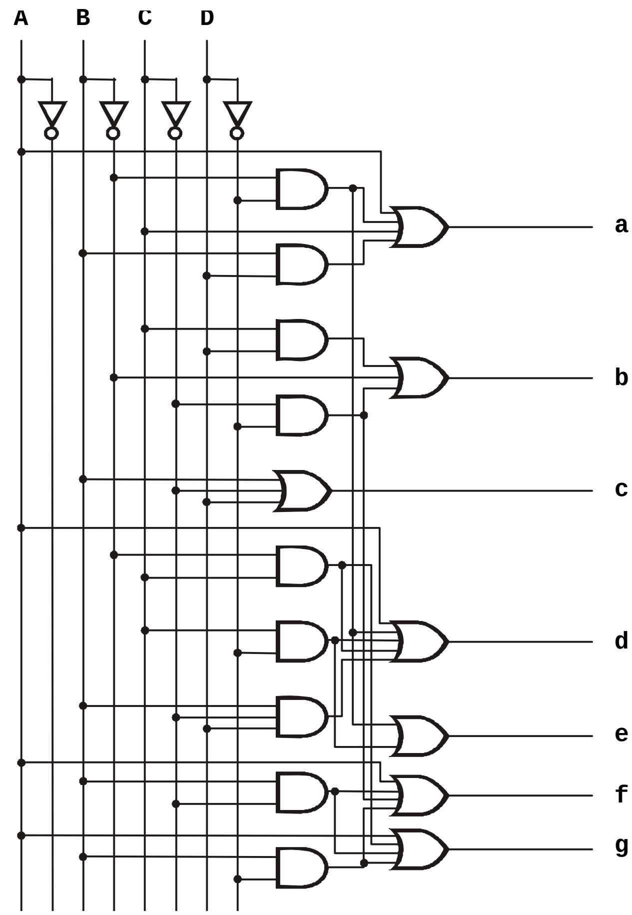
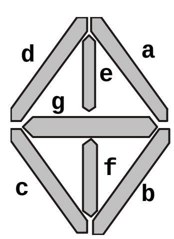
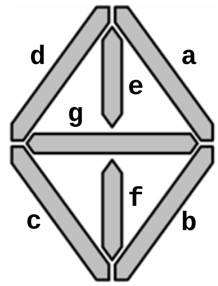


Fig.8.3 Dekoduesi për gjenerimin e shifrave decimale në indikatorin 7-segmentësh

Indikatori **7**-segmentësh mund të ketë edhe organizim tjetër të segmentëve brenda tij, p.sh., kështu:



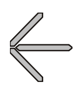


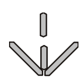
Në një indikator të tillë, me zgjedhjen e kombinimeve të caktuara të segmenteve, mund të gjenerohen edhe simbole të caktuara, siç janë, p.sh., shigjetat.

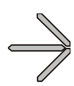
Shembull

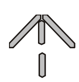
Qarku logjik përmes të cilit në indikatorin **7**-segmentësh të dhënë më sipër, për vlerat e ndryshme hyrëse $\mathbf{h=(ABCD)_2}$, gjenerohen simbolet:

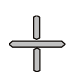
 për **h=0**

 për **h=1,2,3**

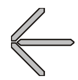
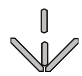
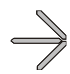
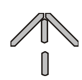
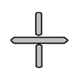
 për **h=4,5,6,7,8**

 për **h=9,10,11**

 për **h=12,13,14**

 për **h=15**

Për realizimin e qarkut logjik shfrytëzohet dekoduesi **4** *në* **16**.

N	A	B	C	D	Simboli	a	b	c	d	e	f	g
0	0	0	0	0								1
1	0	0	0	1				1	1			1
2	0	0	1	0				1	1			1
3	0	0	1	1				1	1			1
4	0	1	0	0			1	1		1	1	
5	0	1	0	1			1	1		1	1	
6	0	1	1	0			1	1		1	1	
7	0	1	1	1			1	1		1	1	
8	1	0	0	0			1	1		1	1	
9	1	0	0	1		1	1					1
10	1	0	1	0		1	1					1
11	1	0	1	1		1	1					1
12	1	1	0	0		1			1	1	1	
13	1	1	0	1		1			1	1	1	
14	1	1	1	0		1			1	1	1	
15	1	1	1	1						1	1	1

$$a = \sum m^1(9-14)$$

$$b = \sum m^1(4-11)$$

$$c = \sum m^1(1-8)$$

$$d = \sum m^1(1-3, 12-14)$$

$$e = \sum m^1(4-8, 12-15)$$

$$f = e$$

$$g = \sum m^1(0-3, 9-11, 15)$$

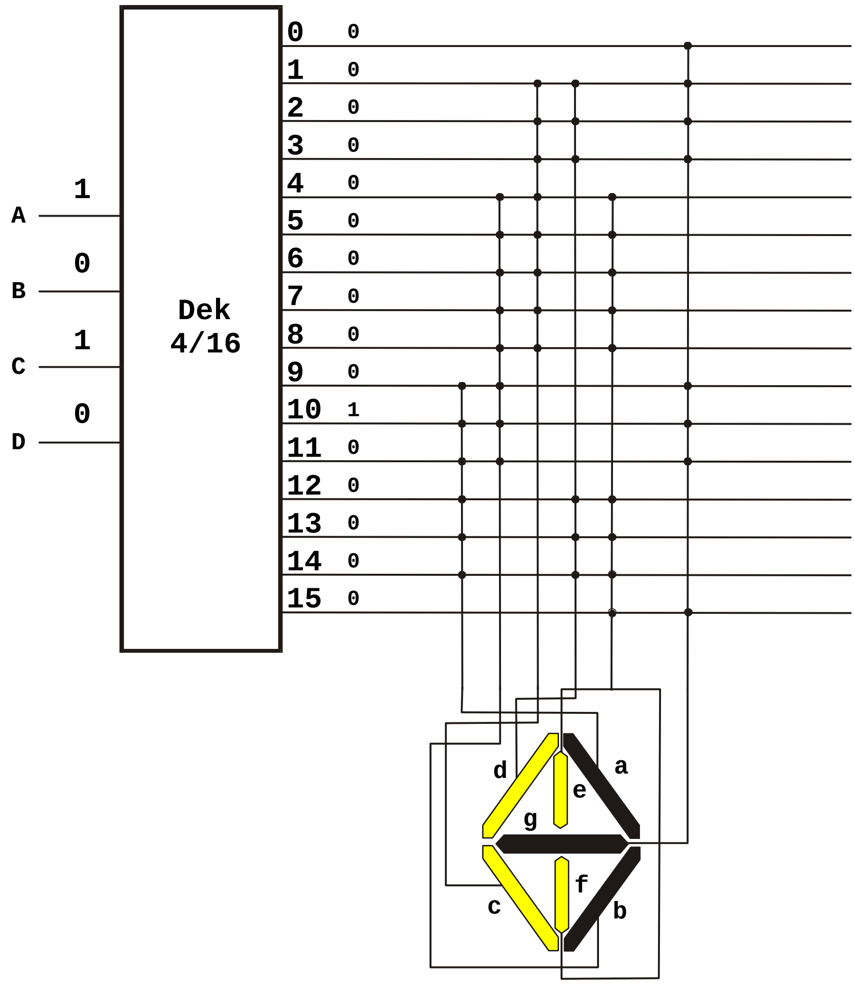
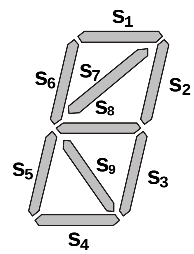


Fig.8.4 Qarku për gjenerimin e simboleve të ndryshme në indikatorin 7-segmentësh

Shigjeta që shihet në vizatimin e dhënë më sipër fitohet nëse në hyrje të qarkut aplikohet kombinimi i vlerave binare **1010**.

Indikatori 9-segmentësh

Organizimi i segmenteve në indikatorin **9**-segmentësh duket kështu:



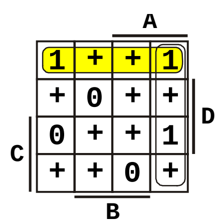
Për realizimin e qarkut logjik përmes të cilit do të ndriçohen kombinime të caktuara të segmenteve brenda indikatorit, duhet ndjekur procedurat e dhëna më sipër, për ndriçimin e segmenteve te indikatorët **7**-segmentësh.

Shembull

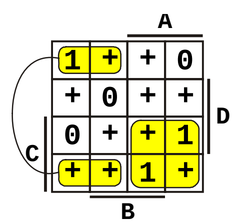
Qarku logjik, përmes të cilit në indikatorin **9**-segmentësh të treguar më sipër do të gjenerohen shkronjat: **P**, **K**, **R**, **L**, **S** dhe **U**, duke zbatuar në hyrje të qarkut për secilën nga shkronjat e dhëna fjalët kodike të përcaktuara përmes tabelës:

CD \ AB	00	01	11	10
00	P			S
01		K		
11	L			R
10			U	

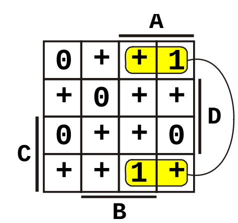
Simboli	Kodi				Segmenti								
	A	B	C	D	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9
P	0	0	0	0	1	1	0	0	1	1	0	1	0
K	0	1	0	1	0	0	0	0	1	1	1	0	1
R	1	0	1	1	1	1	0	0	1	1	0	1	1
L	0	0	1	1	0	0	0	1	1	1	0	0	0
S	1	0	0	0	1	0	1	1	0	1	0	1	0
U	1	1	1	0	0	1	1	1	1	1	0	0	0



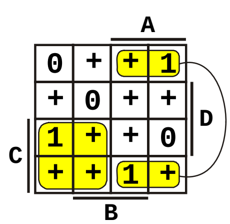
$s_1 = A\bar{B} + \bar{C}\bar{D}$



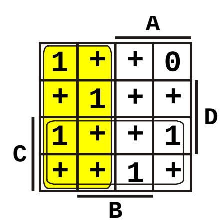
$s_2 = \bar{A}\bar{D} + AC$



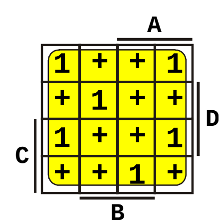
$s_3 = A\bar{D}$



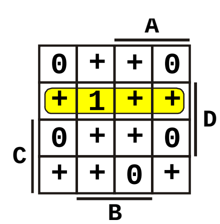
$s_4 = A\bar{D} + \bar{A}C$



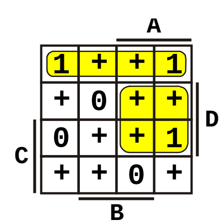
$s_5 = \bar{A} + C$



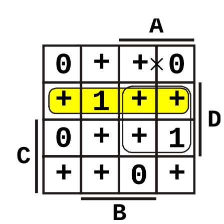
$s_6 = 1$



$s_7 = \bar{C}D$



$s_8 = AD + \bar{C}\bar{D}$



$s_9 = AD + \bar{C}D$

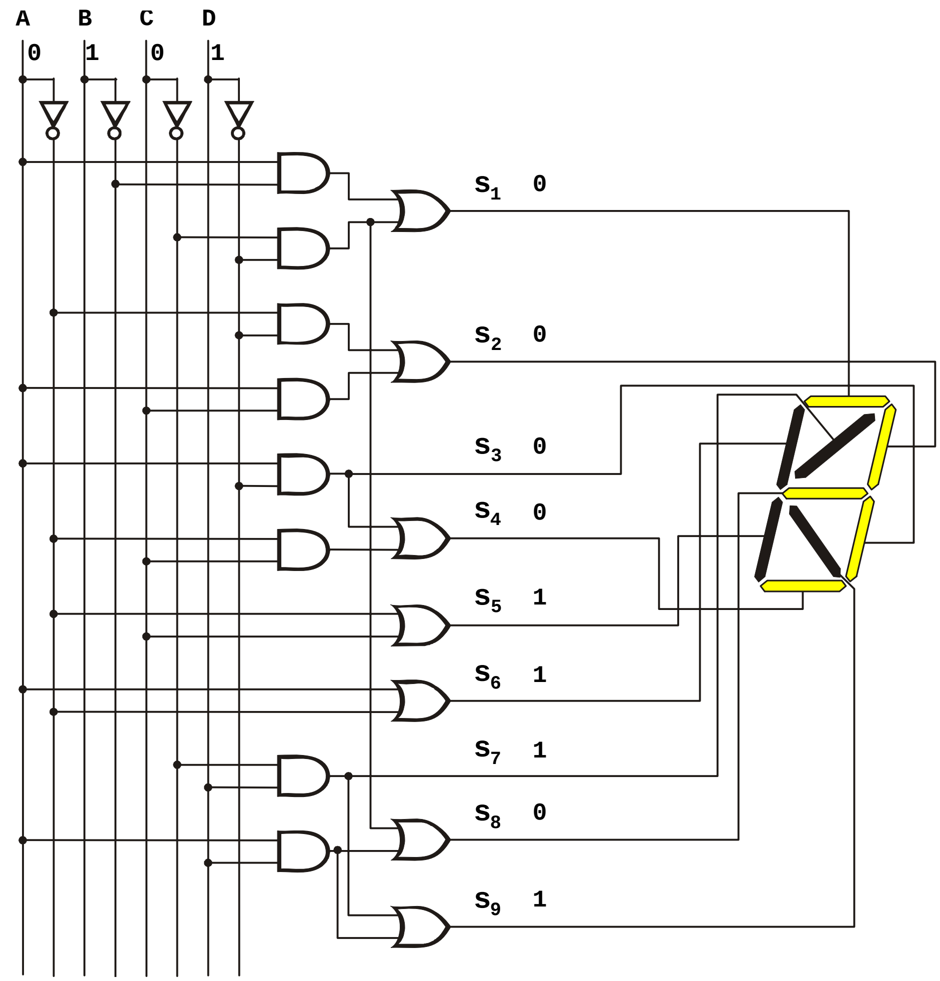


Fig.8.5 Qarku për gjenerimin e shkronjave në indikatorin 9-segmentësh

Shkronja **K**, që është paraqitur në indikator, i përgjigjet kombinimit të vlerave hyrëse **0101**, i cili është aplikuar në hyrje të qarkut.

9 Multiplekserët

Multiplekseri (ang. multiplexer) është qark digjital, i cili, duke shfrytëzuar **n**-sinjale seleksionuese, informatat nga 2^n-hyrjet e tij i përcjell në daljen **d**, sipas një radhe të caktuar. Skematikisht multiplekseri mund të paraqitet kështu:

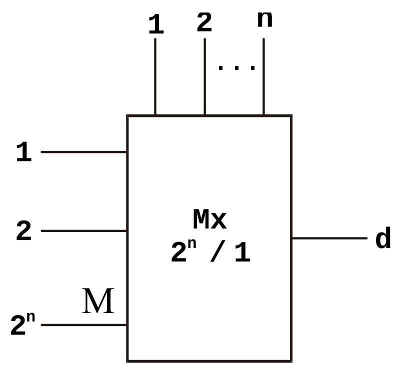


Fig.9.1 Paraqitja skematike e multipleksierit

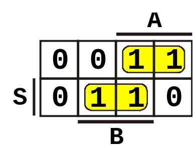
Përmes multipleksierit mundësohet përdorimi i një kanali transmetues të informatave nga më shumë shfrytëzues njëkohësisht. Gjithashtu, gjatë përdorimit të njëkohëshëm të një kompjuteri nga më shumë shfrytëzues, në regjimin e punës me *ndarje kohe* (ang. time-sharing), qasja në kompjuter bëhet me ndërmjetësimin e multipleksierit adekuat.

Multiplekseri 2/1

Dy vargje të informatave elementare mund të paketohen, ashtu që vargu i ri të formohet duke marrë informatat elementare suksesivisht nga njëri dhe vargu tjetër, përmes multipleksierit **2 në 1**, i cili shkurt shënohet edhe si **2/1**.

Shembull Multiplekseri **2/1**, përmes së cilit informatat elementare që vijnë në dy hyrjet e tij **A** dhe **B** përcillen me një radhë të caktuar në daljen **d**. Për zgjedhjen e informatës hyrëse, që duhet përcjellë në daljen e qarkut, përdoret sinjali seleksionues **S**, i cili duket kështu:

A	B	S	d
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1



$$\mathbf{d} = \mathbf{A\bar{S} + BS}$$

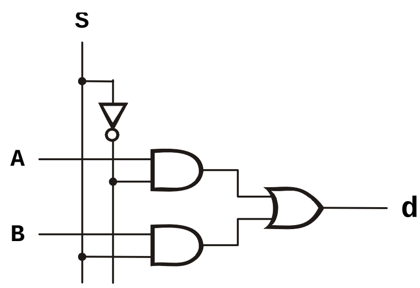


Fig.9.2 Multiplekseri 2/1

Për ta kuptuar mënyrën e funksionimit të qarkut të multiplekserit, p.sh., le të marrim se në dy hyrjet e tij vijnë vargjet e informatave binare:

A=10110...
B=11011...

Në momentin kur sinjali seleksionues **S** ka vlerën **0**, në daljen **d** përcillet vlera e parë **1** nga vargu i informatave hyrëse **A**. Pastaj, meqë në momentin vijues sinjali seleksionues **S** ka vlerën **1**, në daljen **d** përcillet vlera e parë nga vargu i informatave hyrëse **B**. Pas kësaj përsëritet suksesivisht procedura e përcjelljes në daljen **d**, të informatave elementare nga vargjet e sinjaleve hyrës **A** e **B**, për ta fituar në dalje të qarkut vargun e paketuar të informatave elementare:

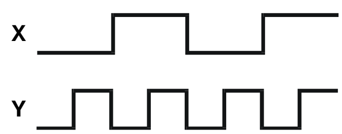
d=1101101101...

Multiplekseri 4/1

Numri i sinjaleve seleksionuese, përkatësisht numri i kombinimeve të vlerave të tyre, e përcakton numrin e hyrjeve të mundshme të multiplekserit, të cilat mund të zgjidhen për t'i lidhur me daljen e tij. Kështu, te multiplekseri **4/1**, përdoren **2** sinjale seleksionuese, meqë numri i kombinimeve të mundshme të vlerave të tyre është $\mathbf{2^2=4}$.

Shembull

Multiplekseri **4/1**, me hyrjet $\mathbf{I_0}$, $\mathbf{I_1}$, $\mathbf{I_2}$ e $\mathbf{I_3}$, daljen **f** dhe sinjalet seleksionuese **X** e **Y**, të cilat ndryshojnë sipas diagrameve kohore të dhëna më poshtë.



$$f = I_0\overline{X}\,\overline{Y} + I_1\overline{X}Y + I_2X\overline{Y} + I_3XY$$

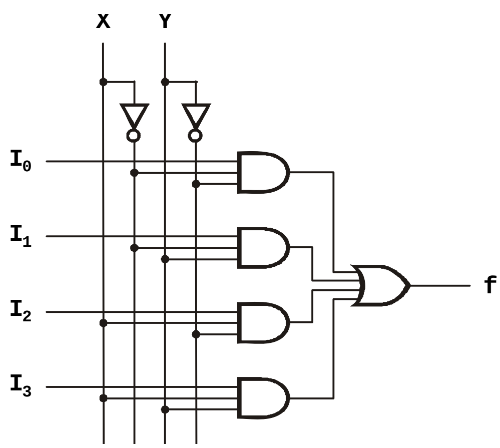


Fig.9.3 Multiplekseri 4/1

Funksionimi i multiplekserit **4/1**, i cili është dhënë më sipër, mund të tregohet në shembullin e vargjeve të informatave binare që vijnë në katër hyrjet e tij:

$\mathbf{I_0=10111...}$
$\mathbf{I_1=00100...}$
$\mathbf{I_2=01011...}$
$\mathbf{I_3=11010...}$

Vargu që fitohet në dalje të multiplekserit, si rezultat i paketimit të këtyre katër vargjeve me informata elementare, do të duket kështu:

f=1001001111001010...

ku katër shifrat e para binare në varg janë katër shifrat e para të vargjeve I_0, I_1, I_2 e I_3 dhe kështu me radhë, për grupin e katër shifrave të dyta, të treta, të katërta dhe të pesta.

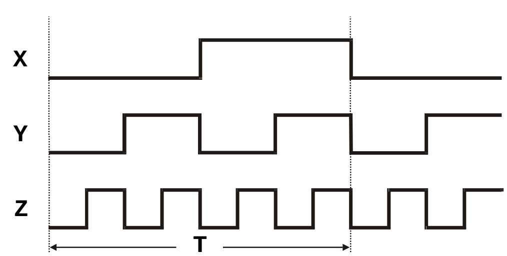
Multipleksheri 8/1

Në rastin e përgjithshëm, funksioni dalës nga multiplekseri me **n**-hyrje mund të shkruhet kështu:

$$f = \sum_{k=0}^{2^n-1} I_k \cdot m_k$$

ku me I_k janë shënuar hyrjet në multiplekser, kurse për selektimin e përcjelljes së informatave në dalje të multiplekserit përdoren mintermat m_k të sinjaleve seleksionuese.

Shembull Multipleksheri **8/1**, me hyrjet I_0, I_1, ..., I_7, daljen **f** dhe sinjalet seleksionuese **X**, **Y** e **Z**, të cilat ndryshojnë sipas diagrameve kohore:



ku me **T** është shënuar perioda e zgjatjes së tyre.

$$
\begin{aligned}
f &= I_0\overline{X}\,\overline{Y}\,\overline{Z} + I_1\overline{X}\,\overline{Y}Z + I_2\overline{X}Y\overline{Z} + I_3\overline{X}YZ \\
&+ I_4X\overline{Y}\,\overline{Z} + I_5X\overline{Y}Z + I_6XY\overline{Z} + I_7XYZ \\
&= I_0m_0 + I_1m_1 + I_2m_2 + I_3m_3 \\
&+ I_4m_4 + I_5m_5 + I_6m_6 + I_7m_7
\end{aligned}
$$

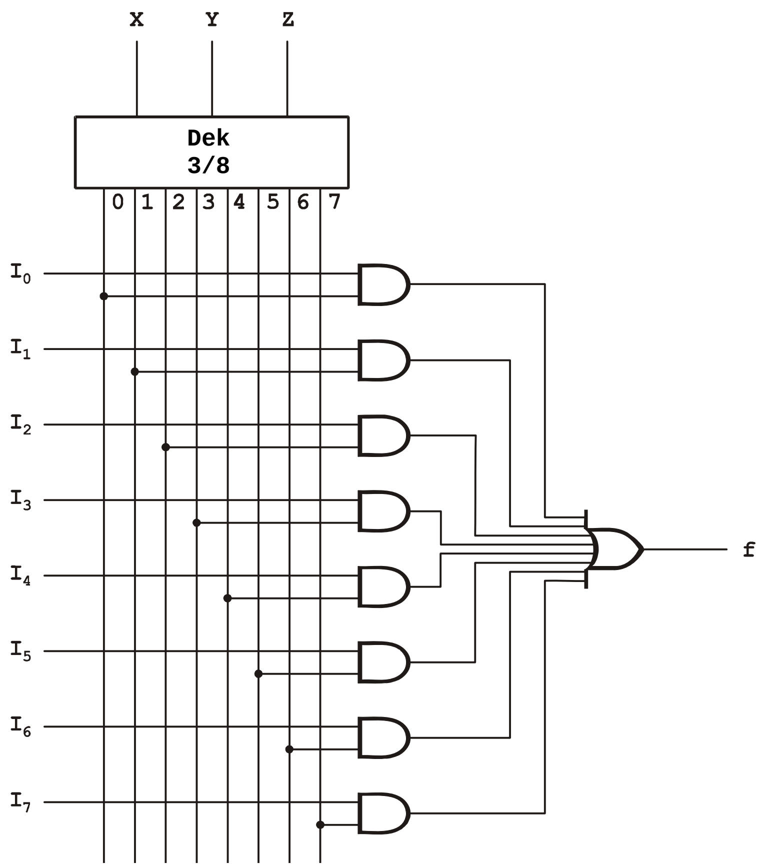


Fig.9.4 Multiplekseri 8/1

Këtu, mintermat m_0, m_1, ..., m_7 të sinjaleve seleksionuese **X**, **Y** dhe **Z** merren në daljet e një dekoduesi **3/8**, gjë që e thjeshton mjaft qarkun logjik. Për ta eliminuar numrin e madh të hyrjeve në elementin e fundit të qarkut, mbledhja mund të realizohet edhe duke shfrytëzuar dy ose më shumë elemente logjike njëkohësisht.

Multiplekseri **8/1**, i dhënë në Fig.9.4, mund të realizohet edhe duke shfrytëzuar dy multipleksere **4/1** dhe një multiplekser **2/1**, ashtu siç është treguar në Fig.9.5.

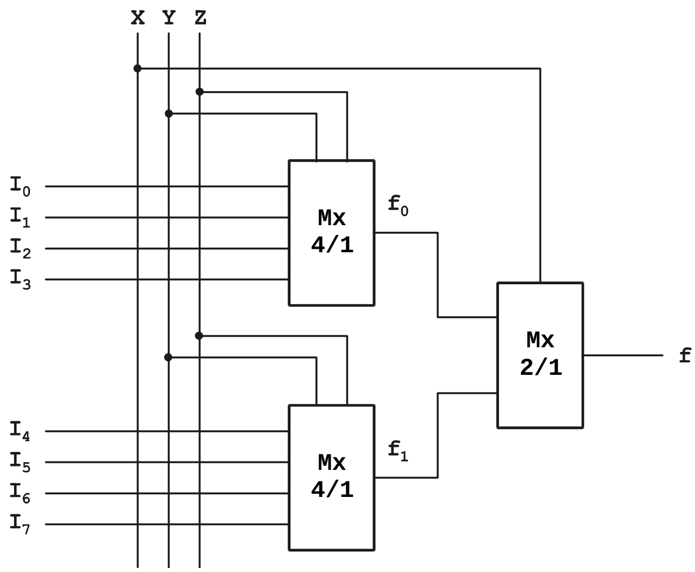


Fig.9.5 Multipleksersi 8/1 i realizuar me multiplekser 4/1 dhe 2/1

$$f_0 = I_0\overline{Y}\overline{Z} + I_1\overline{Y}Z + I_2Y\overline{Z} + I_3YZ$$

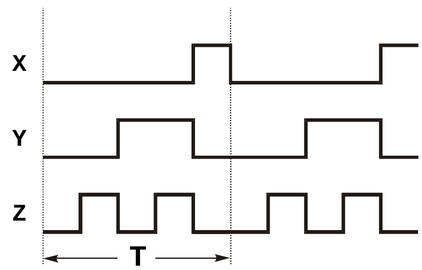
$$f_1 = I_4\overline{Y}\overline{Z} + I_5\overline{Y}Z + I_6Y\overline{Z} + I_7YZ$$

$$f = f_0\overline{X} + f_1X$$

Multiplekseri me numër të çfarëdoshëm hyrjesh

Numri i hyrjeve në multiplekser nuk është e thënë të jetë i barabartë me numrin maksimal të mundshëm të tyre, që lidhet me numrin e sinjaleve seleksionuese.

Shembull Multiplekseri **5/1**, me hyrjet I_0, I_1, ..., I_4, daljen **f** dhe sinjalet seleksionuese **X**, **Y** e **Z**, të cilat ndryshojnë me periodën **T**, sipas diagrameve kohore:



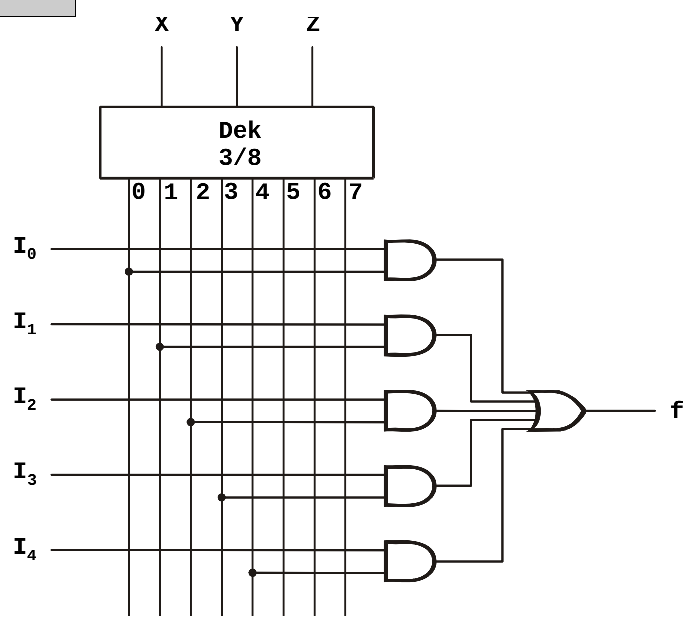


Fig.9.6 Multiplekseri 5/1

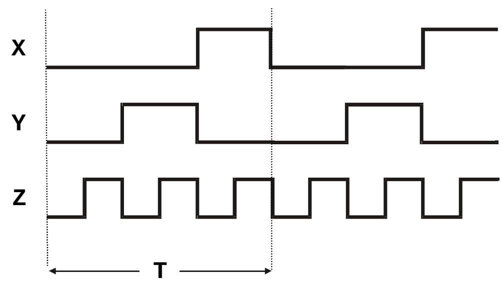
Shprehja e funksionit dalës nga multiplekseri është:

$$f = I_0 m_0 + I_1 m_1 + I_2 m_2 + I_3 m_3 + I_4 m_4$$

Kurse perioda e zgjatjes së sinjaleve seleksionuese, përkatësisht numri i kombinimeve të mundshme të vlerave të tyre, e përcakton numrin e hyrjeve në multiplekser, vlerat e të cilave përcillen në daljen e tij **f**. Tri daljet e fundit nga dekoduesi kanë ngelur të pashfrytëzuara, sepse në hyrje të multiplekserit vijnë vetëm **5** vargje informatash që duhet të paketohen.

Duke përshtatur periodën e sinjaleve seleksionuese që aplikohen në hyrjet përkatëse të multipleksetit, sipas nevojës mund të shfrytëzohen edhe më pak hyrje sesa që ka multiplekseri.

Shembull Sinjalet seleksionuese **X**, **Y** e **Z** dhe shprehja e funksionit dalës **f** nga multiplekseri **8/1**, ashtu që ai të shfrytëzohet si multiplekser **6/1**, përkatësisht vetëm në hyrjet I_0, I_1, ..., I_5, të aplikohen sinjale hyrëse.



Multiplekserët me më shumë hyrje

Për realizimin e multiplekserëve me më shumë hyrje mund të përdoren disa multiplekserë me numër të caktuar hyrjesh, siç janë, p.sh., multiplekserët **16/1**.

Shembull Multiplekseri **32/1**, me hyrjet I_0, I_1, ..., I_{31}, i realizuar duke shfrytëzuar dy multiplekserë **16/1** dhe një multiplekser **2/1**.

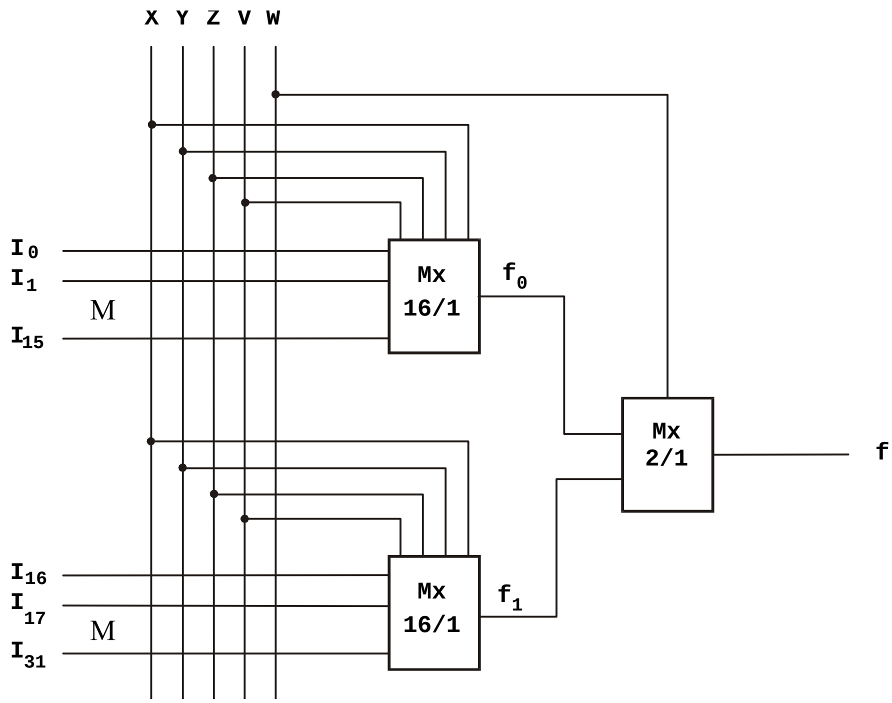


Fig.9.7 Multiplekseri 32/1

Funksionet në daljet e multiplekserëve të veçantë mund të shprehen kështu:

$$f_0 = \sum_{i=0}^{15} I_i \cdot m_i$$

$$f_1 = \sum_{i=16}^{31} I_i \cdot m_{i-16}$$

$$f = f_0\overline{W} + f_1 W = f_0 n_0 + f_1 n_1$$

ku mintermat m_0, m_1, ...,m_{31} fitohen përmes prodhimit të variablave dhe kovariablave të sinjaleve seleksionuese **X**, **Y**, **Z** dhe **V**, kurse mintermat n_0 dhe n_1 janë dy vlerat e mundshme të sinjalit seleksionues **W**. Që multiplekseri **32/1** të funksionojë ashtu që në daljen e tij **f** informatat hyrëse të paraqiten me radhën I_0, I_1, ..., I_{31}, diagramet kohore të sinjaleve seleksionuese duhet t'i marrin vlerat që shihen në Fig.9.8.

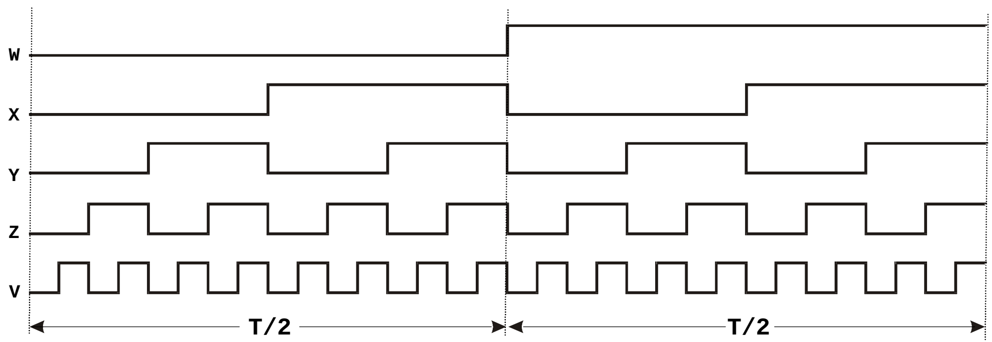


Fig.9.8 Sinjalet seleksionuese te multiplekseri 32/1

Në gjysmëperiodën e parë **T/2** të vlerave të sinjaleve seleksionuese do të jetë aktiv multiplekseri i parë dhe në daljen $\mathbf{f_0}$ të tij do të paraqiten me radhë vlerat e sinjaleve hyrëse $\mathbf{I_0}$, $\mathbf{I_1}$, ..., $\mathbf{I_{15}}$. Në gjysmëperiodën e dytë të vlerave të sinjaleve seleksionuese do të jetë aktiv multiplekseri i dytë, përkatësisht në daljen $\mathbf{f_1}$ të tij përcillen vlerat që aplikohen në hyrjet $\mathbf{I_{16}}$, $\mathbf{I_{17}}$, ..., $\mathbf{I_{31}}$ të multiplekserit. Njëkohësisht, gjatë gjysmëperiodës së parë vlerat e hyrjes $\mathbf{f_0}$ të multiplekserit **2/1** përcillen në daljen **f** të multiplekserit **32/1**. Kurse, në gjysmëperiodën e dytë, në daljen **f** të multiplekserit **32/1** përcillen vlerat e hyrjes $\mathbf{f_1}$ të multiplekserit **2/1**.

Multiplekseri shumëbitësh

Multiplekserët e dhënë më sipër njihen edhe si multiplekserë njëbitësh, sepse informatat në hyrjet e tyre paraqesin vargje informatash binare njëbitëshe. Por, në praktikë, informatat që përpunohen dhe transmetohen janë kryesisht shumëbitëshe. Për paketimin e tyre përdoren multiplekserë adekuat shumëbitësh, të cilët realizohen dhe funksionojnë plotësisht njëlloj si edhe multiplekserët njëbitësh.

Shembull Multiplekseri **2/1**, me dy hyrje të informatave elementare dybitëshe $\mathbf{A=a_1a_0}$ dhe $\mathbf{B=b_1b_0}$.

Këtu, daljet nga multiplekseri janë dybitëshe, ashtu siç janë edhe hyrjet në te, kurse sinjali seleksionues **S** është i njëjtë me sinjalin e përdorur te multiplekseri njëbitësh **2/1**.

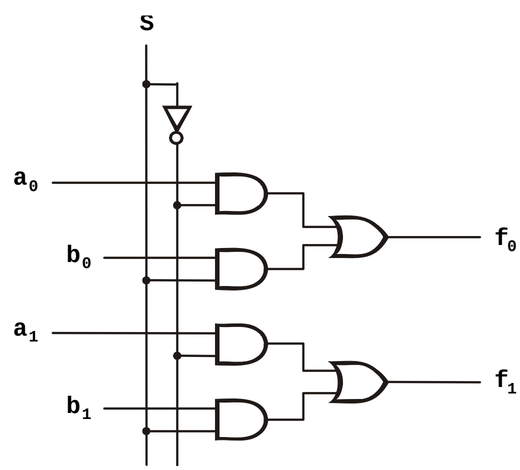


Fig.9.9 Multiplekseri 2/1 dybitësh

$$f_0 = a_0\overline{S} + b_0 S$$
$$f_1 = a_1\overline{S} + b_1 S$$

Për ta kuptuar funksionimin e multipleksserit të dhënë, le të marrim, p.sh., se në dy hyrjet e tij aplikohen vargjet e informatave binare:

a_1**=101...**
a_0**=110...**

b_1**=010...**
b_0**=100...**

Vargjet e informatave binare të cilat do të merren në dy daljet e multipleksserit (për çdo bit një dalje) janë:

f_0**=111000...**
f_1**=100110...**

Nga vargjet e fituara shihet se për **S=0**, në daljet f_1 e f_0 përcillen vlerat e hyrjeve a_1 dhe a_0, kurse për **S=1** vlerat e hyrjeve b_1 dhe b_0 janë vlera dalëse.

Multiplekseri dybitësh në fakt formohet prej dy multiplekserëve njëbitësh. Kështu, skema e multiplekserit të dhënë më sipër, e realizuar përmes dy multiplekserëve **2/1**, do të duket si në Fig.9.10.

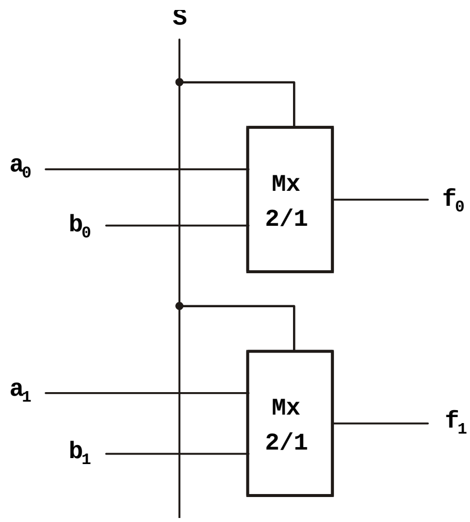


Fig.9.10 Multiplekseri dybitësh 2/1 i realizuar përmes dy multiplekserëve njëbitësh

Plotësisht njëlloj realizohen edhe multiplekserët me disa hyrje shumëbitëshe.

Shembull	Multiplekseri **4/1**, me **4**-hyrje **3**-bitëshe: $A=a_2a_1a_0$ $B=b_2b_1b_0$ $C=c_2c_1c_0$ $D=d_2d_1d_0$ tek i cili, për gjenerimin e mintermave të sinjaleve seleksionuese **X** dhe **Y**, shfrytëzohet dekoduesi **2/4**.

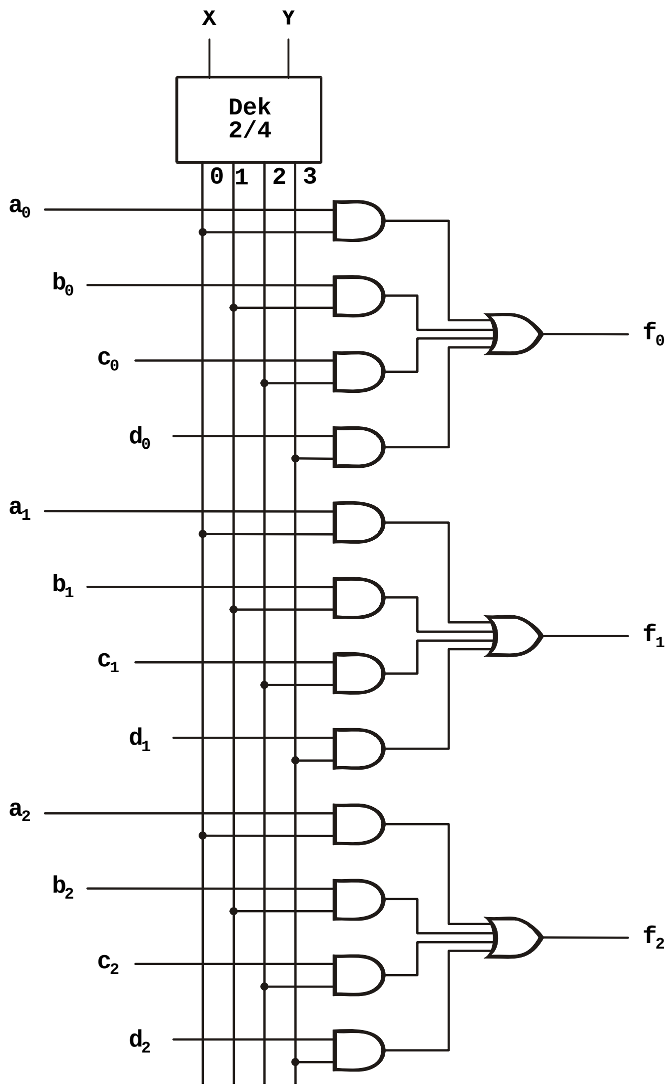


Fig.9.11 Multiplekseri trebitësh 4/1

$$f_0 = a_0\overline{X}\,\overline{Y} + b_0\overline{X}Y + c_0X\overline{Y} + d_0XY$$
$$f_1 = a_1\overline{X}\,\overline{Y} + b_1\overline{X}Y + c_1X\overline{Y} + d_1XY$$
$$f_2 = a_2\overline{X}\,\overline{Y} + b_2\overline{X}Y + c_2X\overline{Y} + d_2XY$$

Multiplekseri **m**-bitësh me **n**-hyrje mund të realizohet përmes **m**-multiplekserëve **n/1** njëbitësh.

Shembull Multiplekseri **2/1**, me **2**-hyrjet **4**-bitëshe:

$$A=a_3a_2a_1a_0$$
$$B=b_3b_2b_1b_0$$

i realizuar përmes **4** multiplekserëve **2/1**, **1**-bitësh, i cili e shfrytëzon sinjalin seleksionues **X**.

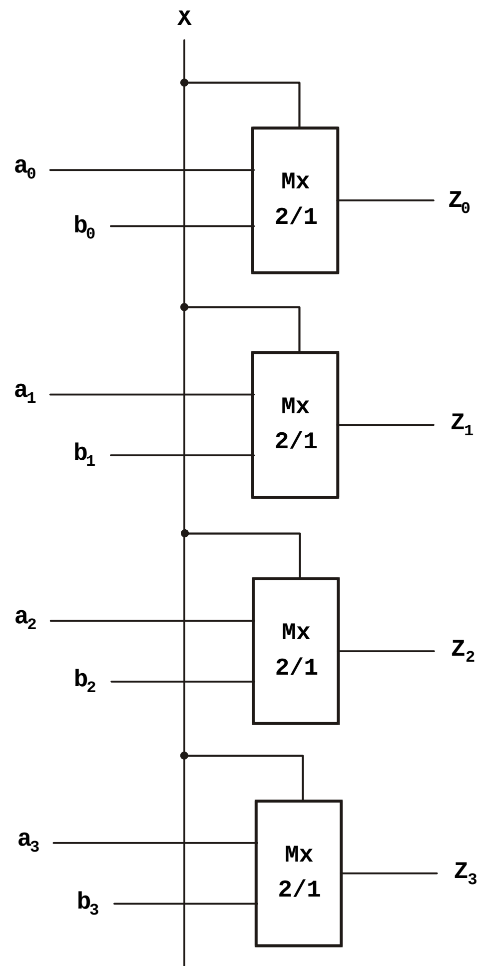


Fig.9.12 Multiplekseri katërbitësh 2/1

$$Z_0 = a_0\overline{X} + b_0X$$
$$Z_1 = a_1\overline{X} + b_1X$$
$$Z_2 = a_2\overline{X} + b_2X$$
$$Z_3 = a_3\overline{X} + b_3X$$

Sinteza e qarqeve përmes multiplekserëve

Multiplekseri me **n**-sinjale seleksionuese dhe me 2^n-hyrje mund të përdoret për realizimin e funksioneve me **n+1** variabla. Gjatë kësaj, së pari zgjidhen variablat që do të përdoren si sinjale seleksionuese, e pastaj caktohen vlerat e sinjaleve në hyrjet e multiplekserit, duke shfrytëzuar edhe variablat e pashfrytëzuara.

Shembull

Sinteza e qarkut logjik përmes së cilit realizohet funksioni:

$$f(A,B,C)=\overline{A}\,\overline{B}+A\,C$$

duke shfrytëzuar multiplekserin **4/1** dhe si sinjale seleksionuese përdoren vlerat e variablave **A** e **B**.

Shprehja e funksionit **f** në dalje të multiplekserit, nëse si sinjale seleksionuese përdoren variablat **A** e **B**, në formë të përgjithshme duket:

$$f=I_0\cdot\overline{A}\,\overline{B}+I_1\overline{A}\,B+I_2\cdot A\,\overline{B}+I_3\cdot A\,B$$

ku me I_0, I_1, I_2 dhe I_3 janë shënuar hyrjet në multiplekser. Që shprehja e funksionit të dhënë të shkruhet në këtë formë, mund të përdoret tabela përkatëse e kombinimeve:

A	B	C	f
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

prej nga, duke veçuar variablën e pavarur **C**, shprehja e kërkuar e funksionit është:

$$f=1\cdot\overline{A}\,\overline{B}+0\cdot\overline{A}\,B+C\cdot A\,\overline{B}+C\cdot A\,B$$

Në fund, nëse krahasohet kjo shprehje me shprehjen e përgjithshme të dhënë më sipër, për hyrjet e veçanta në multiplekser fitohen vlerat:

$$\mathbf{I_0} = \mathbf{1}$$
$$\mathbf{I_1} = \mathbf{0}$$
$$\mathbf{I_2} = \mathbf{C}$$
$$\mathbf{I_3} = \mathbf{C}$$

Kështu, qarku logjik i funksionit **f**, i realizuar përmes multiplekserit **4/1**, nëse vlerat e variablave **A** dhe **B** përdoren si sinjale seleksionuese, duket si në Fig.9.13.

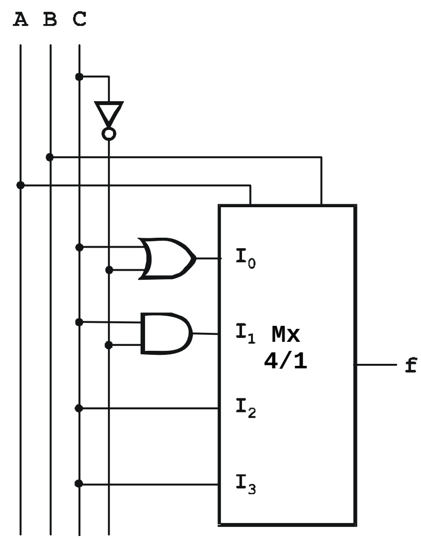


Fig.9.13 Qarku i realizuar duke e shfrytëzuar multiplekserin 4/1

Këtu, vlerat logjike **1** dhe **0**, të cilat aplikohen në hyrjet $\mathbf{I_0}$ dhe $\mathbf{I_1}$, janë gjeneruar duke shfrytëzuar elementet logjike **OSE** e **DHE**.

Duke pasur parasysh shprehjen e përgjithshme të funksionit dalës nga multiplekseri dhe pozitën në **K**-diagram të mintermave që paraqiten në këtë shprehje, çdo fushe në diagram mund t'i shoqërohet sinjali hyrës përkatës.

Shembull Sinteza e qarkut logjik të funksionit:

$$f = AC + \overline{A}\,\overline{B}$$

duke shfrytëzuar multipleksерin **4/1**, nëse si sinjale seleksionuese përdoren vlerat e variablave:

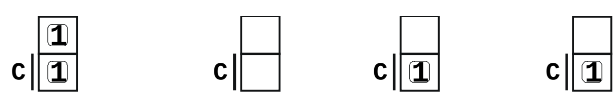
a. **A** dhe **B**
b. **A** dhe **C**
c. **B** dhe **C**

Për funksionin e dhënë, së pari mbushet **K**-diagrami përkatës. Pastaj, në bazë të variablave të cilat janë zgjedhur të përdoren si sinjale seleksionuese, fushave brenda diagramit u shoqërohen sinjalet hyrëse përkatëse. Në fund, duke copëtuar **K**-diagramin në diagrame parciale, të formuara në bazë të shoqërimit të fushave të cilat u përkasin hyrjeve të veçanta në multiplekser, caktohen funksionet hyrësa në multiplekser përmes variablës së pavarur.

C \ AB	00	01	11	10
0	1			
1	1		1	1

a.

AB	00	01	11	10
	I_0	I_1	I_3	I_2
	I_0	I_1	I_3	I_2



$I_o = C + \overline{C} = 1$ $\quad I_1 = 0$ $\quad I_2 = C$ $\quad I_3 = C$

Vlerat e sinjaleve hyrës në multiplekser janë plotësisht të njëjta me ato që u fituan në shembullin paraprak, ashtu që edhe qarku logjik përkatës do të jetë i njëjtë.

b.

C \ A	0	0	1	1
0	I_0	I_0	I_2	I_2
1	I_1	I_1	I_3	I_3

	B
1	

$I_0 = \overline{B}$

	B
1	

$I_1 = \overline{B}$

	B

$I_2 = 0$

	B
1	1

$I_3 = B + \overline{B} = 1$

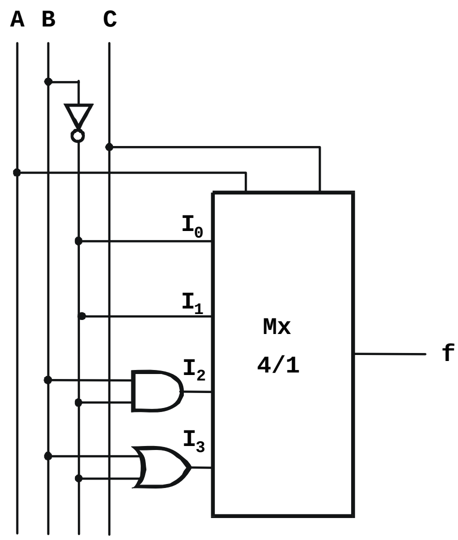


Fig.9.14 Qarku i realizuar me multiplekser **4/1**, *nëse si sinjale seleksionuese përdoren vlerat e variablave* **A** *dhe* **C**

c.

C \ B	0	1	1	0
0	I_0	I_2	I_2	I_0
1	I_1	I_3	I_3	I_1

	A	A	
1			

$I_0 = \overline{A}$

	A	A	
1			1

$I_1 = A + \overline{A} = 1$

	A

$I_2 = 0$

	A
	1

$I_3 = A$

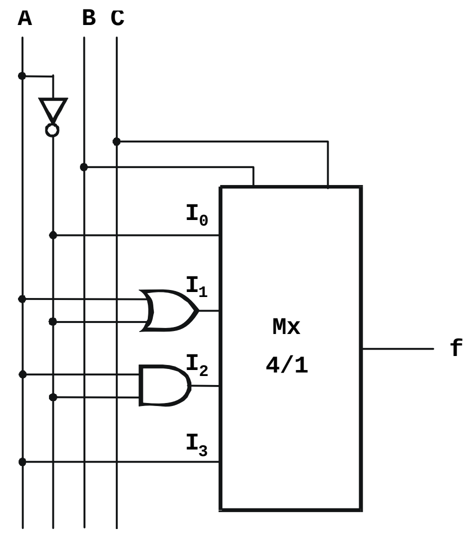


***Fig.9.15 Qarku i realizuar me multiplekser* `4/1`*, nëse si sinjale seleksionuese përdoren vlerat e variablave* `B` *dhe* `C`**

Si variabla të pavarura nuk është e thënë të ngelë vetëm një variabël.

Shembull

Sinteza e qarkut logjik të funksionit:

$$f(A,B,C,D) = \sum m^1(0,3,4,5,9,10,12,13)$$

përmes:

a. multipleksërit **8/1**, nëse si sinjale seleksionuese merren variablat **A**, **B** dhe **C**,
b. multipleksërit **8/1**, duke lidhur hyrjet e sinjaleve seleksionuese me variablat **A**, **C** dhe **D**,
c. multipleksërit **4/1**, nëse si sinjale seleksionuese merren variablat **A** dhe **B**.

A

1	1	1	
	1	1	1
1			
			1

D
C
B

a.

C \ AB	00	01	11	10
0	I_0	I_2	I_6	I_4
0	I_0	I_2	I_6	I_4
1	I_1	I_3	I_7	I_5
1	I_1	I_3	I_7	I_5

$I_0 = \overline{D}$ $\quad I_1 = D$ $\quad I_2 = D + \overline{D} = 1$ $\quad I_3 = 0$

$I_4 = D$ $\quad I_5 = \overline{D}$ $\quad I_6 = D + \overline{D} = 1$ $\quad I_7 = 0$

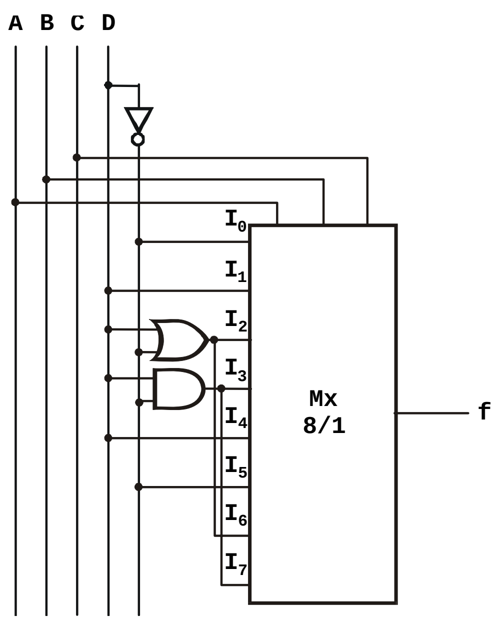


*Fig.9.16 Qarku i realizuar me multiplekser **8/1**, nëse si sinjale seleksionuese përdoren vlerat e variablave **A**, **B** dhe **C***

b.

CD \ A	0	0	1	1
00	I_0	I_0	I_4	I_4
01	I_1	I_1	I_5	I_5
11	I_3	I_3	I_7	I_7
10	I_2	I_2	I_6	I_6

$I_0 = B + \overline{B} = 1$ $I_1 = B$ $I_2 = 0$ $I_3 = \overline{B}$

$I_4 = B$ $I_5 = B + \overline{B} = 1$ $I_6 = \overline{B}$ $I_7 = 0$

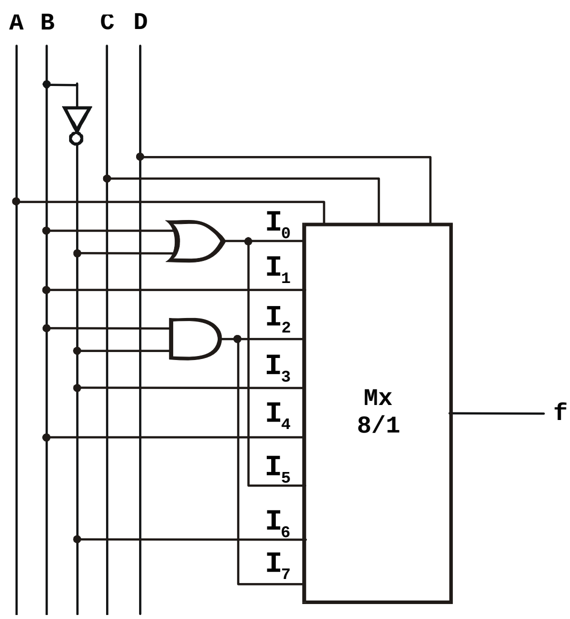
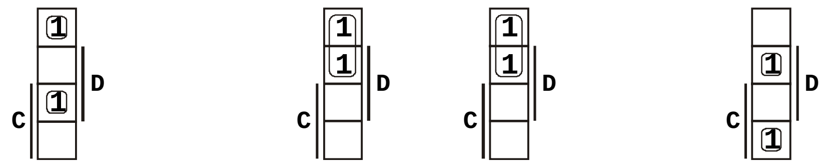


Fig.9.17 Qarku i realizuar me multiplekser **8/1**, *nëse si sinjale seleksionuese përdoren vlerat e variablave* **A, C** *dhe* **D**

c.

AB: 00	01	11	10
I_0	I_1	I_3	I_2
I_0	I_1	I_3	I_2
I_0	I_1	I_3	I_2
I_0	I_1	I_3	I_2



$I_0 = \bar{C}\bar{D} + CD = C \otimes D$ $\quad I_1 = \bar{C}$ $\quad I_3 = \bar{C}$ $\quad I_2 = \bar{C}D + C\bar{D} = C \oplus D$

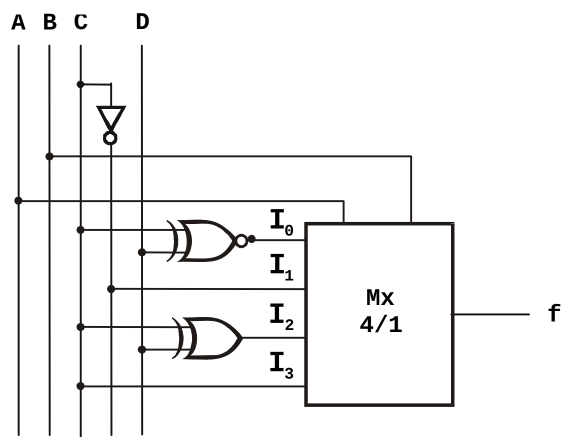


Fig.9.18 Qarku i realizuar me multiplekser **4/1**, *nëse si sinjale seleksionuese përdoren vlerat e variablave* **A** *dhe* **B**

Qarqe të ndryshme të realizuara me multiplekser

Gjatë realizimit të qarqeve logjike të ndryshme, mund të përdoren edhe multiplekserët. Nëse qarqet kanë më shumë dalje, me multiplekser mund të realizohet një dalje, ose edhe disa dalje të tyre njëkohësisht.

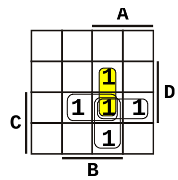
Shembull

Qarku logjik kombinues, me katër hyrje (**A**, **B**, **C** dhe **D**) e tri dalje (**X**, **Y** dhe **Z**), në dalje të së cilit fitohen vlerat vijuese:

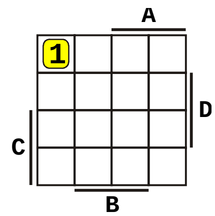
X=1,Y=0,Z=0 nëse në **3** ose **4** hyrje aplikohen vlerat **1**
X=0,Y=1,Z=0 nëse në të **4** hyrjet aplikohet vlera **0**
X=0,Y=0,Z=1 nëse në **1** ose **2** hyrje aplikohet vlera **1**

Funksionet e daljeve **X** e **Y** të realizohen me elemente logjike themelore, kurse për realizimin e funksionit të daljes **Z** të përdoret multiplekseri **8/1**, ashtu që si sinjale seleksionuese të merren variablat **A**, **B** dhe **C**.

Hyrjet				Daljet		
A	B	C	D	X	Y	Z
0	0	0	0	0	1	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	0	0	1
0	1	0	1	0	0	1
0	1	1	0	0	0	1
0	1	1	1	1	0	0
1	0	0	0	0	0	1
1	0	0	1	0	0	1
1	0	1	0	0	0	1
1	0	1	1	1	0	0
1	1	0	0	0	0	1
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	1	0	0



$$X = ABC + ABD + ACD + BCD$$

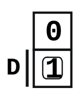


$$y = \overline{A}\,\overline{B}\,\overline{C}\,\overline{D}$$

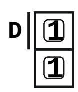
Funksioni ***Z****:*

0	1	1	1
1	1	0	1
1	0	0	0
1	1	0	1

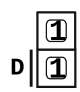
C \ AB	00	01	11	10
0	I_0	I_2	I_6	I_4
0	I_0	I_2	I_6	I_4
1	I_1	I_3	I_7	I_5
1	I_1	I_3	I_7	I_5



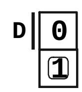
$I_0 = D$



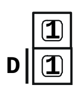
$I_1 = D + \overline{D} = 1$



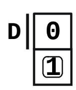
$I_2 = D + \overline{D} = 1$



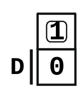
$I_3 = \overline{D}$



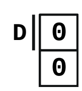
$I_4 = D + \overline{D} = 1$



$I_5 = \overline{D}$



$I_6 = \overline{D}$



$I_7 = 0$

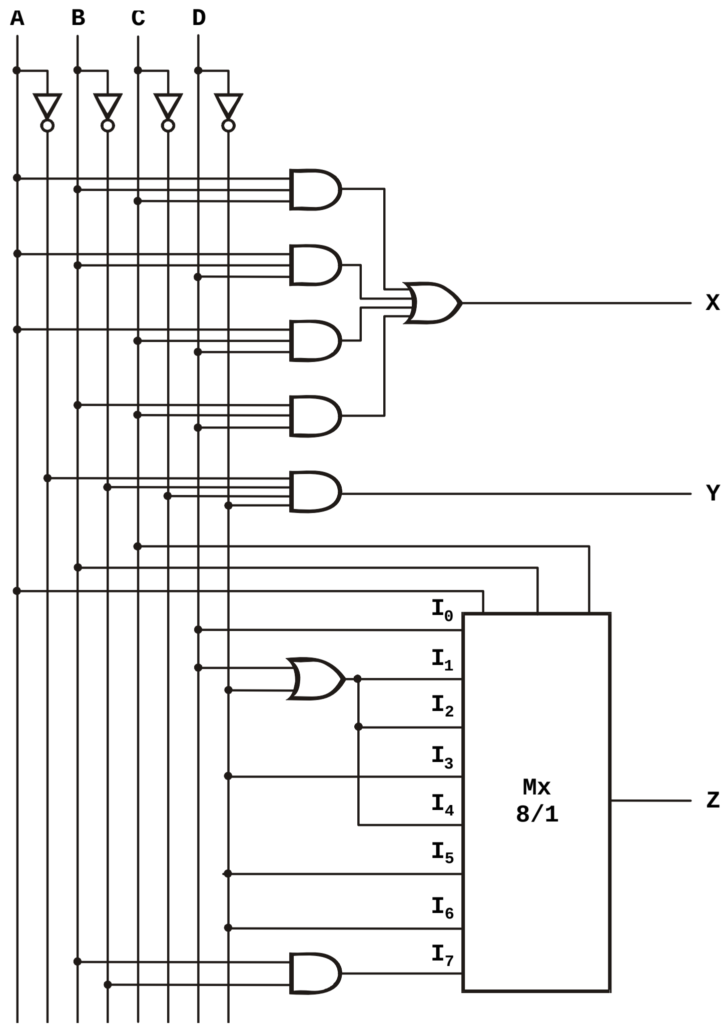
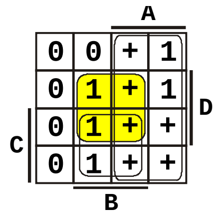


***Fig.9.19 Qarku me një dalje të realizuar përmes multiplekserit* 8/1**

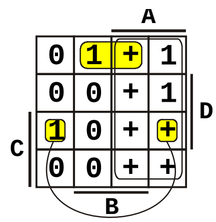
Për realizimin e qarqeve logjike me më shumë dalje, mundet që për çdo dalje të përdoret një multiplekser.

Shembull	Konvertuesi i fjalëve kodike $(ABCD)_2$ të kodit **NBCD**, në fjalë kodike $(XYZV)_2$ të kodit **BCD 5211**.

8421				5211			
A	B	C	D	X	Y	Z	V
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	1
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	1
1	0	0	0	1	1	0	1
1	0	0	1	1	1	1	1
1	0	1	0	+	+	+	+
1	0	1	1	+	+	+	+
1	1	0	0	+	+	+	+
1	1	0	1	+	+	+	+
1	1	1	0	+	+	+	+
1	1	1	1	+	+	+	+

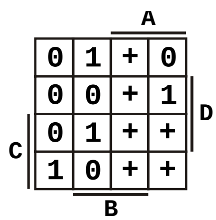


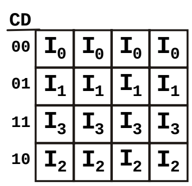
$$X = A + BC + BD$$

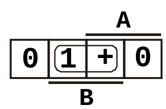


$$Y = A + B\overline{C}\,\overline{D} + \overline{B}CD$$

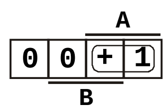
Funksioni ***Z***:



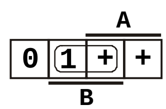




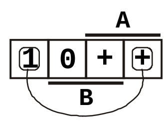
$I_0 = B$



$I_1 = A$



$I_3 = B$

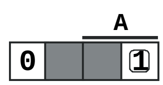


$I_2 = \overline{B}$

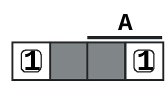
Funksioni **V**:

0	1	+	1
1	0	+	1
1	1	+	+
1	1	+	+

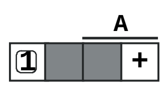
CD \ B	0	1	1	0
00	I_0	I_4	I_4	I_0
01	I_1	I_5	I_5	I_1
11	I_3	I_7	I_7	I_3
10	I_2	I_6	I_6	I_2



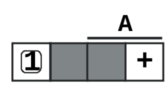
$I_0 = A$



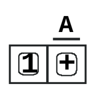
$I_1 = A + \overline{A} = 1$



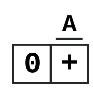
$I_2 = \overline{A}$



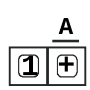
$I_3 = \overline{A}$



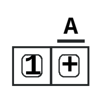
$I_4 = A + \overline{A} = 1$



$I_5 = 0$



$I_6 = A + \overline{A} = 1$



$I_7 = A + \overline{A} = 1$

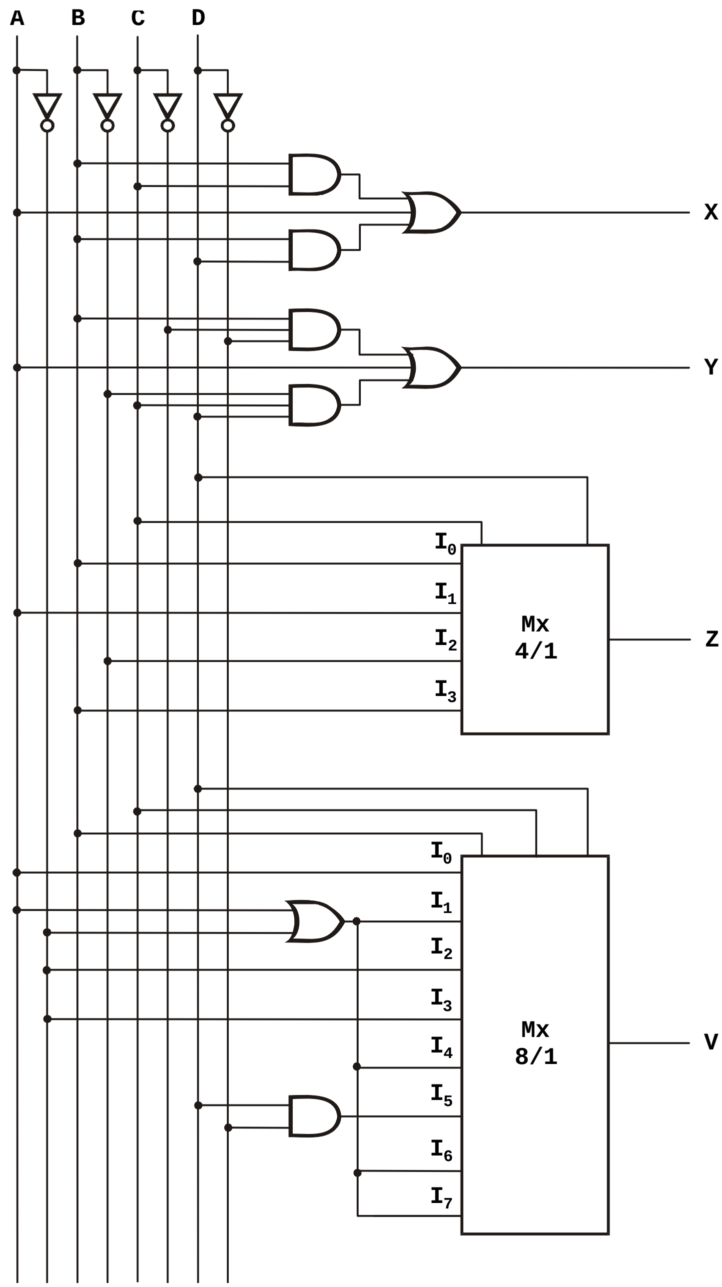
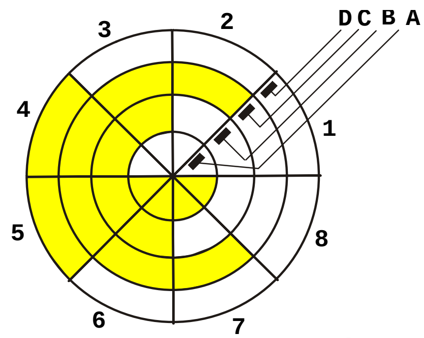


Fig.9.20 Qarku me dy dalje të realizuara përmes multiplekserëve

Gjatë sintezës së qarqeve logjike, për realizimin e funksioneve me të cilët përshkruhen daljet e veçanta të tyre, bashkë me multiplekserët mund të përdoren edhe dekoduesit.

Shembull

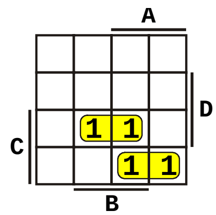
Qarku logjik kombinues përmes të cilit detektohet pozita e brushave **A**, **B**, **C** dhe **D**, të vendosura mbi sipërfaqen e diskut lëvizës, i cili është ndarë në **8** sektor. Sipërfaqet që në figurën vijuese janë të mbushura, paraqesin sipërfaqe përçuese, përkatësisht kur brushat gjenden mbi këto sipërfaqe, në dalje të tyre merret sinjali i vlerës logjike 1.



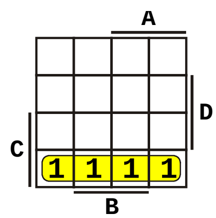
Gjatë punës së qarkut, për vlerat hyrëse $\mathbf{(ABCD)_2}$ të cilat merren përmes brushave, në dalje të tij gjenerohen ekuivalentët binarë $\mathbf{(XYZV)_2}$ të numrave rendorë të sektorëve ku janë pozicionuar brushat. Për kombinimet e vlerave hyrëse të cilat nuk shfrytëzohen, është paraparë që në dalje të qarkut të fitohet ekuivalenti binar **1001** i numrit decimal **9**.

N	Sektori	A	B	C	D	x	y	z	v
0	1	0	0	0	0	0	0	0	1
1	-	0	0	0	1	1	0	0	1
2	2	0	0	1	0	0	0	1	0
3	-	0	0	1	1	1	0	0	1
4	-	0	1	0	0	1	0	0	1
5	-	0	1	0	1	1	0	0	1
6	3	0	1	1	0	0	0	1	1
7	4	0	1	1	1	0	1	0	0
8	8	1	0	0	0	1	0	0	0
9	-	1	0	0	1	1	0	0	1
10	7	1	0	1	0	0	1	1	1
11	-	1	0	1	1	1	0	0	1
12	-	1	1	0	0	1	0	0	1
13	-	1	1	0	1	1	0	0	1

14	6	1	1	1	0	0	1	1	0
15	5	1	1	1	1	0	1	0	1



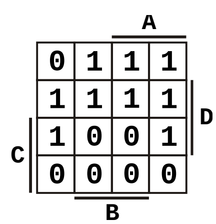
$$y = BCD + AC\overline{D}$$

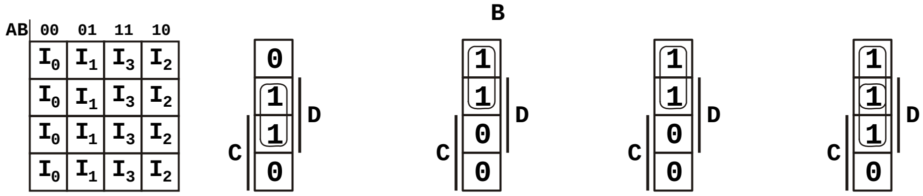


$$z = C\overline{D}$$

$$v = \sum m^1(0-1, 3-6, 9-13, 15)$$

Funksioni **x**:





$I_o = D$ $\quad I_1 = \overline{C}$ $\quad I_2 = \overline{C}$ $\quad I_3 = \overline{C} + D$

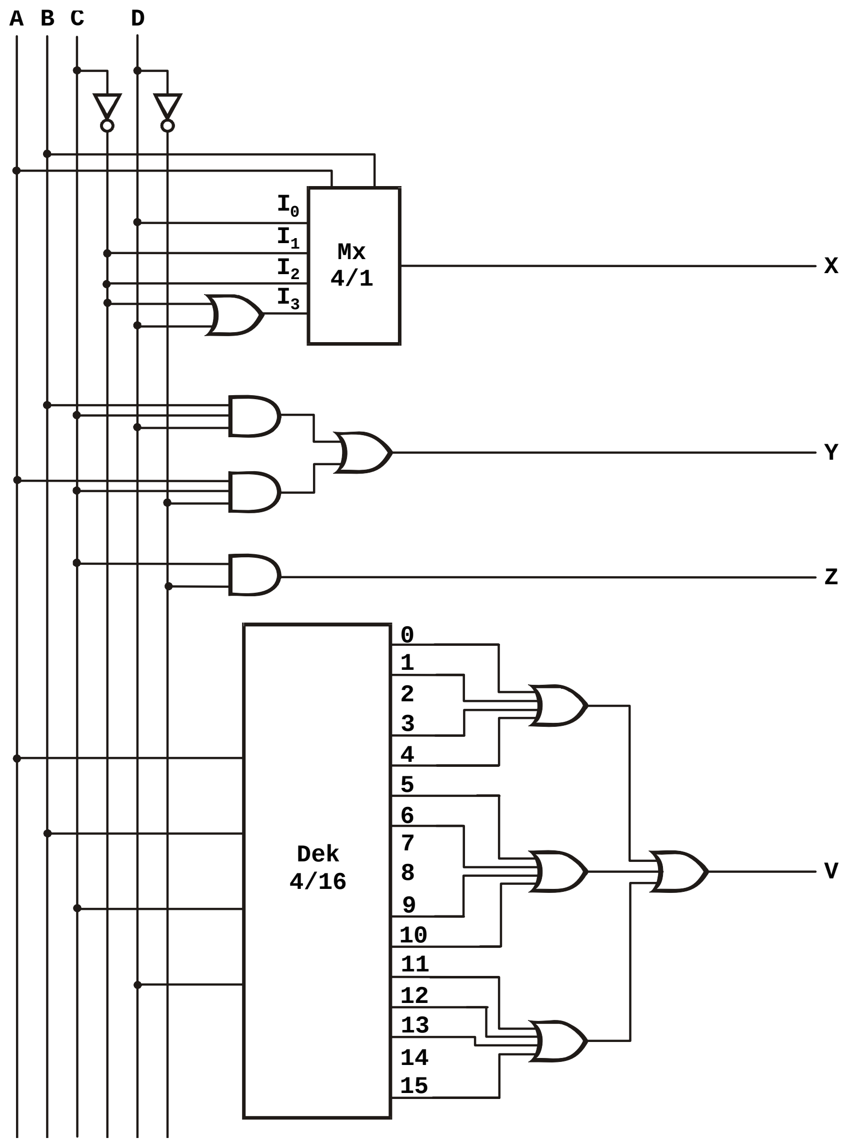


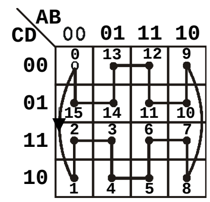
Fig.9.21 Realizimi i qarkut përmes multiplekserit dhe dekoduesit

Gjatë realizimit të qarkut logjik të dhënë më sipër, në dalje të dekoduesit, në vend të **1** elementi janë shfrytëzuar **4** elemente logjike për mbledhje, me qëllim që të mos tejngarkohet hyrja e vetëm një elementi.

Mund të realizohen edhe qarqe logjike me më shumë dalje, duke i realizuar të gjithë funksionet dalëse përmes multiplekserëve të veçantë.

Shembull

Konvertuesi i fjalëve kodike $(ABCD)_2$ të kodit ciklik të dhënë përmes tabelës vijuese:



në numra binarë $(XYZV)_2$, të cilët janë ekuivalentë me numrat decimalë të dhënë në tabelën e mësipërme. Si sinjale seleksionuese janë zgjedhur variablat **A**, **B** dhe **D**.

N	A	B	C	D	X	Y	Z	V
0	0	0	0	0	0	0	0	0
15	0	0	0	1	1	1	1	1
1	0	0	1	0	0	0	0	1
2	0	0	1	1	0	0	1	0
13	0	1	0	0	1	1	0	1
14	0	1	0	1	1	1	1	0
4	0	1	1	0	0	1	0	0
3	0	1	1	1	0	0	1	1
9	1	0	0	0	1	0	0	1
10	1	0	0	1	1	0	1	0
8	1	0	1	0	1	0	0	0
7	1	0	1	1	0	1	1	1
12	1	1	0	0	1	1	0	0
11	1	1	0	1	1	0	1	1
5	1	1	1	0	0	1	0	1
6	1	1	1	1	0	1	1	0

D \ AB	00	01	11	10
0	I_0	I_2	I_6	I_4
1	I_1	I_3	I_7	I_5
1	I_1	I_3	I_7	I_5
0	I_0	I_2	I_6	I_4

	0	1	1	1
	1	1	1	1
C	0	0	0	0
C	0	0	0	1

X

	0	1	1	0
	1	1	0	0
C	0	0	1	1
C	0	1	1	0

Y

	0	0	0	0
	1	1	1	1
C	1	1	1	1
C	0	0	0	0

Z

	0	1	0	1
	1	0	1	0
C	0	1	0	1
C	1	0	1	0

V

	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7
X	0	$\bar{C}$	$\bar{C}$	$\bar{C}$	1	$\bar{C}$	$\bar{C}$	$\bar{C}$
Y	0	$\bar{C}$	1	$\bar{C}$	0	C	1	C
Z	0	1	0	1	0	1	0	1
V	C	$\bar{C}$	$\bar{C}$	C	$\bar{C}$	C	C	$\bar{C}$

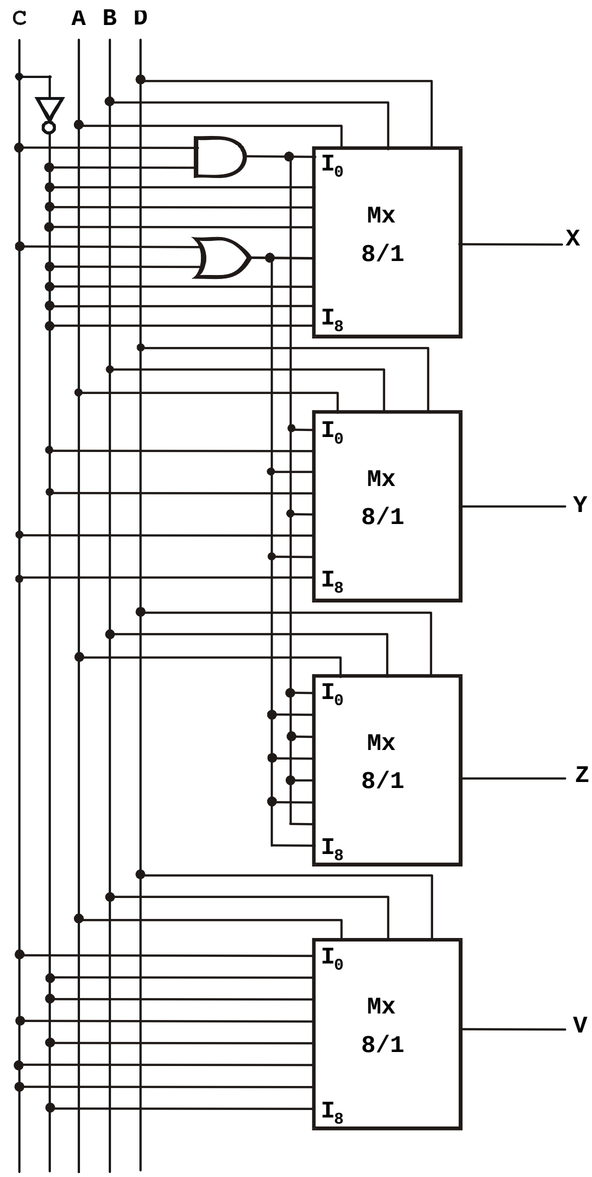


Fig.9.22 Konvertuesi i realizuar me **4** *multiplekser* **8/1**

Demultiplekserët

10

Demultiplekseri (ang. demultiplexer) është qark logjik përmes të cilit kryhet veprim i kundërt me multiplekserin. Përmes demultiplekserit, duke shfrytëzuar **n**-sinjale seleksionuese, vargu i informatave hyrëse **h** copëtohet në sinjale në $\mathbf{2^n}$-daljet e tij. Simbolikisht, demultiplekseri mund të paraqitet kështu:

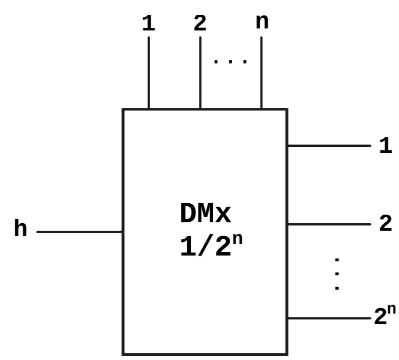


Fig.10.1 Paraqitja skematike e demultiplekserit

Demultiplekseri bashkë me multiplekserin përdoren te sistemet transmetuese, për ta mundësuar përcjelljen në distancë të informatave që burojnë nga më shumë shfrytëzues. Kështu, te dhënësi vargjet e informatave të $\mathbf{2^n}$-shfrytëzuesve të ndryshëm përmes multiplekserit $\mathbf{2^n/1}$ paketohen në një varg të vetëm **d**, i cili si i tillë transmetohet përmes një kanali transmetues. Pastaj, te marrësi përmes demultiplekserit $\mathbf{1/2^n}$ vargu i informatave i cili pranohet **h** (është vargu i njëjtë **d**) zbërthehet në $\mathbf{2^n}$-vargje elementare, të cilat u përcillen shfrytëzuesve të veçantë.

Demultiplekseri 1/2

Përmes demultiplekserit **1/2** vargu i sinjaleve të paketuara para transmetimit të tij përmes një linje transmetuese zbërthehet në dy vargje të pavarura, në bazë të ligjshmërisë së njëjtë, e cila është përdorur gjatë paketimit. Sinjali seleksionues **S** i cili përdoret për zbërthimin e vargut të sinjaleve hyrëse duket:

Shembull Demultiplekseri **1/2**, përmes të cilit zbërthehet sinjali hyrës **h**, në dy sinjale dalës **A** dhe **B**, duke shfrytëzuar sinjalin seleksionues **S** të dhënë më sipër.

h	S	A	B
0	0	0	0
0	1	0	0
1	0	1	0
1	1	0	1

$$A = h\bar{S}$$
$$B = hS$$

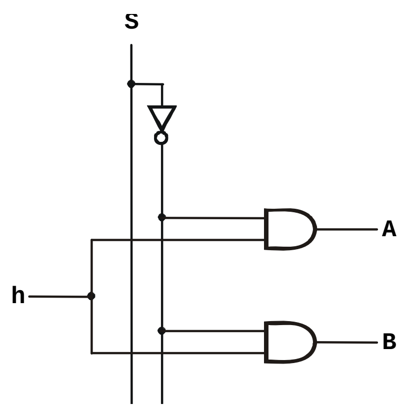


Fig.10.2 Demultiplekseri 1/2

Për ta pasur më të qartë punën e demultiplekserit **1/2** të dhënë më sipër, le të marrim se në hyrje të tij aplikohet vargu i sinjaleve binare:

10110110...

Në dy daljet e demultiplekserit **A** dhe **B**, duke pasur parasysh sinjalin seleksionues **S**, fitohen vargjet e sinjaleve binare:

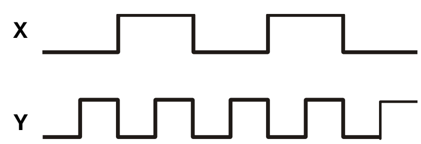
A=1101...
B=0110...

Demultiplekserët me më shumë dalje

Vargu i sinjaleve hyrëse në multiplekser mund të copëtohet edhe në më shumë sinjale dalëse, gjë që përcaktohet përmes sinjaleve seleksionuese.

Shembull

Demultiplekseri **1/4**, përmes së cilit zbërthehet sinjali hyrës **h**, në katër sinjale dalës I_0, I_1, I_2 dhe I_3, duke shfrytëzuar sinjalet seleksionuese vijuese:



$$I_0 = h\overline{X}\,\overline{Y}$$
$$I_1 = h\overline{X}\,Y$$
$$I_2 = h\,X\,\overline{Y}$$
$$I_3 = h\,X\,Y$$

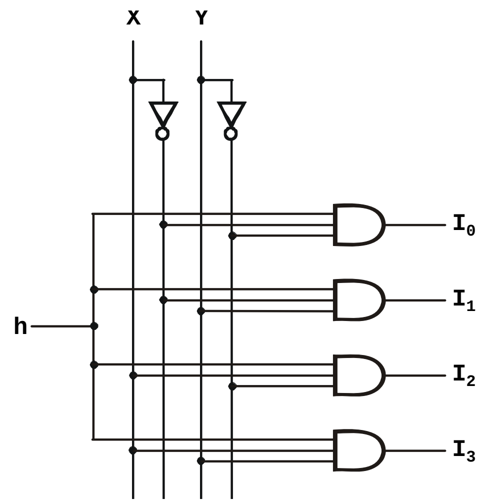


Fig.10.3 Demultiplekseri 1/4

Nëse, p.sh., në hyrje të demultiplekserit **4/1** aplikohet vargu i informatave binare:

h=1011001111000101...

vargjet e informatave dalëse nga demultiplekseri do të jenë:

$\mathbf{I_0=1010...}$
$\mathbf{I_1=0011...}$
$\mathbf{I_2=1100...}$
$\mathbf{I_3=1101...}$

Prej këtu shihet se në daljen $\mathbf{I_0}$ janë përcjellë shifrat binare të vargut **h**, të cilat gjenden në pozitat **1**, **5**, **9**, **13** dhe kështu me radhë. Ngjashëm përcillen edhe sinjalet në daljet e tjera të demultiplekserit, por ashtu që, p.sh., në daljen $\mathbf{I_1}$ fillohet me shifrën në pozitën e **2**-të, për të vazhduar me shifrat në pozitat **6**, **10**, **14** etj.

Për gjenerimin e mintermave të sinjaleve seleksionuese mund të shfrytëzohet edhe dekoduesi me numër të caktuar daljesh.

Shembull	Demultiplekseri **1/8**, përmes të cilit zbërthehet sinjali hyrës **h**, në tetë sinjale dalëse. Për gjenerimin e mintermave të sinjaleve seleksionuese **X**, **Y** dhe **Z** shfrytëzohet dekoduesi **3/8**.

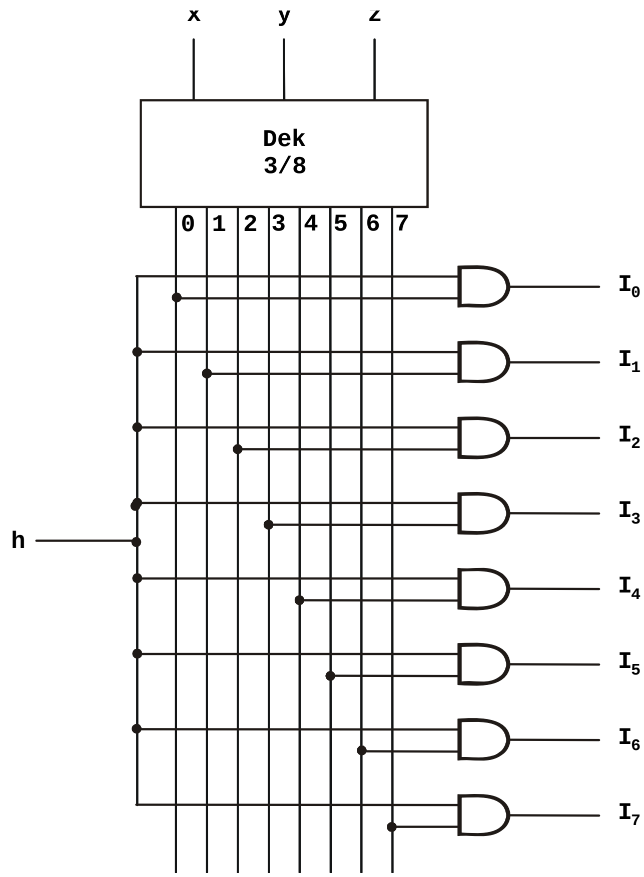
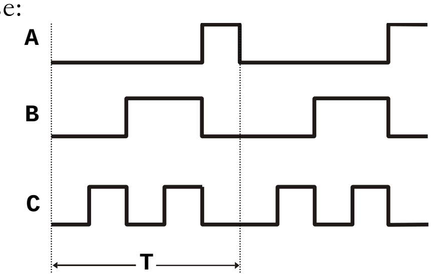


Fig.10.4 Demultiplekseri 1/8

Numri i daljeve nga demultiplekseri, ngjashëm si edhe numri i hyrjeve në multiplekser, nuk është e domosdoshme të jetë i barabartë me numrin e kombinimeve të cilat mund të fitohen me sinjalet seleksionuese.

Shembull

Demultiplekseri **1/5**, përmes të cilit zbërthehet sinjali hyrës **h**, në pesë sinjale dalëse, duke shfrytëzuar sinjalet seleksionuese:



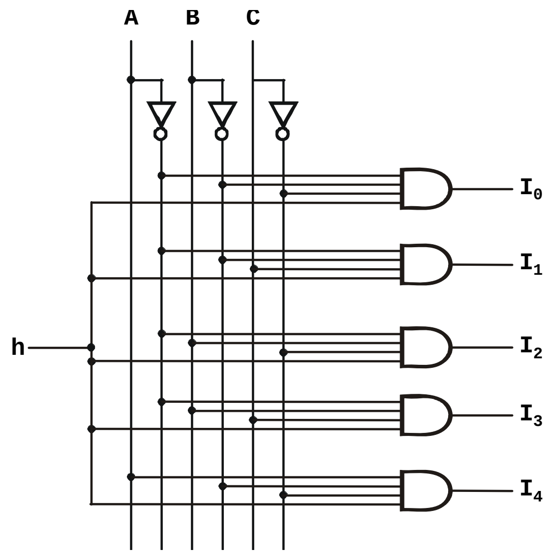


Fig.10.5 Demultiplekseri 1/5

Nëse në hyrjen **h** të demultipleksrit të dhënë më sipër vjen vargu i sinjaleve binare:

h=101110010111101...

vargjet e sinjaleve dalëse do të jenë:

I_0**=101...**
I_1**=001...**
I_2**=111...**
I_3**=100...**
I_4**=111...**

gjë që përcaktohet nga sinjalet seleksionuese.

Demultiplekserët shumëbitësh

Si multiplekserët ashtu edhe demultiplekserët mund të jenë **2** ose më shumë bitësh. Principi i realizimit të tyre është i ngjashëm me realizimin e multiplekserëve shumëbitësh.

Shembull	Demultiplekseri **1/2**, përmes të cilit zbërthehen informatat **3**-bitëshe $h=f_2f_1f_0$, në dy grupe informatash elementare **3**-bitëshe: $$A=a_2a_1a_0$$ $$B=b_2b_1b_0$$

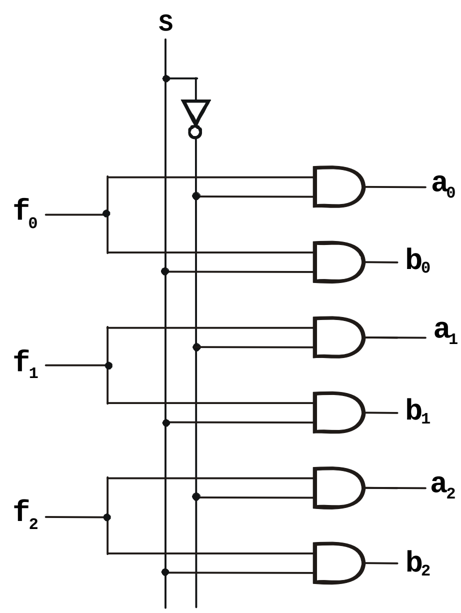


Fig.10.6 Demultiplekseri trebitësh 1/2

Nga qarku logjik i dhënë shihet se për realizimin e multiplekserit në fjalë janë shfrytëzuar **3** demultiplekserë **1**-bitësh, gjegjësisht për çdo bit një demultiplekser. Për ta pasur më të qartë mënyrën e funksionimit të demultiplekserit, le të marrim, p.sh., se në hyrjet e tij aplikohen vargjet e sinjaleve:

$$f_0=1101...$$
$$f_1=0101...$$
$$f_2=1011...$$

Në dalje të demultiplekserit fitohen vargjet e sinjaleve:

a_0=10.. b_0=11...
a_1=00... b_1=11...
a_2=11... b_2=01...

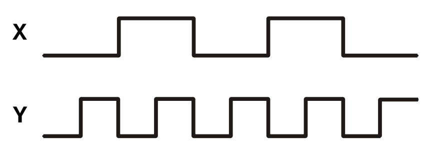
Sinjali seleksionues **S** te demultiplekseri **3**-bitësh **1/2**, i cili është dhënë më sipër, ndryshon plotësisht njëlloj si edhe sinjali seleksionues te demultiplekseri i zakonshëm. Në rastin e përgjithshëm, sinjalet seleksionuese te demultiplekserët shumëbitësh janë plotësisht të njëjtë me sinjalet seleksionuese, të cilat shfrytëzohen te demultiplekserët **1**-bitësh.

Shembull

Demultiplekseri **1/4**, përmes të cilit zbërthehet vargu i informatave **2**-bitëshe hyrëse $H=h_1h_0$ në vargje të informatave dalëse:

$$A = a_1a_0$$
$$B = b_1b_0$$
$$C = c_1c_0$$
$$D = d_1d_0$$

nëse zbërthimi bëhet përmes sinjaleve seleksionuese:



$$a_0 = h_0\,\overline{X}\,\overline{Y} \qquad a_1 = h_1\,\overline{X}\,\overline{Y}$$
$$b_0 = h_0\,\overline{X}\,Y \qquad b_1 = h_1\,\overline{X}\,Y$$
$$c_0 = h_0\,X\,\overline{Y} \qquad c_1 = h_1\,X\,\overline{Y}$$
$$d_0 = h_0\,X\,Y \qquad d_1 = h_1\,X\,Y$$

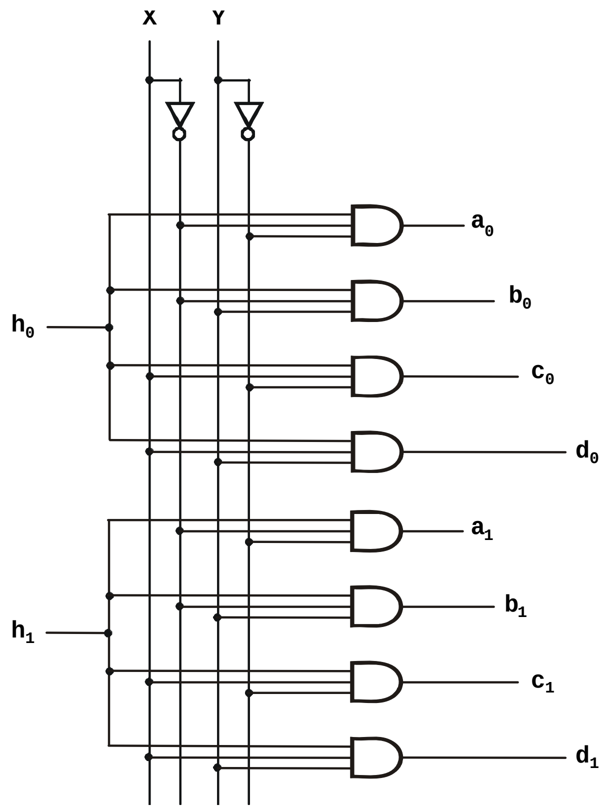


Fig.10.7 Demultiplekseri dybitësh 1/4

Demultiplekseri si dekodues

Demultiplekseri do të punojë si dekodues, nëse në hyrje të tij pandërprerë aplikohet sinjali binar me vlera logjike **1**.

Shembull	Dekoduesi i numrave binarë **(XYZ)**$_2$, në shifra të sistemit oktal të numrave, i realizuar duke shfrytëzuar demultiplekserin **1/8**.

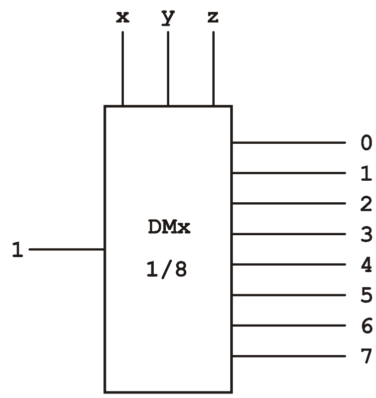


Fig.10.8 Demultiplekseri si dekodues

Për ta vërtetuar se qarku i dhënë funksionon si dekodues, le të nisemi nga shprehjet e funksioneve dalëse nga multiplekseri:

$$\mathbf{0} = \mathbf{h}\overline{\mathbf{X}}\,\overline{\mathbf{Y}}\,\overline{\mathbf{Z}}$$
$$\mathbf{1} = \mathbf{h}\overline{\mathbf{X}}\,\overline{\mathbf{Y}}\mathbf{Z}$$
$$\mathbf{2} = \mathbf{h}\overline{\mathbf{X}}\mathbf{Y}\overline{\mathbf{Z}}$$
$$\mathbf{3} = \mathbf{h}\overline{\mathbf{X}}\mathbf{Y}\mathbf{Z}$$
$$\mathbf{4} = \mathbf{h}\mathbf{X}\overline{\mathbf{Y}}\,\overline{\mathbf{Z}}$$
$$\mathbf{5} = \mathbf{h}\mathbf{X}\overline{\mathbf{Y}}\mathbf{Z}$$
$$\mathbf{6} = \mathbf{h}\mathbf{X}\mathbf{Y}\overline{\mathbf{Z}}$$
$$\mathbf{7} = \mathbf{h}\mathbf{X}\mathbf{Y}\mathbf{Z}$$

Nëse në hyrjen **h** të tij aplikohet vlera logjike **1**, qartë shihet se shprehjet e dhëna e marrin formën e njëjtë me ato të dekoduesit **3/8**.

Komparatorët

11

Komparatori (ang. comparator) është qark digjital kombinues përmes të cilit krahasohen dy numra binar **X** e **Y** dhe si rezultat në njërën nga tri daljet e qarkut merret vlera logjike **1**, për të treguar se është **X<Y**, **X=Y**, ose **X>Y**.

Gjatë vizatimit të qarqeve logjike të ndryshme, për komparatorin mund të përdoret paraqitja skematike e dhënë në Fig.11.1.

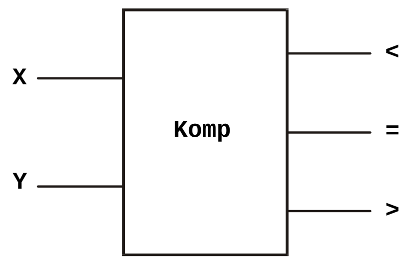


Fig.11.1 Paraqitja skematike e komparatorit

Komparatori 1-bitësh

Krahasimi i dy vlerave **1**-bitëshe bëhet përmes komparatorit digjital **1**-bitësh.

Shembull

Komparatori digjital **1**-bitësh përmes së cilit krahasohen vlerat **1**-bitëshe të variablave hyrëse **X** e **Y**. Daljet e qarkut janë:

v për **X<Y**
b për **X=Y** dhe
m për **X>Y**.

X	Y	v	b	m

$$v = \bar{X}Y$$

0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0

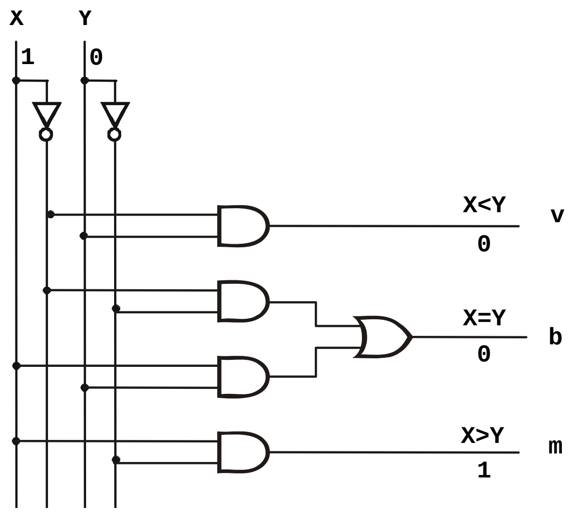


Fig.11.2 Komparatori digjital 1-bitësh

Nëse, p.sh., në dy hyrjet e qarkut logjik të dhënë në Fig.11.2 aplikohen vlerat **1** dhe **0**, në dalje të tij do të fitohen vlerat **v=0**, **b=0** dhe **m=1**, përkatësisht vlera logjike **1** paraqitet në daljen **m**, për të treguar se vlera **X** është më e madhe se vlera **Y**.

Komparatori 2-bitësh

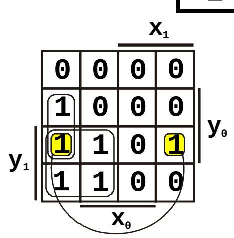
Qarku përmes të cilit mund të krahasohen dy numra binarë **2**-bitësh, përkatësisht numrat binarë $X=x_1x_0$ dhe $Y=y_1y_0$, paraqet komparatorin digjital **2**-bitësh.

Shembull Qarku logjik i komparatorit digjital **2**-bitësh, përmes të cilit krahasohen numrat binarë:

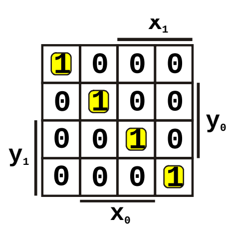
$X=x_1x_0$
$Y=y_1y_0$

Duke i pasur parasyshë ekuivalentët decimal të numrave binar që krahasohen, është plotësuar tabela vijuese, përmes së cilës plotësisht përcaktohet puna e komparatorit digjital **2**-bitësh.

X		Y		X<Y	X=Y	X>Y
x_1	x_0	y_1	y_0	v	b	m
0	0	0	0	0	1	0
0	0	0	1	1	0	0
0	0	1	0	1	0	0
0	0	1	1	1	0	0
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	1	0	0
0	1	1	1	1	0	0
1	0	0	0	0	0	1
1	0	0	1	0	0	1
1	0	1	0	0	1	0
1	0	1	1	1	0	0
1	1	0	0	0	0	1
1	1	0	1	0	0	1
1	1	1	0	0	0	1
1	1	1	1	0	1	0



$$v=\overline{x}_1y_1+\overline{x}_1\overline{x}_0y_0+\overline{x}_0y_1y_0$$



$$\begin{aligned}b&=\overline{x}_1\overline{x}_0\overline{y}_1\overline{y}_0+\overline{x}_1x_0\overline{y}_1y_0+x_1x_0y_1y_0+x_1\overline{x}_0y_1\overline{y}_0\\&=\overline{x}_1\overline{y}_1(\overline{x}_0\overline{y}_0+x_0y_0)+x_1y_1(\overline{x}_0\overline{y}_0+x_0y_0)\\&=(x_1\otimes y_1)\cdot(x_0\otimes y_0)\end{aligned}$$

			x_1	x_1	
	0	1	1	1	
	0	0	1	1	y_0
y_1	0	0	0	0	y_0
y_1	0	0	1	0	
		x_0	x_0		

$$m = x_1\bar{y}_1 + x_0\bar{y}_1\bar{y}_0 + x_1x_0\bar{y}_0$$

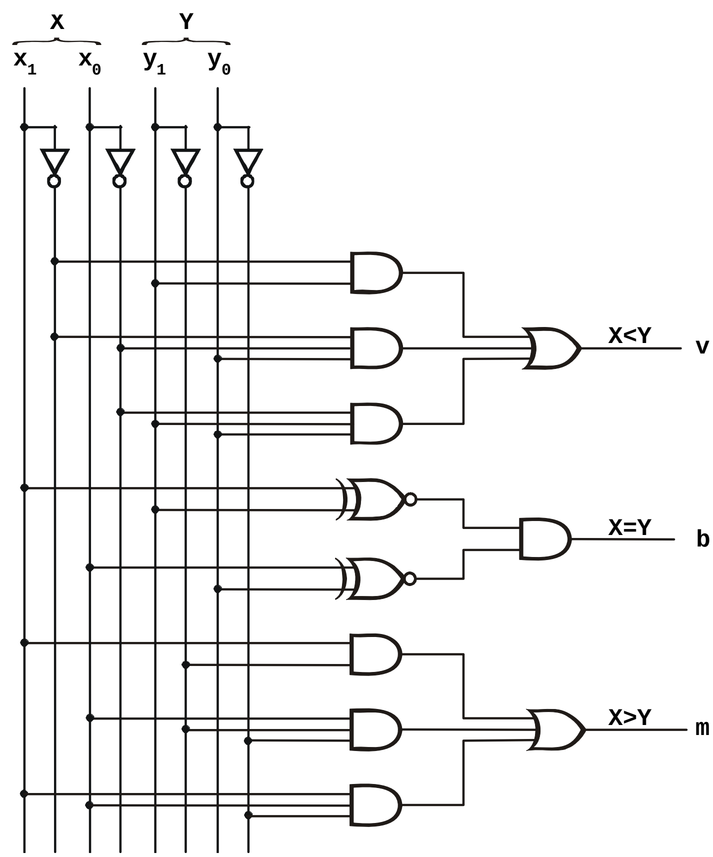


Fig.11.3 Komparatori digjital 2-bitësh

Komparatori digjital **2**-bitësh mund të realizohet edhe duke shfrytëzuar dy komparatorë digjitalë **1**-bitësh. Raportet që ekzistojnë në mes të vlerave në bitët e veçantë të numrave të cilët krahasohen, në formë përmbledhëse mund të shkruhen kështu:

$x < y$

nëse $x_1 < y_1$
ose $x_1 = y_1$ dhe $x_0 < y_0$

$x = y$

nëse $x_1 = y_1$ dhe $x_0 = y_0$

$x > y$

nëse $x_1 > y_1$
ose $x_1 = y_1$ dhe $x_0 > y_0$

Prej nga mund të shkruhen edhe shprehjet e funksioneve të daljeve të veçanta nga komparatori:

$x < y$

$$\begin{aligned} v &= (x_1 < y_1) + (x_1 = y_1)(x_0 < y_0) \\ &= \quad v_1 \quad + \quad b_1 \quad\quad v_0 \end{aligned}$$

$x = y$

$$\begin{aligned} b &= (x_1 = y_1)(x_0 = y_0) \\ &= \quad b_1 \quad\quad b_0 \end{aligned}$$

$x > y$

$$\begin{aligned} m &= (x_1 > y_1) + (x_1 = y_1)(x_0 > y_0) \\ &= \quad m_1 \quad + \quad b_1 \quad\quad m_0 \end{aligned}$$

Në bazë të këtyre shprehjeve është vizatuar qarku përkatës i komparatorit digjital **2**-bitësh, i cili është dhënë në Fig.11.4, i realizuar përmes dy komparatorëve digjitalë **1**-bitësh dhe elemente logjike themelore.

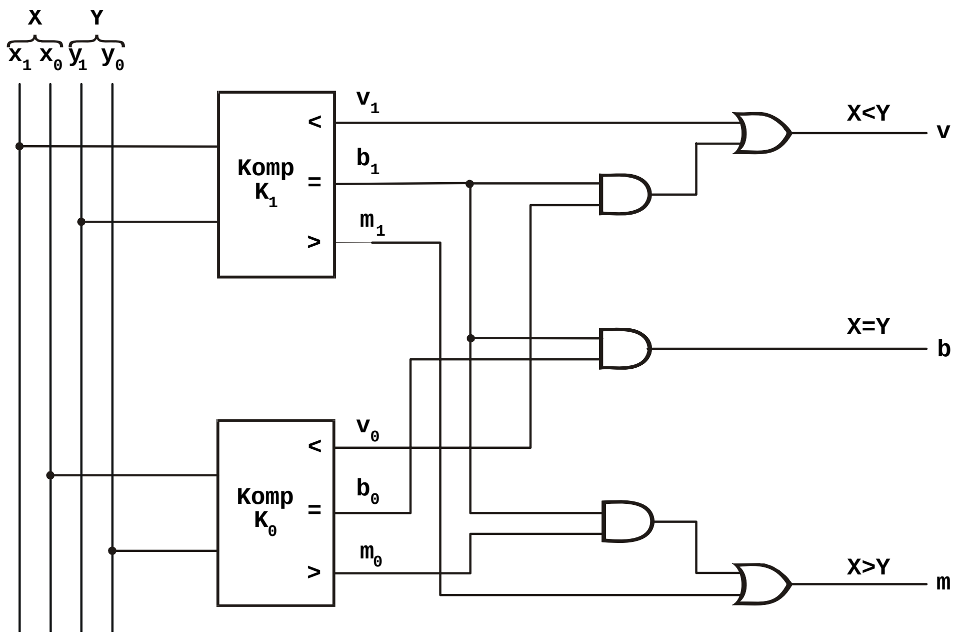


Fig.11.4 Komparatori 2-bitësh i realizuar përmes komparatorëve 1-bitësh

Për ta pasur më të qartë funksionimin e komparatorit digjital **2**-bitësh, le të marrim, p.sh., se në hyrje të tij aplikohen vlerat **X=10** dhe **Y=11**, përkatësisht vlerat decimale **2** dhe **3**, ku $x_1=1$, $x_0=0$, $y_1=1$ e $y_0=1$. Nëse këto vlera aplikohen në hyrjet e qarkut të vizatuar më sipër, në dalje të dy komparatorëve fitohen vlerat:

$v_1=0$	$v_0=0$
$b_1=1$	$b_0=0$
$m_1=0$	$m_0=1$

Pasi këto vlera të aplikohen në hyrjet e elementeve logjike, vlerat të cilat fitohen në daljet e qarkut janë:

$v=0$
$b=0$
$m=1$

gjë që tregon se **X>Y**.

Në dalje të komparatorit rezultati mund të paraqitet edhe në mënyrë vizuale, p.sh., duke shfrytëzuar indikatorë të caktuar.

Shembull

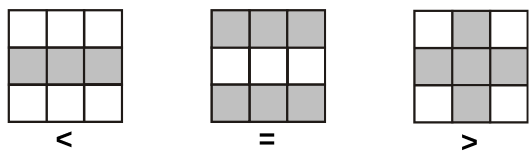
Komparatori digjital **2**-bitësh, përmes të cilit krahasohen numrat binarë:

$$X=x_1x_0$$
$$Y=y_1y_0$$

në dalje të të cilit është vendosur indikatori me **9**-sipërfaqe katrore ndriçuese:

a	b	c
d	e	f
g	h	i

Varësisht nga raporti i vlerave që krahasohen, në indikator gjenerohen simbolet **-**, **=** dhe **+**, kështu:



N	x_1	x_0	y_1	y_0	Simboli	a	b	c	d	e	f	g	h	i
0	0	0	0	0	=	1	1	1	0	0	0	1	1	1
1	0	0	0	1	-	0	0	0	1	1	1	0	0	0
2	0	0	1	0	-	0	0	0	1	1	1	0	0	0
3	0	0	1	1	-	0	0	0	1	1	1	0	0	0
4	0	1	0	0	+	0	1	0	1	1	1	0	1	0
5	0	1	0	1	=	1	1	1	0	0	0	1	1	1
6	0	1	1	0	-	0	0	0	1	1	1	0	0	0
7	0	1	1	1	-	0	0	0	1	1	1	0	0	0
8	1	0	0	0	+	0	1	0	1	1	1	0	1	0
9	1	0	0	1	+	0	1	0	1	1	1	0	1	0
10	1	0	1	0	=	1	1	1	0	0	0	1	1	1
11	1	0	1	1	-	0	0	0	1	1	1	0	0	0
12	1	1	0	0	+	0	1	0	1	1	1	0	1	0
13	1	1	0	1	+	0	1	0	1	1	1	0	1	0
14	1	1	1	0	+	0	1	0	1	1	1	0	1	0
15	1	1	1	1	=	1	1	1	0	0	0	1	1	1

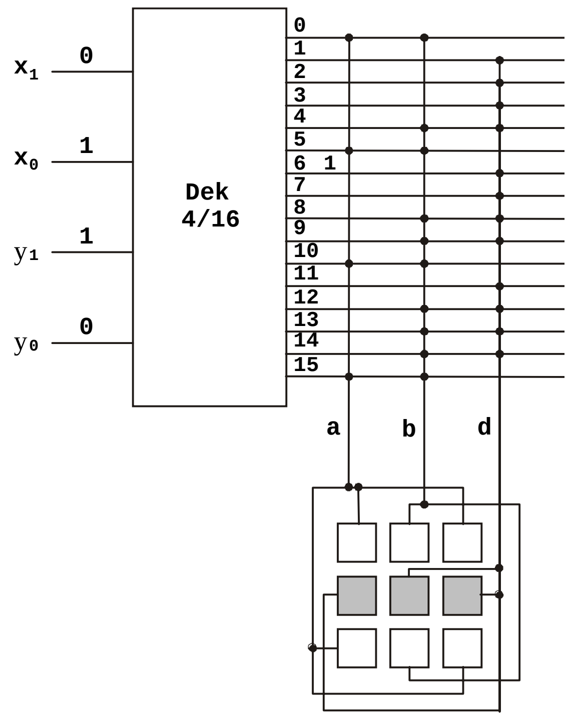


Fig.11.5 Komparatori digjital 2-bitësh me dalje të lidhura në indikator

Komparatori shumëbitësh

Për krahasimin e vlerave numerike, në praktikë përdoren komparatorë digjitalë shumëbitësh. Realizimi i tyre, duke ndjekun rrugën e zakonshme të sintezës së qarqeve logjike, është jopraktik, sepse, p.sh., edhe për komparatorë **3**-bitësh tabela e kombinimeve përkatëse do të jetë e madhe. Për këtë arsye, mund të përdoret procedura e sintezës përmes komparatorëve digjitalë **1**-bitësh, e cila u dha më sipër.

Shembull

Qarku logjik i komparatorit digjital **4**-bitësh, përmes të cilit krahasohen numrat binarë:

$$X=x_3x_2x_1x_0$$
$$Y=y_3y_2y_1y_0$$

Qarku logjik i komparatorit **4**-bitësh përmban **4** komparatorë **1**-bitësh, në dalje të të cilëve vendosen elemente logjike, ngjashëm si edhe te komparatori digjital **2**-bitësh. Ky qark logjik, në formë të përgjithshme është dhënë në Fig.11.6.

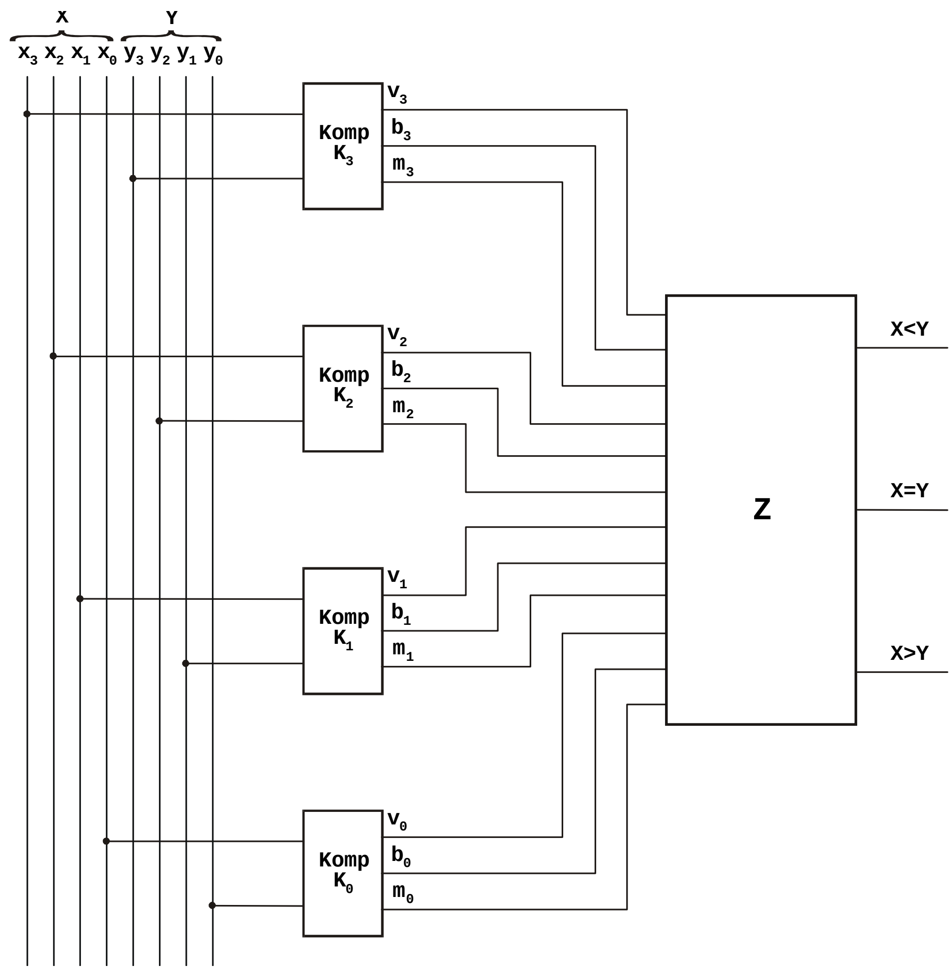


Fig.11.6 Komparatori digjital 4-bitësh

Gjatë sintezës së qarkut kombinues **Z** duhet pasur parasysh tabelën vijuese, ku me **+** nënkuptohen raporte të çfarëdoshme të vlerave hyrëse, përkatësisht përmes këtij simboli tregohet se për vlerat dalëse nga komparatori nuk ka rëndësi raporti mes vlerave hyrëse përkatëse.

Hyrjet				Daljet		
x_3,y_3	x_2,y_2	x_1,y_1	x_0,y_0	v	b	m
$x_3<y_3$	+	+	+	1	0	0
$x_3>y_3$	+	+	+	0	0	1
$x_3=y_3$	$x_2<y_2$	+	+	1	0	0
$x_3=y_3$	$x_2>y_2$	+	+	0	0	1
$x_3=y_3$	$x_2=y_2$	$x_1<y_1$	+	1	0	0
$x_3=y_3$	$x_2=y_2$	$x_1>y_1$	+	0	0	1
$x_3=y_3$	$x_2=y_2$	$x_1=y_1$	$x_0<y_0$	1	0	0
$x_3=y_3$	$x_2=y_2$	$x_1=y_1$	$x_0>y_0$	0	0	1
$x_3=y_3$	$x_2=y_2$	$x_1=y_1$	$x_0=y_0$	0	1	0

Nga kjo tabelë mund të nxirren shprehjet e raporteve mes madhësive hyrëse në komparator, kur $X<Y$, $X=Y$ dhe $X>Y$:

$X<Y$

nëse $x_3<y_3$
ose $x_3=y_3$ dhe $x_2<y_2$
ose $x_3=y_3$, $x_2=y_2$ dhe $x_1<y_1$
ose $x_3=y_3$, $x_2=y_2$, $x_1=y_1$ dhe $x_0<y_0$

$X=Y$

nëse $x_3=y_3$, $x_2=y_2$, $x_1=y_1$ dhe $x_0=y_0$

$X>Y$

nëse $x_3>y_3$
ose $x_3=y_3$ dhe $x_2>y_2$
ose $x_3=y_3$, $x_2=y_2$ dhe $x_1>y_1$
ose $x_3=y_3$, $x_2=y_2$, $x_1=y_1$ dhe $x_0>y_0$

Në bazë të raporteve të dhëna më sipër, në analogji me shprehjet e nxjerra te komparatori digjital **2**-bitësh, mund të shkruhen shprehjet përfundimtare të funksioneve dalëse nga komparatori digjital **4**-bitësh:

$$\mathbf{v} = \mathbf{v_3} + \mathbf{b_3 v_2} + \mathbf{b_3 b_2 v_1} + \mathbf{b_3 b_2 b_1 v_0}$$
$$\mathbf{b} = \mathbf{b_3 b_2 b_1 b_0}$$
$$\mathbf{m} = \mathbf{m_3} + \mathbf{b_3 m_2} + \mathbf{b_3 b_2 m_1} + \mathbf{b_3 b_2 b_1 m_0}$$

Nga shprehjet e fituara, mjaft qartë duket ligjshmëria në bazë të së cilës mund të shkruhen shprehjet për një komparator digjital shumëbitësh.

Në fund, duke përdorur shprehjet e fituara, qarku i komparatorit **4**-bitësh i cili u dha më sipër, mund të plotësohet edhe me pjesën e qarkut kombinues **Z**. Pamja përfundimtare e tij do të jetë e ngjashme me atë të komparatorit digjital **2**-bitësh.

Për realizimin e komparatorëve shumbitësh, në praktikë përdoren komparatorë me numër të caktuar bitësh, por të cilët kanë edhe hyrje të veçanta për raportin e komparimit të kryer më parë. Me lidhjen serike ose paralele të më shumë komparatorëve të tillë mund të realizohen komparatorë të numrave më të mëdhenj.

Shembull

Komparatori digjital për krahasimin e vlerave numerike **X** dhe **Y**, **12**-bitësh, i realizuar duke shfrytëzuar një komponente krahasuese **4**-bitëshe, e cila ka edhe **3** hyrje të veçanta për marrjen e vlerave nga krahasimi paraprak.

Komponentja krahasuese **K** në fjalë, përveç **3** hyrjeve (**r**, **s**, **t**) për vlerat e krahasimit paraprak, për krahasimin e dy numrave **4**-bitësh, duhet t'i ketë edhe **8** hyrje të tjera, ashtu që do të duket si në Fig.11.7, ku në hyrjen **r** aplikohet vlera e cila i përgjigjet rezulatit **X<Y** nga krahasimi paraprak, në hyrjen **s** - vlera e rezultatit **X=Y** dhe në hyrjen **t** - vlera e rezultatit **X>Y**.

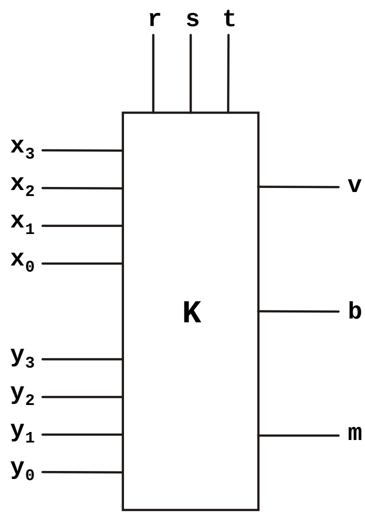


Fig.11.7 Komponeta krahasuese 4-bitëshe

Puna e kësaj komponente mund të përshkruhet përmes tabelës vijuese.

Hyrjet							Daljet		
x_3,y_3	x_2,y_2	x_1,y_1	x_0,y_0	r	s	t	v	b	m
$x_3<y_3$	+	+	+	+	+	+	1	0	0
$x_3>y_3$	+	+	+	+	+	+	0	0	1
$x_3=y_3$	$x_2<y_2$	+	+	+	+	+	1	0	0
$x_3=y_3$	$x_2>y_2$	+	+	+	+	+	0	0	1
$x_3=y_3$	$x_2=y_2$	$x_1<y_1$	+	+	+	+	1	0	0
$x_3=y_3$	$x_2=y_2$	$x_1>y_1$	+	+	+	+	0	0	1
$x_3=y_3$	$x_2=y_2$	$x_1=y_1$	$x_0<y_0$	+	+	+	1	0	0
$x_3=y_3$	$x_2=y_2$	$x_1=y_1$	$x_0>y_0$	+	+	+	0	0	1
$x_3=y_3$	$x_2=y_2$	$x_1=y_1$	$x_0=y_0$	1	0	0	1	0	0
$x_3=y_3$	$x_2=y_2$	$x_1=y_1$	$x_0=y_0$	0	1	0	0	1	0
$x_3=y_3$	$x_2=y_2$	$x_1=y_1$	$x_0=y_0$	0	0	1	0	0	1

Për realizimin e komparatorit digjital **12**-bitësh duhet të shfrytëzohen **3** komponente krahasuese, të lidhura sipas skemës të dhënë në Fig.11.8.

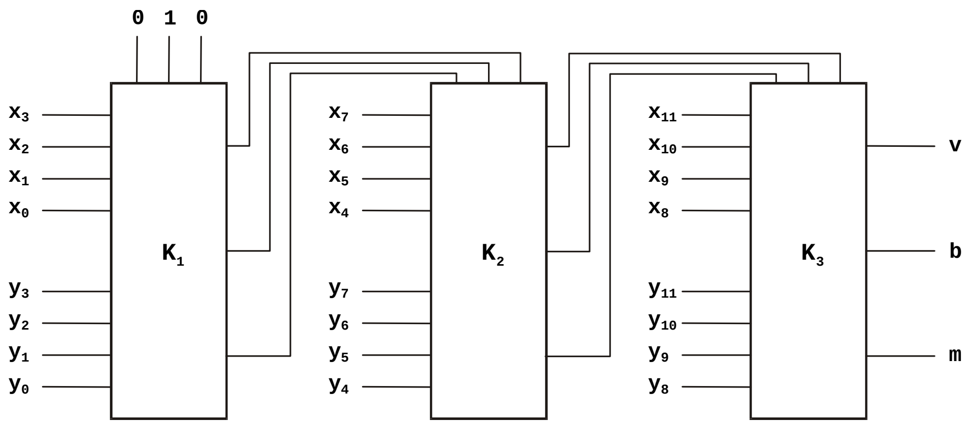


Fig.11.8 Komparatori digjital 12-bitësh

Në hyrjet **r**, **s** dhe **t**, te komponentja krahasuese $\mathbf{K_1}$, si vlera fillestare janë marrë vlerat **0**, **1** dhe **0** - përkatësisht, me qëllim që të mos ndikojë në rezultatin e krahasimit.

Gjeneratorët e paritetit

12

Gjatë zhvendosjes së të dhënave brenda pajisjeve, ose mes pajisjeve të ndryshme digjitale, si dhe gjatë transmetimit të tyre në distancë, nevojitet një shkallë e lartë e sigurisë, përkatësisht duhet të eliminohen gabimet e mundshme. Njëra nga mundësitë që shpesh aplikohet në praktikë për eliminimin e gabimeve është shtimi i bitit për paritet në çdo informatë elementare që transmetohet. Qarqet përmes të cilave gjenerohet biti për paritet, quhen *gjeneratorë të paritetit* (ang. parity generator).

Gjeneratori i paritetit për fjalët kodike **n**-bitëshe në hyrje të tij, skematikisht mund të paraqitet si në Fig.12.1, ku në daljen **p** merret vlera e bitit për paritet që gjenerohet.

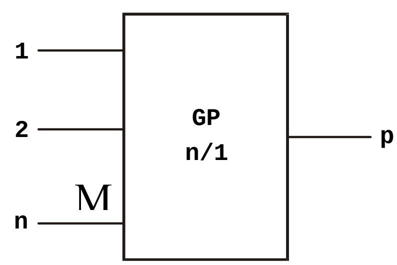


Fig.12.1 Paraqitja skematike e gjeneratorit të paritetit

Gjeneratorët e zakonshëm

Siç është përmendur edhe në pjesën e librit për kodet, gjatë transmetimit të informatave binare, me qëllim të detektimit të gabimeve, ose të detektimit dhe të korrigjimit të tyre, çdo fjale kodike mund t'i shtohet edhe një shifër për *paritet të njësheve*. Pariteti mund të jetë *paritet çift* (ang. even parity) ose *paritet tek* (ang. odd parity), varësisht nga ajo se a *bëhet çift ose tek numri i njësheve*, pas shtimit të shifrës për paritet, përfshirë këtu edhe shifrën e shtuar.

Shembull

Qarku përmes të cilit gjenerohet biti për *paritet tek* **g**, për fjalët kodike të kodit binar natyror **3**-bitësh.

A	B	C	g
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

$$
\begin{aligned}
g &= \bar{A}\bar{B}\bar{C} + \bar{A}BC + A\bar{B}C + AB\bar{C} \\
&= \bar{A}(\bar{B}\bar{C} + BC) + A(\bar{B}C + B\bar{C}) \\
&= \bar{A}(B \otimes C) + A(B \oplus C) \\
&= \bar{A}(\overline{B \oplus C}) + A(B \oplus C) \\
&= A \otimes (B \oplus C)
\end{aligned}
$$

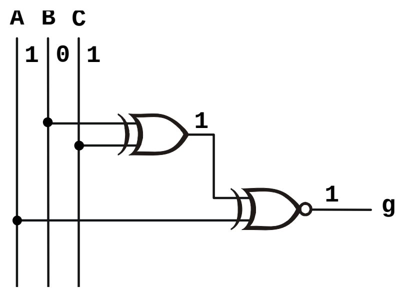


Fig.12.2 Gjeneratori i paritetit tek për fjalët kodike të kodit binar natyror 3-bitësh

Nëse, p.sh., në hyrje të qarkut të dhënë në Fig.12.2 aplikohet fjala kodike **101**, në dalje të tij do të fitohet vlera **g=1**, gjë që mund të vërtetohet nëse merren parasysh tabelat e funksionimit të elementeve logjike speciale që paraqiten në qark.

Plotësisht njëlloj veprohet edhe gjatë sintezës së gjeneratorëve të bitit për paritetin çift të njësheve.

Shembull

Qarku përmes të cilit gjenerohet biti për paritet çift **P**, te fjalët kodike të kodit **NBCD**. Gjatë sintezës së qarkut, për fjalët kodike që nuk i përkasin këtij kodi, biti për paritet merret i çfarëdoshëm.

A	B	C	D	P

0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	+
1	0	1	1	+
1	1	0	0	+
1	1	0	1	+
1	1	1	0	+
1	1	1	1	+

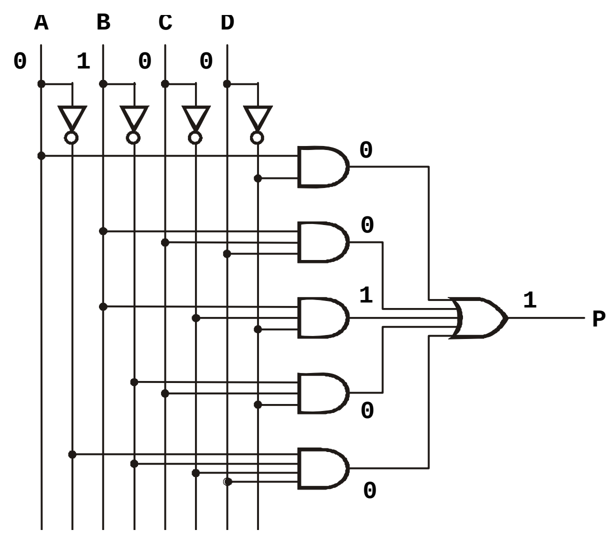


Fig.12.3 Gjeneratori i bitit për paritet çift për fjalët kodike të kodit ***NBCD***

Puna e qarkut të dhënë në Fig.12.3 mund të testohet, nëse në hyrje të tij aplikohet një fjalë kodike e kodit **NBCD**. Kështu, p.sh., për fjalën kodike **0100** në dalje të qarkut do të merret vlera **P=1**, me çrast, pasi kjo shifër t'i shtohet fjalës kodike, numri i njësheve të saj bëhet çift. Nëse në hyrje të qarkut aplikohet kombinimi i shifrave binare i cili nuk i përket kodit **NBCD**, vlera që fitohet në dalje të qarkut do të jetë e çfarëdoshme, përkatësisht **0** ose **1** dhe nuk lidhet me numrin e shifrave me vlerë **1** të kombinimit në fjalë.

Qarku i dhënë më sipër për gjenerimin e shifrës për paritet mund të thjeshtohet mjaft, nëse edhe kombinimeve që nuk i përkasin kodit **NBCD** u shtohet shifra për paritet, përkatësisht nëse merret kodi binar natyror.

Shembull Gjeneratori i bitit për paritet çift **P**, te fjalët kodike të kodit binar natyror **4**-bitësh.

A	B	C	D	P
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

$$
\begin{aligned}
P &= \bar{A}\bar{B}\bar{C}D+\bar{A}\bar{B}C\bar{D}+\bar{A}B\bar{C}\bar{D}+\bar{A}BCD+A\bar{B}\bar{C}\bar{D}+A\bar{B}CD+AB\bar{C}D+ABC\bar{D} \\
&= \bar{A}\bar{B}(\bar{C}D+C\bar{D})+\bar{A}B(\bar{C}\bar{D}+CD)+A\bar{B}(\bar{C}\bar{D}+CD)+AB(\bar{C}D+C\bar{D}) \\
&= \bar{A}\bar{B}(C\oplus D)+\bar{A}B(C\otimes D)+A\bar{B}(C\otimes D)+AB(C\oplus D) \\
&= (\bar{A}\bar{B}+AB)\cdot(C\oplus D)+(\bar{A}B+A\bar{B})\cdot(C\otimes D) \\
&= (A\otimes B)\cdot(C\oplus D)+(A\oplus B)\cdot(C\otimes D) \\
&= (\overline{A\oplus B})\cdot(C\oplus D)+(A\oplus B)\cdot(\overline{C\oplus D}) \\
&= (A\oplus B)\oplus(C\oplus D)
\end{aligned}
$$

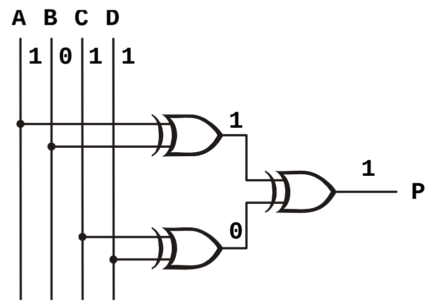


Fig.12.4 Gjeneratori i paritetit çift për fjalët kodike të kodit binar natyror 4-bitësh

Nëse në hyrje të gjeneratorit të dhënë në Fig.12.4 aplikohet fjala kodike **1011**, në dalje të tij do të fitohet vlera **P=1**. Nëse kjo vlerë, p.sh., shtohet para fjalës kodike së dhënë, numri i njësheve në të bëhet çift, përkatësisht fjala e re kodike është **11011**.

Gjeneratorët për fjalët kodike shumëbitëshe

Kur fjalët kodike janë shumëbitëshe, për sintezë të gjeneratorit të bitit për paritet, nuk është e nevojshme të zbatohet procedura e dhënë më sipër, sepse tabela përkatëse e kombinimeve do të jetë e madhe. Por, për këtë qëllim gjenerimi i bitit për paritet realizohet duke copëtuar gjeneratorin në dy ose edhe në më shumë grupe shifrash binare.

Shembull	Gjeneratori i bitit për paritet çift **P**, te fjalët kodike të kodit binar natyror **8**-bitësh.

Nëse në hyrje të qarkut aplikohen kombinimet **8**-bitëshe $(ABCDEFGH)_2$, gjeneratori i bitit për paritet mund të realizohet duke copëtuar atë në dy pjesë, me hyrjet **4**-bitëshe $(ABCD)_2$ dhe $(EFGH)_2$. Kështu, në dy daljet P_1 dhe P_2 të pjesëve të veçanta të qarkut, bitët përkatës për paritet gjenden përmes shprehjeve:

$$\mathbf{P_1} = (\mathbf{A} \oplus \mathbf{B}) \oplus (\mathbf{C} \oplus \mathbf{D})$$
$$\mathbf{P_2} = (\mathbf{E} \oplus \mathbf{F}) \oplus (\mathbf{G} \oplus \mathbf{H})$$

kurse për llogaritjen e bitit për paritet në dalje të qarkut mund të përdoret shprehja:

$$\begin{aligned}\mathbf{P} &= \mathbf{P_1} \oplus \mathbf{P_2} \\ &= [(\mathbf{A} \oplus \mathbf{B}) \oplus (\mathbf{C} \oplus \mathbf{D})] \oplus [(\mathbf{E} \oplus \mathbf{F}) \oplus (\mathbf{G} \oplus \mathbf{H})]\end{aligned}$$

Në bazë të shprehjeve të dhëna, mund të vizatohet qarku logjik i gjeneratorit të bitit për paritet, i cili është dhënë në Fig.12.5.

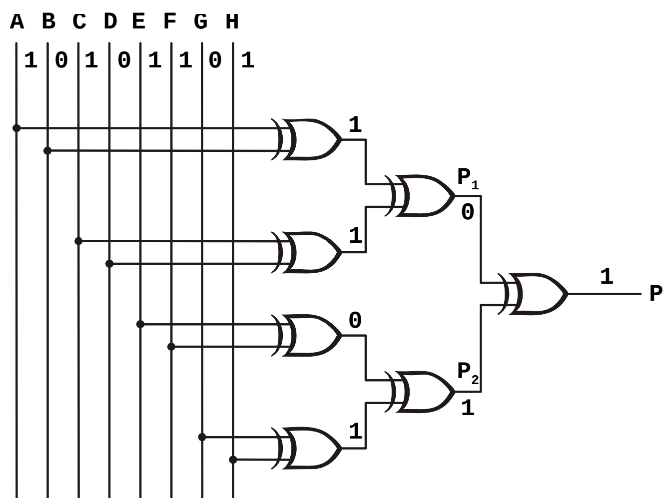


Fig.12.5 Gjeneratori i paritetit çift për fjalët kodike të kodit binar natyror 8-bitësh

Në Fig.12.5 shihen qartë dy pjesët e qarkut, për gjenerimin e bitëve për paritet të dy tërësive të veçanta **4**-bitëshe. Kështu, p.sh., nëse në hyrje të qarkut aplikohet fjala kodike **10101101**, për **4** bitat e parë **1010**, në daljen $\mathbf{P_1}$ do të fitohet vlera **0**, kurse për **4** bitët e tjerë **1101**, biti për paritet në daljen $\mathbf{P_2}$ do të jetë **1**. Në bazë të këtyre dy vlerave dhe shprehjes përkatëse, në daljen **P** do të fitohet vlera **1**, e cila i përgjigjet bitit për paritet të fjalës kodike **8**-bitëshe të aplikuar në hyrje të qarkut.

Copëtimi i gjeneratorit në pjesë me më pakë bitë mund të bëhet edhe nëse numri i bitëve nuk është çift, ose duke marrë numër të ndryshëm bitësh për pjesët e veçanta të qarkut.

Shembull

Gjeneratori i bitit për paritet tek **Q**, te fjalët kodike të kodit **ASCII**, të dhënë në tabelën e Fig.2.21.

Me qëllim që realizimi i gjeneratorit të bitit për paritet tek të fjalëve kodike **7**-bitëshe të kodit **ASCII**, pa përpilimin e tabelës së kombinimeve për **7** variabla, qarku përkatës mund të copëtohet në dy pjesë, ashtu që njëra pjesë me daljen Q_1 të jetë **4**-bitëshe, kurse pjesa tjetër me daljen Q_2 të jetë **3**-bitëshe.

Nëse për sintezën e gjeneratorit të bitit për paritet tek për fjalët kodike **4**-bitëshe përdoret procedura e sintezës së gjeneratorit për paritet çift, e dhënë më sipër, tabela e kombinimeve, shprehja e funksionit dalës nga qarku dhe qarku logjik përkatës do të duken si në vijim.

A	B	C	D	Q_1
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

$$
\begin{aligned}
Q_1 &= \bar{A}\bar{B}\bar{C}\bar{D}+\bar{A}\bar{B}CD+\bar{A}B\bar{C}D+\bar{A}BC\bar{D}+A\bar{B}\bar{C}D+A\bar{B}C\bar{D}+AB\bar{C}\bar{D}+ABCD \\
&= \bar{A}\bar{B}\,(\bar{C}\bar{D}+CD)+\bar{A}B\,(\bar{C}D+C\bar{D})+A\bar{B}\,(\bar{C}D+C\bar{D})+AB\,(\bar{C}\bar{D}+CD) \\
&= \bar{A}\bar{B}\,(C\otimes D)+\bar{A}B\,(C\oplus D)+A\bar{B}\,(C\oplus D)+AB\,(C\otimes D) \\
&= (\bar{A}\bar{B}+AB)\,(C\otimes D)+(\bar{A}B+A\bar{B})\,(C\oplus D) \\
&= (A\otimes B)\,(C\otimes D)+(A\oplus B)\,(C\oplus D) \\
&= (\overline{A\oplus B})\,(\overline{C\oplus D})+(A\oplus B)\,(C\oplus D) \\
&= (A\oplus B)\otimes(C\oplus D)
\end{aligned}
$$

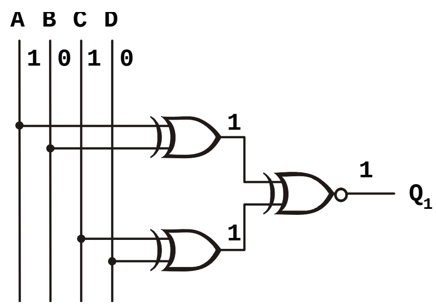


***Fig.12.6 Gjeneratori i paritetit për pjesën 4-bitëshe të fjalëve kodike të kodit* ASCII**

Për ta testuar punën e qarkut të dhënë në Fig.12.6, le ta aplikojmë në hyrje të tij, p.sh., fjalën kodike **1010**. Vlera dalëse nga qarku do të jetë $Q_1=1$, përkatësisht pas shtimit të kësaj shifre fjala e re kodike është **11010** dhe ka numër tek njëshesh.

Duke pasur parasysh gjeneratorin **3**-bitësh të dhënë në fillim, për pjesën **3**-bitëshe të qarkut, shprehja e funksionit dalës do të jetë:

$$Q_2 = E\otimes(F\oplus G)$$

Në fund, duke i bashkuar dy pjesët e gjeneratorit në një, ashtu që dy daljet e tyre të lidhen në një element logjik **JODHE**, do të fitohet gjeneratori i bitit për paritet tek të fjalëve kodike të kodit **ASCII**, i cili është dhënë në Fig.12.7.

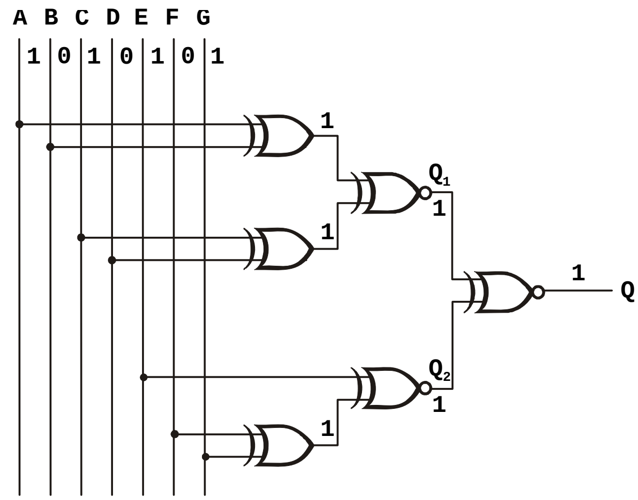


***Fig.12.7 Gjeneratori i paritetit tek për fjalët kodike të kodit* ASCII**

Për ta testuar punën e qarkut të dhënë në Fig.12.7, le të aplikohet, p.sh., në hyrje të tij fjala kodike **1010101**, e cila në kodin **ASCII** i përket shkronjës **U**. Si rezultat, në daljet e dy pjesëve të qarkut do të gjenerohen vlerat:

$$Q_1 = 1$$
$$Q_2 = 1$$

kurse vlera e bitit për paritet tek të fjalës kodike të dhënë është **Q=1**. Nëse biti për paritet shtohet para, fjala kodike me bit të shtuar për paritet tek është **11010101**.

13 Detektorët e paritetit

Pasi informatat binare bashkë me bitët e paritetit të transmetohen në destinacion, përmes qarqeve logjike të cilat njihen si *detektorë të paritetit* (ang. parity checker), mund të kontrollohet pariteti i shifrave binare brenda tyre.

Detektori i paritetit për fjalët kodike **m**-bitëshe në hyrje të tij, skematikisht mund të paraqitet si në Fig.13.1.

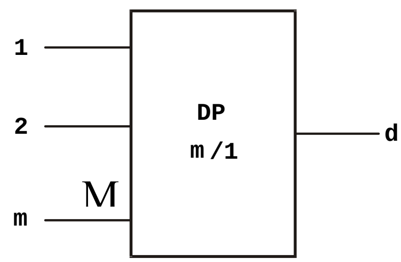


Fig.13.1 Paraqitja skematike e detektorit të paritetit

Krahasuar me gjeneratorin e paritetit me **n**-hyrje, detektori i paritetit ka **m=n+1** hyrje, sepse në hyrje të tij vjen edhe biti për çiftësi i shtuar nga gjeneratori i paritetit. Nëse gjatë kontrollës zbulohet se nuk është ruajtur pariteti, në daljen **d** të detektorit të paritetit merret vlera logjike **1**, për ta treguar gabimin.

Detektorët e zakonshëm

Në pjesën vijuese do të flitet për detektorët e paritetit të sintetizuar *duke marrë se numri maksimal i gabimeve të mundshme gjatë transmetimit të informatave elementare është një*, përkatësisht se brenda një informate e cila kontrollohet mund të ndryshojë vetëm një shifër binare.

Shembull

Detektori i *paritet tek* brenda fjalëve kodike të kodit binar natyror **3**-bitësh, të cilat e përmbajnë edhe bitin për *paritet tek* **g**.

A	B	C	g	d
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

$$
\begin{aligned}
d &= \overline{A}\,\overline{B}\,\overline{C}\,\overline{g} + \overline{A}\,\overline{B}\,C\,g + \overline{A}\,B\,\overline{C}\,g + \overline{A}\,B\,C\,\overline{g} \\
&\quad + A\,\overline{B}\,\overline{C}\,g + A\,\overline{B}\,C\,\overline{g} + A\,B\,\overline{C}\,\overline{g} + A\,B\,C\,g \\
&= \overline{A}\,\overline{B}(\overline{C}\,\overline{g} + Cg) + \overline{A}\,B(\overline{C}\,g + C\overline{g}) \\
&\quad + A\,\overline{B}(\overline{C}\,g + C\,\overline{g}) + A\,B(\overline{C}\,\overline{g} + Cg) \\
&= (A \otimes B)\,(C \otimes g) + (A \oplus B)\,(C \oplus g) \\
&= (A \otimes B)\,(C \otimes g) + (\overline{A \otimes B})\,(\overline{C \otimes g}) \\
&= (A \otimes B) \otimes (C \otimes g)
\end{aligned}
$$

Këtu, për zbulimin e gabimeve të cilat kanë ndodhur brenda fjalëve kodike gjatë transmetimit të tyre kontrollohet çiftësia e njësheve. Nëse zbulohet numër çift i njësheve, ose nëse fjala kodike ka vetëm zero, në dalje të detektorit gjenerohet vlera logjike **1**, për të sinjalizuar se ka ndodhur gabim. Qarku logjik i dhënë në Fig.13.2, është vizatuar në bazë të shprehjes së fituar më sipër për funksionin **d**, dhe merret si detektor i paritetit **4**-bitësh.

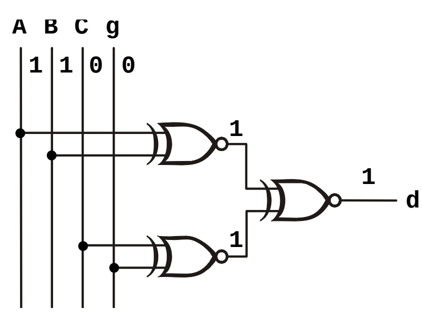


Fig.13.2 Detektori i paritetit tek për fjalët kodike 4-bitëshe

Për ta testuar funksionimin e qarkut, nëse p.sh. në hyrje të tij aplikohet kombinimi i shifrave binare **1100**, në dalje të detektorit do të gjenerohet shifra binare **d=1**, me të cilën sinjalizohet gabimi në informatën e pranuar.

Detektori i realizuar përmes gjeneratorëve

Problemi i detektimit të gabimeve në transmetim përmes paritetit mund të zgjidhet edhe duke shfrytëzuar gjeneratorë të paritetit, të vendosur te dhënësi dhe marrësi i informatave binare.

Shembull

Detektori i realizuar përmes *dy gjeneratorëve të paritetit çift* **4**-bitësh, të vendosur te dhënësi dhe te marrësi i informatave që transmetohen.

Nëse me **P** shënohet gjeneratori i bitit për paritet çift, duke shfrytëzuar dy gjenerator të tillë (**P_1** dhe **P_2**), gabimi në transmetim mund të detektohet përmes lidhjeve që shihen në Fig.13.3.

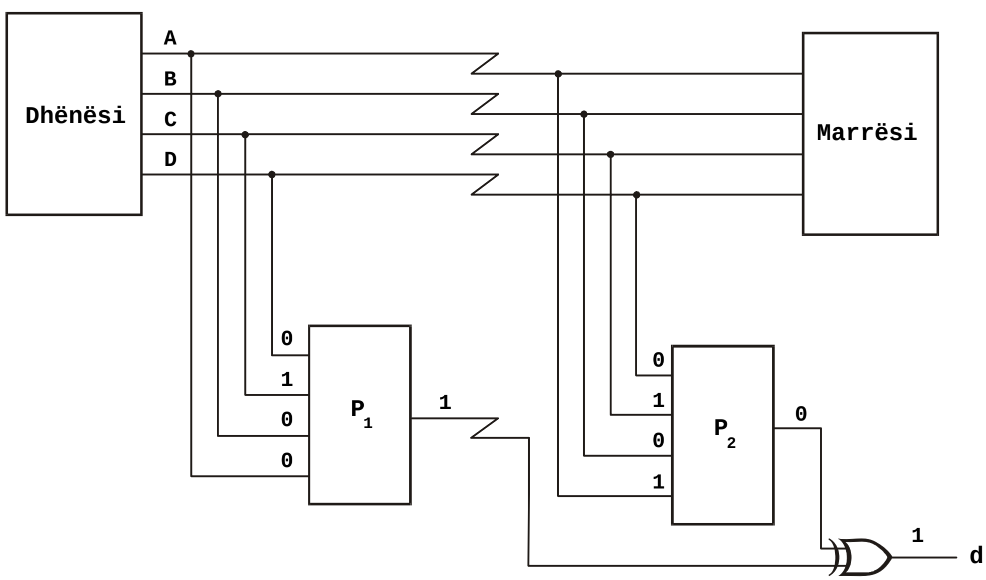


Fig.13.3 Detektori i i realizuar përmes dy gjeneratorëve të paritetit çift

Për ta pasur më të qartë funksionimin e qarkut të dhënë në Fig.13.3, le të marrim, p.sh., se në daljet **A**, **B**, **C** dhe **D** të dhënësit, jepet informata binare **0100**. Pas shtimit të bitit për paritet çift përmes gjeneratorit P_1, informata që duhet të transmetohet është **10100**. Por, nëse në vend të informatës së transmetuar pranohet informata e gabuar **10101**, në dalje të gjeneratorit P_2 do të fitohet biti për paritet me vlerë **0**, kurse në daljen **d** të elementit logjik ekskluziv-**OSE** merret vlera **1**, për të sinjalizuar se ka ndodhur gabim gjatë transmetimit.

Komplementuesit

14

Siç është shpjeguar në kapitullin e parë, operacionet aritmetikore mund të kryhen në rrugë direkte, ose duke i shfrytëzuar numrat komplementarë. Në rastin e dytë, gjatë realizimit të qarqeve për kryerjen e operacioneve aritmetikore, duhet përdorur edhe qarqe logjike për komplementimin e numrave, të cilët njihen si *komplementues*.

Këtu do të bëhet fjalë për qarqet të cilët e bëjnë komplementimin e numrave binarë natyror dhe numrave të shprehur në kodin **NBCD**. Qarku përmes së cilit për numrin **m**-bitësh gjendet komplementi përkatës **n**-bitësh, skematikisht mund të paraqitet si në Fig.14.1.

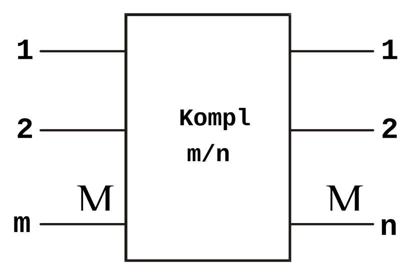


Fig.14.1 Paraqitja skematike e komplementuesit

Komplementuesi binar

Për realizimin e qarqeve përmes së cilave bëhet komplementimi i numrave binarë, shfrytëzohet principi i komplementimit direkt, i shpjeguar në kapitullin e parë të librit.

Shembull

Qarku logjik përmes të cilit për numrat binarë $(ABCD)_2$, në daljet **x**, **y**, **z** dhe **v**, merren **1**-komplementet e tyre.

A	B	C	D	x	y	z	v
0	0	0	0	1	1	1	1
0	0	0	1	1	1	1	0
0	0	1	0	1	1	0	1
0	0	1	1	1	1	0	0
0	1	0	0	1	0	1	1
0	1	0	1	1	0	1	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	0	0
1	0	0	0	0	1	1	1
1	0	0	1	0	1	1	0
1	0	1	0	0	1	0	1
1	0	1	1	0	1	0	0
1	1	0	0	0	0	1	1
1	1	0	1	0	0	1	0
1	1	1	0	0	0	0	1
1	1	1	1	0	0	0	0

$$x = \overline{A}$$
$$y = \overline{B}$$
$$z = \overline{C}$$
$$v = \overline{D}$$

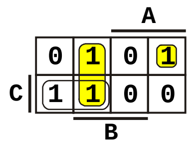
Nga shprehjet e fituara shihet qartë se qarku për gjetjen e **1**-komplementit të numrit binar të dhënë do të përmbaj vetëm katër invertor.

Ligjshmëria e gjetjes së **2**-komplementit nuk është aq e thjeshtë sa ajo që u dha më sipër, sepse **1**-komplementit përkatës duhet shtuar vlerën **1**.

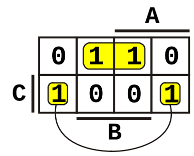
Shembull

Qarku logjik përmes të cilit gjendet **2**-komplementi $(gxyz)_2$, i numrit binar natyror $(ABC)_2$, i cili aplikohet në hyrje të tij, ku **g** është derdhja që paraqitet gjatë komplementimit.

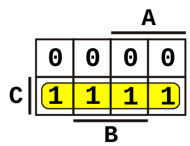
A	B	C	g	x	y	z
0	0	0	1	0	0	0
0	0	1	0	1	1	1
0	1	0	0	1	1	0
0	1	1	0	1	0	1
1	0	0	0	1	0	0
1	0	1	0	0	1	1
1	1	0	0	0	1	0
1	1	1	0	0	0	1



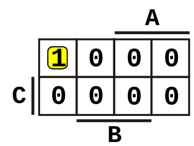
$$x = A\bar{B}\bar{C} + \bar{A}B + \bar{A}C$$



$$y = B\bar{C} + \bar{B}C$$



$$z = C$$



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

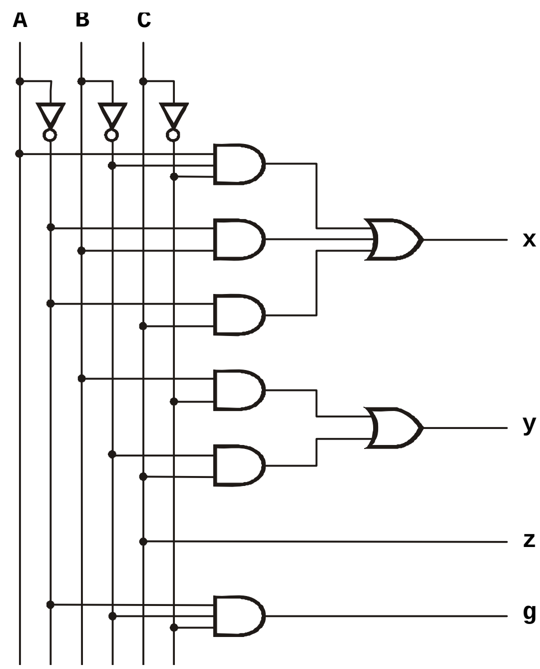


Fig.14.2 Qarku për gjetjen e 2-komplementit të numrit binar natyror 3-bitësh

Për realizimin e qarqeve për gjetjen e **2**-komplementit, sidomos gjatë komplementimit të numrave shumëbitësh, mund të përdoren edhe multiplekserët.

Shembull

Qarku logjik digjital përmes të cilit për numrin binar $(ABCD)_2$ në hyrje të tij gjendet **2**-komplementi përkatës $(QRSTV)_2$, i realizuar duke shfrytëzuar **5** multiplekserë **4/1**.

A	B	C	D	Q	R	S	T	V
0	0	0	0	1	0	0	0	0
0	0	0	1	0	1	1	1	1
0	0	1	0	0	1	1	1	1
0	0	1	1	0	1	1	0	1
0	1	0	0	0	1	1	0	0
0	1	0	1	0	1	0	1	1
0	1	1	0	0	1	0	1	0
0	1	1	1	0	1	0	0	1
1	0	0	0	0	1	0	0	0
1	0	0	1	0	0	1	1	1
1	0	1	0	0	0	1	1	0
1	0	1	1	0	0	1	0	1
1	1	0	0	0	0	1	0	0
1	1	0	1	0	0	0	1	1
1	1	1	0	0	0	0	1	0
1	1	1	1	0	0	0	0	1

CD				
00	I_0	I_0	I_0	I_0
01	I_1	I_1	I_1	I_1
11	I_3	I_3	I_3	I_3
10	I_2	I_2	I_2	I_2

Q (A, B)

1			

R (A, B)

0	1	0	1
1	1	0	0
1	1	0	0
1	1	0	0

S (A, B)

0	1	1	0
1	0	0	1
1	0	0	1
1	0	0	1

T (A, B)

0	0	0	0
1	1	1	1
0	0	0	0
1	1	1	1

V (A, B)

0	0	0	0
1	1	1	1
1	1	1	1
0	0	0	0

	I_0	I_1	I_2	I_3
Q	$\bar{A}\bar{B}$	0	0	0
R	$A \otimes B$	$\bar{A}$	$\bar{A}$	$\bar{A}$
S	B	$\bar{B}$	$\bar{B}$	$\bar{B}$
T	0	1	1	0
V	0	1	0	1

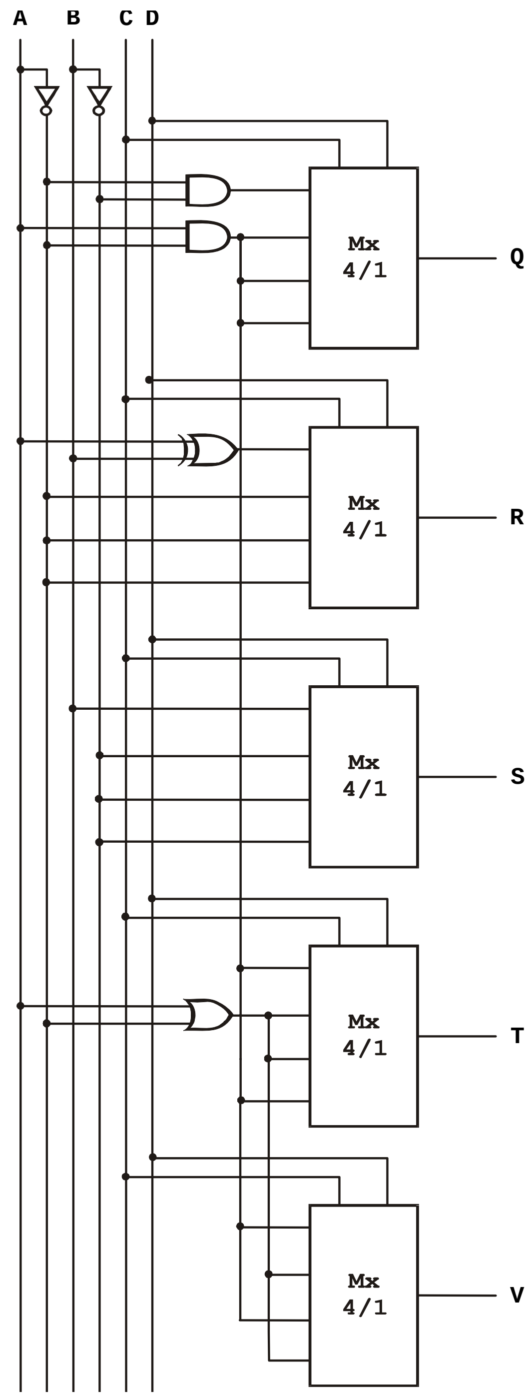


Fig.14.3 Qarku për komplementim i realizuar me multiplekser

Në praktikë, takohen shpeshherë qarqe të cilat janë në gjendje të kryejnë dy ose edhe më shumë operacione, gjë që komandohet duke aplikuar vlera të caktuara në hyrje të veçanta të tyre.

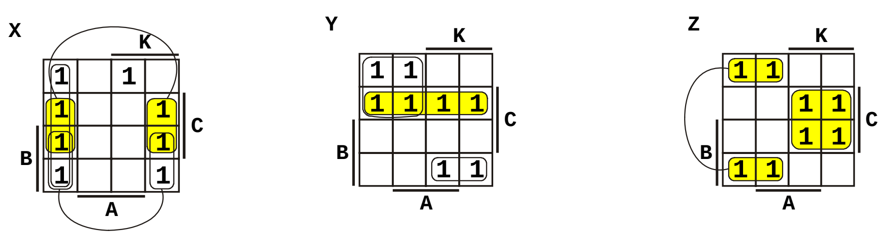
Shembull

Qarku digjital përmes të cilit gjendet **1**-komplementi dhe **2**-komplementi i numrave binarë **3**-bitësh $(ABC)_2$. Për komandimin e punës së qarkut shfrytëzohet hyrja **K** në qark, kështu:

$$K = \begin{cases} 0 & \textit{gjendet } \mathbf{1}\textit{ - komplementi} \\ 1 & \textit{gjendet } \mathbf{2}\textit{ - komplementi} \end{cases}$$

kurse derdhja e cila paraqitet gjatë komplementimit merret në daljen **d** të qarkut.

K	A	B	C	X	Y	Z	d	
0	0	0	0	1	1	1	0	
0	0	0	1	1	1	0	0	
0	0	1	0	1	0	1	0	
0	0	1	1	1	0	0	0	$\overline{1}$
0	1	0	0	0	1	1	0	
0	1	0	1	0	1	0	0	
0	1	1	0	0	0	1	0	
0	1	1	1	0	0	0	0	
1	0	0	0	0	0	0	1	
1	0	0	1	1	1	1	0	
1	0	1	0	1	1	0	0	
1	0	1	1	1	0	1	0	$\overline{2}$
1	1	0	0	1	0	0	0	
1	1	0	1	0	1	1	0	
1	1	1	0	0	1	0	0	
1	1	1	1	0	0	1	0	



$$X = \overline{K}\,\overline{A} + \overline{A}\,C + \overline{A}\,B + K\,A\,\overline{B}\,\overline{C}$$

$$Y = \overline{K}\,\overline{B} + \overline{B}\,C + K\,B\,\overline{C}$$

$$Z = K\,C + \overline{K}\,\overline{C}$$

$$d = K\,\overline{A}\,\overline{B}\,\overline{C}$$

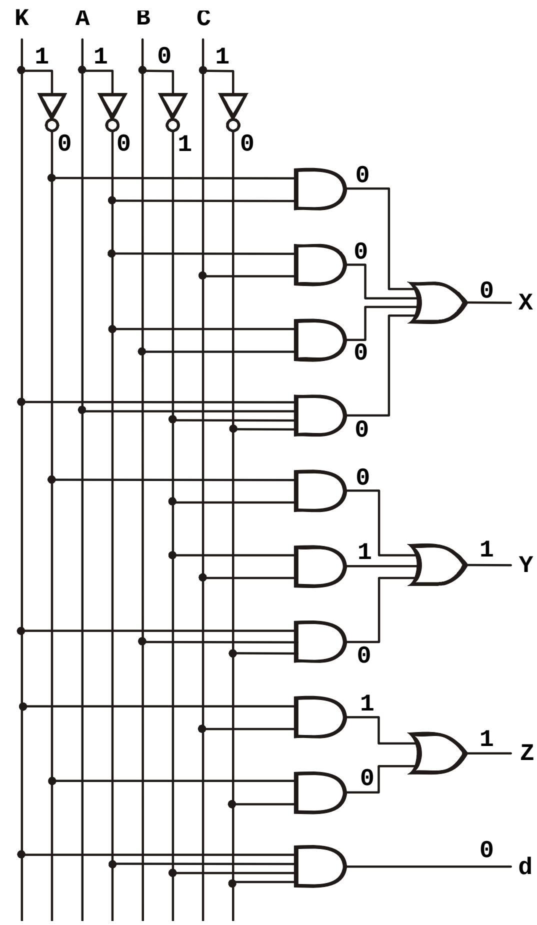


Fig.14.4 Qarku për gjetjen e 1 dhe 2-komplementit

Për ta pasur më të qartë mënyrën e funksionimit të qarkut të dhënë, p.sh., nëse kërkohet të gjendet **2**-komplementi i numrit **101**, në hyrjen **K** të qarkut duhet të aplikohet vlera **1**. Si rezultat, në dalje të qarkut do të fitohet numri **011**, kurse vlera e derdhjes **d** është **0**.

Komplementuesi BCD

Numrat të cilët paraqiten në kodet **BCD**, në fakt janë numra decimalë të shkruar përmes fjalëve kodike binare. Prandaj, komplementet e fjalëve kodike të kodeve **BCD**, përkatësisht **1**-komplementi dhe **2**-komplementi, gjenden duke i gjetur **10**-komplementin dhe **9**-komplementin e ekuivalentëve decimalë përkatës.

Shembull

Qarku logjik përmes të cilit për fjalët kodike $(\mathbf{ABCD})_2$ të kodit **NBCD**, gjenden **1**-komplementet përkatës $(\mathbf{RSTV})_2$.

Këtu, për gjetjene e **1**-komplementit të kërkuar, për çdo fjalë kodike merret ekuivalenti decimal përkatës dhe pastaj gjendet **9**-komplementi i saj, duke e shfrytëzuar shprehjen:

$$N_{\bar{9}} = 10^m - 10^{-n} - N$$

ku janë:

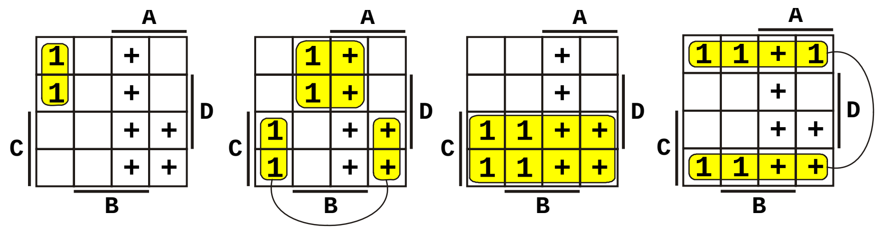
N - ekuivalenti decimal i fjalës kodike
m - numri i shifrave para pikës dhjetore
n - numri i shifrave pas pikës dhjetore.

Kështu, p.sh., për fjalën kodike **0101**, meqë **m=1** dhe **n=0**, komplementi i kërkuar për ekuivalentin decimal përkatës **N=5** është:

$$N_{\bar{9}} = 10^1 - 10^0 - 5 = 4$$

të cilit i përgjigjet ekuivalenti binar **0100**. Duke shkuar në këtë rrugë për të gjitha fjalët kodike të kodit **NBCD**, është plotësuar tabela e dhënë në vijim, janë nxjerrur shprehjet e funksioneve logjike përkatëse, në bazë të të cilave pastaj është vizatuar edhe qarku logjik i komplementuesit në fjalë.

	A	B	C	D	R	S	T	V
0	0	0	0	0	1	0	0	1
1	0	0	0	1	1	0	0	0
2	0	0	1	0	0	1	1	1
3	0	0	1	1	0	1	1	0
4	0	1	0	0	0	1	0	1
5	0	1	0	1	0	1	0	0
6	0	1	1	0	0	0	1	1
7	0	1	1	1	0	0	1	0
8	1	0	0	0	0	0	0	1
9	1	0	0	1	0	0	0	0



$R = \overline{A}\,\overline{B}\,\overline{C}$ $S = B\overline{C} + \overline{B}C = B \oplus C$ $T = C$ $V = \overline{D}$

Nga shprehjet e fituara për funksionet logjike, mund të vizatohet qarku logjik i komplementuesit **NBCD**, i dhënë në Fig.14.5.

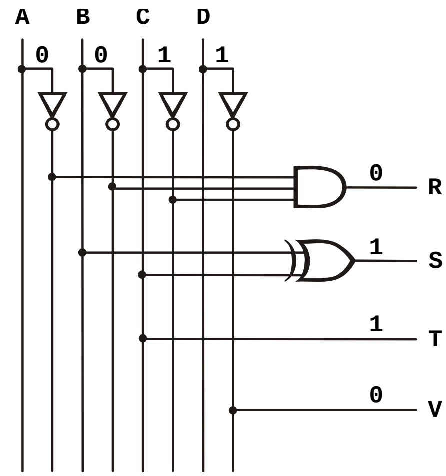


Fig.14.5 Qarku për gjetjen e 1-komplementit të fjalëve kodike të kodit NBCD

Për ta testuar qarkun logjik të komplementuesit, le të marrim se në hyrje të tij aplikohet fjala kodike **0011**. Në daljet e qarkut, si rezultat do të fitohet fjala kodike e komplementit përkatës **0110**.

Plotësisht njëlloj mund të sintetizohet edhe qarku logjik për gjetjen e **2**-komplementit të fjalëve kodike të kodit **NBCD**.

Shembull

Qarku logjik përmes së cilit gjindet **1**-komplementi dhe **2**-komplementi i fjalëve kodike **ABCD** të kodit **NBCD**. Për komandimin e punës së qarkut, përdoret hyrja e veçantë **K**, ashtu që për **K=0** gjendet **1**-komplementi, kurse për **K=1** në dalje të qarkut merret **2**-komplementi i fjalës kodike e cila aplikohet në hyrje të tij.

m	f	K	A	B	C	D	R	S	T	V	
0	0	0	0	0	0	0	1	0	0	1	
1	1	0	0	0	0	1	1	0	0	0	
2	2	0	0	0	1	0	0	1	1	1	
3	3	0	0	0	1	1	0	1	1	0	
4	4	0	0	1	0	0	0	1	0	1	
5	5	0	0	1	0	1	0	1	0	0	$\bar{1}$
6	6	0	0	1	1	0	0	0	1	1	
7	7	0	0	1	1	1	0	0	1	0	
8	8	0	1	0	0	0	0	0	0	1	
9	9	0	1	0	0	1	0	0	0	0	
10		0	1	0	1	0	+	+	+	+	
11		0	1	0	1	1	+	+	+	+	
12		0	1	1	0	0	+	+	+	+	
13		0	1	1	0	1	+	+	+	+	
14		0	1	1	1	0	+	+	+	+	
15		0	1	1	1	1	+	+	+	+	
16	0	1	0	0	0	0	0	0	0	0	
17	1	1	0	0	0	1	1	0	0	1	
18	2	1	0	0	1	0	1	0	0	0	
19	3	1	0	0	1	1	0	1	1	1	
20	4	1	0	1	0	0	0	1	1	0	$\bar{2}$
21	5	1	0	1	0	1	0	1	0	1	
22	6	1	0	1	1	0	0	1	0	0	
23	7	1	0	1	1	1	0	0	1	1	
24	8	1	1	0	0	0	0	0	1	0	
25	9	1	1	0	0	1	0	0	0	1	
26		1	1	0	1	0	+	+	+	+	
27		1	1	0	1	1	+	+	+	+	
28		1	1	1	0	0	+	+	+	+	
29		1	1	1	0	1	+	+	+	+	
30		1	1	1	1	0	+	+	+	+	
31		1	1	1	1	1	+	+	+	+	

$$\mathbf{R} = \sum \mathbf{m}^1(\mathbf{0-1, 17-18})$$
$$\mathbf{S} = \sum \mathbf{m}^1(\mathbf{2-5, 19-22})$$
$$\mathbf{T} = \sum \mathbf{m}^1(\mathbf{2-3, 6-7, 19-20, 23-24})$$
$$\mathbf{V} = \sum \mathbf{m}^1(\mathbf{0, 2, 4, 6, 8, 17, 19, 21, 23, 25})$$

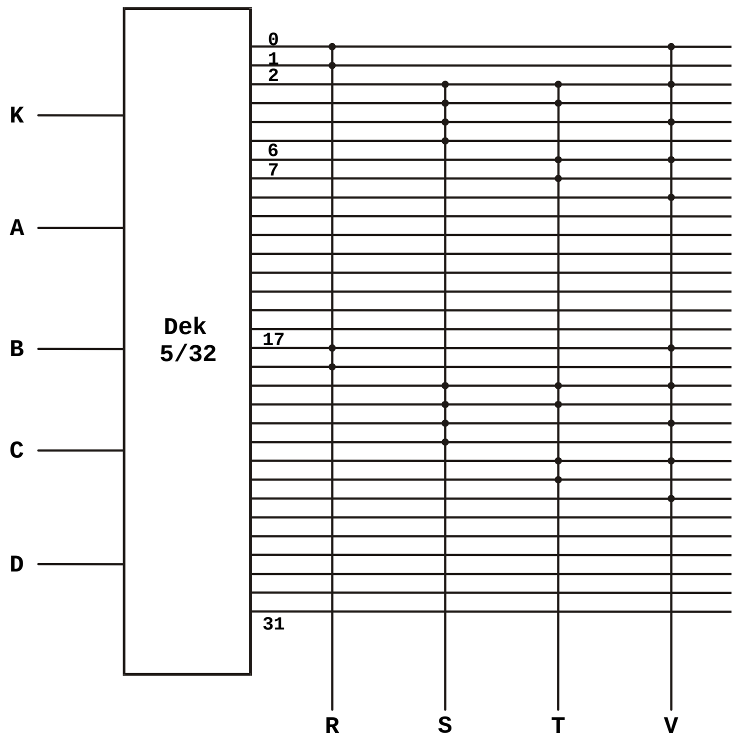


Fig.14.6 Komplementuesi për gjetjen e 1 dhe 2-komplementit të fjalëve kodike të kodit NBCD

Qarku logjik i komplementuesit është realizuar duke e shfrytëzuar një dekodues **5/32** për shkak se shprehjet e funksioneve dalëse nga qarku janë më të komplikuara. Për ta parë mënyrën e funksionimit të qarkut, le të marrim, p.sh., se në hyrje të tij aplikohet fjala kodike **0110**. Për **K=1**, në dalje të qarkut do të merret fjala kodike **0100**, e cila i përgjigjet **2**-komplementit të fjalës kodike të aplikuar në hyrje të tij. Kjo shihet edhe në qark, sepse për kombinimin hyrës **10110**, sinjali binar me vlerë **1** paraqitet vetëm në daljen e **22**-të të dekoduesit, përkatësisht sinjalet në krejt daljet tjera të dekoduesit kanë vlerë binare **0**.

15 Qarqet aritmetikore

Te pajisjet e ndryshme digjitale, për kryerjen e llogaritjeve të shumta përdoren qarqe përkatëse aritmetikore. Këtu do të përmenden disa nga qarqet elementare aritmetikore të realizuara si qarqe kombinuese.

Mbledhësi

Qarku përmes të cilit kryhet mbledhja e numrave binarë quhet *mbledhës* (ang. adder). Si module elementare mbledhëse të cilat përdoren në praktikë janë *gjysmëmbledhësi* dhe *mbledhësi i plotë*, përmes të cilëve mblidhen dy shifra binare, duke mos e marrur, ose duke e marrur parasyshë bartjen nga niveli paraprak.

Gjysmëmbledhësi

Qarku përmes të cilit mblidhen dy shifra binare njihet si *gjysmëmbledhës* (ang. half-adder). Në dy hyrjet e këtij qarku aplikohen shifrat të cilat mblidhen, kurse në daljet e tij merret shuma dhe vlera e bartjes.

Tabela e kombinimeve përmes së cilës përshkruhet puna e gjysmëmbledhësit është:

X	Y	s	b
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

ku janë:

X, **Y** - numrat që mblidhen
s - shuma
b - bartja.

Nga tabela e dhënë, përmes mintermave mund të shkruhen shprehjet e funksioneve dalëse nga gjysmëmbledhësi:

$$\begin{aligned} s &= \overline{X}Y + X\overline{Y} \\ &= X \oplus Y \qquad (15.1) \\ b &= XY \end{aligned}$$

si dhe të vizatohet qarku logjik përkatës i cili është dhënë në Fig.15.1.

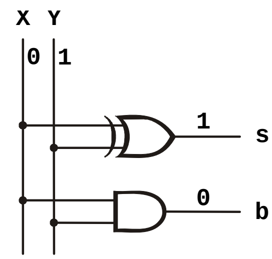


Fig.15.1 Gjysmëmbledhësi

Funksionimin e qarkut logjik të gjysmëmbledhësit mund ta testojmë nëse në hyrje të tij aplikohen vlerat e mundshme hyrëse, ashtu siç janë dhënë në tabelën përkatëse të kombinimeve. P.sh., nëse si vlera hyrëse merren vlerat **0** dhe **1**, vlerat dalëse për shumën dhe bartjen do të jenë **1** dhe **0** - përkatësisht.

Në qarqe të ndryshme, gjysmëmbledhësi skematikisht mund të paraqitet si në Fig.15.2.

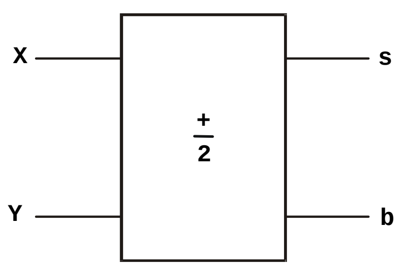


Fig.15.2 Paraqitja skematike e gjysmëmbledhësit

Mbledhësi i plotë

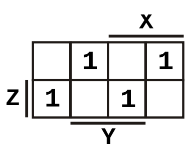
Edhe pse në dalje të gjysmëmbledhësit merret vlera e bartjes e cila paraqitet gjatë mbledhjes së dy shifrave binare, ai nuk mund të përdoret për krijimin e qarkut për mbledhje të numrave shumëbitësh. Qarku digjital i cili gjatë mbledhjes së dy shifrave binare e merr parasysh edhe *bartjen hyrëse*, njihet si *mbledhës i plotë* (ang. full-adder). Tabela e kombinimeve për funksionet dalëse nga mbledhësi i plotë është:

X	Y	Z	s	b
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

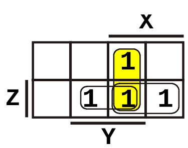
ku janë:

X, Y - numrat që mblidhen
Z - bartja hyrëse
s - shuma
b - bartja dalëse

Nga tabela mund të nxirren shprehjet minimale të funksioneve dalëse **s** e **b** dhe pastaj në bazë të tyre të vizatohet qarku logjik i mbledhësit të plotë, i cili është dhënë në Fig.15.3.



$$s = \overline{X}\,\overline{Y}Z + \overline{X}Y\overline{Z} + X\overline{Y}\,\overline{Z} + XYZ \qquad \textbf{(15.2a)}$$



$$b = XY + XZ + YZ \qquad \textbf{(15.2b)}$$

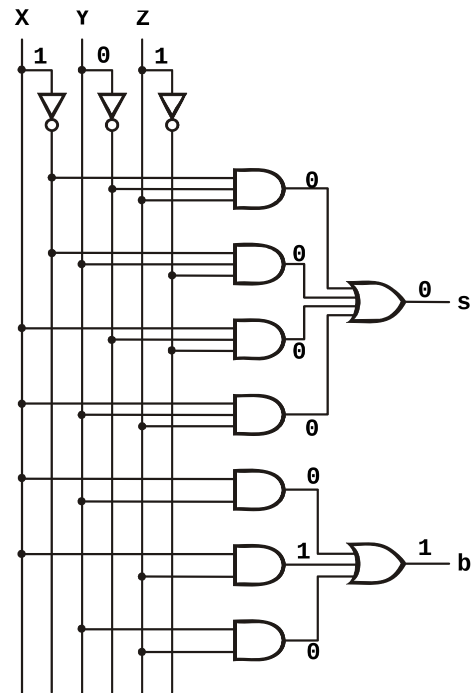


Fig.15.3 Mbledhësi i plotë

Nëse në tri hyrjet e qarkut të vizatuar më sipër aplikohen vlerat **1**, **0** dhe **1**, në daljen **s** të shumës do të fitohet vlera **0**, kurse vlera e bartjes **b** në daljen përkatëse do të jetë **1**. Vlerat në të dy daljet e qarkut do të jenë **1**, nëse vlerat e aplikuara në tri hyrjet e tij janë **1**, gjë që shihet edhe në tabelën e kombinimeve përmes së cilës përshkruhet funksionimi i mbledhësit të plotë.

Shprehjet e dhëna më sipër mund të shkruhen në një formë të tillë që gjatë realizimit të qarkut logjik të mbledhësit të plotë të përdoren edhe elemente logjike *ekskluziv*-**OSE**, ashtu siç shihet në Fig.15.4.

$$
\begin{aligned}
\mathbf{s} &= \overline{X}\,\overline{Y}Z + \overline{X}Y\overline{Z} + X\overline{Y}\,\overline{Z} + XYZ \\
&= (\overline{X}\,\overline{Y} + XY)Z + (\overline{X}Y + X\overline{Y})\overline{Z} \\
&= (X \otimes Y)Z + (X \oplus Y)\overline{Z} \\
&= (\overline{X \oplus Y})Z + (X \oplus Y)\overline{Z} \\
&= (X \oplus Y) \oplus Z \\
\mathbf{b} &= \overline{X}YZ + X\overline{Y}Z + XY\overline{Z} + XYZ \\
&= (\overline{X}Y + X\overline{Y})Z + XY(\overline{Z} + Z) \\
&= (X \oplus Y)Z + XY
\end{aligned}
\qquad (15.3)
$$

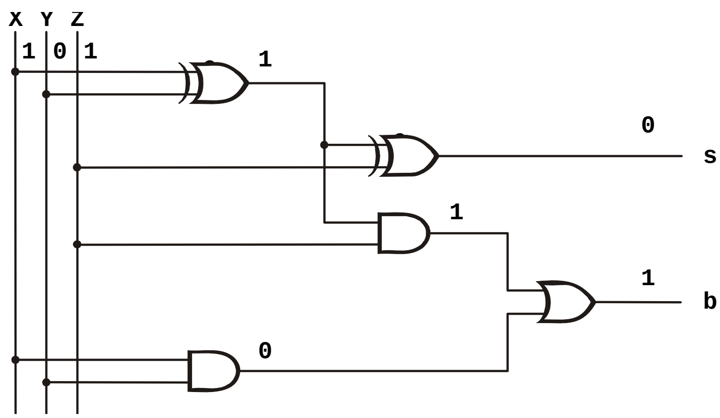


***Fig.15.4 Realizimi i mbledhësit të plotë duke shfrytëzuar edhe elemente logjike* EX-OSE**

Nëse edhe në hyrje të këtij versioni të qarku logjik të mbledhësit të plotë zbatohen vlerat **1**, **0** dhe **1**, rezultati dalës nga qarku do të jetë i njëjtë me atë që fitohet te versionit i mbledhësit të plotë që shihet në Fig.15.3.

Për paraqitjen skematike të mbledhësit të plotë brenda qarqeve logjike të ndryshme, mund të përdoret simboli i dhënë në Fig.15.5.

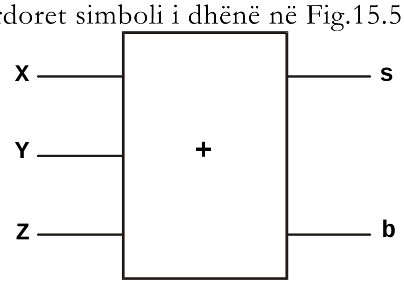


Fig.15.5 Paraqitja skematike e mbledhësit të plotë

Realizimi përmes gjysmëmbledhësve

Duke e pasur parasysh qarkun e gjysmëmbledhësit, si dhe shprehjet përkatëse me të cilat përshkruhet ky qark, mbledhësi i plotë mund të realizohet edhe përmes dy gjysmëmbledhësve, ashtu siç është treguar në Fig.15.6.

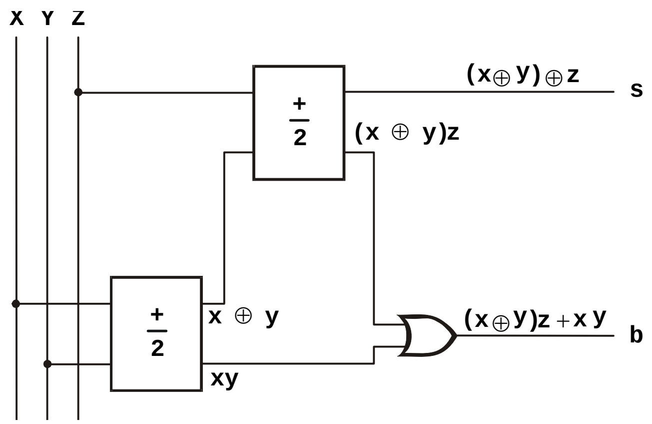


Fig.15.6 Realizimi i mbledhësit të plotë përmes gjysmëmbledhësve

Mbledhësi serik

Për realizimin e mbledhësit **n**-bitësh mund të përdoren **n**-mbledhës të plotë, të lidhur ashtu që të formojnë një seri mbledhësish. Qarku i cili formohet në këtë mënyrë paraqetë *mbledhës serik* **n**-bitësh.

Shembull	Qarku logjik i mbledhësit serik **4**-bitësh, përmes të cilit mblidhen numrat binarë: $X=x_3x_2x_1x_0$ $Y=y_3y_2y_1y_0$ i realizuar përmes **4** mbledhësve të plotë.

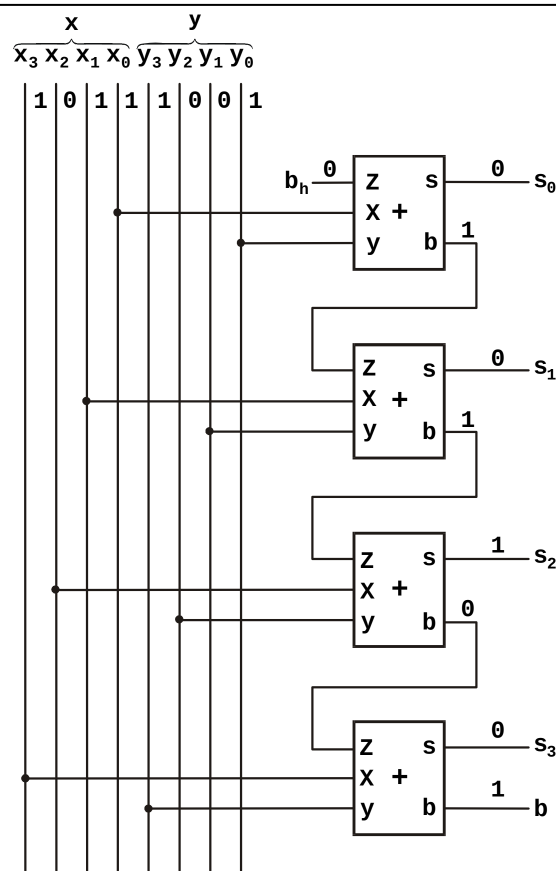


Fig.15.7 Mbledhësi serik 4-bitësh

Për ta pasur më të qartë funksionimin e qarkut logjik të mbledhësit **4**-bitësh, të dhënë më sipër, le të marrim, p.sh., se në hyrjen e tij zbatohen vlerat numerike binare:

$$\mathbf{X} = \mathbf{1011}$$

$$\mathbf{Y} = \mathbf{1001}$$

kurse bartja fillestare hyrëse $\mathbf{b_h}$ në nivelin e parë të mbledhësit serik merret **Z=0**. Rezultati i mbledhjes në dalje të qarkut do të jetë **10100**, gjë që fitohet edhe nëse mbledhja kryhet me laps:

```
1011   Bartja
 1011
+1001
-----
10100  Shuma
```

Bartja te mbledhësi serik i cili është dhënë në Fig.15.7 do të shkaktoi një vonesë të konsiderueshme kohore, proporcionale me numrin e mbledhësve, sepse ajo llogaritet në mënyrë serike.

Për llogaritjen e bartjes mund të realizohet edhe një qark i veçant, ashtu siç është dhënë në vijim.

Shembull

Qarku logjik për *llogaritjen serike të bartjes*, te mbledhësi **4**-bitësh, nëse në hyrje të tij aplikohen numrat:

$$X=x_3x_2x_1x_0$$
$$Y=y_3y_2y_1y_0$$

Duke u mbështetur në ekuacionin **(15.2b)**, për llogaritjen serike të bartjes, te mbledhësi **4**-bitësh i dhënë më sipër, mund të nxirren shprehjet:

$$\begin{aligned} b_0 &= x_0y_0 + (x_0 + y_0)b_h \\ b_1 &= x_1y_1 + (x_1 + y_1)b_0 \\ b_2 &= x_2y_2 + (x_2 + y_2)b_1 \\ b_3 &= x_3y_3 + (x_3 + y_3)b_2 \end{aligned} \qquad \textbf{(15.4)}$$

ku me $\mathbf{b_h}$ është shënuar *bartaja hyrëse* fillestare.

Qarku logjik për llogaritjen serike të bartjes te mbledhësi binar **4**-bitësh, i realizuar në bazë të shprehjeve **(15.4)**, është dhënë në Fig.15.8.

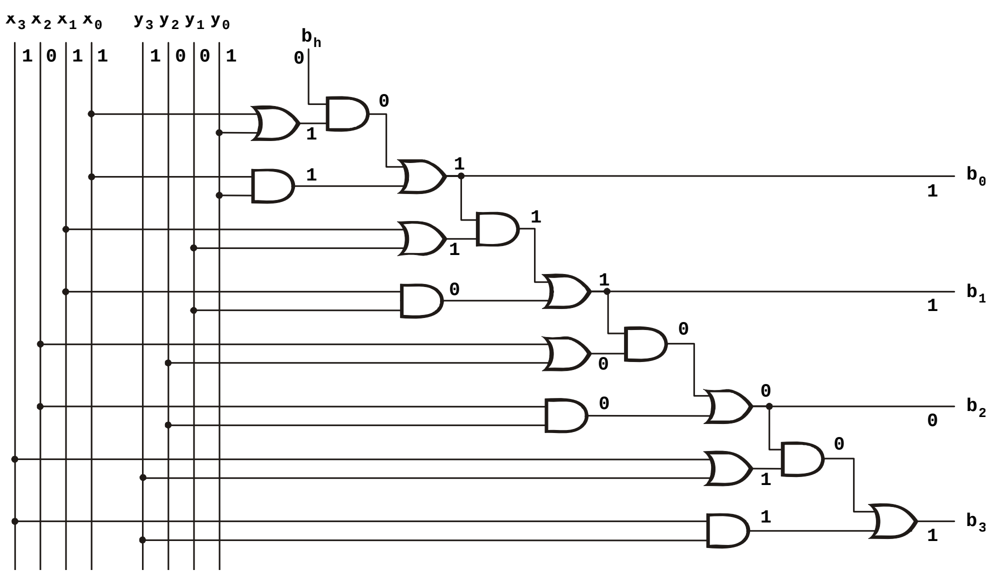


Fig.15.8 Qarku për llogaritjen serike të bartjes te mbledhësi binar 4-bitësh

Nëse në hyrje të qarkut për llogaritje serike të bartjes aplikohen vlerat nga shembulli paraprak:

X=1011
Y=1001

dhe bartja hyrëse fillestare merret $b_h=0$, në dalje të qarkut, për vargun e bartjeve $b_3b_2b_1b_0$, do të fitohen vlerat **1011**, të cilat janë të njëjta me vlerat e fituara në daljet përkatëse të qarkut të mbledhësit serik **4**-bitësh që u dha në Fig.15.7.

Mbledhësi paralelë

Puna e qarkut të mbledhësit mund të shpejtohet shumë, nëse bartjet e nevojshme llogariten njëkohësisht, përmes *qarkut për llogaritje paralele të bartjes*.

Shembull

Qarku logjik për llogaritjen paralele të bartjes, te mbledhësi **4**-bitësh, nëse në hyrje të tij aplikohen vlerat:

$X=x_3x_2x_1x_0$
$Y=y_3y_2y_1y_0$

Nëse në shprehjet **(15.4)**, shënohen me:

$$Q_i = x_i + y_i$$
$$R_i = x_i y_i$$

për **i=0**, **1**, **2** e **3**, shprehjet për llogaritjen e bartjeve e marrin formën:

$$\begin{aligned} b_0 &= R_0 + Q_0 b_h \\ b_1 &= R_1 + Q_1 b_0 \\ b_2 &= R_2 + Q_2 b_1 \\ b_3 &= R_3 + Q_3 b_2 \end{aligned} \qquad (15.5)$$

Duke e zëvendësuar te shprehja për llogaritjen e bartjes b_i, bartjen b_{i-1} me shprehjen përkatëse, për llogaritjen paralele të bartjeve fitohen shprehjet vijuese.

$$
\begin{aligned}
b_0 &= R_0 + Q_0 b_h \\
b_1 &= R_1 + Q_1(R_0 + Q_0 b_h) \\
&= R_1 + Q_1 R_0 + Q_1 Q_0 b_h \\
b_2 &= R_2 + Q_2(R_1 + Q_1 R_0 + Q_1 Q_0 b_h) \\
&= R_2 + Q_2 R_1 + Q_2 Q_1 R_0 + Q_2 Q_1 Q_0 b_h \\
b_3 &= R_3 + Q_3\left(R_2 + Q_2 R_1 + Q_2 Q_1 R_0 + Q_2 Q_1 Q_0 b_h\right) \\
&= R_3 + Q_3 R_2 + Q_3 Q_2 R_1 + Q_3 Q_2 Q_1 R_0 + Q_3 Q_2 Q_1 Q_0 b_h
\end{aligned}
\tag{15.6}
$$

Në bazë të shprehjeve **(15.6)** është vizatuar qarku për llogaritjen paralele të bartjes te mbledhësi binar **4**-bitësh, i cili shihet në Fig.15.9.

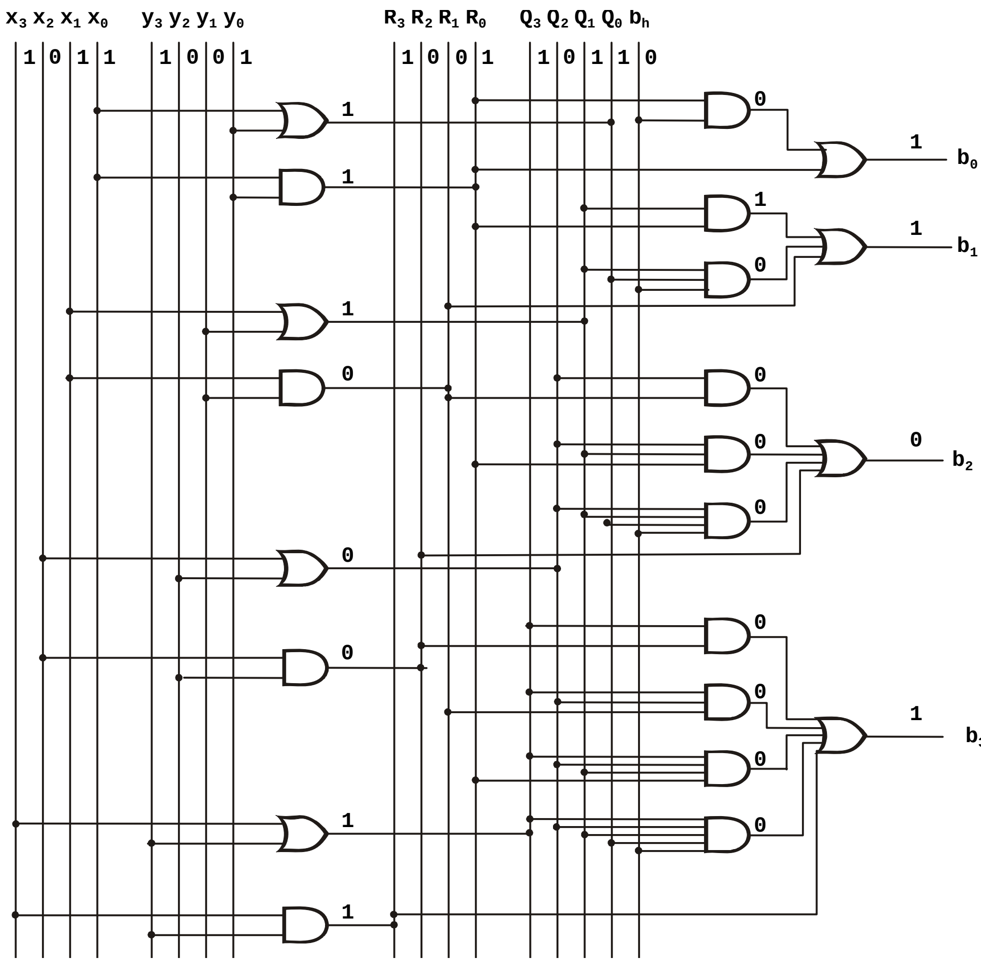


Fig.15.9 Qarku për llogaritjen paralele të bartjes te mbledhësi 4-bitësh

Për ta testuar punën e qarkut të dhënë, le të marrim se në hyrje të tij aplikohen vlerat e njëjta si edhe te qarku për llogaritje serike të bartjes:

X=1011
Y=1001

dhe bartja fillestare hyrëse merret $b_h=0$. Në dalje të qarkut, për vargun e bartjeve $b_3b_2b_1b_0$, do të fitohen vlerat binare **1011**, të cilat janë të njëjta me vargun e vlerave të bartjeve që fitohen në dalje të qarkut për llogaritje serike të bartjeve (Fig.15.8).

Nëse dihen vlerat e bartjeve, mbledhësi **n**-bitësh mund të vizatohet duke u mbështetur në funksionin logjik të shumës së dhënë me shprehjet **(15.3)**, i cili në formë të përgjithshme shkruhet kështu:

$$s_i = (x_i \oplus y_i) \oplus b_{i-1} \tag{15.7}$$

për $i=0$, **1**, ..., **n**, ku në vend të bartjes b_{i-1}, për $i=0$, në shprehjen e shumës s_0 duhet të figurojë bartja fillestare b_h.

Shembull	Qarku logjik i mbledhësit **4**-bitësh, nëse në hyrje të tij aplikohen vlerat binare: $X=x_3x_2x_1x_0$ $Y=y_3y_2y_1y_0$ Bartjet që nevojiten në nivelet e veçanta të mbledhësit: $B=b_3b_2b_1b_0$ merren nga daljet e qarkut për llogaritjen paralele të bartjeve.

Mbështetur në shprehjen **(15.7)** për funksionin e shumës, mund të vizatohet qarku logjik i mbledhësit paralel **4**-bitësh, ashtu siç shihet në Fig.15.10.

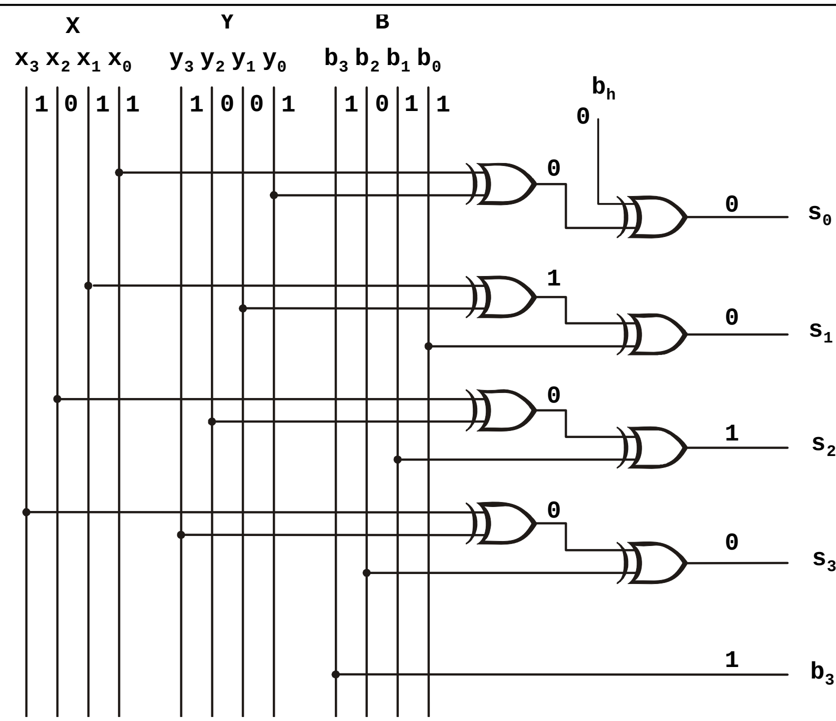


Fig.15.10 Mbledhësi 4-bitësh i cili i shfrytëzon bartjet nga qarku për llogaritje paralele të bartjes

Funksionimi i qarkut logjik të dhënë mund të testohet, nëse në hyrje të tij aplikohen vlerat binare të numrave që mblidhen, si dhe vlerat e bartjeve të llogaritura përmes qarkut për llogaritje paralele të tyre. Nëse merret shembulli i vlerave të shfrytëzuara edhe gjatë testimit të qarqeve të mbledhësve **4**-bitësh të dhënë më sipër:

X=1011
Y=1001

si dhe vlerat e bartjeve **B=1011**, të llogaritura përmes qarkut për llogaritje paralele të bartjeve, në dalje të qarkut do të fitohet shuma **10100**, ku biti i parë i përket bartjes së fundit.

Në praktikë, për realizimin e mbledhësve shumëbitësh, përdoren komponente mbledhëse standarde, siç janë, p.sh., ato për mbledhje të numrave **4**- bitësh, të cilat simbolikisht mund të shënohen si në Fig.15.11.

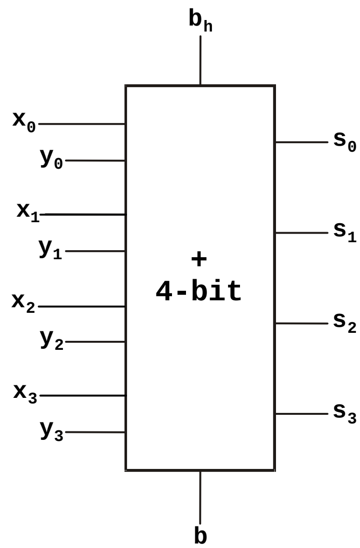


Fig.15.11 Komponenta standarde për mbledhje 4-bitëshe

Shembull	Qarku logjik i mbledhësit **8**-bitësh, i realizuar duke shfrytëzuar dy mbledhës **4**-bitësh.

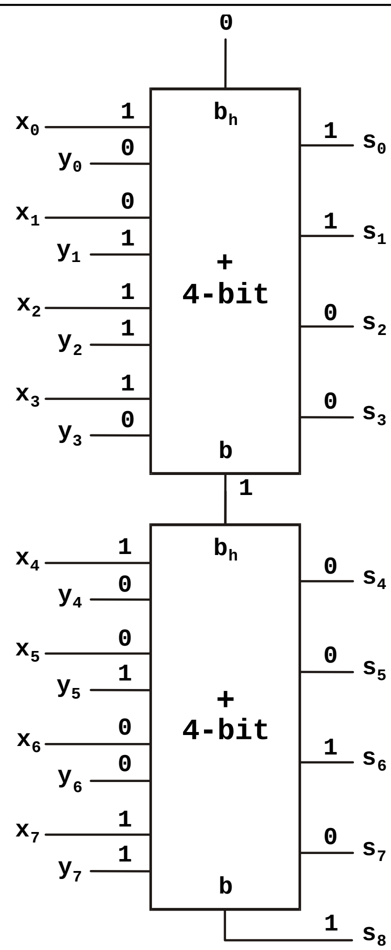


Fig.15.12 Mbledhësi 8-bitësh i realizuar duke shfrytëzuar dy mbledhës 4-bitësh

Puna e qarkut është testuar për vlerat hyrëse:

X = 10011101

Y = 10100110

kurse rezultati i mbledhjes në dalje të qarkut do të jetë:

```
1 111 1   Bartja
 1001 1101
 1010 0110
----------
10100 0011
```

Zbritësi

Për zbritjen e numrave binarë shfrytëzohet qarku logjik i cili quhet *zbritës* (ang. subtractor). Si module elementare zbritëse në praktikë përdoren *gjysmëzbritësi* dhe *zbritësi i plotë*.

Gjysmëzbritësi

Qarku përmes të cilit zbriten dy shifra binare, pa e marrë parasysh huan paraprake, njihet si *gjysmëzbritës* (ang. half-substractor). Puna e gjysmëzbritësit përshkruhet përmes tabelës së kombinimeve:

X	Y	d	h
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

ku janë:

X, **Y** - numrat të cilët zbriten
d - diferenca
h - huaja dalëse.

Shprehjet e funksioneve dalëse nga qarku i gjysmëzbritësit, të nxjerra nga tabela e mësipërme, duken kështu:

$$\begin{aligned} \mathbf{d} &= \overline{\mathbf{X}}\mathbf{Y} + \mathbf{X}\overline{\mathbf{Y}} \\ &= \mathbf{X} \oplus \mathbf{Y} \qquad \mathbf{(15.8)} \\ \mathbf{h} &= \overline{\mathbf{X}}\mathbf{Y} \end{aligned}$$

në bazë të së cilëve mund të vizatohet edhe qarku logjik i gjysmëzbritësit, ashtu siç shihet në Fig.15.13.

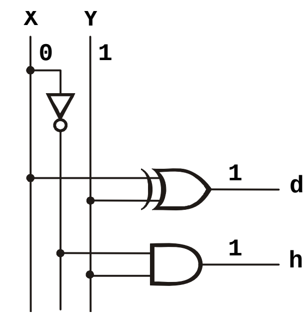


Fig.15.13 Gjysmëzbritësi

Për ta testuar funksionimin e qarkut logjik të gjysmëzbritësit, në hyrje të tij mund të aplikohen vlerat e kombinimeve të mundshme hyrëse. Kështu, p.sh., nëse në hyrjet **X** e **Y** të qarkut si numra që zbriten aplikohen vlerat **0** dhe **1**, të dy vlerat në dalje të qarkut do të jenë **1**, gjë që është e njëjtë me vlerat përkatëse në tabelën e kombinimeve.

Brenda qarqeve të ndryshme, gjysmëzbritësi skematikisht mund të paraqitet si në Fig.15.14.

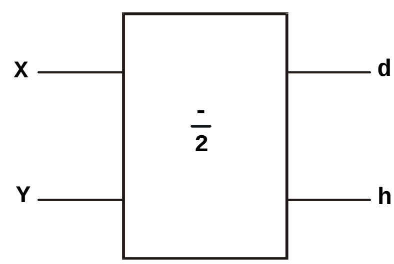


Fig.15.14 Paraqitja skematike e gjysmëzbritësit

Zbritësi i plotë

Për sintetizimin e qarqeve përmes së cilave mund të zbriten numra shumëbitësh, përdoret qarku njëbitësh që njihet si *zbritës i plotë* (ang. full-substractor), tek i cili merret parasysh edhe *huaja hyrëse*.

Tabela e kombinimeve sipas së cilës funksionon zbritësi i plotë, është:

X	Y	Z	d	h
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

ku janë:

X, Y - numrat që zbriten
Z - huaja hyrëse
d - diferenca
h - huaja dalëse

Raporti matematikor mes madhësive hyrëse dhe atyre dalëse, në tabelën e mësipërme, për të gjitha vlerat hyrëse, mund të paraqitet kështu:

$$\underline{\mathbf{X - (Y + Z) + h = d}}$$
$$\mathbf{0 - (0 + 0) + 0 = 0}$$
$$\mathbf{0 - (0 + 1) + 2 = 1}$$
$$\mathbf{0 - (1 + 0) + 2 = 1}$$
$$\mathbf{0 - (1 + 1) + 2 = 0}$$
$$\mathbf{1 - (0 + 0) + 0 = 1}$$
$$\mathbf{1 - (0 + 1) + 0 = 0}$$
$$\mathbf{1 - (1 + 0) + 0 = 0}$$
$$\mathbf{1 - (1 + 1) + 2 = 1}$$

ku me **2** simbolikisht është shënuar huazimi i një "*dhjetësheje*" binare, për t'u realizuar zbritja.

Duke e pasur parasysh tabelën e kombinimeve që u dha më sipër për madhësitë dalëse nga zbritësi i plotë, mund të nxirren shprehjet:

$$
\begin{aligned}
\mathbf{d} &= \overline{X}\,\overline{Y}Z + \overline{X}Y\overline{Z} + X\overline{Y}\,\overline{Z} + XYZ \\
&= (\overline{X}\,\overline{Y} + XY)Z + (\overline{X}Y + X\overline{Y})\overline{Z} \\
&= (X \otimes Y)Z + (X \oplus Y)\overline{Z} \\
&= (\overline{X \oplus Y})Z + (X \oplus Y)\overline{Z} \\
&= (X \oplus Y) \oplus Z \qquad\qquad (15.9) \\
\mathbf{h} &= \overline{X}\,\overline{Y}Z + \overline{X}Y\overline{Z} + \overline{X}YZ + XYZ \\
&= (\overline{X}\,\overline{Y} + XY)Z + (\overline{Z} + Z)\overline{X}Y \\
&= (X \otimes Y)Z + \overline{X}Y \\
&= (\overline{X \oplus Y})Z + \overline{X}Y
\end{aligned}
$$

Qarku logjik i zbritësit të plotë, i vizatuar në bazë të shprehjeve të nxjerra më sipër, duket si në Fig.15.15.

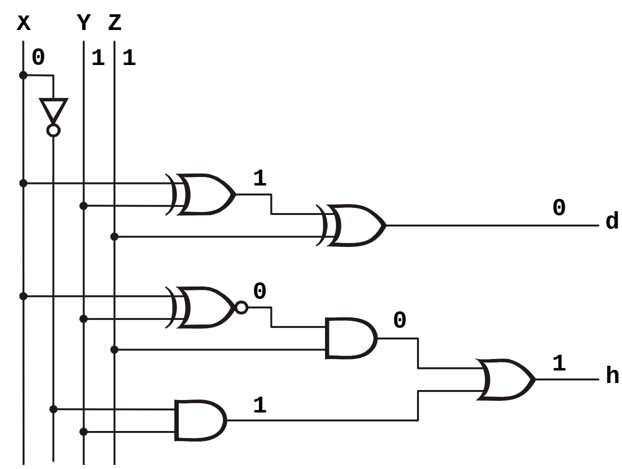


Fig.15.15 Zbritësi i plotë

Për ta testuar qarkun e dhënë të zbritësit të plotë, në tri hyrjet e tij, le t'i aplikojmë, p.sh., vlerat **0**, **1** dhe **1**. Si rezultat, në dy daljet e qarkut do të fitohen vlerat **0** dhe **1**, gjë që përputhet me vlerat përkatëse në tabelë.

Zbritësi i plotë, brenda qarqeve logjike të ndryshme, skematikisht mund të paraqitet si në Fig.15.16.

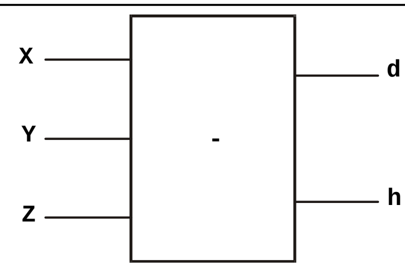


Fig.15.16 Paraqitja skematike e zbritësit të plotë

Realizimi përmes gjysmëzbritësve

Njëlloj si mbledhësi i plotë, edhe zbritësi i plotë mund të realizohet duke shfrytëzuar dy gjysmëzbritës, ashtu siç shihet në Fig.15.17.

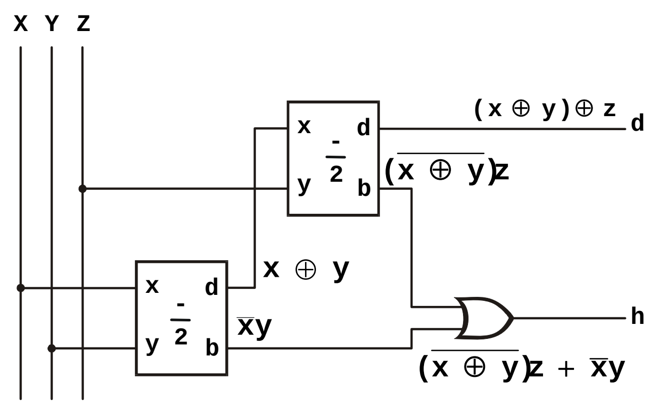


Fig.15.17 Zbritësi i plotë i realizuar përmes dy gjysmëzbritësve

Saktësia e funksionimit të qarkut të dhënë si zbritës i plotë vërtetohet me shprehjet e fituara për funksionet dalëse **d** dhe **h**. Këto shprehje janë të njëjta me ato **(15.9)**, të nxjerra nga tabela e kombinimeve, të cilat e përshkruajnë punën e zbritësit të plotë.

Zbritësi i plotë përmes mbledhësit të plotë

Nëse krahasohen shprehjet **(15.3)** dhe **(15.9)** të funksioneve dalëse nga mbledhësi dhe zbritësi i plotë, shihet se shprehjet e shumës **s** dhe të diferencës

d janë të njëjta. Por, meqë shprehjet për bartjen **b** dhe huanë **h** dallojnë, që mbledhësi i plotë të funksionojë si zbritës i plotë, duhet të invertohet vlera e variablës së parë hyrëse dhe vlera në daljen që i përket shumës **s**. Kështu, qarku përkatës i zbritësit të plotë i realizuar duke e shfrytëzuar mbledhësin e plotë, do të duket si në Fig. 15.18.

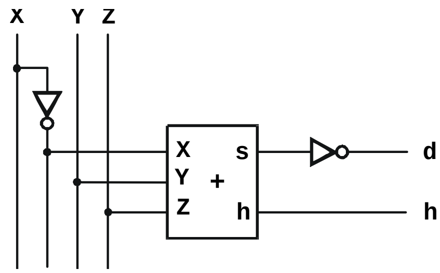


Fig.15.18 Zbritësi i plotë i realizuar përmes mbledhësit të plotë

Për funksionet dalëse nga qarku mund të nxirren shprehjet:

$$
\begin{aligned}
\mathbf{d} &= \overline{(\overline{X} \oplus Y) \oplus Z} \\
&= (\overline{X} \oplus Y) \otimes Z \\
&= (XY + \overline{X}\,\overline{Y})Z + (\overline{X}Y + X\overline{Y})\overline{Z} \\
&= (X \otimes Y)Z + (X \oplus Y)\overline{Z} \\
&= (\overline{X \oplus Y})Z + (X \oplus Y)\overline{Z} \\
&= (X \oplus Y) \oplus Z \\
\mathbf{h} &= (\overline{X} \oplus Y)Z + \overline{X}Y \\
&= (\overline{X}\,\overline{Y} + XY)Z + \overline{X}Y \\
&= (X \otimes Y)Z + \overline{X}Y \\
&= (\overline{X \oplus Y})Z + \overline{X}Y
\end{aligned}
\qquad \textbf{(15.10)}
$$

Nga shprehjet e fituara shihet se ato janë të njëjta me shprehjet **(15.9)**, të cilat u përkasin funksioneve dalëse të zbritësit të plotë, me çka vërtetohet funksionimi si zbritës i plotë i qarkut të dhënë në Fig.15.18

Skematikisht, qarku i zbritësit të plotë i realizuar përmes mbledhësit të plotë, mund të paraqitet si në Fig.15.19.

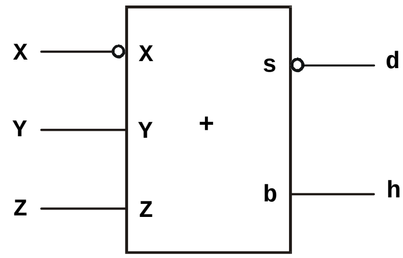


Fig.15.19 Paraqitja skematike e zbritësit të plotë i cili është realizuar përmes mbledhësit të plotë

Mbledhësi/zbritësi

Duke përdorur procedurën e zakonshme të sintezës së qarqeve kombinuese, mund të projektohet një qark kombinues i vetëm, i cili njëkohësisht do të shfrytëzohet si mbledhës dhe zbritës i plotë. Komandimi i punës së qarkut të tillë duhet të bëhet përmes një hyrjeje të veçantë **K**, në të cilën duke i aplikuar dy vlerat e mundshme **0** dhe **1**, qarku do të punojë si mbledhës ose zbritës, p.sh. kështu:

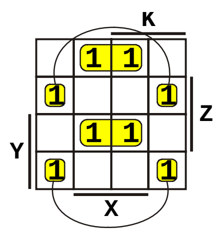
$$\mathbf{K} = \begin{cases} \mathbf{0} & \textit{mbledhës i plotë} \\ \mathbf{1} & \textit{zbritës i plotë} \end{cases}$$

Në vijim është dhënë tabela e kombinimeve përmes së cilës përshkruhet funksionimi i qarkut në fjalë, prej ku pastaj janë nxjerrur edhe shprehjet e funksioneve përkatëse të dy daljeve nga qarku.

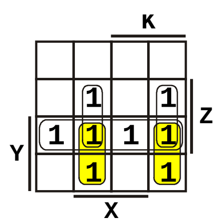
K	X	Y	Z	s/d	b/h
0	0	0	0	0	0
0	0	0	1	1	0
0	0	1	0	1	0
0	0	1	1	0	1
0	1	0	0	1	0
0	1	0	1	0	1
0	1	1	0	0	1
0	1	1	1	1	1
1	0	0	0	0	0
1	0	0	1	1	1
1	0	1	0	1	1
1	0	1	1	0	1
1	1	0	0	1	0
1	1	0	1	0	0
1	1	1	0	0	0
1	1	1	1	1	1

Mbledhës i plotë

Zbritës i plotë



$$\begin{aligned} s/d &= X\overline{Y}\overline{Z} + \overline{X}\overline{Y}Z + XYZ + \overline{X}Y\overline{Z} \\ &= (X\overline{Y} + \overline{X}Y)\overline{Z} + (\overline{X}\overline{Y} + XY)Z \\ &= (X \oplus Y)\overline{Z} + (X \otimes Y)Z \\ &= (X \oplus Y) \oplus Z \end{aligned}$$



$$\begin{aligned} b/h &= YZ + \overline{K}XZ + K\overline{X}Z + \overline{K}XY + K\overline{X}Y \\ &= YZ + (\overline{K}X + K\overline{X})Z + (\overline{K}X + K\overline{X})Y \\ &= YZ + (\overline{K}X + K\overline{X})(Y + Z) \\ &= YZ + (K \oplus X)(Y + Z) \end{aligned}$$

Në bazë të shprehjeve të fituara, është vizatuar qarku logjik i mbledhësit/zbritësit të plotë, ashtu siç shihet në Fig.15.20. Funksionimi i qarkut si mbledhës (**K=0**) dhe si zbritës (**K=1**) është testuar me dy grupe vlerash hyrëse, **0101** dhe **1011**, për të cilat në dalje të tij janë fituar vlerat **sb=01** dhe **dh=01**, që përputhen me vlerat përkatëse në tabelën e kombinimeve.

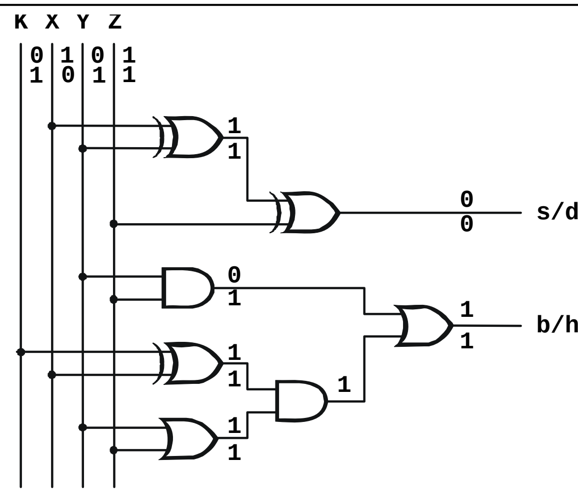


Fig.15.20 Mbledhësi/zbritësi i plotë

Mbledhësi NBCD

Mbledhja e numrave të cilët shkruhen në kodin **NBCD** kërkon sintetizimin e mbledhësit adekuat, mbështetur në principet e mbledhjes në këtë kod. Por, meqë numrat në kodin **NBCD** paraqiten me fjalë kodike **4**-bitëshe, për mbledhjen e dy fjalëve kodike mbledhësi përkatës duhet të ketë **9** hyrje, përkatësisht **2** herë nga **4** hyrje, për numrat që mblidhen, dhe **1** hyrje për bartjen hyrëse, si dhe **5** dalje.

Sintetizimi i qarkut logjik të mbledhësit **NBCD** nuk mund të bëhet duke e shfrytëzuar procedurën e zakonshme, sepse tabela përkatëse e kombinimeve përmban $2^9=512$ kombinime. Për gjetjen e shprehjeve të funksioneve dalëse nga qarku, duhet të shfrytëzohet metoda e minimizimit tabelar përmes kompjuterit. Por, këtu do të përdoret procedura e sintetizimit të qarkut, duke e copëtuar atë në pjesë më të vogla, të cilat në fund komponohen në një tërësi, për ta formuar qarkun e dëshiruar.

Nëse analizohet metoda e mbledhjes në kodin **NBCD**, e cila është shpjeguar në kapitullin mbi kodet, qarku i mbledhësit **NBCD** mund të copëtohet në **3** pjesë funksionale:

- mbledhja binare e dy fjalëve kodike
- gjenerimi i sinjalit binar për korrigjimin e fjalëve kodike
- korrigjimi i fjalëve kodike duke ua shtuar vlerën **0110**.

Nëse fjalët kodike të cilat mblidhen shënohen me **X=$x_8x_4x_2x_1$** dhe **Y=$y_8y_4y_2y_1$**, pjesa e parë funksionale përmes së cilës kryhet mbledhja binare e tyre duhet të përmbajë **4** mbledhës të plotë, ashtu siç është treguar në qarkun logjik të mbledhësit **NBCD**, të dhënë në Fig.15.21. Në **8** hyrjet e qarkut aplikohen shifrat binare të fjalëve kodike **X** dhe **Y**, kurse në hyrjen e **9**-të - edhe bartja hyrëse **b_h**, nëse qarku lidhet me mbledhës të tjerë **NBCD**, përndryshe vlera në këtë hyrje merret zero.

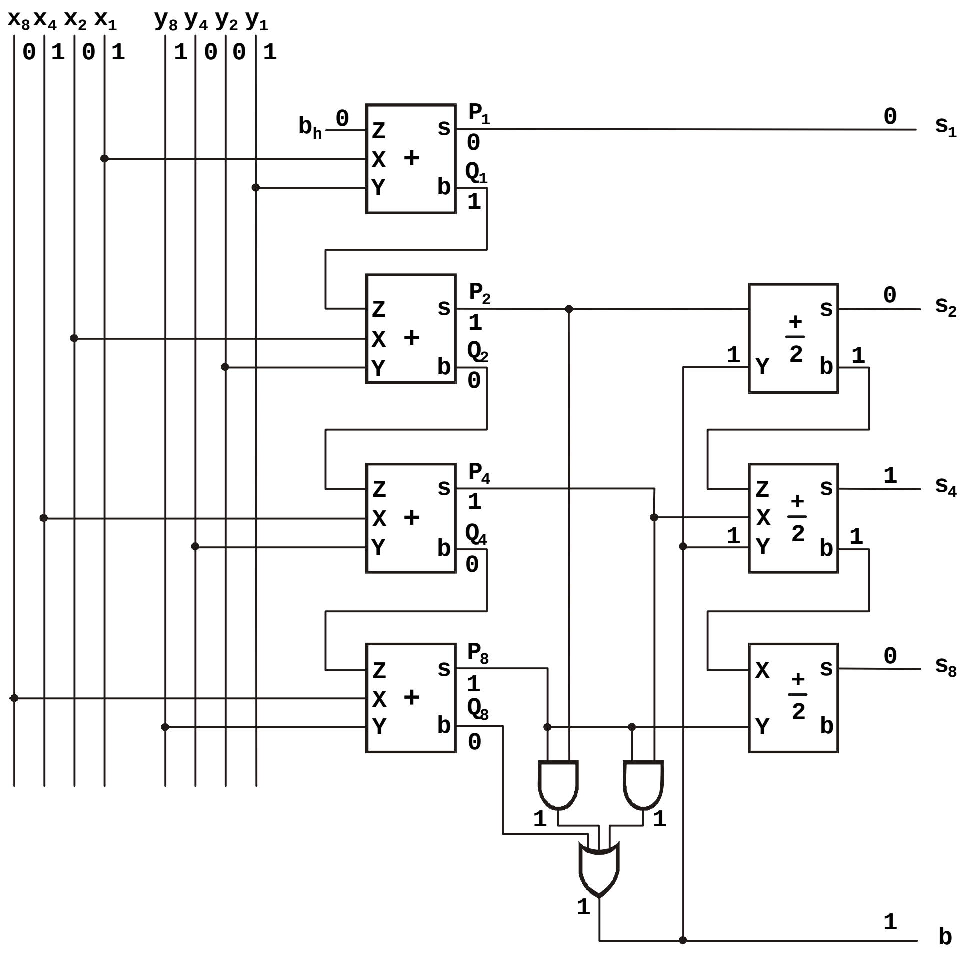


Fig.15.21 Mbledhësi **NBCD**

Shumat $\mathbf{P_1}$, $\mathbf{P_2}$, $\mathbf{P_4}$ dhe $\mathbf{P_8}$, të cilat fitohen në dalje të mbledhësve, korrigjohen nëse paraqesin fjalë kodike të ndaluara, ose kanë bartje, ashtu siç është dhënë në tabelën vijuese.

Q_8	P_8	P_4	P_2	P_1	b	S_8	S_4	S_2	S_1	
0	0	0	0	0	0	0	0	0	0	
0	0	0	0	1	0	0	0	0	1	
0	0	0	1	0	0	0	0	1	0	
0	0	0	1	1	0	0	0	1	1	
0	0	1	0	0	0	0	1	0	0	
0	0	1	0	1	0	0	1	0	1	**(1)**
0	0	1	1	0	0	0	1	1	0	
0	0	1	1	1	0	0	1	1	1	
0	1	0	0	0	0	1	0	0	0	
0	1	0	0	1	0	1	0	0	1	
0	1	0	1	0	1	0	0	0	0	
0	1	0	1	1	1	0	0	0	1	
0	1	1	0	0	1	0	0	1	0	**(2)**
0	1	1	0	1	1	0	0	1	1	
0	1	1	1	0	1	0	1	0	0	
0	1	1	1	1	1	0	1	0	1	
1	0	0	0	0	1	0	1	1	0	
1	0	0	0	1	1	0	1	1	1	**(3)**
1	0	0	1	0	1	1	0	0	0	
1	0	0	1	1	1	1	0	0	1	

Nëse analizohet tabela lidhur me korrigjimin e shumave të veçanta, ajo mund të ndahet në **3** pjesë karakteristike:

(1) - nuk korrigjohen.
(2) - korrigjohen sepse janë fjalë kodike të ndaluara.
(3) - korrigjohen për shkak të bartjes.

Kjo do të thotë se korrigjimin duhet bërë nëse plotësohet njëri nga **3** kushtet:

$$\mathbf{Q_8 = 1}$$
$$\mathbf{P_8P_4 = 1}$$
$$\mathbf{P_8P_2 = 1}$$

përkatësisht nëse vlera e funksionit:

$$\mathbf{b} = \mathbf{Q_8} + \mathbf{P_8P_4} + \mathbf{P_8P_2}$$

është **1**. Kjo vlerë njëkohësisht përcillet edhe në daljen **b** të qarkut dhe e paraqet bartjen e përgjithshme, e cila figuron në rezultatin përfundimtar të mbledhjes.

Për realizimin e funksionit **b**, në pjesën e dytë funksionale të qarkut, janë përdorur dy elemente logjike **DHE** e një element logjik **OSE**.

Korrigjimi i fjalëve kodike, përkatësisht i vlerave $\mathbf{P_8}$, $\mathbf{P_4}$ dhe $\mathbf{P_2}$, bëhet përmes shtimit të ekuivalentit binar **0110** të numrit decimal **6**, duke shfrytëzuar dy gjysmëmbledhës dhe një mbledhës të plotë, në pjesën e tretë funksionale të qarkut. Korrigjimi nuk ka ndikim te vlera $\mathbf{P_1}$, e cila përcillet direkt në shumën dalëse $\mathbf{s_1}$ nga qarku.

Për ta testuar funksionimin e qarkut të mbledhësit **NBCD** të dhënë më sipër, le të marrim se në hyrje të tij aplikohen fjalët kodike:

X=0101
Y=1010

dhe se bartja hyrëse në qark është $\mathbf{b_h=0}$. Në daljet $\mathbf{P_8P_4P_2P_1}$ të mbledhësve të plotë të qarkut, do të merren vlerat **1110**. Meqë vlera në daljen **b** është **1**, fjala kodike që fitohet pas mbledhjes binare duhet të korrigjohet duke ia shtuar numrin binar **0110**, gjë që bëhet përmes një mbledhësi të plotë dhe dy gjysmëmbledhësve, të cilët janë të vendosur në dalje të qarkut. Rezultati i mbledhjes fitohet në daljet **b**, $\mathbf{s_8}$, $\mathbf{s_4}$, $\mathbf{s_2}$ e $\mathbf{s_1}$ dhe për shembullin e marrë të madhësive hyrëse është **10100**, përkatësisht pasi t'i shtohen zerot para, gjendet vlera reale **0001 0100**. Ky rezultat do të fitohet edhe nëse mbledhja kryhet me dorë:

```
     1  Bartja
  0101
+ 1001
------
  1110
```

Meqë fjala kodike e fituar është *fjalë e ndaluar*, ajo duhet të korrigjohet duke ia shtuar vlerën binare **0110**:

```
 111   Bartja
  1110
+ 0110
------
 10100
```

Nëse bëhet prova, duke i marrë ekuivalentët decimalë të fjalëve kodike të cilat mblidhen:

$$
\begin{array}{r}
\mathbf{5} \\
+\mathbf{9} \\
\hline
\mathbf{14}
\end{array}
$$

Numrit **14** i përgjigjet edhe kombinimi i dy fjalëve kodike:

0001 0100

të cilat u fituan më sipër përmes mbledhjes **NBCD**.

Mbledhësi **NBCD** mund të realizohet edhe duke shfrytëzuar **2** mbledhës **4**-bitësh, në vend të mbledhësve të plotë **1**-bitësh, ashtu siç shihet në Fig.15.22.

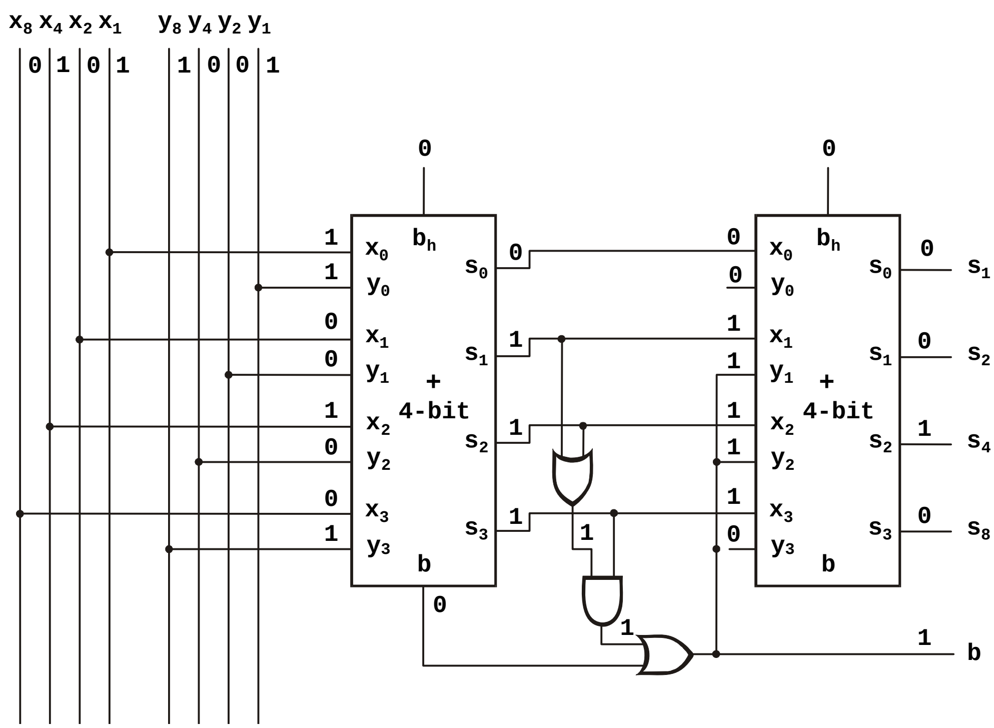


*Fig.15.22 Mbledhësi **NBCD** i realizuar përmes dy mbledhësve të plotë*

Për ta testuar punën e qarkut të dhënë, si mbledhës të dy shifrave decimale, përkatësisht të fjalëve kodike përkatëse, le të marrim, p.sh., se në hyrje të tij aplikohen fjalët kodike **X=0101** dhe **Y=1001**. Në dalje të mbledhësit të parë fitohet vargu i shifrave binare **1110** si dhe bartja **0**, sepse:

```
   1  Bartja
 0101
+1001
-----
 1110
```

Meqë rezultati i fituar nuk bënë pjesë në fjalët kodike të kodit **NBCD**, ai duhet të korrigjohet. Gjenerimi i sinjalit **1** në daljen **b** të qarkut realizohet përmes lidhjes së tri elementeve logjike themelore, ashtu siç shihet në qark, kurse për korrigjim shfrytëzohet mbledhësi i dytë, përmes të cilit fitohet rezultati përfundimtar i mbledhjes së dy fjalëve kodike të kodit **NBCD**.

Qarku logjik i mbledhësit **NBCD** të dhënë në Fig.15.22, paraqet mbledhës njëshifror paralel. Për realizimin e mbledhësit shumëshifror duhet të shfrytëzohen më shumë mbledhës **NBCD** njëshifror, të lidhur ashtu që bartja dalëse nga mbledhësi i numrave me peshë më të ulët të përcillet në hyrje të mbledhësit me peshë më të madhe, plotësisht njëlloj siç lidhen mbledhësit e plotë për të krijuar mbledhës binarë shumëbitësh.

Shembull

Qarku logjik i mbledhësit paralel **3**-shifror, nëse në hyrje të tij aplikohen numrat **X** dhe **Y**, të koduar në kodin **NBCD** dhe bartja hyrëse fillestare në qark është $\mathbf{b_h}$.

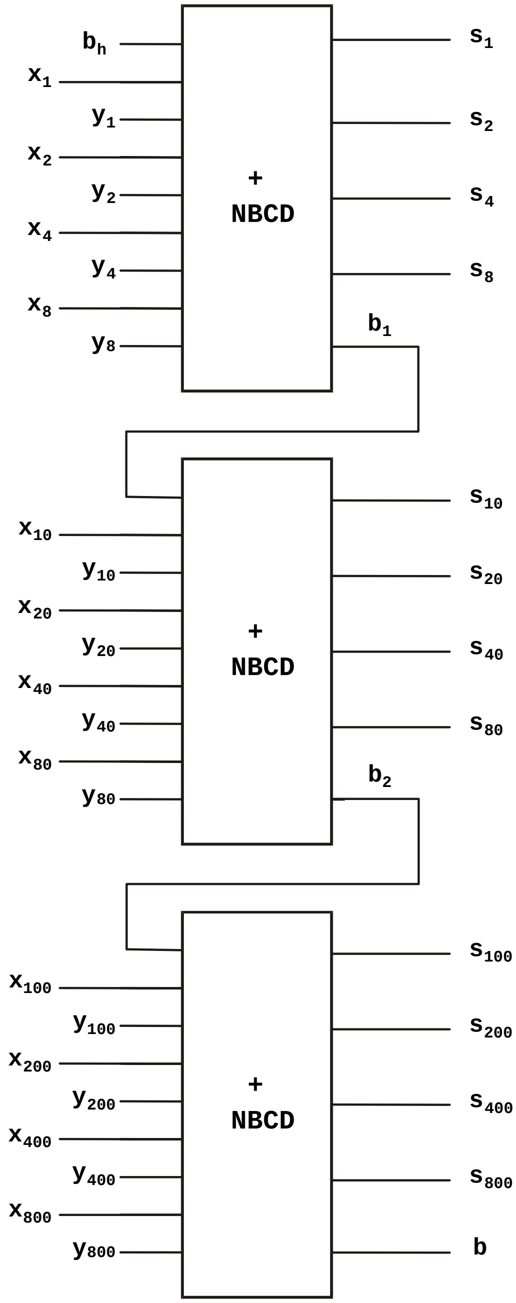


Fig.15.23 Mbledhësi paralel **NBCD** *3-shifror*

Mbledhësi Excess-3

Siç është shpjeguar në kapitullin për kodet, mbledhja në kodin **Excess-3** kryhet në dy hapa:

1. Mblidhen numrat e dhënë, duke shfrytëzuar rregullat e mbledhjes binare.
2. Korrigjohen të gjitha fjalët kodike të cilat fitohen pas mbledhjes, kështu:
- Fjalët kodike që kanë pasur bartje u shtohet numri **0011**.
- Fjalëve kodike që nuk kanë pasur bartje u zbritet numri **0011**, ose u shtohet numri **1101**, ashtu siç është vepruar në pjesën vijuese gjatë sintetizimit të qarkut të mbledhësit **Excess-3**.

Duke e ndjekur procedurën e sintezës së mbledhësit **NBCD**, për mbledhësin **Excess-3**, mbështetur në rregullat e mbledhjes, të dhëna më sipër, mund të vizatohet qarku logjik i cili shihet në figurën vijuese.

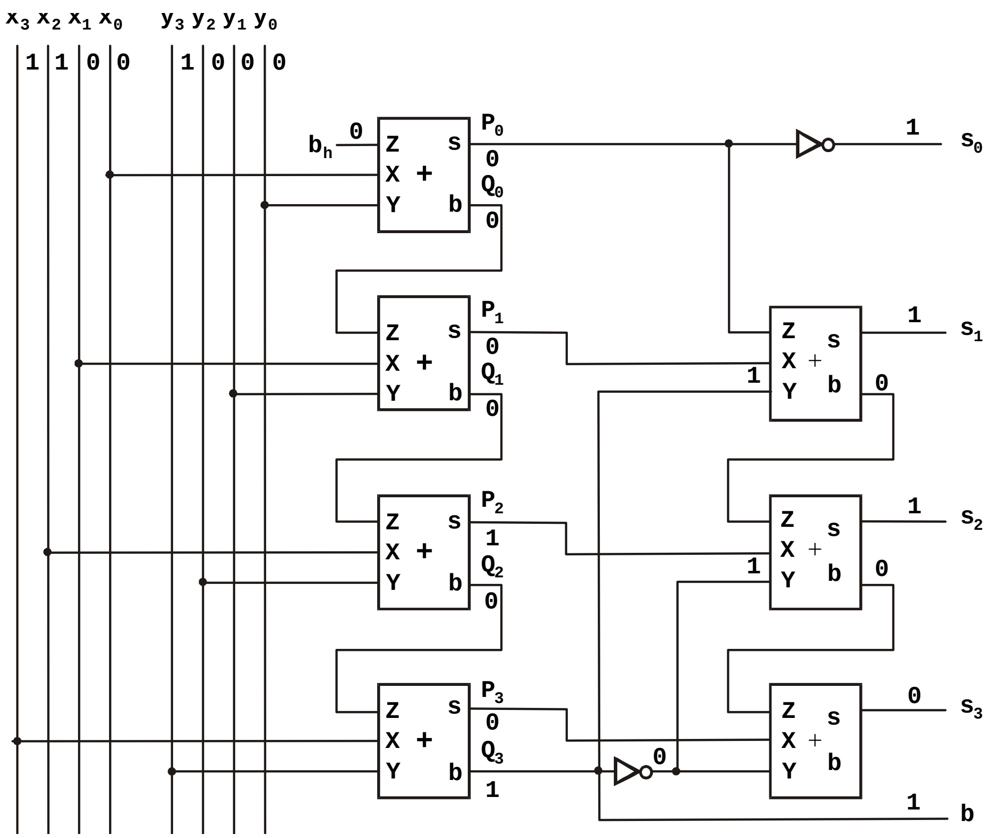


Fig.15.24 Mbledhësi EXcess-3

Bartja që fitohet në mbledhësin e fundit, nga i cili merret shuma s_3, si edhe te mbledhësi **NBCD**, nuk përdoret.

Qarku i dhënë më sipër mund të testohet, nëse në hyrje të tij aplikohen fjalëve kodike të kodit **Excess-3**. Kështu, p.sh., nëse në hyrje të qarkut aplikohen fjalët kodike **1100** dhe **1000**, si rezultat në dalje të tij fitohet vargu i shifrave binare **10111**, përkatësisht fjalët kodike:

0001 0111

pas shtimit të tri zerove para vlerës së bartjes. Ky rezultat do të fitohet edhe nëse mbledhja kryhet me dorë:

```
 1     Bartja
  1100
+ 1000
 -----
 10100
```

Meqë ka bartje, rezultati i fituar korrigjohet duke ia shtuar numrin **0011**:

```
 10100
+ 0011
 -----
 10111
```

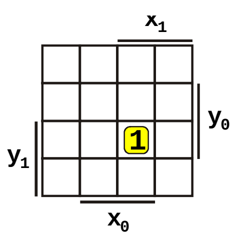
Shumëzuesi

Për shumëzimin e dy shifrave binare përdoret thjesht një element logjik **DHE**. Si qarqe kombinuese të zakonshme mund të realizohen shumëzues **2**-bitësh, ose eventualisht edhe shumëzues **3**-bitësh, sepse qarku përkatës komplikohet shumë.

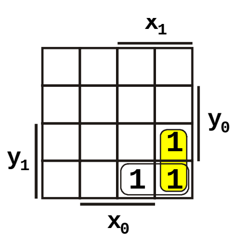
Shembull

Qarku logjik i shumëzuesit **2**-bitësh, nëse në hyrje të tij aplikohen numrat binarë $X=x_1x_0$ dhe $Y=y_1y_0$, kurse rezultati binar i cili merret në dalje të qarkut është $P=p_3p_2p_1p_0$.

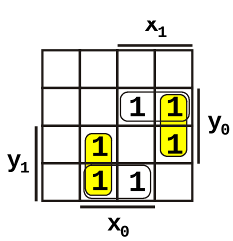
X		Y		P			
x_1	x_0	y_1	y_0	p_3	p_2	p_1	p_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	1	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	1	1	0
1	1	1	1	1	0	0	1



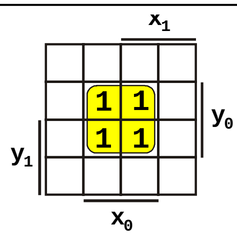
$$p_3 = x_1 x_0 y_1 y_0$$



$$p_2 = x_1 \overline{x}_0 y_1 + x_1 y_1 \overline{y}_0$$



$$p_1 = x_1 \overline{x}_0 y_0 + x_1 \overline{y}_1 y_0 + x_0 y_1 \overline{y}_0 + \overline{x}_1 x_0 y_1$$



$$p_0 = x_0 y_0$$

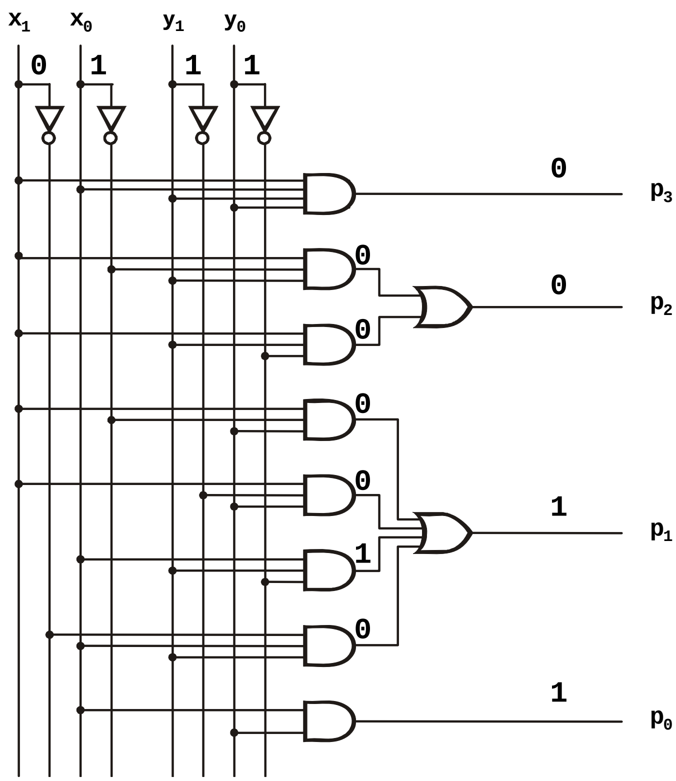


Fig.15.25 Shumëzuesi 2-bitësh

Për ta testuar funksionimin e qarkut logjik të shumëzuesit digjitalë **2**-bitësh të dhënë në Fig.15.25, le të marrim se në hyrje të tij aplikohen vlerat numerike binare **X=01** dhe **Y=11**. Në dalje të qarkut, si rezultat i shumëzimit fitohet numri binar **P=0011**, gjë që shihet edhe në figurë.

Shumëzuesi shumëbitësh

Siç është theksuar edhe në kapitullin mbi sistemet numerike, shumëzuese i dy numrave binarë reduktohet në një numër të caktuar mbledhjesh, gjë që varet nga numri i bitëve të shumëzuesit. Sinteza e qarkut logjik të shumëzuesit shumëbitësh në rrugë të zakonshme kërkon operim me një tabelë të madhe të kombinacioneve. Por, problemi i sintezës së qarqeve logjike shumëbitëshe, ashtu siç u tha edhe më parë, mund të zgjidhet duke ndjekur rrugë tjetër.

Shembull

Qarku logjik i shumëzuesit **4**-bitësh, nëse në hyrje të tij aplikohen numrat binar $\mathbf{X=x_3x_2x_1x_0}$ dhe $\mathbf{Y=y_3y_2y_1y_0}$, kurse rezultati binar që merret në dalje të qarkut është $\mathbf{P=p_7p_6p_5p_4p_3p_2p_1p_0}$.

Për t'i nxjerrë shprehjet e funksioneve dalëse të shumëzimit binar **4**-bitësh, mund të nisemi nga shumëzimi me dorë i numrave **X** e **Y**, duke operuar me vlera të përgjithshme:

$$
\begin{array}{cccccccc}
 & & & & x_3 & x_2 & x_1 & x_0 \\
 & & & & y_3 & y_2 & y_1 & y_0 \\
\cline{5-8}
 & & & & x_3y_0 & x_2y_0 & x_1y_0 & x_0y_0 \\
 & & & x_3y_1 & x_2y_1 & x_1y_1 & x_0y_1 & \\
 & & x_3y_2 & x_2y_2 & x_1y_2 & x_0y_2 & & \\
 & x_3y_3 & x_2y_3 & x_1y_3 & x_0y_3 & & & \\
\hline
p_7 & p_6 & p_5 & p_4 & p_3 & p_2 & p_1 & p_0
\end{array}
$$

Shifrat binare $p_0, p_1, \Lambda\ , p_7$ të cilat fitohen gjatë prodhimit, gjenden përmes shprehjeve matematikore:

$$x_0y_0$$
$$x_1y_0 + x_0y_1$$
$$x_2y_0 + x_1y_1 + x_0y_2$$
$$x_3y_0 + x_2y_1 + x_1y_2 + x_0y_3$$
$$x_3y_1 + x_2y_2 + x_1y_3$$
$$x_3y_2 + x_2y_3$$
$$x_3y_3$$

si dhe bartjeve të cilat paraqiten në procesin e mbledhjeve parciale. Për mbledhje të komponenteve brenda këtyre shprehjeve, si dhe bartjeve përcjellëse, mund të përdoren mbledhës të plotë. Kështu, qarku logjik i shumëzuesit të dy numrave **4**-bitësh, në fomën e tij përfundimtare do të duket si në Fig.15.26.

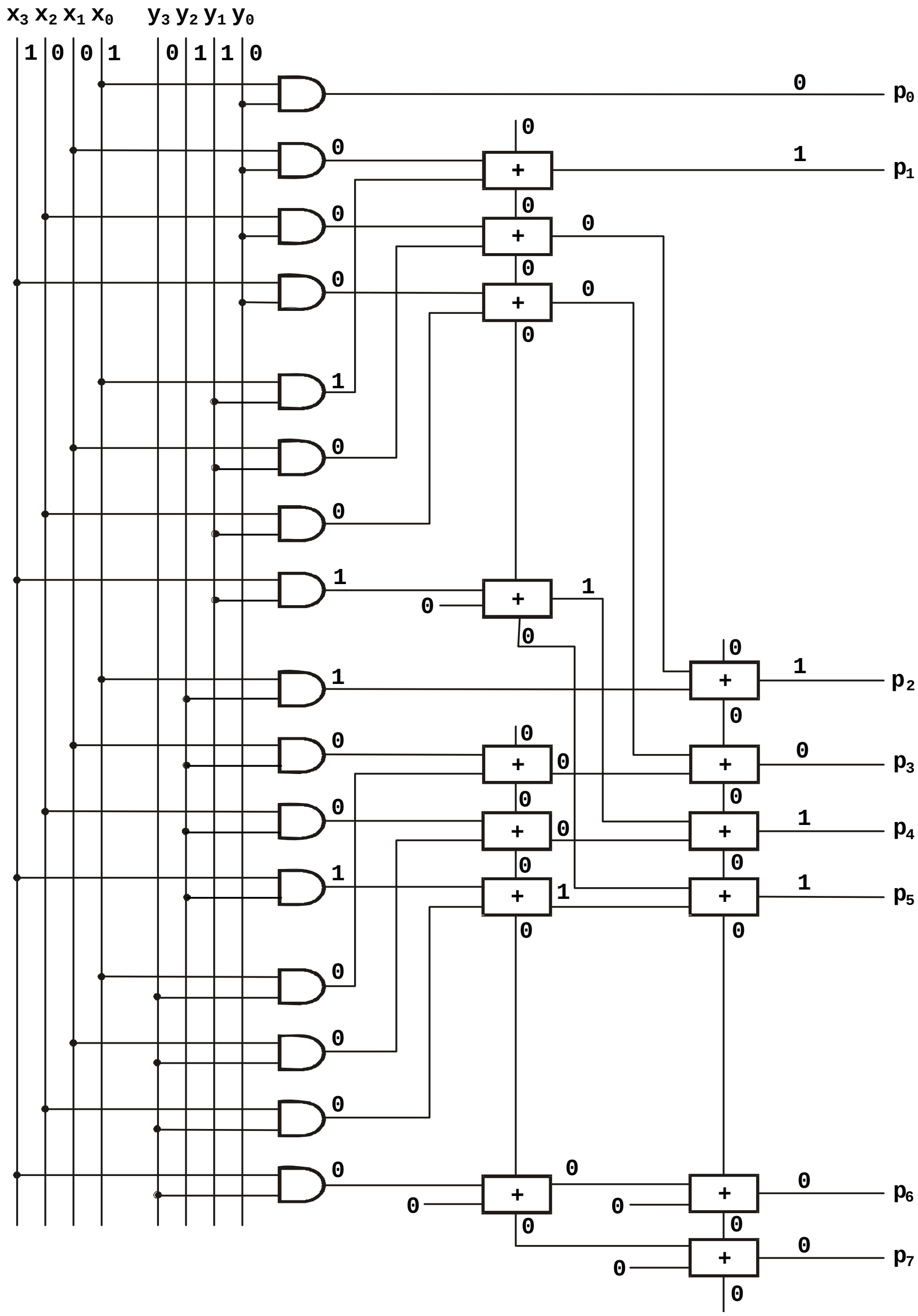
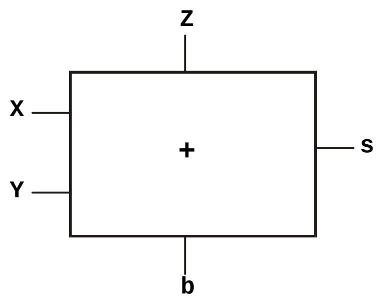


Fig.15.26 Shumëzuesi 4-bitësh

Struktura e hyrjeve dhe e daljeve te mbledhësit e plotë të cilët janë shfrytëzuar te qarku i shumëzuesit shihet në Fig.15.27.



ku janë:

X, **Y** - shifrat binare që mblidhen
Z - bartja hyrëse
s - shuma
b - bartja dalëse

Fig.15.27 Mbledhësi i plotë që shfrytëzohet te shumëzuesi 4-bitësh

Gjatë vizatimit të qarkut të dhënë, në hyrjet e mbledhësve të plotë të cilat nuk shfrytëzohen është shënuar vlera **0**, gjë që nuk ka ndikim në rezultatin e mbledhjes, sepse shuma nuk ndryshon nëse i shtohet vlera **0**.

Realizimi i prodhuesit shumëbitësh mund të thjeshtohet nëse përdoren mbledhës të gatshëm shumëbitësh.

Shembull

Qarku logjik i shumëzuesit **(4x3)**-bitësh, përmes të cilit shumëzohen dy numra binarë, njëri **4**-bitësh $X=x_3x_2x_1x_0$ dhe tjetri **3**-bitësh $Y=y_2y_1y_0$, duke shfrytëzuar si komponente të gatshme dy mbledhës binarë **4**-bitësh.

$$
\begin{array}{ccccccc}
 & & & x_3 & x_2 & x_1 & x_0 \\
 & & & & y_2 & y_1 & y_0 \\
\cline{4-7}
 & & & x_3y_0 & x_2y_0 & x_1y_0 & x_0y_0 \\
 & & x_3y_1 & x_2y_1 & x_1y_1 & x_0y_1 & \\
 & x_3y_2 & x_2y_2 & x_1y_2 & x_0y_2 & & \\
\hline
p_6 & p_5 & p_4 & p_3 & p_2 & p_1 & p_0
\end{array}
$$

Realizimi i qarkut logjik të shumëzuesit mbështetet në idenë e gjetjes së vlerave të shprehjevet matematikore:

$$x_0y_0$$
$$x_1y_0 + x_0y_1$$
$$x_2y_0 + x_1y_1 + x_0y_2$$
$$x_3y_0 + x_2y_1 + x_1y_2$$
$$x_3y_1 + x_2y_2$$
$$x_3y_2$$

kurse **12** komponentet e prodhimeve parciale brenda tyre gjenden duke shfrytëzuar elemente logjike **DHE**.

Në bazë të shprehjeve të dhëna më sipër mund të vizatohet qarku logjik i shumëzuesit **(4x3)**-bitësh, ashtu siç shihet në Fig.15.28. Për ta testuar funksionimin e këtij qarku në hyrjet e tij mund të aplikohen kombinimet e vlerave të mundshme, të cila sillen mes vlerës minimale **0** dhe vlerave maksimale **X=1111** e **Y=111**, përkatësisht ekuivalentëvet decimal **15** e **7**. Kështu, p.sh., nëse në hyrje të qarkut aplikohen vlerat **X=1001** dhe **Y=110**, në dalje të tij do të merret prodhimi i tyre **P=0110110**, gjë që mund të vërtetohet edhe përmes llogaritjes me dorë:

```
  1001
   110
  ----
  0000
 1001
1001
------
110110
```

Këtu, rezultatit duhet shtuar edhe shifrën e parë **0** e cila fitohet si bartje në daljen p_6 të qarkut, por meqë është **0**, gjatë shumëzimit me dorë nuk shënohet.

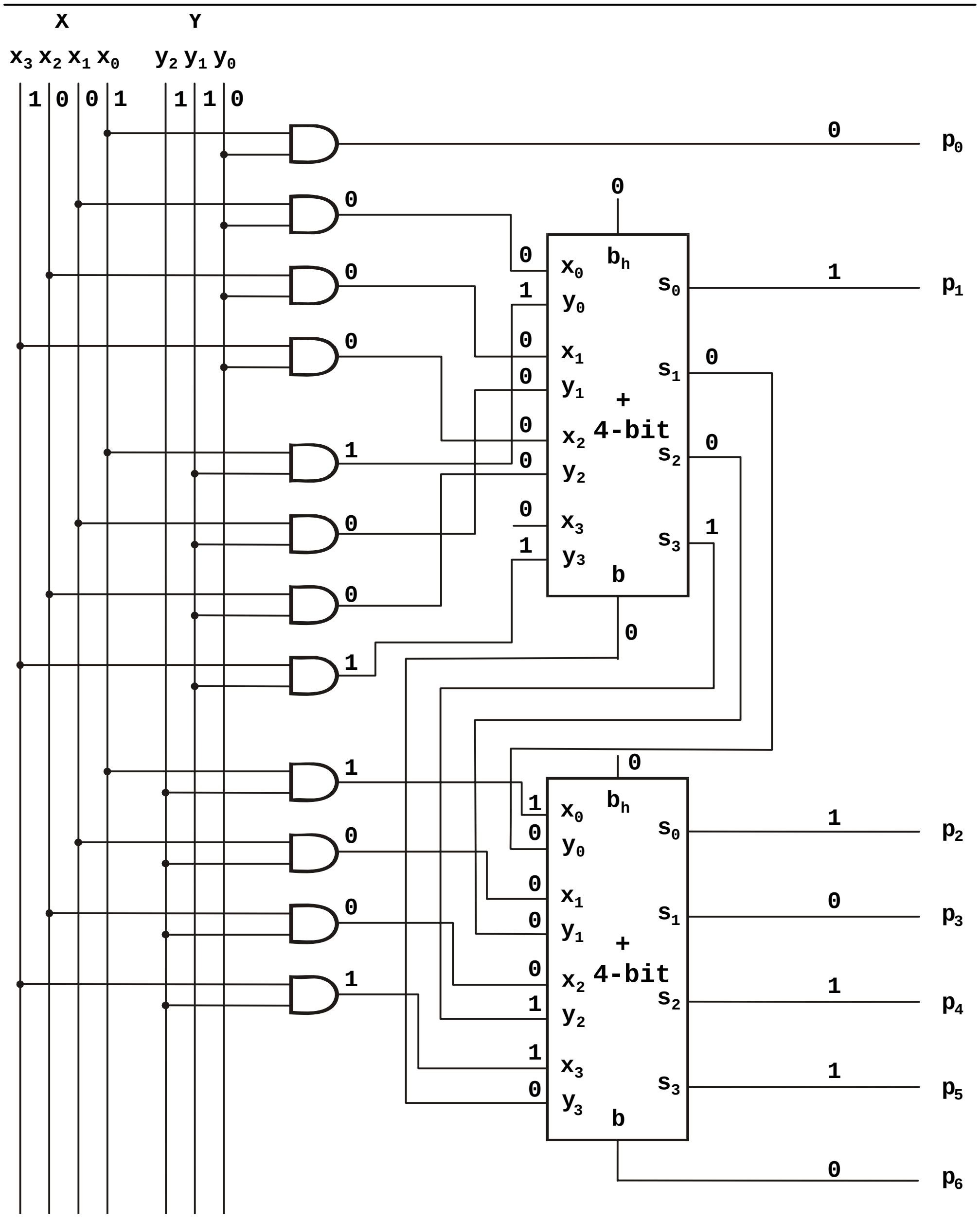


Fig.15.28 Shumzuesi **(4x3)** *-bitësh i realizuar me dy mbledhës* **4** *-bitësh*

Mbledhësit e plotë **4**-bitësh të cilët janë përdorur në qarkun e dhënë në Fig.15.28, funksionojnë sikurse mbledhja e numrave të cilët aplikohen në hyrje të tyre kryhet me dorë. Bartjet hyrëse b_h te të dy mbledhësit janë marrë **0**, kurse bartjet dalëse prej tyre përcillen në daljet **b**. Kështu, p.sh., meqë në hyrjet e mbledhësit të dytë, si rezultat i vlerave hyrëse në qark, aplikohen vlerat **1001** dhe **0100**, rezultati i mbledhjes është:

$$\begin{array}{r} 1001 \\ \underline{0100} \\ 1101 \end{array}$$

gjatë së cilës bartja **b=0** përcillet në daljen p_6 dhe njëkohësisht e paraqet edhe bartjen e përgjithshme nga shumëzuesi.

Pjesëtimi

Meqë pjesëtimi është operacion më i komplikuar, realizimi i qarqeve përmes të cilëve do të pjesëtohen numrat, si qarqe kombinuese të zakonshme, është e pamundshme. Qarku për pjesëtimin e dy numrave do të mund të realizohet, p.sh., përmes metodës së zbritjes suksesive të pjesëtuesit, e cila është shpjeguar në fillim të librit. Por, sinteza e një qarku të tillë imponon nevojën e përdorimit edhe të elementeve memoruese, përkatësisht regjistrave, për të cilët do të bëhet fjalë në pjesën e dytë të këtij libri.

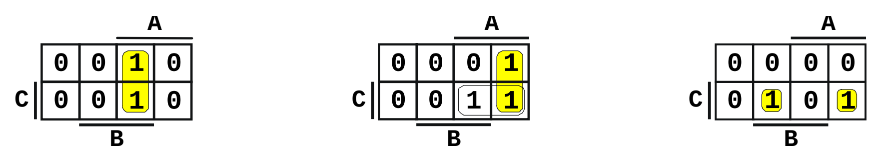
Fuqizimi

Për projektimin e qarqeve përmes të cilave gjenden fuqitë e caktuara të numrave të vegjël, mund të përdoret procedura e zakonshme e sintezës së qarqeve kombinuese.

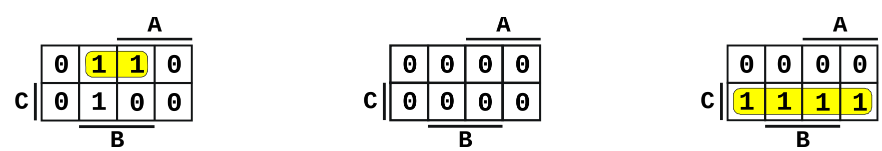
Shembull

Qarku logjik përmes të cilit gjenden katrorët **XYZVWT** e numrave binarë **3**-bitësh **ABC**.

N	A	B	C	X	Y	Z	V	W	T
0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	1
2	0	1	0	0	0	0	1	0	0
3	0	1	1	0	0	1	0	0	1
4	1	0	0	0	1	0	0	0	0
5	1	0	1	0	1	1	0	0	1
6	1	1	0	1	0	0	1	0	0
7	1	1	1	1	1	0	0	0	1



$X = AB$ $\qquad$ $Y = A\overline{B} + AC$ $\qquad$ $Z = \overline{A}BC + A\overline{B}C$



$V = B\overline{C}$ $\qquad$ $W = 0$ $\qquad$ $T = C$

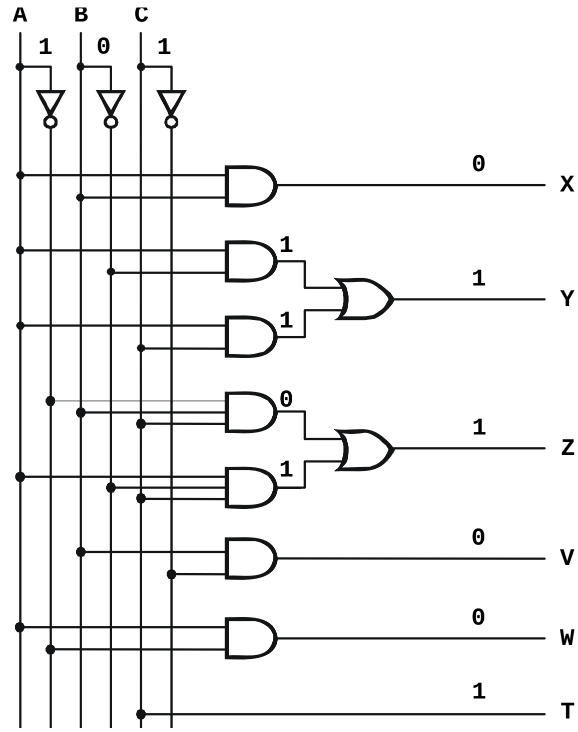


Fig.15.29 Qarku për gjetjen e katrorëve të numrave **3***-bitësh*

Për ta testuar funksionimin e qarkut të dhënë, në hyrje të tij është aplikuar numri decimal **5**, përkatësisht ekuivalenti binar përkatës **101**. Si rezultat, në **6** daljet e qarkut është fituar numri binar **011001**, ashtu siç figuron edhe në tabelën e kombinimeve, të cilit i përgjigjet numri decimal **25**, përkatësisht katrori i numrit të aplikuar në hyrje të qarkut.

Në praktikë, sinteza e qarqeve për llogaritjen e katrororëve të një numri mbështetet në idenë e shumëzimit të numrit me vetveten. Kjo do të thotë se për këtë qëllim mund të përdoret qarku i zakonshëm për shumëzim, por në hyrje të tij duhet të aplikohet dy herë vlera e numrit që ngritet në katror. Kështu, p.sh., nëse duhet të ngritet në katror numri $K=k_n...k_2k_1k_0$, përkatësisht të llogaritet K^2, përmes shumëzimit $K \cdot K$, hyrjet e qarkut logjik duhet të lidhen shkurt, ashtu siç shihet në Fig.15.30.

Fig.15.30 Fuqizimi duke e shfrytëzuar shumëzuesin

Për ngritjen në fuqi të caktuar të numrave binarë, mund të realizohen qarqe të veçanta. Por, edhe në këtë rast fuqizimi mund të bëhet përmes qarkut për shumëzim, duke e përsëritur shumëzimin aq sa është fuqia në të cilën duhet ngritur numrin.

Plotpjesëtimi

Me qëllim të gjetjes së plotpjestueshmërisë së numrave me një numër të caktuar, përdoren qarqe logjike të cilat paraqesin *detektorë të plotpjesëtueshmërisë*.

Shembull

Detektori i plotpjesëtueshmërisë të ekuivalentëve decimal **N** të numrave binar **ABCD**, me numrin decimal **3**. Për realizimin e qarkut logjik përkatës shfrytëzohet edhe dekoduesi **4/16**.

Në daljen **X** të qarkut merret vlera logjike **1**, nëse numri që aplikohet në hyrje të tij është i plotpjesëtueshëm me **3**, përndryshe vlera dalëse është **0**.

N	A	B	C	D	X
0	0	0	0	0	0
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	0
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	0
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

$$X = \sum m^1(3,6,9,12,15)$$

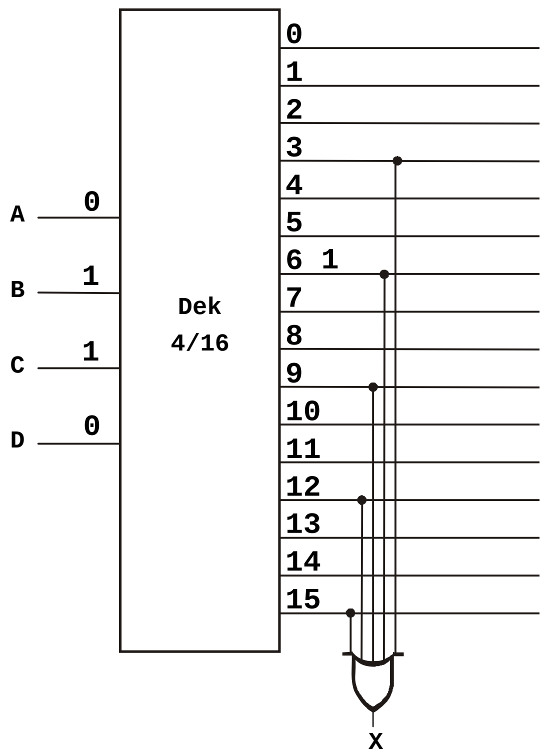


*Fig.15.31 Detektori i plotpjesëtueshmërisë me **3***

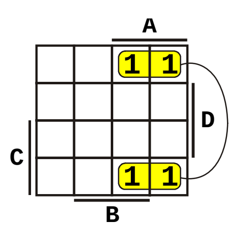
Funksionimi i qarkut mund të testohet duke aplikuar në hyrje të tij vlera numerike të ndryshme. Kështu, p.sh., nëse në hyrje të qarkut aplikohet vlera binare **0110**, në dalje të tij do të merret vlera **X=1**, sepse numri **6** është i plotëpjesëtueshëm me **3**.

Rreth plotpjesëtueshmërisë mund të realizohen edhe qarqe logjike përmes së cilave gjendet edhe rezultati i plotpjesëtimit.

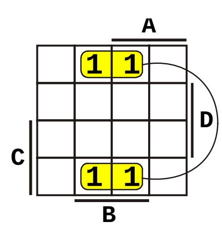
Shembull

Qarku logjik për gjetjen e vlerës binare **XYZ**, të numrit i cili fitohet nga plotpjesëtimi i ekuivalentit binar të numrit decimal **N** me numrin **2**. Për numrat që nuk plotpjesëtohen me **2**, në dalje të qarkut merret vlera **000**.

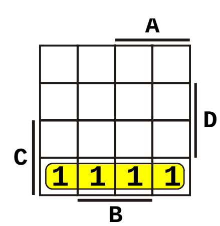
N	A	B	C	D	X	Y	Z
0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0
2	0	0	1	0	0	0	1
3	0	0	1	1	0	0	0
4	0	1	0	0	0	1	0
5	0	1	0	1	0	0	0
6	0	1	1	0	0	1	1
7	0	1	1	1	0	0	0
8	1	0	0	0	1	0	0
9	1	0	0	1	0	0	0
10	1	0	1	0	1	0	1
11	1	0	1	1	0	0	0
12	1	1	0	0	1	1	0
13	1	1	0	1	0	0	0
14	1	1	1	0	1	1	1
15	1	1	1	1	0	0	0



$X = A\overline{D}$



$Y = B\overline{D}$



$Z = C\overline{D}$

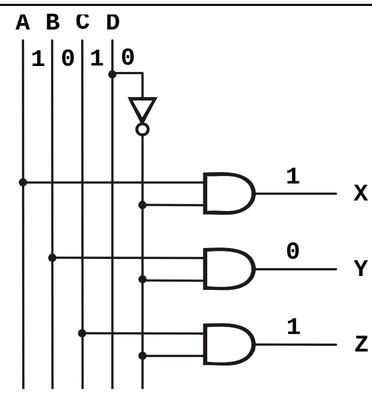


Fig.15.32 Qarku për gjetjen e vlerës së plotë të pjestimit me 2

Qarku logjik i dhënë më sipër duket mjaft i thjeshtë, gjë që është rezultat i ligjshmërisë që përsëritet për numrat çift. Nëse, p.sh., në hyrje të qarkut aplikohet vlera binare **1010**, ekuivalenti decimal i të cilës është **10**, në dalje të qarkut si rezultat fitohet numri **101**, ekuivalenti decimal i të cilit është numri **5**, meqë vlera hyrëse **10** është e plotpjesëtueshme me **2**.

16 Vlerat e funksioneve

Qarqet logjike kombinuese mund të përdoren edhe si gjeneratorë të vlerave numerike të funksioneve të ndryshme aritmetikore.

Funksionet e zakonshme

Procedura e sintezës së qarqeve për gjenerimin e vlerave të funksioneve të zakonshme fillon me përpilimin e tabelës së kombinimeve për të gjitha vlerat e mundshme të variablave që paraqiten në shprehjet e tyre.

Shembull Qarku logjik përmes të cilit gjenerohen vlerat e funksionit:

$$\mathbf{f = 2a + b + 1}$$

nëse variablat **a** dhe **b** i marrin vlerat **0**, **1**, **2** dhe **3**.

	a		b		f			
	A	B	C	D	x	y	z	v
0	0	0	0	0	0	0	0	1
1	0	0	0	1	0	0	1	0
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	1	0	0
4	0	1	0	0	0	0	1	1
5	0	1	0	1	0	1	0	0
6	0	1	1	0	0	1	0	1
7	0	1	1	1	0	1	1	0
8	1	0	0	0	0	1	0	1
9	1	0	0	1	0	1	1	0
10	1	0	1	0	0	1	1	1
11	1	0	1	1	1	0	0	0
12	1	1	0	0	0	1	1	1
13	1	1	0	1	1	0	0	0
14	1	1	1	0	1	0	0	1
15	1	1	1	1	1	0	1	0

$$x = \sum m^1(11,13-15)$$
$$y = \sum m^1(3,5-10,12)$$
$$z = \sum m^1(1-2,4,7,9-10,12,15)$$
$$v = \sum m^1(0,2,4,6,8,10,12,14)$$

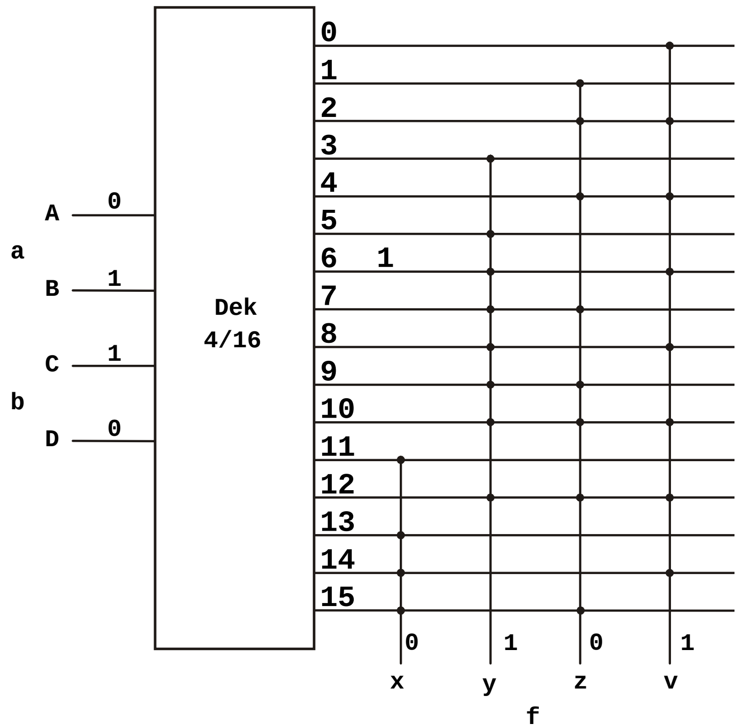


Fig.16.1 Qarku për gjenerimin e vlerave të një funksioni

Për ta testuar funksonimin e qarkut të vizatuar më sipër, le të marrim, p.sh., se në hyrje të tij aplikohen vlerat decimale **a=1** dhe **b=2**, përkatësisht kombinimi i vlerave hyrëse **0110**. Vlera decimale e funksionit **f** do të jetë:

$$\mathbf{f} = \mathbf{2} \cdot \mathbf{1} + \mathbf{2} + \mathbf{1} = \mathbf{5}$$

kurse në dalje të qarkut merret numri binar **0101**, të cilit i përgjigjet vlera decimale **5**.

Përmes një qarku logjik njëkohësisht mund të gjenerohen edhe vlerat e më shumë funksioneve.

Shembull

Qarku logjik në dalje të të cilit gjenerohen vlerat e funksioneve:

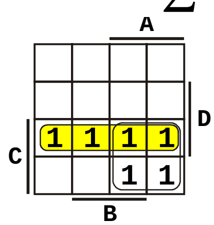
$$\mathbf{R} = \mathbf{a} + \mathbf{3b} + \mathbf{1}$$

$$\mathbf{S} = \mathbf{2a} + \mathbf{b}^2$$

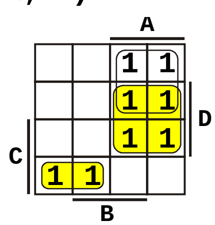
nëse variablat **a** dhe **b** i marrin vlerat **0**, **1**, **2** dhe **3**.

	a		b		R				S			
	A	B	C	D	e	f	g	h	x	y	z	v
0	0	0	0	0	0	0	0	1	0	0	0	0
1	0	0	0	1	0	1	0	0	0	0	0	1
2	0	0	1	0	0	1	1	1	0	1	0	0
3	0	0	1	1	1	0	1	0	1	0	0	1
4	0	1	0	0	0	0	1	0	0	0	1	0
5	0	1	0	1	0	1	0	1	0	0	1	1
6	0	1	1	0	1	0	0	0	0	1	1	0
7	0	1	1	1	1	0	1	1	1	0	1	1
8	1	0	0	0	0	0	1	1	0	1	0	0
9	1	0	0	1	0	1	1	0	0	1	0	1
10	1	0	1	0	1	0	0	1	1	0	0	0
11	1	0	1	1	1	1	0	0	1	1	0	1
12	1	1	0	0	0	1	0	0	0	1	1	0
13	1	1	0	1	0	1	1	1	0	1	1	1
14	1	1	1	0	1	0	1	0	1	0	1	0
15	1	1	1	1	1	1	0	1	1	1	1	1

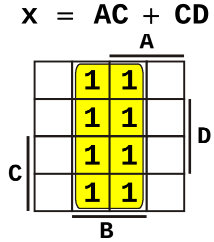
$$e = \sum m^1(3, 6-7, 10-11, 14-15)$$
$$f = \sum m^1(1-2, 5, 9, 11-13, 15)$$
$$g = \sum m^1(2-4, 7-9, 13-14)$$
$$h = \sum m^1(0, 2, 5, 7-8, 10, 13, 15)$$



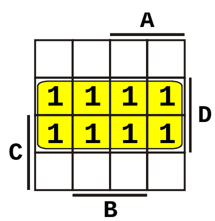
$x = AC + CD$



$y = A\overline{C} + AD + \overline{A}C\overline{D}$



$z = B$



$v = D$

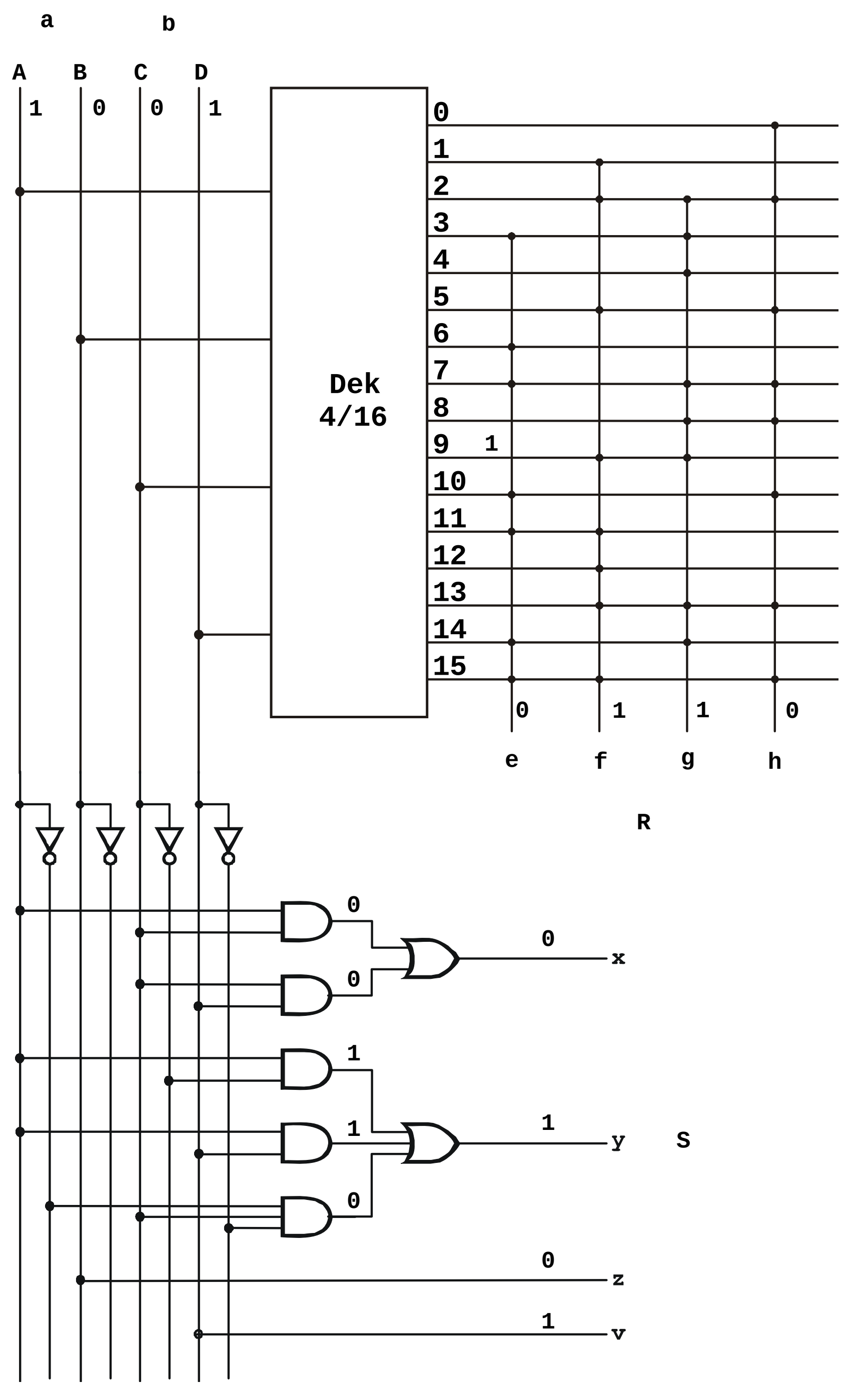


Fig.16.2 Qarku për gjenerimin e vlerave të dy funksioneve

Nëse qarku logjik testohet për vlerat hyrëse **a=2** dhe **b=1**, përkatësisht nëse në hyrje të tij aplikohet vargu i vlerave binare **1001**, në dalje të qarkut do të fitohen vlerat **0110** dhe **0101**, të cilat u përgjigjen vlerave decimale të funksioneve:

$$\mathbf{R} = \mathbf{2} + \mathbf{3} \cdot \mathbf{1} + \mathbf{1} = \mathbf{6}$$

$$\mathbf{S} = \mathbf{2} \cdot \mathbf{2} + \mathbf{1}^{2} = \mathbf{5}$$

Për funksionet që përcaktohen me dy ose edhe me më shumë shprehje, gjenerimi i vlerave gjithashtu mund të bëhet përmes qarqeve kombinuese, duke e përcaktuar kushtin përmes vlerave në një ose më shumë hyrje të qarkut.

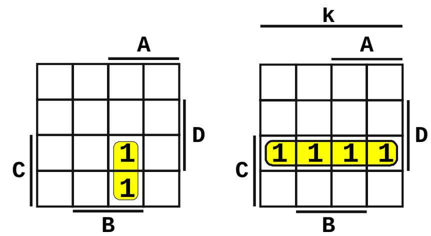
Shembull

Qarku logjik për gjenerimin e vlerave të funksionit:

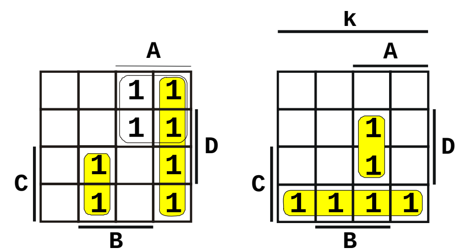
$$\mathbf{g} = \begin{cases} \mathbf{2x + y} & \text{për} \quad \mathbf{k = 0} \\ \mathbf{x + y^2} & \text{për} \quad \mathbf{k = 1} \end{cases}$$

nëse variablat **x** dhe **y** marrin vlerat **0**, **1**, **2** dhe **3.**

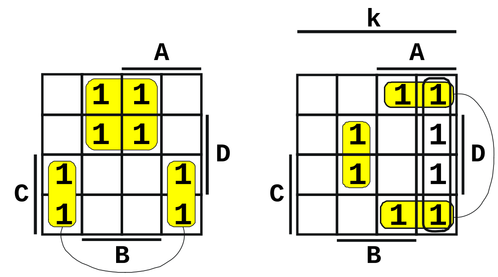
	x		y		g			
k	A	B	C	D	r	s	t	v
0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	1
0	0	0	1	0	0	0	1	0
0	0	0	1	1	0	0	1	1
0	0	1	0	0	0	0	1	0
0	0	1	0	1	0	0	1	1
0	0	1	1	0	0	1	0	0
0	0	1	1	1	0	1	0	1
0	1	0	0	0	0	1	0	0
0	1	0	0	1	0	1	0	1
0	1	0	1	0	0	1	1	0
0	1	0	1	1	0	1	1	1
0	1	1	0	0	0	1	1	0
0	1	1	0	1	0	1	1	1
0	1	1	1	0	1	0	0	0
0	1	1	1	1	1	0	0	1
1	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
1	0	0	1	0	0	1	0	0
1	0	0	1	1	1	0	0	1
1	0	1	0	0	0	0	0	1
1	0	1	0	1	0	0	1	0
1	0	1	1	0	0	1	0	1
1	0	1	1	1	1	0	1	0
1	1	0	0	0	0	0	1	0
1	1	0	0	1	0	0	1	1
1	1	0	1	0	0	1	1	0
1	1	0	1	1	1	0	1	1
1	1	1	0	0	0	0	1	1
1	1	1	0	1	0	1	0	0
1	1	1	1	0	0	1	1	1
1	1	1	1	1	1	1	0	0



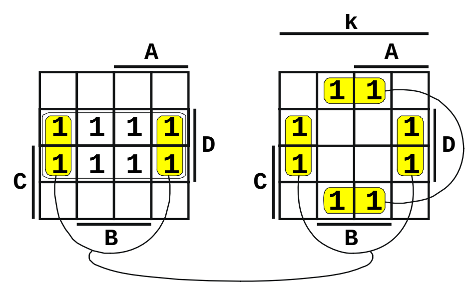
$$r = \bar{k}ABC + kCD$$



$$s = \bar{k}A\bar{B} + \bar{k}A\bar{C} + \bar{k}\bar{A}BC + kABD + kC\bar{D}$$



$$t = \bar{k}B\bar{C} + \bar{k}\bar{B}C + kA\bar{B} + kA\bar{D} + k\bar{A}BD$$



$$v = kB\bar{D} + \bar{B}D + \bar{k}D$$

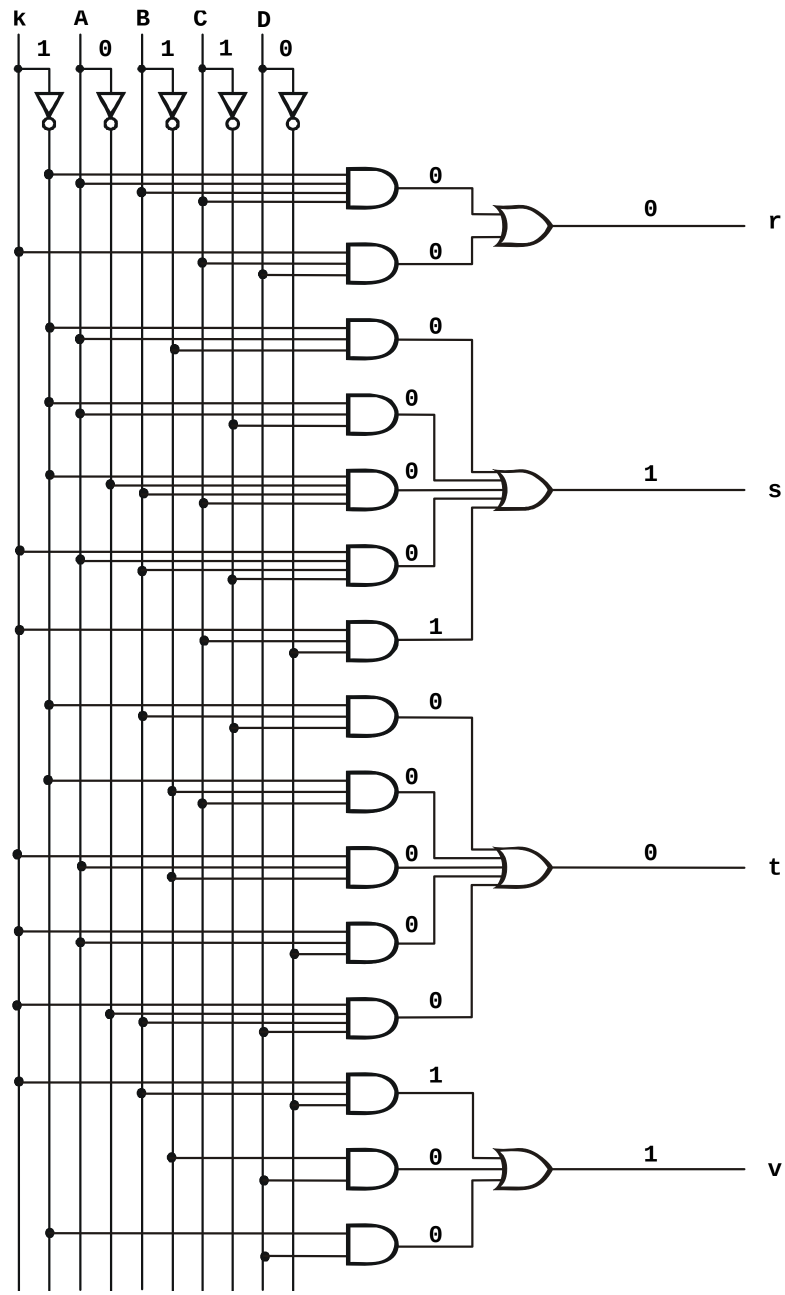


Fig.16.3 Qarku për gjenerimin e vlerave të funksionit që definohet me dy shprehje

Për ta testuar funksionimin e qarkut, le të marrim, p.sh., se në hyrje të tij aplikohen vlerat **x=1**, **y=2** dhe **k=1**, përkatësisht vlera binare **10110**. Në dalje të qarkut do të fitohet vlera e funksionit:

$$\mathbf{g} = \mathbf{1} + \mathbf{2}^{2} = \mathbf{5}$$

ekuivalenti binar i së cilës është **0101**.

Funksionet trigonometrike

Qarqet kombinuese mund të përdoren edhe për gjenerimin e vlerave të funksioneve trigonometrike. Si vlera hyrëse te këto qarqe paraqiten këndet e dhëna në radianë, por në formën e tyre binare. Vlerat në dalje të qarqeve të tilla gjithashtu janë numra binarë. Saktësia e vlerave hyrëse dhe e vlerave të funksioneve trigonometrike në dalje të qarqeve varet nga numri i bitëve që përdoren për paraqitjen e tyre.

Shembull

Qarku logjik për gjenerimin e vlerave të funksionit **Y=sin(πX)**, ku $\mathbf{0} \leq \mathbf{X} \leq \mathbf{1}$.

Në formë binare, vlerat e përafërta të variablës **X** paraqiten duke shfytyëzuar **4** shifra binare $\mathbf{x_3}$, $\mathbf{x_2}$, $\mathbf{x_1}$ dhe $\mathbf{x_0}$, peshat e të cilave janë: **1/2**, **1/4**, **1/8** dhe **1/16** - përkatësisht. Kështu, p.sh., për ta paraqitur vlerën e këndit prej $\mathbf{45^0}$, variabla **X** duhet ta ketë vlerën **1/4**, përkatësisht **0.25** ose në formë binare vlerën **0100**, sepse:

$$\mathbf{0 \cdot (1/2) + 1 \cdot (1/4) + 0 \cdot (1/8) + 0 \cdot (1/16) = 0.25}$$

Për ta paraqitur këndin prej $\mathbf{30^0}$, vlera e variablës **X** duhet të jetë **1/6=0.1666**. Por, meqë paraqitja e saktë e kësaj vlere me **4** bitë është e pamundshme, si vlerë më e përafërt e saj mund të merret vlera **3/16=0.1873** ose **0011**, sepse:

$$\begin{aligned}\mathbf{0 \cdot (1/2) + 0 \cdot (1/4) + 1 \cdot (1/8) + 1 \cdot (1/16)} &= \mathbf{0.1250 + 0.0625} \\ &= \mathbf{0.1875}\end{aligned}$$

Duke vepruar në këtë mënyrë për të gjithë këndet, është përpiluar tabela vijuese në të cilën janë përfshirë vlerat hyrëse dhe vlerat dalëse nga qarku.

N	x_3	x_2	x_1	x_0	X	y_3	y_2	y_1	y_0	Y(radian)	Y(shkallë)
0	0	0	0	0	0.0000	0	0	0	0	0.0000	0.00
1	0	0	0	1	0.0625	0	0	1	1	0.1875	11.25
2	0	0	1	0	0.1250	0	1	1	0	0.3750	22.50
3	0	0	1	1	0.1875	1	0	0	0	0.5000	33.75
4	0	1	0	0	0.2500	1	0	1	1	0.6875	45.00
5	0	1	0	1	0.3125	1	1	0	1	0.8125	56.25
6	0	1	1	0	0.3750	1	1	1	0	0.8750	67.50
7	0	1	1	1	0.4375	1	1	1	1	0.9375	78.75
8	1	0	0	0	0.5000	1	1	1	1	0.9375	90.00
9	1	0	0	1	0.5625	1	1	1	1	0.9375	101.25
10	1	0	1	0	0.6250	1	1	1	0	0.8750	112.50
11	1	0	1	1	0.6875	1	1	0	1	0.8125	123.75
12	1	1	0	0	0.7500	1	0	1	1	0.6875	135.00
13	1	1	0	1	0.8125	1	0	0	0	0.5000	146.25
14	1	1	1	0	0.8750	0	1	1	0	0.3750	157.50
15	1	1	1	1	0.9375	0	0	1	1	0.1875	168.75

Përmes vlerave të dhëna në tabelë realisht nuk mund të merren të gjitha vlerat e mundshme të funksionit **Y**, por vetëm vlerat në **16** pikat e zgjedhura në lakoren përkatëse, ashtu siç shihet në paraqitjen grafike të dhënë në Fig.16.4.

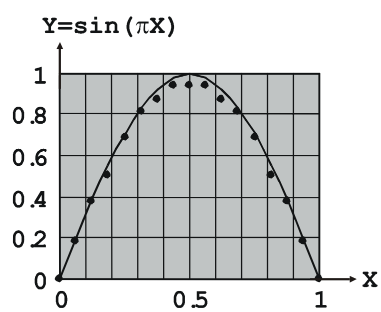
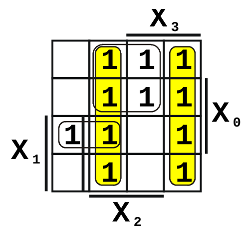
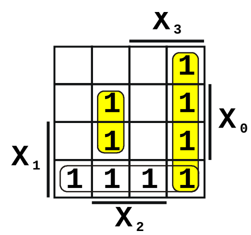


Fig.16.4 Paraqitja grafike e funksionit **sin(πX)**

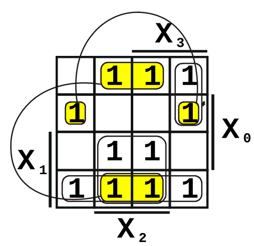
Nëse bazë të vlerave të dhëna në formën binare të variablave **X** dhe **Y**, janë mbushur **K**-diagramet përkatëse, prej ku pastaj janë gjetur shprehjet e funksioneve dhe është vizatuar qarku logjik i gjeneratorit të vlerave të funksionit **Y=sin(πX)**, i cili shihet në Fig.16.5.



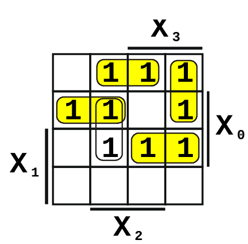
$$y_3 = x_3\overline{x}_2 + x_2\overline{x}_1 + \overline{x}_3x_2 + \overline{x}_3x_1x_0$$



$$y_2 = x_3\overline{x}_2 + x_1\overline{x}_0 + \overline{x}_3x_2x_0$$



$$y_1 = x_1\overline{x}_0 + x_2x_1 + x_2\overline{x}_0 + x_3\overline{x}_2\overline{x}_1 + \overline{x}_2\overline{x}_1x_0$$



$$y_0 = x_3\overline{x}_2\overline{x}_1 + x_3x_1x_0 + \overline{x}_3x_1x_0 + x_2\overline{x}_1\overline{x}_0 + \overline{x}_3\overline{x}_1x_0$$

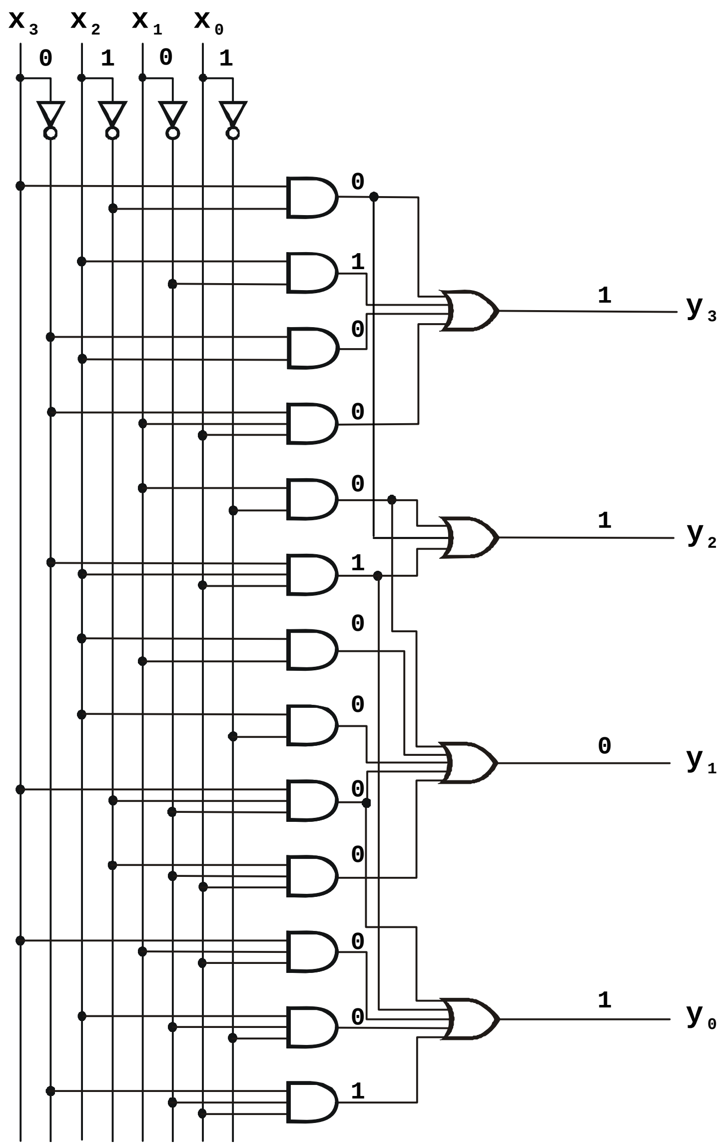


Fig.16.5 Gjeneratori i vlerave të funksionit $Y=\sin(\pi X)$

Memoriet fikse

17

Qarqet kombinuese të cilat janë në gjendje të mbajnë në mend grumbuj informatash në formë tabelash të ndyshme, njihen si *memorie fikse* (ang. Read-Only Memory, **ROM**). Përmes tyre mund të gjenerohen fjalët kodike të kodeve të ndryshme, të bëhet konvertimi i kodeve, të realizohen qarqet aritmetikore, të gjenerohen funksione, ose edhe të mbahen në mend programe kompjuterike.

Të dhënat e vendosura në memorien fikse ruhen edhe pas ndërprerjes së furnizimit të tyre me energji elektrike. Prandaj, memoriet fikse janë të përshtatshme për ruajtjen e të dhënave fikse, siç janë, p.sh., të dhënat për komandimin e punës së pajisjeve të ndryshme digjitale, programet për inicimin e punës së kompjuterëve etj.

Forma e përgjithshme

Organizimi i brendshëm i memories fikse me 2^n-fjalë m-bitëshe, në formë të përgjithshme, duket si në Fig.17.1.

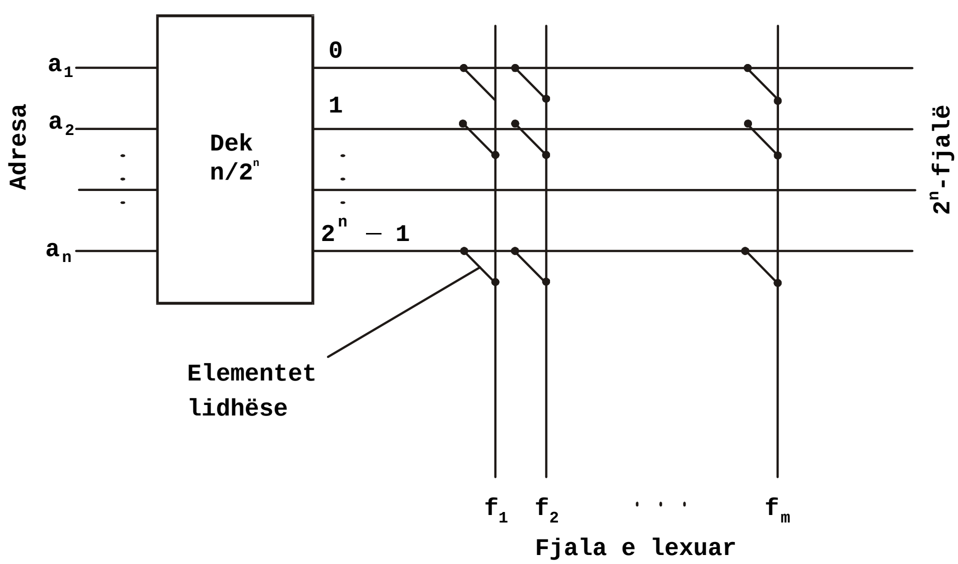


Fig.17.1 Forma e përgjithshme e memories fikse

Nëse në hyrje të memories aplikohet adresa $a_n a_{n-1} ... a_2 a_1$, në dalje të dekoduesit do të zgjidhet njëra nga 2^n fjalët e memories me numra rendor **0**, **1**, **2**,**...**,2^n-1, përkatësisht do të lexohet fjala **m**-bitëshe $f_m f_{m-1} ... f_2 f_1$ dhe përmbajtja e saj do të përcillet në dalje të memories.

Memoria fikse e dhënë në Fig.17.1, skematikisht mund të paraqitet si në Fig.17.2, ku me $2^n \times m$ është shënuar kapaciteti i memories fikse, i cili në fakt është 2^n-fjalë **m**-bitëshe.

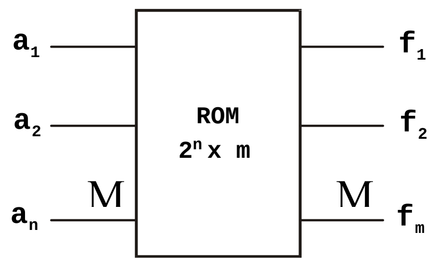


Fig.17.2 Paraqitja skematike e memories fikse

Elementet lidhëse

Si elemente lidhëse, te memoriet fikse përdoren kryesisht *diodat* dhe *transistorët*. Por, në literaturë përmenden edhe realizime të memorieve fikse te të cilat si elemente lidhëse përdoren *rezistorët*, *kondensatorët* ose edhe *bërthamat magnetike*. Në pjesën vijuese të librit, gjatë shpjegimit të mënyrës së realizimit të memorieve fikse, si elemente lidhëse brenda tyre do të përdoren diodat dhe transistorët bipolarë.

Struktura fillestare e memories fikse me dioda, duket si në Fig.17.3.

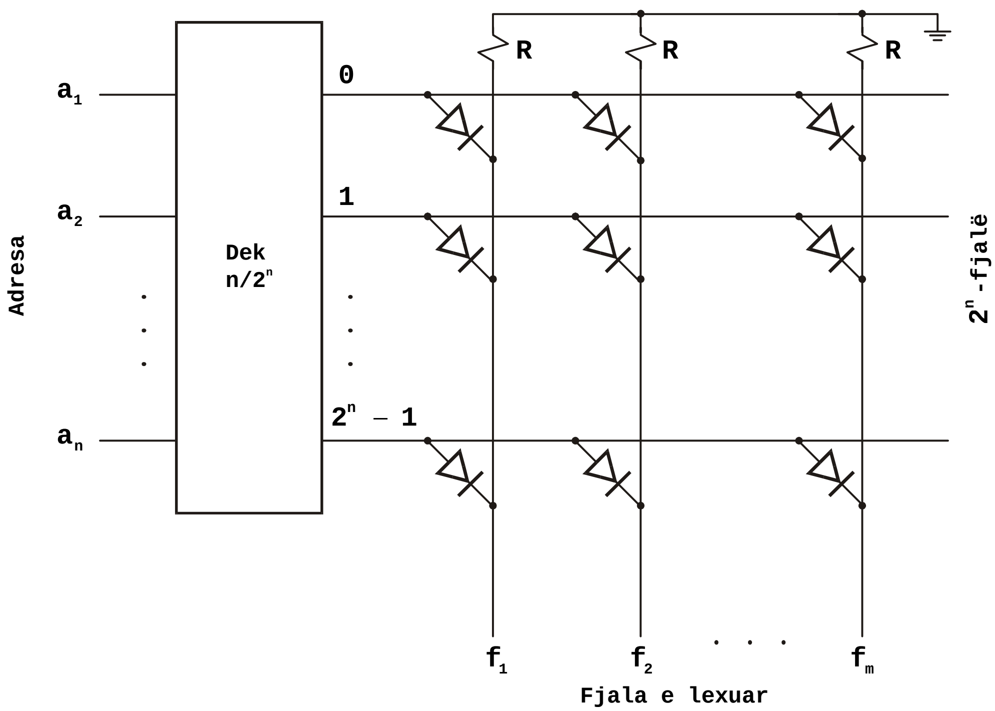


Fig.17.3 Forma e përgjithshme e memories fikse me dioda

Njëri nga realizimet e mundshme të memories fikse me transistor bipolar, në formën e saj të përgjithshme, është dhënë në Fig.17.4.

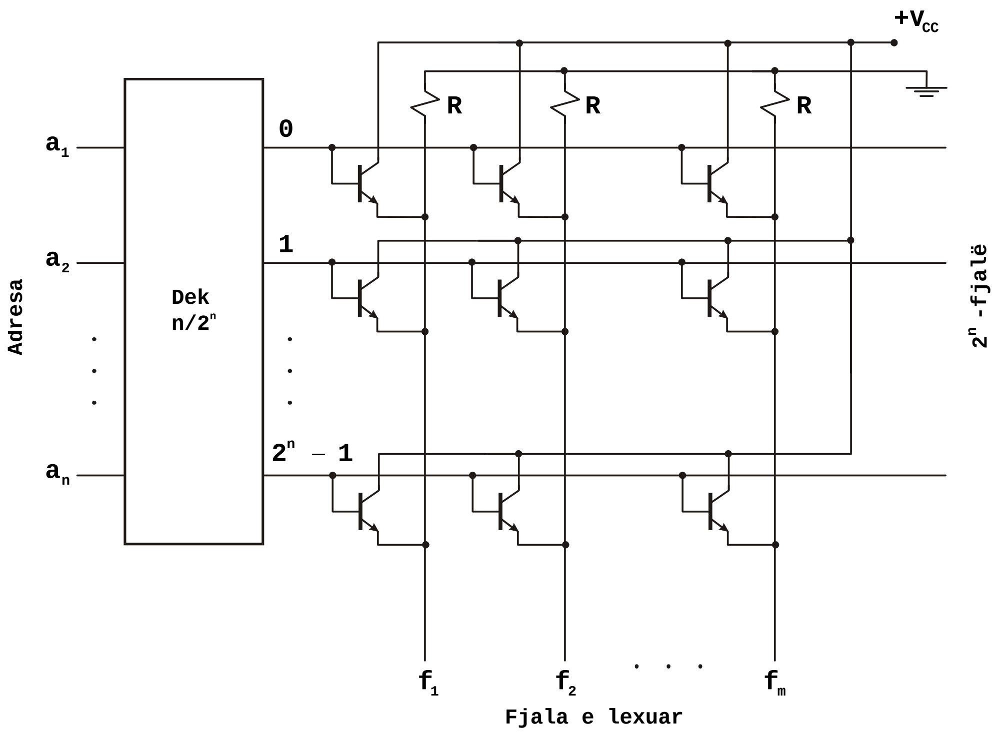


Fig.17.4 Forma e përgjithshme e memories fikse me transistor

Programimi i memories fikse

Fillimisht, memoria fikse është e mbushur me vlera binare **1**. Procesi i vendosjes së të dhënave në memorien fikse njihet si *programim i memories fikse*. Gjatë programimit, në memorie eliminohen lidhjet elektrike të c*elulave memoruese* në të cilat, në vend të vlerave binare **1**, duhet të memorohen vlera binare **0**. Ky eliminim mund të bëhet, p.sh., me djegien e diodave, përkatësisht të transistorëve që gjenden në celulat ku duhet të memorohen vlerat binare **0**. Për këtë qëllim, pasi të zgjidhen celulat përmes adresave përkatëse, nëpër to aplikohet një puls i tensionit të lartë (**10-30V**), me ç'rast shkaktohet rrjedhja e një rryme jonormale dhe djegia e lidhjeve elektrike të celulave.

Programimi mund të bëhet nga *prodhuesi*, ose edhe nga vetë *shfrytëzuesit* e memorieve fikse.

Memoriet fikse që programohen nga *prodhuesi* njihen edhe me shkurtesën **MROM**, sepse gjatë programimit të tyre, për gjenerimin e lidhjeve, shfrytëzohet *maska* përkatëse. Pasi të programohen një herë, përmbajtjet e memorieve të tilla fikse vetëm mund të lexohen, por jo edhe të riprogramohen.

Memoriet që mund të programohen nga *shfrytëzuesi* njihen me shkurtesën **PROM** (nga Programmable **ROM**) dhe prodhohen në dy versione bazike:

- ato që mund të programohen vetëm një herë dhe
- ato që riprogramohen, duke e fshirë së pari përmbajtjen ekzistuese.

Në versionin e memorieve fikse që riprogramohen bëjnë pjesë memoriet **EPROM** (nga Erase Programmable **ROM**), në celulat memoruese të së cilave vendosen transistorët **MOS**, me *gejt silici*. Këtu, gjatë programimit të memories, përmes aplikimit të impulseve të tensionit të lartë (**10-25V**), injektohen elektrone me energji të lartë, për ta vendosur transistorin në gjendjen logjike **0**.

Fshirja e përmbajtjes së memorieve **EPROM** bëhet duke i ekspozuar ato në një *dritë ultravjollcë*, për një kohë të caktuar, përmes dritares që gjendet mbi *çipin* përkatës. Kurse, për programim, përkatësisht për mbushje me të dhëna binare të celulave të saj, përdoret *programatori* përkatës, i cili lidhet në dalje të kompjuterit, ku përcaktohet përmbajtja e memories fikse.

Ndryshimi i përmbajtjes së vetëm një pjese të caktuar të memories **EPROM** nuk është e mundshme të bëhet pa e fshirë dhe pa e riprogramuar komplet memorien. Por, te memoriet fikse të tipit **EEPROM** (nga Electrically Erasable **PROM**) mund të ndryshohet përmbajtja e vetëm një bajti. Fshirja dhe rishkruarja në celulat memoruese të memories **EEPROM** bëhet direkt, pa përdorur *dritë ultaravjollcë* dhe *programator special*, me një shpejtësi shumë më të madhe se te memoriet **EPROM**.

Realizimi i qarqeve me memorie fikse

Për të parë realizimin e qarqeve që përmbajnë edhe memorie fikse, me dioda dhe transistorë, fillimisht le të marrim një shembull elementar.

Shembull

Realizimi i tabelës së kombinimeve:

N	A	B	x	y	z
0	0	0	1	0	1
1	0	1	0	1	0
2	1	0	0	0	1
3	1	1	1	1	0

përmes memories fikse, nëse si elemente lidhëse përdoren diodat, ose transistorët.

$$x = \sum m^1(0,3)$$
$$y = \sum m^1(1,3)$$
$$z = \sum m^1(0,2)$$

Realizimi me dioda

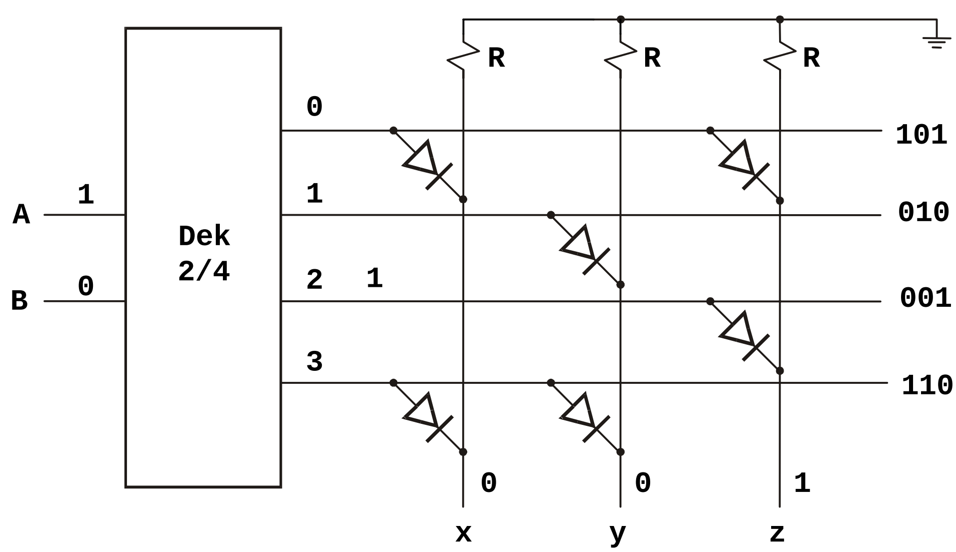


Fig.17.5 Realizimi i memories fikse me dioda

Nga qarku i dhënë shihet se memoria fikse përmban **4**-fjalë **3**-bitëshe: **101**, **010**, **001** dhe **110**. Për leximin e fjalës së caktuar nga memoria, duhet të zgjidhet adresa përkatëse **AB**. Kështu, p.sh., nëse në hyrje të dekoduesit aplikohet vlera binare e adresës **10**, në daljet **xyz** të memories do të merret përmbajtja **001** e fjalës me numër rendor **2**, sepse vetëm në daljen përkatëse të dekoduesit fitohet sinjali me vlerë binare **1**, i cili përmes diodës përcillet vetëm në daljen **z** të qarkut.

Realizimi me transistorë

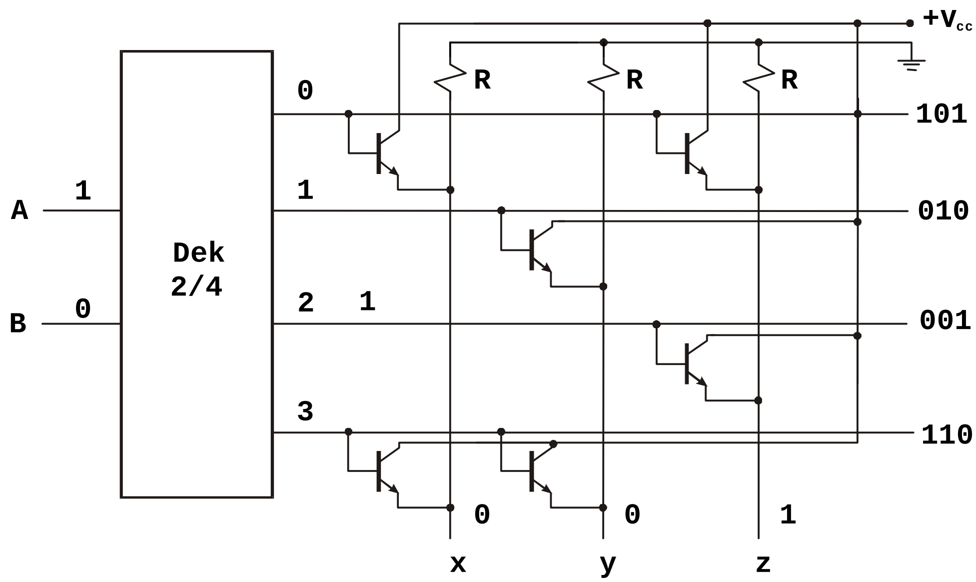


Fig.17.6 Realizimi i memories fikse me transistorë

Për testimin e funksionimit të qarkut të dhënë, si edhe në rastin e versionit të qarkut me dioda, në hyrje të tij mund të aplikohet njëra nga adresat e mundshme **AB**. Nëse, p.sh., në hyrje të qarkut aplikohet adresa **10**, në daljet **xyz** do të merret përmbajtja **001** e fjalës së tretë të saj, sepse sinjali i vlerës binare **1** paraqitet vetëm në daljen **2** të dekoduesit dhe pastaj përmes transistorit përkatës përcillet në daljen **z** të qarkut.

Në pjesën vijuese, që të mos përsëritet vizatimi i diodave, ose i transistorëve, elementet lidhëse do të paraqiten përmes vizave, kurse rezistorët dhe lidhjet për furnizim elektrik nuk do të vizatohen. P.sh., qarku i mëspërm do të paraqitet si në Fig.17.7.

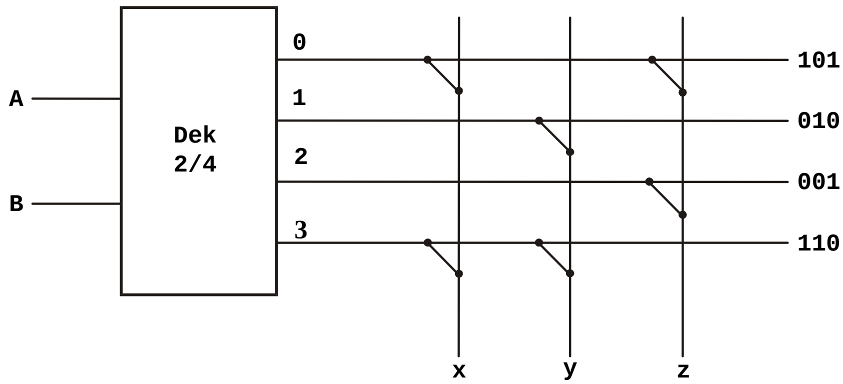


Fig.17.7 Paraqitja e elementeve lidhëse përmes vizave

Në vend të vizave, lidhjet e nevojshme mund të paraqiten edhe duke e shënuar një pikë në vendin e prerjes së vijave horizontale dhe vertikale, ashtu siç është paraqitur në Fig.17.8 qarku nga shembulli i mësipërm.

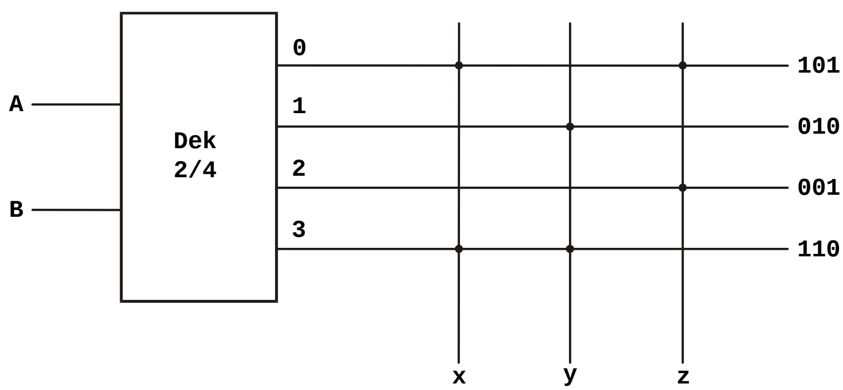
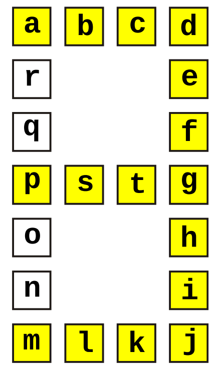


Fig.17.8 Paraqitja e elementeve lidhëse përmes pikave

Memoria fikse sidomos përdoret kur qarku logjik përkatës duhet të gjenerojë tabela me më shumë vlera numerike.

Shembull

Qarku logjik me një memorie fikse, përmes të cilit në indikatorin me **20** katrorë:



gjenerohen shifrat decimale **0**, **1**, **2** dhe **3**.

N	A	B	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
1	0	1	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
2	1	0	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	1	0	0	1	1
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	0	0	1	1

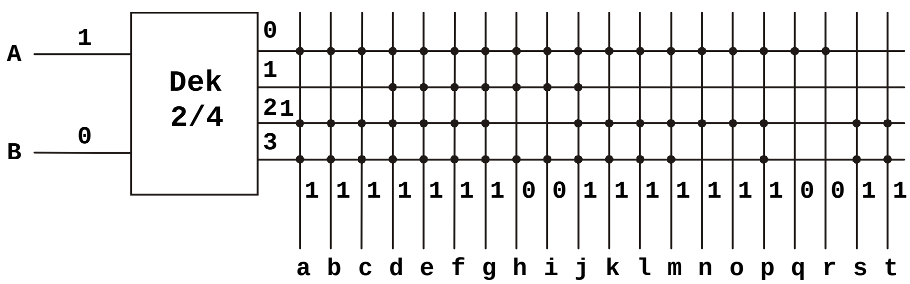


Fig.17.9 Gjeneratori i shifrave decimale në indikatorin me 20 katrorë

Gjenerimi i **4** shifrave decimale përmes qarkut të dhënë mund të testohet, nëse në hyrje të tij aplikohen **4** kombinimet e mundshme të adresave të **4**-fjalëve që janë vendosur në memorie. P.sh., nëse zgjidhet adresa **AB=10**, në **20** daljet e qarkut do të merren vlerat e rreshtit të tretë të tabelës, sepse në daljen e **3** të dekoduesit paraqitet sinjali i vlerës logjike **1**.

Përdorimi i memorieve fikse

Disa tipe të qarqeve kombinuese, që janë përmendur në pjesët paraprake të librit, shumë më thjesht mund të realizohen përmes memorieve fikse, gjë që do të shpjegohet me shembuj të qarqeve që jepen në vijim.

Koduesit

Realizimi i koduesit përmes memories fikse formalisht dallohet nga realizimi i zakonshëm i memorieve fikse, për shkak se zgjedhja e fjalëve brenda memories bëhet në rrugë direkte, pa ndërmjetësimin e dekoduesit të adresës hyrëse.

Shembull

Koduesi i shifrave të sistemit oktal të numrave, me numra binarë **3**-bitësh.

N	0	1	2	3	4	5	6	7	x	y	z
0	1	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	1
2	0	0	1	0	0	0	0	0	0	1	0
3	0	0	0	1	0	0	0	0	0	1	1
4	0	0	0	0	1	0	0	0	1	0	0
5	0	0	0	0	0	1	0	0	1	0	1
6	0	0	0	0	0	0	1	0	1	1	0
7	0	0	0	0	0	0	0	1	1	1	1

$$x = \sum m^1(4-7)$$
$$y = \sum m^1(2-3,6-7)$$
$$z = \sum m^1(1,3,5,7)$$

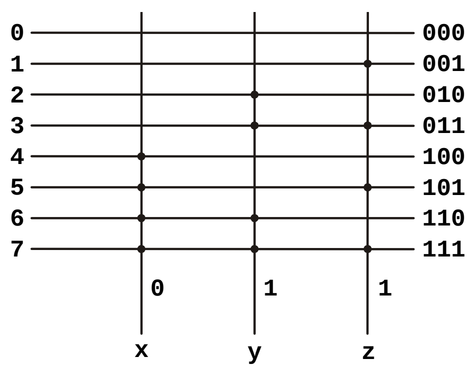


Fig.17.10 Koduesi i shifrave të sitemit oktal

Nga qarku i dhënë shihet se për zgjedhjen e fjalëve të vendosura në memorie adresat përkatëse jepen direkt, pa ndërmjetësimin e dekoduesit. Kështu, p.sh., nëse sinjali me vlerë binare **1** aplikohet në hyrjen **3**, kurse në të gjitha hyrjet e tjera aplikohen sinjale me vlera logjike **0**, në dalje të qarkut do të gjenerohet fjala kodike **011**, përkatësisht ekuivalenti binar i shifrës decimale **3**.

Realizimi i koduesve, kur informatat elementare kodohen me fjalë kodike shumëbitëshe, është më i thjeshtë nëse shfrytëzohen memoriet fikse.

Shembull

Qarku logjik me memorie fikse, përmes të cilit, në kodin **ASCII** kodohen shifrat e sistemit heksadecimal (**0**, **1**, **2**, ..., **A**, **B**, **C**, **D**, **E**, **F**), duke ua shtuar edhe bitin për *paritet tek* të njësheve.

N	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	p	r	s	t	x	y	z	v
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1
2	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0
3	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	1	1
4	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0
5	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	1	0	1
6	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	1	0	1	1	0
7	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1
8	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0
9	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	1	1	0	0	1
10	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0	1
11	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	1	0
12	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	1
13	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	1	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	0	1
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	1	0

$$p = \sum m^1(0, 3, 5-6, 9-11, 13)$$

$$r = \sum m^1(10-15)$$

$$s = \sum m^1(0-9)$$

$$t = \sum m^1(0-9)$$

$$x = \sum m^1(8-9)$$

$$y = \sum m^1(4-7, 13-15)$$

$$z = \sum m^1(2-3, 6-7, 11-12, 15)$$

$$v = \sum m^1(1, 3, 5, 7, 9-10, 12, 14)$$

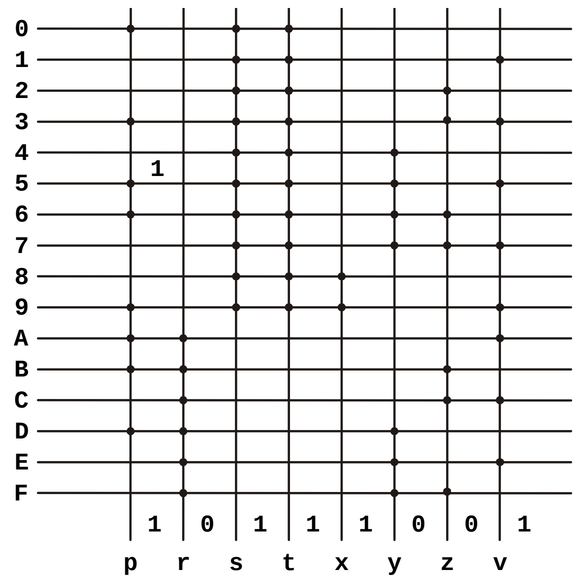


***Fig.17.11 Kodimi i shifrave heksadecimale në kodin* `ASCII`**

Nëse, p.sh., në hyrje të qarkut të dhënë aplikohet sinjali me vlerën **`1`** vetëm në hyrjen **`9`**, në dalje të qarkut do të merret fjala kodike **`0111001`** në kodin **`ASCII`**, të shifrës heksadecimale **`9`**, të cilës në fillim i është shtuar edhe biti **`p=1`** për paritet tek.

Dekoduesit

Për dallim nga koduesit, te qarqet logjike të dekoduesve të realizuar me memorie fikse, zgjedhja e fjalës e cila lexohet nga memoria bëhet përmes adresës përkatëse dhe dekoduesit që e gjeneron sinjalin me vlerën binare **`1`**, për lexim.

Shembull	Qarku logjik përmes të cilit dekodohen në shifra të sistemit decimal: **`0`**, **`1`**, **`2`**, ..., **`8`**, **`9`** fjalët kodike të ekuivalentëve binarë **`xyzv`** përkatës. Për numrat binarë, të cilët nuk u përkasin shifrave decimale, në daljen e veçantë të qarkut **`d`** të gjenerohet vlera binare **`1`**.

x	y	z	v	0	1	2	3	4	5	6	7	8	9	d
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	1	0	0	0	0
0	1	1	1	0	0	0	0	0	0	0	1	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	1	0	0
1	0	0	1	0	0	0	0	0	0	0	0	0	1	0
1	0	1	0	0	0	0	0	0	0	0	0	0	0	1
1	0	1	1	0	0	0	0	0	0	0	0	0	0	1
1	1	0	0	0	0	0	0	0	0	0	0	0	0	1
1	1	0	1	0	0	0	0	0	0	0	0	0	0	1
1	1	1	0	0	0	0	0	0	0	0	0	0	0	1
1	1	1	1	0	0	0	0	0	0	0	0	0	0	1

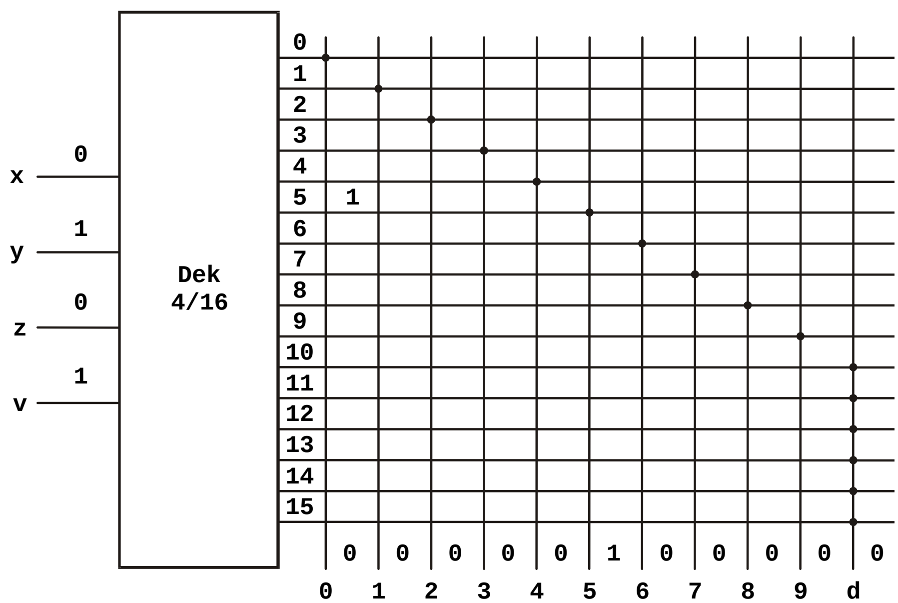


Fig.17.12 Dekoduesi i fjalëve kodike binare në shifra decimale

Për ta testuar funksionimin e qarkut të dhënë, në hyrjet e tij duhet të aplikohen vlerat binare nga grupi i vlerave të cilat i paraqesin ekuivalentët binarë të shifrave decimale dhe të atyre që nuk i takojnë këtij grupi. P.sh., nëse në hyrje aplikohet vlera binare **0101**, sinjali i vlerës binare **1** do të gjenerohet në daljen **5** të dekoduesit **4/16** dhe të qarkut i cili gjithashtu punon si dekodues.

Konvertuesit e kodeve

Memoriet fikse veçanërisht janë të përshtatshme nëse përdoren si konvertues të kodeve.

Shembull

Konvertuesi i fjalëve kodike të kodit **BCD 5221** në ato të kodit **BCD 5311**, të cilat janë dhënë në tabelën e Fig.2.1. Për fjalët kodike që nuk i përkasin kodit **BCD 5221** në dalje të qarkut është paraparë të gjenerohet vlera binare **1111**.

N	x	y	z	m	x	y	z	v
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	1	0	0
4	0	1	0	0	1	1	1	1
5	0	1	0	1	1	1	1	1
6	0	1	1	0	0	1	0	1
7	0	1	1	1	1	1	1	1
8	1	0	0	0	1	0	0	0
9	1	0	0	1	1	0	0	1
10	1	0	1	0	1	0	1	1
11	1	0	1	1	1	1	0	0
12	1	1	0	0	1	1	1	1
13	1	1	0	1	1	1	1	1
14	1	1	1	0	1	1	0	1
15	1	1	1	1	1	1	1	1

$$x = \sum m^1(4-5, 7-15)$$

$$y = \sum m^1(3-7, 11-15)$$

$$z = \sum m^1(2, 4-5, 7, 10, 12-13, 15)$$

$$v = \sum m^1(1-2, 4-7, 9-10, 12-15)$$

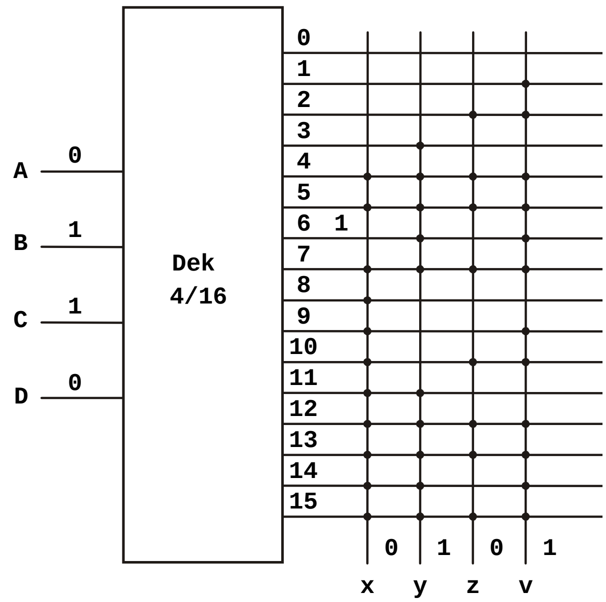


Fig.17.13 Konvertuesi i kodit* BCD *5221 në kodin* BCD *5311

Këtu, për fjalët kodike që nuk shfrytëzohen është paraparë që në dalje të konvertuesit të gjenerohet fjala kodike **1111**. Në fakt, në këto raste vlerat hyrëse në qark janë të gabueshme, prandaj zgjidhje më e mirë do të ishte sikur të parashihet një dalje e veçantë, ku për fjalët kodike që nuk i takojnë kodit **BCD 5221**, do të gjenerohej sinjali me vlerën **1**.

Për ta testuar funksionimin e qarkut të dhënë, mund të marrim kombinime të fjalëve kodike të cilat do të aplikohen në hyrje të qarkut. Kështu, p.sh., nëse në hyrje të qarkut aplikohet fjala kodike **0110** e kodit **BCD 5221**, në dalje të tij do të merret fjala kodike përkatëse **0101**, në kodin **BCD 5311**. Por, sikur në hyrje të qarkut të aplikohej kombinimi i shifrave binare **0101**, në dalje të tij merret vargu i shifrave binare **1111**, për të treguar se fjala kodike hyrëse është e gabueshme.

Për kombinimet që nuk bëjnë pjesë në grupin e fjalëve kodike të kodit që konvertohet, gjatë konvertimit mund të mos zgjidhet asgjë, përkatësisht të merren si arbitrare.

Shembull	Konvertuesi i fjalëve kodike të kodit **Excess-3**, në ato të kodit **2** prej **5**. Për fjalët kodike që nuk i përkasin kodit **Excess-3**, merren vlera arbitrare.

N	Excess-3				2 prej 5				
	A	B	C	D	x	y	z	v	t
0	0	0	0	0	+	+	+	+	+
1	0	0	0	1	+	+	+	+	+
2	0	0	1	0	+	+	+	+	+
3	0	0	1	1	1	1	0	0	0
4	0	1	0	0	0	0	0	1	1
5	0	1	0	1	0	0	1	0	1
6	0	1	1	0	0	0	1	1	0
7	0	1	1	1	0	1	0	0	1
8	1	0	0	0	0	1	0	1	0
9	1	0	0	1	0	1	1	0	0
10	1	0	1	0	1	0	0	0	1
11	1	0	1	1	1	0	0	1	0
12	1	1	0	0	1	0	1	0	0
13	1	1	0	1	+	+	+	+	+
14	1	1	1	0	+	+	+	+	+
15	1	1	1	1	+	+	+	+	+

$$x = \sum m^1(3, 10-12)$$
$$y = \sum m^1(3, 7-9)$$
$$z = \sum m^1(5-6, 9, 12)$$
$$v = \sum m^1(4, 6, 8, 11)$$
$$t = \sum m^1(4-5, 7, 10)$$

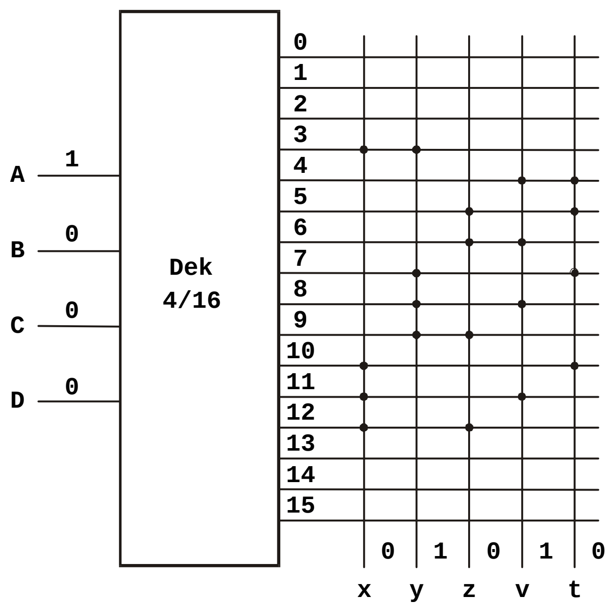


Fig.17.14 Konvertuesi i kodit **Excess-3** *në kodin* **2** *prej* **5**

Pavarësisht se në tabelën e kombinimeve për fjalët kodike që nuk i takojnë kodit **Excess-3**, në kodin **2 prej 5** janë shënuar vlera arbitrare **+**, gjatë realizimit të qarkut një gjë e tillë nuk është marrë parasysh. Për këto kombinime në hyrje të qarkut, vlerat dalëse në të gjitha daljet e qarkut do të jenë **0**. Kështu, p.sh., nëse në hyrje të qarkut aplikohet kombinimi **1101**, i cili nuk bën pjesë në grumbullin e fjalëve kodike të kodit **Excess-3**, kombinimi dalës nga qarku do të jetë **00000**, sepse asnjëra nga daljet e qarkut nuk lidhet në daljen **13** të dekoduesit **4/16**. Por, nëse, p.sh., në hyrje të qarkut aplikohet vlera **1000**, në dalje të qarkut do të gjenerohet fjala kodike përkatëse **01010** në kodin **2 prej 5**, sepse sinjali me vlerën **1** paraqitet vetëm në daljen **8** të dekoduesit **4/16**.

Qarqet aritmetikore

Shfrytëzimi i memorieve fikse për kryerjen e operacioneve aritmetikore, është i përshtatshëm sidomos tek operacioni i shumëzimit, i pjesëtimit ose fuqizimit, meqë këto operacione janë më të ndërlikuara.

Shembull

Qarku për shumëzim të numrit **2**-bitësh **m=AB** me numrin **3**-bitësh **n=CDE**, i realizuar duke e shfrytëzuar një memorie fikse.

Në faqen vijuese është dhënë tabela e kombinimeve, në bazë të së cilës është vizatuar edhe qarku përkatës, ashtu siç shihet në Fig.17.15.

Nga qarku i dhënë shihet se realizimi i tij është shumë më i thjeshtë se realizimi i mundshëm me elemente logjike. Për ta testuar punën e qarkut, le të marrim, p.sh., se në hyrje të tij aplikohen vlerat **m=11** dhe **n=011**, përkatësisht se **ABCDE=11011**. Sinjali me vlerë logjike **1** do të paraqitet në daljen **27** të dekoduesit **5/32**, kurse vlera në dalje të qarkut është **xyzvt=01001**, e cila i përgjigjet prodhimit të kërkuar:

```
   11   m
  011   n
 ------
   11
  11
 00
 ------
 1001   m·n
```

	m		n			m·n				
N	A	B	C	D	E	x	y	z	v	t
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	0
2	0	0	0	1	0	0	0	0	0	0
3	0	0	0	1	1	0	0	0	0	0
4	0	0	1	0	0	0	0	0	0	0
5	0	0	1	0	1	0	0	0	0	0
6	0	0	1	1	0	0	0	0	0	0
7	0	0	1	1	1	0	0	0	0	0
8	0	1	0	0	0	0	0	0	0	0
9	0	1	0	0	1	0	0	0	0	1
10	0	1	0	1	0	0	0	0	1	0
11	0	1	0	1	1	0	0	0	1	1
12	0	1	1	0	0	0	0	1	0	0
13	0	1	1	0	1	0	0	1	0	1
14	0	1	1	1	0	0	0	1	1	0
15	0	1	1	1	1	0	0	1	1	1
16	1	0	0	0	0	0	0	0	0	0
17	1	0	0	0	1	0	0	0	1	0
18	1	0	0	1	0	0	0	1	0	0
19	1	0	0	1	1	0	0	1	1	0
20	1	0	1	0	0	0	1	0	0	0
21	1	0	1	0	1	0	1	0	1	0
22	1	0	1	1	0	0	1	1	0	0
23	1	0	1	1	1	0	1	1	1	0
24	1	1	0	0	0	0	0	0	0	0
25	1	1	0	0	1	0	0	0	1	1
26	1	1	0	1	0	0	0	1	1	0
27	1	1	0	1	1	0	1	0	0	1
28	1	1	1	0	0	0	1	1	0	0
29	1	1	1	0	1	0	1	1	1	1
30	1	1	1	1	0	1	0	0	1	0
31	1	1	1	1	1	1	0	1	0	1

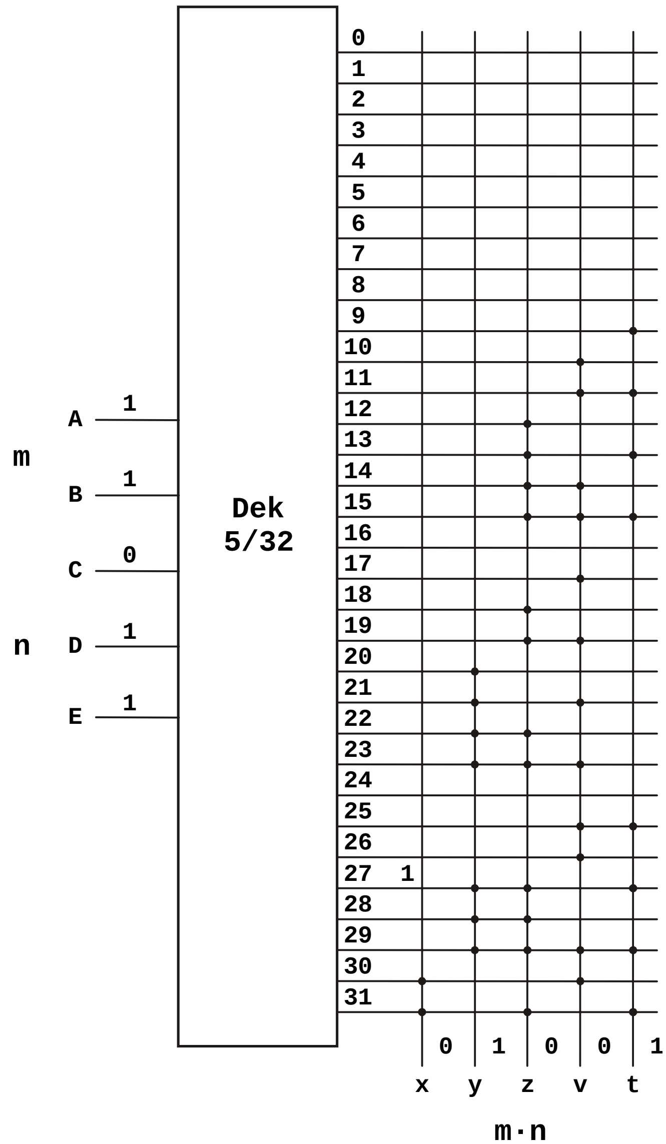


Fig.17.15 Shumëzuesi i numrit 2-bitësh me numër 3-bitësh

Ngjashëm mund të realizohen edhe qarqet e tjera aritmetikore.

Shembull Qarku logjik për gjetjen e kubeve të numrave **3**-bitësh, i realizuar përmes një memorie fikse.

N	A	B	C	x_8	x_7	x_6	x_5	x_4	x_3	x_2	x_1	x_0	N^3
0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	0	0	1	1
2	0	1	0	0	0	0	0	0	1	0	0	0	8
3	0	1	1	0	0	0	0	1	1	0	1	1	27
4	1	0	0	0	0	1	0	0	0	0	0	0	64
5	1	0	1	0	0	1	1	1	1	1	0	1	125
6	1	1	0	0	1	1	0	1	1	0	0	0	216
7	1	1	1	1	0	1	0	1	0	1	1	1	345

$$x_0 = \sum m^1(1,3,5,7)$$

$$x_1 = \sum m^1(3,7)$$

$$x_2 = \sum m^1(5,7)$$

$$x_3 = \sum m^1(2-3,5-6)$$

$$x_4 = \sum m^1(3,5-7)$$

$$x_5 = \sum m^1(5)$$

$$x_6 = \sum m^1(4-7)$$

$$x_7 = \sum m^1(6)$$

$$x_8 = \sum m^1(7)$$

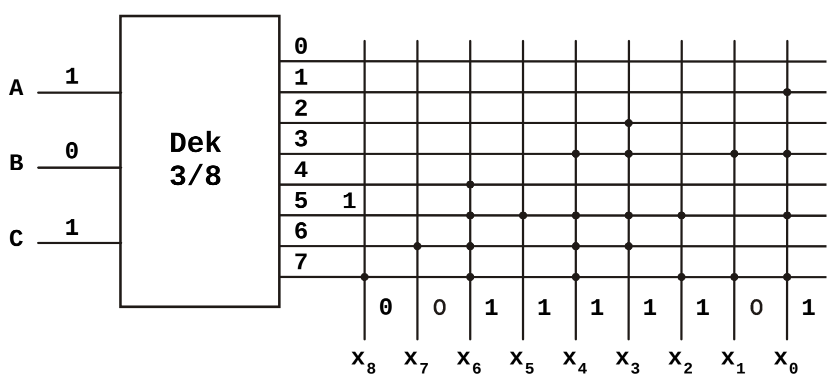


Fig.17.16 Qarku për gjetjen e kubeve të numrave **3***-bitësh*

Në qark, për testim është marrë vlera hyrëse **101**, së cilës i përgjigjet numri decimal **5**. Gjatë kësaj, sinjali me vlerën **1** do të paraqitet në daljen **5** të dekoduesit **3/8**, kurse në dalje të qarkut fitohet numri binar **001111101**, të cilit i përgjigjet numri decimal **125**, përkatësisht kubi i numrit **5**.

Shembull

Qarku logjik përmes të cilit gjinden vlerat numerike të funksionit:

$$z = xy^3 + x^2y + 1$$

nëse **x** dhe **y** paraqesin vlera numerike **2**-bitëshe. Gjatë realizimit të qarkut është përdorur memoria fikse me **16**-fjalë **8**-bitëshe.

N	x		y		z	z						
	A	B	C	D		z_6	z_5	z_4	z_3	z_2	z_2	z_0
0	0	0	0	0	1	0	0	0	0	0	0	1
1	0	0	0	1	1	0	0	0	0	0	0	1
2	0	0	1	0	1	0	0	0	0	0	0	1
3	0	0	1	1	1	0	0	0	0	0	0	1
4	0	1	0	0	1	0	0	0	0	0	0	1
5	0	1	0	1	3	0	0	0	0	0	1	1
6	0	1	1	0	11	0	0	0	1	0	1	1
7	0	1	1	1	31	0	0	1	1	1	1	1
8	1	0	0	0	1	0	0	0	0	0	0	1
9	1	0	0	1	7	0	0	0	0	1	1	1
10	1	0	1	0	25	0	0	1	1	0	0	1
11	1	0	1	1	67	1	0	0	0	0	1	1
12	1	1	0	0	1	0	0	0	0	0	0	1
13	1	1	0	1	13	0	0	0	1	1	0	1
14	1	1	1	0	43	0	1	0	1	0	1	1
15	1	1	1	1	109	1	1	0	1	1	0	1

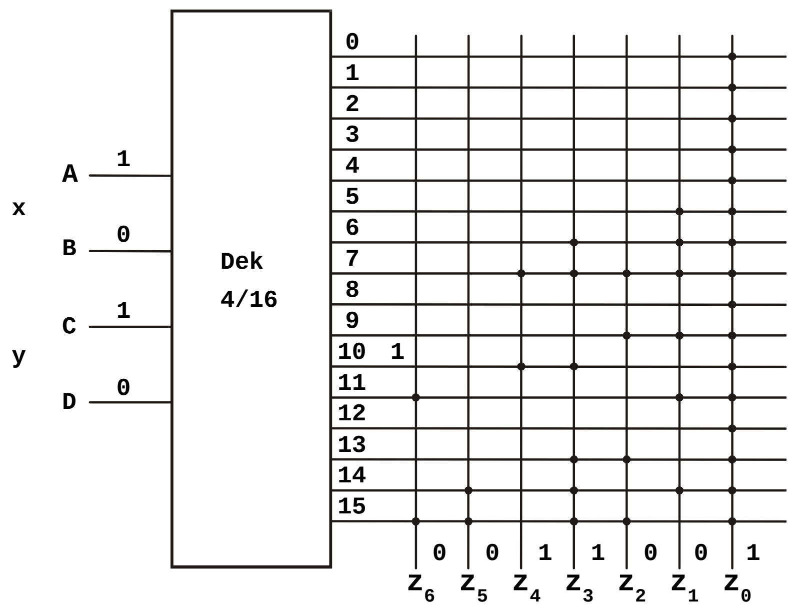


Fig.17.17 Qarku për gjetjen e vlerave të një funksioni

Për ta testuar qarkun e dhënë, në hyrje të tij janë aplikuar vlerat binare **x=10** dhe **y=10**, të cilave u përgjigjet vlera decimale **2**. Vlera e shprehjes në këtë rast është:

$$z = 2 \cdot 2^3 + 2^2 \cdot 2 + 1 = 25$$

Kësaj vlere i përgjigjet ekuivalenti binar **11001**, i cili në memorie është vendosur te fjala me numrin rendor **10**, prej nga lexohet dhe përcillet në dalje të saj.

Qarqe të ndryshme

Duke i shfrytëzuar memoriet fikse, mund të realizohen edhe qarqe të ndryshme, gjë që do të shpjegohet me disa shembuj të qarqeve që jepen në vijim.

Shembull

Qarku logjik përmes të cilit numrit binar hyrës **pABC** i shtohet ose i zbritet numri decimal **2**, varësisht nga ajo se a është numri hyrës negativ ose pozitiv, ku parashenja e numrit përcaktohet me bitin e parë **p**, kështu:

$$\mathbf{p} = \begin{cases} \mathbf{0} & \textit{numër pozitiv} \\ \mathbf{1} & \textit{numër negativ} \end{cases}$$

Gjatë realizimit të qarkut shfrytëzohet memoria fikse me **16**-fjalë **4**-bitëshe, kurse në fjalët dalëse nga memoria **rxyz**, përmes bitit të parë ruhet parashenja e rezultatit.

N	p	A	B	C	r	x	y	z	
0	0	0	0	0	1	0	1	0	0-2=-2
1	0	0	0	1	1	0	0	1	1-2=-1
2	0	0	1	0	0	0	0	0	2-2=0
3	0	0	1	1	0	0	0	1	3-2=1
4	0	1	0	0	0	0	1	0	4-2=2
5	0	1	0	1	0	0	1	1	5-2=3
6	0	1	1	0	0	1	0	0	6-2=4
7	0	1	1	1	0	1	0	1	7-2=5
8	1	0	0	0	0	0	1	0	0+2=2
9	1	0	0	1	0	0	0	1	-1+2=1
10	1	0	1	0	0	0	0	0	-2+2=0
11	1	0	1	1	1	0	0	1	-3+2=-1
12	1	1	0	0	1	0	1	0	-4+2=-2
13	1	1	0	1	1	0	1	1	-5+2=-3
14	1	1	1	0	1	1	0	0	-6+2=-4
15	1	1	1	1	1	1	0	1	-7+2=-5

$$\mathbf{r} = \sum \mathbf{m}^1(\mathbf{0}-\mathbf{1},\mathbf{11}-\mathbf{15})$$

$$\mathbf{x} = \sum \mathbf{m}^1(\mathbf{6}-\mathbf{7},\mathbf{14}-\mathbf{15})$$

$$\mathbf{y} = \sum \mathbf{m}^1(\mathbf{0},\mathbf{4}-\mathbf{5},\mathbf{8},\mathbf{12}-\mathbf{13})$$

$$\mathbf{z} = \sum \mathbf{m}^1(\mathbf{1},\mathbf{3},\mathbf{5},\mathbf{7},\mathbf{9},\mathbf{11},\mathbf{13},\mathbf{15})$$

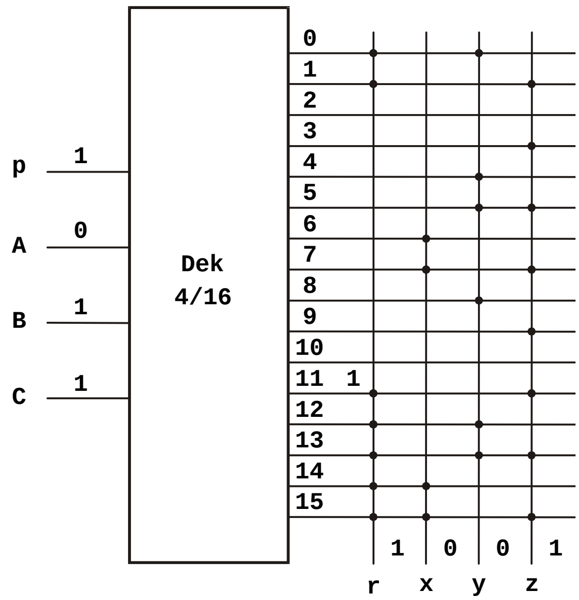


***Fig.17.18 Qarku për shtim ose zbritje të numrit* 2**

Këtu, gjatë përpilimit të tabelës së kombinimeve, janë shfrytëzuar llogaritjet e dhëna pranë saj. Kështu, p.sh., për numrin negativ **-3**, në hyrje të qarkut, pasi t'i shtohet numri pozitiv **2**, kemi:

-3+2=-1

Nga kjo del se për vlerën binare **1011** në hyrje të qarkut, ku shifra e parë e paraqet parashenjën negative të numrit, në dalje të qarkut, nga memoria fikse duhet të merret vlera **-1**, ose vlera binare **1001**, sepse shifra e parë **1** gjithashtu e tregon parashenjën negative të numrit.

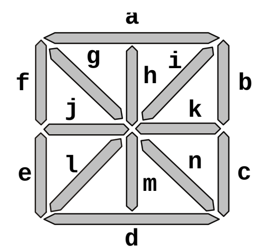
Rezultatet që fitohen gjatë llogaritjeve të ndryshme nuk është e thënë të paraqiten vetëm si vlera numerike. Ato mund të paraqiten, p.sh., përmes **LED**-diodave, ose edhe në indikatorë të ndryshëm - si vlera ose si simbole.

Shembull

Qarku logjik përmes të cilit vlerat numerike të funksionit:

$$y = \begin{cases} x+1 & \textit{për} \;\; z=0 \\ 2x & \textit{për} \;\; z=1 \end{cases}$$

paraqiten në indikatorin **14**-segmentësh, ashtu siç është treguar më poshtë. Për gjenerimin e simboleve të veçanta në indikator, shfrytëzohet memoria fikse me **16**-fjalë **14**-bitëshe.



Indikatori **14** - segmentësh

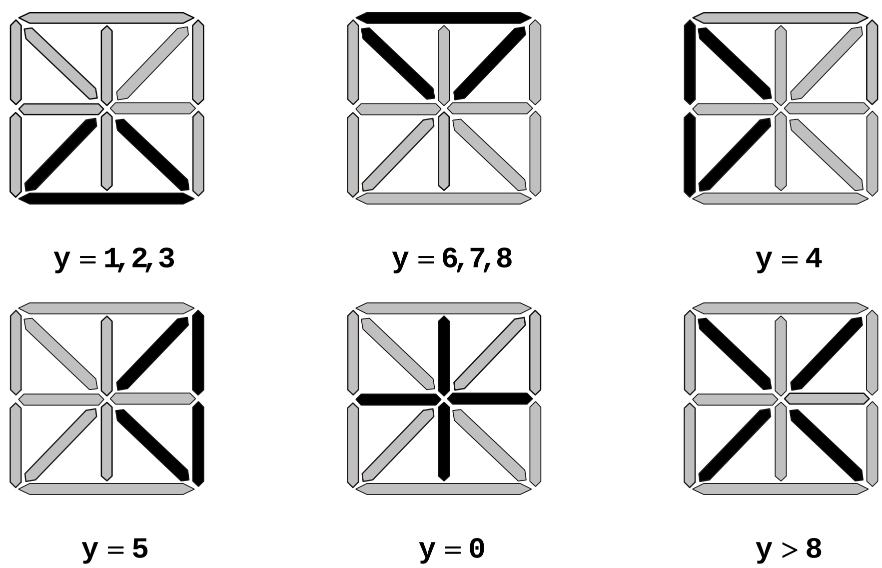


Fig.17.19 Gjenerimi i simboleve në indikatorin 14-segmentësh

N	z	x_2	x_1	x_0	a	b	c	d	e	f	g	h	i	j	k	l	m	n	y
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	1
1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	1	2
2	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	3
3	0	0	1	1	0	0	0	0	1	1	1	0	0	0	0	1	0	0	4
4	0	1	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	1	5
5	0	1	0	1	1	0	0	0	0	0	1	0	1	0	0	0	0	0	6
6	0	1	1	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	7
7	0	1	1	1	1	0	0	0	0	0	1	0	1	0	0	0	0	0	8
8	1	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	1	0	0
9	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	1	2
10	1	0	1	0	0	0	0	0	1	1	1	0	0	0	0	1	0	0	4
11	1	0	1	1	1	0	0	0	0	0	1	0	1	0	0	0	0	0	6
12	1	1	0	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	8
13	1	1	0	1	0	0	0	0	0	0	1	0	1	0	0	1	0	1	10
14	1	1	1	0	0	0	0	0	0	0	1	0	1	0	0	1	0	1	12
15	1	1	1	1	0	0	0	0	0	0	1	0	1	0	0	1	0	1	14

$$a = \sum m^1(5-7, 11-12)$$

$$b = c = \sum m^1(4)$$

$$d = \sum m^1(0-2, 9)$$

$$e = f = \sum m^1(3, 10)$$

$$g = \sum m^1(3, 5-7, 10-15)$$

$$h = j = k = m = \sum m^1(8)$$

$$i = \sum m^1(4-7, 11-15)$$

$$l = \sum m^1(0-3, 8-9, 13-15)$$

$$n = \sum m^1(0-2, 4, 9, 13-15)$$

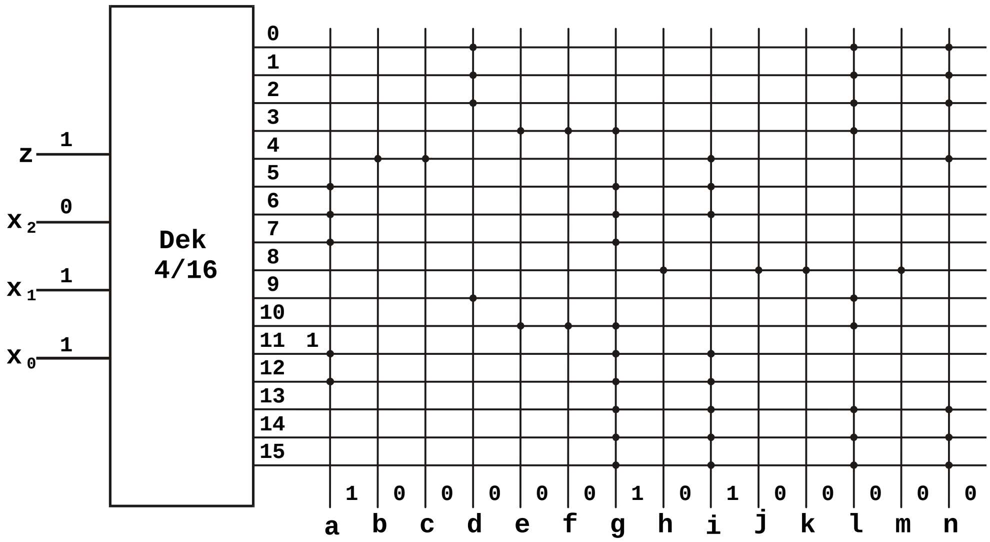


Fig.17.20 Qarku për gjenerimin e simboleve në indikatorin 14-segmentësh

Vlerat e funksionit **y**, të llogaritura përmes shprehjes përkatëse, duke shfrytëzuar vlera numerike decimale, janë dhënë në kolonën e fundit të tabelës. Në bazë të këtyre vlerave, në memorien fikse është programuar gjenerimi i simboleve, ashtu siç është theksuar më sipër. Kështu, p.sh., për vlerat numerike **z=1** dhe **x=011**, funksioni **y=6** dhe gjenerohet simboli i dytë, i cili fitohet nëse ndriçohen segmentet **a**, **g** dhe **i**, përkatësisht nëse vlerat logjike në këto dalje janë **1**.

18

Qarqet që programohen

Prodhuesit e qarqeve të integruara ofrojnë një numër të madh qarqesh logjike, kryesisht në teknologjitë **SSI** (nga Small-Scale Integration) dhe **MSI** (nga Medium-Scale Integration), të cilat mund të përdoren si të gatshme, vetëm ose të kombinuara edhe me qarqe të tjera. Por, procedura e krijimit të pllakës ku ato vendosen, e ngjitjes si dhe e testimit të qarqeve është një punë e mundimshme. Për tejkalimin e problemeve të kësaj natyre, në teknologjitë **LSI** (nga Large-scale Integration) dhe **VLSI** (nga Very Large-Scale Integration) janë prodhuar qarqe të integruara me më shumë funksione standarde në një çip të vetëm, të cilët përmbajnë qindra dhe mija elemente logjike, të lidhura në një tërësi funksionale. Të tillë, p.sh., janë çipat për kalkulatorë, lojëra të ndryshme, kujtesa, sintetizim të zërit si dhe mikroprocesorët.

PLD

Kërkesat e prodhuesve shpeshherë janë të atilla që mund të gjenden zgjidhje optimale nëse shfrytëzohen elemente logjike të qarqeve të integruara standarde. Për këtë qëllim janë prodhuar qarqe të integruara, të cilat përmbajnë një numër të madh elementesh logjike, përfshirë këtu edhe elemente memoruese, të cilat janë të lidhura mes vete brenda çipit. Por, disa prej lidhjeve janë të atilla që sipas nevojës mund të këputen, ngjashëm siç bëhet programimi i kujtesave fikse. Qarqet e tilla njihen si *pajisje logjike që programohen* (ang. Programmable Logic Device, **PLD**), sepse për realizimin e funksioneve përmes tyre duhet të përcaktohen lidhjet në mes të elementeve *që këputen* dhe lidhjet *që ngelin* të paképutura. Procesi i përcaktimit të tillë të lidhjeve në mes të elementeve të qarkut quhet *programim* dhe mund të bëhet nga prodhuesi, ose edhe nga shfrytëzuesi i qarkut të integruar.

Shembull

Forma e përgjithshme e **PLD**-së, me një fushë **DHE** e cila përmban **4** elemente logjike **DHE**, si dhe një fushë **OSE** me **4** elemente logjike **OSE**.

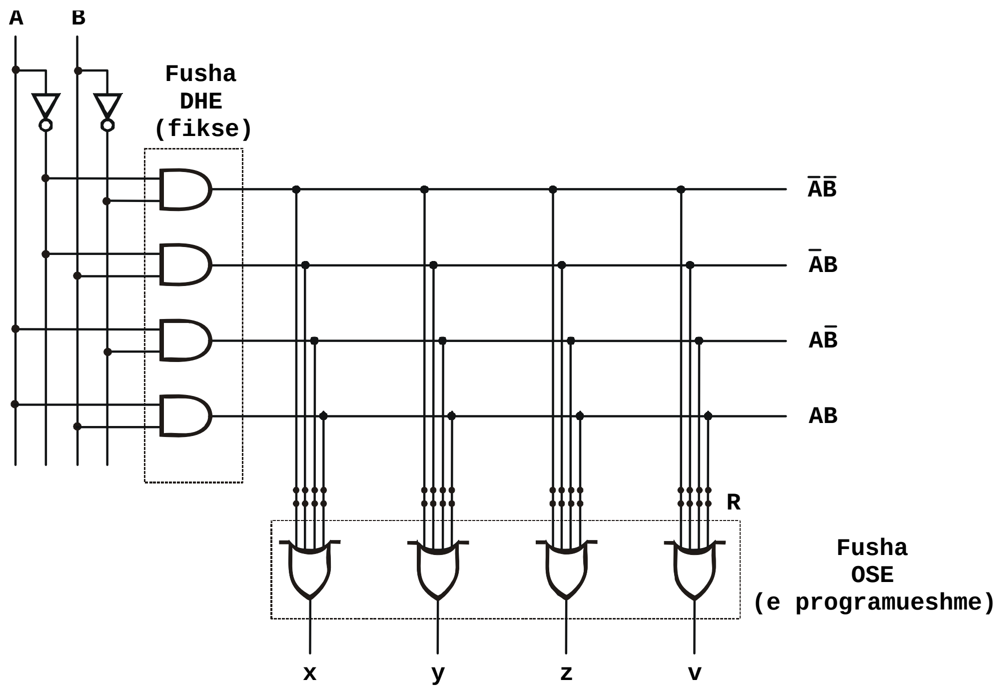


Fig.18.1 Forma e përgjithshme e ***PLD****-së*

Te qarku i dhënë fusha **DHE** është fikse, kurse fusha **OSE** mund të programohet duke i eliminuar lidhjet **R** në hyrjet e elementeve logjike **OSE**. Fillimisht, kur ekzistojnë të gjitha lidhjet e vizatuara në hyrjet e elementeve logjike **OSE**, në **4** daljet e qarkut **x**, **y**, **z** dhe **v**, merren vlerat logjike **1**, gjë që mund të vërtetohet, p.sh., për daljen **x**, kështu:

$$\begin{aligned} x &= \overline{A}\,\overline{B} + \overline{A}\,B + A\,\overline{B} + A\,B \\ &= \overline{A}(\overline{B} + B) + A(\overline{B} + B) \\ &= \overline{A} + A = 1 \end{aligned}$$

PLD-të janë disajnuar ashtu që lidhjet e këputura të elementeve logjike **OSE** sillen sikur në hyrjet përkatëse të jenë aplikuar vlerat logjike **0**. Kështu, p.sh., për ta programuar daljen **x**, ashtu që në të të merret vlera e funksionit logjik:

$$x = \overline{A}\,\overline{B} + A\,B$$

në hyrje të elementit logjik **OSE** përkatës duhet të këputet lidhja e dytë dhe lidhja e tretë, përkatësisht dy lidhjet e mesme, sepse në atë rast kemi:

$$\begin{aligned} x &= \overline{A}\,\overline{B} + 0 + 0 + A\,B \\ &= \overline{A}\,\overline{B} + A\,B \end{aligned}$$

Plotësisht njëlloj mund të programohen edhe **3** elementet logjike të tjera, për të fituar edhe **3** funksione dalëse. Pas programimit, qarku me funksionet e programuara mund të shfrytëzohet sikur edhe qarqet e realizuara me elemente logjike të zakonshme.

Shembull

Qarku logjik përmes të cilit gjenden vlerat e funksionit:

$$x = \overline{A} + A\overline{B}$$

i realizuar duke programuar daljen e parë të **PLD**-së, i cili në formë të përgjithshme u dha më sipër.

A	B	$A\overline{B}$	x
0	0	0	1
0	1	0	1
1	0	1	1
1	1	0	0

$$x = \overline{A}\,\overline{B} + \overline{A}B + A\overline{B}$$

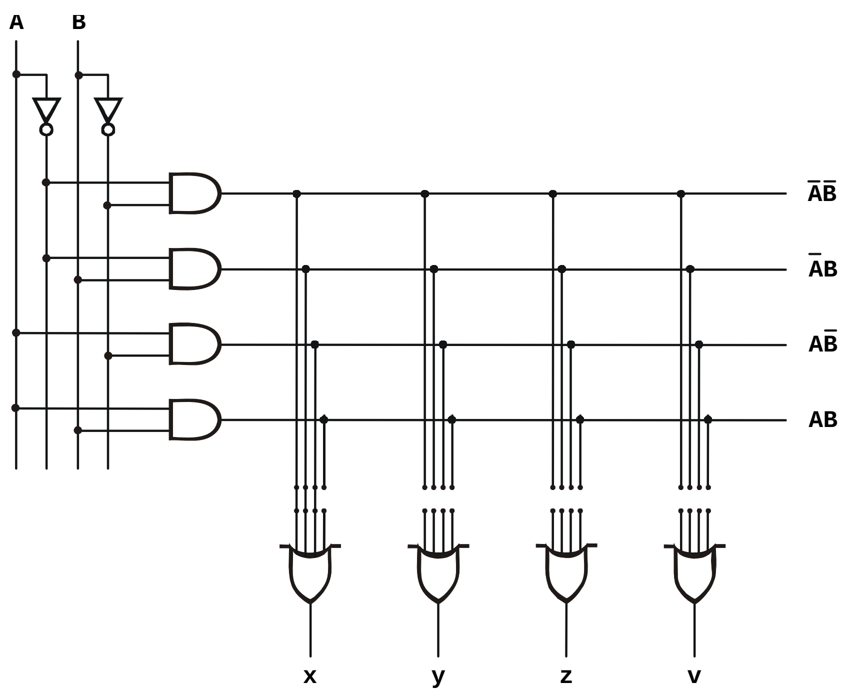


Fig.18.2 Programimi i PLD-së në bazë të funksionit të dhënë

Para se të vizatohej qarku i mësipërm, është përpiluar tabela përkatëse e kombinimeve, prej nga pastaj funksioni është shprehur përmes shumës së mintermave me vlerën **1**. Në bazë të shprehjeve të fituara është programuar **PLD**-ja, duke i lënë të pakëputura vetëm **3** lidhjet e para të hyrjeve në elementin logjik **OSE**, i cili i përket funksionit **x**. Janë këputur edhe lidhjet në të gjitha hyrjet e **3** elementeve logjike **OSE**, sepse në këtë qark programohet vetëm një funksion.

Që të mos vizatohet komplet qarku, qarku i dhënë në Fig.18.1 në formën e tij të përgjithshme, vizatohet si në Fig.18.3.

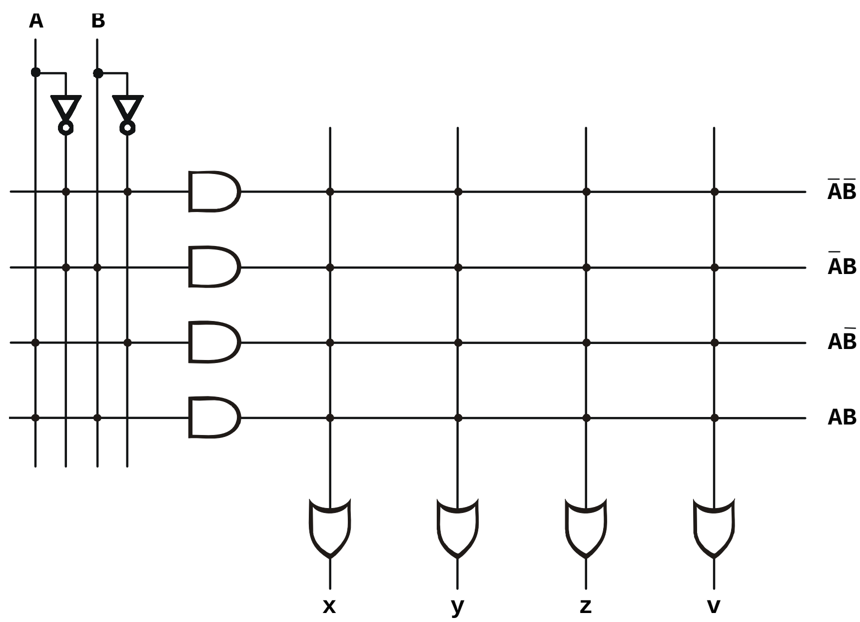


Fig.18.3 Paraqitja e thjeshtuar e formës së përgjithshme të PLD-së

Kështu, **PLD**-ja e programuar te shembulli i mësipërm, që është dhënë në Fig.18.2, më thjeshtë mund të paraqitet si në Fig.18.4.

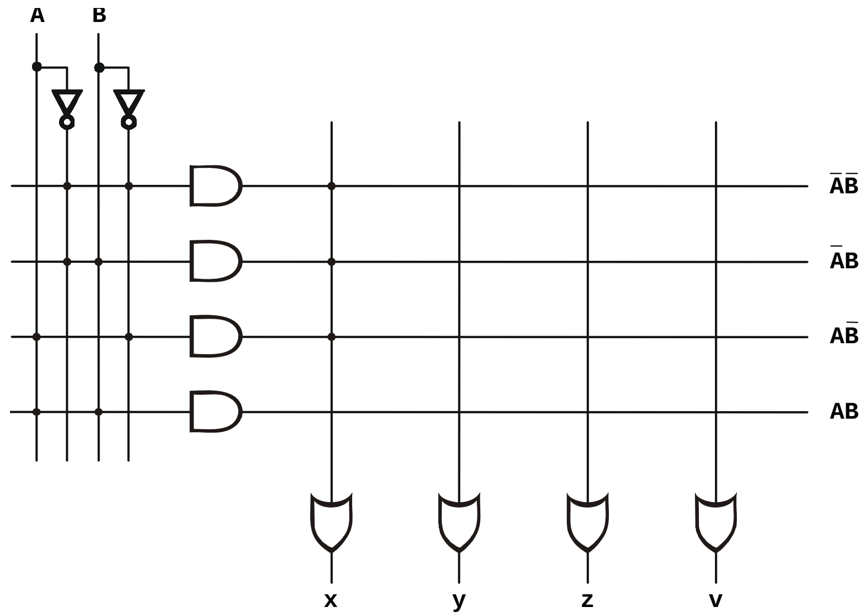


Fig.18.4 Paraqitja e thjeshtuar e qarkut në Fig.18.2

Në figurën e dhënë shihet qartë se të pakëputura kanë ngelur vetëm **3** lidhjet e para të elementit të parë **OSE**, gjë e cila tregohet me **3** pikat e vizatuara.

Kjo formë e paraqitjes së thjeshtuar të qarqeve të realizuara përmes **PLD**-ve do të shfrytëzohet edhe në pjesën vijuese të librit.

Shembull Qarku logjik i mbledhësit të plotë, i realizuar duke programuar një **PLD**.

N	X	Y	Z	s	b
0	0	0	0	0	0
1	0	0	1	1	0
2	0	1	0	1	0
3	0	1	1	0	1
4	1	0	0	1	0
5	1	0	1	0	1
6	1	1	0	0	1
7	1	1	1	1	1

$$s = \sum m^1(1-2,4,6)$$

$$b = \sum m^1(3,5-7)$$

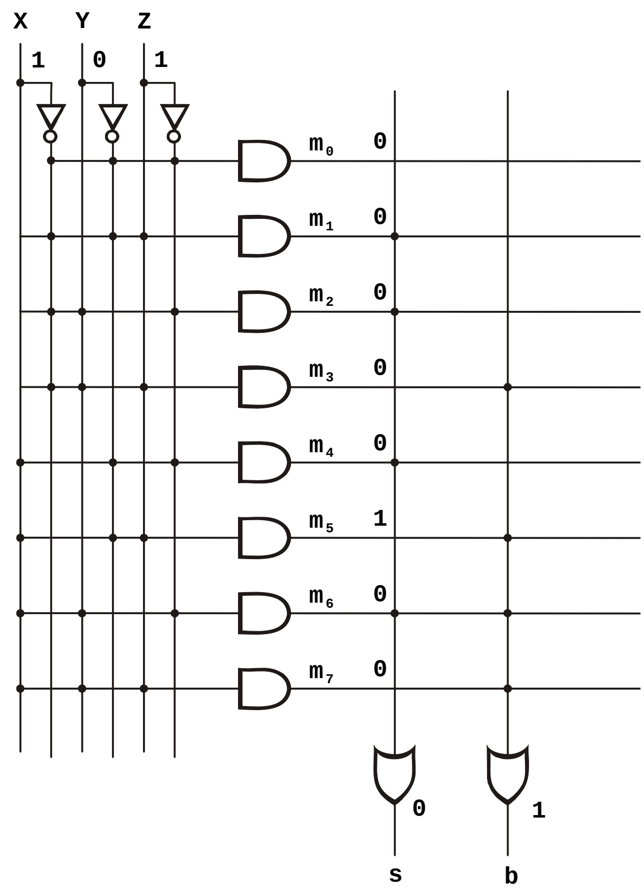


***Fig.18.5 Mbledhësi i plotë i realizuar me* `PLD`**

Në qarkun e dhënë në Fig.18.5 variablat e shfrytëzuara janë:

X, **Y** - numrat që mblidhen
Z - bartja hyrëse
s - shuma
b - bartja.

Nëse në hyrje të qarkut aplikohen vlerat **X=1**, **Y=0** dhe **Z=1**, vetëm në daljen m_5 të fushës **DHE** do të gjenerohet sinjali me vlerën **1**, i cili pastaj përmes fushës **OSE** përcillet në daljen **b** të qarkut. Nga rezultati i fituar në dalje shihet se qarku funksionon si mbledhës i plotë.

Shembull

Konvertuesi i kodit **BCD 2421** në kodin **BCD 4221**, duke shfrytëzuar një **PLD**.

N	2421				4221			
	A	B	C	D	x	y	z	v
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	0
3	0	0	1	1	0	0	1	1
4	0	1	0	0	0	1	1	0
5	0	1	0	1	+	+	+	+
6	0	1	1	0	+	+	+	+
7	0	1	1	1	+	+	+	+
8	1	0	0	0	+	+	+	+
9	1	0	0	1	+	+	+	+
10	1	0	1	0	+	+	+	+
11	1	0	1	1	0	1	1	1
12	1	1	0	0	1	1	0	0
13	1	1	0	1	1	1	0	1
14	1	1	1	0	1	1	1	0
15	1	1	1	1	1	1	1	1

$$x = \sum m^1(12-15)$$

$$y = \sum m^1(4, 11-15)$$

$$z = \sum m^1(2-4, 11, 14-15)$$

$$v = \sum m^1(1, 3, 11, 13, 15)$$

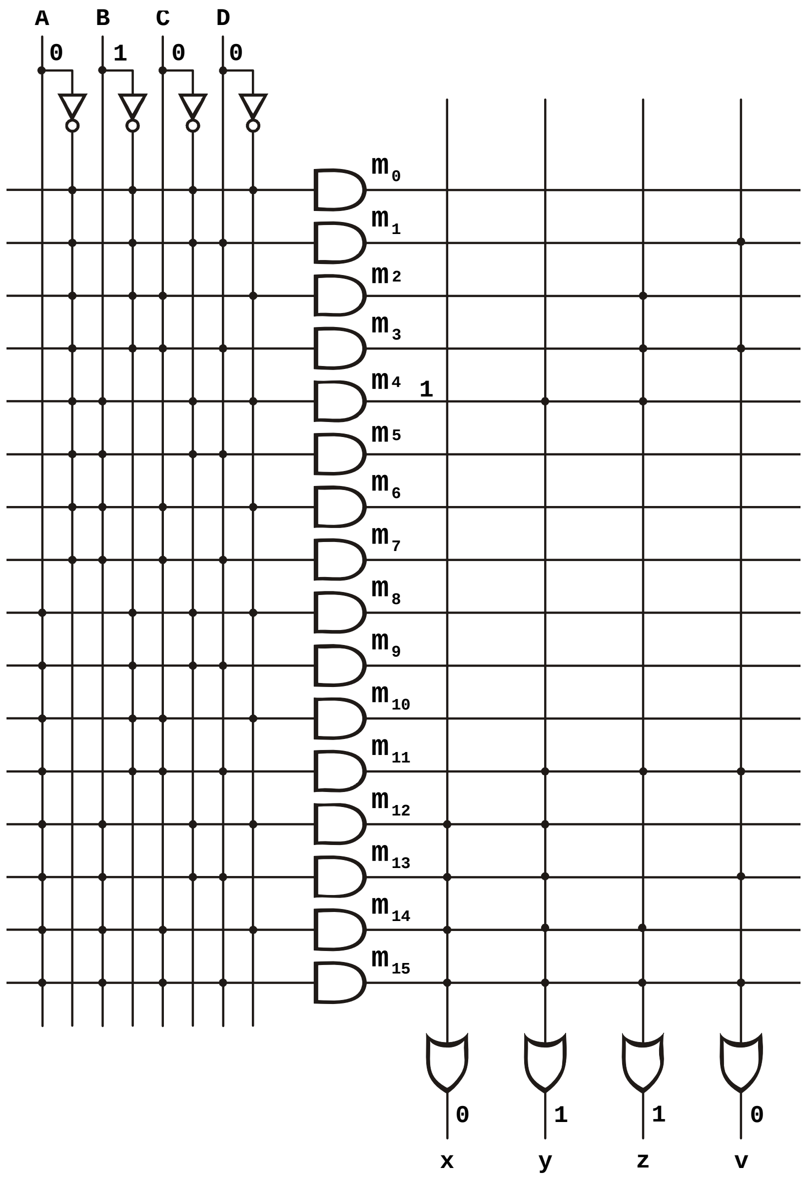


Fig.18.6 Konvertuesi i kodit **2421** *në kodin* **4221**

Për fjalët kodike të cilat nuk i takojnë kodit **BCD 2421**, në kodin **BCD 4221** janë marrë vlera arbitrare, të cilat nuk shfrytëzohen aspak gjatë programimit të **PLD**-së. Për ta testuar funksionimin e qarkut të dhënë në Fig.18.6, në hyrje të tij është aplikuar fjala kodike **0100**. Si rezultat, sinjali me vlerën **1** paraqitet në daljen m_4 të fushës **DHE**, e cila pastaj përcillet në daljet **y** dhe **z** të qarkut, përkatësisht në dalje të tij fitohet fjala kodike **0110** e kodit **BCD 4221**.

PAL

Në rastin e përgjithshëm, siç mund të shihet edhe në shembujt e qarqeve **PLD**, shumica e daljeve të fushës **DHE** nuk shfrytëzohen. Prandaj, janë prodhuar qarqe të integruara me elemente logjike të shumta, si edhe te **PLD**-të, por te të cilat lidhjet e fushës **OSE** janë *fikse*, kurse lidhjet e fushës **DHE** janë *të programueshme*. Qarqet e tilla **PLD** njihen si **PAL** (nga Programmable Array Logic). **PAL**-i në formën e tij të përgjithshme, p.sh., mund të duket si në Fig.18.7.

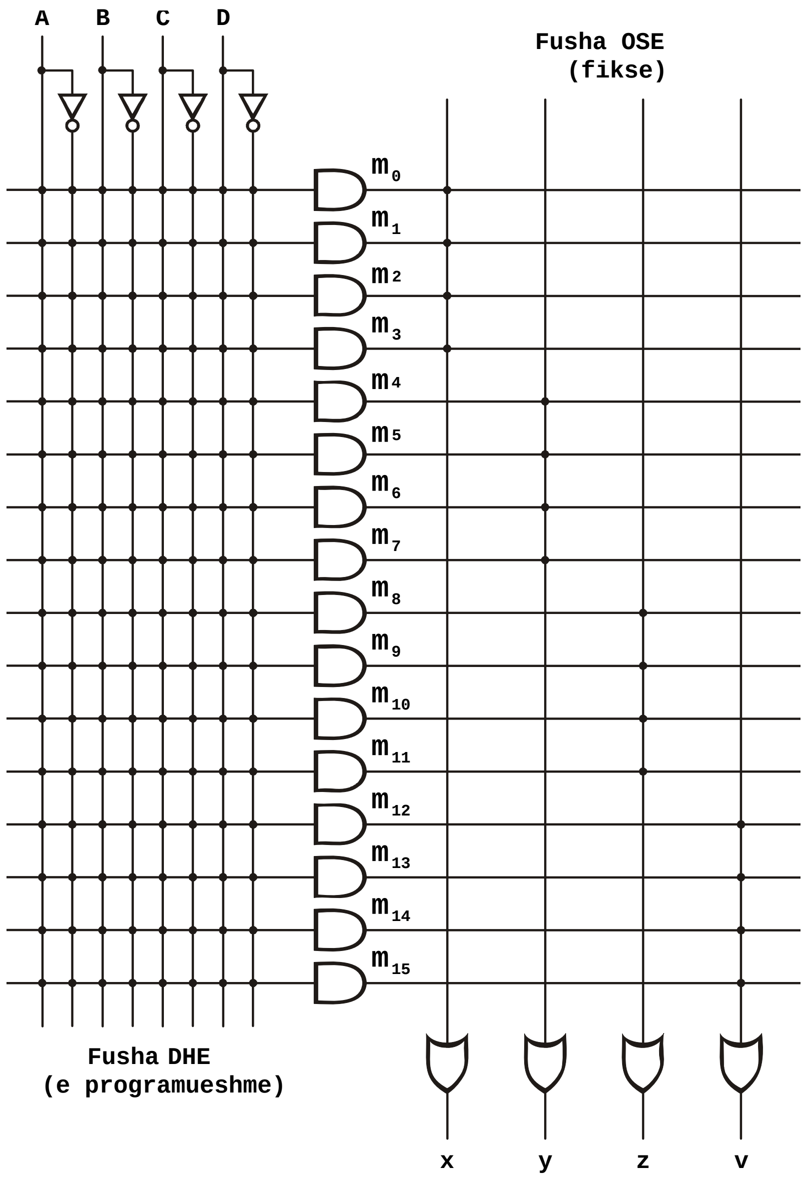


Fig.18.7 Forma e përgjithshme e PAL-it

Nga forma e përgjithshme e **PAL**-it shihet se në fushën **DHE** fillimisht ekzistojnë të gjitha lidhjet. Gjatë procesit të programimit, në këtë fushë eliminohen lidhjet e panevojshme, njëlloj siç programohet fusha **OSE** te **PLD**-ja. Lidhjet në fushën **OSE** janë fikse, gjë që duhet të merret parasysh gjatë programimit të fushës **DHE**.

Shembull

Qarku përmes të cilit gjenden vlerat e funksioneve:

$$x = \overline{A}B + AC + \overline{B}CD$$

$$y = B + \overline{C}D + ABC\overline{D}$$

$$z = A\overline{C} + BC\overline{D}$$

$$v = \overline{A}B\overline{C} + AC + B\overline{D}$$

i realizuar përmes një **PAL**-i.

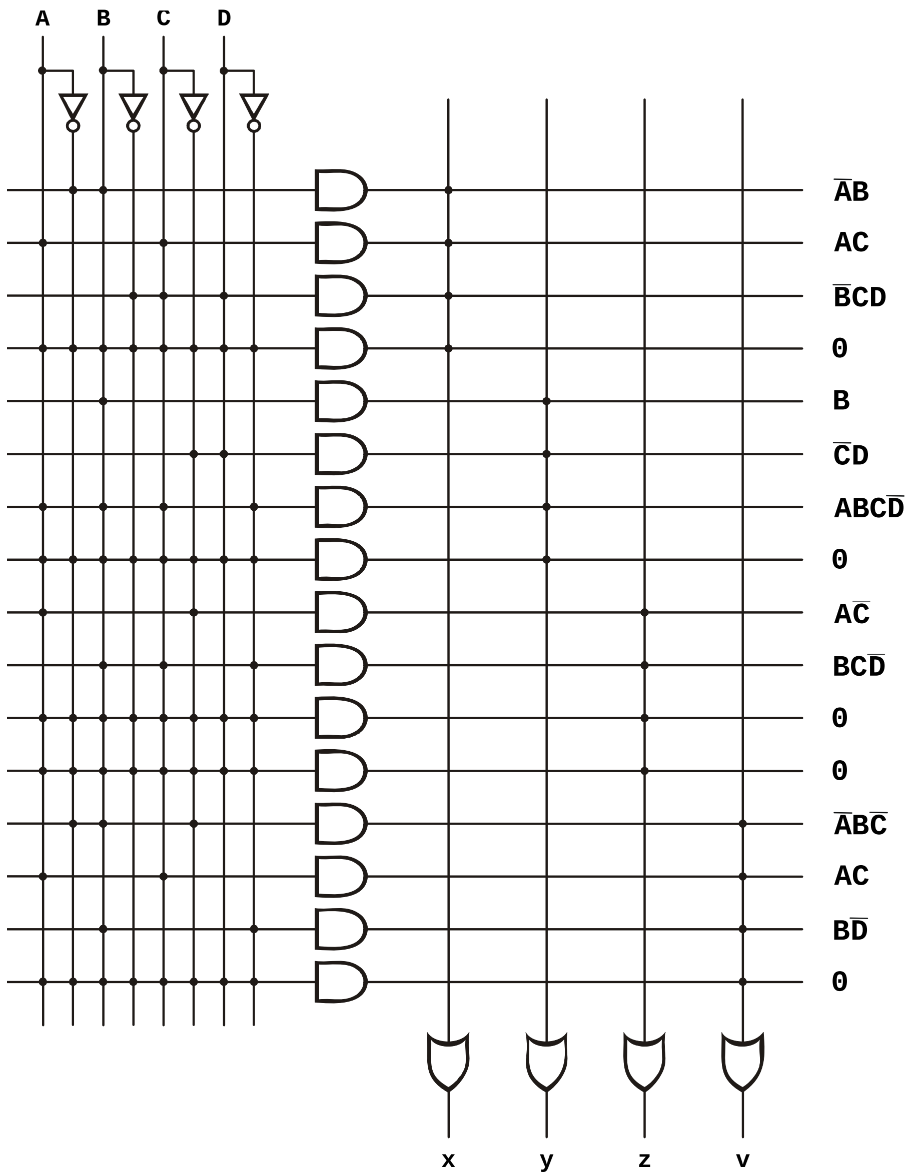


Fig.18.8 Qarku për gjetjen e vlerave të disa funksioneve

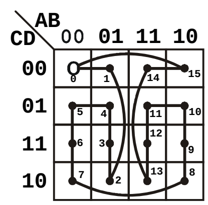
Siç shihet edhe nga realizimi i qarkut në Fig.18.8, për dallim nga qarqet me **PLD**, këtu nuk nevojitet që funksionet të shprehen përmes shumës së mintermave të plota, por gjatë programimit mintermat merren ashtu siç paraqiten në shprehjet e funksioneve. Tek elementet **DHE** të cilat nuk përdoren, të gjitha lidhjet kanë ngelur të pakëputura, për çka në dalje të tyre merret vlera logjike **0**, sepse:

$$A\bar{A}B\bar{B}C\bar{C}D\bar{D} = 0$$

Nga kjo që u dha më sipër shihet se programimi i **PAL**-it gjatë realizimit të qarqeve të ndryshme mund të bëhet pasi të gjenden paraprakisht shprehjet minimale të funksioneve me të cilat ato përshkruhen.

Shembull

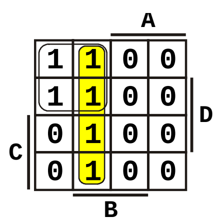
Qarku logjik i realizuar me një ***PAL***, ashtu që në njërën nga **3** daljet e tij merret vlera logjike **1**, varësisht nga vlera e madhësisë **x**, e cila vjen prej një instrumenti matës, e shprehur në formën binare **ABCD**, në kodin ciklik të dhënë me tabelën:



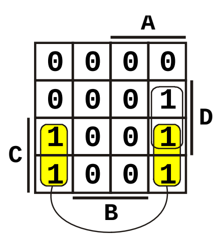
Dalja e qarkut me vlerën logjike **1** përcaktohet kështu:

$$d_1 = 1, \text{ nëse } 0 \le x \le 5$$
$$d_2 = 1, \text{ nëse } 6 \le x \le 10$$
$$d_3 = 1, \text{ nëse } 11 \le x \le 15$$

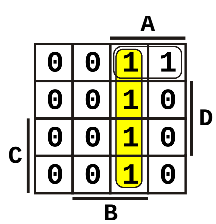
x	A	B	C	D	d_1	d_2	d_3
0	0	0	0	0	1	0	0
1	0	1	0	0	1	0	0
2	0	1	1	0	1	0	0
3	0	1	1	1	1	0	0
4	0	1	0	1	1	0	0
5	0	0	0	1	1	0	0
6	0	0	1	1	0	1	0
7	0	0	1	0	0	1	0
8	1	0	1	0	0	1	0
9	1	0	1	1	0	1	0
10	1	0	0	1	0	1	0
11	1	1	0	1	0	0	1
12	1	1	1	1	0	0	1
13	1	1	1	0	0	0	1
14	1	1	0	0	0	0	1
15	1	0	0	0	0	0	1



$$d_1 = \bar{A}B + \bar{A}C$$



$$d_2 = A\bar{B}D + \bar{B}C$$



$$d_3 = AB + A\bar{C}\bar{D}$$

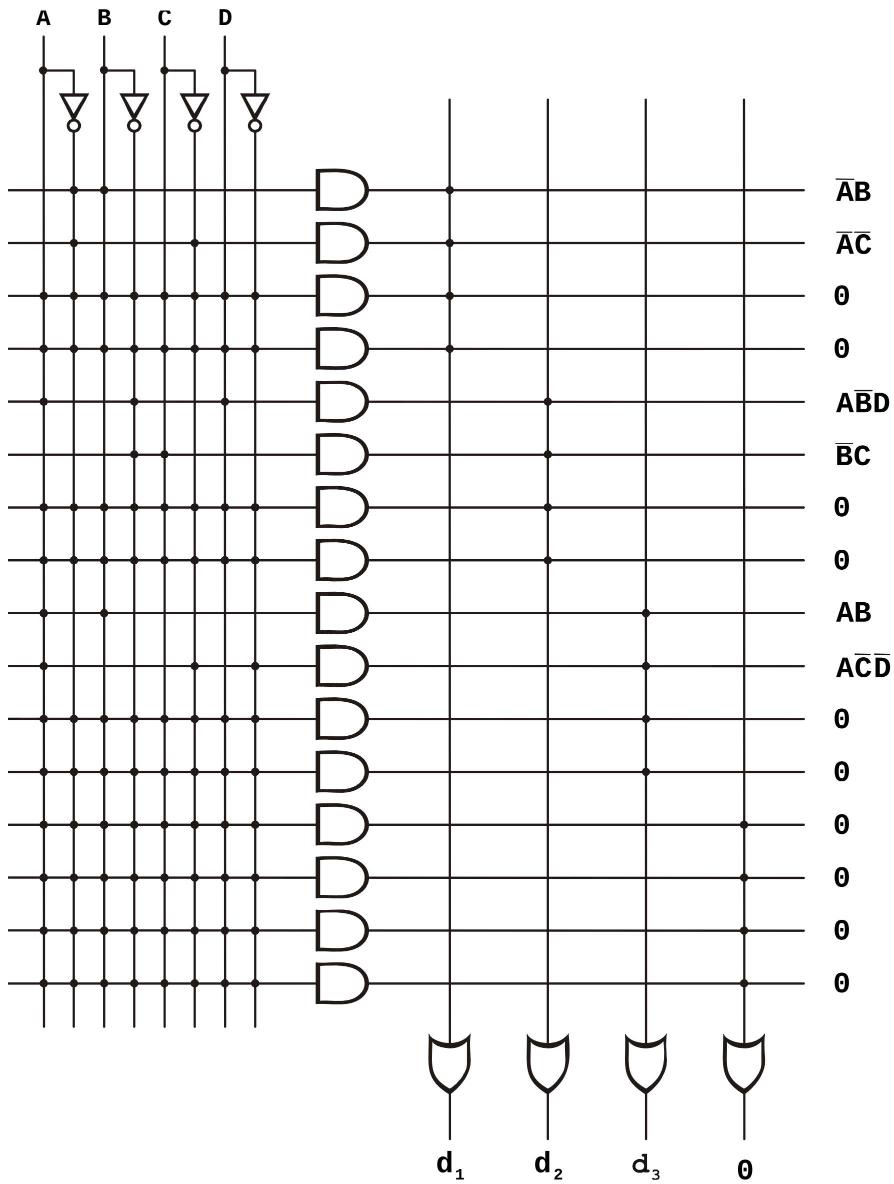


Fig.18.9 Programimi i* PAL*-it pas minimizimit të funksioneve

Gjatë programimit të **PAL**-it të shfrytëzuar në Fig.18.9, janë përdorur vetëm **3** elementet e para **OSE**, kurse elementi i fundit nuk përdoret aspak, përkatësisht vlera në dalje të tij mbetet **0**, sepse pjesa e fushës **DHE**, që i përket këtij elementi, nuk programohet.

PLA

Si zgjidhje më optimale e pajisjeve logjike që programohen janë **PLA**-të (nga Programmable Logic Array), te të cilat mund të programohet njëkohësisht fusha **DHE** dhe fusha **OSE**. Arkitektura e brendshme e **PLA**-së, në formën e saj të përgjithshme, është dhënë në Fig.18.10.

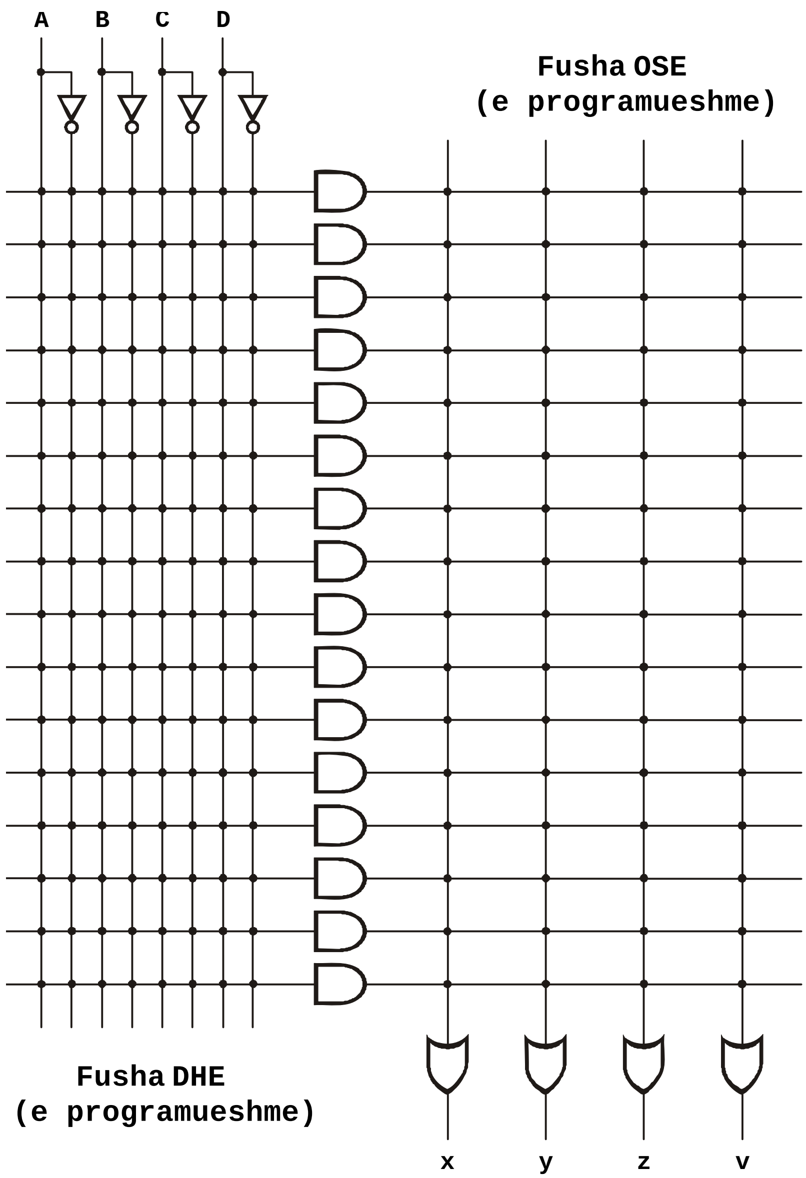


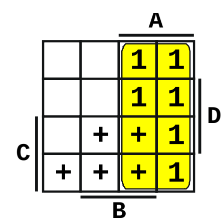
Fig.18.10 Forma e përgjithshme e ***PLA****-së*

Siç shihet nga vizatimi i dhënë, fillimisht ekzistojnë të gjitha lidhjet e mundshme, si në fushën **DHE** ashtu edhe në fushën **OSE**. Gjatë programimit të dy fushave eliminohen lidhjet e tepërta, përkatësisht zgjedhja e lidhjeve që duhet të ngelin është e lirë.

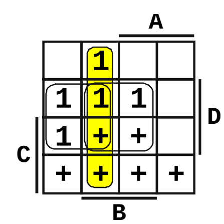
Shembull

Qarku logjik i konvertuesit të fjalëve kodike të kodit **BCD 5311**, në fjalë të kodit **BCD 84-2-1**, i realizuar përmes **PLA**-së.

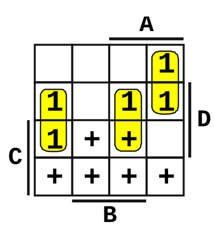
N	5311				84-2-1			
	A	B	C	D	x	y	z	v
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	1	1	1
2	0	0	1	0	+	+	+	+
3	0	0	1	1	0	1	1	0
4	0	1	0	0	0	1	0	1
5	0	1	0	1	0	1	0	0
6	0	1	1	0	+	+	+	+
7	0	1	1	1	+	+	+	+
8	1	0	0	0	1	0	1	1
9	1	0	0	1	1	0	1	0
10	1	0	1	0	+	+	+	+
11	1	0	1	1	1	0	0	1
12	1	1	0	0	1	0	0	0
13	1	1	0	1	1	1	1	1
14	1	1	1	0	+	+	+	+
15	1	1	1	1	+	+	+	+



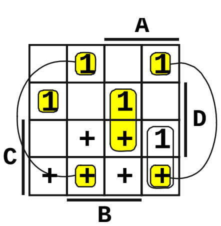
$$x = A$$



$$y = \overline{A}B + BD + \overline{A}D$$



$$z = A\overline{B}\,\overline{C} + ABD + \overline{A}\,\overline{B}D$$



$$v = A\overline{B}\,\overline{D} + A\overline{B}C + ABD + \overline{A}B\overline{D} + \overline{A}\,\overline{B}\,\overline{C}D$$

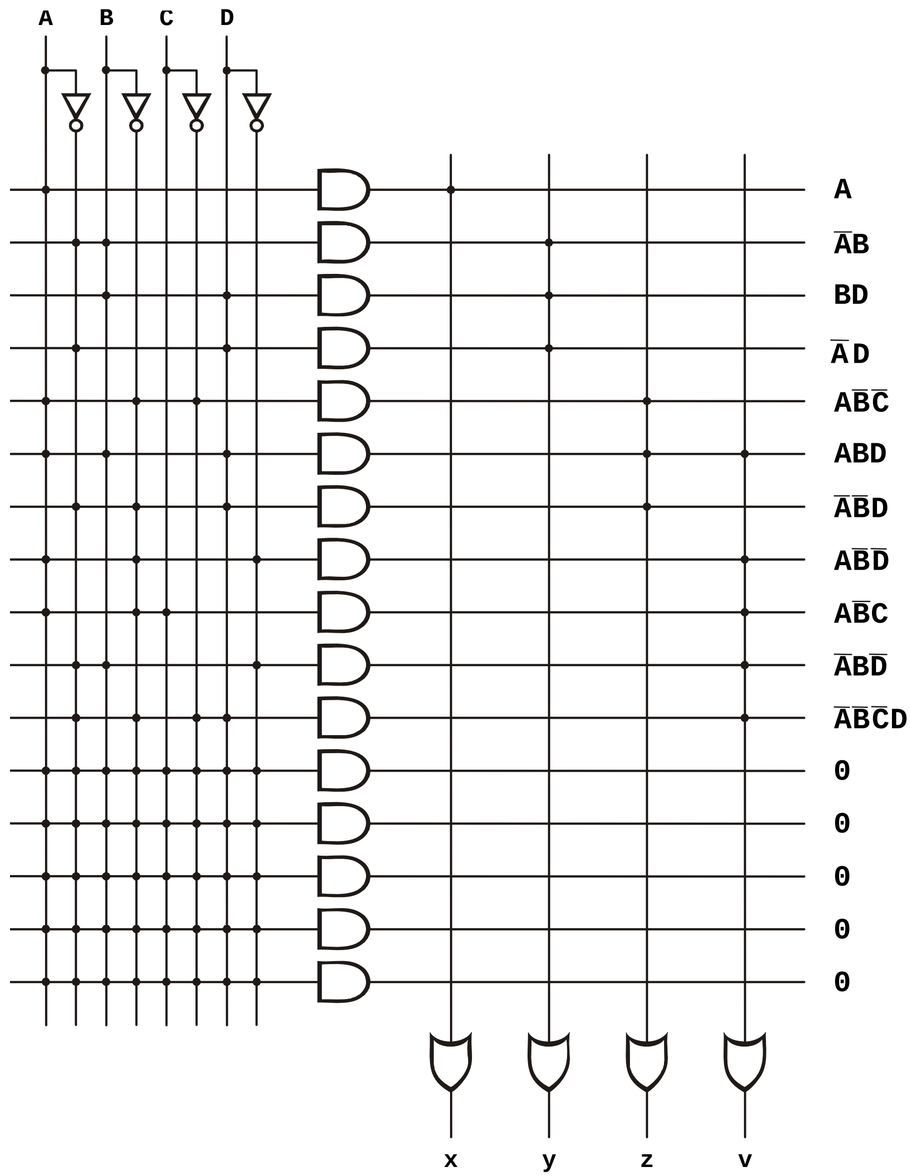


Fig.18.11 Konvertuesi i kodit* 5311 *në kodin* 84-2-1 *i realizuar përmes* PLA*-së

Radha e programimit të mintermave në pjesën e fushës **DHE** nuk ka rëndësi, sepse shfrytëzimi i tyre gjatë programimit të fushës **OSE** i përcakton funksionet dalëse, përkatësisht mintermat të cilat marrin pjesë në formimin e shprehjeve të tyre.

Për dallim nga pajisjet e tjera logjike të cilat programohen, **PLA**-të janë më të ndërlikuara, si gjatë prodhimit, ashtu edhe gjatë programimit, sepse përmbajnë dy fusha që programohen.

Numri i hyrjeve në **PLA**, si dhe numri i elementeve logjike brenda tyre mund të jetë i madh. P.sh., **PLA**-të mund të kenë prej **10** deri në **20** hyrje, **30** deri në **60** elemente logjike **DHE** dhe **10** deri në **20** elemente logjike **OSE**. Kështu, me një **PLA** mund të realizohen më shumë funksione, për një ose edhe më shumë qarqe të ndryshme.

PLS

Qarqet me fusha që programohen mund të përmbajnë edhe *elemente memoruese*, *regjistra hyrës* dhe *regjistra dalës*. Qarqet e tilla shkurt quhen **PLS** (nga Programmable Logic Sequencer) dhe përdoren për realizimin e qarqeve kompjuterike komplekse.

Dalje të invertuara

Me qëllim të invertimit të vlerave dalëse, në daljet e qarqeve logjike me fusha që programohen, siç janë ato **PLD**, **PAL** dhe **PLA**, vendosen elemente logjike speciale **EX-OSE**. Në njërën hyrje të këtyre elementeve lidhen daljet nga qarku, kurse hyrja tjetër **T** e tyre *tokësohet*, por me mundësi të programimit, ashtu siç shihet në shembullin e qarkut **PLD** me fushë të programueshme për invertimin e daljeve, i cili është dhënë në Fig.18.12.

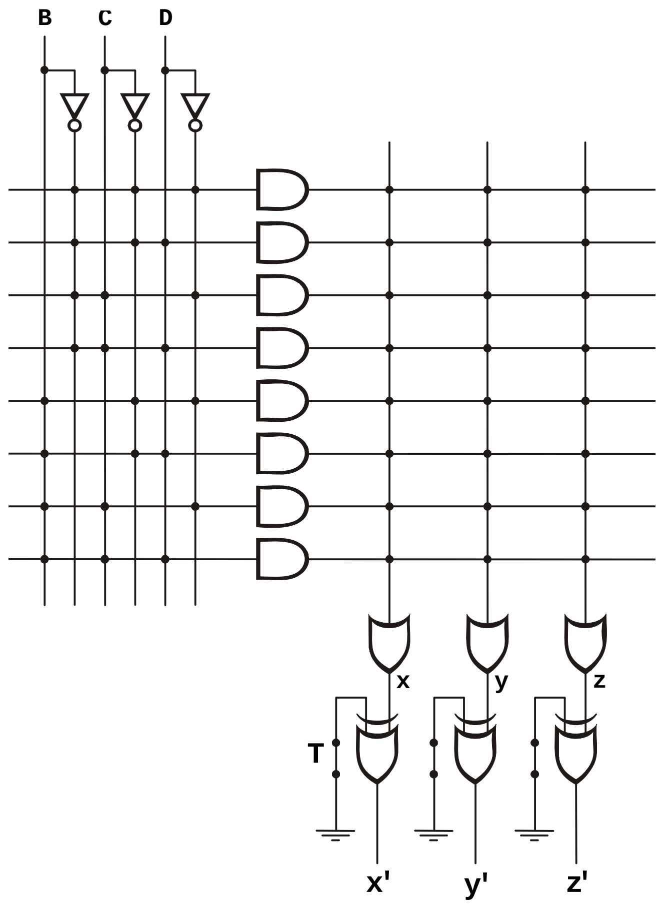


Fig.18.12 Qaku* PLD *te i cili mund të programohet invertimi i daljeve

Nëse njëra hyrje e elementeve logjike **EX-OSE** ngel e tokësuar, gjatë së cilës vlera e sinjalit përkatës në këtë hyrje është **0**, në dalje të tyre do të merren vlerat dalëse të qarkut, të pandryshueshme. Kështu, p.sh., për daljen **x'** kemi:

$$
\begin{aligned}
\mathbf{x'} &= \mathbf{0} \oplus \mathbf{x} \\
&= \overline{\mathbf{0}} \cdot \mathbf{x} + \mathbf{0} \cdot \overline{\mathbf{x}} \\
&= \mathbf{1} \cdot \mathbf{x} + \mathbf{0} \\
&= \mathbf{x}
\end{aligned}
$$

Por, nëse gjatë programimit hyrja **T** shkëputet nga tokësimi, vlera e sinjalit hyrës përkatës në elementin **EX-OSE** është **1**, kurse në dalje të tij fitohet vlera e invertuar e daljes, p.sh., kështu:

$$
\begin{aligned}
x' &= 1 \oplus x \\
&= \overline{1} \cdot x + 1 \cdot \overline{x} \\
&= 0 \cdot x + \overline{x} \\
&= \overline{x}
\end{aligned}
$$

Programimi

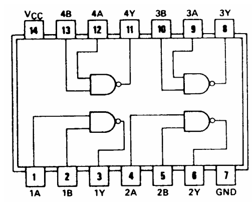
Për programimin e qarqeve me fusha që programohen shfrytëzohen pajisje universale për programim, përmes së cilave mund të programohen **PROM**-et, **PAL**-et dhe **FPLA**-të. Përgatitja e të dhënave të cilat shfrytëzohen për programim bëhet në kompjuter, duke shfrytëzuar softverin adekuat.

Gjatë programimit të më shumë **PLA**-ve për një zbatim të caktuar shfrytëzohen *matrica programuese*, për çka **PLA**-të e tilla shpesh quhen edhe **FPLA** (nga Field-Programmable Logic Arrays).

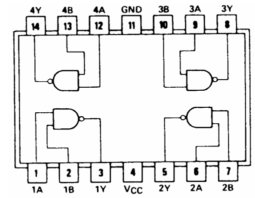
Janë prodhuar edhe qarqe me fusha që programohen, te të cilat mund të fshihet programi ekzistues dhe të bëhet riprogramimi i tyre, plotësisht njëlloj siç fshihen dhe programohen **EEPROM**-et. Qarqet e tilla njihen me shkurtesën **EPLD** (nga Erasable Programmable Logic Device).

Shtesë

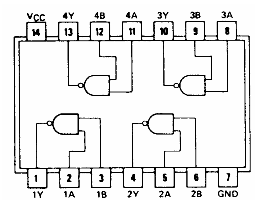
Qarqe të integruara të familjes 54/74



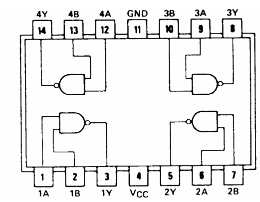
00A



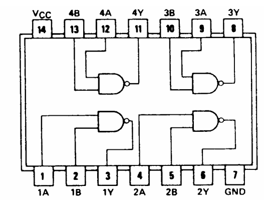
00B



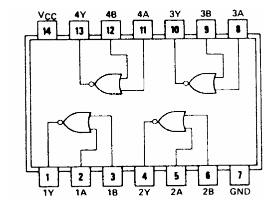
01A



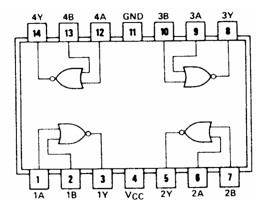
01B



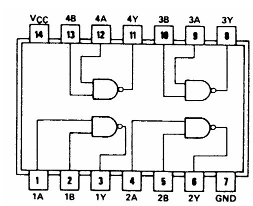
01C



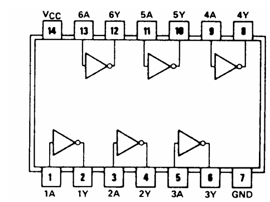
02A



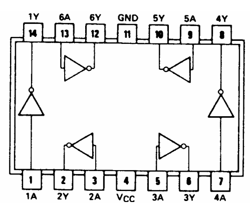
02B



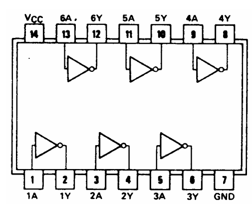
03



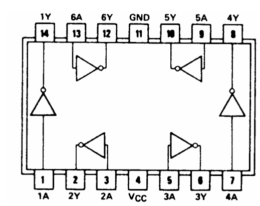
04A



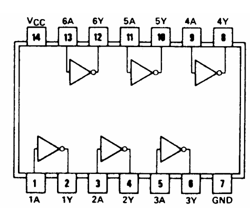
04B



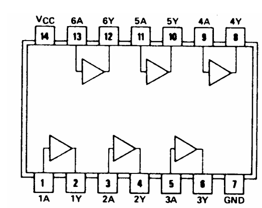
05A



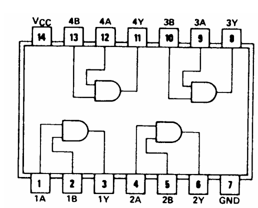
05B



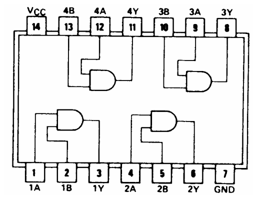
06



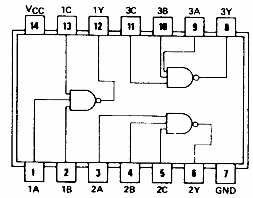
07



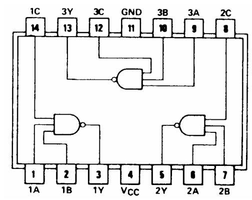
08



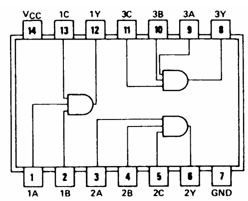
09



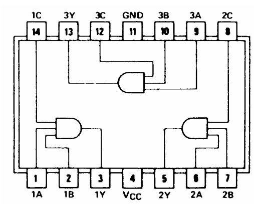
10A



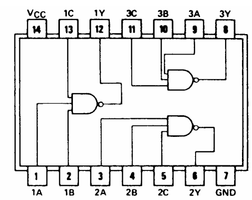
10B



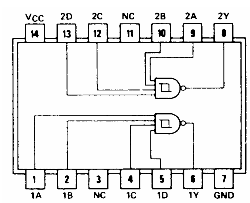
11A



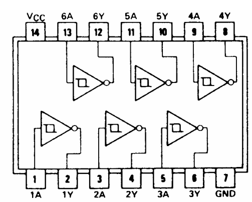
11B



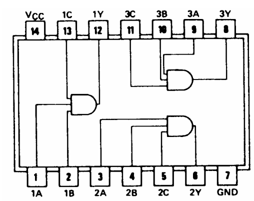
12



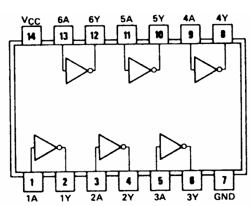
13



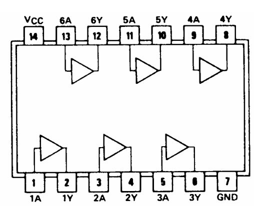
14



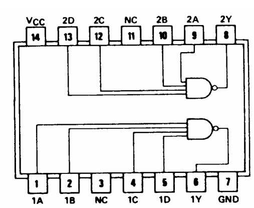
15



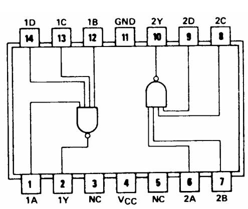
16



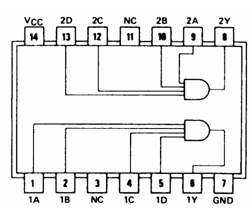
17



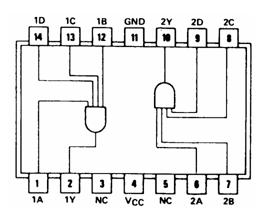
20A



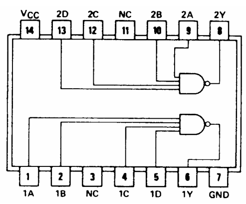
20B



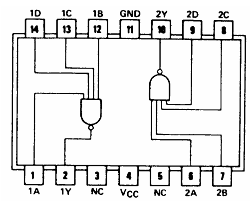
21A



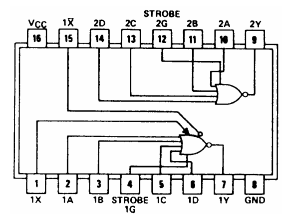
21B



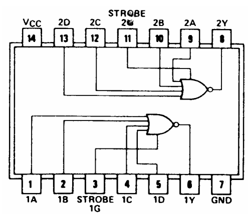
22A



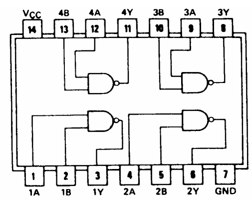
22B



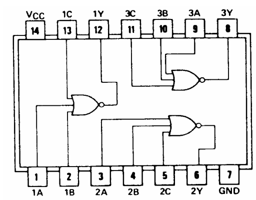
23



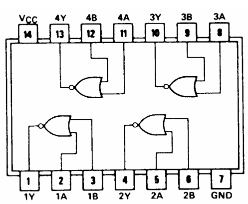
25



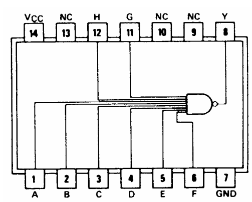
26



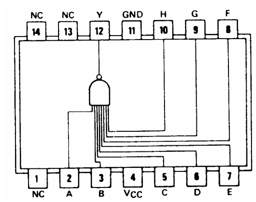
27



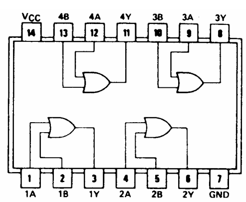
28



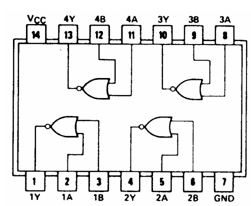
30A



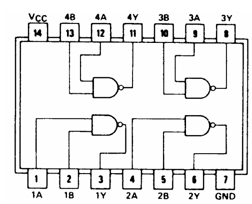
30B



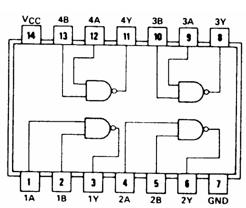
32



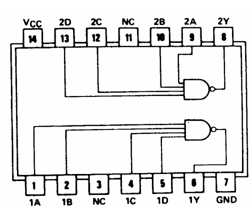
33



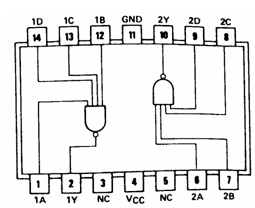
37



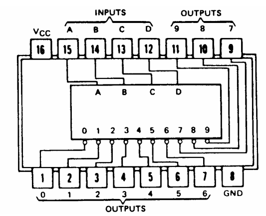
38



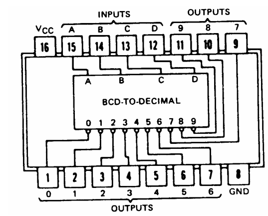
40A



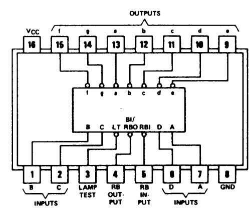
40B



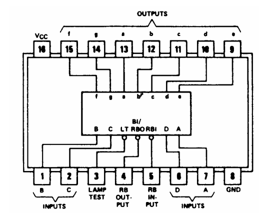
42,43,44



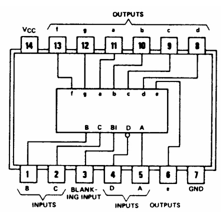
45



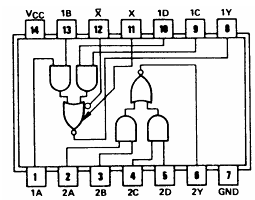
46,47



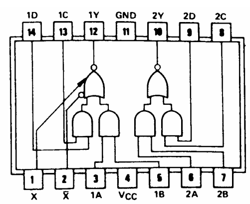
48



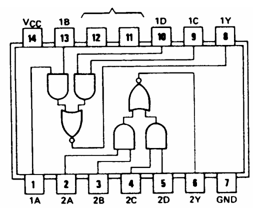
49



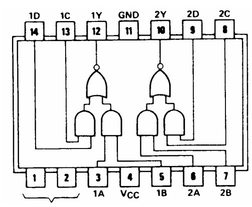
50A



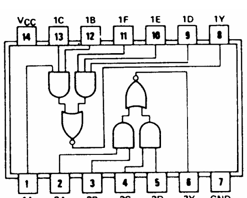
50B



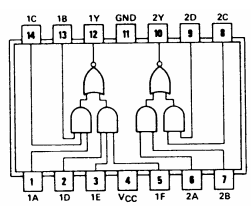
51A



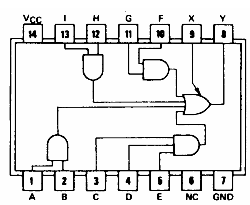
51B



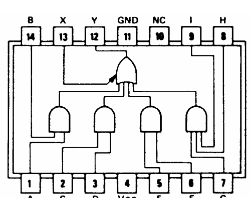
51C



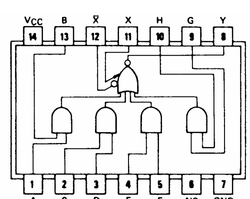
51D



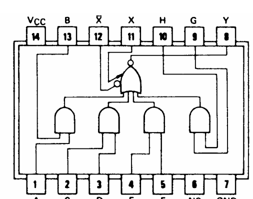
52A



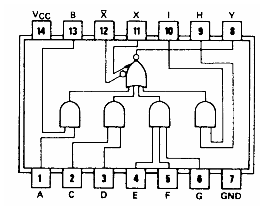
52B



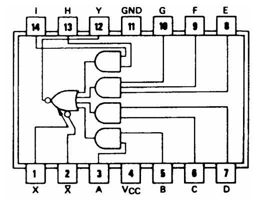
53A



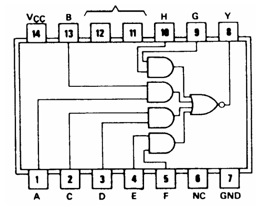
53B



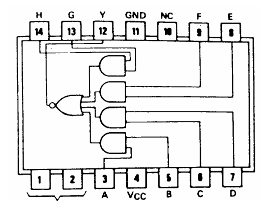
53C



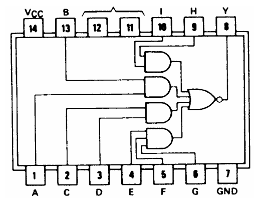
53D



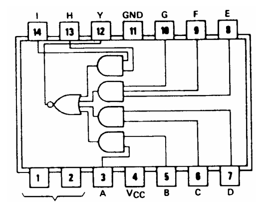
54A



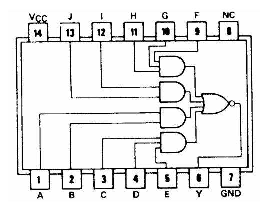
54B



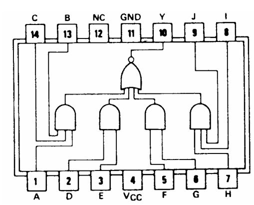
54C



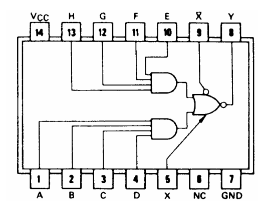
54D



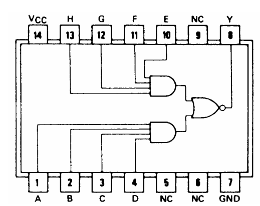
54E



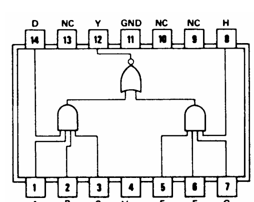
54F



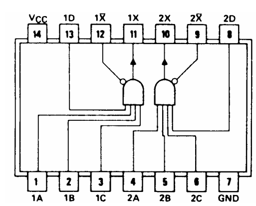
55A



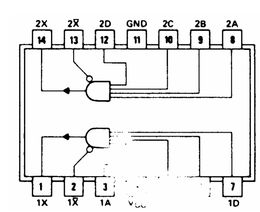
55C



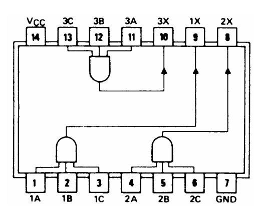
55D



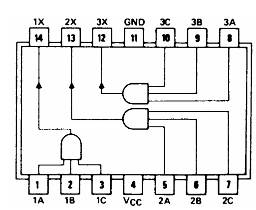
60A



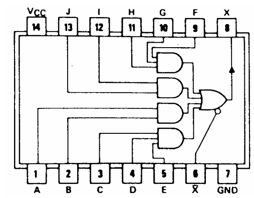
60B



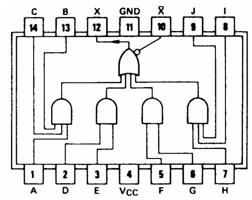
61A



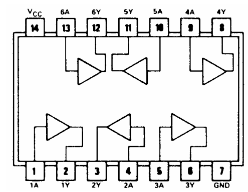
61B



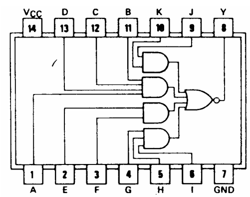
62A



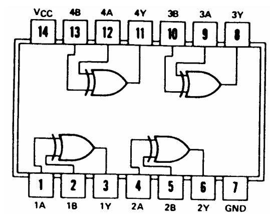
62B



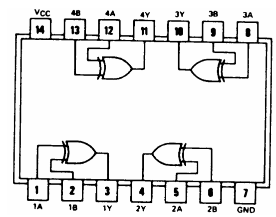
63



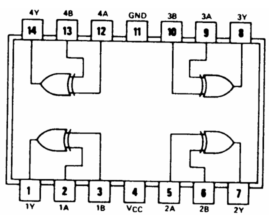
64,65



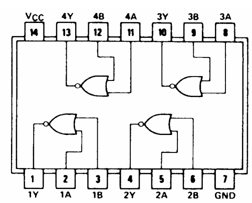
86A



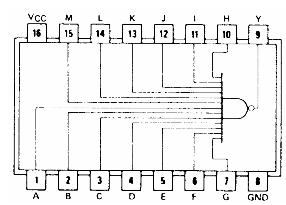
86B



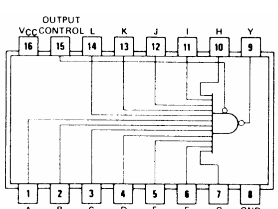
86C



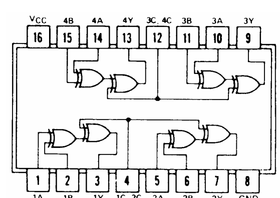
128



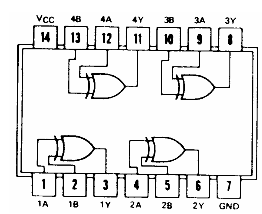
133



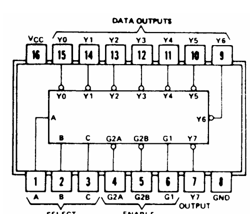
134



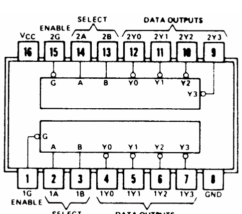
135



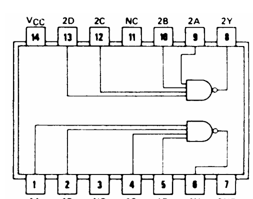
136



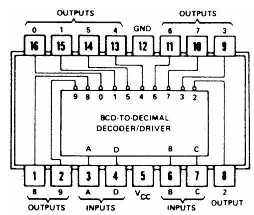
138



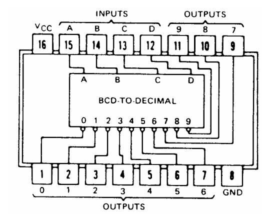
139



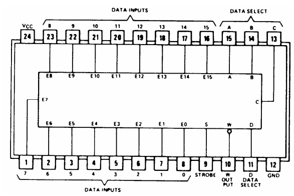
140



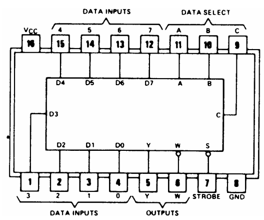
141



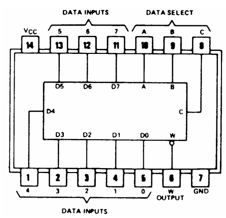
145



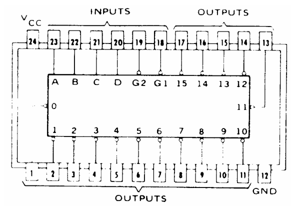
150



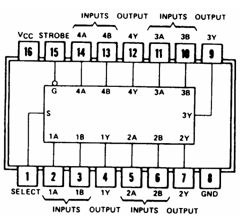
151



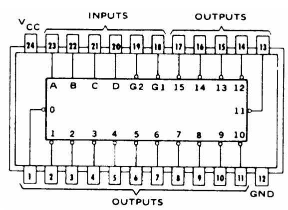
152



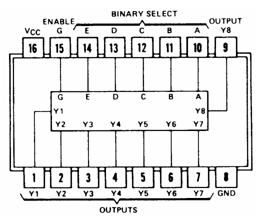
154



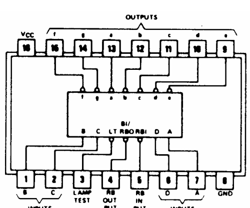
157,158



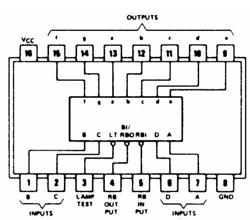
159



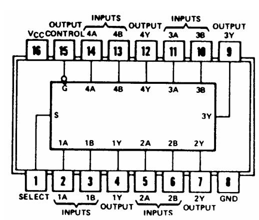
184,185



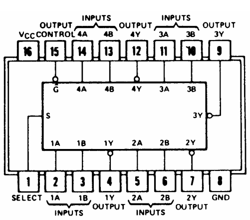
246,247



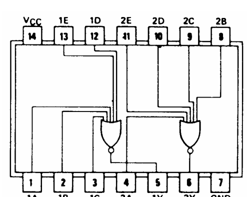
248,249



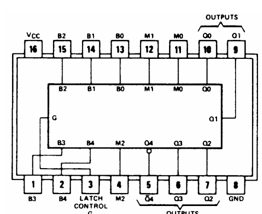
257



258



260



261

Literatura

1. **Ronald J. Tocci**
Digital Systems, Principles & Applications
Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1995.

2. **Logic symbols and Diagrams**
The Institute of Electrical and Electronics Engineers, New York 1987.

3. **Charles H. Roth, Jr.**
Fundamentals of Logic Design, Thirt Editon
West Publishing Co., St. Paul, Minnesota, 1985.

4. **PAL Device Handbook and Data Book**
Monolithic Memories
Advenced Micro Devices Inc., Sunnyvale, California 1988.

5. **System Design Handbook**
Monolithic Memories
Advenced Micro Devices Inc., Sunnyvale, California 1985.

6. **Douglas V. Hall**
Microprocessors and Interfacing, Programming and Hardvare
McGraw-Hill, New York 1986.

7. **Microprocessors and Peripherals Handbook**
Intel Corporation, Santa Clara, California, 1988.

8. **Microprocessors**
Intel, 1990.

9. **Abd-elfattah M. Abd-alla, Arnold C. Meltzer**
Principles of Digital Computer Design
Prentice Hall, Inc., Englewood Cliffs, New Jersey 1976.

10. **M. Morris Mano**
Digital Logic and Computer Design
Prentice Hall, Inc., Englewood Cliffs, New Jersey 1979.

11. Taylor L. Booth
Digital Networks and Computer Systems
John Wiley and Sons, Inc., New York, 1971.

12. Thomas C. Bartee
Digital Computer Fundamentals
McGraw-Hill, New York 1977.

13. David A. Hodges, H. G. Jackson
Analysis and Design of Digital Integrated Circuits
McGraw-Hill, New York 1988.

14. J. E. Oleksy, G. B. Rutkowski
Microprocessor and Digital Computer Technology
Prentice Hall, Inc., Englewood Cliffs, New Jersey 1989.

15. H. Troy Nagle, Jr., B. D. Carrol, J. D. Irwin
An Introduction to Computer Logic
Prentice Hall, Inc., Englewood Cliffs, New Jersey 1981.

16. J. F. Wakerly
Digital Design: Principles and Practices
Prentice Hall, Inc., Englewood Cliffs, New Jersey 1994.

17. P. K. Chan, S. Mourad
Digital Design Using Field Programmable Gate Arrays
Prentice Hall, Inc., Englewood Cliffs, New Jersey 1994.

18. J. M. Rabaey
Digital Integrated Circuits: A Design Perspective
Prentice Hall, Inc., Englewood Cliffs, New Jersey 1996.

19. C. H. Chen (Editor)
Computer Engineering Handbook
McGraw-Hill, New York 1992.

20. M. Morris Mano
Digital Design
Prentice Hall, Inc., Englewood Cliffs, New Jersey 1991.

21. Zvi Kohavi

Switching and Finite Automata Theory
McGraw-Hill, New York 1986.

22. M. Morris Mano
Computer System Architecture
Prentice Hall, Englewood Cliffs, New Jersey 1993.

23. John Y. Hsu
Computer Logic, Design Principles and Applications
Springer, New York 2002.

24. Charles H. Roth, Jr.
Fundamentals of Logic Design, 5th Editon
Brook/Cole - Thomson Learning
Belmont, CA USA 2004.

25. William Kleitz
Digital Electronics, a Practical Approach
Pearson Prentice Hall, New Jersey 2005.

26. Thomas L. Floyd
Digital Fundamentals
Pearson Prentice Hall, New Jersey 2005.

Dr. Agni H. Dika

Qarqet Kompjuterike
Kombinuese
1

Lektor
Dr. Ilaz Metaj

Kopertina
AfiDesign

Shtypi
Adea
Prishtinë