

**Dr. Vörösházi Zsolt**

[voroshazi.zsolt@virt.uni-pannon.hu](mailto:voroshazi.zsolt@virt.uni-pannon.hu)

## Tervezési módszerek programozható logikai eszközökkel

**3. Xilinx Vivado Design Suite integrált fejlesztő rendszer használata.  
Vivado integrált szimulátor használata**

# Tárgyalt ismeretkörök

## 3. előadás

- I.) Xilinx Vivado fejlesztő környezet,
- II.) Xilinx Vivado Simulator - integrált szimulátor használata.



Rövid általános ismertetés

# **I. XILINX VIVADO DESIGN SUITE (2020.1)**



# Felhasznált irodalom:

-  Xilinx Vivado Design Suite:
  - <http://www.xilinx.com/products/design-tools/vivado.html>
-  Vivado Design Suite User Guide (UG910):
  - [https://www.xilinx.com/support/documentation/sw\\_manuals/xilinx2018\\_3/ug910-vivado-getting-started.pdf](https://www.xilinx.com/support/documentation/sw_manuals/xilinx2018_3/ug910-vivado-getting-started.pdf)
-  Xilinx Vivado DS design flows (UG892):
  - [https://www.xilinx.com/support/documentation/sw\\_manuals/xilinx2019\\_2/ug892-vivado-design-flows-overview.pdf](https://www.xilinx.com/support/documentation/sw_manuals/xilinx2019_2/ug892-vivado-design-flows-overview.pdf)
-  Xilinx Vivado simple tutorial (Nexys-4):
  - [http://www.xilinx.com/support/documentation/university/Vivado-Teaching/HDL-Design/2013x/Nexys4/Verilog/docs-pdf/Vivado\\_tutorial.pdf](http://www.xilinx.com/support/documentation/university/Vivado-Teaching/HDL-Design/2013x/Nexys4/Verilog/docs-pdf/Vivado_tutorial.pdf)
-   Xilinx Vivado oktató anyagok / videók:
  - <http://xilinxprod-catalog.netexam.com/>
  - <https://www.youtube.com/user/XilinxInc/featured>

# Xilinx Vivado Design Suite 2020.1

Laborokon a Xilinx Vivado Design Suite HLx 2020.1 integrált beágyazott rendszer fejlesztő környezetet használjuk!

- **\* Vivado: integrált fejlesztő környezet (IDE), amelyből más program modulok indíthatók, mint pl:**
  - IP Integrator / IP Catalog – IP magok paraméterezése
  - Source Editor: forráskód szerkesztő, stb.
  - ...
- **\* Vivado Simulator: integrált szimulátor**
  - Vivado (~EDK) – Beágyazott rendszer (firmware) fejlesztő környezete
  - Timing Analyzer – Időzítési analízátor
  - FPGA Editor (chip layout, floorplan)
  - **\* Hardware manager – Bitstream (FPGA konfiguráció) letöltő program**
  - ChipScope: logikai analízátor



\* Következő előadásokon részletesen tárgyaljuk és labor gyakorlatokon használjuk

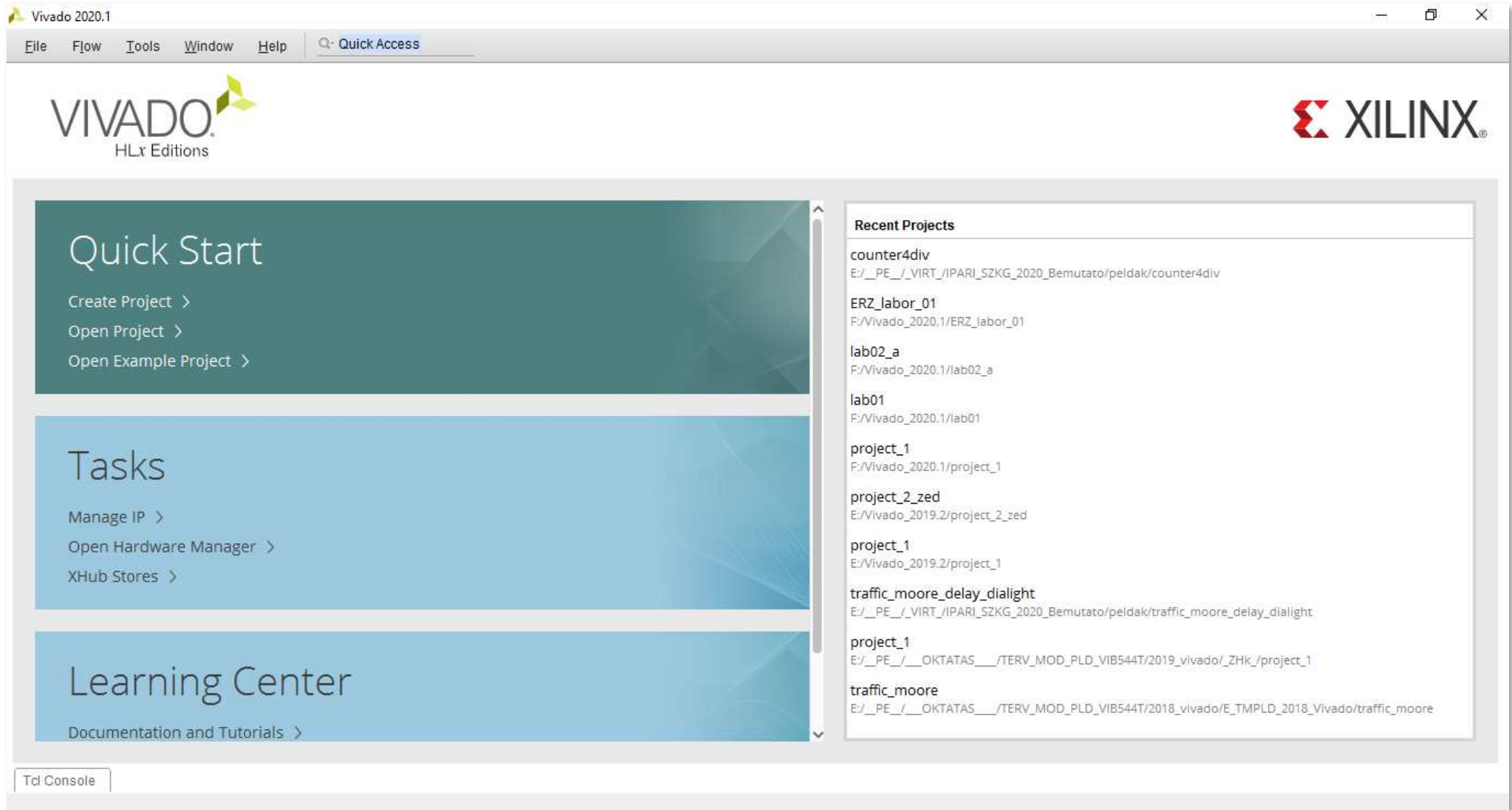
# Vivado IP integrator / IP Catalog

- Paraméterezhető IP-k (**Intellectual Property, azaz ún. „Szellemi termékek”**) adatbázisa, amelyeket kifejezette Xilinx FPGA eszközökre optimalizáltak. Különböző előre elkészített funkciók katalógusa, melyek paraméterei a Vivado GUI-n keresztül konfigurálhatóak. Ezekből szimulálható, illetve szintetizálható, ált. titkosított (encrypted) forrásokat, HDL leírókat generálhatunk, melyek beépíthetők a FW tervekbe.
- A generálható IP-k rendkívül sokoldalúak, komplexitásuk az egyszerű FIFO-tól egészen a memória vezérlőig, szűrőig terjedhet. Van közöttük:
  -  Legtöbb szabadon elérhető (freely available), mások
  -  Licenz kötelesek (licensed), de időkorláttal kipróbálhatóak\*
- *Low-level - alap blokkok/funkciók:*
  - Komparátorok, számlálók, Shift Regiszterek, Ált. Regiszterek, Akkumulátorok
  - FIFOs, (soft/hard-core) memória vezérlők
- *High-level - rendszer szintű funkciók:*
  - Reed-Solomon Dekódoló – Kódoló, FIR szűrő, FFT (DSP alkalmazásokhoz)
  - Szabványos busz interfészek (PCI™ , PCI-X™ ...)
  - Hálózati interfészek, összeköttetések (Ethernet, SPI-4.2, PCI ...)
  - PCI EXPRESS® mikroprocesszor interfészek

# Xilinx Vivado: Egyszerű példa

- Digilent ZyBo FPGA-s kártyán található LED-ek, illetve kapcsolók összekötése (egyelőre anélkül, hogy a VHDL nyelvi elemeit részletesen tárgyalnánk)
  - Feladat: 1 kapcsoló (`sw`) összekötése 1 LED kijelzővel (`led`)
- Főbb lépések a következők:
  - 1.) Projekt létrehozása (Create New project).
  - 2.) HDL fileok létrehozása (újat), vagy hozzáadása (már meglévőt).
  - 3. HDL nyelvi szintaxis ellenőrzése (Check HDL syntax).
- Xilinx Vivado keretrendszer indítása:
  - *Programok → Xilinx Design Tools → Vivado 2020.1*

# Vivado – Kezdő képernyő





# 1.a.) Projekt létrehozása

Create Project >

Egy Vivado projekt alapvető információkat tartalmaz a forrás fileokról (pl. HDL, XDC), illetve a használt FPGA eszközről (target device)

- a.) *Vivado* → *File* menü → *Project* → *NEW* (*Create a New project...*) → majd pedig *Next* gomb

*New project* ablaka jelenik meg. Adjuk meg a projekt nevét („sw2led”), és elérési útját.

- **Figyeljünk arra hogy az elérési út NE tartalmazzon ékezetet és white-space karaktereket!**

- **A projekt neve, és majd a HDL forrás neve NE kezdődjön számmal, csak betűvel (de lehet bennük szám).**

- **Lehetőség szerint a projekt neve és a forrás(ok) neve legyen eltérő, a későbbi hibaüzenetekben szereplő azonosítás végett.**

- Ellenőrizze, hogy a „*Top-level*” típus, azaz a legfelső hierarchia szinten lévő forrás típusa a *HDL* (lehet még EDIF/Schematic/NGC/NGO formátumot is hozzárendelni a projekthez).

# 1.a.) Projekt létrehozása

1

New Project

**Project Name**  
Enter a name for your project and specify a directory where the project data files will be stored.

Project name: sw2led

Project location: F:/TM\_2020\_Vivado\_2020.1

☒ Create project subdirectory

Project will be created at: F:/TM\_2020\_Vivado\_2020.1/sw2led

? < Back Next > Finish Cancel

2

# 1.a.) Projekt létrehozása (folyt.)

**New Project**

**Project Type**  
Specify the type of project to create.

1

☒ **RTL Project**  
You will be able to add sources, create block designs in IP Integrator, generate IP, run RTL analysis, synthesis, implementation, design planning and analysis.  
☒ **Do not specify sources at this time**

☐ **Post-synthesis Project:** You will be able to add sources, view device resources, run design analysis, planning and implementation.  
☐ **Do not specify sources at this time**

☐ **I/O Planning Project**  
Do not specify design sources. You will be able to view part/package resources.

☐ **Imported Project**  
Create a Vivado project from a Synplify, XST or ISE Project File.

☐ **Example Project**  
Create a new Vivado project from a predefined template.

2

RTL project kiválasztása: mivel HDL forrás fájlt akarunk készíteni - VHDL nyelven. Nem adunk hozzá forrásokat (sources).

# 1.b.) Projekt beállításai

## FPGA kártya kiválasztása

New Project

**Default Part**  
Choose a default Xilinx part or board for your project.

1 **Boards**

2 **Reset All Filters**

Vendor:  Name:  Board Rev:

Search:  (3 matches)

| Display Name | Preview | Vendor          | File Version | Part            |
|--------------|---------|-----------------|--------------|-----------------|
|              |         | digilentinc.com | 1.0          | xc7z010clg400-1 |
| Zybo Z7-20   |         | digilentinc.com | 1.0          | xc7z020clg400-1 |
| 3 Zybo       |         | digilentinc.com | 2.0          | xc7z010clg400-1 |

4 **Next >**

Fontos az **FPGA** kártya kiválasztása megadása!

\*Fontos: a Zybo kártya támogatásához a Digilent oldaláról a támogató csomagot le kell tölteni, és a megfelelő helyre másolni!

\* <https://reference.digilentinc.com/reference/software/vivado/board-files?redirect=1>



# 1.c.) Projekt beállításai

## FPGA típus kiválasztása (alternatív módszer)

New Project

**Default Part**  
Choose a default Xilinx part or board for your project.

1 **Parts** Boards

2 [Reset All Filters](#)

Category: All Package: clg400 Temperature: All Remaining  
Family: Zynq-7000 Speed: -1

Search: Q-

| Part             | I/O Pin Count | Available IOBs | LUT Elements | FlipFlops | Block RAMs | Ultra RAMs | DSPs | Gt |
|------------------|---------------|----------------|--------------|-----------|------------|------------|------|----|
| xc7z007sclg400-1 | 400           | 100            | 14400        | 28800     | 50         | 0          | 66   | 0  |
| xc7z010clg400-1  | 400           | 100            | 17600        | 35200     | 60         | 0          | 80   | 0  |
| xc7z014sclg400-1 | 400           | 125            | 40600        | 81200     | 107        | 0          | 170  | 0  |
| xc7z020clg400-1  | 400           | 125            | 53200        | 106400    | 140        | 0          | 220  | 0  |

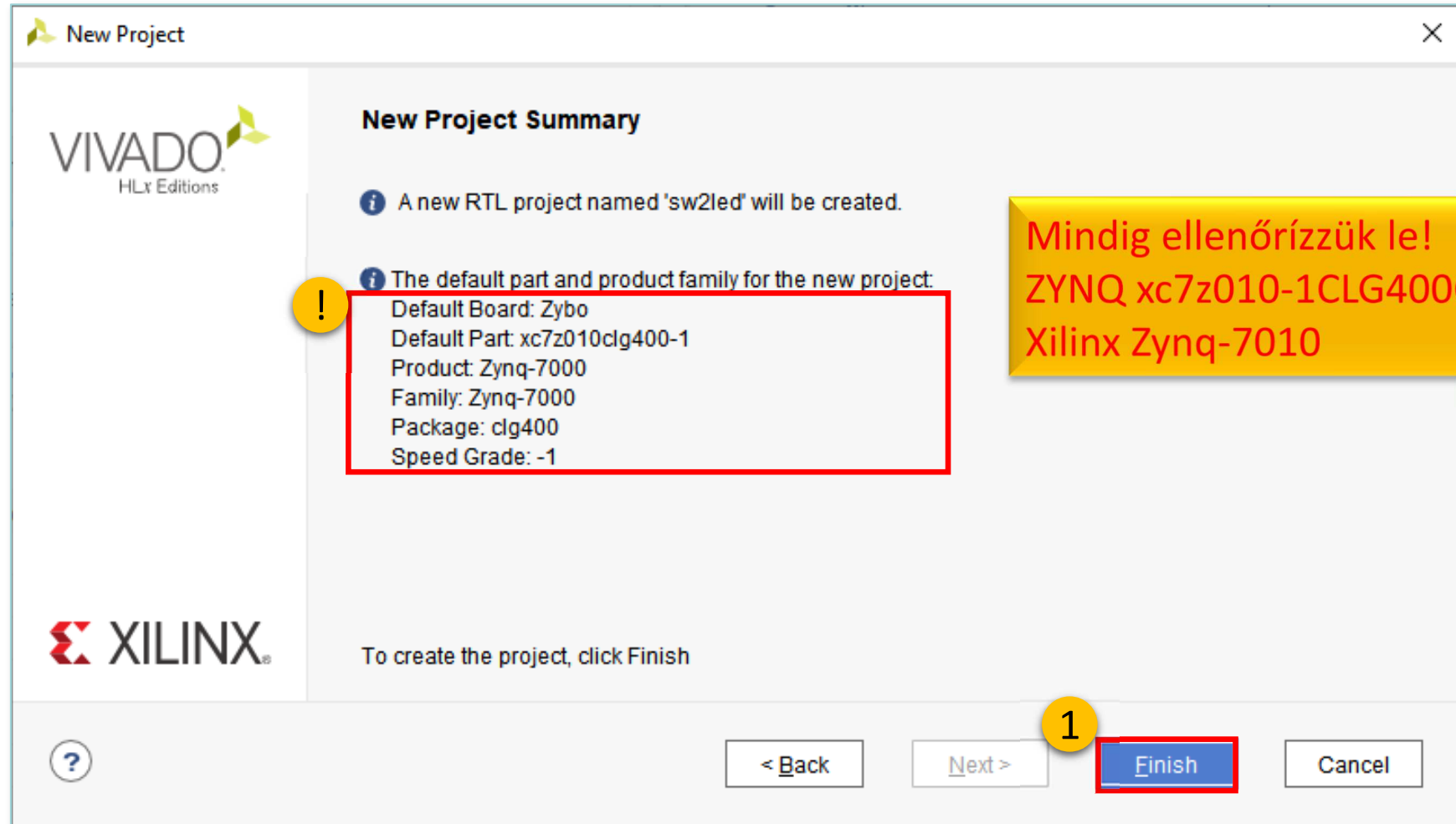
3

4

< Back Next > Finish Cancel

Fontos az **FPGA** típusának pontos megadása!

# 1.d.) Összegzés



# Vivado GUI

Flow Navigator

Project Manager

Layout selector

Workplace window

The screenshot displays the Vivado 2020.1 GUI interface. The top menu bar includes File, Edit, Flow, Tools, Reports, Window, Layout, View, and Help. Below the menu bar is a toolbar with various icons. The main workspace is divided into several panes:

- Flow Navigator:** Located on the left, it shows a tree view of the project structure. The 'PROJECT MANAGER' section is expanded, showing 'Settings', 'Add Sources', 'Language Templates', and 'IP Catalog'. The 'IP INTEGRATOR' section is also expanded, showing 'Create Block Design', 'Open Block Design', and 'Generate Block Design'. The 'SIMULATION' section shows 'Run Simulation'. The 'RTL ANALYSIS' section shows 'Open Elaborated Design'. The 'SYNTHESIS' section shows 'Run Synthesis' and 'Open Synthesized Design'. The 'IMPLEMENTATION' section shows 'Run Implementation'.
- Project Manager:** Located in the center, it shows the 'Sources' pane with 'Design Sources', 'Constraints', and 'Simulation Sources'. The 'Hierarchy' pane is selected, showing a tree view of the project hierarchy. The 'Properties' pane is empty, with a message 'Select an object to see properties'.
- Design Runs:** Located at the bottom, it shows a table of design runs. The 'Design Runs' tab is selected, showing a table with columns for Name, Constraints, Status, WNS, TNS, WHS, THS, TPWS, Total Power, Failed Routes, LUT, FF, BRAM, URAM, DSP, Start, Elapsed, and Run Strategy. The table contains two rows: 'synth\_1' and 'impl\_1', both with a status of 'Not started'.

The 'Project Summary' pane on the right shows the project details:

- Project name: sw2led
- Project location: F:/TM\_2020\_Vivado\_2020.1/sw2led
- Product family: Zynq-7000
- Project part: Zybo (xc7z010cpg400-1)
- Top module name: Not defined
- Target language: VHDL
- Simulator language: Mixed

The 'Layout selector' pane at the top right shows a dropdown menu with 'Default Layout' selected.

Result window

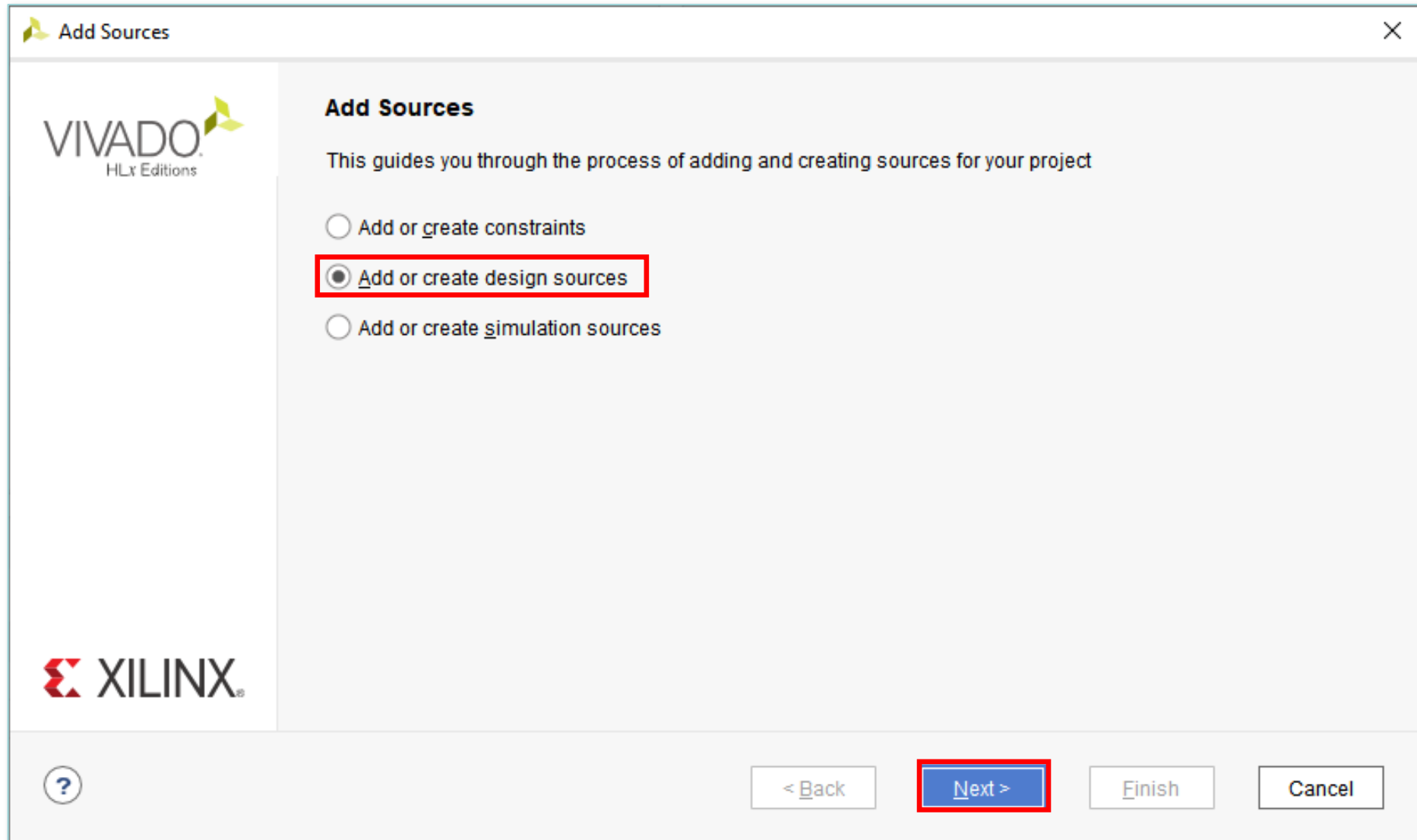
## 2. Tervezői file (HDL) létrehozása

- Itt kell hozzáadni a meglévő, vagy létrehozni egy HDL forrás file.
  - a.) *File menü + Add Sources. Add or create design sources* – Hozzá kell adni egy új forrásfilet. Itt a 'VHDL'-t kell kiválasztani. A file neve legyen „sw2led\_top.vhd”. OK, majd Finish.
  - b.) A következő ablakban a port neveket lehet megadni, típusaikkal, illetve busz szélességükkel. „+” a hozzáadás:
    - Port Type/Name/Bus: Input 'sw' – nem busz típus
    - Port Type/Name/Bus: Output 'led' – nem busz típus
    - Architektúra neve maradhat *Behavioral* (viselkedési). OK.
- Az Vivado IDE automatikusan generálja le a **VHDL** forrásfile tartalmát, amely a headert, library neveket, entitás deklarációkat és egy üres architektúra vázat tartalmaz még csak.
  - c.) Alapértelmezettként, az Vivado a következő library hozzárendeléseket generálja le :  
`library IEEE;`  
`use IEEE.STD_LOGIC_1164.ALL;`  
  
A fenti könyvtár IEEE szabvány, de lehet helyette még alkalmazni a `IEEE.STD-LOGIC-ARITH.ALL`, vagy a `IEEE.STD-LOGIC-UNSIGNED.ALL` könyvtárakat is.
  - d.) a szöveges editor használható a HDL forráskód szerkesztéséhez ,
  - e.) Ha szükséges ezek a lépések megismételhetők, amikor több forrásfilet is hozzá kell adni a projekthez. De ezt akár egy lépésben is meg lehet tenni.



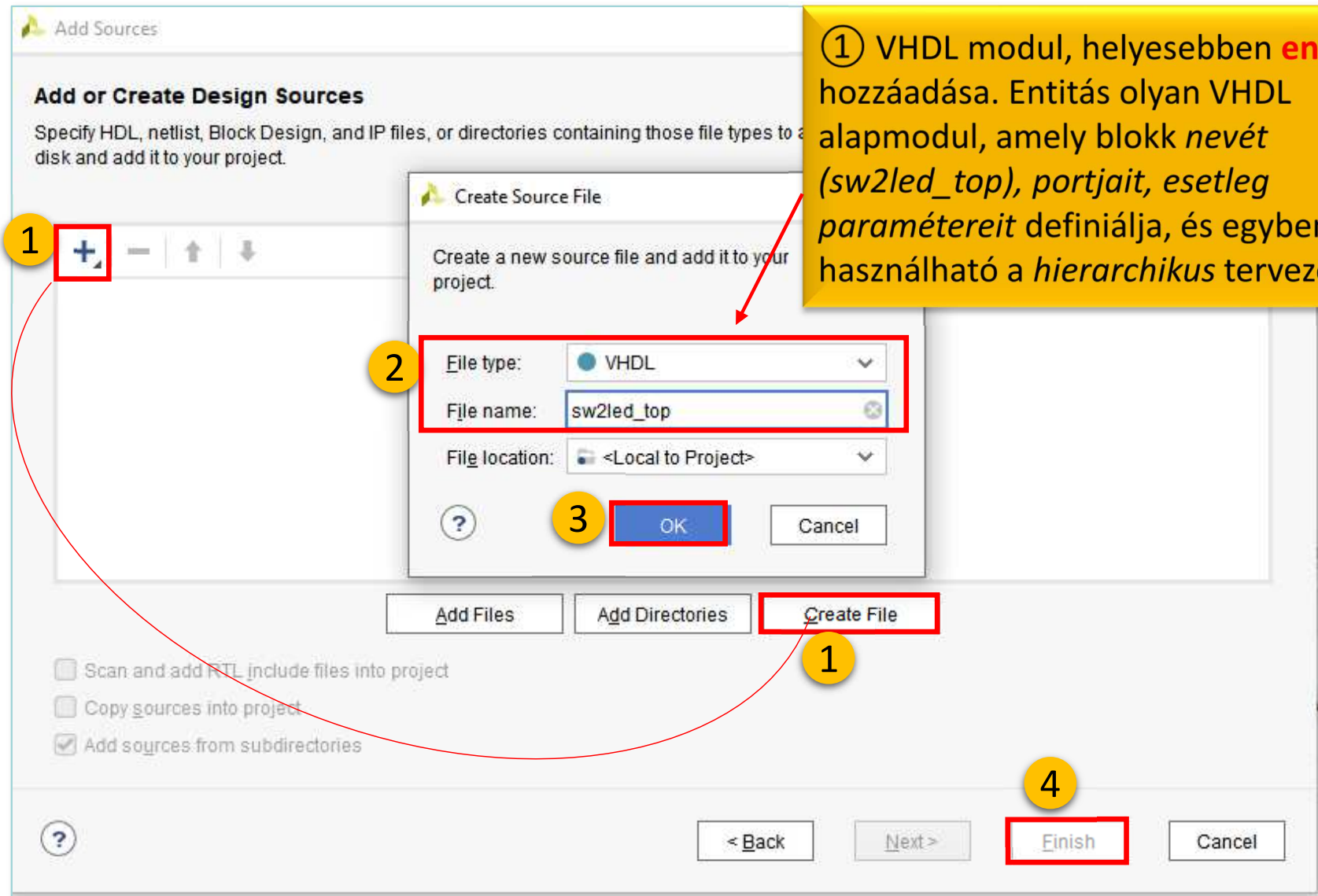
## 2.a.) HDL tervezői entitás létrehozása

*File menü -> Add Sources -> Add or Create Design Sources +*



## 2.a.) HDL entitás létrehozása (folyt)

*File menü -> Add Files -> Create Source Files*



## 2.b.) Portok megadása

**Define Module**

Define a module and specify I/O Ports to add to your source file.  
For each port specified:  
MSB and LSB values will be ignored unless its Bus column is checked.  
Ports with blank names will not be written.

**Module Definition**

Entity name:

Architecture name:

**I/O Port Definitions**

**+** **-** **↑** **↓**

| Port Name | Direction |   | Bus                      | MSB | LSB |
|-----------|-----------|---|--------------------------|-----|-----|
| sw        | in        | ▼ | <input type="checkbox"/> | 0   | 0   |
| led       | out       | ▼ | <input type="checkbox"/> | 0   | 0   |

**?** **OK** **Cancel**

## 2.c.) Generált HDL forrásfile

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
-- use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells/primitives in this code.
-- library UNISIM;
-- use UNISIM.VComponents.all;

entity sw2led_top is
    Port ( sw    : in    STD_LOGIC ;
          led    : out   STD_LOGIC );
end sw2led_top;

architecture Behavioral of sw2led_top is

begin
    } Ide kerülnek a felhasználói kódrészletek
end Behavioral;
```



### 3.) Szintaxis/Szintézis ellenőrzés

- A **Messages / TCL console / Log ablakok** is megjelenítik az ellenőrzés eredményét (riportolják az esetleges hibákat, és figyelmeztetéseket adnak, stb.)
  - Hiba/leáll a futtatás =  Error (3) , Figyelmeztetés/tovább lép =  2 warnings
- Mindaddig amíg hiba van a forrásfile-ban, el kell végezni a korrekciót és azután a szintézis futtatását.
  - a. ) Próbaképpen: a **begin** és **end** közötti részre adjuk meg először:  

```
sw <= led;
```

 //Ekkor szintaxis hibát kapunk mivel az inputhoz rendeltük az output-ot, tehát a portok irányultságához képest ellentétesen van hajtva.  
 [HDL 9-806] Syntax error near "sw". [sw2led\_top.vhd:40]  
 [HDL 9-806] Syntax error near "sw". [sw2led\_top.vhd:40]
  - b. ) Majd próbáljuk ki:  

```
led <= sw;
```

 // Ekkor OK.
- Végül a HDL forrás(ok) szerkesztése után a *Run Synthesis -el* lehet a nyelvi szintaxis ellenőrzést is elvégeztetni.
  - a.) A kívánt HDL forrás kiválasztása a **Sources** ablakban
    - [+] megjeleníthetők az egyes hierarchia szintek .
  - b.) Majd alatta a **Project Manager** ablakban -> Run Synthesis,
    - synth\_design completed successfully

# 3.a.) Szintézis futtatása

The screenshot shows the Vivado 2018.3 IDE interface. The top menu bar includes File, Edit, Flow, Tools, Reports, Window, Layout, View, and Help. The toolbar contains various icons for file operations and synthesis. The Flow Navigator on the left shows the project hierarchy, with the 'SYNTHESIS' section expanded. The 'Run Synthesis' button is highlighted with a red box and a yellow circle with the number 1. The 'Open Synthesized Design' section is also visible. The Sources panel in the center shows the project sources, with 'sw2led\_top(Behavioral) (sw2led\_top.vhd)' highlighted. The Source File Properties panel below it shows the file 'sw2led\_top.vhd' with a library of 'xilinx\_defaultlib' and a size of 1.0 KB. The code editor on the right shows the VHDL code for the entity 'sw2led\_top'. The code is as follows:

```
29 -- Uncomment the following library declaration if instantiating
30 -- any Xilinx leaf cells in this code.
31 --library UNISIM;
32 --use UNISIM.VComponents.all;
33
34 entity sw2led_top is
35     Port ( sw : in STD_LOGIC;
36           led : out STD_LOGIC);
37 end sw2led_top;
38
39 architecture Behavioral of sw2led_top is
40
41 begin
42     led <= sw;
43
44 end Behavioral;
45
```

The I/O Ports panel at the bottom shows the port configuration for the entity. It includes columns for Name, Direction, Board Part Pin, Board Part Interface, Neg Diff Pair, Package Pin, Fixed, Bank, I/O Std, Vcco, Vref, Drive Strength, Slew Type, Pull Type, and Off-Chip Termination. The 'All ports (2)' section is expanded, showing 'Scalar ports (2)'.

## 3.b.) Szintetizálási hiba

The screenshot displays the Vivado IDE interface. The top window shows the source code for `sw2led_top.vhd`. The code defines an entity `sw2led_top` with an input port `sw` and an output port `led`. In the behavioral architecture, line 42 contains the assignment `sw <= led;`, which is highlighted in yellow. A red box around this line points to the error messages in the Messages window below.

The Messages window shows the following errors:

- [Synth 8-1789] cannot update 'in' object sw [sw2led\_top.vhd:42]
- [Synth 8-1779] cannot read from 'out' object led ; use 'buffer' or 'inout' [sw2led\_top.vhd:42]
- [Common 17-69] Command failed: Vivado Synthesis failed

A "Synthesis Failed" dialog box is also visible, indicating that the synthesis process did not complete successfully. The dialog box includes options to "View Messages" and "Don't show this dialog again".

## 3.b.) Hibák szűrése (*Messages* ablak) folyt.





# 3. c.) Schematic – kapcsolási rajz

Szintézis futtatása után

The screenshot displays the Xilinx Vivado IDE interface after synthesis. The top status bar indicates 'Synthesis Complete' with a green checkmark. The 'Flow Navigator' on the left shows the 'SYNTHESIS' phase completed, with the 'Schematic' option highlighted under 'Open Synthesized Design' (marked with a red box and a yellow circle with the number 1). The main workspace shows the 'SYNTHESIZED DESIGN - xc7z010clg400-1 (active)' project. The 'Sources' pane on the left lists the design sources, including 'sw2led\_top(Behavioral) (sw2led\_top.vhd)'. The 'Schematic' view is open, showing a circuit diagram with a switch 'sw' connected to an input buffer 'sw\_IBUF\_inst' (labeled 'IBUF'), which is connected to an output buffer 'led\_OBUF\_inst' (labeled 'OBUF'), which is connected to an LED 'led'. The 'Schematic' tab is highlighted in the top window bar (marked with a red box and a yellow circle with the number 2). A yellow banner at the bottom of the schematic view reads 'Technology Schematic-nak megfelelő felépítés'. The bottom status bar shows the 'I/O Ports' tab, displaying a table of ports.

| Name             | Direction | Board Part Pin | Board Part Interface | Neg Diff Pair | Package Pin | Fixed | Bank | I/O Std | Vcco | Vref | Drive Strength | Slew Type | Pull Type | Off-Chip Termination |
|------------------|-----------|----------------|----------------------|---------------|-------------|-------|------|---------|------|------|----------------|-----------|-----------|----------------------|
| All ports (2)    |           |                |                      |               |             |       |      |         |      |      |                |           |           |                      |
| Scalar ports (2) |           |                |                      |               |             |       |      |         |      |      |                |           |           |                      |

# 4.) Felhasználói megkötések (XDC = Xilinx Design Constraints)

Flow Navigator

- SYNTHESIS
  - Run Synthesis
  - Open Synthesized Design
    - Constraints Wizard
    - Edit Timing Constraints
  - Set Up Debug
  - Report Timing Summary
  - Report Clock Networks
  - Report Clock Interaction
  - Report Methodology
  - Report DRC
  - Report Noise
  - Report Utilization
  - Report Power
  - Schematic
- IMPLEMENTATION
  - Run Implementation
  - Open Implemented Design
- PROGRAM AND DEBUG
  - Generate Bitstream

SYNTHESIZED DESIGN - xc7z010clg400-1 (active)

Sources x Netlist Device Constraints

Design Sources (1)

- sw2led\_top(Behavioral) (sw2led\_top.vhd)

Constraints

- constrs\_1

Simulation Sources (1)

- sim\_1 (1)

Utility Sources

Jobb gomb (constrs\_1) → Add Sources

Package x Device x sw2led\_top.vhd x Schematic

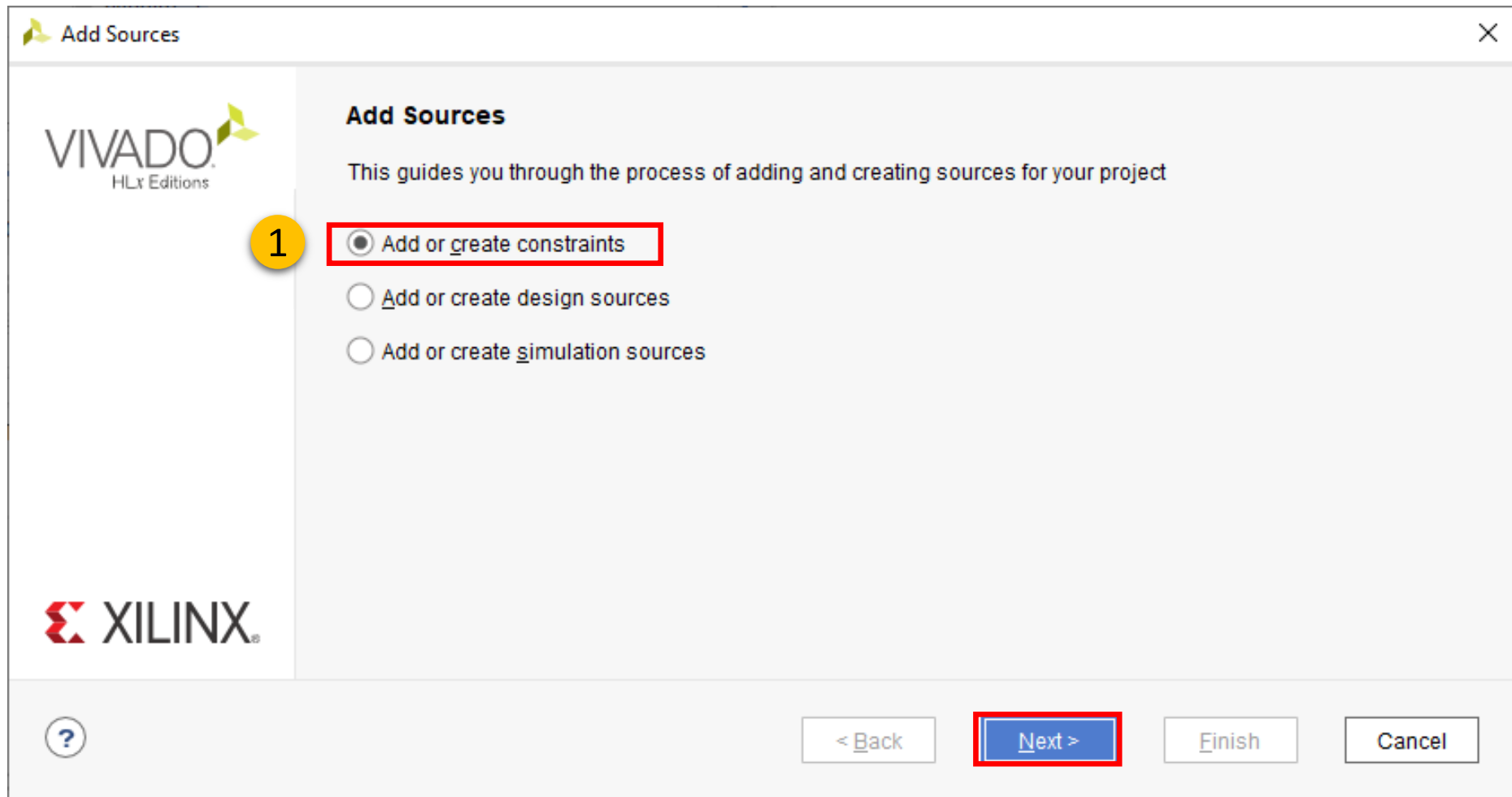
2 Cells 2 I/O Ports 3 Nets

sw sw\_IBUF\_inst IBUF led\_OBUF\_inst OBUF led

Tcl Console Messages Log Reports Design Runs Package Pins I/O Ports

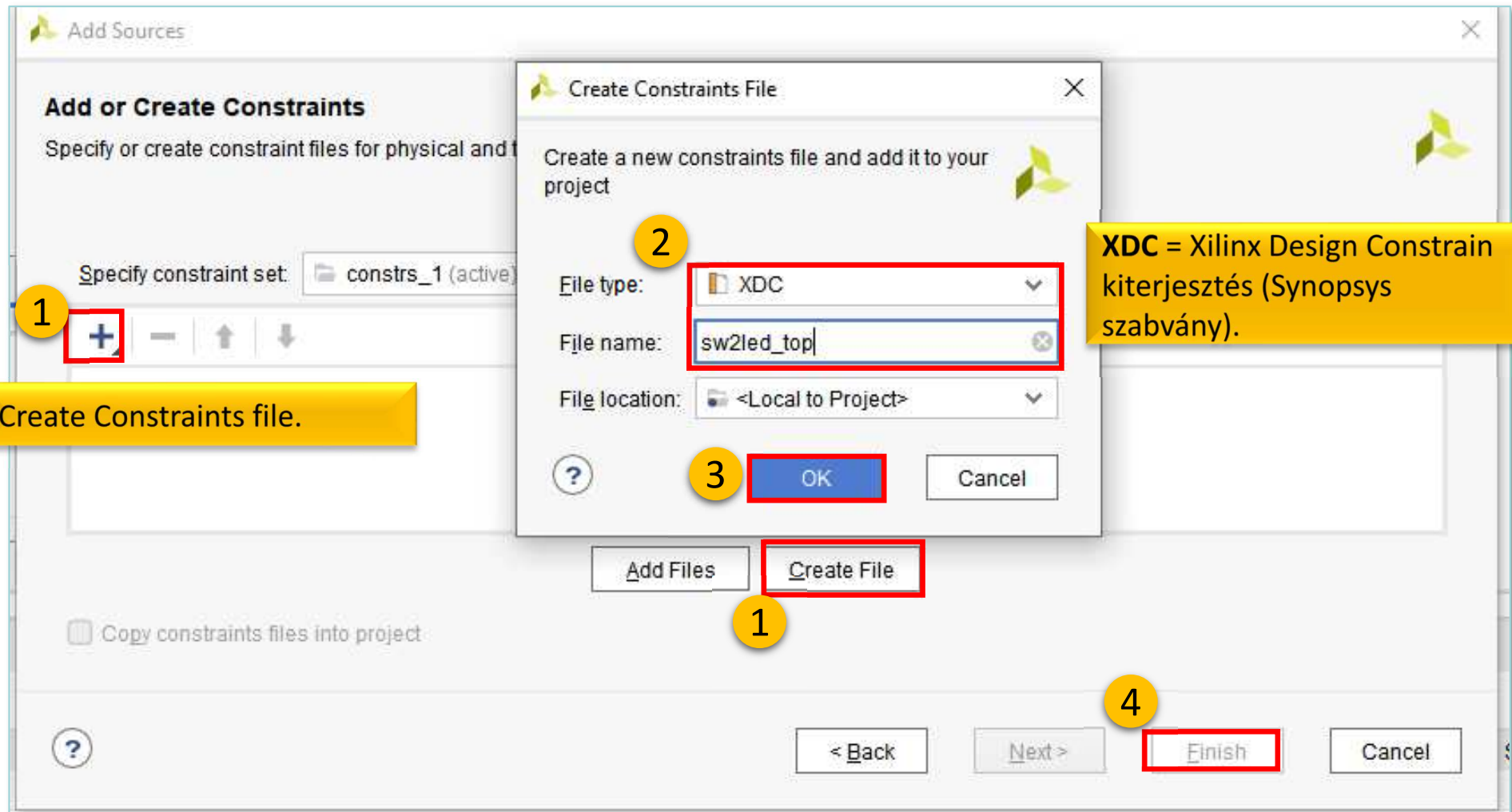
| Name             | Direction | Board Part Pin | Board Part Interface | Neg Diff Pair | Package Pin | Fixed | Bank | I/O Std | Vcco | Vref | Drive Strength | Slew Type | Pull Type | Off-Chip Termination |
|------------------|-----------|----------------|----------------------|---------------|-------------|-------|------|---------|------|------|----------------|-----------|-----------|----------------------|
| All ports (2)    |           |                |                      |               |             |       |      |         |      |      |                |           |           |                      |
| Scalar ports (2) |           |                |                      |               |             |       |      |         |      |      |                |           |           |                      |

# 4.a.) Felhasználói megkötések (Add or create constraints – XDC)





## 4.b.) Xilinx Constraint file (XDC) létrehozása





## 4.c.) Lábak hozzárendelése

The screenshot shows the Xilinx Vivado IDE interface. The 'Flow Navigator' on the left has 'SYNTHESIS' selected. The 'Design Sources' pane shows 'sw2led\_top.xdc' highlighted with a red box and a yellow circle labeled '1'. The 'Sources' pane shows the same file. The 'Schematic' pane shows the 'sw2led\_top.xdc' file open, with the following code:

```
#Switches
set_property -dict { PACKAGE_PIN G15 IOSTANDARD LVCMOS33 } [get_ports { sw }]; #IO_L19N_T3_VREF_35 Sch=SW0
#LEDs
set_property -dict { PACKAGE_PIN M14 IOSTANDARD LVCMOS33 } [get_ports { led }]; #IO_L23P_T3_35 Sch=LED0
```

The code is highlighted with a red box and a yellow circle labeled '2'. A red arrow points from the 'sw' and 'led' variables in the code to a yellow box containing the following text:

**SW0 és a LED0 részek bemásolása a ZYBO Master.xdc alapján (forrása a Digilent GIT-en). Átírni:**  
**Sw[0] → sw**  
**Led[0] → led**

Below this text is a table titled 'Letölthető gyakorlati anyag' (Downloadable practical material).

| Letölthető gyakorlati anyag                                                                                                     |
|---------------------------------------------------------------------------------------------------------------------------------|
| -                                                                                                                               |
| Digilent ZyBo Fejlesztőkártyákhoz (BSP, XDC):<br><b>XDC (FPGA lábkiosztás - GIT master):</b><br><a href="#">Zybo-Master.xdc</a> |

# 5.) Implementáció futtatása

The screenshot shows the Xilinx IDE interface. On the left, the 'Flow Navigator' pane is visible, with the 'IMPLEMENTATION' section expanded. The 'Run Implementation' button is highlighted with a red box and a yellow circle containing the number 1. The main editor displays the VHDL code for 'sw2led\_top.vhd'.

```
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity sw2led_top is
    Port ( sw : in STD_LOGIC;
          led : out STD_LOGIC);
end sw2led_top;

architecture Behavioral of sw2led_top is
begin
    led <= sw;
end Behavioral;
```

The bottom pane shows the 'I/O Ports' table, which lists the ports for the design.

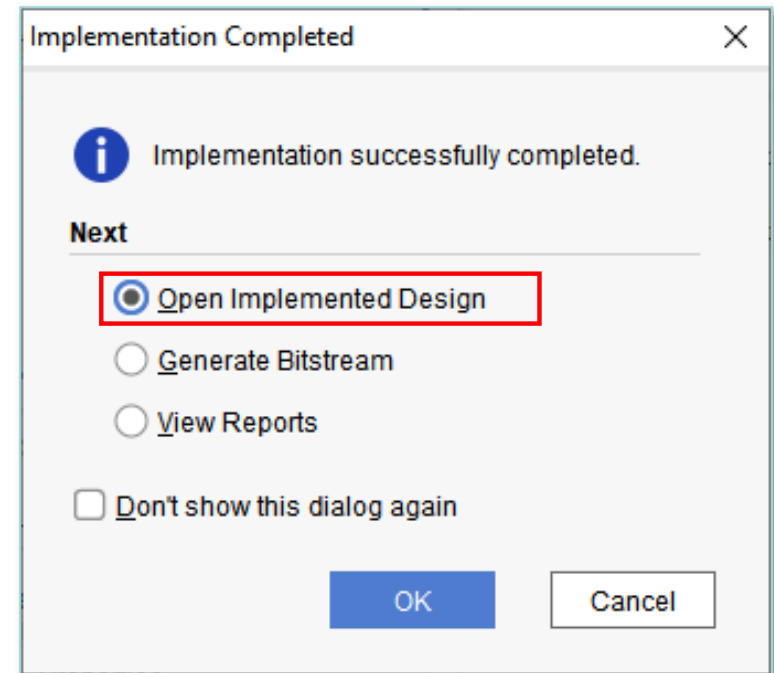
| Name             | Direction | Board Part Pin | Board Part Interface | Neg Diff Pair | Package Pin | Fixed | Bank | I/O Std | Vcco | Vref | Drive Strength | Slew Type | Pull Type | Off-Chip Termination |
|------------------|-----------|----------------|----------------------|---------------|-------------|-------|------|---------|------|------|----------------|-----------|-----------|----------------------|
| All ports (2)    |           |                |                      |               |             |       |      |         |      |      |                |           |           |                      |
| Scalar ports (2) |           |                |                      |               |             |       |      |         |      |      |                |           |           |                      |

Flow manager → Implementation → Run implementation (kb 3-5 perc!)

## 5.) Implementáció (folyt.)

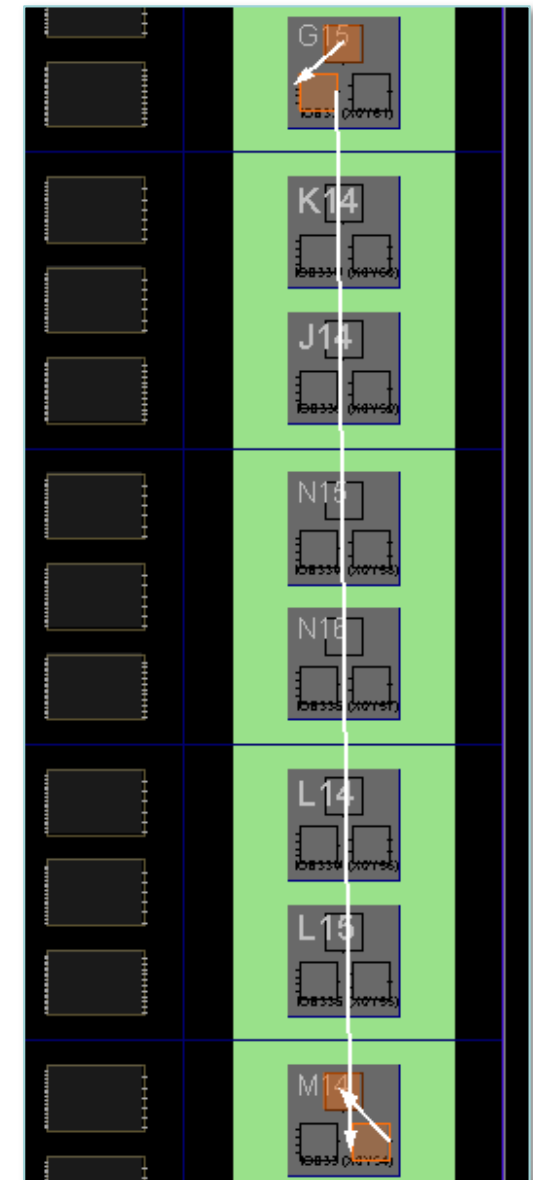
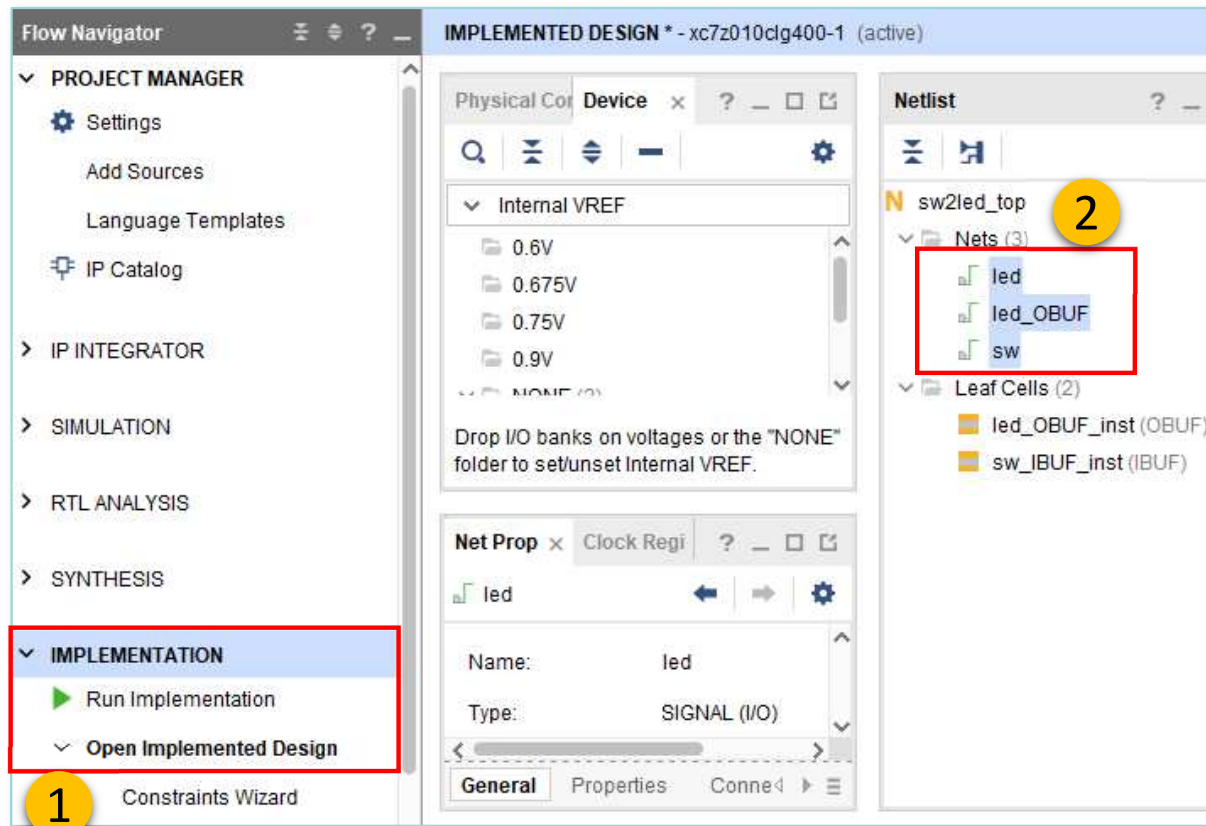
### Futtatás végén rákérdez:

- az implementált (MAP, PLACE & ROUTE) megnyitására,
  - Open Implemented Design,
- Bitstream generálására,
  - Generate Bitstream (.bit),
- Fordítási riportok megtekintésére
  - View Riports.





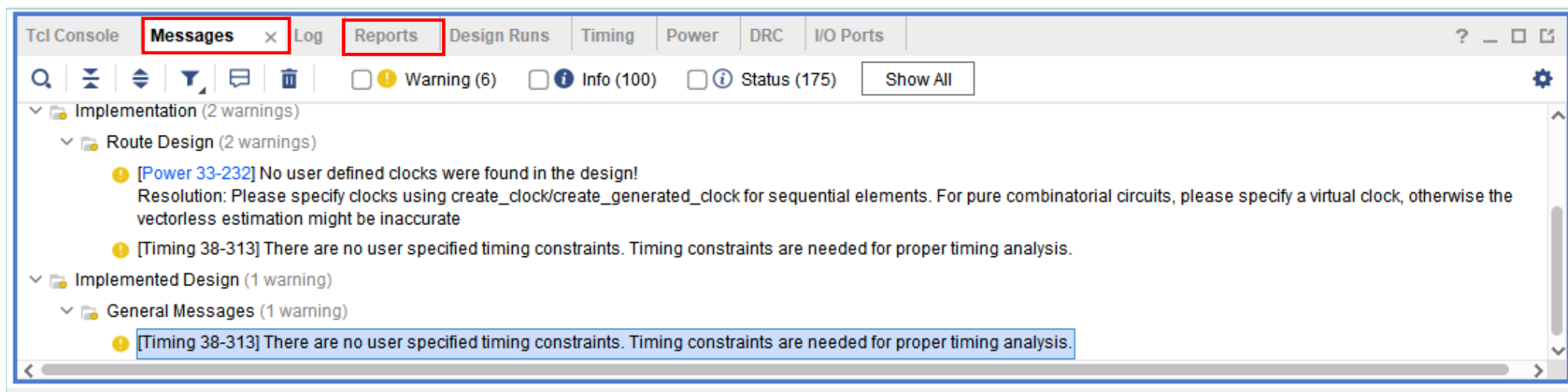
# 5.a.) Implementált – „Floorplan” design megnyitása



Elérhető még a *Layout* menü -> *Floorplanning* opció alól is!



## 5.b.) Üzenetek - Riportok



OK. Most nincsenek megadva a clock constraint-ek.

DE: Szinkron órajeles feladatok tervezése esetén (S.H.) meg kell majd adnunk!

Most még nem használtunk FPGA logikát sem, csak I/O Buffereket.

## 5.c.) I/O Planning

Implementation Complete ✓

1 I/O Planning

Flow Navigator

- Open Synthesized Design
  - Constraints Wizard
  - Edit Timing Constraints
- Set Up Debug
- Report Timing Summary
- Report Clock Networks
- Report Clock Interaction
- Report Methodology
- Report DRC
- Report Noise
- Report Utilization
- Report Power
- Schematic

IMPLEMENTATION

- Run Implementation
- Open Implemented Design
  - Constraints Wizard
  - Edit Timing Constraints
  - Report Timing Summary
  - Report Clock Networks
  - Report Clock Interaction

Sources Netlist Device Constraints

Internal VREF

- 0.6V
- 0.675V
- 0.75V
- 0.9V
- NONE (2)

Drop I/O banks on voltages or the "NONE" folder to set/unset Internal VREF.

I/O Port Properties

led

General Properties Configure Power

Package Device sw2led\_top.vhd sw2led\_top.xdc

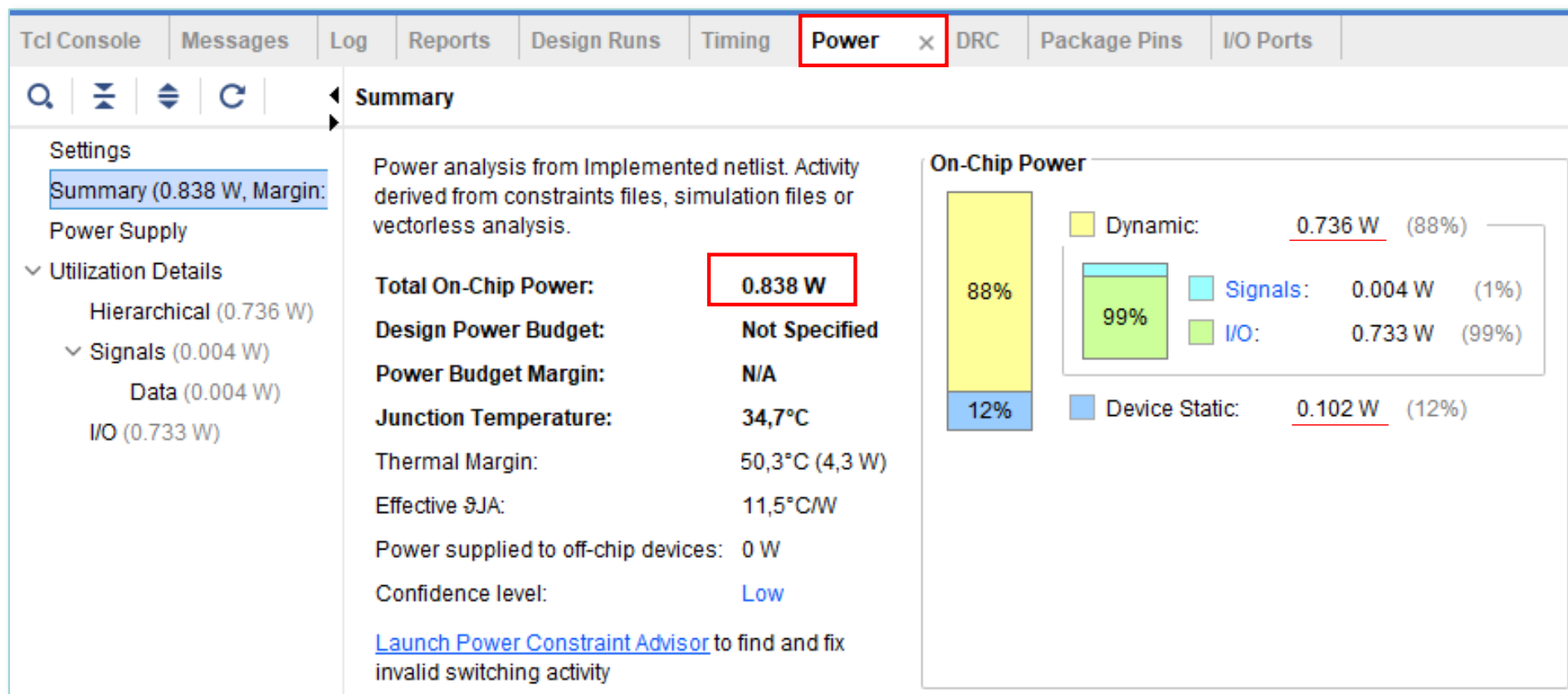
Tcl Console Messages Log Reports Design Runs Timing Power DRC Package Pins I/O Ports

DRC - impl\_1 (1 violation) (saved)

| Name             | Direction | Board Part Pin | Board Part Interface | Neg Diff Pair | Package Pin | Fixed | Bank | I/O Std    | Vcco  | Vref | Drive Strength | Slew Type |
|------------------|-----------|----------------|----------------------|---------------|-------------|-------|------|------------|-------|------|----------------|-----------|
| All ports (2)    |           |                |                      |               |             |       |      |            |       |      |                |           |
| Scalar ports (2) |           |                |                      |               |             |       |      |            |       |      |                |           |
| led              | OUT       |                |                      |               | M14         | ✓     | 35   | LVC MOS33* | 3.300 | 12   | SLOW           |           |
| sw               | IN        |                |                      |               | G15         | ✓     | 35   | LVC MOS33* | 3.300 |      |                |           |

Elérhető a *Layout* menü → *I/O planning* opció alól is!

## 5.d.) Power – Disszipált teljesítmény





# 6.) Bitstream generálás

Implementation Complete ✓

I/O Planning

Flow Navigator

- RTL ANALYSIS
- SYNTHESIS
- IMPLEMENTATION
  - Run Implementation
  - Open Implemented Design
    - Constraints Wizard
    - Edit Timing Constraints
    - Report Timing Summary
    - Report Clock Networks
    - Report Clock Interaction
    - Report Methodology
    - Report DRC
    - Report Noise
    - Report Utilization
  - Report Power
  - Schematic
- PROGRAM AND DEBUG
  - Generate Bitstream**
  - Open Hardware Manager

IMPLEMENTED DESIGN \* - xc7z010clg400-1 (active)

Sources Netlist Device Constraints

Internal VREF

0.75V

0.9V

Drop I/O banks on voltages or the "NONE" folder to set/unset Internal VREF.

I/O Port Properties

led

Name: led

General Properties Configure Power

Package Pins

| ion       | Board Part Pin | Board Part Interface | Neg Diff Pair | Package Pin | Fixed | Bank | I/O Std | Vcco   | Vref | Drive Strength | Slew Type | Pull Type | Off-Chip Termination | IN_TERM |
|-----------|----------------|----------------------|---------------|-------------|-------|------|---------|--------|------|----------------|-----------|-----------|----------------------|---------|
| ports (2) |                |                      |               | M14         | ✓     | 35   | L...    | 3.3... | 12   | SLOW           | NONE      | FP_VTT_50 |                      |         |
|           |                |                      |               | G15         | ✓     | 35   | L...    | 3.3... |      |                | NONE      | NONE      |                      |         |

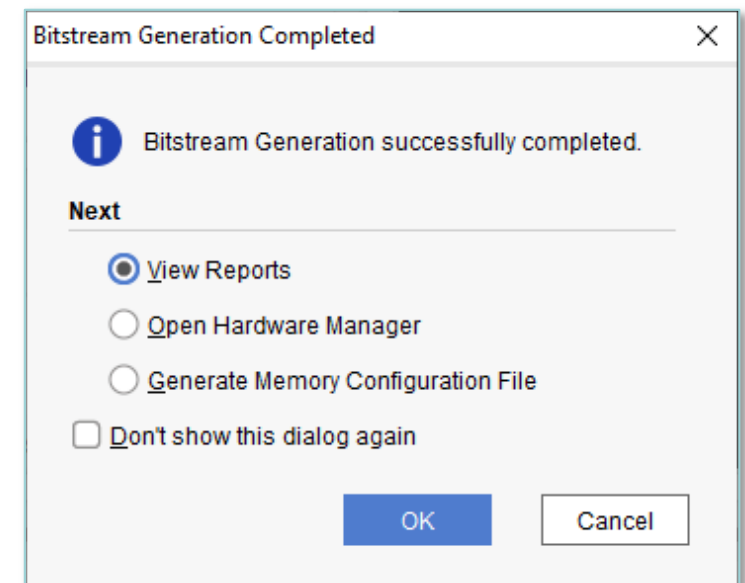


## 6.) Bitstream generálás (folyt.)

**Bitstream = FPGA konfiguráció generálás végén két lehetőség:**

- Fordítási riport megtekintése,
  - View Reports
- Hardver Manager megnyitása (felprogramozáshoz)
  - Open Hardware Manager
- Memória konfiguráció generálása

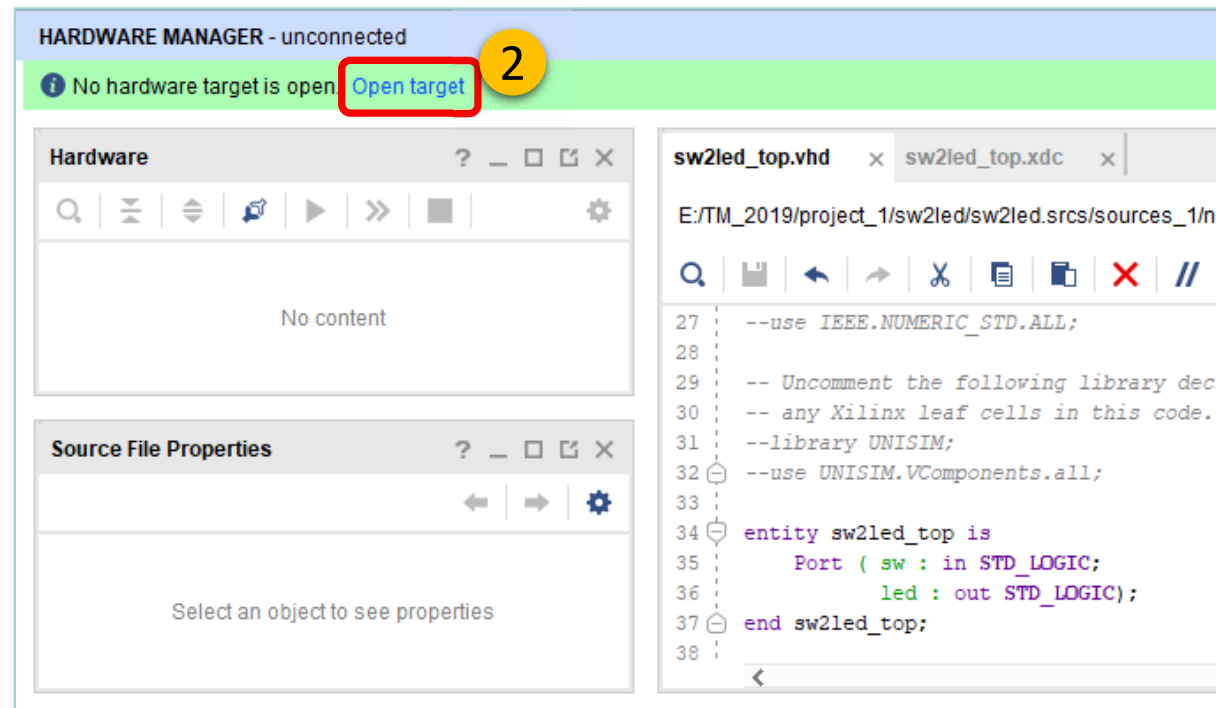
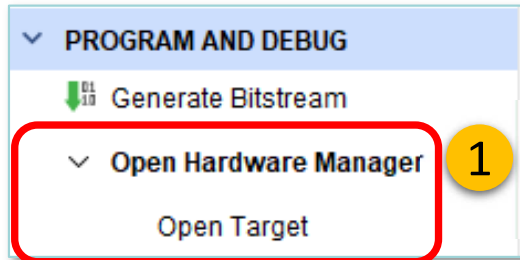
write\_bitstream Complete ✓



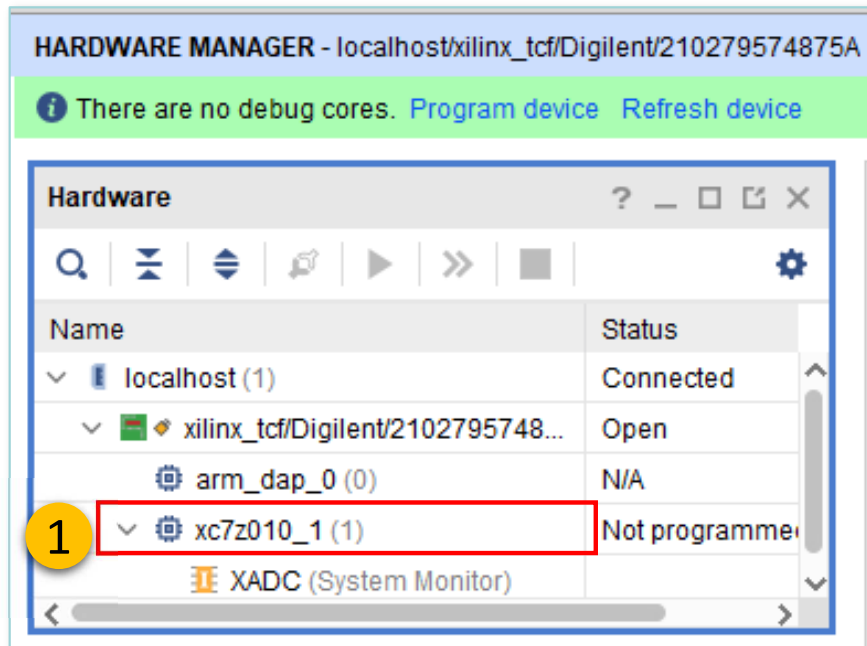
# 7.) FPGA „programozása”



- Open Hardware manager
  - Hardware Manager (Open target → Auto Connect)
- JTAG lánc két eleme (platformonként eltérő lehet)
  - ARM\_DAP: Debug Access Port (nem ezt használjuk)
  - FPGA (APSoC): xc7z010 = Xilinx Zynq 7010



# 7.a.) Hardware manager

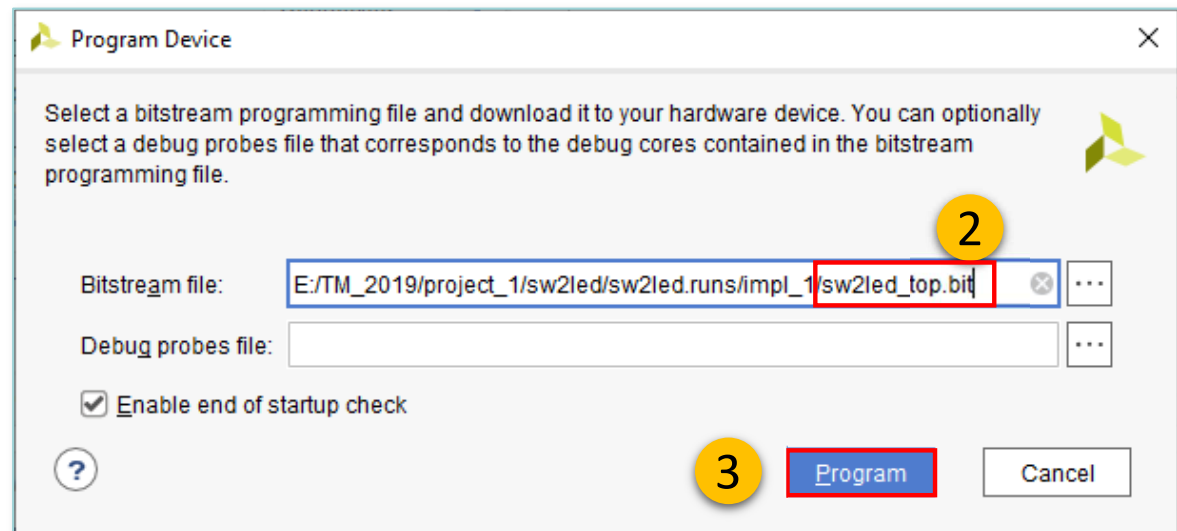


1.) xc7z010 kiválasztása (FPGA)

Jobb gomb → Program device

2.) Bitstream kiválasztása (.bit)

3.) Programozás



Bemutató egyszerű példán keresztül

## **II. XILINX VIVADO INTEGRÁLT SZIMULÁTOR**



# Felhasznált irodalom:

-  Xilinx Vivado Design Suite IN depth tutorial (UG937)  
[https://www.xilinx.com/support/documentation/sw\\_manuals/xilinx2020\\_1/ug937-vivado-design-suite-simulation-tutorial.pdf](https://www.xilinx.com/support/documentation/sw_manuals/xilinx2020_1/ug937-vivado-design-suite-simulation-tutorial.pdf)
-  Xilinx Vivado User Guide – Logic simulation (UG990)  
[www.xilinx.com/support/documentation/sw\\_manuals/xilinx2020\\_1/ug900-vivado-logic-simulation.pdf](http://www.xilinx.com/support/documentation/sw_manuals/xilinx2020_1/ug900-vivado-logic-simulation.pdf)
-  Vivado Design Suite User Guide: Using the Vivado IDE (UG893)  
[https://www.xilinx.com/support/documentation/sw\\_manuals/xilinx2020\\_1/ug893-vivado-ide.pdf](https://www.xilinx.com/support/documentation/sw_manuals/xilinx2020_1/ug893-vivado-ide.pdf)
-   Xilinx Vivado oktató anyagok / videók:
  - <https://www.xilinx.com/video/hardware/logic-simulation.html>

# Xilinx Vivado szimulátor

- Xilinx Vivado Design Suite (2020.1)

## Xilinx Vivado integrált HDL szimulátor

- A Vivado keretrendszerbe integrált modul (sok funkció, egyszerűség),
- Léteznek a *kereskedelmi forgalomban* kapható szimulátorok, pl. Mentor ModelSim (drága, de profi),
- Létezik néhány *ingyenes* HDL szimulációs eszköz is, pl. Icarus Verilog, Verilator, GHDL, stb.
- On-line szimulátorok, pl. :
  - <http://www.edaplayground.com>

# 1.a.) Új Vivado project létrehozása

- **File** menü → **New Project**
- Projekt neve: „myand2”
- Elérési út megadása. Majd **Next>**

New Project

**Project Name**  
Enter a name for your project and specify a directory where the project data files will be stored.

1 Project name: myand2

Project location: F:/TM\_2020\_Vivado\_2020.1/lab02

☒ Create project subdirectory

Project will be created at F:/TM\_2020\_Vivado\_2020.1/lab02/myand2

2

< Back Next > Finish Cancel

# 1.b.) Projekt létrehozása (folyt)

**1**

 New Project

**Project Type**  
Specify the type of project to create.

☒ **RTL Project**  
You will be able to add sources, create block designs in IP Integrator, generate IP, run RTL analysis, synthesis, implementation, design planning and analysis.

☐ Do not specify sources at this time

☐ **Post-synthesis Project:** You will be able to add sources, view device resources, run design analysis, planning and implementation.  
☐ Do not specify sources at this time

☐ **I/O Planning Project**  
Do not specify design sources. You will be able to view part/package resources.

☐ **Imported Project**  
Create a Vivado project from a Synplify, XST or ISE Project File.

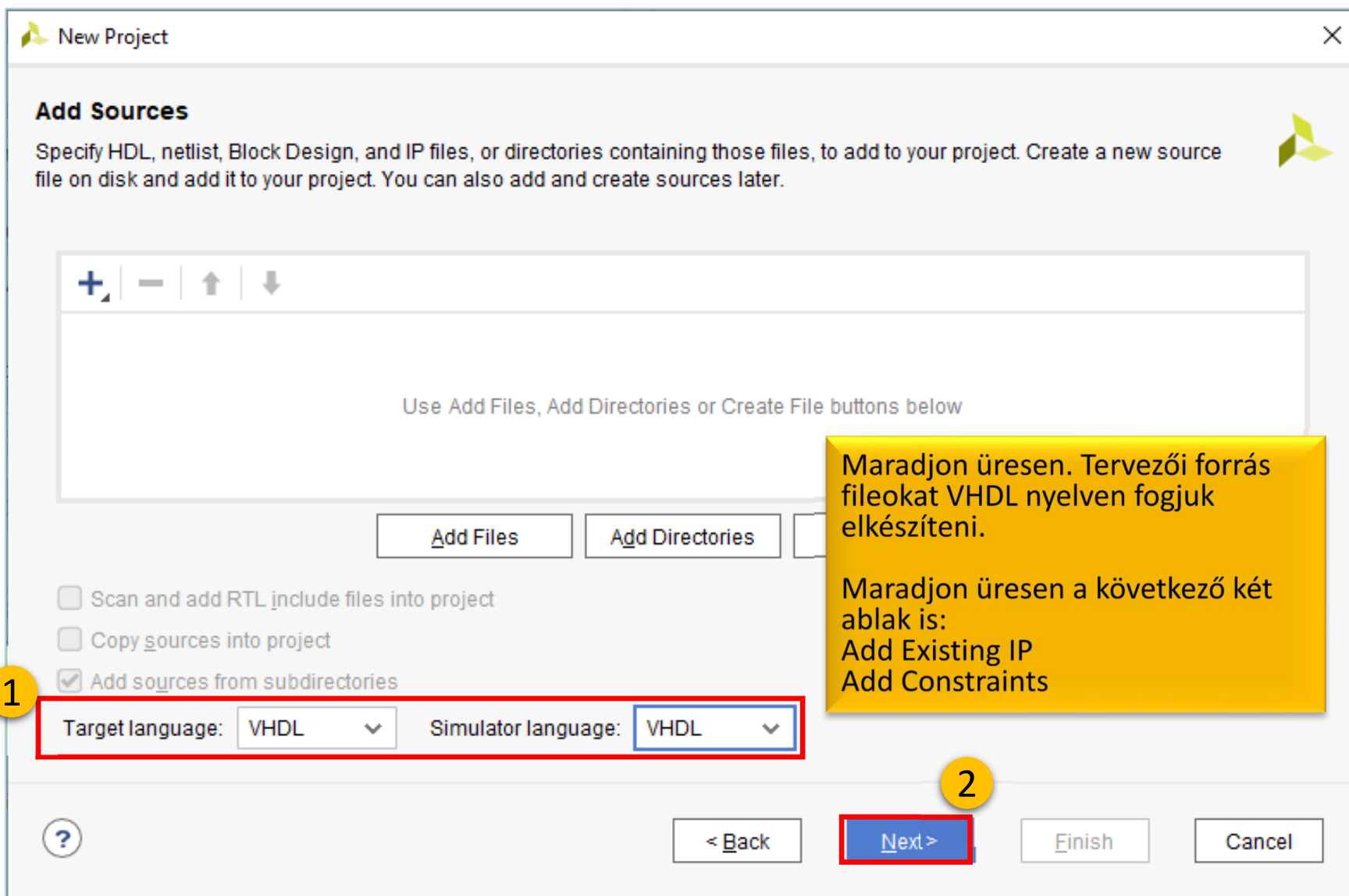
☐ **Example Project**  
Create a new Vivado project from a predefined template.

**2**

RTL Project kiválasztása:  
HDL fájlt akarunk szerkeszteni  
VHDL nyelven.



# 1.c.) Projekt beállításai



**New Project**

**Add Sources**

Specify HDL, netlist, Block Design, and IP files, or directories containing those files, to add to your project. Create a new source file on disk and add it to your project. You can also add and create sources later.

Use Add Files, Add Directories or Create File buttons below

☐ Scan and add RTL include files into project  
☐ Copy sources into project  
☒ Add sources from subdirectories

Target language: **VHDL** Simulator language: **VHDL**

? < Back **Next >** Finish Cancel

Maradjon üresen. Tervezői forrás fileokat VHDL nyelven fogjuk elkészíteni.

Maradjon üresen a következő két ablak is:  
Add Existing IP  
Add Constraints

# 1.d.) Projekt beállításai

## FPGA kártya kiválasztása

Fontos az **FPGA** kártya kiválasztása megadása!

\*Fontos: a Zybo kártya támogatásához a Digilent oldaláról a támogató csomagot le kell tölteni, és a megfelelő helyre másolni!

New Project

**Default Part**  
Choose a default Xilinx part or board for your project.

1 **Boards**

2 Vendor:  Name:  Board Rev:

3

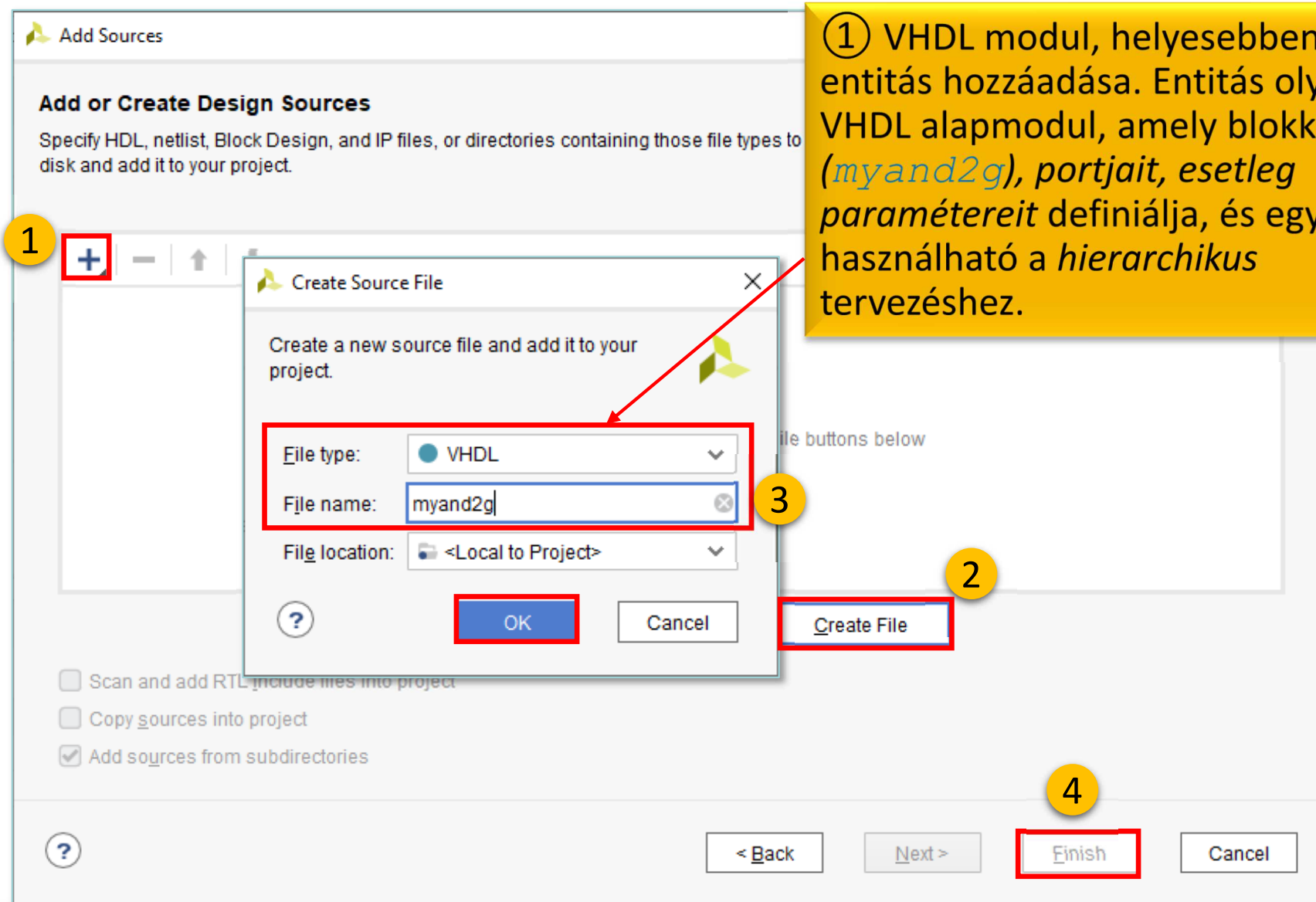
| Display Name | Preview | Vendor          | File Version | Part            |
|--------------|---------|-----------------|--------------|-----------------|
| Zybo         |         | digilentinc.com | 2.0          | xc7z010clg400-1 |

4 **Next >**

\* <https://reference.digilentinc.com/reference/software/vivado/board-files?redirect=1>

## 2.a.) HDL modul létrehozása

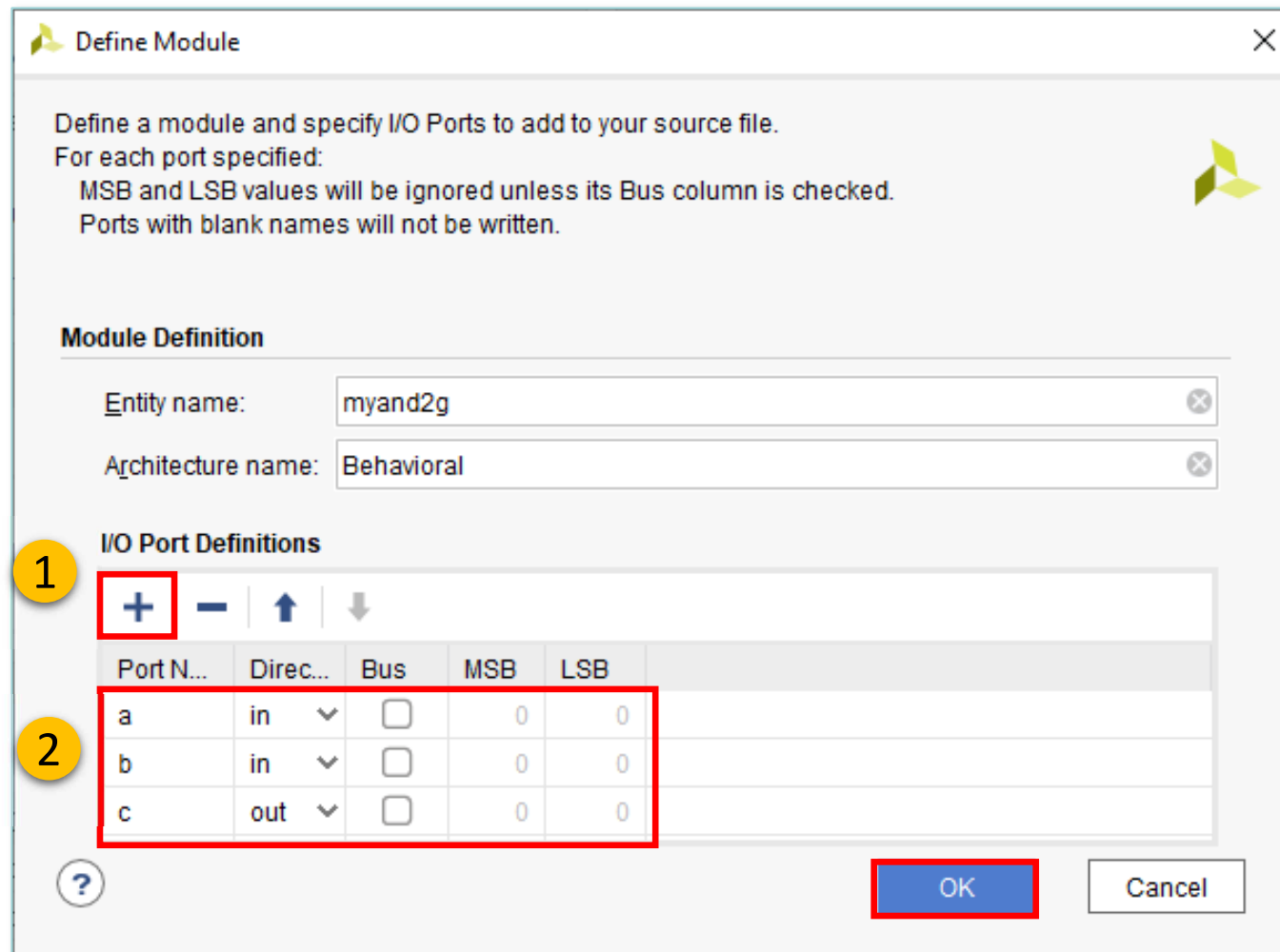
*File menü → Add Sources → Add or Create Design Sources*



## 2.b.) Portok megadása

Legyen két bemenete: (a, b) **input** irányultságú

Legyen egy kimenete: (c) **output** irányultságú



Define Module

Define a module and specify I/O Ports to add to your source file.  
For each port specified:  
MSB and LSB values will be ignored unless its Bus column is checked.  
Ports with blank names will not be written.

**Module Definition**

Entity name: myand2g

Architecture name: Behavioral

**I/O Port Definitions**

1. + - ↑ ↓

2.

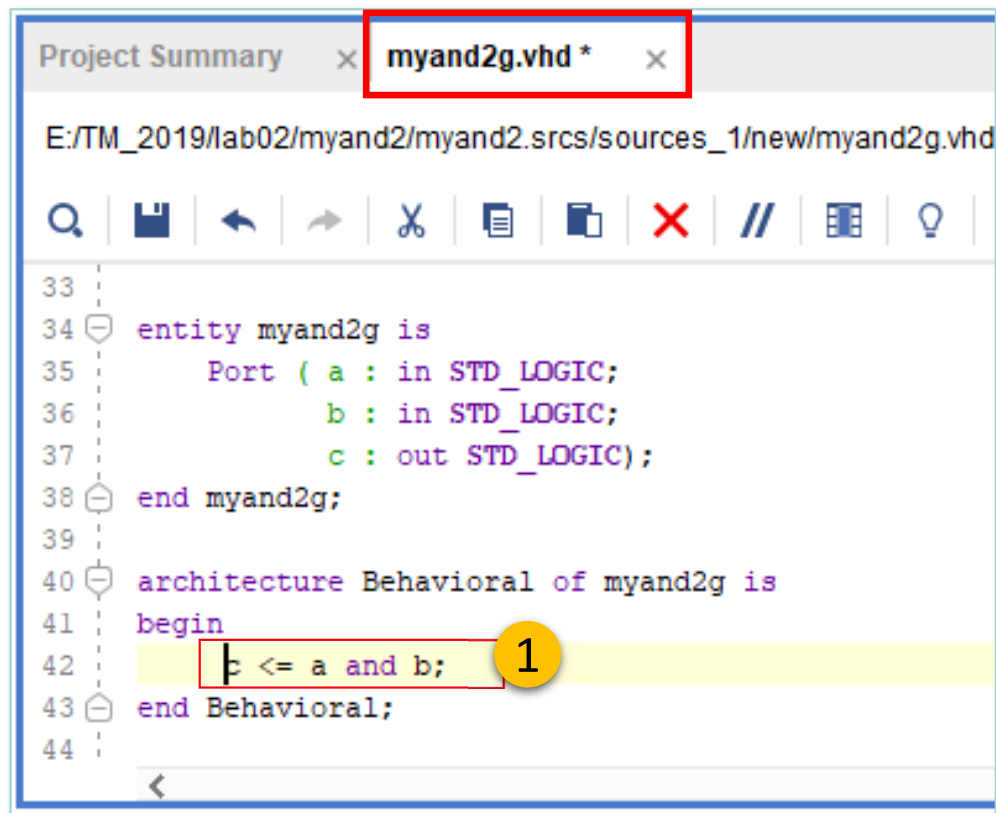
| Port N... | Direc... | Bus | MSB                      | LSB |
|-----------|----------|-----|--------------------------|-----|
| a         | in       | ▼   | <input type="checkbox"/> | 0 0 |
| b         | in       | ▼   | <input type="checkbox"/> | 0 0 |
| c         | out      | ▼   | <input type="checkbox"/> | 0 0 |

?

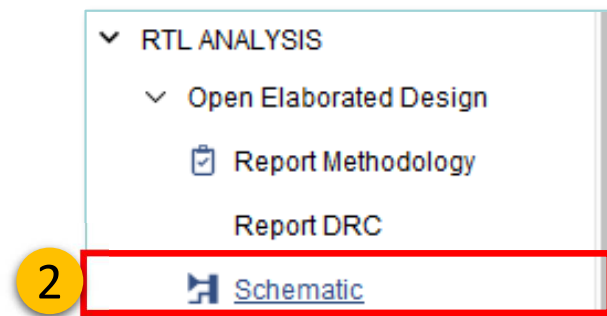
OK Cancel



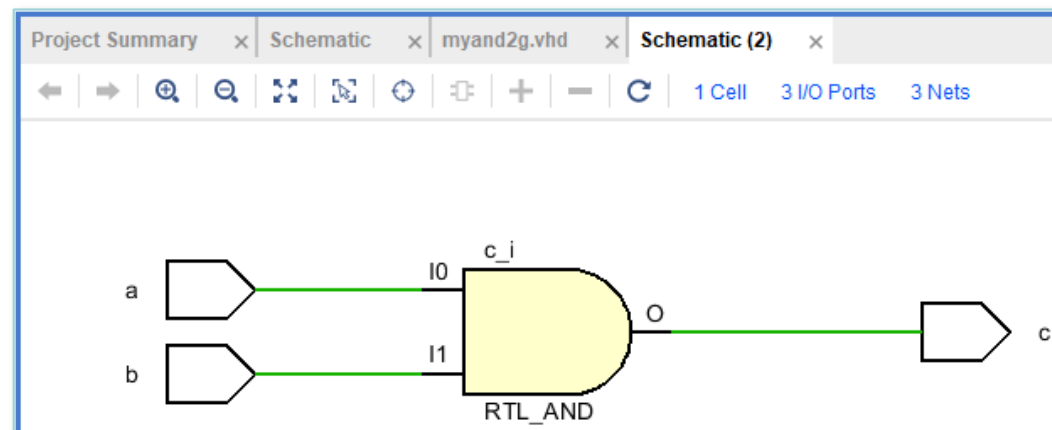
### 3.) VHDL forráskód megírása



```
33
34 entity myand2g is
35     Port ( a : in STD_LOGIC;
36           b : in STD_LOGIC;
37           c : out STD_LOGIC);
38 end myand2g;
39
40 architecture Behavioral of myand2g is
41 begin
42     c <= a and b;
43 end Behavioral;
44
```



- *CTRL+S*: VHDL forrásfile mentése (\*)
- *Open Elaborated design* → *Schematic*.

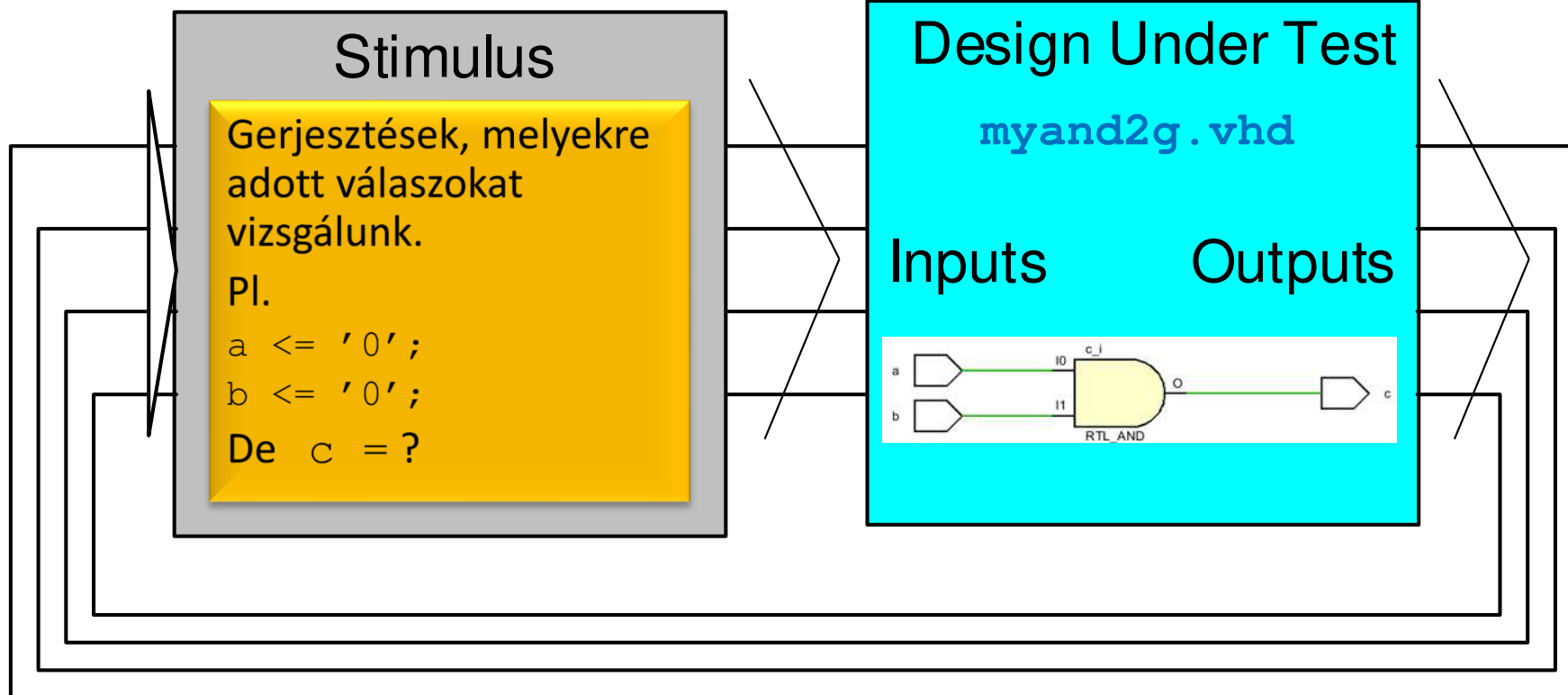


## 4.) TestBench: VHDL szimuláció

Testbench / TestPad

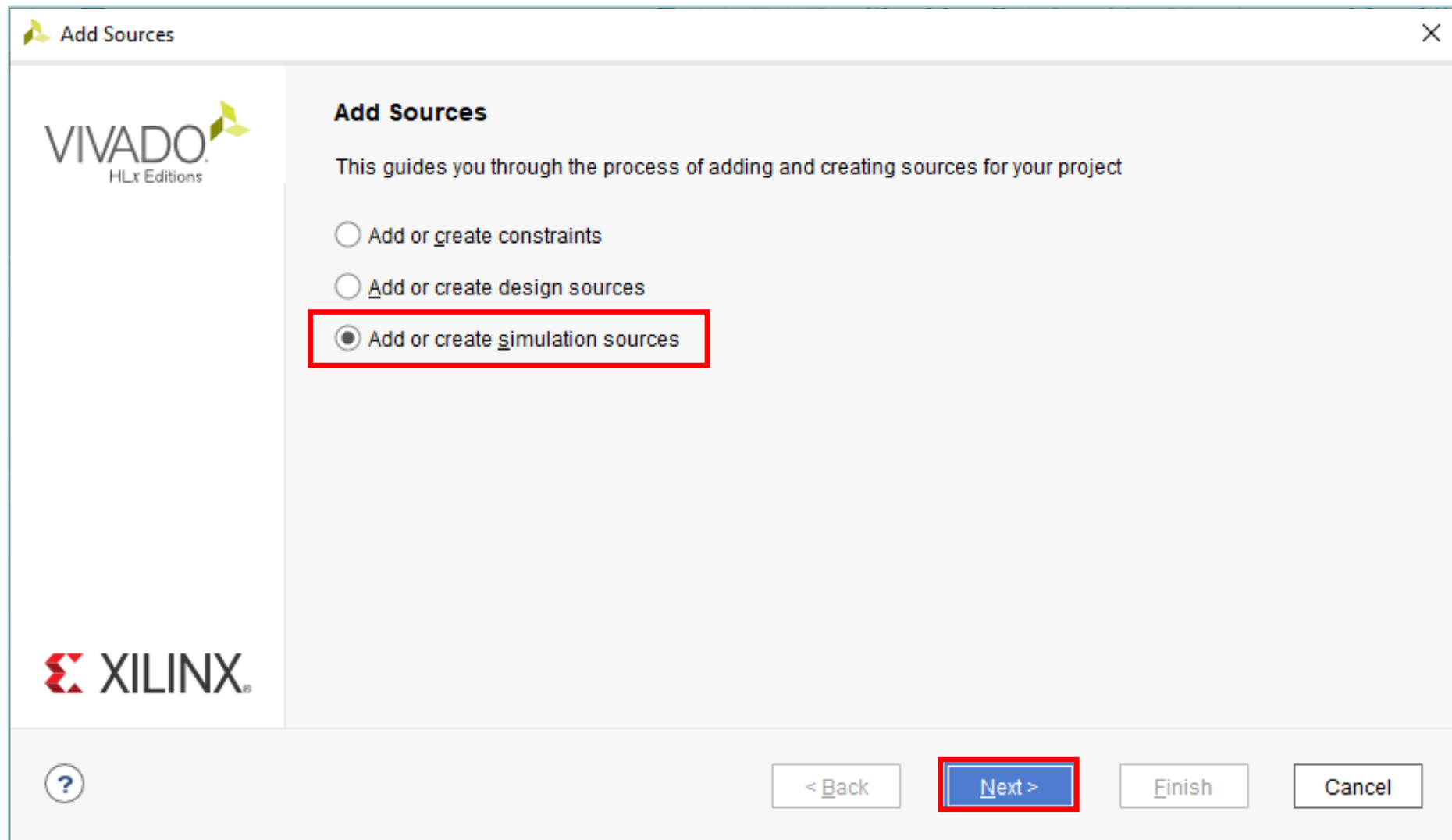
`myand2g_tb.vhd`

\*DUT = UUT (Design / Unit under Test)



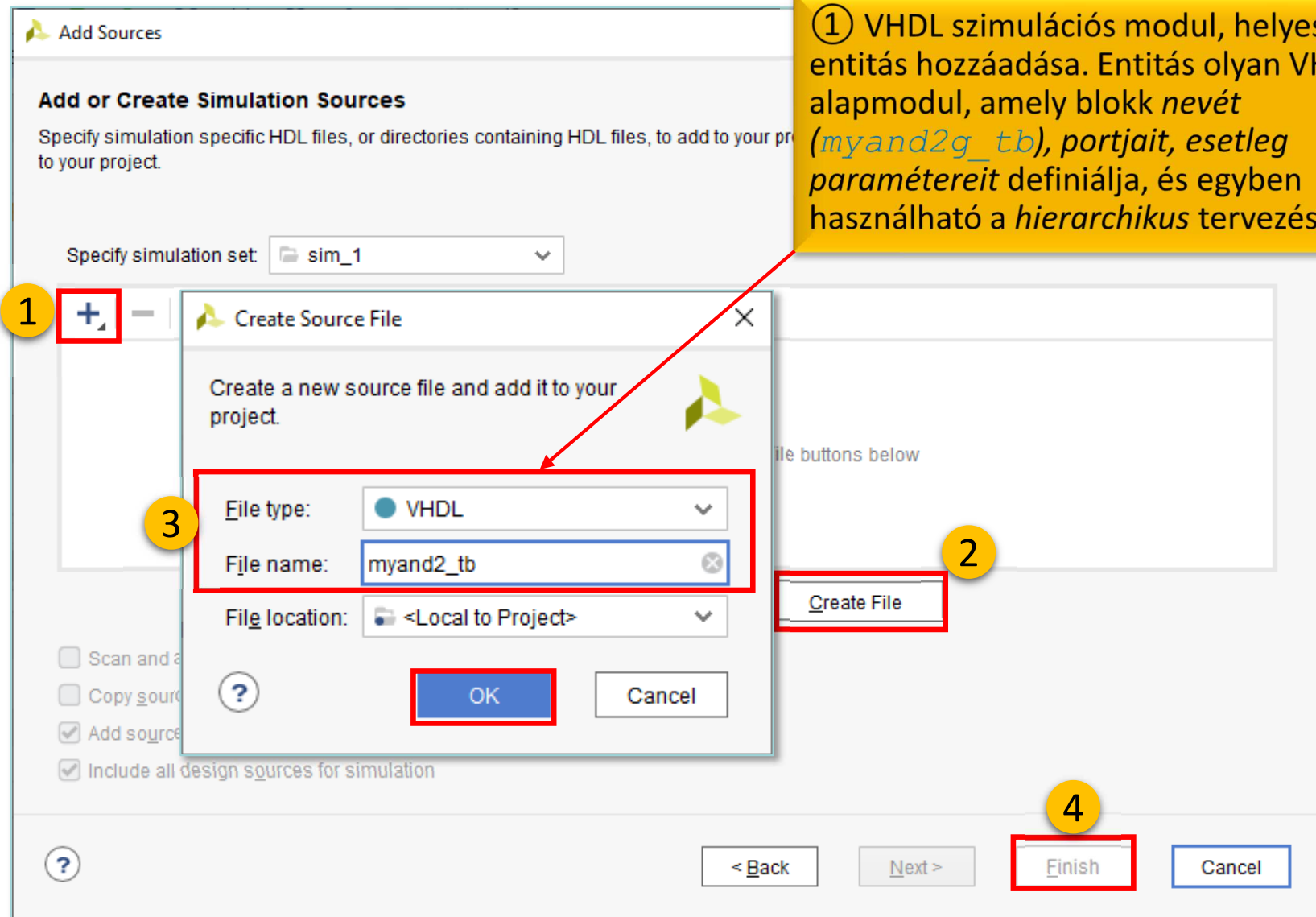
## 4.a.) HDL szimulációs modul létrehozása

*File menü → Add Sources → Add or Create Simulation Sources*



# 4.a.) HDL testbench létrehozása

*File menü → Add Sources → Add or Create Simulation Sources*



① VHDL szimulációs modul, helyesebben entitás hozzáadása. Entitás olyan VHDL alapmodul, amely blokk *nevét* (*myand2g\_tb*), *portjait*, *esetleg paramétereit* definiálja, és egyben használható a *hierarchikus* tervezéshez.



## 4.b.) HDL testbench (folyt)

Fontos: Vivado-ban a teljes test-bench automatikus generálása sajnos nem lehetséges. Ezért mi magunk írjuk meg a test-bench forrását is!

Define Module

Define a module and specify I/O Ports to add to your source file.  
For each port specified:  
MSB and LSB values will be ignored unless its Bus column is checked.  
Ports with blank names will not be written.

**Module Definition**

Entity name: myand2\_tb

Architecture name: Behavioral

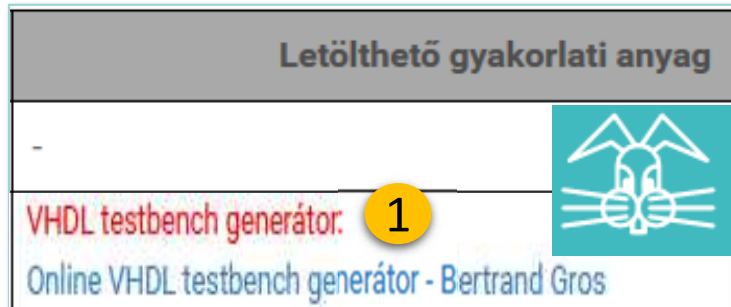
**I/O Port Definitions**

| Port Name | Direction | Bus                      | MSB | LSB |
|-----------|-----------|--------------------------|-----|-----|
| in        | in        | <input type="checkbox"/> | 0   | 0   |
|           |           |                          |     |     |
|           |           |                          |     |     |

OK Cancel

Az I/O port-ok definícióit szimulációs modul szerkesztésekor hagyjuk most üresen! Mivel a testbenchen belül generáljuk a gerjesztéseket, és vizsgáljuk az azokra adott válaszokat. OK.

## 4.c.) Testbench modul generálása



myand2.vhd VHDL forrásfile  
tartalmának bemásolása a szöveg  
dobozba → végül **Generate**



Online VHDL Testbench Template Generator

© 2011-2020 by Bertrand Gros

This tool automatically generates a template file for a testbench for the simulation of a VHDL entity.

A testbench is a VHDL code that simulates the environment around your DUT (design under test). The testbench generates stimuli to the inputs of the DUT and allows to check its functionality and outputs within a simulator. The declarative part of the testbench is quite boring to write, hence the existence of this automatic generator.

The generator is in constant development, but for now it seems to work pretty well for most standard source files, but it has not yet been extensively tested with a lot of different sources with different writing styles.

Feel free to test it with your files and to report problems to help to improve it.

This is for VHDL only, don't try to paste Verilog code, it will not work ;-)

Simply copy and paste your VHDL code below, then press the Generate button.

```
library IEEE; use IEEE.STD_LOGIC_1164.ALL; -- Uncomment the following library declaration if using -- arithmetic functions with Signed or Unsigned values --use IEEE
```

2

- ☐ Generate clock and try to automatically guess signal name
- ☐ Generate clock without guessing signal name
- ☒ No clock generation
- ☐ Generate reset (positive polarity)
- ☐ Generate reset (negative polarity)
- ☒ No reset generation

3

Generate

A megjelenő új ablak tartalmának (testbench) bemásolása a Vivado-ba, a myand2\_tb.vhd VHDL **testbench** üres forrásába.

Bertrand Gros: <https://vhdl.lapinoo.net/testbench/tb.php>

-- EDIT Adapt initialization as needed  
-- EDIT Adapt initialization as needed  
-- EDIT Adapt initialization as needed

## 4.d.) Testbench modul módosítása

ARCHITECTURE tb OF tb\_myand2 IS

-- Component Declaration for the  
Design Under Test (DUT) - myand2g

COMPONENT myand2g

PORT (

a : IN std\_logic;

b : IN std\_logic;

c : OUT std\_logic

);

END COMPONENT;

signal a : std\_logic;

signal b : std\_logic;

signal c : std\_logic;

BEGIN

<clock> részek később itt lesznek majd:  
egyelőre nem használunk órajelet a  
tesztbench-ben (K.H)

-- Instantiate the Design Under  
Test (DUT)

dut: myand2g PORT MAP (

a => a,

b => b,

c => c

);

-- Stimulus process

stimuli: process

begin

-- hold reset state for 100 ns.

wait for 100 ns;

-- insert stimulus here

a <= '1';

wait for 10 ns;

b <= '1';

wait for 5 ns;

a <= '0';

wait for 20 ns;

b <= '0';

wait for 10 ns;

wait;

end process;

END;

## 4.d.) Testbench modul **megírása**

**ARCHITECTURE** tb **OF** myand2\_tb **IS**

-- Component Declaration for the  
Design Under Test (DUT) - myand2g

**COMPONENT** myand2g

**PORT** (

a : **IN** std\_logic;

b : **IN** std\_logic;

c : **OUT** std\_logic

);

**END COMPONENT**;

**signal** a : std\_logic; --:= '0';

**signal** b : std\_logic; --:= '0';

**signal** c : std\_logic;

**BEGIN**

clock részek később itt lesznek majd: egyelőre  
nem használunk órajelet a testbench-ben

**FONTOS:** File végéről a „configuration” részek  
(4 utolsó sor) törölhető!

-- Instantiate the (DUT)

dut: myand2g

**PORT MAP** (

a => a,

b => b,

c => c );

-- Stimulus process

stim\_proc: **process**

**begin**

-- EDIT Adapt initialization as needed

a <= '0';

b <= '0';

**wait for** 10 ns;

-- EDIT Add stimuli here

a <= '1';

**wait for** 10 ns;

b <= '1';

**wait for** 5 ns;

a <= '0';

**wait for** 20 ns;

b <= '0';

**wait for** 10 ns;

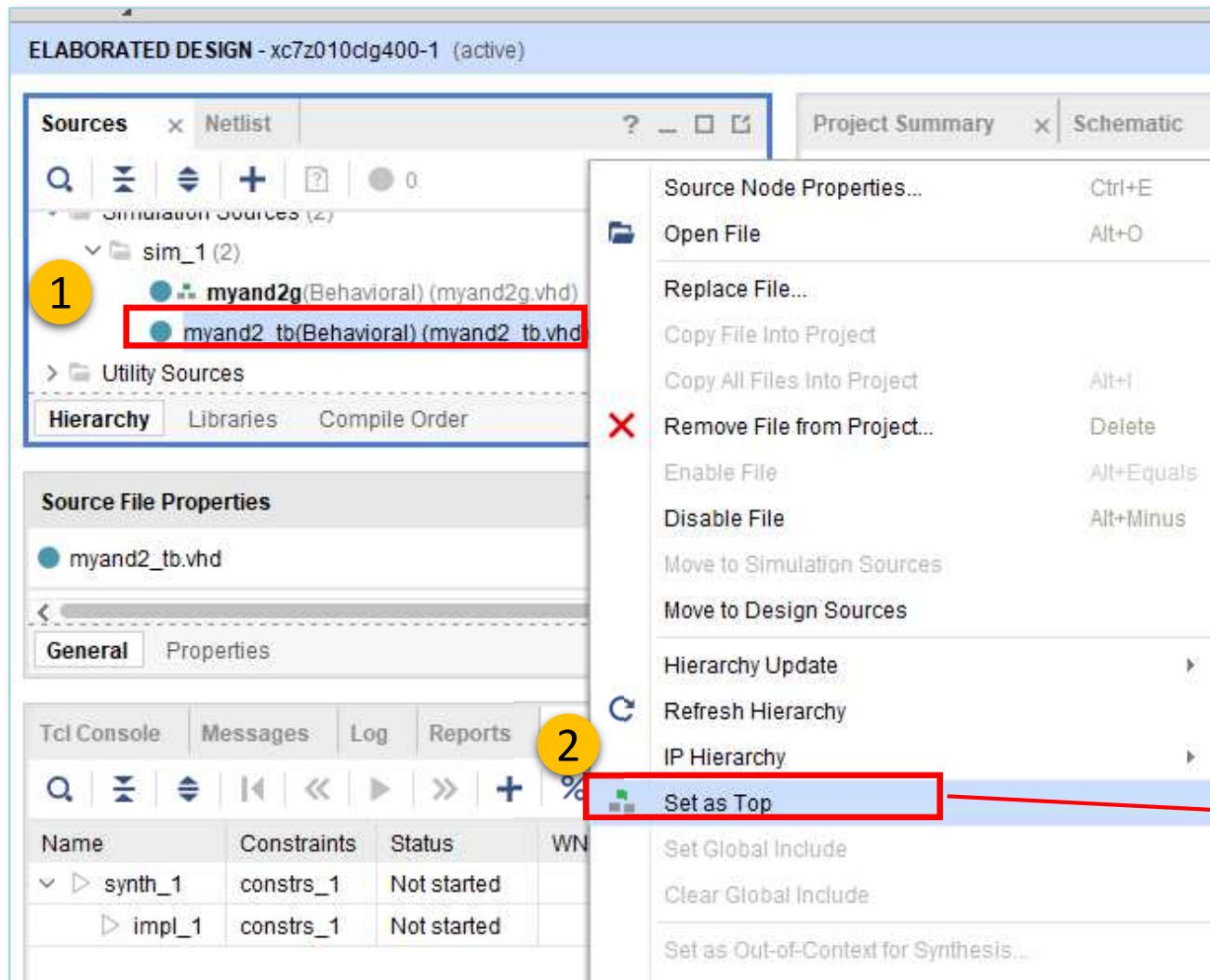
**wait**;

**end process**;

**END** tb;



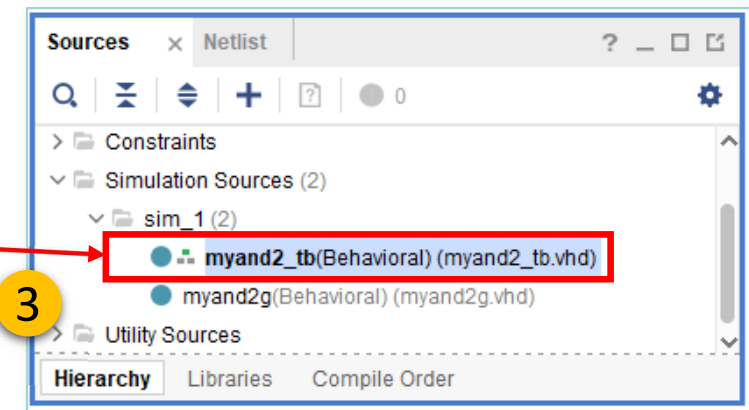
## 4.e.) Testbench: tesztkörnyezet összeállítása



1.) Sources ablakban a Hierarchy nézetben a `myand2_tb` szimulációs modul kiválasztásakor → **Jobb gomb** → **Set as Top.**

2.) Kijelöljük, hogy a testbench legyen a top-szintű modul.

3.) Mindig a test-benchen futtassuk a szimulációt!

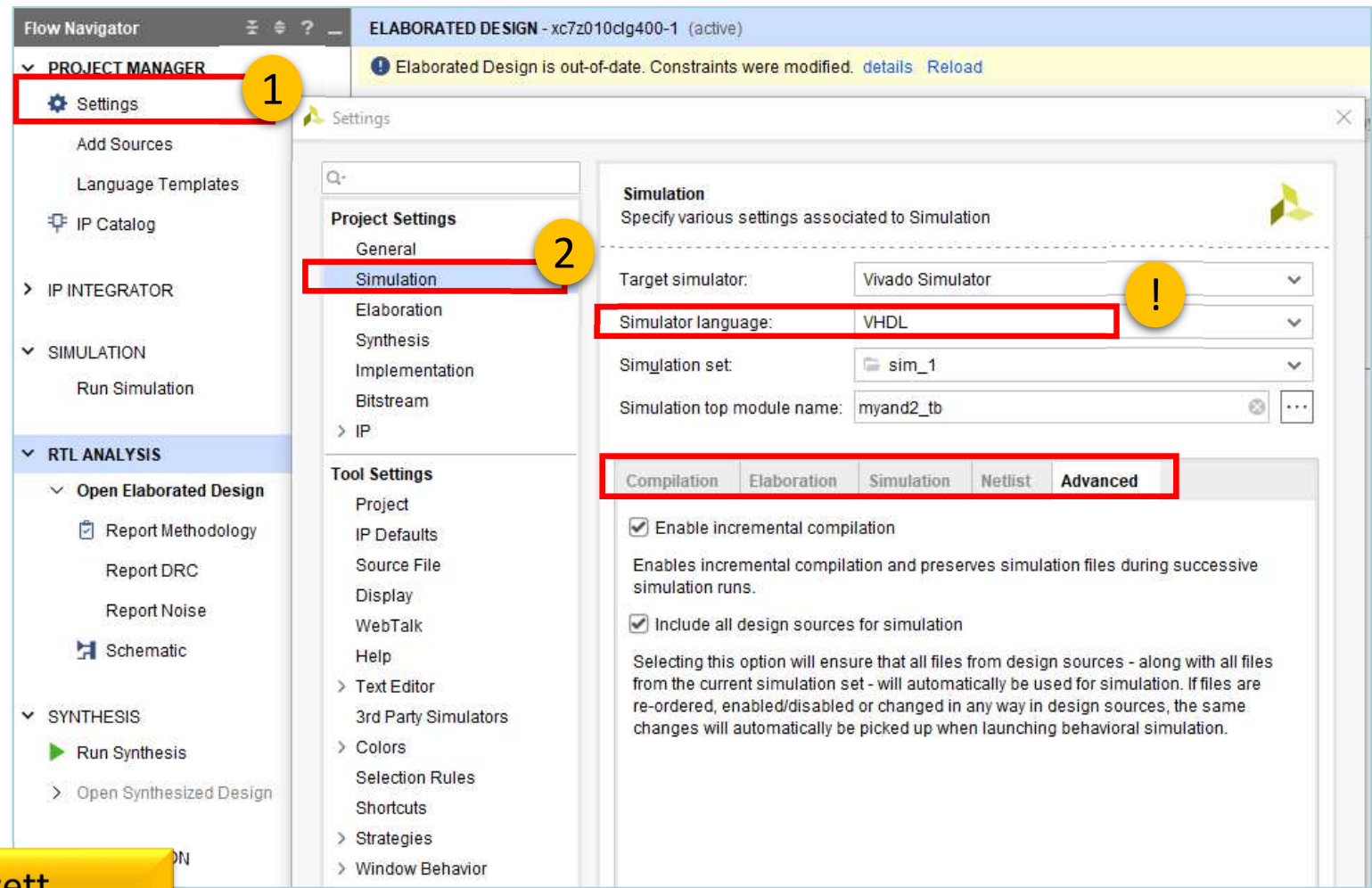


## 4.f.) Vivado szimulátor beállításai

- Fontos a Testbench kiválasztása

`myand2_tb`

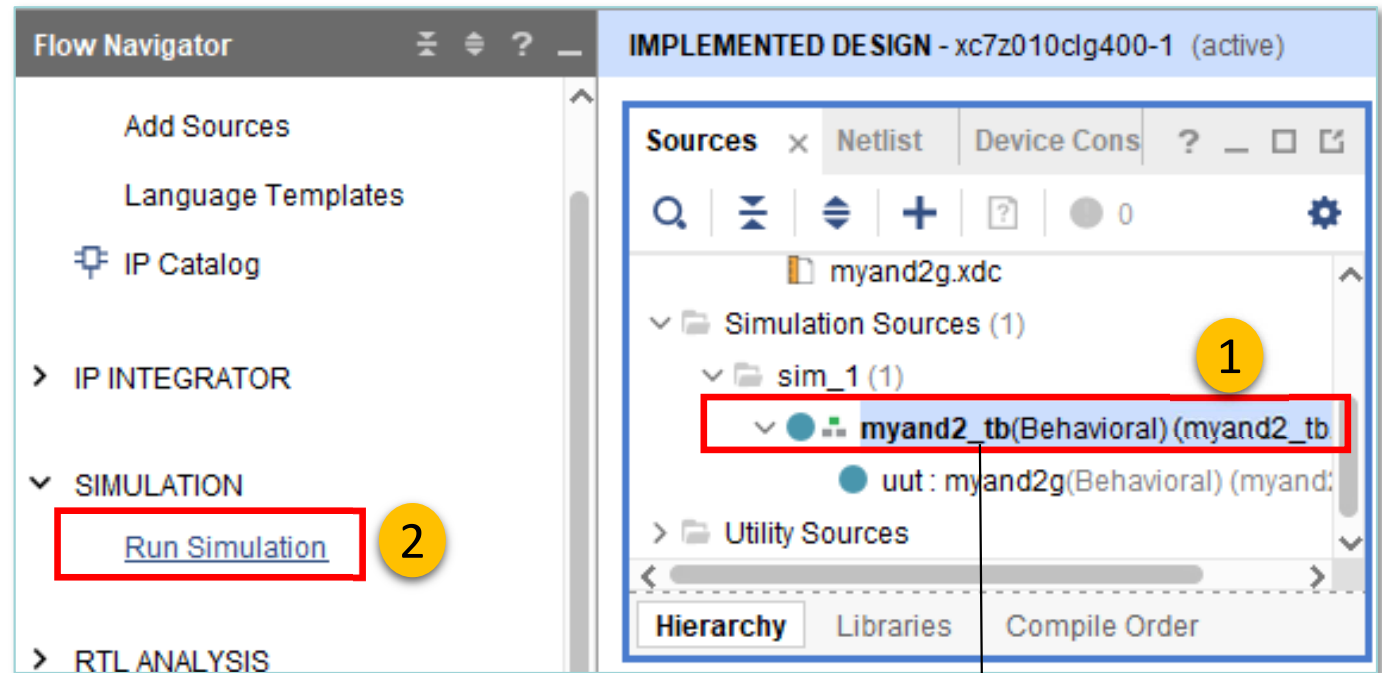
- Flow Navigator panel → „Settings”
  - Simulation



A szimulátor alapértelmezett időbeli felbontása 1 ps.

## 5.) Vivado szimulátor indítása

- 1.) Sources ablak – *myand2\_tb* kiválasztása
- 2.) Flow Navigator panel → Simulation
  - Run Simulation
    - Run behavioral simulation
  - Futtatáskor a WaveForm ablak jelenik meg, melyen a gerjesztéseknek (stimulus) megfelelő kimeneti jelek vizsgálhatóak.

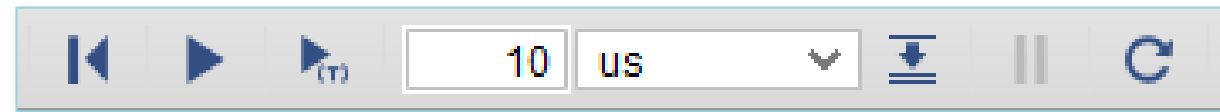


 **myand2\_tb**

Mindig a testbench legyen a top-szintű modul.  
Mindig a test-benchen futtassuk a szimulációt!



## 5.a.) Szimulátor GUI: WaveForm ablak



SIMULATION - Behavioral Simulation - Functional - sim\_1 - myand2\_tb

myand2\_tb

| Name | Value |
|------|-------|
| a    | 0     |
| b    | 0     |
| c    | 0     |

myand2\_tb

| Name | Value |
|------|-------|
| a    | 1     |
| b    | 0     |
| c    | 0     |

Waveform plot showing signals a, b, and c over time (80 ns to 160 ns). Signal a is high (1) from 100.000 ns to 120 ns. Signal b is high (1) from 100 ns to 140 ns. Signal c is high (1) from 100 ns to 110 ns.



# 6.) XDC - Felhasználói megkötések

1

Add Sources, "+" vagy CTRL+A.

2

Create Constraints File

Create a new constraints file and add it to your project

File type: XDC

File name: myand2g

File location: <Local to Project>

OK Cancel

myand2g.vhd

```
28
29 -- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;
33
34 entity myand2g is
35     Port (a : in STD_LOGIC;
```

| LUT | FF | BRAMs | URAM | DSP | Start | Elapsed | Run Strategy |
|-----|----|-------|------|-----|-------|---------|--------------|
|     |    |       |      |     |       |         | Vivado Synth |
|     |    |       |      |     |       |         | Vivado Imple |

Constraint Set: constrs\_1

# 6.a.) XDC - Lábak hozzárendelése

The screenshot shows the Vivado IDE interface. On the left, the 'Flow Navigator' pane is open, showing the 'PROJECT MANAGER' section. Under 'Sources', the 'myand2g' project is listed. In the 'Constraints' section, the 'constrs\_1' folder is expanded, and the 'myand2g.xdc' file is highlighted. A yellow circle with the number '1' is placed next to this file. The main editor window shows the 'myand2g.xdc' file content. A red box highlights the code, and a yellow circle with the number '2' is placed next to it. The code defines properties for switches and LEDs. A yellow callout box contains text about copying SW0/SW1 and LED0 parts. At the bottom right, there is a table with download links for Digilent ZyBo boards.

1

2

SW0/SW1 és a LED0 részek bemásolása a létrehozott `myand2.xdc` -be. ZYBO Master.xdc alapján (forrása a tárgy oldalán)

| Letölthető gyakorlati anyag                   |  |  |  |  |  |  |  |  |  |
|-----------------------------------------------|--|--|--|--|--|--|--|--|--|
| Run Strategy                                  |  |  |  |  |  |  |  |  |  |
| Vivado Synth                                  |  |  |  |  |  |  |  |  |  |
| Vivado Imple                                  |  |  |  |  |  |  |  |  |  |
| -                                             |  |  |  |  |  |  |  |  |  |
| Digilent ZyBo Fejlesztőkártyákhoz (BSP, XDC): |  |  |  |  |  |  |  |  |  |
| XDC (FPGA lábkiosztás - GIT master):          |  |  |  |  |  |  |  |  |  |
| <a href="#">Zybo-Master.xdc</a>               |  |  |  |  |  |  |  |  |  |

62

## 6.b.) Lábak hozzárendelése: **myand2g.xdc**

- Master XDC fileok megnyitása (ZyBo platformra):

- [https://github.com/Digilent/ZYBO/blob/master/Resources/XDC/ZYBO\\_Master.xdc](https://github.com/Digilent/ZYBO/blob/master/Resources/XDC/ZYBO_Master.xdc)

#LEDs és #Switches szekciók átmásolása a **myand2g.xdc** fájlba:

**##Switches**

```
set_property -dict { PACKAGE_PIN G15    IOSTANDARD LVCMOS33 } [get_ports { a }];  
    #IO_L19N_T3_VREF_35 Sch=SW0  
set_property -dict { PACKAGE_PIN P15    IOSTANDARD LVCMOS33 } [get_ports { b }];  
    #IO_L24P_T3_34 Sch=SW1
```

**##LEDs**

```
set_property -dict { PACKAGE_PIN M14    IOSTANDARD LVCMOS33 } [get_ports { c }];  
    #IO_L23P_T3_35 Sch=LED0
```

„sw<0>” átnevezése → „a”,

„sw<1>” → „b”.

„led<0>” átnevezése → „c”,

Kommentek # eltávolítása minden sorból! //

Végül: **myand2g.xdc** mentése.



# 7.) Implementáció futtatása

The screenshot displays the Xilinx Vivado IDE interface. The 'Flow Navigator' on the left shows the 'RTL ANALYSIS' tab selected, with 'Run Implementation' highlighted under the 'IMPLEMENTATION' section. A yellow banner at the top indicates 'Elaborated Design is out-of-date. Constraints were modified. details Reload'. The central pane shows the 'myand2g.vhd' source file with the following VHDL code:

```
32 --use UNISIM.VComponents.all;
33
34 entity myand2g is
35     Port ( a : in STD_LOGIC;
36           b : in STD_LOGIC;
37           c : out STD_LOGIC);
38 end myand2g;
39
40 architecture Behavioral of myand2g is
41 begin
42     c <= a and b;
43 end Behavioral;
44
```

The bottom pane shows the 'Design Runs' table:

| Name    | Constraints | Status      | WNS | TNS | WHS | THS | TPWS | Total Power | Failed Routes | LUT | FF | BRAMs | URAM | DSP | Start | Elapsed | Run Strategy |
|---------|-------------|-------------|-----|-----|-----|-----|------|-------------|---------------|-----|----|-------|------|-----|-------|---------|--------------|
| synth_1 | constrs_1   | Not started |     |     |     |     |      |             |               |     |    |       |      |     |       |         | Vivado Synth |
| impl_1  | constrs_1   | Not started |     |     |     |     |      |             |               |     |    |       |      |     |       |         | Vivado Imple |



## 7.a.) Implement Design

- *Reports → Place Design → place\_report\_utilization*

- *Kötött – Bounded IOB-k ellenőrzése. Miért 3?*

| Site Type       | Used | Fixed | Available | Util% |
|-----------------|------|-------|-----------|-------|
| Bonded IOB      | 3    | 3     | 100       | 3.00  |
| IOB Master Pads | 2    |       |           |       |
| IOB Slave Pads  | 1    |       |           |       |
| ...             |      |       |           |       |

Példák

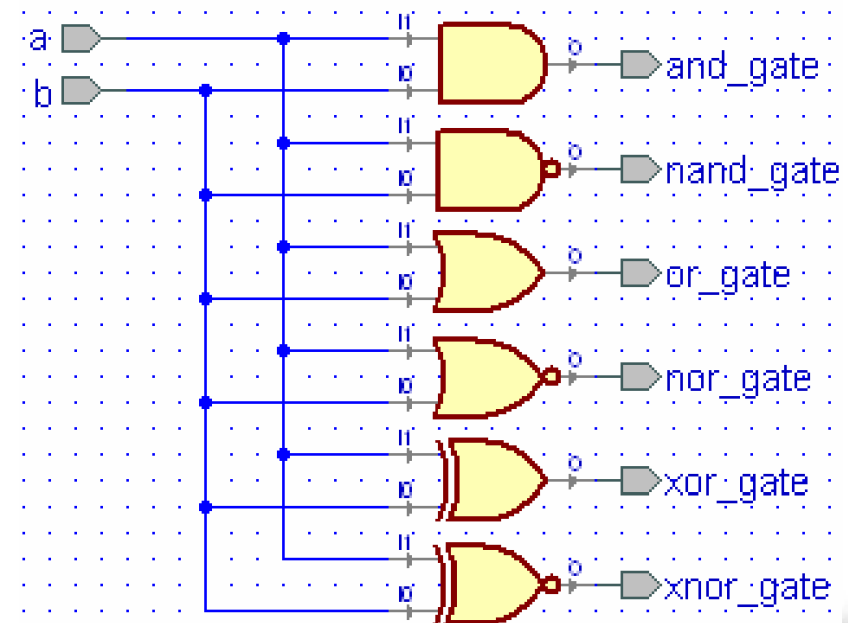
# **TOVÁBBI FELADATOK**

# Feladat 1.): Teljes „Design Flow”

- Switch/Button - Led feladat implementálása FPGA-n
  - SW(3:0) összekapcsolása LED(3:0)-el, vagy
  - BTN (3:0) összekapcsolása LED(3:0)-el.
  - Elaborate -> Synthesis -> Implementation -> Generate bitstream lépések végrehajtása,
  - VHDL Testbench készítése („`sw4toled4_tb`” néven) és a .vhd forrásfile viselkedési szimulációja Vivado segítségével,
  - .bit (bitstream) letöltése a Zynq7010 APSoC-ra (ZyBo kártya) az Vivado Hardware Manager segítségével.
  - Tesztelés és ellenőrzés.

## Feladat 2.)

- A lenti képen látható kombinációs logikai hálózat megtervezése és implementálása VHDL-ben („`multi_gates.vhd`” néven), amely a következő kapukat és összeköttetéseket tartalmazza:
  - SW(1:0) port -> ‘a’ és ‘b’ bemenetekre
  - LED(3:0) port -> az egyes ‘<xyz>\_gate(s)’ kezelésére (Te választod meg, hogy a lehetséges 6 kimenet közül, melyik négyet kötöd ki a LED-ekre!)
  - Elaborate -> Synthesis -> Implementation -> Generate bitstream
  - VHDL Testbench készítése („`multi_gates_tb.vhd`” néven) és a .vhd forrásfile viselkedési szimulációja Vivado XSim segítségével
  - .bit (bitstream) letöltése a Zynq-7010 APSoC-re (ZyBo kártya) a Vivado HW Manager segítségével.
  - Ellenőrzés (FPGA).





# Feladat 2.) VHDL megoldása

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity multi_gates is
port (
    a : in STD_LOGIC;
    b : in STD_LOGIC;
    and_gate : out STD_LOGIC;
    nand_gate : out STD_LOGIC;
    or_gate : out STD_LOGIC;
    nor_gate : out STD_LOGIC;
    xor_gate : out STD_LOGIC;
    xnor_gate : out STD_LOGIC
);
end multi_gates;

architecture Behav of multi_gates is
begin
    and_gate <= a and b;
    nand_gate <= a nand b;
    or_gate <= a or b;
    nor_gate <= a nor b;
    xor_gate <= a xor b;
    xnor_gate <= a xnor b;
end Behav;
```

Két-bemenetű logikai kapuk:  
foglalt szavak a VHDL-ben.