

DIGITÁLIS TECHNIKA

Dr. Oniga István

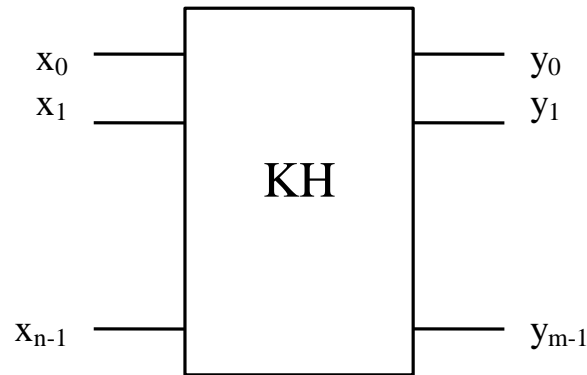
A tananyag elkészítését az EFOP-3.4.3-16-2016-00021 számú projekt támogatta.
A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósult meg.

Kombinációs logikai hálózatok

- Logikai hálózat = olyan hálózat, melynek bemenetei és kimenetei logikai állapotokkal jellemezhetők
- **Kombinációs logikai hálózat:** olyan logikai hálózat, mely kimenetei csak a bemenetek állapotaitól, kombinációjától függenek, semmi mástól.
- **Szekvenciális logikai hálózat** (sorrendi): kimenetei nem csak a bemenetek kombinációjától, hanem az előzményektől, a különböző kombinációk sorrendjétől is függenek.

Kombinációs logikai hálózatok

- A kimeneti jelek értékei csak a bemeneti jelek pillanatnyi értékétől függenek.

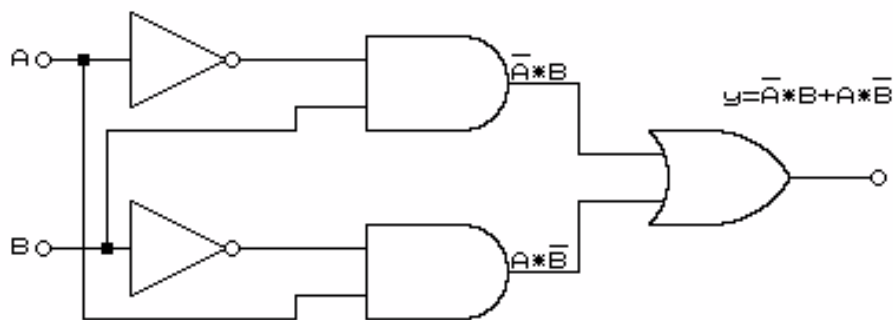


A kimenetek egy-egy függvénykapcsolattal írhatók le.

$$y_k = f_k(x_0, x_1, \dots, x_{n-1})$$

Kombinációs hálózatok vizsgálata

- Az áramkör kapcsolási rajzából kiindulva a működés leírását (igazságtáblázatot vagy a logikai függvényt) eredményezi.



A	B	$\bar{A} \cdot B$	$A \cdot \bar{B}$	Y
L	L	L	L	L
L	H	H	L	H
H	L	L	H	H
H	H	L	L	L

Kombinációs hálózatok tervezése

- Igazságtábla felállítása (n jel esetén 2^n sorral rendelkező táblázat)
- Logikai függvény felírása
- (Logikai függvény minimalizálása)
- (Hazárdmentesítés)
- Megvalósítás logikai kapukkal

Kombinációs hálózatok tervezése I

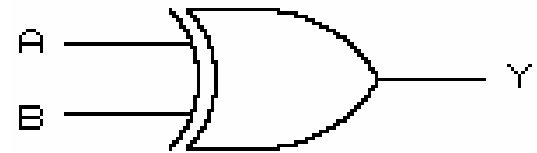
- Igazságtábla felállítása (n jel esetén 2^n sorral rendelkező táblázat)
- Logikai függvény felírása
- (Logikai függvény minimalizálása)
- (Hazárdmentesítés)
- Megvalósítás logikai kapukkal
 - NOT, AND, OR kapuk használatával
 - NOT, OR, AND kapuk használatával
 - NAND kapuk használatával
 - NOR kapuk használatával

Kombinációs hálózatok tervezése II

PI.

- *A kizáró-vagy (XOR) függvény megvalósítása*

A	B	$Y=A \oplus B$
L	L	L
L	H	H
H	L	H
H	H	L



B \ A	0	1
0	0	1
1	1	0

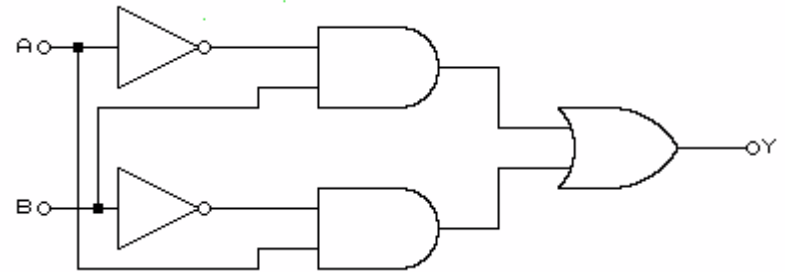
NOT, AND, OR kapuk alkalmazása

1. Szorzatok összege (NOT, AND, OR):

- Karnaugh diagram

B \ A	0	1
0	0	1
1	1	0

$$Y = (\bar{A} \bullet B) + (A \bullet \bar{B})$$

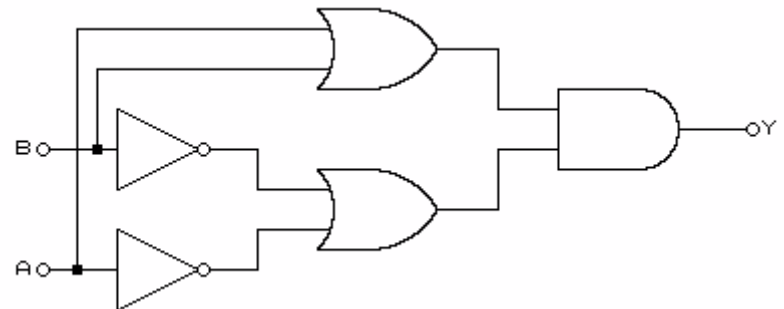


2. Összegek szorzata (NOT, OR, AND)

- Karnaugh diagram

B \ A	0	1
0	0	1
1	1	0

$$Y = (A + B) \bullet (\bar{A} + \bar{B})$$



XOR megvalósítása NAND vagy NOR kapukkal

• De Morgan tételeit alkalmazva az előző két kifejezésen:

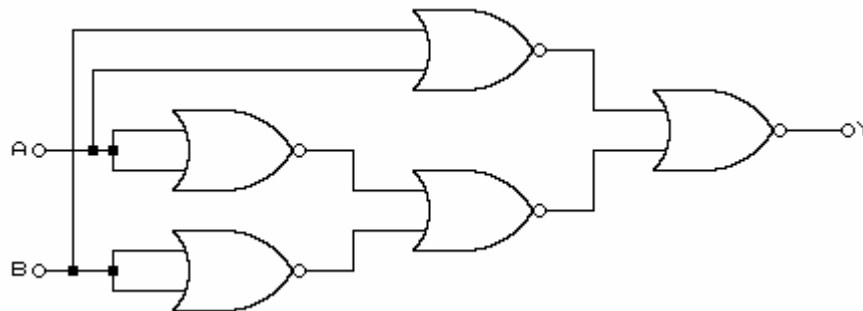
3. NAND kapuk alkalmazása:

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



4. NOR kapuk alkalmazása

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



Funkcionális kombinációs egységek

- A következő funkcionális egységek logikai felépítésével, és működésével foglalkozunk:
 - *kódoló*k,
 - *dekódoló*k,
 - *multiplex*erek,
 - *demultiplex*erek,
 - *komparátor*ok,
 - *összeadó*k,
 - *paritásvizsgáló áramkör*ök.

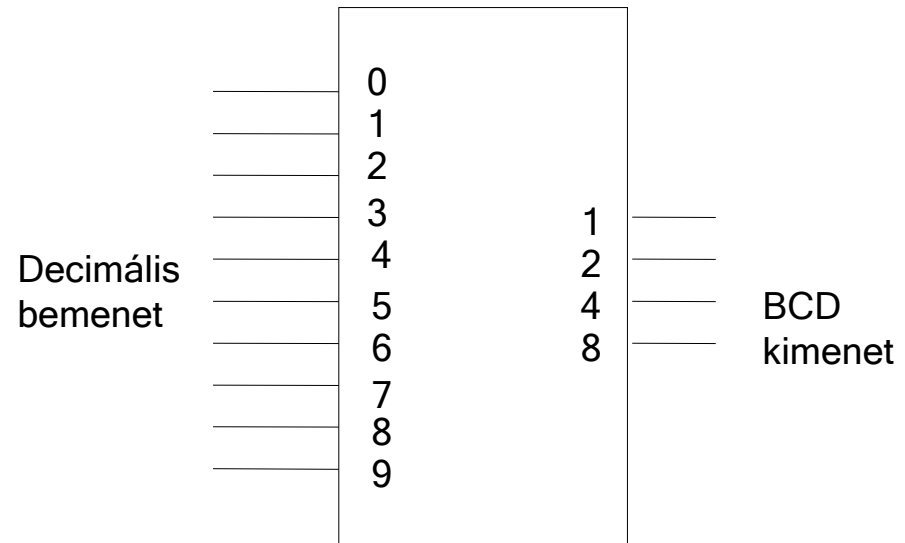
Kódolók

- A kódoló egy olyan áramkör amelynek bármelyik 1 az m -ből bemenetének az aktiválása esetén egy k bites kódot szolgál

$$Y_r = \sum_0^{m-1} a_n I_n \quad r = 0, 1, \dots, (k-1),$$

- **Decimális – BCD átalakító**

I	Y ₃	Y ₂	Y ₁	Y ₀
I ₀	0	0	0	0
I ₁	0	0	0	1
I ₂	0	0	1	0
I ₃	0	0	1	1
I ₄	0	1	0	0
I ₅	0	1	0	1
I ₆	0	1	1	0
I ₇	0	1	1	1
I ₈	1	0	0	0
I ₉	1	0	0	1

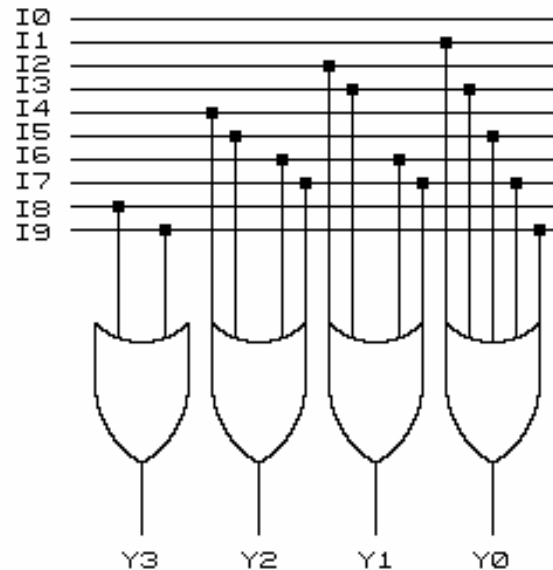


Decimális – BCD átalakító

- A táblázat alapján felírhatók az egyes kimeneteket megvalósító logikai függvények

- $Y_0 = I_1 + I_3 + I_5 + I_7 + I_9$
- $Y_1 = I_2 + I_3 + I_6 + I_7$
- $Y_2 = I_4 + I_5 + I_6 + I_7$
- $Y_3 = I_8 + I_9$

I	Y ₃	Y ₂	Y ₁	Y ₀
I ₀	0	0	0	0
I ₁	0	0	0	1
I ₂	0	0	1	0
I ₃	0	0	1	1
I ₄	0	1	0	0
I ₅	0	1	0	1
I ₆	0	1	1	0
I ₇	0	1	1	1
I ₈	1	0	0	0
I ₉	1	0	0	1



Prioritásos kódoló I

- Ha egyidejűleg egynél több bemenet aktív, az eredmény nem meghatározható.
- A megoldás: a bemenetekhez prioritást rendelünk
- Ha egy vagy több bemenet aktív, akkor a legmagasabb prioritásúnak megfelelő eredményt kapjuk

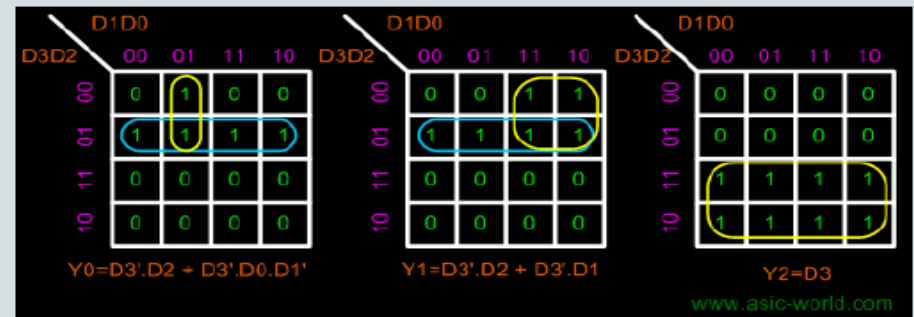
4-ből 3-ba prioritásos logika igazságtáblája

D3	D2	D1	D0	Y2	Y1	Y0
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	x	0	1	0
0	1	x	x	0	1	1
1	x	x	x	1	0	0

Prioritásos kódoló II

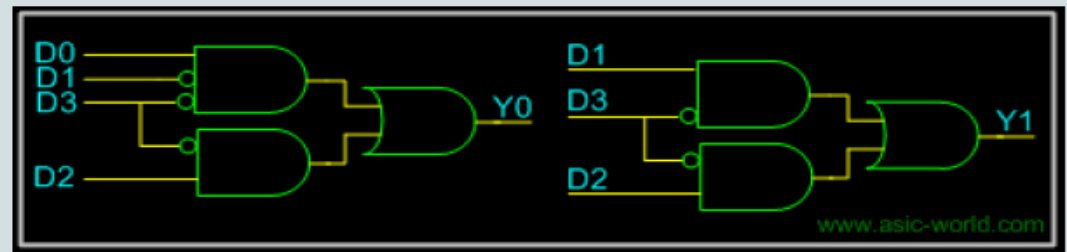
Az igazságtáblázat alapján megrajzolhatjuk a Karnaugh-diagramot és a kapcsolási rajzot.

A 4-ből 3-ba prioritásos logika Karnaugh-diagramja



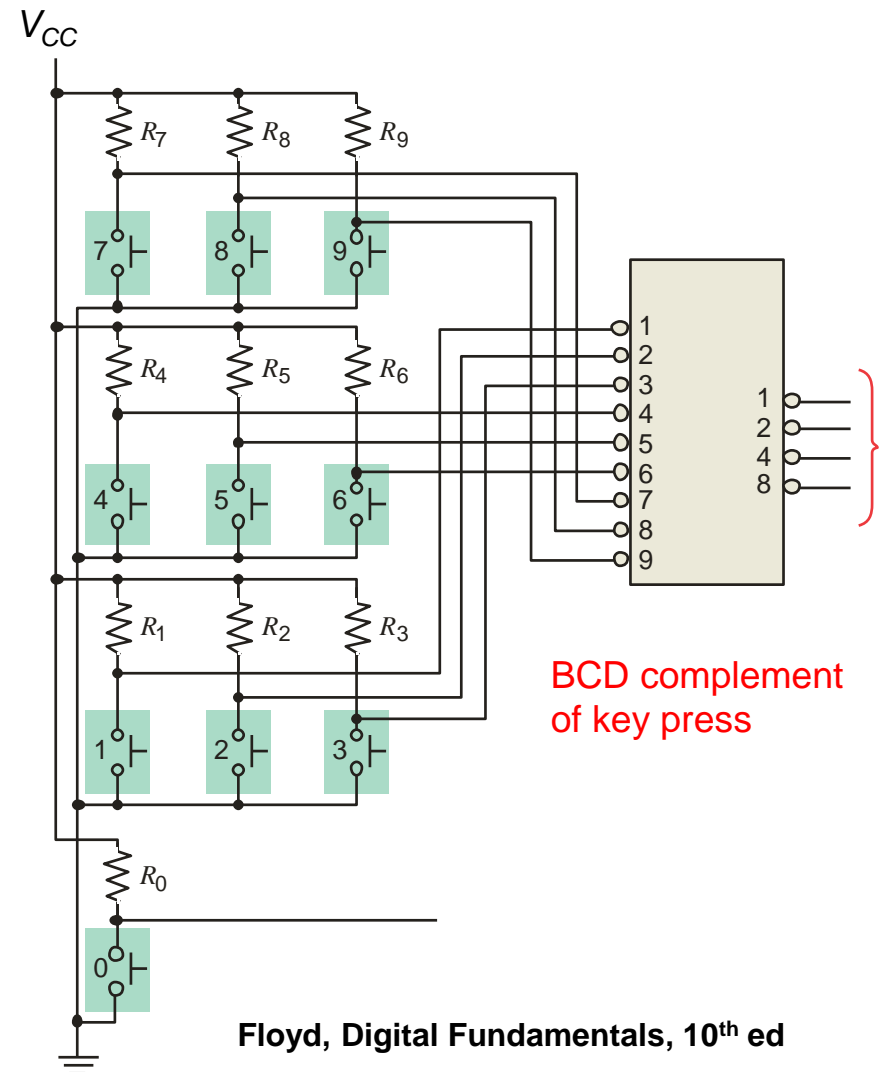
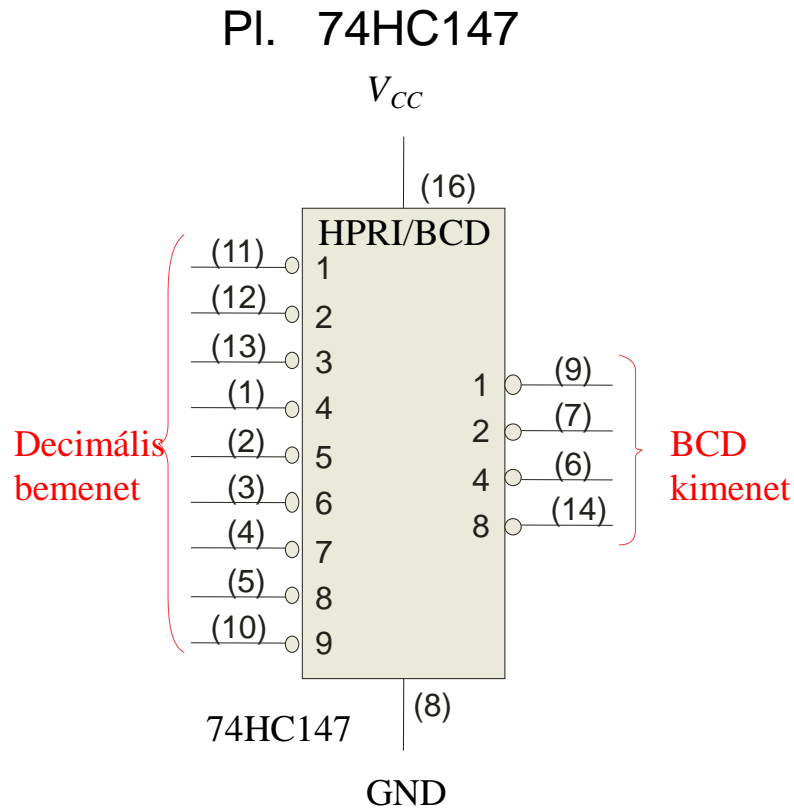
©1998-2008, Deepak Kumar Tala, <http://asic-world.com>

Az $Y2$ kimenetet közvetlenül $D3$ -ra kötjük.



Prioritásos kódoló III

Alkalmazási példa: billentyűzet kódoló



Dekódoló

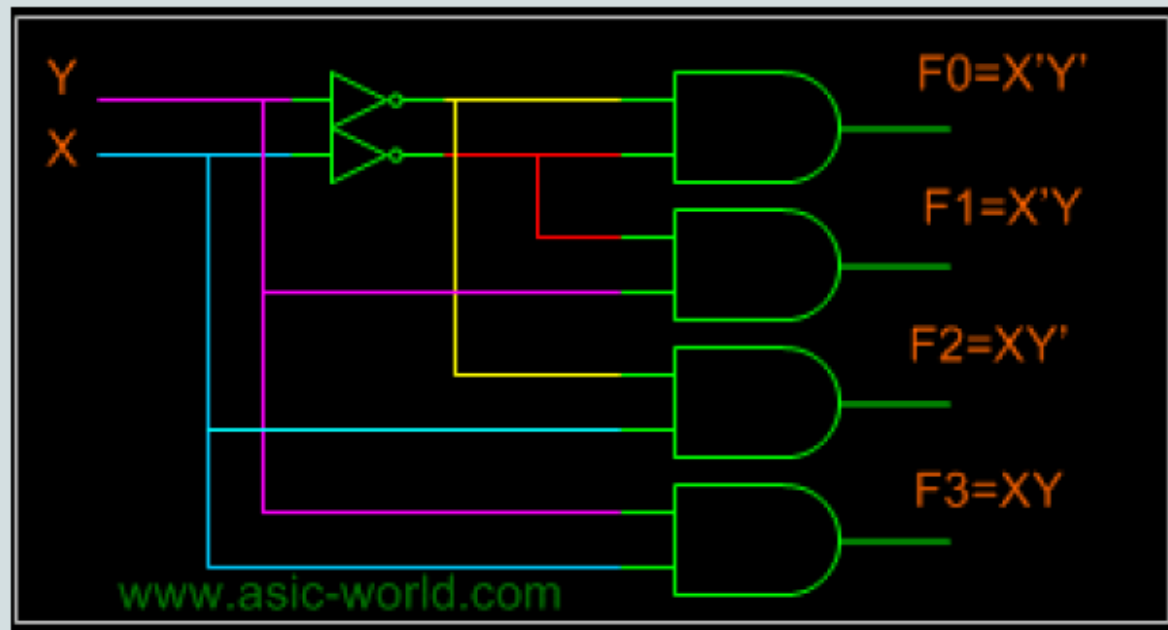
- A dekódoló (dekóder) egy olyan áramkör amely a bemenetére adott kód alapján **egyetlen kimenetet tesz aktívvá**
- Egy bináris dekódolónak n bemenete és 2^n kimenete van (ezek közül egyidejűleg csak **egy lehet aktív logikai értékű**).

Bináris $n - 2^n$ dekóder



©1998-2008, Deepak Kumar Tala, <http://asic-world.com>

Bináris dekódoló 2-ről 4-re

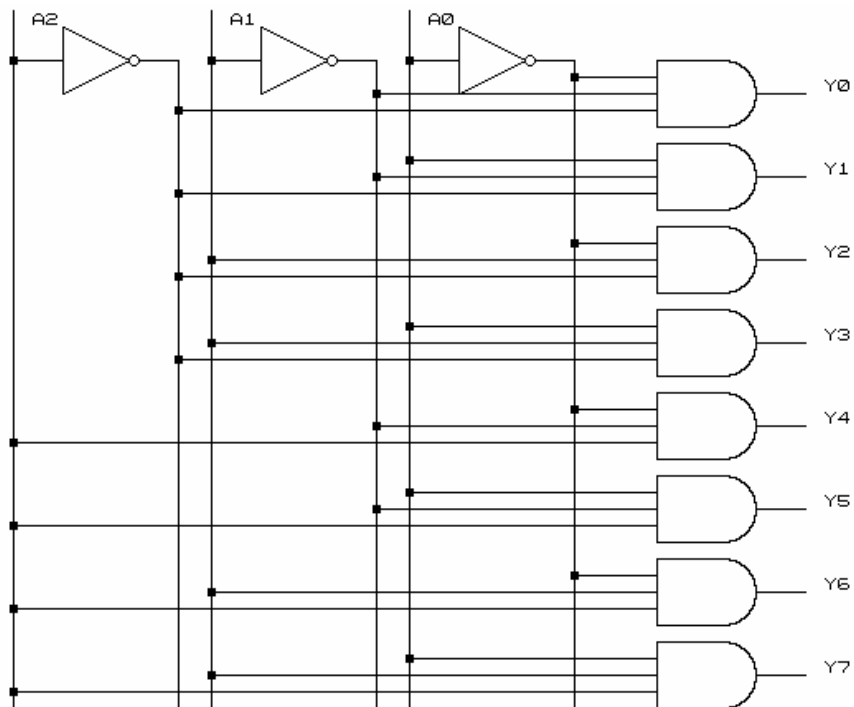
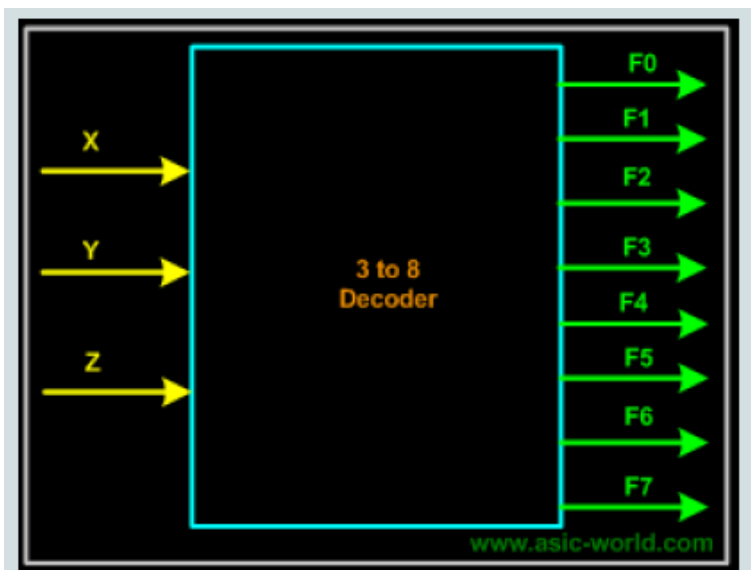


©1998-2008, Deepak Kumar Tala, <http://asic-world.com>

Bináris dekóder 3-ról 8-ra

A dekóder elvi rajza

A_2	A_1	A_0	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_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

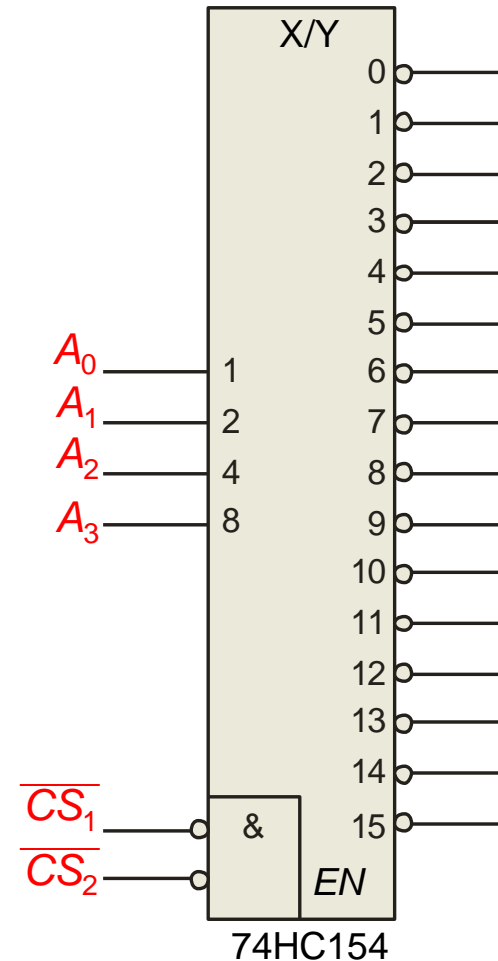


©1998-2008, Deepak Kumar Tala,
<http://asic-world.com>

4-ről 16-ra dekódoló

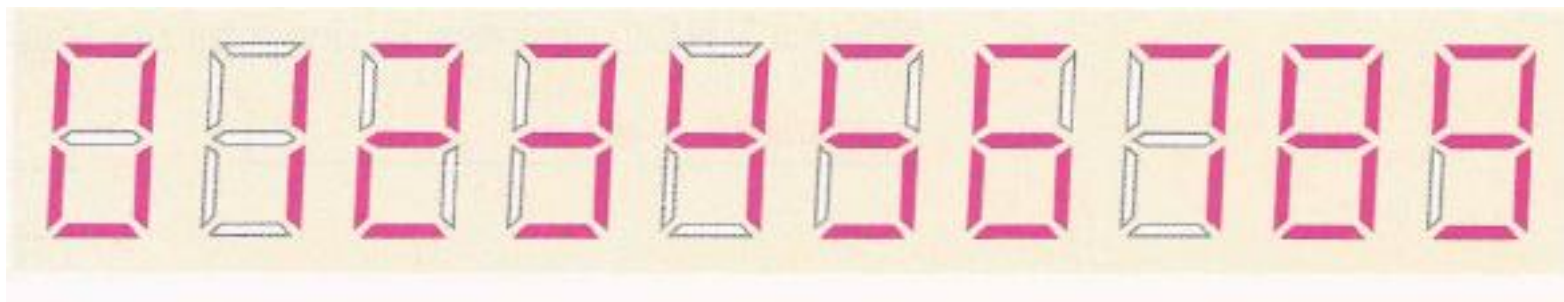
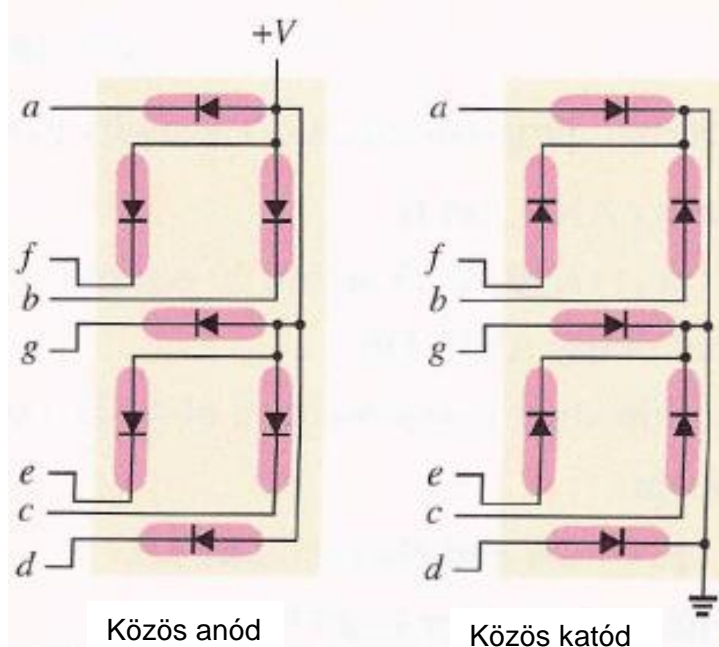
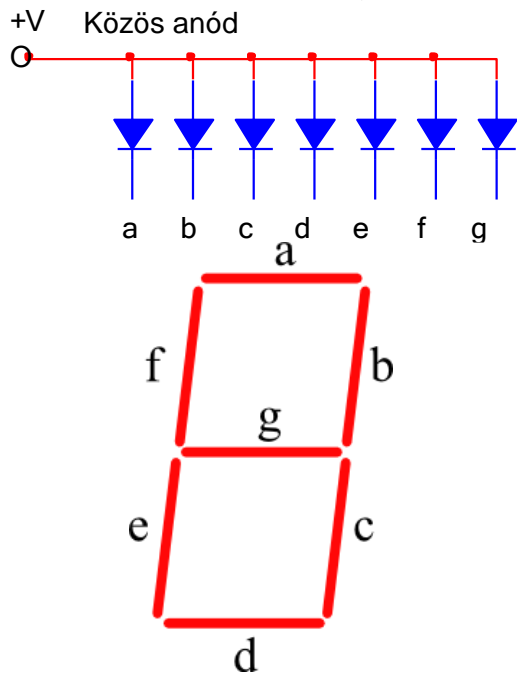
Az 74HC154 típusú dekóder:

- a két engedélyező CS bemenet logikai ÉS kapcsolatban van,
- a bemenet akkor igaz, ha mindkettő alacsony (ezért vannak negáltan jelölve)
- Ez az áramkör az adat szempontjából alacsony logikai szintű, azaz az L az aktív szint a H szint az nem aktív.



BCD – 7 szegmenses dekódoló

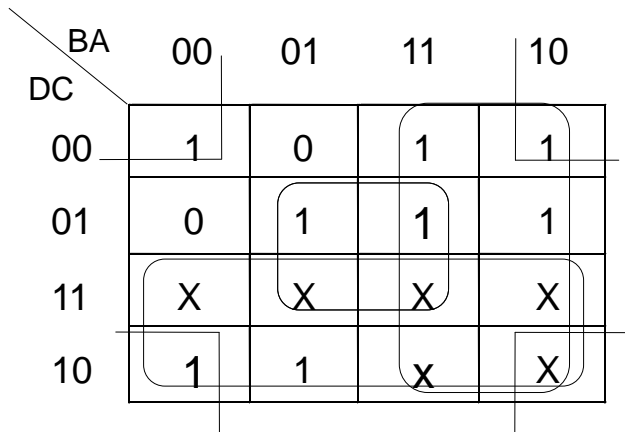
A dekóder, **BCD 8421** súlyozású kódból állítja elő a 7 szegmensű kijelző vezérlésére alkalmas jeleket az **a, b, c, d, e, f, g** jelű kimenetein.



BCD – 7 szegmenses dekódoló II

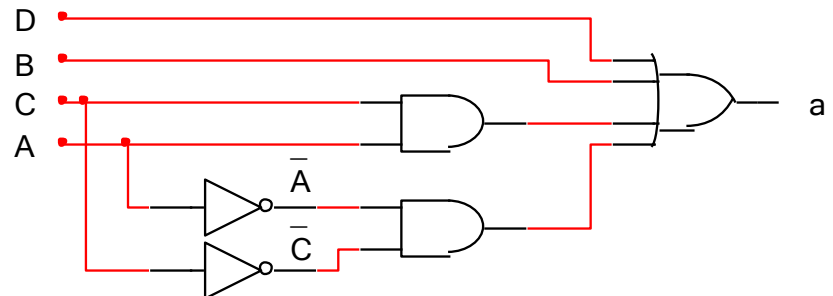
Az igazságtáblázatból felírhatjuk a következő logikai függvényt:

$$a = \overline{D}\overline{C}\overline{B}\overline{A} + \overline{D}\overline{C}B\overline{A} + \overline{D}C\overline{B}\overline{A} + \overline{D}CBA + \overline{D}C\overline{B}A + \overline{D}CBA + DC\overline{B}\overline{A} + DC\overline{B}A$$



$$a = D + B + CA + \overline{C}\overline{A}$$

Zecimal	D	C	B	A	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1
10	1	0	1	0	X	X	X	X	X	X	X
11	1	0	1	1	X	X	X	X	X	X	X
12	1	1	0	0	X	X	X	X	X	X	X
13	1	1	0	1	X	X	X	X	X	X	X
14	1	1	1	0	X	X	X	X	X	X	X
15	1	1	1	1	X	X	X	X	X	X	X

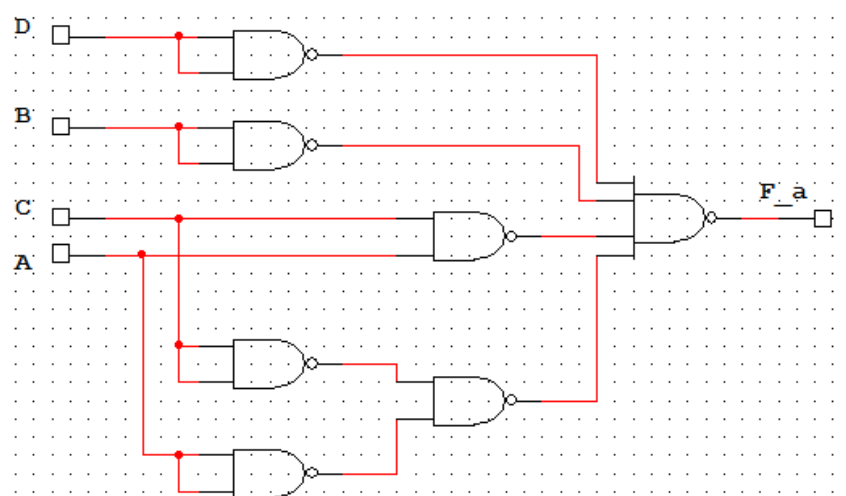
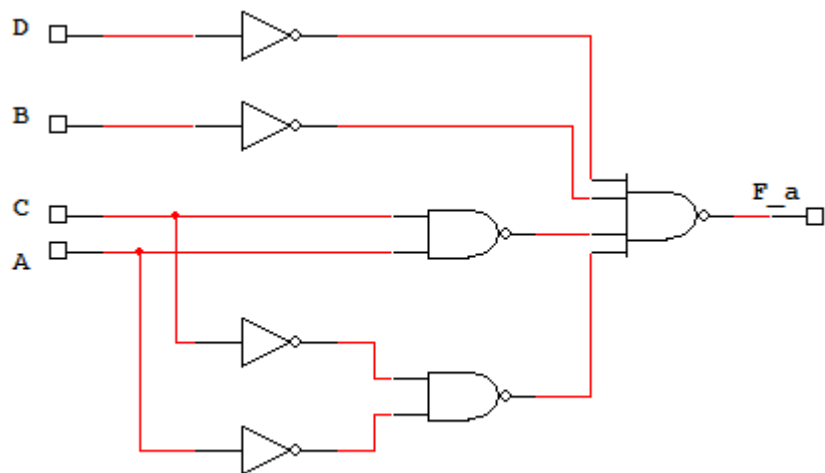


BCD – 7 szegmenses dekódoló III

A függvény megvalósítása NAND kapukkal

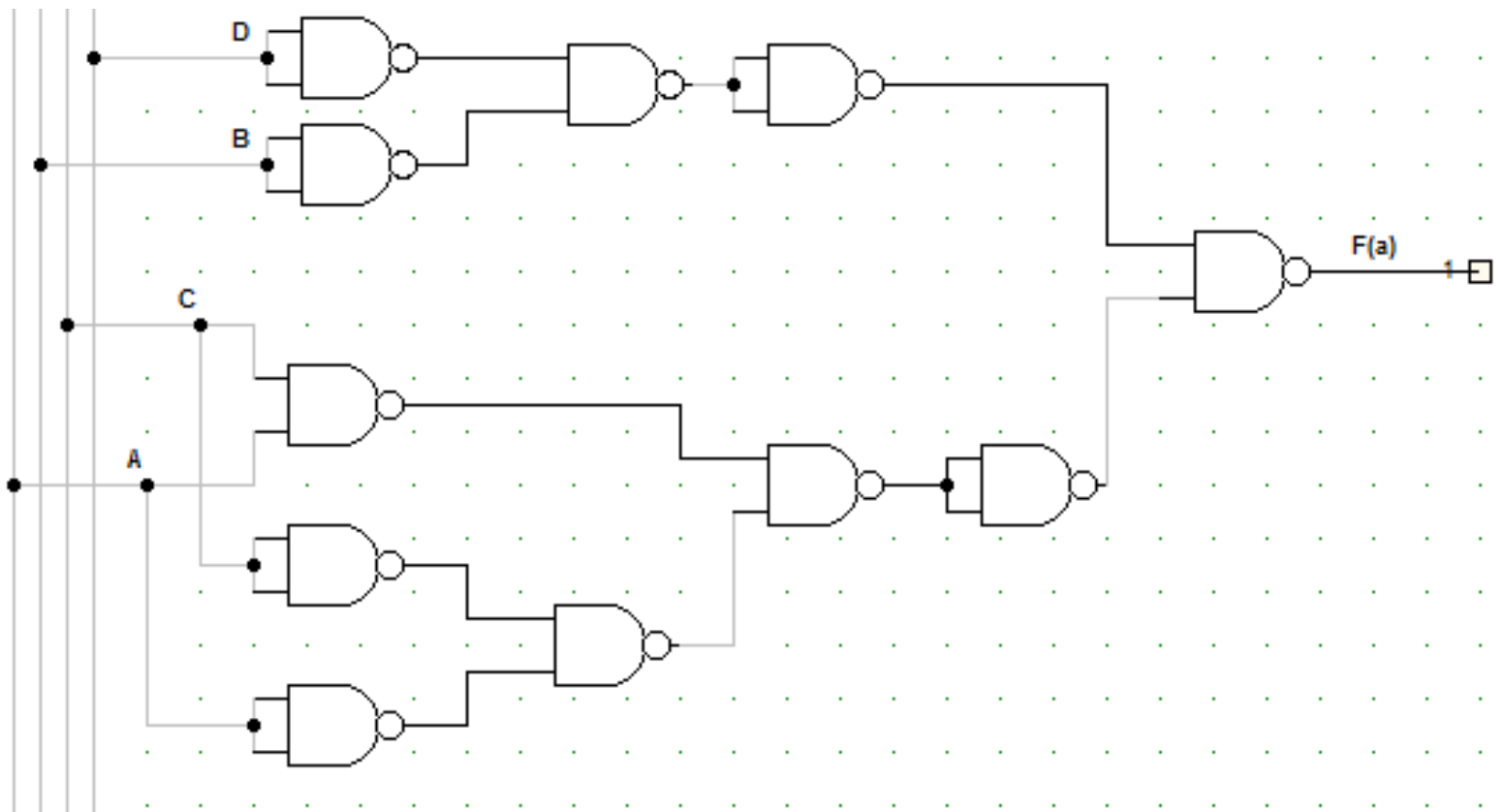
- Az előző függvény átalakítható De Morgan tételek használatával

$$a = D + B + CA + \overline{C}\overline{A} = \overline{\overline{D + B + CA + \overline{C}\overline{A}}} = \overline{\overline{D} \cdot \overline{B} \cdot \overline{C} \cdot \overline{A} \cdot \overline{C} \cdot \overline{A}}$$

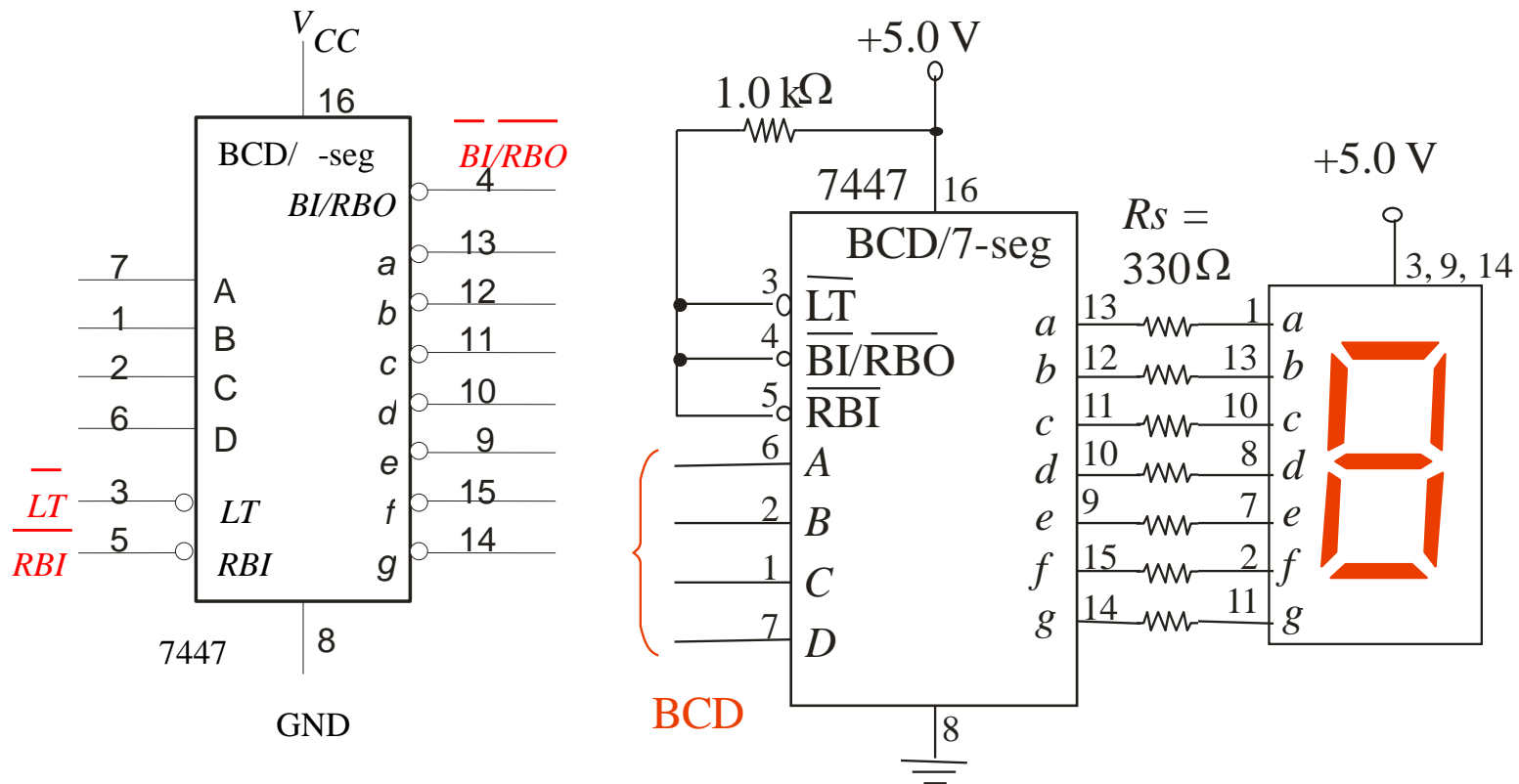


BCD – 7 szegmenses dekódoló IV

A függvény megvalósítása 2 bemenetű NAND kapukkal

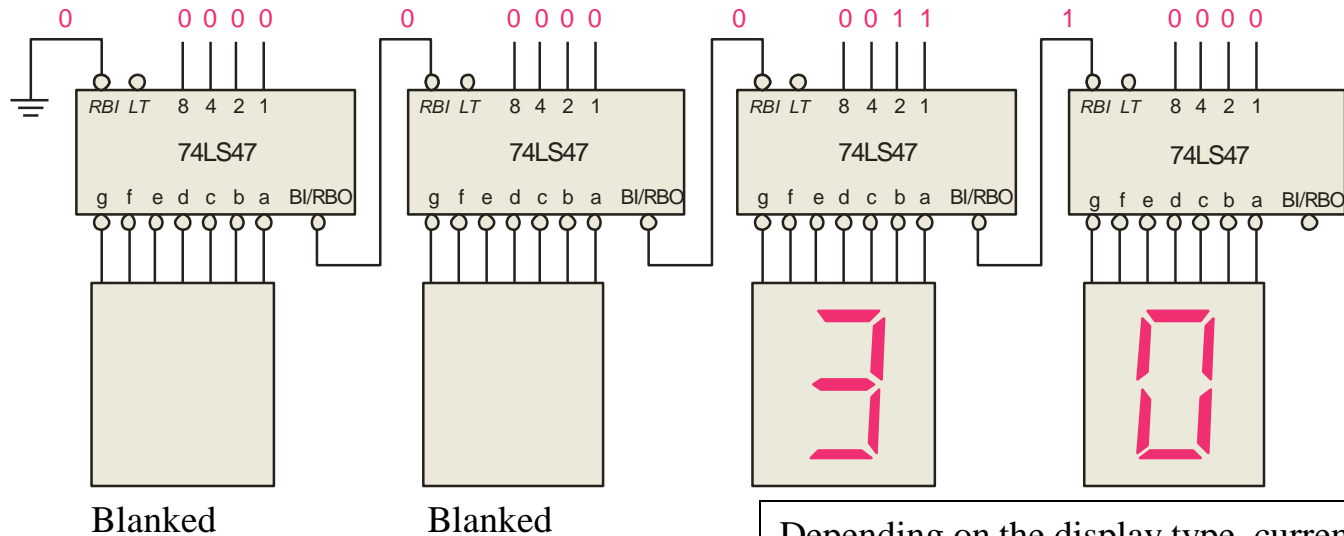


7447, BCD – 7 szegmenses dekódoló



BCD Decoder/Driver

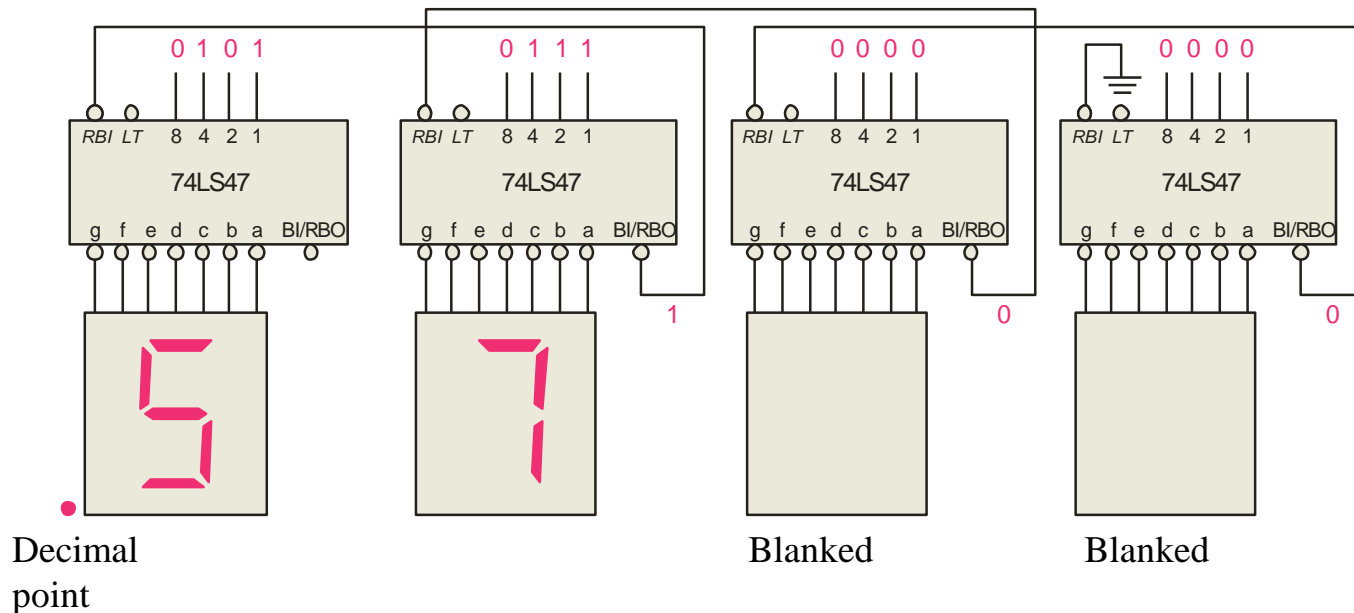
The 74LS47 features leading zero suppression, which blanks unnecessary leading zeros but keeps significant zeros as illustrated here. The $\overline{BI/RBO}$ output is connected to the RBI input of the next decoder.



Depending on the display type, current limiting resistors may be required.

BCD Decoder/Driver

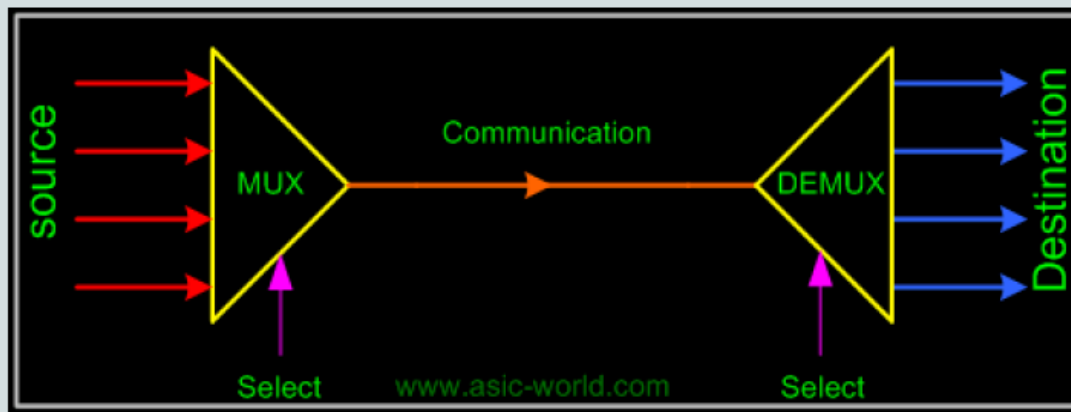
Trailing zero suppression blanks unnecessary trailing zeros to the right of the decimal point as illustrated here. The *RBI* input is connected to the *BI/RBO* output of the following decoder.



MUX-DEMUX

- Kevés számú adathordozó (vezeték, rádióhullám, stb.) igénybevételével - nagy számú jelek továbbítására alkalmas.

Mux-Demux alkalmazási példa

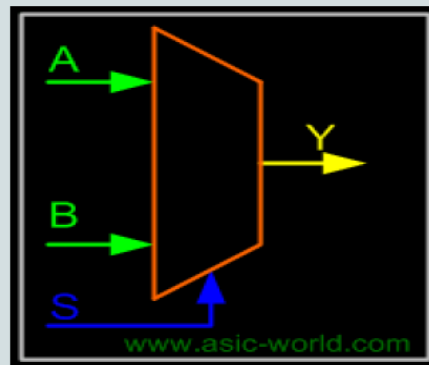


- Adatgyűjtés, adatelosztás
 - Analóg multiplexelés, **digitális multiplexelés**
 - Byte vagy **bitszervezésű multiplexelés**
 - Időmultiplexelés vagy **címszerinti multiplexelés**

MULTIPLEXEREK

- A multiplexer olyan digitális kapcsoló, amely m ($m=2^n$) különböző forrásból származó adatokat kapcsol egy kimenetre.
- A kiválasztó bemenetek határozzák meg, hogy melyik adatforrás jele kerül a kimenetre
- 2^n féle adathoz 2^n különböző cím kell
- 2^n különböző címet n bittel tudunk előállítani,

Kétvonalas
multiplexer
blokkdiagramja



©1998-2008, Deepak Kumar Tala,
<http://asic-world.com>

S	Y
0	A
1	B

Példa – MUX (1.)

. 2:1 multiplexer

```
module mux_21 (input in0, in1, sel, output r);  
  assign r = (sel==1'b1) ? in1 : in0;  
endmodule
```

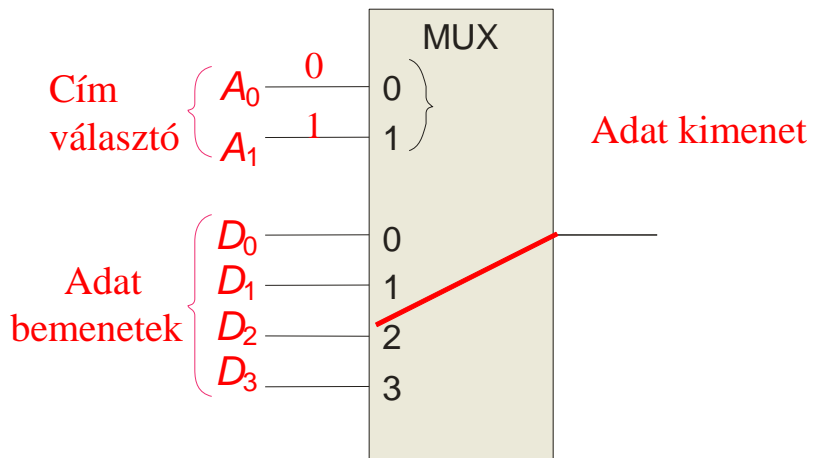
```
module mux_21 (input in0, in1, sel, output reg r);  
  always @ (*)  
  if (sel==1'b1) r <= in1;  
  else          r <= in0;  
endmodule
```

```
module mux_21 (input in0, in1, sel, output reg r);  
  always @ (*)  
  case(sel)  
    1'b0: r <= in0;  
    1'b1: r <= in1;  
  endmodule
```

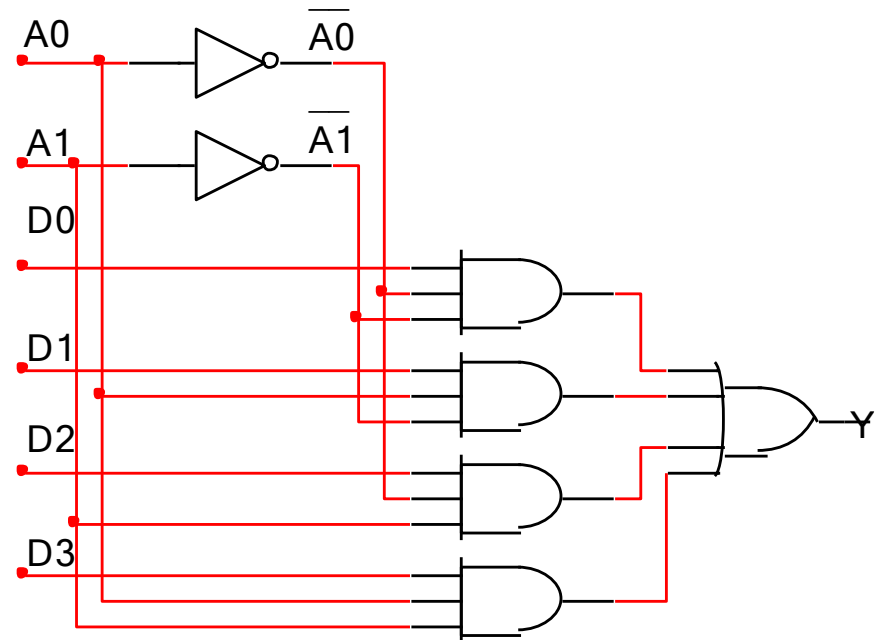
4 x1 MULTIPLEXER

- 4 féle adathoz két különböző címbit kell

4 vonalas multiplexer blokkdiagramja



4 vonalas multiplexer áramköri megvalósítása



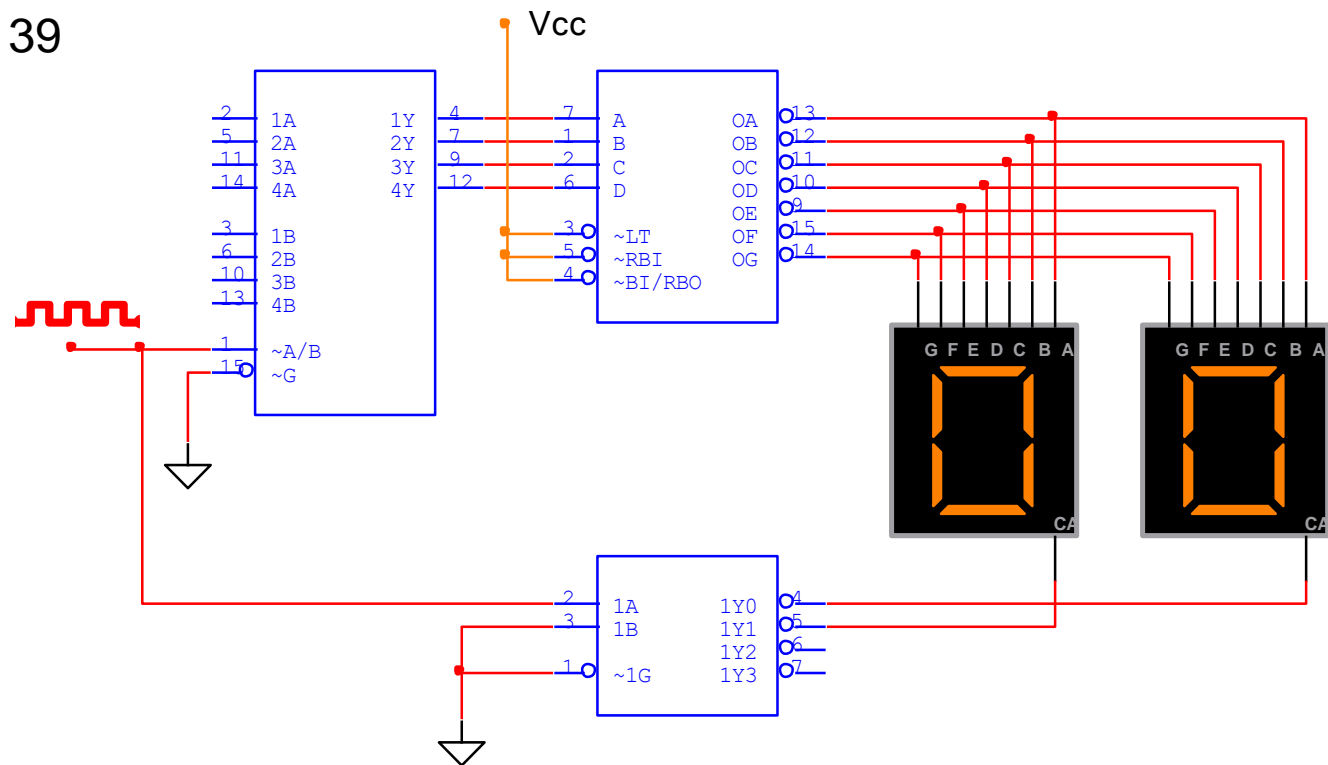
4 x1 MULTIPLEXER

```
module mux_41 (input in0, in1, in2, in3, input [1:0] sel, output reg r);  
always @ (*)  
case(sel)  
    2'b00: r <= in0;  
    2'b01: r <= in1;  
    2'b10: r <= in2;  
    2'b11: r <= in3;  
endcase  
endmodule
```

Multiplexer alkalmazási példák I

Több digités 7 szegmenses kijelzők meghajtása egy BCD/7 szegmenses dekóderrel

- 4 két bemenetű (A és B) multiplexer - 74LS157
- BCD/7 szegmenses dekóder - 74LS47
- 2:4 dekóder 74LS139



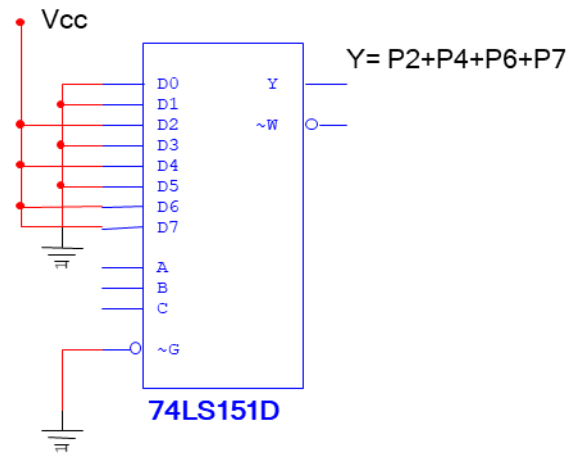
Multiplexer alkalmazási példák II

Logikai függvény megvalósítása multiplexerrel

- A multiplexer kétszintű ÉS-VAGY felépítésű hálózat tehát multiplexer is alkalmazható a ÉS-VAGY függvények megvalósításához.
- A multiplexer kiválasztó bemeneteire a függvény változóit kötjük
- Az adatbemenetekre az igazságtáblázatban megadót értékeket kötjük

Bemenetek			
C	B	A	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

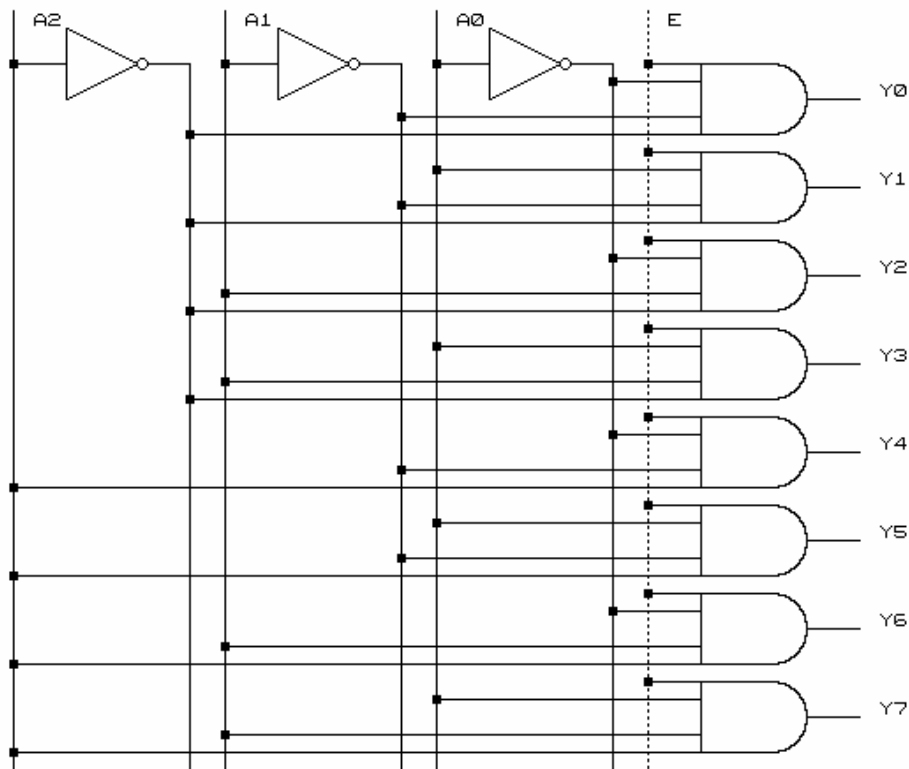
a.



b.

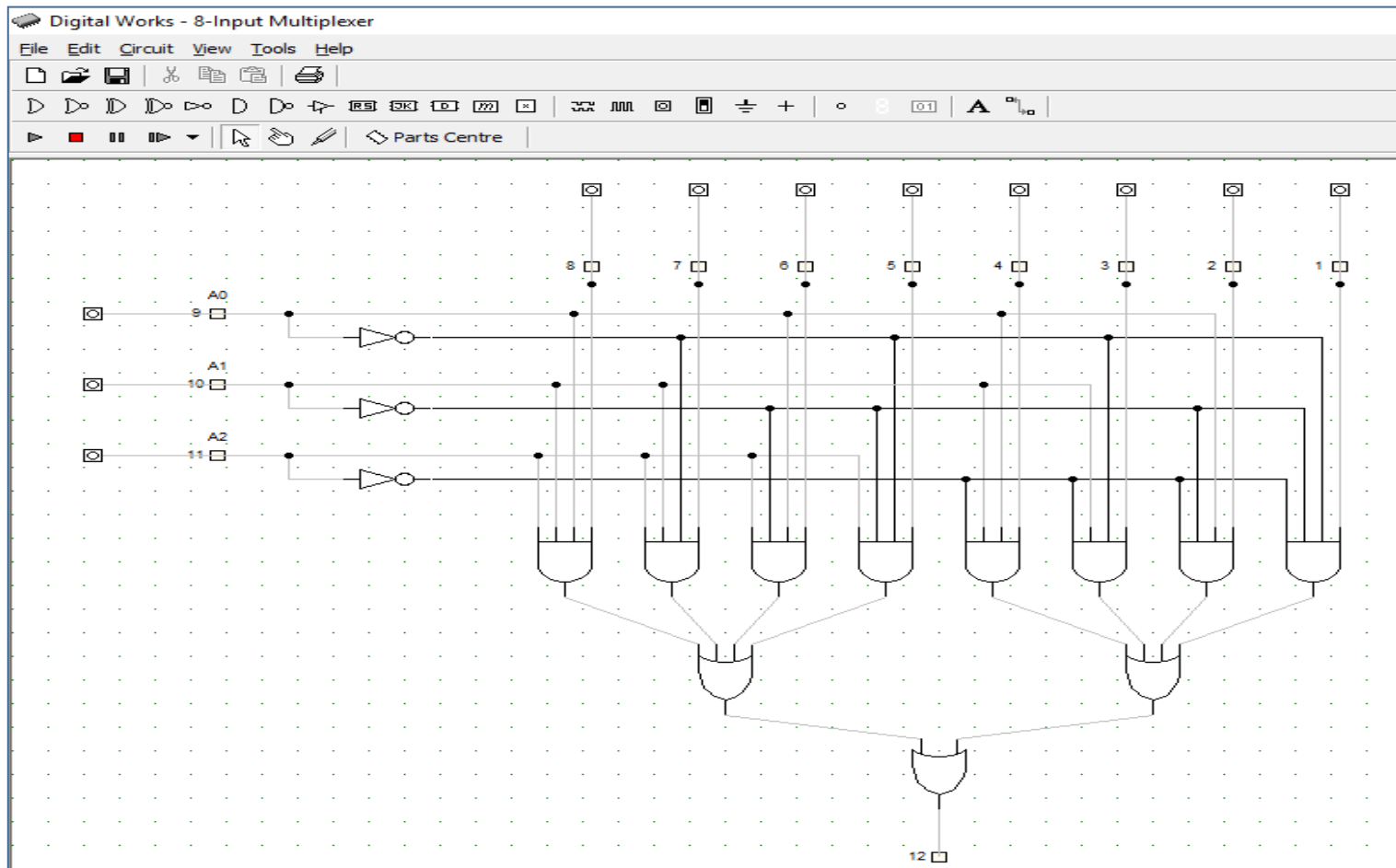
Demultiplexerek

- A demultiplexer egy olyan digitális kapcsoló, amely egy bemenő forrást n kimenet valamelyikére kapcsolja.
- Általában n -ből 2^n -re dekódolót használunk.
- A dekódoló engedélyező jele a demultiplexer adatbemenete.



Gyakorlat

- http://www.play-hokey.com/digital/combinational/multiplexer_four_input.html
- Digital Works szimulációs program



Gyakorlat

http://www.play-hookey.com/digital/combinational/decoder_demux_four.html

