

Digitális Technika

Dr. Oniga István
Debreceni Egyetem, Informatikai Kar

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.

2. Laboratóriumi gyakorlat

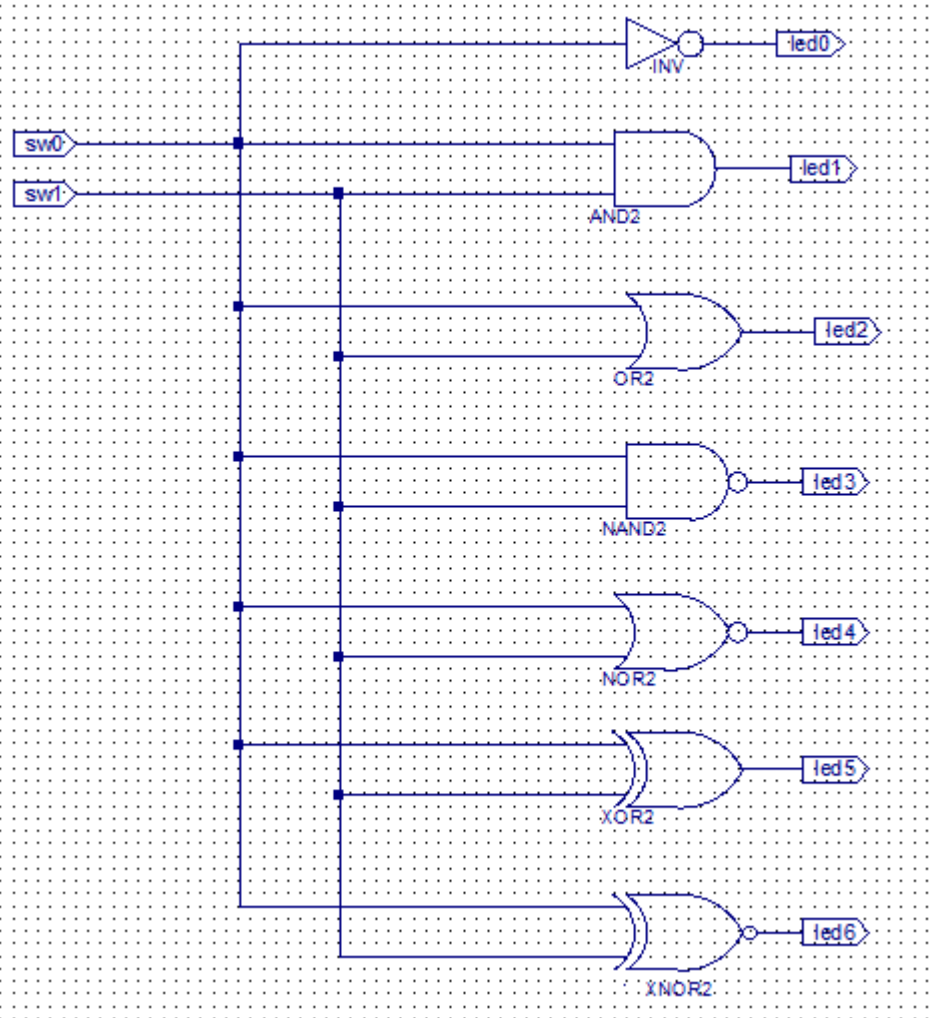
A gyakorlat célja:

- Logikai kapuk implementációja HDL nyelven
- Boolean algebra
 - Asszociativitás tétel
 - Disztributivitás tétel
 - Abszorpció tétel
 - De Morgan tételek

Lab2_1 feladat:

2 változós logikai függvények implementációja
- HDL terv leírással -

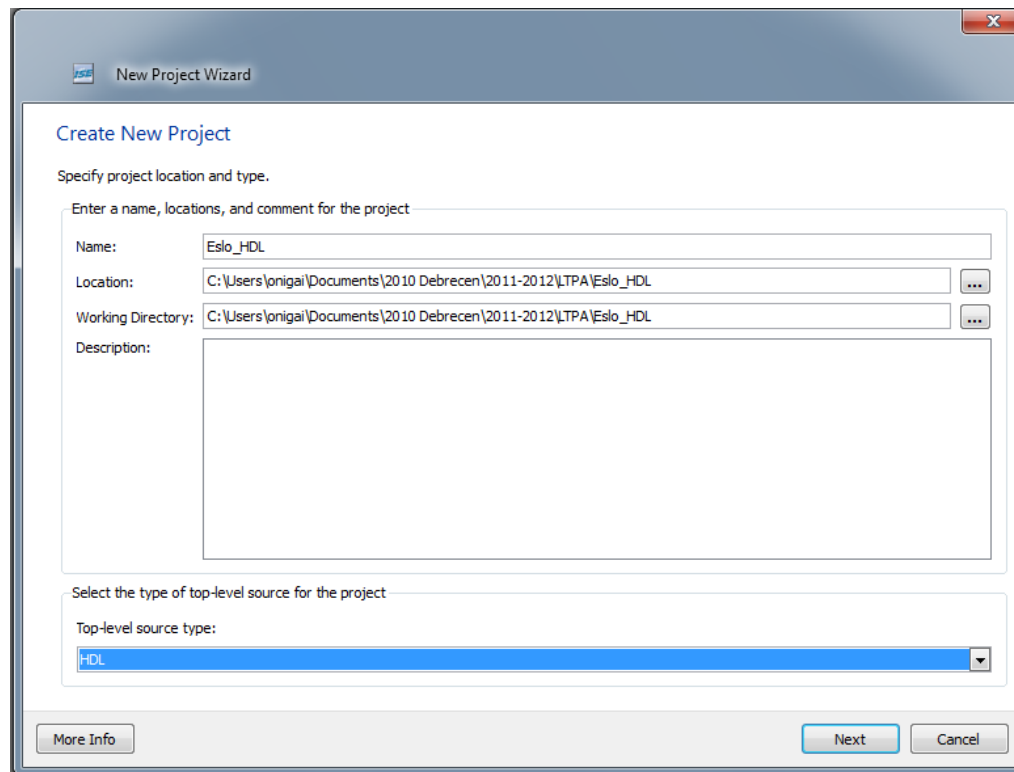
Logikai kapuk implementációja HDL nyelven



```
module elsoHDL(  
    input sw0,  
    input sw1,  
    output led0, led1, led2, led3, led4, led5, led6  
  
    );  
assign led0 = ~ sw0;  
assign led1 = sw0 & sw1;  
assign led2 = sw0 | sw1;  
assign led3 = ~ (sw0 & sw1);  
assign led4 = ~ (sw0 | sw1);  
assign led5 = sw0 ^ sw1;  
assign led6 = sw0 ~^ sw1;  
  
endmodule
```

A project létrehozása

- **A fejlesztőkörnyezet elindítása:** Start -> Programs\Xilinx ISE Design Suite 14.7\ISE Design Tools\Project Navigator.
- Új projekt (*File*→*New Project*) - minden projektnek külön könyvtárat hoz létre,
- Projektünk neve legyen „*EsloHDL*” (javaslat),
- **Legmagasabb szintű** forrásként HDL típust adjunk meg!



FPGA tulajdonságainak beállítása

- A **Next** gombra kattintás után megjelenő **Device Properties** mezőben a **Value** oszlop legördülő listáiból válassza az alábbi értékeket:

NEXYS 2 board

Device Family: Spartan3E

Device: xc3s500E

Package: FG320

Speed Grade: -4

Synthesis Tool: XST (VHDL/Verilog)

Simulator: ISim (VHDL/Verilog)

Preferred Language: Verilog

NEXYS 4 DDR board

Family: Artix7

Device: XC7A100T

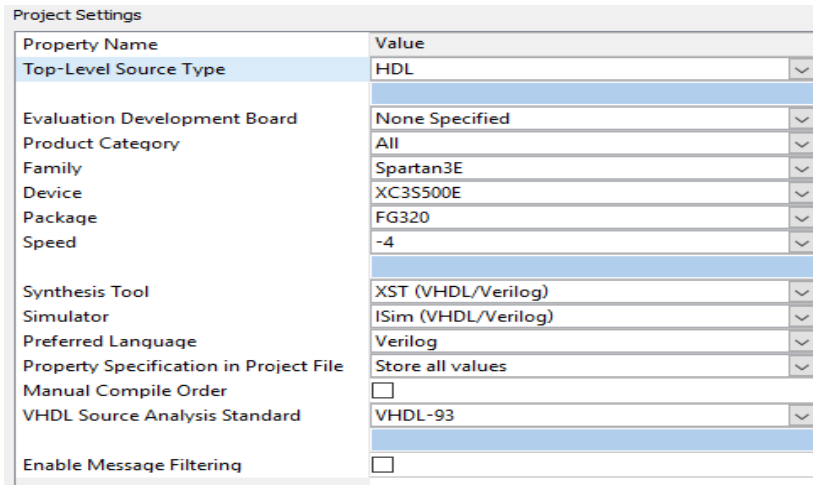
Package: CSG324

Speed Grade: -3

Synthesis Tool: XST (VHDL/Verilog)

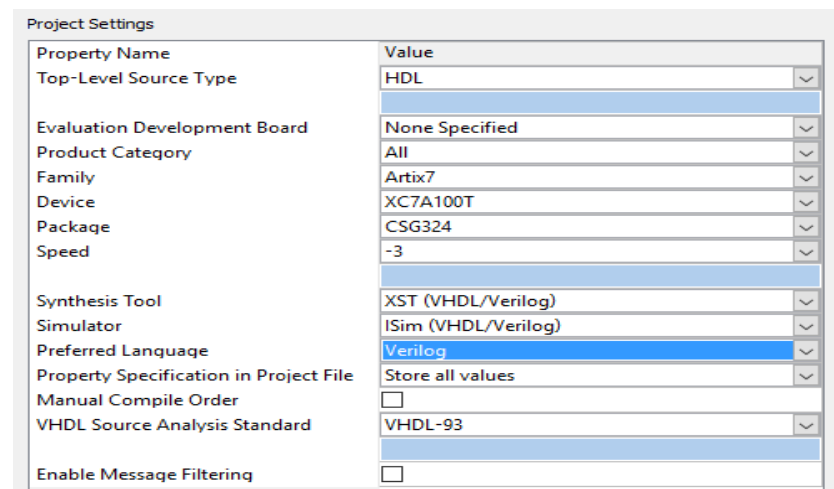
Simulator: ISim (VHDL/Verilog)

Preferred Language: Verilog



Project Settings

Property Name	Value
Top-Level Source Type	HDL
Evaluation Development Board	None Specified
Product Category	All
Family	Spartan3E
Device	XC3S500E
Package	FG320
Speed	-4
Synthesis Tool	XST (VHDL/Verilog)
Simulator	ISim (VHDL/Verilog)
Preferred Language	Verilog
Property Specification in Project File	Store all values
Manual Compile Order	<input type="checkbox"/>
VHDL Source Analysis Standard	VHDL-93
Enable Message Filtering	<input type="checkbox"/>



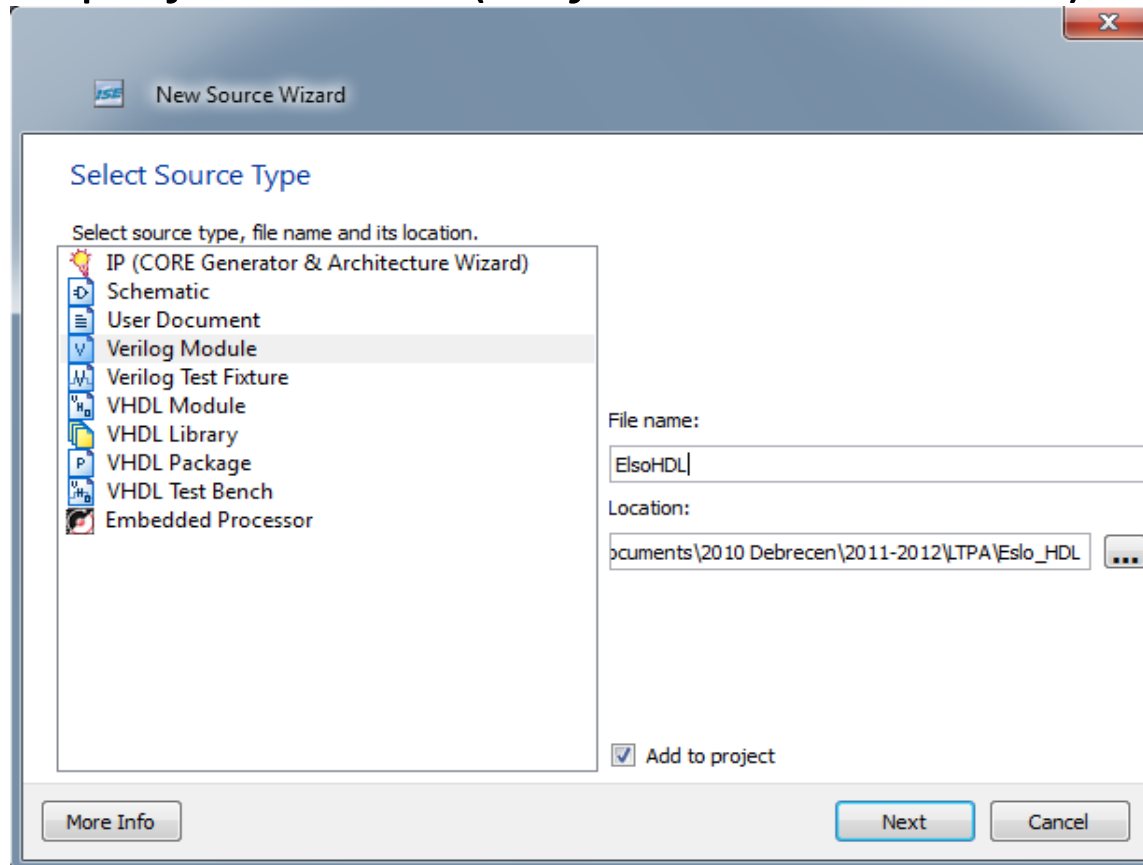
Project Settings

Property Name	Value
Top-Level Source Type	HDL
Evaluation Development Board	None Specified
Product Category	All
Family	Artix7
Device	XC7A100T
Package	CSG324
Speed	-3
Synthesis Tool	XST (VHDL/Verilog)
Simulator	ISim (VHDL/Verilog)
Preferred Language	Verilog
Property Specification in Project File	Store all values
Manual Compile Order	<input type="checkbox"/>
VHDL Source Analysis Standard	VHDL-93
Enable Message Filtering	<input type="checkbox"/>

A **Next**, majd **Finish** gombra kattintva elkészül az üres project.

Tervezői file (HDL) hozzáadása

- Hozzuk létre forrás fájlt (*Project*→*New Source...*)!
- A forrásunk típusa **Verilog Module**, a neve ElsoHDL!
- Amennyiben nem történt meg automatikusan, adjuk hozzá a forrás fájlt a projektünkhöz (*Project*→*Add Source...*)



Portok megadása

New Source Wizard



← Define Module

Specify ports for module.

Module name

Port Name	Direction	Bus	MSB	LSB
sw0	input	<input type="checkbox"/>		
sw1	input	<input type="checkbox"/>		
led0	output	<input type="checkbox"/>		
	input	<input type="checkbox"/>		
	input	<input type="checkbox"/>		
	input	<input type="checkbox"/>		
	input	<input type="checkbox"/>		
	input	<input type="checkbox"/>		
	input	<input type="checkbox"/>		
	input	<input type="checkbox"/>		
	input	<input type="checkbox"/>		
	input	<input type="checkbox"/>		
	input	<input type="checkbox"/>		
	input	<input type="checkbox"/>		
	input	<input type="checkbox"/>		

More Info

< Back

Next >

Cancel

Generált HDL fájl

```
1  `timescale 1ns / 1ps
2  ///////////////////////////////////////////////////////////////////
3  // Company:
4  // Engineer:
5  //
6  // Create Date:    11:35:13 02/18/2021
7  // Design Name:
8  // Module Name:    alsoHDL
9  // Project Name:
10 // Target Devices:
11 // Tool versions:
12 // Description:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 ///////////////////////////////////////////////////////////////////
21 module alsoHDL(
22     input sw0,
23     input sw1,
24     output led0
25 );
26
27
28 endmodule
29
```

Generált HDL fájl kiegészítése

```
1 `timescale 1ns / 1ps
2 ///////////////////////////////////////////////////////////////////
3 // Company:
4 // Engineer:
5 //
6 // Create Date:      11:35:13 02/18/2021
7 // Design Name:
8 // Module Name:      elsoHDL
9 // Project Name:
10 // Target Devices:
11 // Tool versions:
12 // Description:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 ///////////////////////////////////////////////////////////////////
21 module elsoHDL(
22     input sw0,
23     input sw1,
24     output led0
25 );
26 Ide kerül a felhasználói kódrészlet
27
28 endmodule
29
```

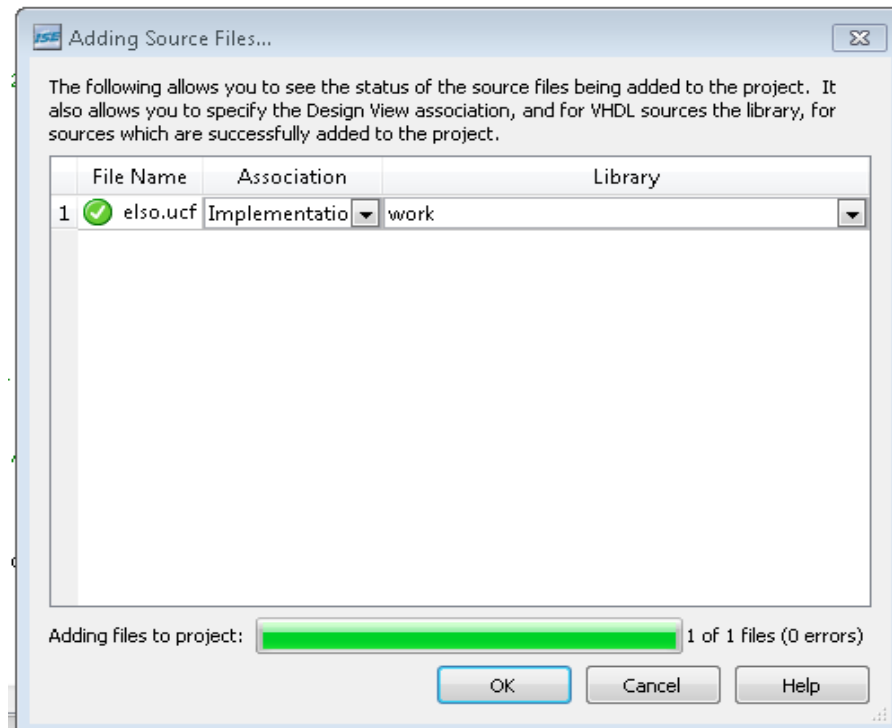
- Egészítsük ki a generált kódot az összes led hozzáadásával illetve a működést leíró résszel

```
module elsoHDL(
    input sw0,
    input sw1,
    output led0, led1, led2, led3, led4, led5, led6
);
    assign led0 = ~ sw0;
    assign led1 = sw0 & sw1;
    assign led2 = sw0 | sw1;
    assign led3 = ~ (sw0 & sw1);
    assign led4 = ~ (sw0 | sw1);
    assign led5 = sw0 ^ sw1;
    assign led6 = sw0 ~^ sw1;

endmodule
```

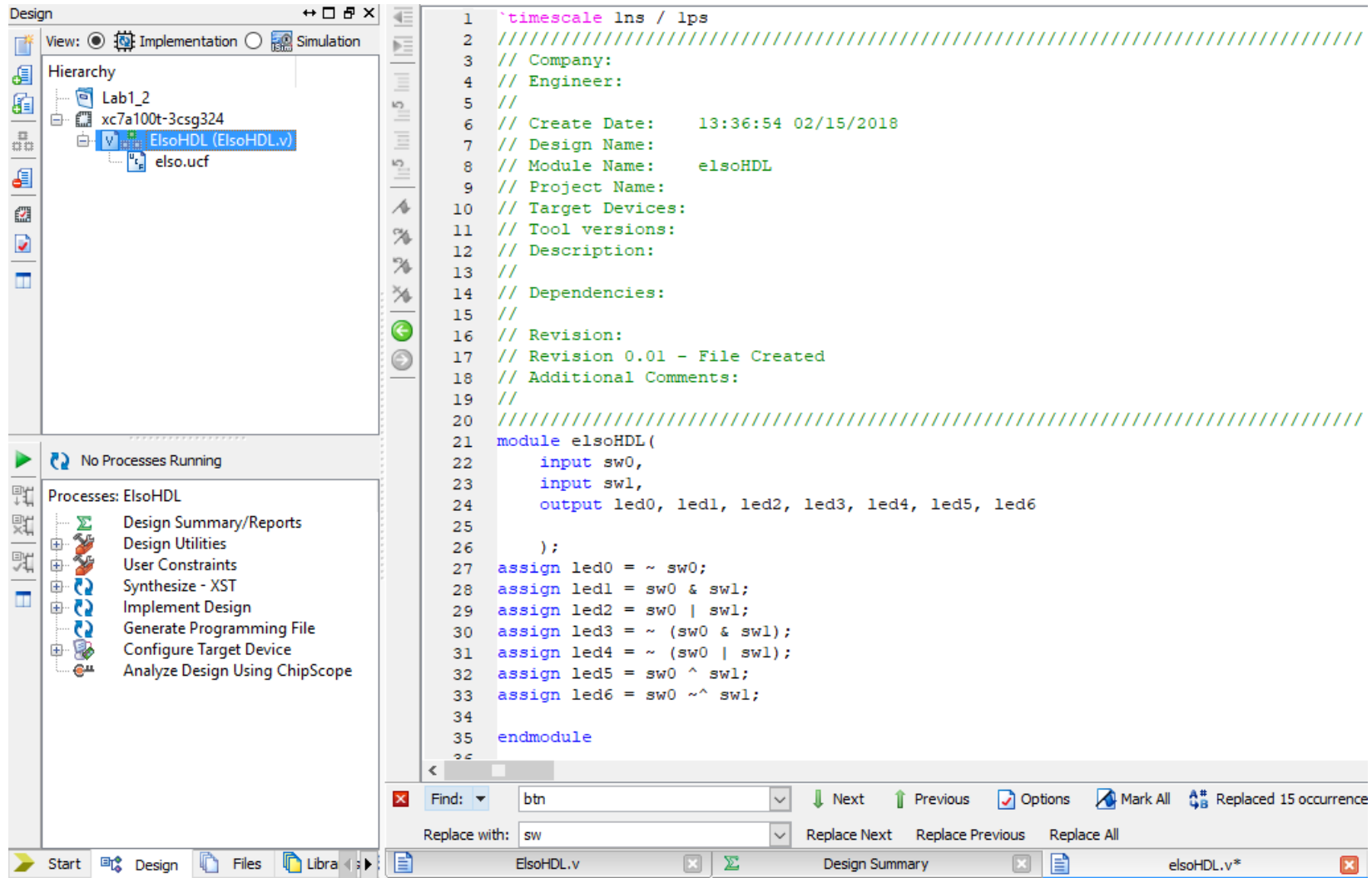
Láb-hozzárendelés

- egy constraint fájlt adunk a projekthez.
- Válasszuk ki a **Project / Add Copy of Source** menüpontot, a felbukkanó ablakban pedig **keresük meg az előző projektben használt eslo.ucf fájlt**.
- **OK** gomb megnyomása után a **Sources** ablakban meg is jelenik a *elso.ucf* fájl.



Láb-hozzárendelés

- egy constraint fájlt adunk a projekthez.



The screenshot displays the Xilinx Vivado IDE interface. On the left, the 'Design' window shows a project hierarchy for 'Lab1_2' on a 'xc7a100t-3csg324' device. The 'Els0HDL (Els0HDL.v)' file is selected, and its associated 'els0.ucf' constraint file is visible below it. The 'Processes' window shows 'No Processes Running' and a list of tasks including 'Design Summary/Reports', 'Design Utilities', 'User Constraints', 'Synthesize - XST', 'Implement Design', 'Generate Programming File', 'Configure Target Device', and 'Analyze Design Using ChipScope'.

The main editor window displays the Verilog code for 'els0HDL.v'. The code includes a header with a timescale and various comments, followed by a module definition for 'els0HDL'. The module has two inputs, 'sw0' and 'sw1', and six outputs, 'led0' through 'led6'. The logic is implemented using 'assign' statements:

```
1 `timescale 1ns / 1ps
2 ///////////////////////////////////////////////////////////////////
3 // Company:
4 // Engineer:
5 //
6 // Create Date:    13:36:54 02/15/2018
7 // Design Name:
8 // Module Name:    els0HDL
9 // Project Name:
10 // Target Devices:
11 // Tool versions:
12 // Description:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 ///////////////////////////////////////////////////////////////////
21 module els0HDL(
22     input sw0,
23     input sw1,
24     output led0, led1, led2, led3, led4, led5, led6
25 );
26
27     assign led0 = ~ sw0;
28     assign led1 = sw0 & sw1;
29     assign led2 = sw0 | sw1;
30     assign led3 = ~ (sw0 & sw1);
31     assign led4 = ~ (sw0 | sw1);
32     assign led5 = sw0 ^ sw1;
33     assign led6 = sw0 ~^ sw1;
34
35 endmodule
```

At the bottom, a search and replace dialog is open, showing 'Find: btn' and 'Replace with: sw'. The 'Replaced 15 occurrence' status is visible.

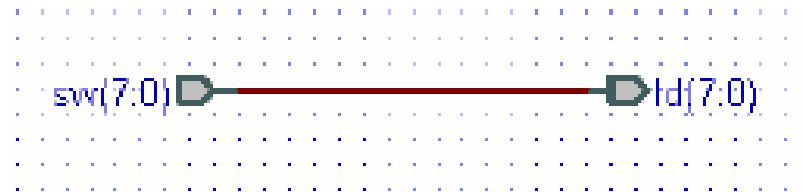
Eredmények

- A sw0 és sw1 nyomógombok megnyomásával alítsa elő a négy lehetséges bemeneti kombinációt és írja be a led-ek megfelelő állapotjait
- Ellenőrizze a minden logikai függvények igazság táblázatát

sw0	sw1	led0 NOT	led1 AND	led2 OR	led3 NAND	led4 NOR	led5 XOR	led6 XNOR
0	0							
0	1							
1	0							
1	1							

Lab2_2 feladat:

8 db LED vezérlése a 8 bites DIP kapcsolóval
- HDL terv leírással -

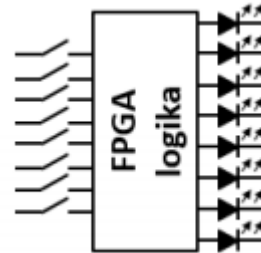


Lab2_2

- A 8 bemenet sw [7:0] és a 8 kimenetet ld [7:0] kezelhetjük egyedi bitenként mind az előző két példában (egyenkénti megadással többet kell gépelni) vagy vektorosan.

- egyedi bitenkénti leírás:

```
module sw2led(  
    input sw0, sw1, sw2, sw3, sw4, sw5, sw6, sw7,  
    output ld0, ld1, ld2, ld3, ld4, ld5, ld6, ld7  
);  
assign ld0 = sw0;  
assign ld1 = sw1;  
assign ld2 = sw2;  
assign ld3 = sw3;  
assign ld4 = sw4;  
assign ld5 = sw5;  
assign ld6 = sw6;  
assign ld7 = sw7;  
endmodule
```



NEXYS 4 board

Switches

NET "sw0"	LOC=J15 IOSTANDARD=LVCMOS33;
NET "sw1"	LOC=L16 IOSTANDARD=LVCMOS33;
NET "sw2"	LOC=M13 IOSTANDARD=LVCMOS33;
NET "sw3"	LOC=R15 IOSTANDARD=LVCMOS33;
NET "sw4"	LOC=R17 IOSTANDARD=LVCMOS33;
NET "sw5"	LOC=T18 IOSTANDARD=LVCMOS33;
NET "sw6"	LOC=U18 IOSTANDARD=LVCMOS33;
NET "sw7"	LOC=R13 IOSTANDARD=LVCMOS33;

LEDs

NET "ld0"	LOC=H17 IOSTANDARD=LVCMOS33;
NET "ld1"	LOC=K15 IOSTANDARD=LVCMOS33;
NET "ld2"	LOC=J13 IOSTANDARD=LVCMOS33;
NET "ld3"	LOC=N14 IOSTANDARD=LVCMOS33;
NET "ld4"	LOC=R18 IOSTANDARD=LVCMOS33;
NET "ld5"	LOC=V17 IOSTANDARD=LVCMOS33;
NET "ld6"	LOC=U17 IOSTANDARD=LVCMOS33;
NET "ld7"	LOC=U16 IOSTANDARD=LVCMOS33;

NEXYS 2 board

8 kapcsoló, balról jobbra számozva

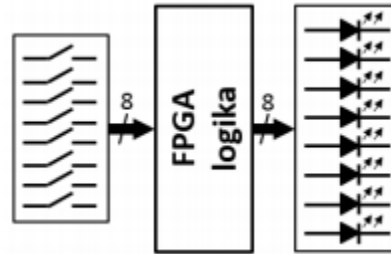
NET "sw7"	LOC = "R17";
NET "sw6"	LOC = "N17";
NET "sw5"	LOC = "L13";
NET "sw4"	LOC = "L14";
NET "sw3"	LOC = "K17";
NET "sw2"	LOC = "K18";
NET "sw1"	LOC = "H18";
NET "sw0"	LOC = "G18";

8 LED, balról jobbra számozva

NET "ld7"	LOC = "R4";
NET "ld6"	LOC = "F4";
NET "ld5"	LOC = "P15";
NET "ld4"	LOC = "E17";
NET "ld3"	LOC = "K14";
NET "ld2"	LOC = "K15";
NET "ld1"	LOC = "J15";
NET "ld0"	LOC = "J14";

Lab2_2

- Bitvektor leírás:



```
module sw2led(  
    input [7:0] sw,  
    output [7:0] ld  
);  
assign ld = sw;  
endmodule
```

NEXYS 2 board

8 kapcsoló, balról jobbra számozva

NET "sw<7>" LOC = "R17";
NET "sw<6>" LOC = "N17";
NET "sw<5>" LOC = "L13";
NET "sw<4>" LOC = "L14";
NET "sw<3>" LOC = "K17";
NET "sw<2>" LOC = "K18";
NET "sw<1>" LOC = "H18";
NET "sw<0>" LOC = "G18";

8 LED, balról jobbra számozva

NET "ld<7>" LOC = "R4";
NET "ld<6>" LOC = "F4";
NET "ld<5>" LOC = "P15";
NET "ld<4>" LOC = "E17";
NET "ld<3>" LOC = "K14";
NET "ld<2>" LOC = "K15";
NET "ld<1>" LOC = "J15";
NET "ld<0>" LOC = "J14";

NEXYS 4 board

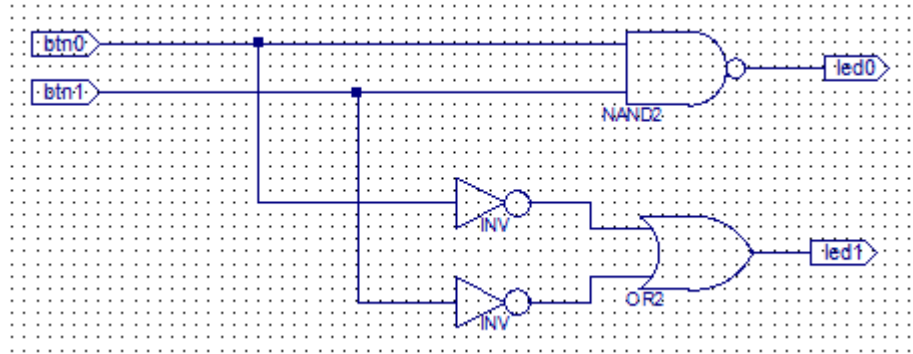
Switches

NET "sw<0>" LOC=J15 | IOSTANDARD=LVCMOS33;
NET "sw<1>" LOC=L16 | IOSTANDARD=LVCMOS33;
NET "sw<2>" LOC=M13 | IOSTANDARD=LVCMOS33;
NET "sw<3>" LOC=R15 | IOSTANDARD=LVCMOS33;
NET "sw<4>" LOC=R17 | IOSTANDARD=LVCMOS33;
NET "sw<5>" LOC=T18 | IOSTANDARD=LVCMOS33;
NET "sw<6>" LOC=U18 | IOSTANDARD=LVCMOS33;
NET "sw<7>" LOC=R13 | IOSTANDARD=LVCMOS33;

LEDs

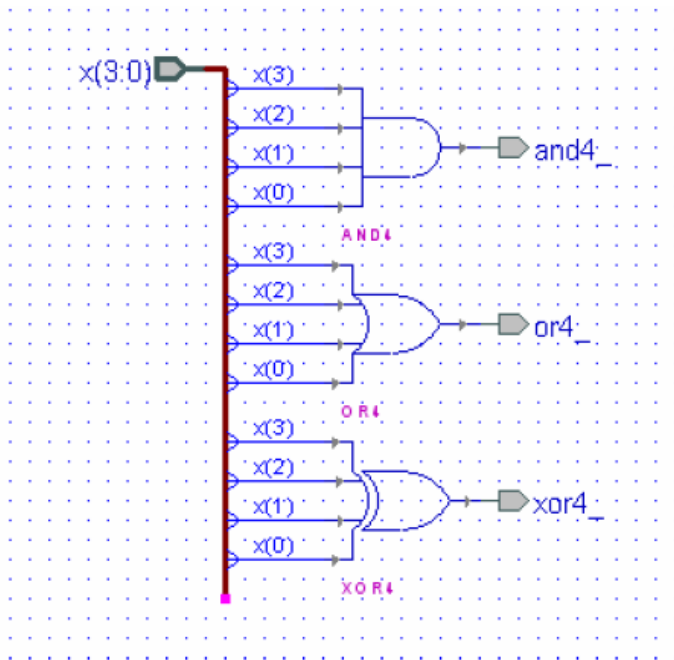
NET "ld<0>" LOC=H17 | IOSTANDARD=LVCMOS33;
NET "ld<1>" LOC=K15 | IOSTANDARD=LVCMOS33;
NET "ld<2>" LOC=J13 | IOSTANDARD=LVCMOS33;
NET "ld<3>" LOC=N14 | IOSTANDARD=LVCMOS33;
NET "ld<4>" LOC=R18 | IOSTANDARD=LVCMOS33;
NET "ld<5>" LOC=V17 | IOSTANDARD=LVCMOS33;
NET "ld<6>" LOC=U17 | IOSTANDARD=LVCMOS33;
NET "ld<7>" LOC=U16 | IOSTANDARD=LVCMOS33;

Szorgalmi „HDL” feladat



btn0	btn1	led0=(AB)'	led1=A'+B'
0	0		
0	1		
1	0		
1	1		

Szorgalmi feladat



```
module gates4b (  
    input [3:0] x ,  
    output and4_ ,  
    output or4_ ,  
    output xor4_  
);  
    assign and4_ = &x;  
    assign or4_ = |x;  
    assign xor4_ = ^x;  
endmodule
```

Lab2_3a feladat:

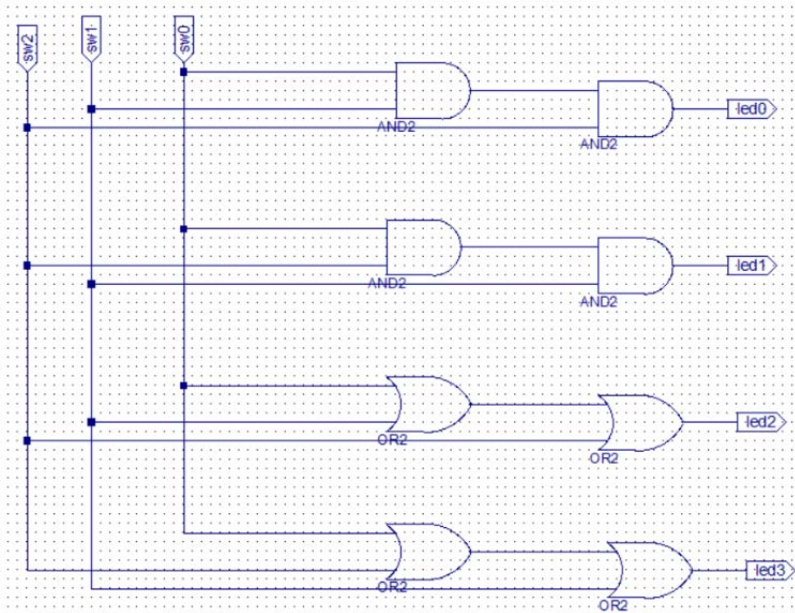
3 változós logikai függvények tesztelése - Asszociativitás tétele -

- Hozunk létre egy új projektet
- Adjunk hozzá egy új "schematic" forrásfájlt
- Rajzoljuk le az ábrán látható áramköröket.
- A NexysX.UCF fájl hozzáadása és adaptálása
 - Bemenetek: **sw[2:0]**
 - sw0 -> A; sw1 -> B; sw2->C
 - Kimenetek: **led[3:0]**
- Konfiguráció generálás, letöltés, működés tesztelése
- A tapasztalatokat jegyezzük fel a Laboratórium 2. eredmények kérdőíven

$$A \bullet (B \bullet C) = (A \bullet B) \bullet C$$



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



Lab2_3a Eredmények

- A sw0, sw1 és sw2 kapcsolókkal alítsa elő a 8 lehetséges bemeneti kombinációt és írja be a led-ek megfelelő állapotait

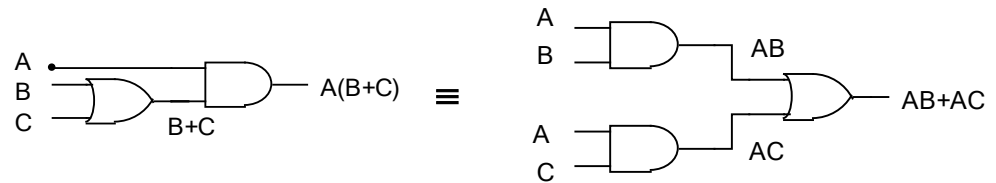
sw0	sw1	sw2	led0 $A(BC)$	led1 $(AB)C$	led2 $A+(B+C)$	led3 $(A+B)+C$
0	0	0				
0	0	1				
0	1	0				
0	1	1				
1	0	0				
1	0	1				
1	1	0				
1	1	1				

Lab2_3b feladat:

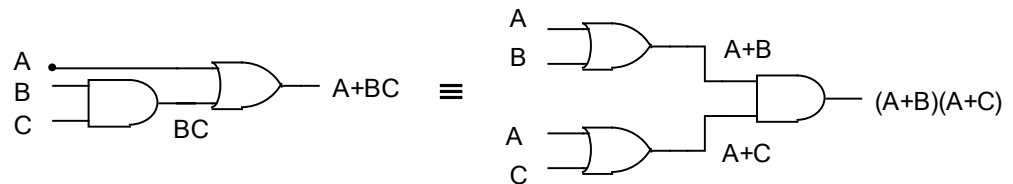
3 változós logikai függvények tesztelése - Disztributivitás tétele -

- Hozunk létre egy új projektet
- Adjunk hozzá egy új "schematic" forrásfájlt
- Rajzoljuk le az ábrán látható áramköröket.
- A NexysX.UCF fájl hozzáadása és adaptálása
 - Bemenetek: **sw[2:0]**
 - sw0 -> A; sw1 -> B; sw2->C
 - Kimenetek: **led[3:0]**
- Konfiguráció generálás, letöltés, működés tesztelése
- A tapasztalatokat jegyezzük fel a Laboratórium 2. eredmények kérdőíven

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



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



Lab2 3b Eredmények

- A sw0, sw1 és sw2 kapcsolókkal alítsa elő a 8 lehetséges bemeneti kombinációt és írja be a led-ek megfelelő állapotait

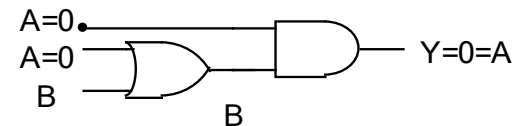
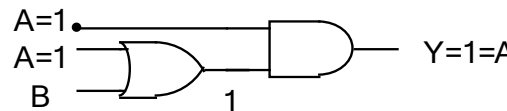
sw0	sw1	sw2	led0 $A(B+C)$	led1 $AB+AC$	led2 $A+BC$	led3 $(A+B)(A+C)$
0	0	0				
0	0	1				
0	1	0				
0	1	1				
1	0	0				
1	0	1				
1	1	0				
1	1	1				

Lab2_3c feladat:

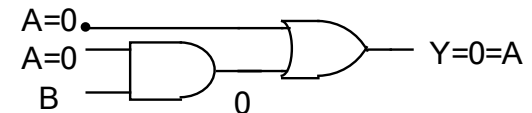
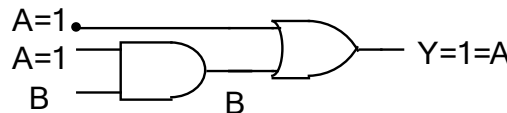
3 változós logikai függvények tesztelése - Abszorpció tételei -

- Hozunk létre egy új projektet
- Adjunk hozzá egy új "schematic" forrásfájlt
- Rajzoljuk le az ábrán látható 2 áramkört.
- A NexysX.UCF fájl hozzáadása és adaptálása
 - Bemenetek: **sw[1:0]**
 - sw0 -> A; sw1 -> B
 - Kimenetek: **led[1:0]**

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



$$A + A \bullet B = A$$



- Konfiguráció generálás, letöltés, működés tesztelése
- A tapasztalatokat jegyezzük fel a Laboratórium 2. eredmények kérdőíven

Lab2_3c Eredmények

- A sw0 és sw1 kapcsolókkal alítsa elő a 4 lehetséges bemeneti kombinációt és írja be a led-ek megfelelő állapotait

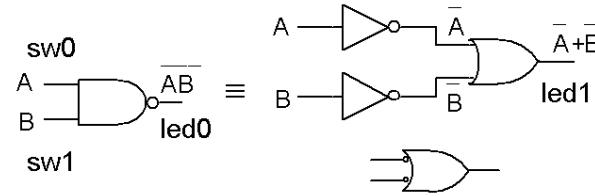
sw0	sw1	led0 $A(A+B)$	led1 $A+AB$
0	0		
0	1		
1	0		
1	1		

Lab2_4a feladat:

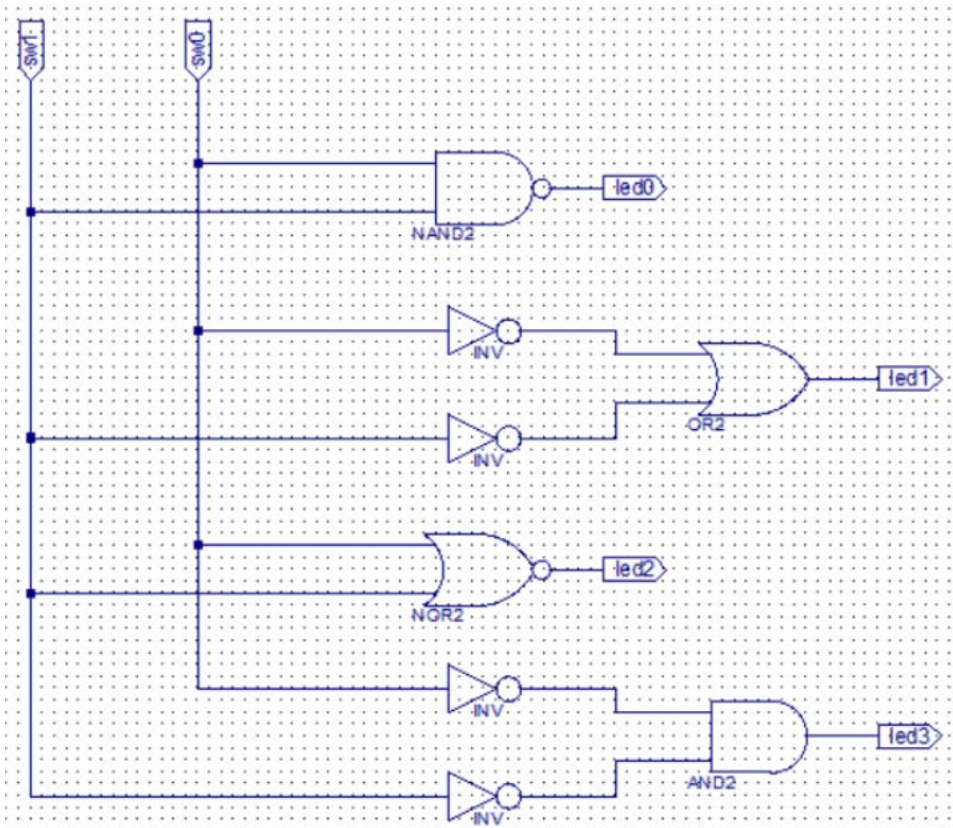
De Morgan tételei tesztelése - 2 változóra

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

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



- Hozunk létre egy új projektet
- Adjunk hozzá egy új "schematic" forrásfájlt
- Rajzoljuk le az ábrán látható áramköröket.
- A Nexysx.UCF fájl hozzáadása és adaptálása
 - Bemenetek: **sw[1:0]**
 - sw0 -> A; sw1 -> B; sw2->C
 - Kimenetek: **led[3:0]**
- Konfiguráció generálás, letöltés, működés tesztelése
- A tapasztalatokat jegyezzük fel a Laboratórium 2. eredmények kérdőíven



Lab2_4a Eredmények

- A sw0 és sw1 kapcsolókkal alítsa elő a 4 lehetséges bemeneti kombinációt és írja be a led-ek megfelelő állapotait

sw0	sw1	led0 $/(AB)$	led1 $/A+/B$	led2 $/(A+B)$	led3 $/A*/B$
0	0				
0	1				
1	0				
1	1				

Lab2_4b feladat:

De Morgan tételei tesztelése - 3 változóra

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

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

- Hozunk létre egy új projektet
- Adjunk hozzá egy új "schematic" forrásfájlt
- Rajzoljuk le az áramköröket.
- A NexysX.UCF fájl hozzáadása és adaptálása

– Bemenetek: **sw[2:0]**

sw0 -> A; sw1 -> B; sw2 -> C

– Kimenetek: **led[3:0]**

sw0	sw1	sw2	led0 /(ABC)	led1 /A+/B+/C	led2 /(A+B+C)	led3 /A*/B*/C
0	0	0				
0	0	1				
0	1	0				
0	1	1				
1	0	0				
1	0	1				
1	1	0				
1	1	1				

- Konfiguráció generálás, letöltés, működés tesztelése
- A sw0, sw1 és sw2 kapcsolókkal alítsa elő a 8 lehetséges bemeneti kombinációt és írja be a led-ek megfelelő állapotait
- A tapasztalatokat jegyezzük fel a Laboratórium 2. eredmények kérdőíven

Lab2_4c (szorgalmi feladat):

De Morgan tételei általánosítása

$$X = \overline{A \cdot B + A \cdot \overline{C} + ABC} = \overline{A \cdot B} \cdot \overline{A \cdot \overline{C}} \cdot \overline{ABC}$$

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

- Hozunk létre egy új projektet
- Adjunk hozzá egy új "schematic" forrásfájlt
- Rajzoljuk le az X, Y függvények által leírt áramköröket.
- A NexysX.UCF fájl hozzáadása és adaptálása
 - Bemenetek: **sw[2:0]**
sw0 -> A; sw1 -> B; sw2 -> C
 - Kimenetek: **led[3:0]**
- Konfiguráció generálás, letöltés, működés tesztelése
- A sw0, sw1 és sw2 kapcsolókkal alítsa elő a 8 lehetséges bemeneti kombinációt és írja be a led-ek megfelelő állapotait
- A tapasztalatokat jegyezzük fel a Laboratórium 2. eredmények kérdőíven

sw0	sw1	sw2	led0 X1	led1 X2	led2 Y1	led3 Y2
0	0	0				
0	0	1				
0	1	0				
0	1	1				
1	0	0				
1	0	1				
1	1	0				
1	1	1				