

Digitális Technika

Dr. Oniga István
Debreceni Egyetem, Informatikai Kar

4. Laboratóriumi gyakorlat

A gyakorlat célja:

- Kétszintű hálózatok
- SOP realizáció
- Logikai függvények egyszerűsítése

Lab4_1 feladat:

Kétszintű hálózatok

- XOR függvény implementálása 5 féle képen -

- Hozunk létre egy új projektet
- Adjunk hozzá egy új "schematic" forrásfájlt
- Rajzoljuk le a következő dián látható áramköröket.
- A NexysX.UCF fájl hozzáadása és adaptálása
 - Bemenetek: **sw[1:0]** (A és B az ábrán)
 - Kimenetek: **led[4:0]**

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

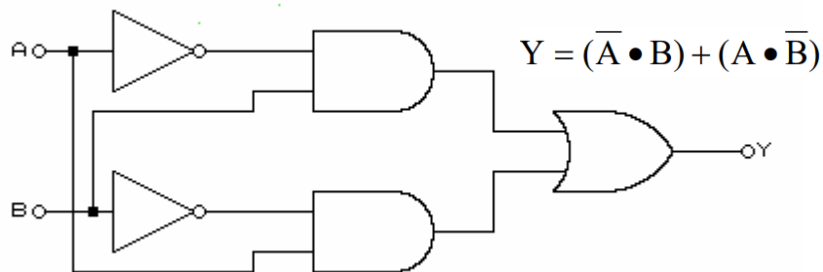


XOR függvény implementálása

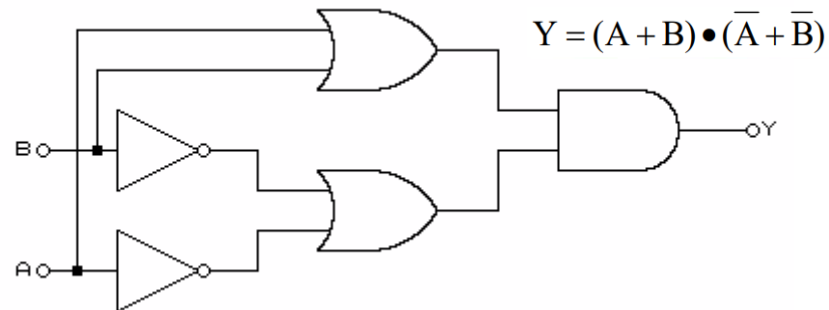
A	B	$Y=A\oplus B$
L	L	L
L	H	H
H	L	H
H	H	L



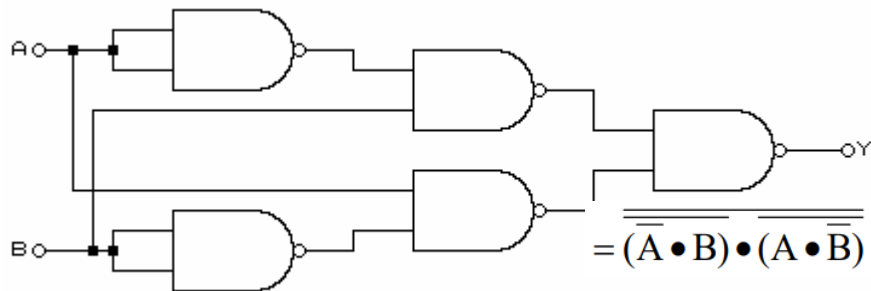
SOP Implementálás



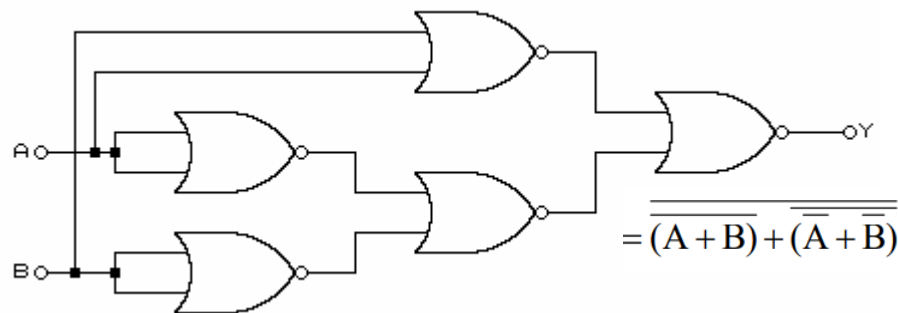
POS Implementálás



Implementálás NAND kapukkal

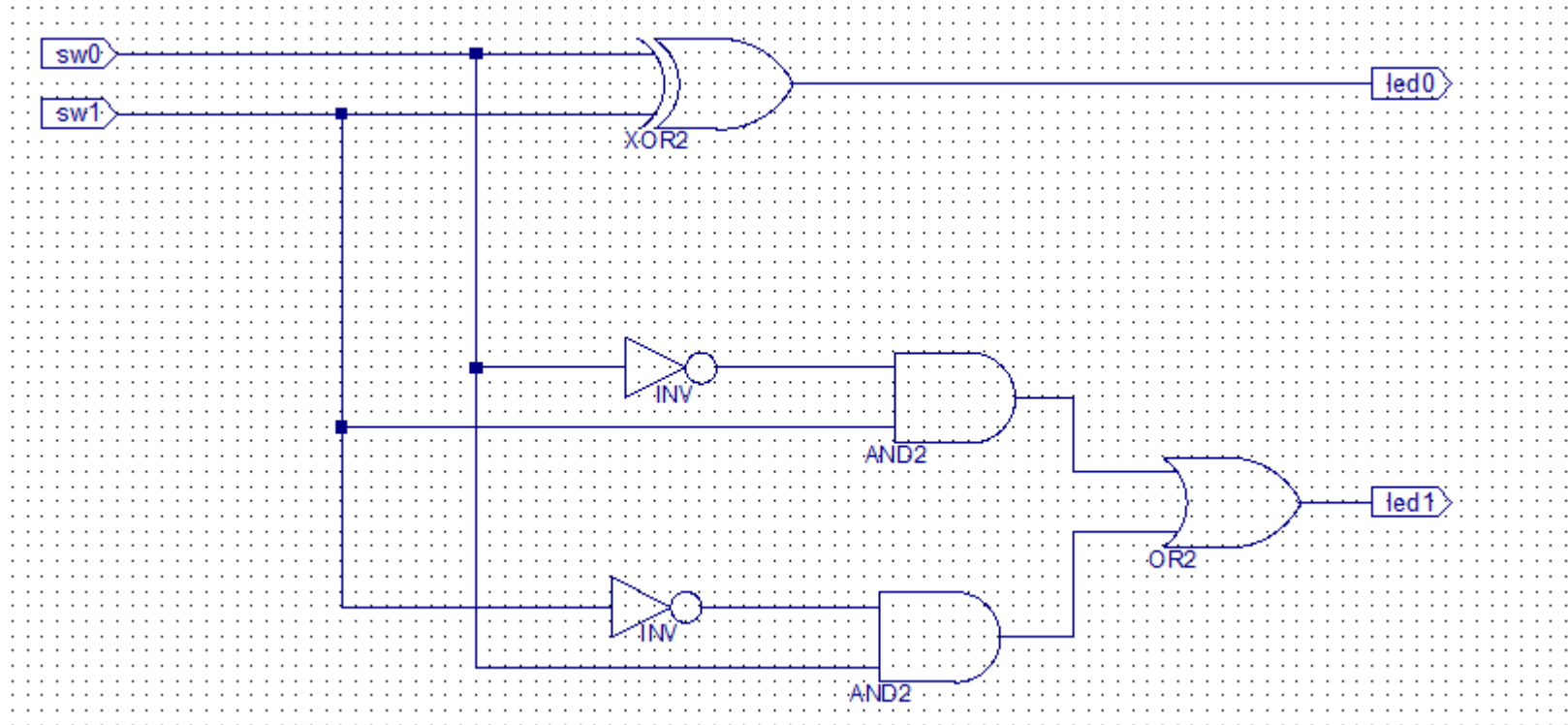


Implementálás NOR kapukkal



XOR függvény implementálása

- Egészítse ki az alábbi rajzot a többi 3 megvalósítás rajzával



Lab4_1 Eredmények

- Konfiguráció generálás, letöltés, működés tesztelése
- A sw0 és sw1 kapcsolókkal alítsa elő a 4 lehetséges bemeneti kombinációt és írja be a led-ek megfelelő állapotait

sw0	sw1	led0 AxorB	led1 (SOP)	led2 (POS)	led3 ("NAND")	led4 ("NOR")
0	0					
0	1					
1	0					
1	1					

- A tapasztalatokat jegyezzük fel a Laboratórium 3. hét eredmények kérdőíven
 - A kimenetek megfelelnek minden esetben?
 - Melyik megvalósítást tartja előnyösebbnek és miért?

Lab4_2 feladat:

BCD – 7 szegmenses dekódoló „a” szegmens

- Hozunk létre egy új projektet
- Adjunk hozzá egy új "schematic" forrásfájlt
- Tervezzük meg az „a” szegmens-t vezérlő áramkört a [Digitális példatár](#) program segítségével.
- Az előző függvényt alakítsuk át De Morgan tételek használatával úgy hogy csak NOT és NAND kaput tartalmazón.
- Helyettesítsük a NOT kapukat NAND kapukkal
- Rajzoljuk le mind a három áramkört az áramkört a schematic editorban.
 - Bemenetek BCD kód: **sw3, sw2, sw1, sw0**
 - Kimenetek:
 - „ca” szegmens katódja = az első áramkör negált kimenete
 - „an0” az első 7 szegmenses kijelző közös anódja – ezt alacsony szintre kell kötni (GND)
 - **led1** és **led0** a második és a harmadik áramkör kimenetei
- A NexysX.UCF fájl hozzáadása és adaptálása

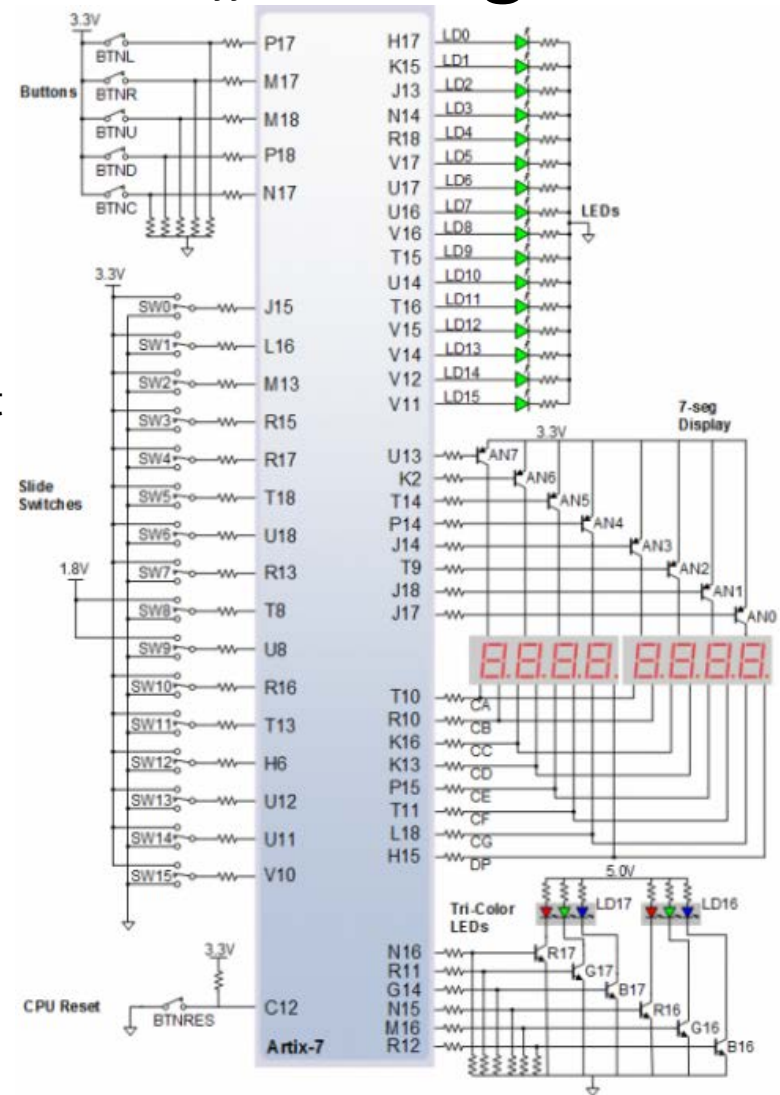
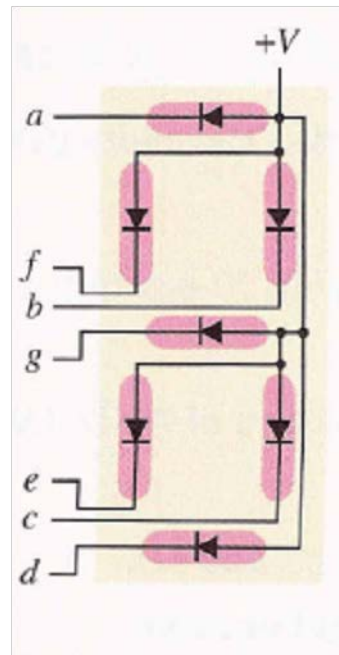
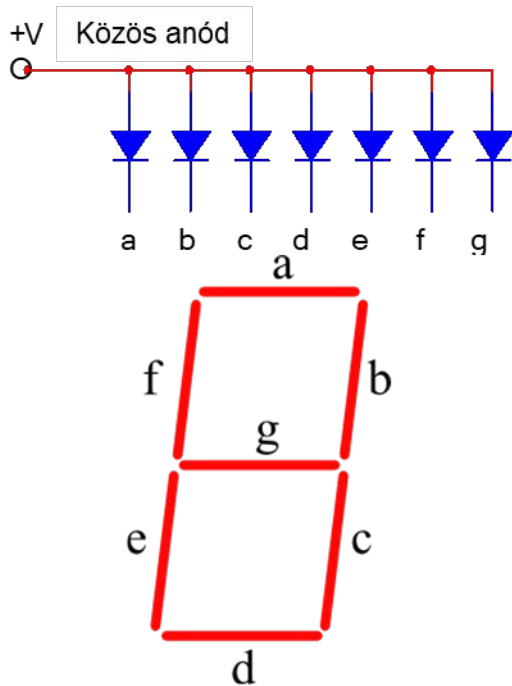
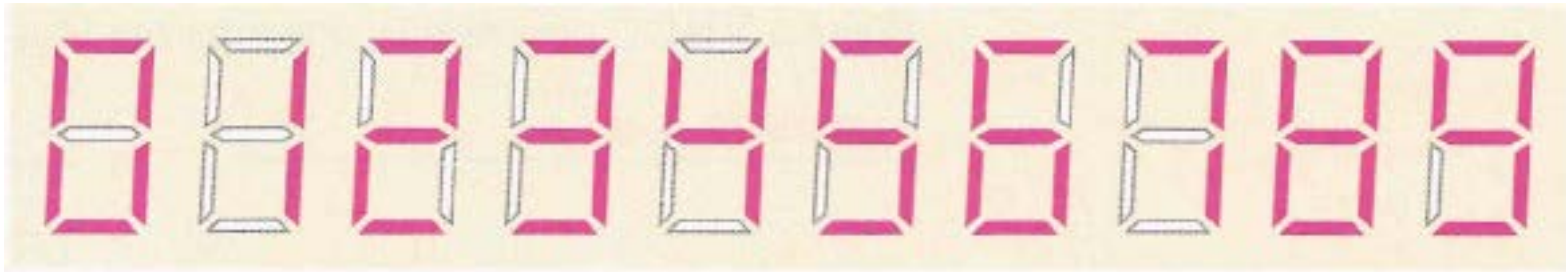


Figure 16. General Purpose I/O devices on the Nexys4 DDR.

BCD – 7 szegmenses dekódoló „a” szegmens



Decimális	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

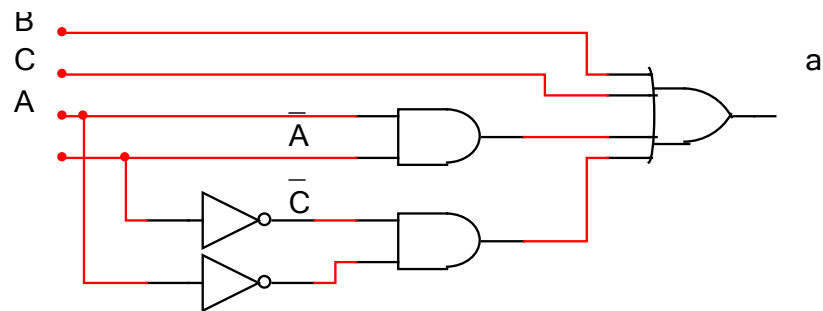
$$a = \overline{DCBA} + \overline{DCB\bar{A}} + \overline{DC\bar{B}A} + \overline{DCBA} + \overline{DCB\bar{A}} + \overline{DC\bar{B}A} + \overline{DCBA} + \overline{DCBA}$$

BCD – 7 szegmenses dekódoló „a” szegmens

$$a = \overline{DCBA} + \overline{DCB\bar{A}} + \overline{DCBA} + \overline{DC\bar{B}A} + \overline{DCB\bar{A}} + \overline{DCBA} + \overline{DC\bar{B}A} + \overline{DCBA}$$

BA \ DC	00	01	11	10
00	1	0	1	1
01	0	1	1	1
11	X	X	X	X
10	1	1	X	X

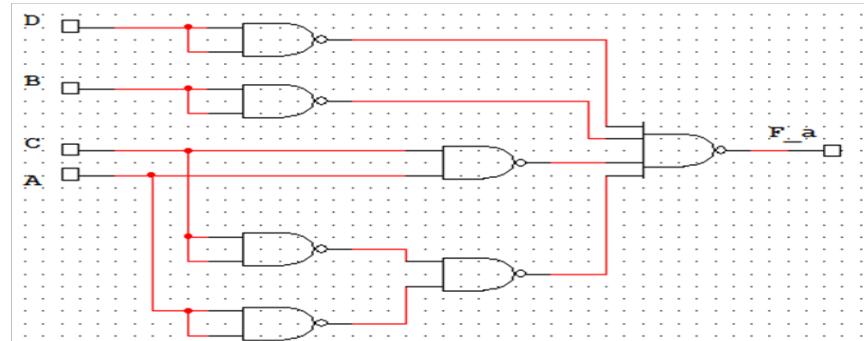
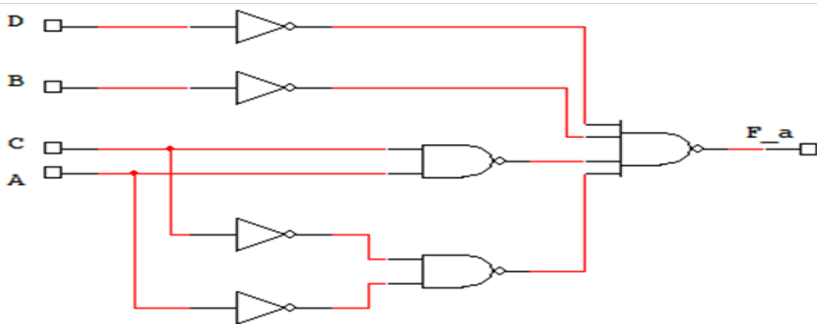
$$a = D + B + CA + \overline{CA}$$



A függvény megvalósítása NAND kapukkal

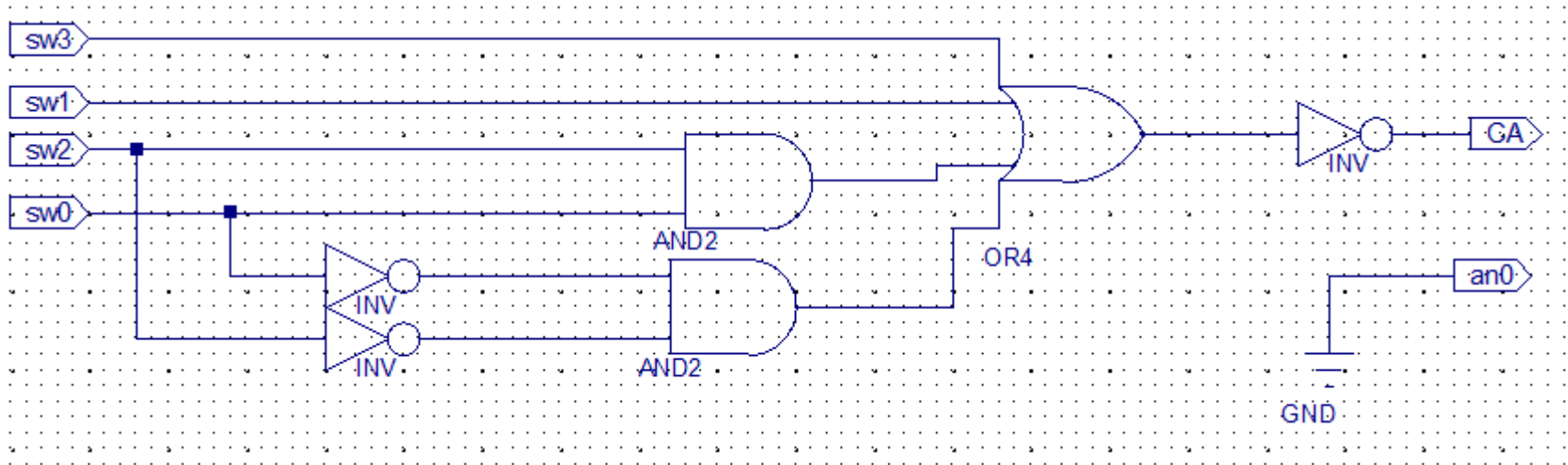
- Az előző függvény átalakítható De Morgan tételek használatával

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



BCD – 7 szegmenses dekódoló „a” szegmens

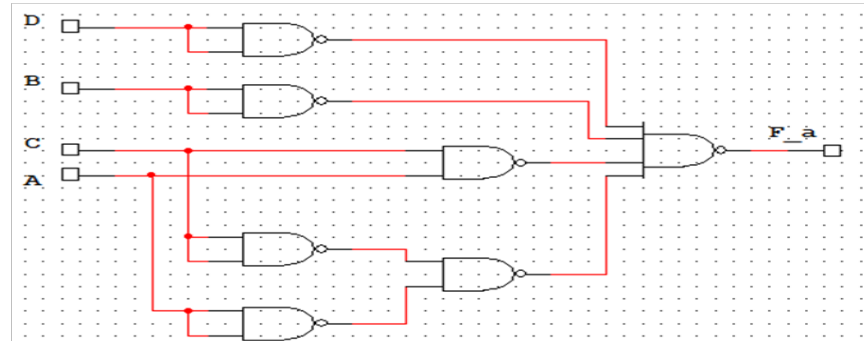
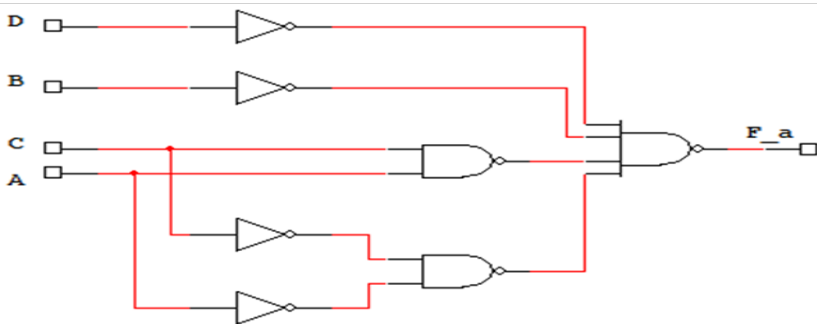
$$a = D + B + CA + \overline{CA}$$



A függvény megvalósítása NAND kapukkal

- Az előző függvény átalakítható De Morgan tételek használatával

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



Lab4_2 Eredmények

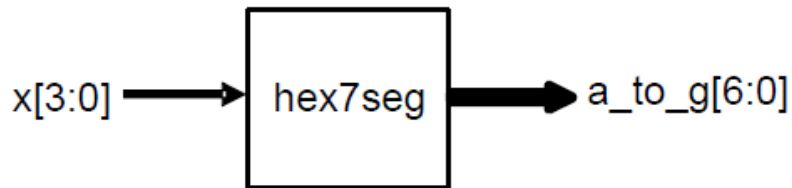
- Konfiguráció generálás, letöltés, működés tesztelése
- A sw0, sw1, sw2 és sw3 kapcsolókkal állítsa elő a 10 érvényes BCD kódot illetve a 6 érvénytelen kódot, figyelje meg és írja be táblázatba az első kijelző „a” szegmens és a led0 illetve a led1 megfelelő állapotait.

sw3	sw2	sw1	sw0	„a”	led0	led1	„a”=led0=led1?
0	0	0	0				
0	0	0	1				
0	0	1	0				
0	0	1	1				
0	1	0	0				
0	1	0	1				
0	1	1	0				
0	1	1	1				
1	0	0	0				
1	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				

- A tapasztalatokat jegyezzük fel a Laboratórium 3. eredmények kérdőíven

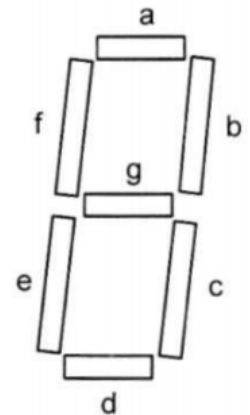
Lab4_3 feladat:

7 szegmenses kijelző vezérlése



x	a	b	c	d	e	f	g
0	0	0	0	0	0	0	1
1	1	0	0	1	1	1	1
2	0	0	1	0	0	1	0
3	0	0	0	0	1	1	0
4	1	0	0	1	1	0	0
5	0	1	0	0	1	0	0
6	0	1	0	0	0	0	0
7	0	0	0	1	1	0	1
8	0	0	0	0	0	0	0
9	0	0	0	0	1	0	0
A	0	0	0	1	0	0	0
b	1	1	0	0	0	0	0
C	0	1	1	0	0	0	1
d	1	0	0	0	0	1	0
E	0	1	1	0	0	0	0
F	0	1	1	1	0	0	0

1 = off
0 = on

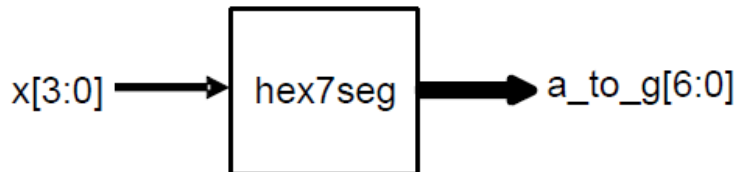
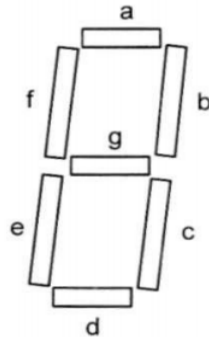


- A szegmenseket külön-külön vezéreljük
- A dot pontokat szintén
- Minden 7 szegmens-t külön kell aktiválni alacsony szinttel (active low)

7 szegmenses kijelző vezérlése

x	a	b	c	d	e	f	g
0	0	0	0	0	0	0	1
1	1	0	0	1	1	1	1
2	0	0	1	0	0	1	0
3	0	0	0	0	1	1	0
4	1	0	0	1	1	0	0
5	0	1	0	0	1	0	0
6	0	1	0	0	0	0	0
7	0	0	0	1	1	0	1
8	0	0	0	0	0	0	0
9	0	0	0	0	1	0	0
A	0	0	0	1	0	0	0
b	1	1	0	0	0	0	0
C	0	1	1	0	0	0	1
d	1	0	0	0	0	1	0
E	0	1	1	0	0	0	0
F	0	1	1	1	0	0	0

1 = off
0 = on



```

module hex7seg ( input [3:0] x,
                 output reg [6:0] a_to_g );
always @(*)
case(x)
0: a_to_g = 7'b0000001;
1: a_to_g = 7'b1001111;
2: a_to_g = 7'b0010010;
3: a_to_g = 7'b0000110;
4: a_to_g = 7'b1001100;
5: a_to_g = 7'b0100100;
6: a_to_g = 7'b0100000;
7: a_to_g = 7'b0001111;
8: a_to_g = 7'b0000000;
9: a_to_g = 7'b0000100;
'hA: a_to_g = 7'b0001000;
'hb: a_to_g = 7'b1100000;
'hC: a_to_g = 7'b0110001;
'hd: a_to_g = 7'b1000010;
'hE: a_to_g = 7'b0110000;
'hF: a_to_g = 7'b0111000;
default: a_to_g = 7'b0000001; // 0
endcase
endmodule

```

- Hozunk létre egy új projektet
- Adjunk hozzá egy új Verilog forrásfájlt
 - Bemenetek: x[3:0]
 - Kimenetek: a_to_g[6:0]
- Adjunk hozzá egy újabb Verilog forrásfájlt, ez lesz a „top module”
 - Bemenetek: x[3:0]
 - Kimenetek: a_to_g[6:0]
- A forráskódok letölthetők a labor oldaláról is.

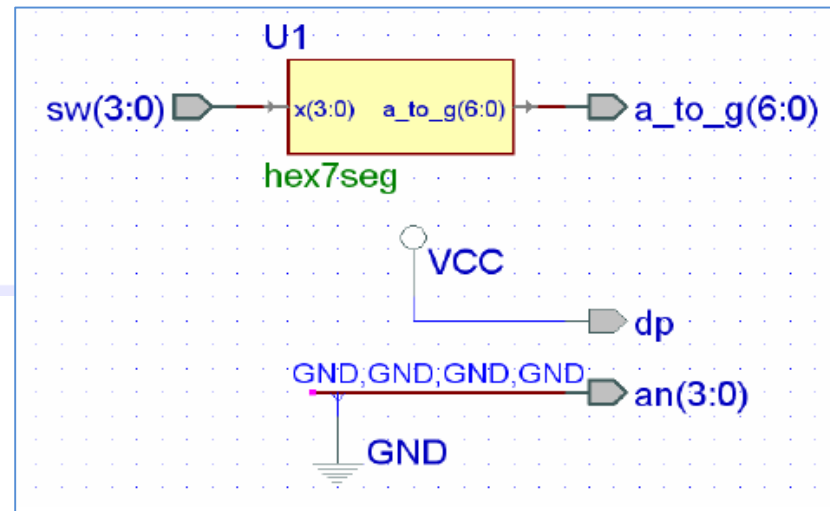
7 szegmenses kijelző vezérlése

```
module hex7seg_top (input [3:0] sw, output [6:0] a_to_g ,
output [3:0] an, output dp);

assign an = 4'b0000; // all digits on
assign dp = 1; // dp off

hex7seg D4 (.x(sw), .a_to_g(a_to_g));
endmodule
```

- A NexysX.UCF fájl hozzáadása és adaptálása
 - Bemenetek: **sw[3:0]**
 - Kimenetek: **a_to_g [6:0], an(3:0), dp.**



- Konfiguráció generálás, letöltés, működés tesztelése
- A sw0, sw1, sw3 és sw3 kapcsolókkal állítsa elő a 16 kódot, ellenőrizze a kijelzőn megjelenő számok és betűk helyességét
- Változtassa a kódot úgy hogy csak egy kijelző működjön.
- A tapasztalatokat jegyezzük fel a Laboratórium 3. eredmények kérdőíven
 - Hogyan érte el azt hogy csak egy kijelzőn jelenjenek meg a számok?
 - Milyen szinttel lehet egy kijelzőt „bekapcsolni” a Nexys 4 kártyán?
 - Mit vezérel a „dp” jel?