



**BUDAPESTI MŰSZAKI ÉS GAZDASÁGTUDOMÁNYI EGYETEM**  
**VILLAMOSMÉRNÖKI ÉS INFORMATIKAI KAR**  
**MÉRÉSTECHNIKA ÉS INFORMÁCIÓS RENDSZEREK TANSZÉK**

# **Digitális technika (VIMIAA01)**

## **A Xilinx ISE 14.6 használata**

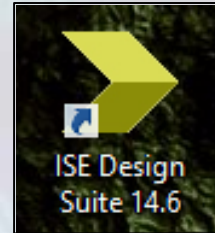
**Fehér Béla**  
**Raikovich Tamás**

**BME MIT**

# Projekt létrehozása (1)

## A Xilinx ISE 14.6 elindítása

- Az asztalon lévő ikonnal
- A Start menüből



All programs →

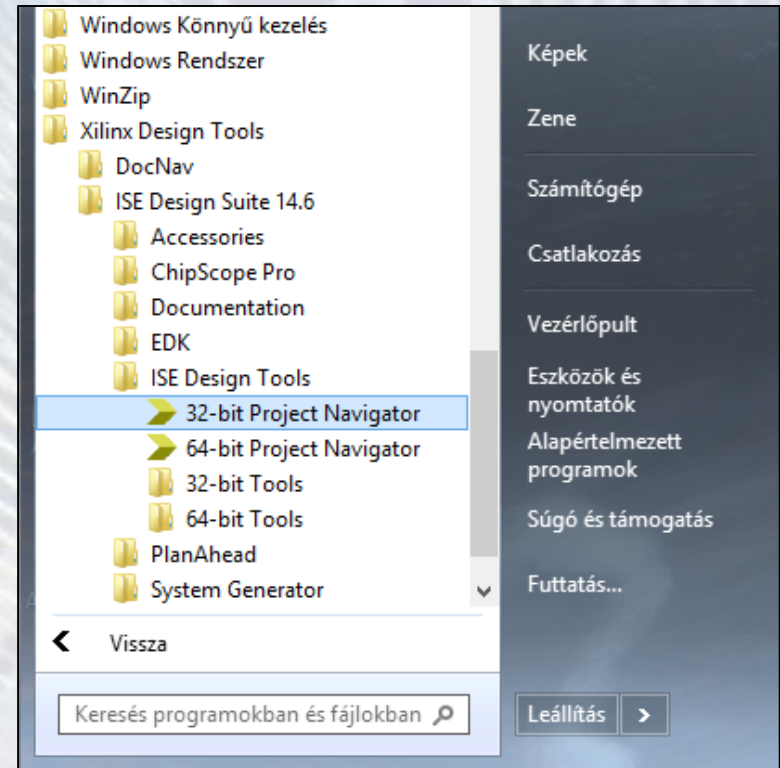
Xilinx Design Tools →

ISE Design Suite 14.6 →

ISE Design Tools →

### ***32-bit Project Navigator***

- Az ISE 64 bites verziója nem működik 64 bites Windows 8 operációs rendszer alatt!



# Projekt létrehozása (2)

- Az új projekt létrehozása  
File → New Project
- Projekt neve: **Lab1**
- Helye: **D:\DTLab\**
- Projekt típusa: **HDL**

New Project Wizard

Create New Project

Specify project location and type.

Enter a name, locations, and comment for the project

Name: Lab1

Location: D:\DTLab\Lab1

Working Directory: D:\DTLab\Lab1

Description:

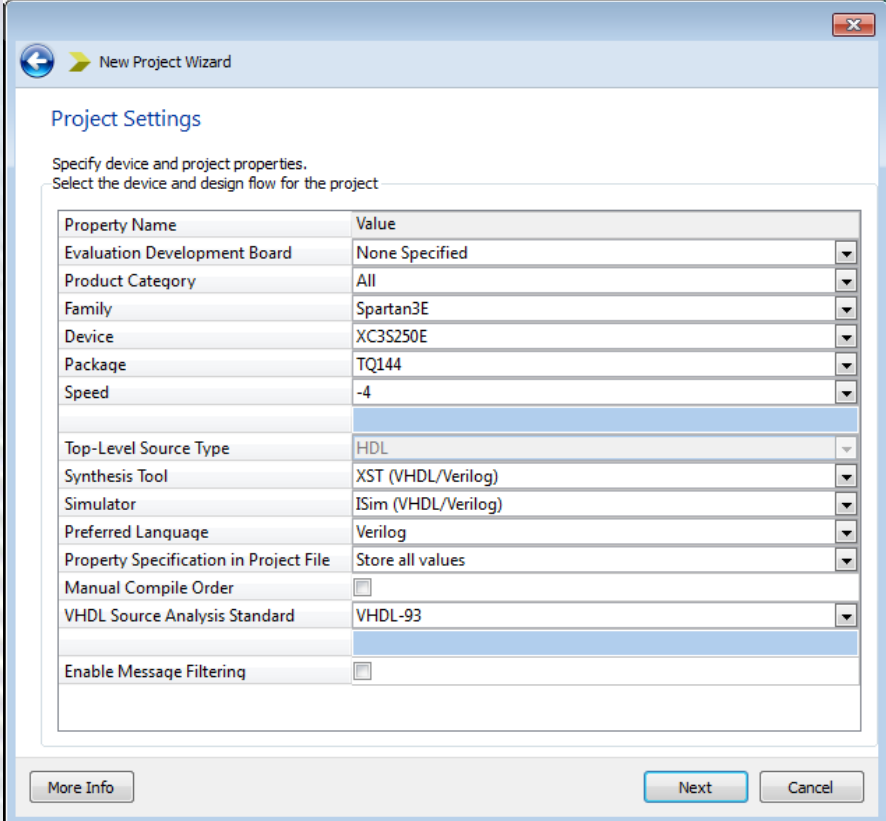
Select the type of top-level source for the project

Top-level source type: HDL

More Info Next Cancel

# Projekt létrehozása (3)

- **Ami fontos:** a fizikai alkatrész specifikálása a LOGSYS Spartan-3E FPGA kártyához
  - Family: **Spartan3E**
  - Device: **XC3S250E**
  - Package: **TQ144**
  - Speed: **-4**
- **A feldolgozási technológia specifikálása**
  - Synthesis tool: **XST**
  - Simulator: **ISim**
  - Preferred Language: **Verilog**



Project Settings

Specify device and project properties.  
Select the device and design flow for the project

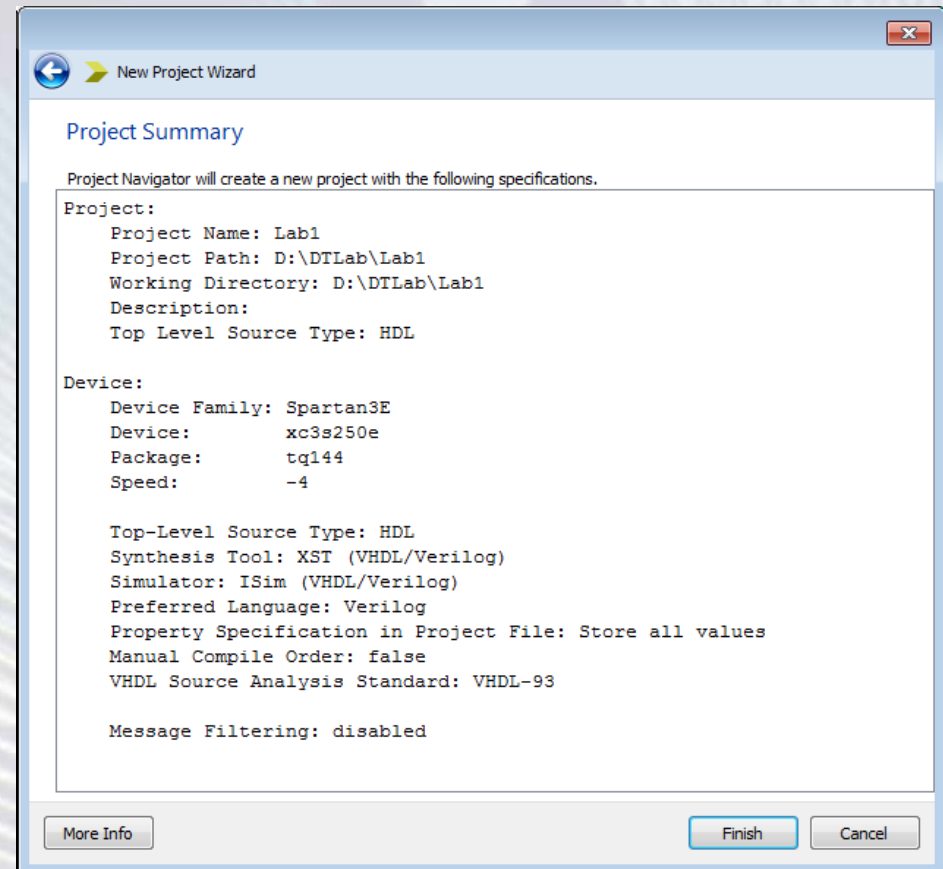
| Property Name                          | Value                    |
|--|--------------------------|
| Evaluation Development Board           | None Specified           |
| Product Category                       | All                      |
| Family                                 | Spartan3E                |
| Device                                 | XC3S250E                 |
| Package                                | TQ144                    |
| Speed                                  | -4                       |
| Top-Level Source Type                  | HDL                      |
| 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"/> |

More Info      Next      Cancel



# Projekt létrehozása (4)

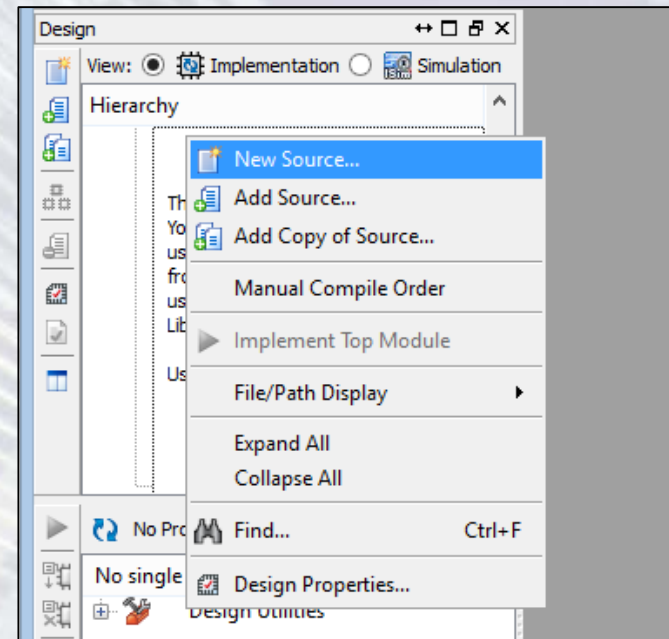
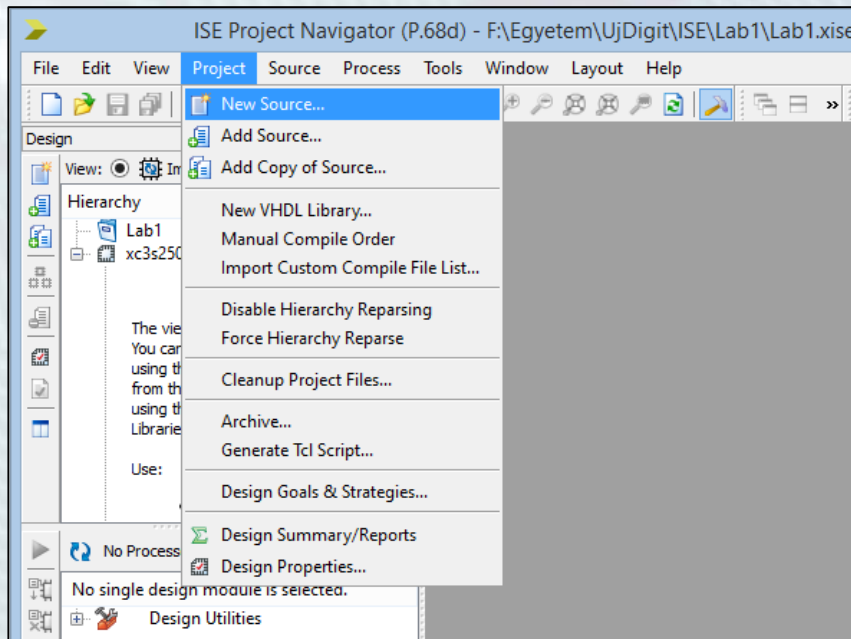
- Összefoglalás a projekt beállításokról
- A Finish után létrejön a projekt struktúra
- Ezután következik a projekt forrásfájlok előkészítése



# Projekt forrásfájlok létrehozása (1)

## A tervet a HDL forrásfájlok specifikálják

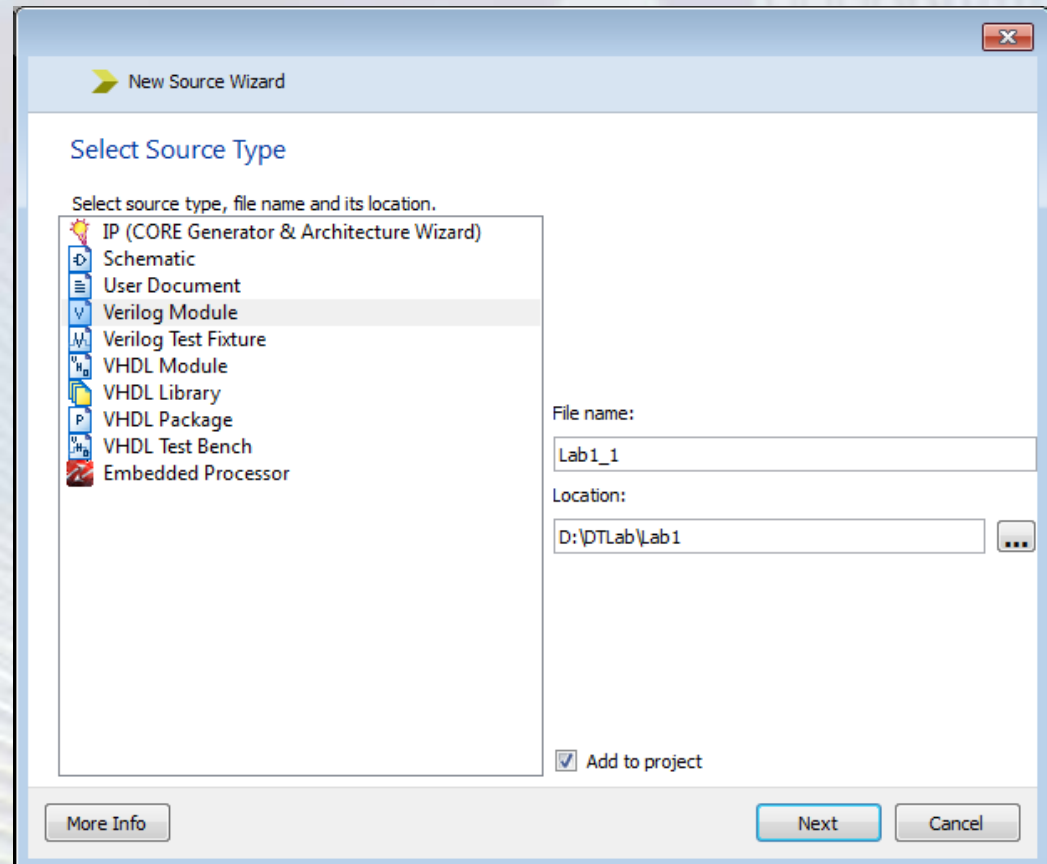
- Project menü → New Source... vagy
- A Design ablakban jobb klikk → New Source...



# Projekt forrásfájlok létrehozása (2)

## Az első forrásfájl típusa

- **Verilog Module**
- Fájl neve: Lab1\_1  
(kiterjesztés: .v)
- Helye a projekt könyvtár
- Az **Add to project** legyen bejelölve



# Projekt forrásfájlok létrehozása (3)

## Definiáljuk a bemeneti és kimeneti jeleket

- Kapcsolók
  - Port neve: sw
  - Irány: input
  - Bus bejelölve
  - MSB: 7, LSB: 0
- LED-ek
  - Port neve: ld
  - Irány: output
  - Bus bejelölve
  - MSB: 7, LSB: 0

New Source Wizard

Define Module

Specify ports for module.


Module name: Lab1\_1

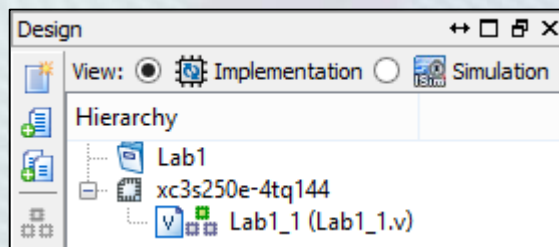
| Port Name | Direction | Bus                                 | MSB | LSB |
|-----------|-----------|-------------------------------------|-----|-----|
| sw        | input     | <input checked="" type="checkbox"/> | 7   | 0   |
| ld        | output    | <input checked="" type="checkbox"/> | 7   | 0   |
|           | 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 Next Cancel

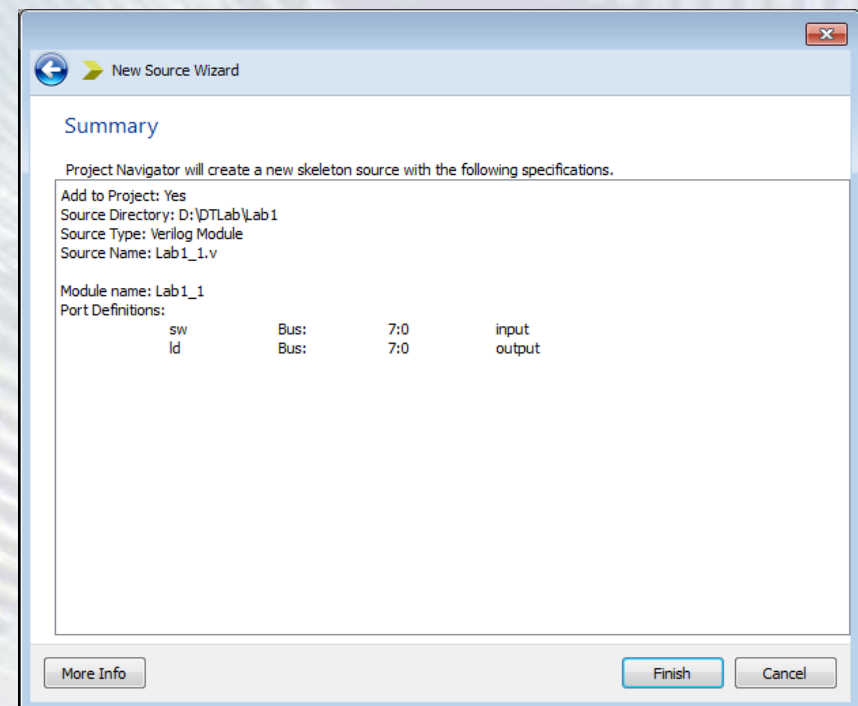


# Projekt forrásfájlok létrehozása (4)

- Összefoglalás a beállításokról
- A Finish után létrejön a Lab1\_1.v Verilog HDL forrásfájl
- A fájl melletti  ikon jelzi, hogy ez a projekt hierarchia csúcsán lévő „top module” forrásfájl



- Minden más projektfájl ez alá fog rendeződni



# Projekt forrásfájlok létrehozása (5)

## A Lab1\_1.v Verilog minta szövegfájl tartalma

- ``timescale 1ns/1ps`: a szimuláció során az időfelbontás 1 ps, az értékek pedig ns-ban értendők (pl. 1,23456789 us = 1234,568 ns)
- A megjegyzés mező nem lényeges
- A Verilog fájl `module...endmodule` törzse tartalmazza az általunk megadott `input` és `output` paramétereket, továbbá ide kerül majd a forráskód lényeges, a működést specifikáló része is (lásd később)

```
`timescale 1ns / 1ps
////////////////////////////////////
// Company:
// Engineer:
//
// Create Date:      23:5
// Design Name:
// Module Name:      Lab1
// Project Name:
// Target Devices:
// Tool versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File
// Additional Comments:
//
////////////////////////////////////
module Lab1_1(
    input [7:0] sw,
    output [7:0] ld
);

endmodule
```

# Projekt forrásfájlok létrehozása (6)

- **Összetett feladatok esetén a terv részleteit önálló modulokba érdemes elhelyezni**
- **A modulok legyenek önálló fájlok**
  - A partícionálás finomsága (az egyes modulok komplexitása) egyéni döntés kérdése
  - A lényeg, hogy segítse a terv megértését
- **Sok esetben használhatunk már meglévő, könyvtári modulokat. Ezek vagy csak olvashatók, vagy ha nem, akkor csak a másolatukat adjuk hozzá a projekthez.**
- **Léteznek paraméterezhető modulok is (lásd később)**



# Projekt forrásfájlok létrehozása (7)

## A projekt források elsősorban

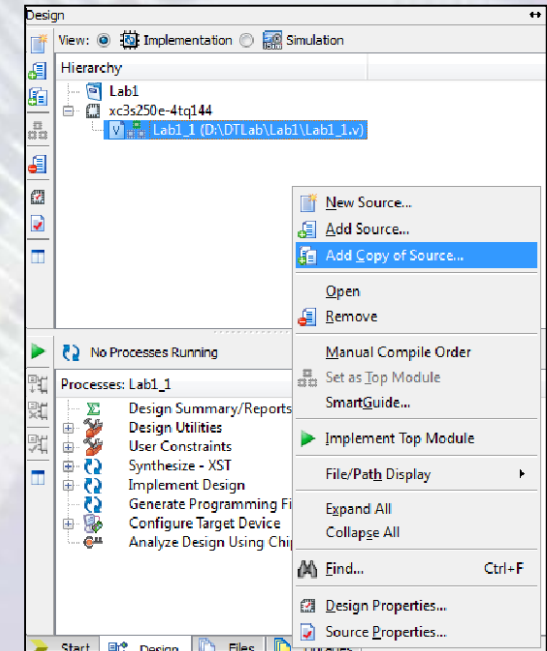
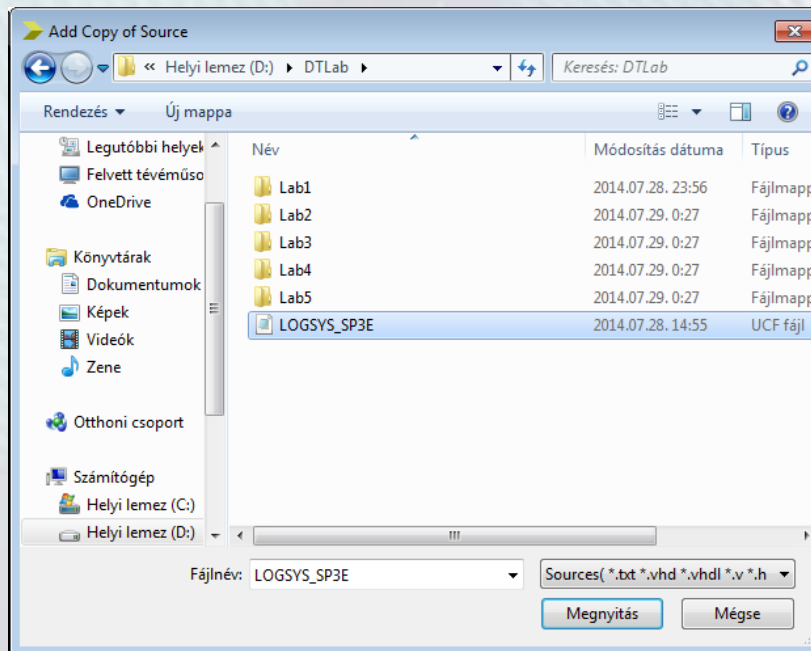
- HDL nyelvű tervezői fájlok (Verilog vagy VHDL)
  - Funkcionális szinten keverhető a kétféle specifikáció
- Közbenső technológiai szintű leírások
- Felhasználói tervezői előírások
  - Az eszköz kivezetéseinek használata (I, O, I/O) és kiosztása
  - Az interfészek speciális tulajdonságai (jelszint, áram, jelváltási sebesség, stb.)
  - Előírt globális rendszerműködési sebesség (órajel)
  - Optimalizálási előírások a belső áramköri részletekre



# Projekt forrásfájlok létrehozása (8)

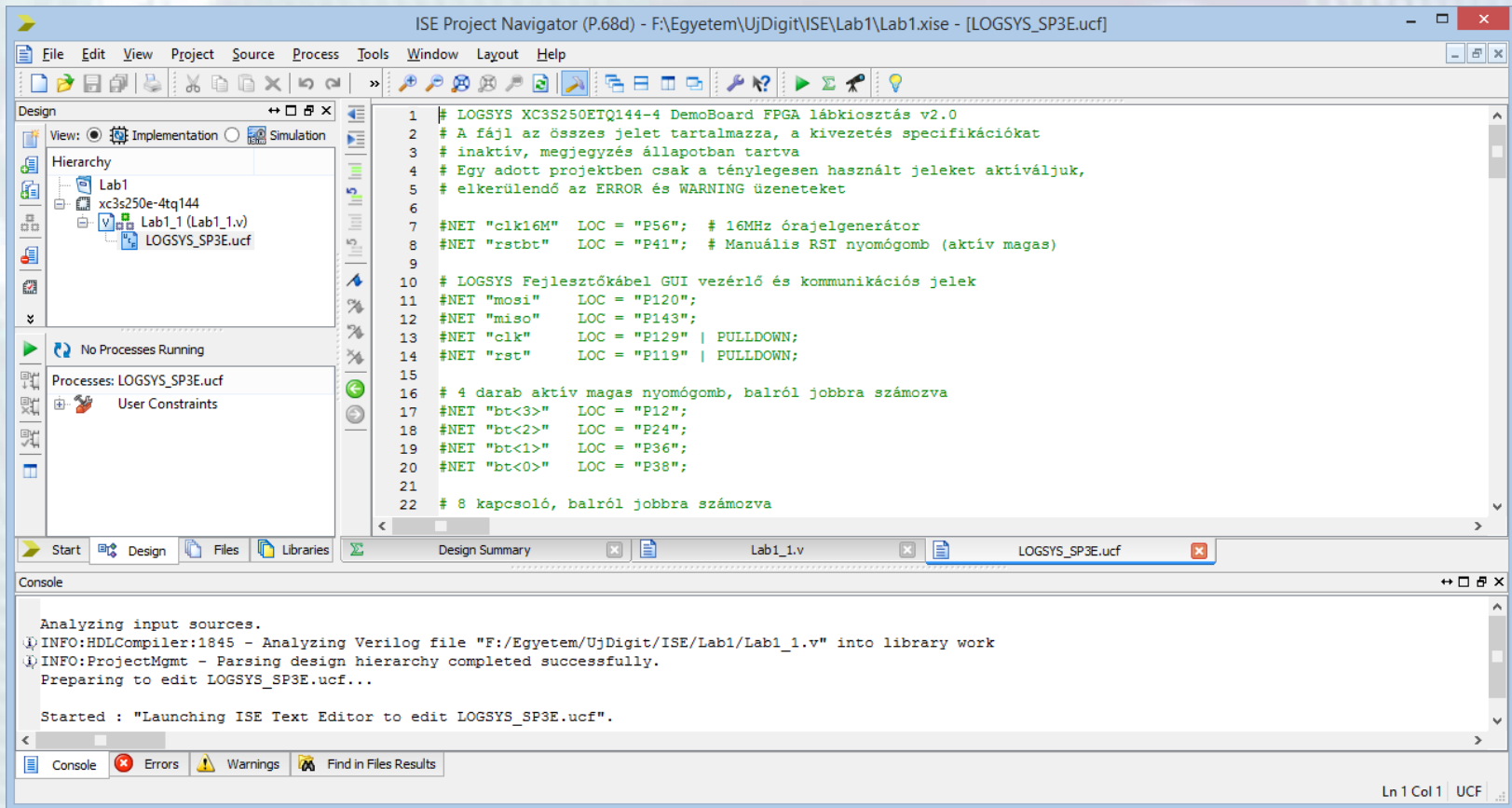
A külső jelek bekötésének megadása (UCF fájl) – ez a LOGSYS Spartan-3E FPGA kártyához előre el lett készítve

- Project menü → Add Copy of Source... vagy
- A Design ablakban jobb klikk → Add Copy of Source...



# Projekt forrásfájlok létrehozása (9)

## A projekt struktúra elkészült



The screenshot displays the Xilinx ISE Project Navigator interface. The main window shows the Verilog source code for the project, which is a demo board FPGA lab kit. The code includes comments in Hungarian and defines several nets for the design.

```
1 # LOGSYS XC3S250ETQ144-4 DemoBoard FPGA lábkiosztás v2.0
2 # A fájl az összes jelet tartalmazza, a kivezetés specifikációkat
3 # inaktív, megjegyzés állapotban tartva
4 # Egy adott projektben csak a ténylegesen használt jeleket aktiváljuk,
5 # elkerülendő az ERROR és WARNING üzeneteket
6
7 #NET "clk16M" LOC = "P56"; # 16MHz órajelgenerátor
8 #NET "rstbt" LOC = "P41"; # Manuális RST nyomógomb (aktív magas)
9
10 # LOGSYS Fejlesztőkábel GUI vezérlő és kommunikációs jelek
11 #NET "mosi" LOC = "P120";
12 #NET "miso" LOC = "P143";
13 #NET "clk" LOC = "P129" | PULLDOWN;
14 #NET "rst" LOC = "P119" | PULLDOWN;
15
16 # 4 darab aktív magas nyomógomb, balról jobbra számozva
17 #NET "bt<3>" LOC = "P12";
18 #NET "bt<2>" LOC = "P24";
19 #NET "bt<1>" LOC = "P36";
20 #NET "bt<0>" LOC = "P38";
21
22 # 8 kapcsoló, balról jobbra számozva
```

The left sidebar shows the project hierarchy with the following structure:

- Lab1
  - xc3s250e-4tq144
    - Lab1\_1 (Lab1\_1.v)
    - LOGSYS\_SP3E.ucf

The bottom console window shows the following output:

```
Analyzing input sources.
INFO:HDLCompiler:1845 - Analyzing Verilog file "F:/Egyetem/UjDigit/ISE/Lab1/Lab1_1.v" into library work
INFO:ProjectMgmt - Parsing design hierarchy completed successfully.
Preparing to edit LOGSYS_SP3E.ucf...

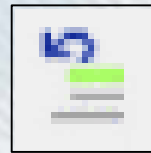
Started : "Launching ISE Text Editor to edit LOGSYS_SP3E.ucf".
```

# Projekt forrásfájlok létrehozása (10)

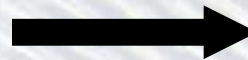
## A külső jelek bekötésének specifikálása

- A LOGSYS\_SP3E.ucf fájlt a projekt könyvtárába másoltuk és a használandó jeleket aktiváljuk (a projektre szabjuk)

```
19 #NET "bt<1>" LOC = "P36";
20 #NET "bt<0>" LOC = "P38";
21
22 # 8 kapcsoló, balról jobbra számozva
23 #NET "sw<7>" LOC = "P47";
24 #NET "sw<6>" LOC = "P48";
25 #NET "sw<5>" LOC = "P69";
26 #NET "sw<4>" LOC = "P78";
27 #NET "sw<3>" LOC = "P84";
28 #NET "sw<2>" LOC = "P89";
29 #NET "sw<1>" LOC = "P95";
30 #NET "sw<0>" LOC = "P101";
31
32 # 8 LED, balról jobbra számozva
33 #NET "ld<7>" LOC = "P43";
34 #NET "ld<6>" LOC = "P50";
35 #NET "ld<5>" LOC = "P51";
36 #NET "ld<4>" LOC = "P52";
37 #NET "ld<3>" LOC = "P53";
38 #NET "ld<2>" LOC = "P54";
39 #NET "ld<1>" LOC = "P58";
40 #NET "ld<0>" LOC = "P59";
41
42 # 4 digités kijelző aktív ALACSONY szegmens vez
43 # Ugyanezeket a lábakat használjuk a 7x5 matrix
```



Uncomment



```
19 #NET "bt<1>" LOC = "P36";
20 #NET "bt<0>" LOC = "P38";
21
22 # 8 kapcsoló, balról jobbra számozva
23 NET "sw<7>" LOC = "P47";
24 NET "sw<6>" LOC = "P48";
25 NET "sw<5>" LOC = "P69";
26 NET "sw<4>" LOC = "P78";
27 NET "sw<3>" LOC = "P84";
28 NET "sw<2>" LOC = "P89";
29 NET "sw<1>" LOC = "P95";
30 NET "sw<0>" LOC = "P101";
31
32 # 8 LED, balról jobbra számozva
33 NET "ld<7>" LOC = "P43";
34 NET "ld<6>" LOC = "P50";
35 NET "ld<5>" LOC = "P51";
36 NET "ld<4>" LOC = "P52";
37 NET "ld<3>" LOC = "P53";
38 NET "ld<2>" LOC = "P54";
39 NET "ld<1>" LOC = "P58";
40 NET "ld<0>" LOC = "P59";
41
42 # 4 digités kijelző aktív ALACSONY szegme
43 # Ugyanezeket a lábakat használjuk a 7x5
```

# Projekt terv megvalósítása

- **A tervezési feladat alapján megírjuk a „top module” és az esetleges egyéb modulok funkcionalitását realizáló kódrészleteket**
  - Lásd pl. a Lab1\_1a, Lab1\_1b és Lab1\_1c feladatokat
- **Mentés, szintaktikai ellenőrzések, javítások**
- **Ha minden rendben, akkor**
  - Az elvi (funkcionális) terv ellenőrzése szimulációval
  - A terv technológiai feldolgozása
  - A generált konfiguráció letöltése és tesztelés a működő hardveren (LOGSYS GUI)



# Példa: Lab1\_1 tervezési feladatok

A létrehozott Lab1\_1.v Verilog HDL modul üres vázába készítjük el az első tervspecifikációt

```
`timescale 1ns / 1ps
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// Digitális Technika Laboratórium 1. hét
// 1. bemutató projekt: A környezet használatának bemutatása
// LED-ek vezérlése DIP kapcsolókkal
// Részfeladatok:
// 1_1_1   Egyszerű vezetékezés, műveletvégzés nélkül, 8 bites vektor jelekkel
// 1_1_2   Kettes komplement képzés
// 1_1_3   Aritmetikai műveletek vizsgálata (+, *, /, %, **) 4 bites operandusokon
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
module Lab1_1(
    input   [7:0] sw,
    output  [7:0] ld
);

// IDE ÍRANDÓ A VERILOG KÓD
// ....
// ....

endmodule
```

# Lab1\_1a feladat

## Egyszerű jelátvezetés a kapcsolókról a LED-ekre

- Kezelhető vektorosan vagy egyedi bitenként
- $LD[i] = SW[i]$  esetén a vektoros egyszerűbb
- Más leképezéseknél (pl.  $i \neq j$  esetén) többet kell gépelni

```
module Lab1_1(  
    input    [7:0] sw,  
    output   [7:0] ld  
);  
  
////////////////////////////////////  
// 1_1_1 A kimeneti LED-ek vezérlése közvetlenül a kapcsolókkal  
// LD[i] = SW[i] , ez az áramkörben egy direkt vezérlés, a Verilog HDL nyelvben  
// folytonos értékadásnak nevezzük  
////////////////////////////////////  
  
//?????????????  
  
endmodule
```

# Lab1\_1a feladat

## Egyszerű jelátvezetés a kapcsolókról a LED-ekre

- Ez egy folytonos vezérlés SW → LED bitjei között
- Egy egyszerűsített modellje a HW működésének

```
module Lab1_1(  
    input    [7:0] sw,  
    output   [7:0] ld  
);  
  
/////////////////////////////////////  
// 1_1_1 A kimeneti LED-ek vezérlése közvetlenül a kapcsolókkal  
// LD[i] = SW[i] , ez az áramkörben egy direkt vezérlés, a Verilog HDL nyelvben  
// folytonos értékadásnak nevezzük  
/////////////////////////////////////  
  
assign ld = sw;                // Vektoros folytonos értékadás  
  
endmodule
```

# A terv ellenőrzése szimulációval

- A szimulátor egy számítógépes program, amely a digitális tervleírás funkcionális működését ellenőrzi
- Funkcionális szimulációnál csak a modulok bemeneti – kimeneti összefüggéseit szimulálja, a valós fizikai hatásokat (időzítés, meghajtáserősség) nem kezeli
- A szimulátor egy tesztkörnyezetet ad, amelyben a bemeneteket jelforrásokkal, „generátorokkal” vezéreljük és vizsgáljuk, „monitorozzuk” a kimenetek állapotát.
- A terv technológiai feldolgozása után akár teljes időzítési szimuláció is végezhető (nem része a jelen labornak)



# A szimulációs környezet (1)

- **A Xilinx ISE beépített szimulátora az ISim**
  - Verilog és VHDL tervek szimulációját biztosítja
  - Gyári alkatrészmodell könyvtárakat használ
- **A tesztkörnyezet neve Verilog Test Fixture**
  - Speciális modul, nincsenek bemenetei/kimenetei, azaz minden „benne” van, ami a szimulációhoz kell
  - Persze, szinte mindent nekünk kell beletenni, mert kezdetben csak egy üres mintakeret
  - Specifikáljuk a meghajtó jeleket, tesztvektorokat, gondoskodunk az alaphelyzetbe állításról és a kimeneti eredmények értékelési módjáról

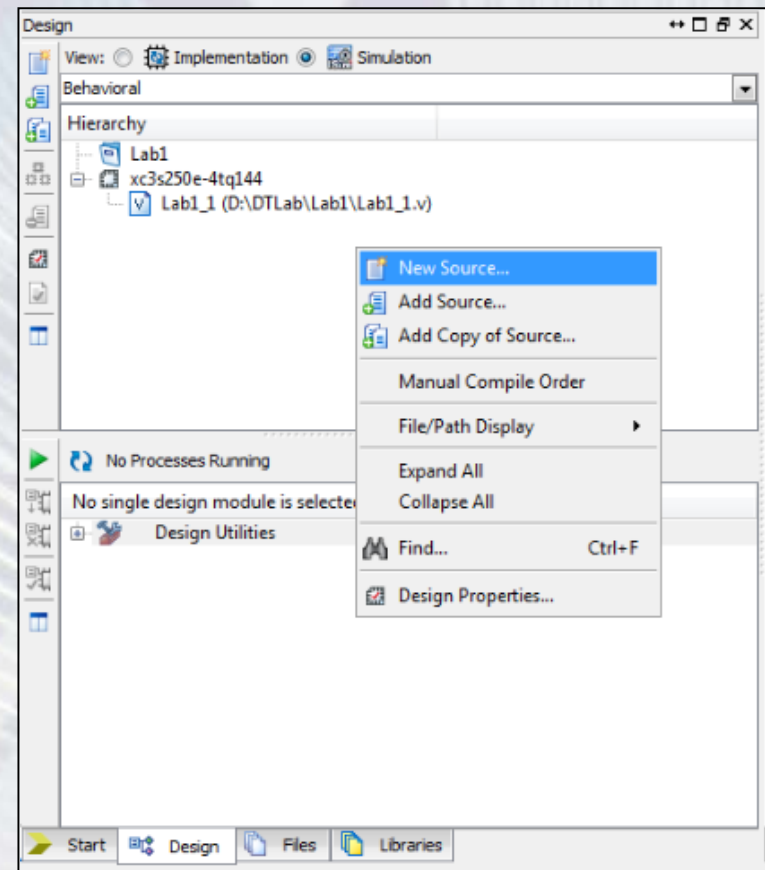
# A szimulációs környezet (2)

## A tesztkörnyezet hierarchia felépítése

- Szimulációs környezet = Verilog Test Fixture
- A beágyazott Verilog module, a tesztelt tervfájl neve, azonosítója UUT („Unit Under Test”)
  - Ez egy általános azonosító a tesztkörnyezetbe beillesztett tetszőleges tervezési fájlra (pl. Lab1\_1.v)
- **Megjegyzés:** A szimulációs környezet felépítése általában összetettebb munka, mint a tervfájl elkészítése. Sokan nem is szeretik ezt a feladatot.
- **De** ellenőrzés nélkül a terv ritkán működik helyesen! Ez jellemzően minden területen igaz!

# A szimuláció előkészítése (1)

- A projekt nézetet átváltjuk szimulációs módba
- **View: Implementation → Simulation**
  - Hatás: UCF nem látható
  - Alul Process ablak „kiürül”
  - Csak a legfelső szintű (jelen esetben egyetlen) Lab1\_1.v tervfájl marad látható
  - Ehhez rendeljük hozzá új forrásként a teszt-környezetet (az ismert módon...)

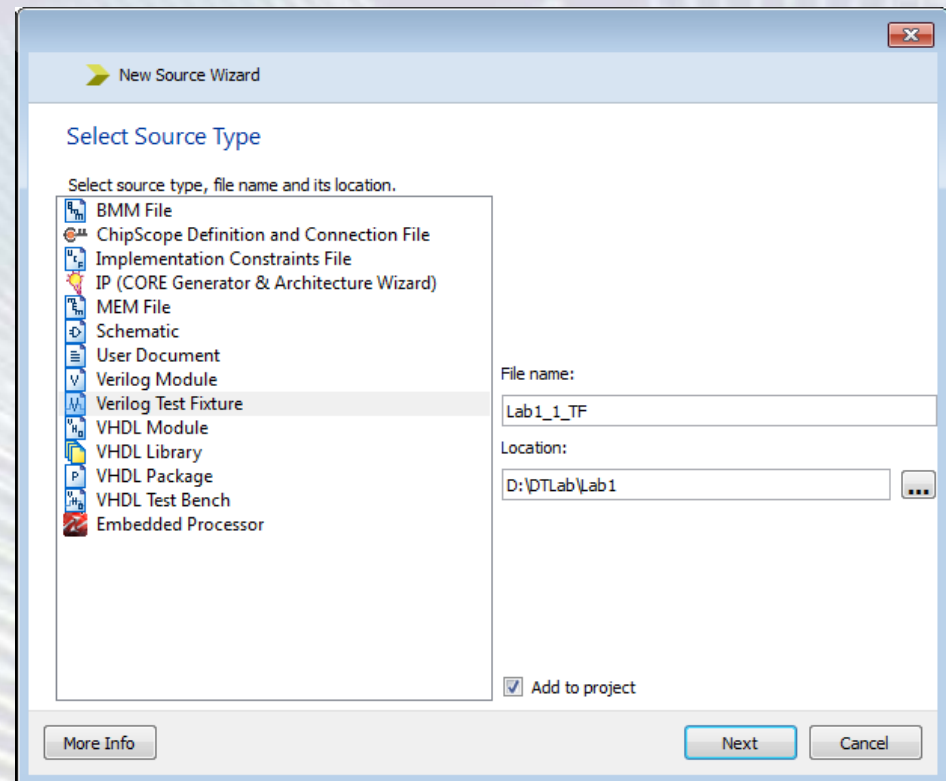




# A szimuláció előkészítése (2)

## Az új forrás típusa Verilog Test Fixture

- Neve legyen a tervhez kapcsolódó, \_TF kiegészítéssel (pl. Lab1\_1\_TF.v)
- Ez is egy Verilog HDL fájl, csak a használata speciális
- Csak az ellenőrzés során használjuk

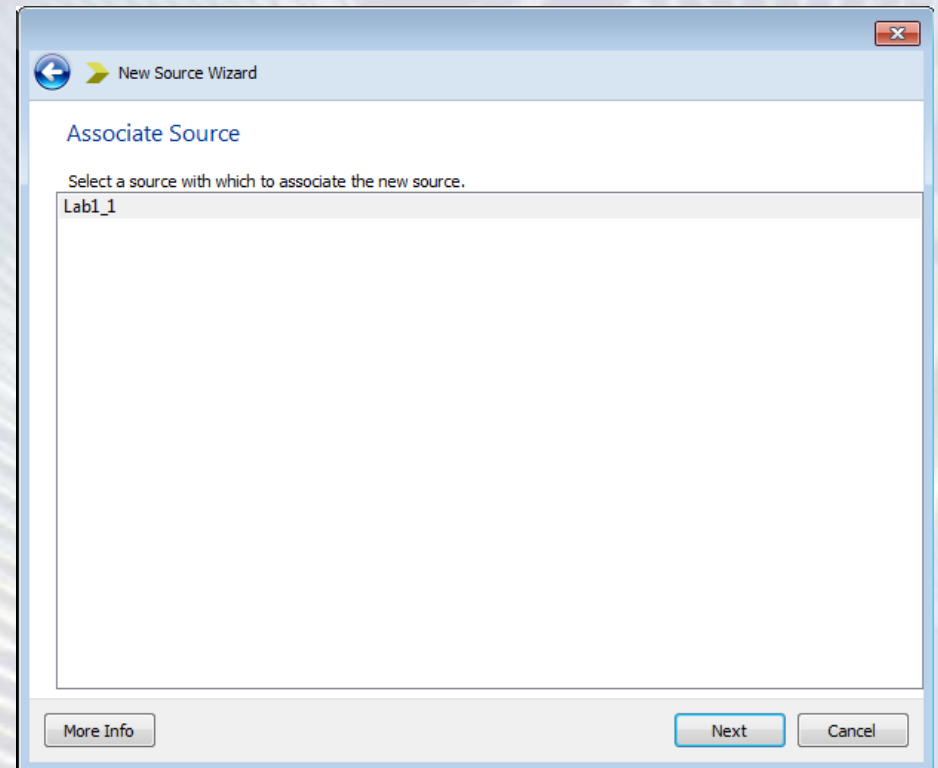




# A szimuláció előkészítése (3)

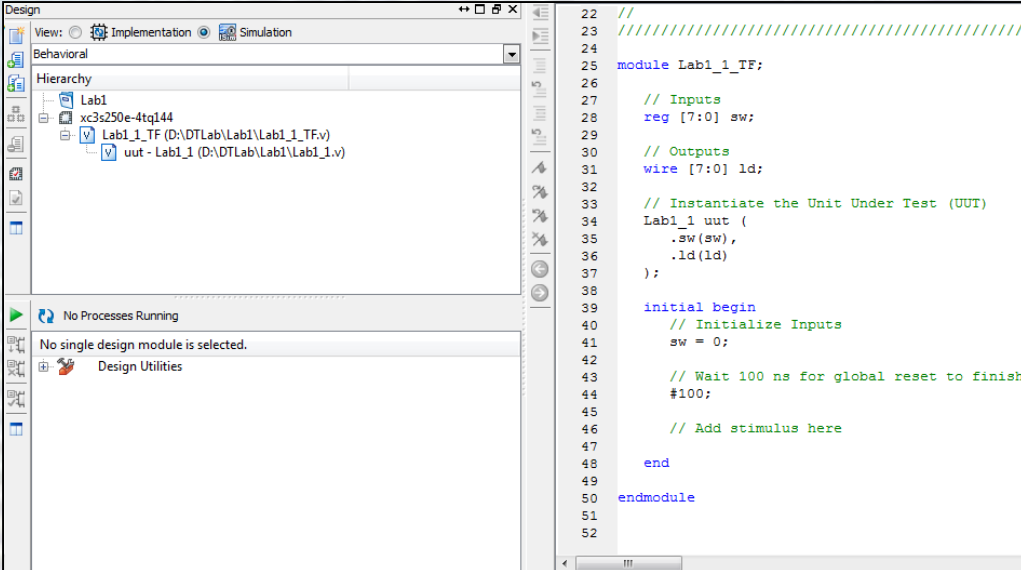
A Verilog Test Fixture mintafájl generálása a tesztelendő fájl (UUT) interfészjellemzőitől függ, ezért meg kell adni a tesztelendő fájlt.

- A példában csak egy fájl van, ezért ez egyértelmű
- Összetettebb projektnél kiválasztható, mi legyen a teszt célpontja



# A szimuláció előkészítése (4)

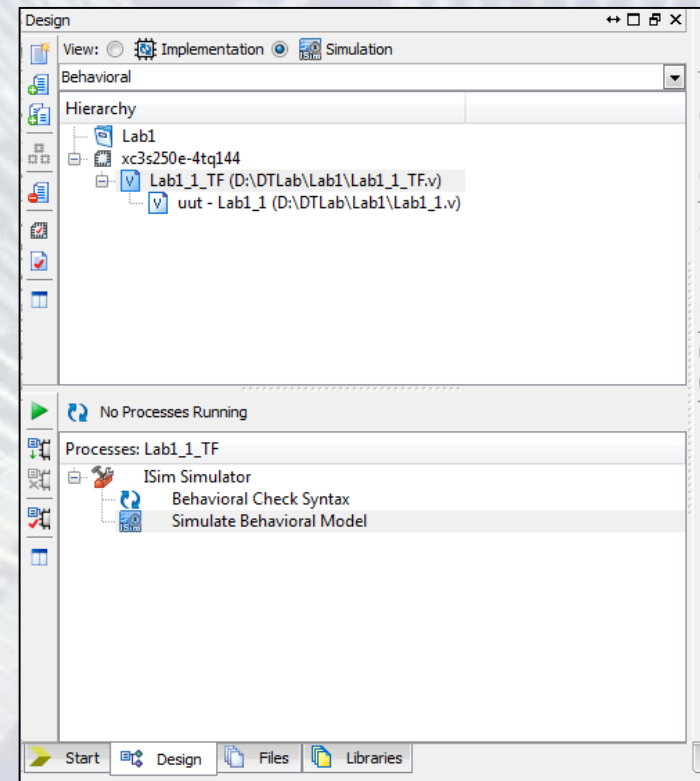
- A létrehozott Lab1\_1\_TF a szimulációs üzemmód beágyazó környezetét biztosítja, beköti a jeleket
- Vezérli a bemeneteket, monitorozza a kimeneteket
- Beépíti a Lab1\_1.v tervet, mint UUT
- Inicializálja a változókat és futtat 100ns szimulációt
- Ezután várja a saját tesztvektorok, teszt-előírások megadását az **initial begin...end** blokkon belül



```
22 //
23 ///////////////////////////////////////////////////////////////////
24
25 module Lab1_1_TF;
26
27 // Inputs
28 reg [7:0] sw;
29
30 // Outputs
31 wire [7:0] ld;
32
33 // Instantiate the Unit Under Test (UUT)
34 Lab1_1 uut (
35     .sw(sw),
36     .ld(ld)
37 );
38
39 initial begin
40     // Initialize Inputs
41     sw = 0;
42
43     // Wait 100 ns for global reset to finish
44     #100;
45
46     // Add stimulus here
47
48 end
49
50 endmodule
51
52
```

# A szimuláció indítása

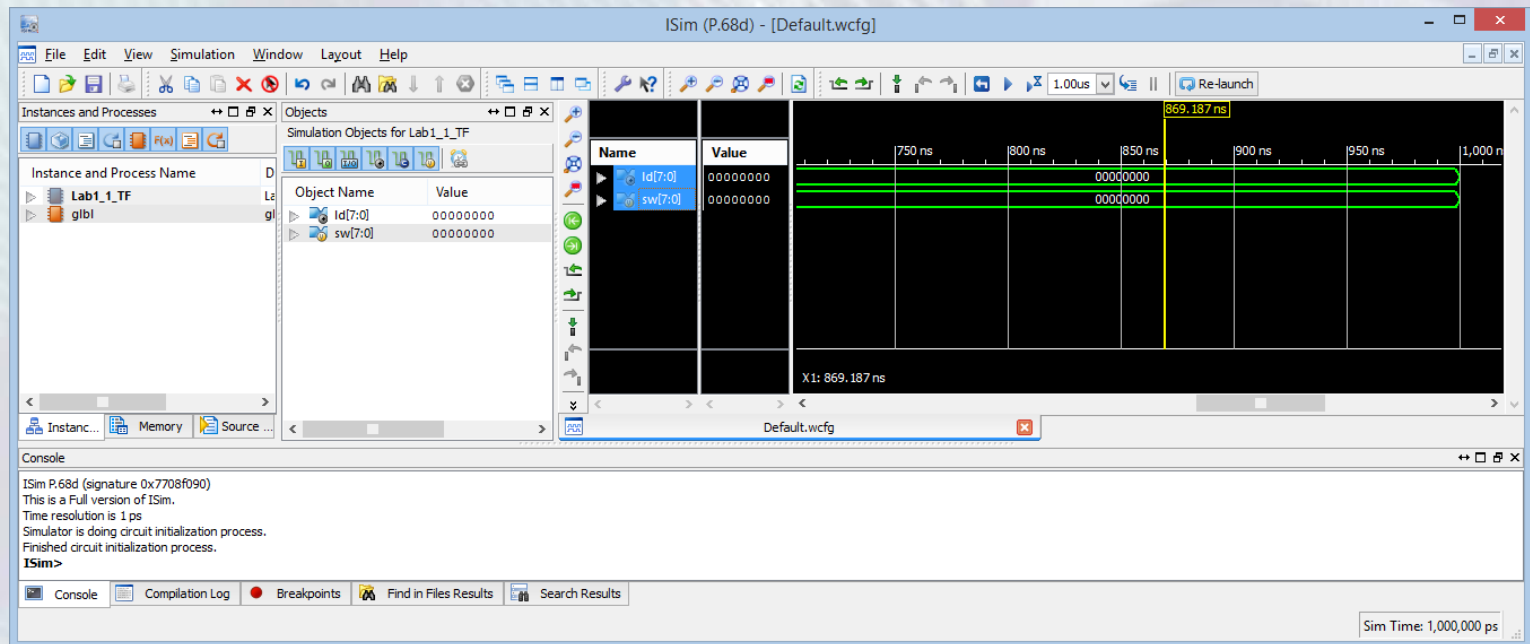
- A szimuláció indítása a Lab1\_1\_TF.v Verilog Test Fixture fájl kiválasztásával és a Process ablakban a Simulate Behavioral Model parancs kiadásával lehetséges
- Ekkor egy önálló program, az ISim szimulátor indul el
- Ez beolvassa a forrásokat és szintaktikai ellenőrzés után a szimulációs projekt futtatható modelljét generálja, amit az ISim GUI-ban tesztelhetünk



# A Xilinx ISim szimulátor

## A szimulátor egy összetett programkörnyezet

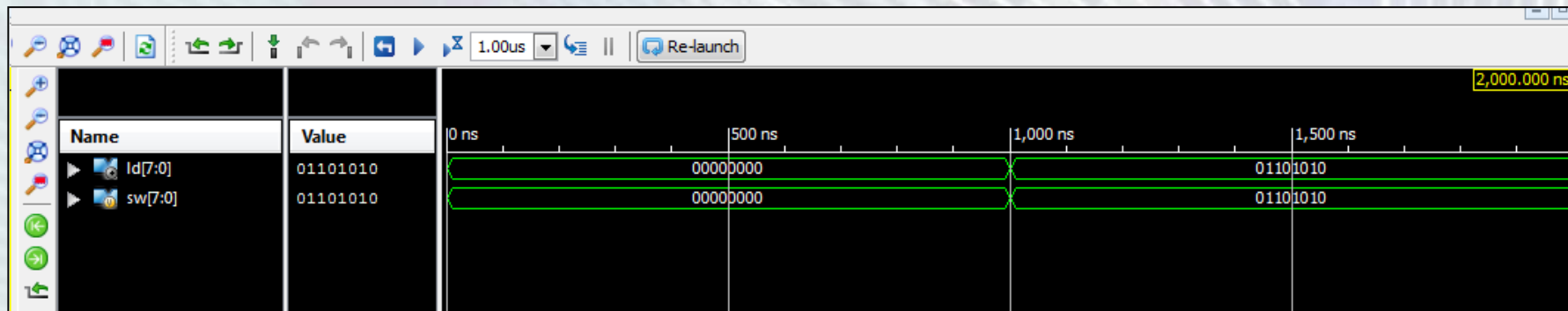
- Legfontosabb részlete a hullámforma ablak, ahol az idő függvényében látjuk, hogy a bemeneti vezérlésre hogyan reagál a tesztelt UUT.
- Általában programozott tesztekkel dolgozunk
- De ehhez kell a Verilog nyelv ismerete...





# Az ISim interaktív használata (1)

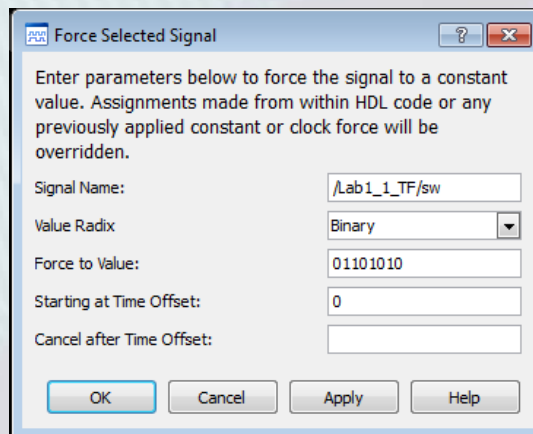
- Nagyon egyszerű esetekben használható interaktív módban
  - Új bemeneti érték előírása → Force Constant...
  - Szimuláció adott ideig → Run
  - Ellenőrzés a válasz alapján → A Waveform ablakban
- Például SW = 01101010 bemenetre 1 us futás után

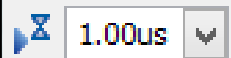



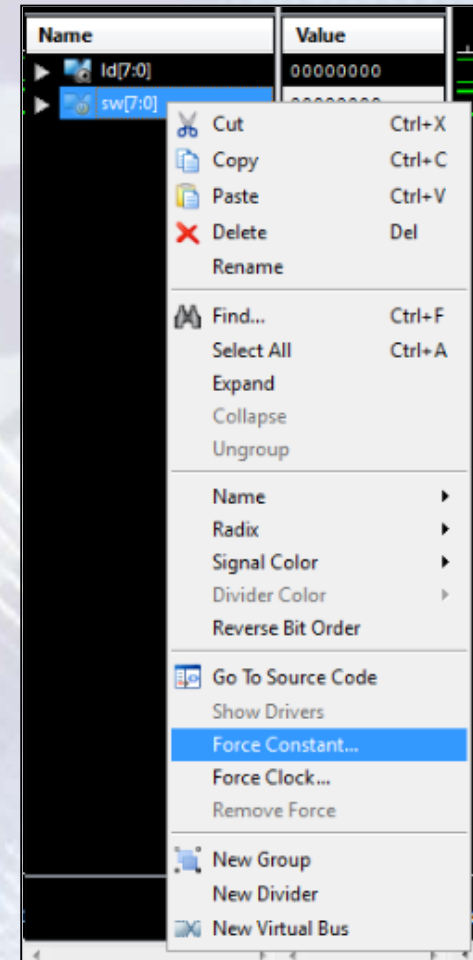
# Az ISim interaktív használata (2)

## A bemeneti SW jel vezérlése

- Jobb klikk az SW jelen → Force Constant... → OK

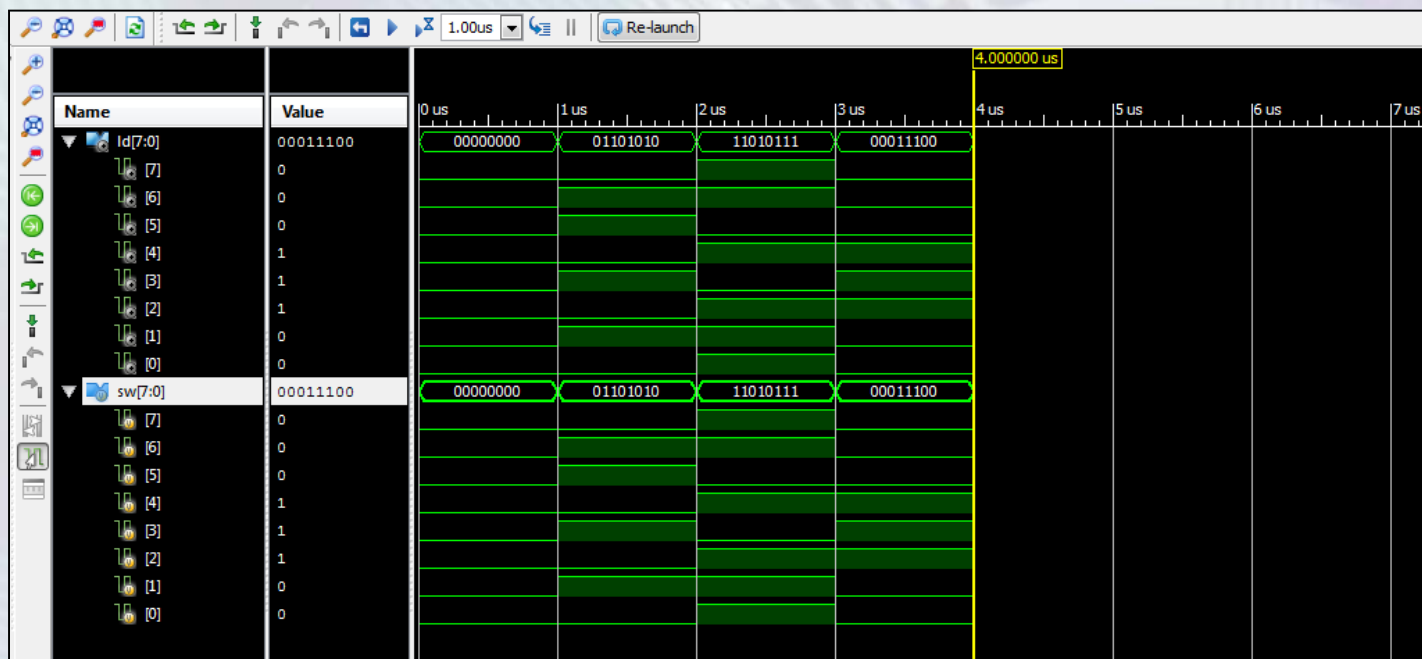


- Szimuláció adott ideig
- A  1.00us menüsor gomb 1us ideig futtatja a szimulációt
- Ezután a  gombokkal a hullámforma analizálható



# Az ISim interaktív használata (3)

- Egyszerű tervünk különböző tesztvektorokkal ellenőrizhető, a jelek vektorosan, vagy bitenként is vizsgálhatóak
- A megjelenítési opciók a felugró menüben megtekinthetők (név, szín, számábrázolás, bitsorrend, vezérlés)
- Ha a tesztelés kész (mikor kész?), akkor kiléphetünk az ISim-ből



# Az ISim hatékonyabb használata

- A HDL alapú tesztelés a programozott tesztvektor generálással aknázható ki igazán
  - A fáradtságos, nem hatékony Force parancsok helyett a Test Fixture fájlban előírjuk a teszt időbeli lefutását
  - Az `initial begin...end` egyfajta ütemezett lefutást biztosít, azaz az előírt időzítéssel (`#100`) kiadja az új tesztvektorokat, majd leáll
- A Verilog Test Fixture fájl módosítása miatt csak újrafordítás után használható (Re-launch)

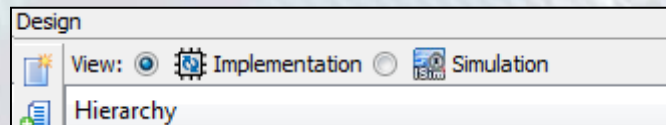
```
`timescale 1ns / 1ps
module Lab1_1_TF;
    // Inputs
    reg [7:0] sw;
    // Outputs
    wire [7:0] ld;
    // Instantiate the Unit Under Test (UUT)
    Lab1_1 uut (
        .sw(sw),
        .ld(ld)
    );
    initial begin
        // Initialize Inputs
        sw = 0;
        // Wait 100 ns for global reset to finish
        #100;
        // Add stimulus here
        #100 sw = 8'b01010101;
        #100 sw = 8'b01111000;
        #100 sw = 8'b11001100;
        #100 sw = 8'b10001011;
    end
endmodule
```



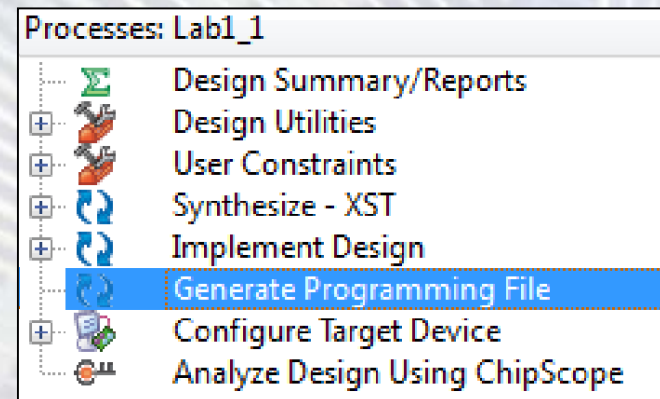
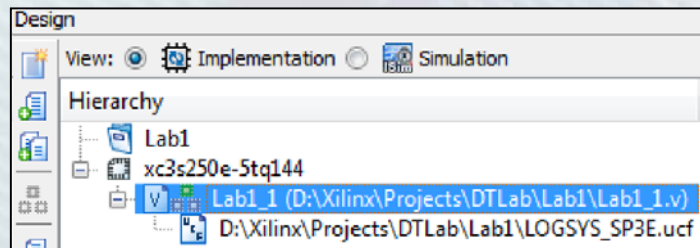
# 1.1.a feladat

A szimuláció befejezése után generáljuk a specifikációhoz tartozó programozó adatfájlt

- Implementációs mód (NEM SZIMULÁCIÓS!)



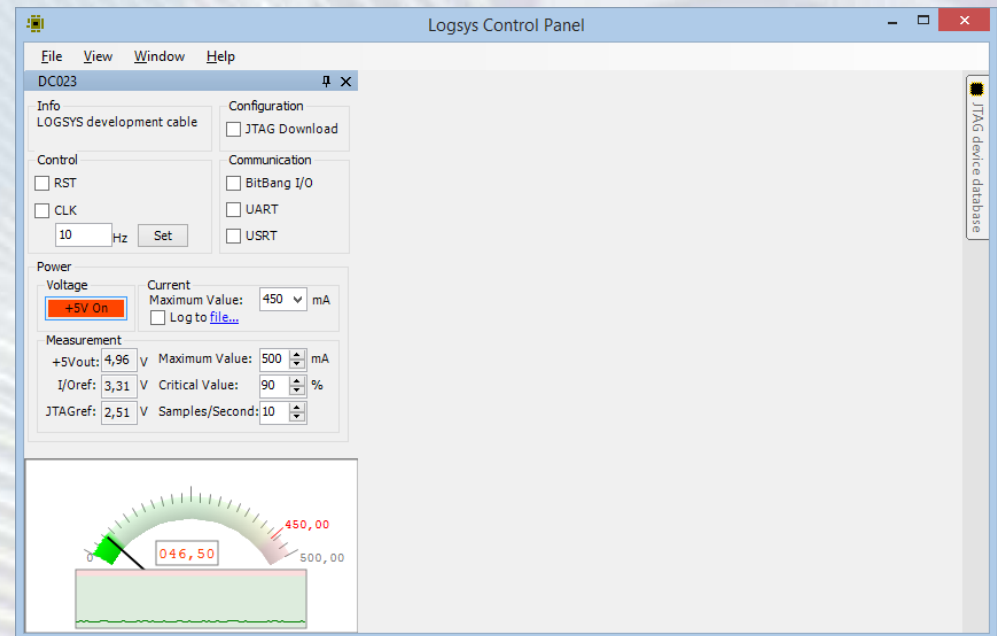
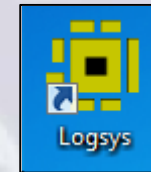
- A Lab1\_1.v tervezői fájl aktív (és látszik az UCF is)
- Konfigurációs fájl generálás



# Az FPGA felkonfigurálása (1)

- Csatlakoztassuk a fejlesztői kábelt a számítógéphez és az FPGA kártyához
- Indítsuk el a LOGSYS GUI alkalmazást
- Kapcsoljuk be a +5V tápfeszültséget, ellenőrizzük a mért adatokat:

- Vout: 5 V
- Vref I/O: 3,3 V
- Vref JTAG: 2,5 V
- Iout: kb. 50 mA



# Az FPGA felkonfigurálása (2)

- Nyissuk meg a JTAG Download konfigurációs interfészt
- Azonosítsuk az elérhető eszközöket → Query JTAG chain
- Válasszuk ki az FPGA-t a listából (csak egy eszköz van)
- Töltsük le az FPGA-ra a projekt könyvtárból a lab1\_1.bit fájlt (Configure the selected device...)
- A letöltés végén az FPGA kártyán a zöld színű DONE LED kigyulladás jelzi a sikeres konfigurálást
- Teszteljük a működést!

