lab 01

Introduction: language VHDL and Xilinx Vivado

Digital design – laboratory exercises

assistant: Nejc Ilc

Overview

- digital design using VHDL language
- interaction with peripheral devices (momentary switches, buttons, LED, 7-segment display, …)
- integration of PicoBlaze CPU
- implementation of the UART controller
- project work

- Week challenges
  - we give instructions at labs
  - you submit the solution via Moodle

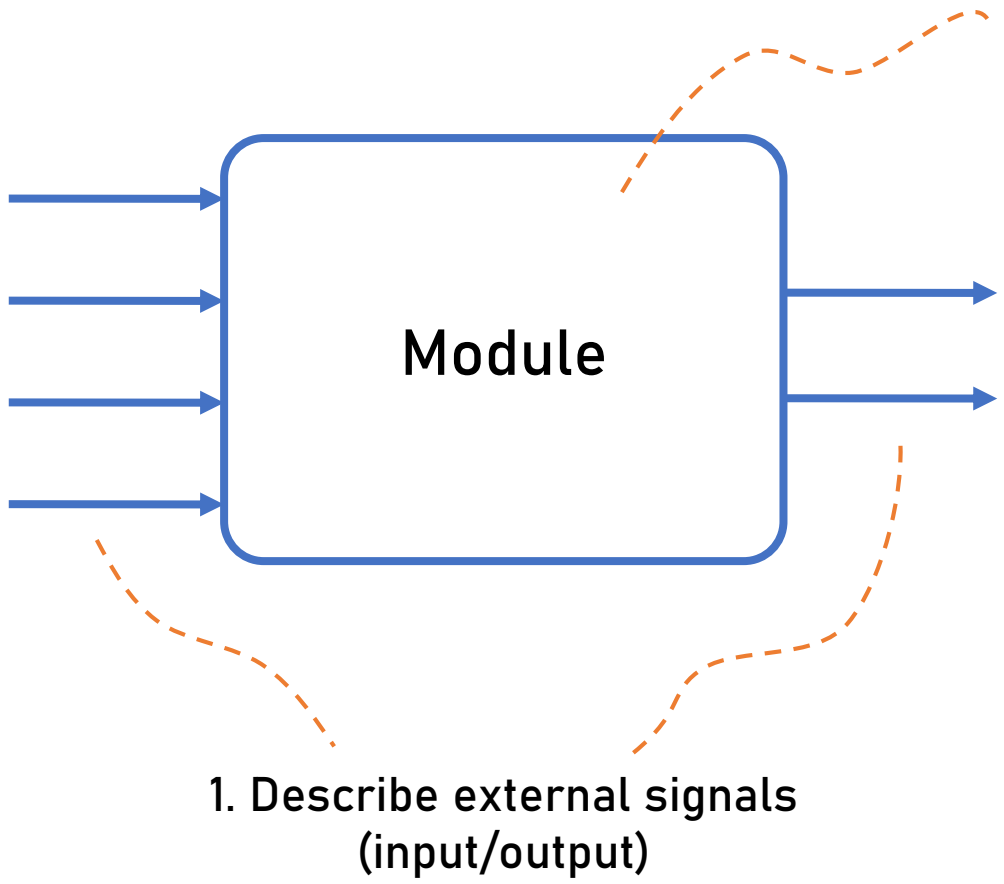
VHDL

- VHDL = VHSIC Hardware Description Language
- VHSIC = very-high-speed integrated circuits
- High-level language for the description and simulation of digital circuits
- The VHDL description will be synthesized and used to program an FPGA

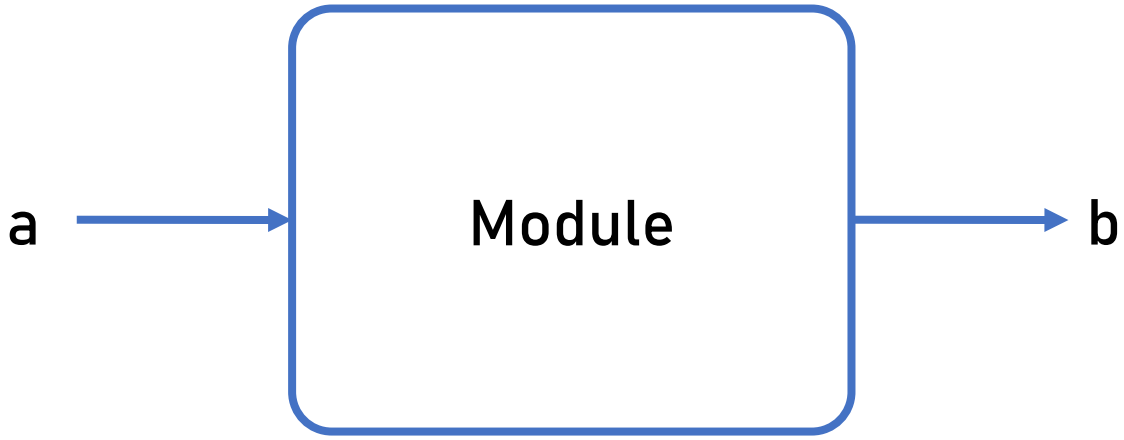
Tools

- Development with Xilinx Vivado ML Standard Edition (free)
- Boards Digilent Nexys
  - Nexys4
  - Nexys4 DDR
  - Nexys A7 50T in 100T

Description of a module



Description of external signals

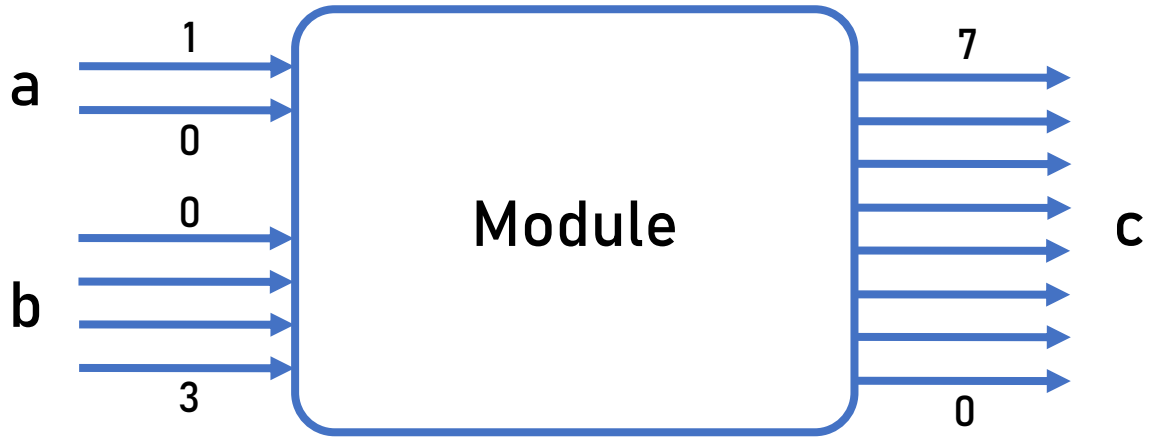



```
entity Module is
    port(
        a: in  std_logic;
        b: out std_logic
    );
end Module;
```

```
entity module_name is
 port(
  signal_name_1 : direction type;
  signal_name_2 : direction type;
  ...
  signal_name_n : direction type;
 );
end module_name;
```

- Direction:
  in, out, inout
- Type:
  std_logic, std_logic_vector()

Description of external signals: example




```
entity Module is
    port(
        a: in  std_logic_vector(1 downto 0);
        b: in  std_logic_vector(0 to 3);
        c: out std_logic_vector(7 downto 0)
    );
end Module;
```

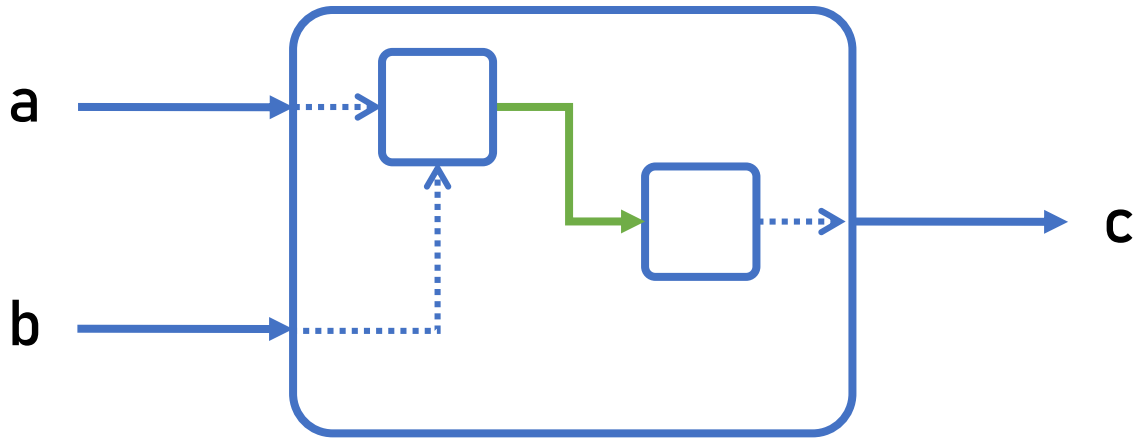
Description of a module behaviour

```
architecture behaviour_name of module_name is
  -- declaration of internal signals

begin
  -- statements

end behaviour_name;
```

Declaration of internal signals




```
architecture behaviour_name of module_name is
    signal signal_name: signal_type;
begin
...
```

Signal assignment

- Syntax

```
signal <= expression;
```

- Examples

  Assign a constant:

```
a <= '0';
```

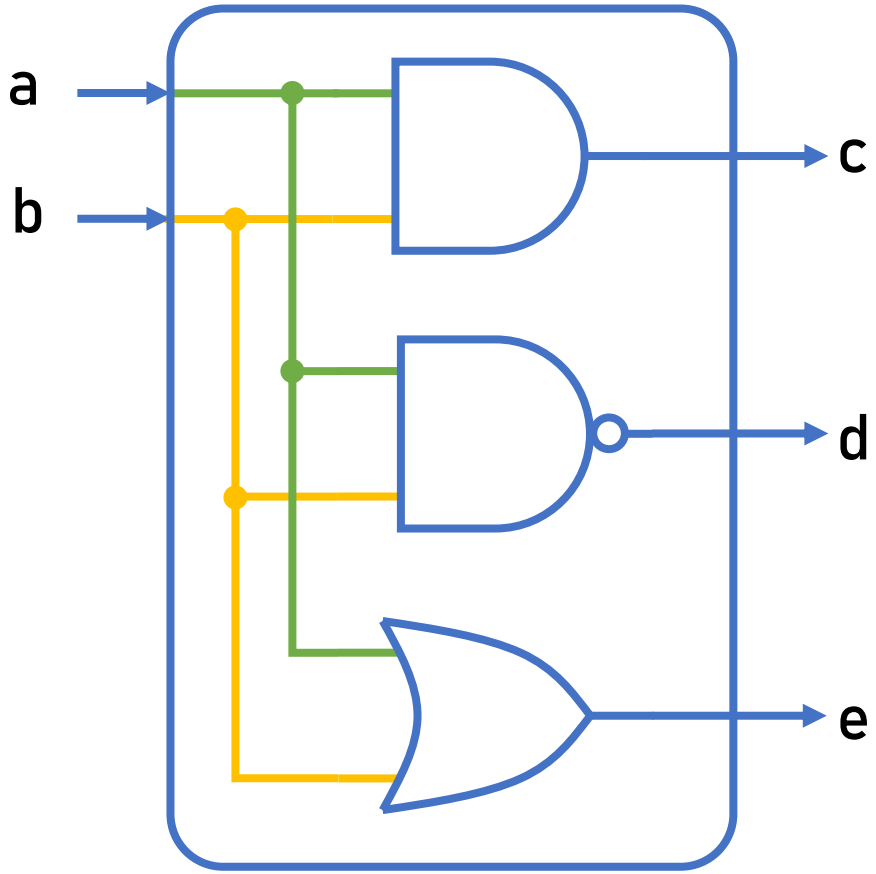
  Assign a constant to a vector:

```
b <= "01001";
```

  Assignment to the portion of a vector:

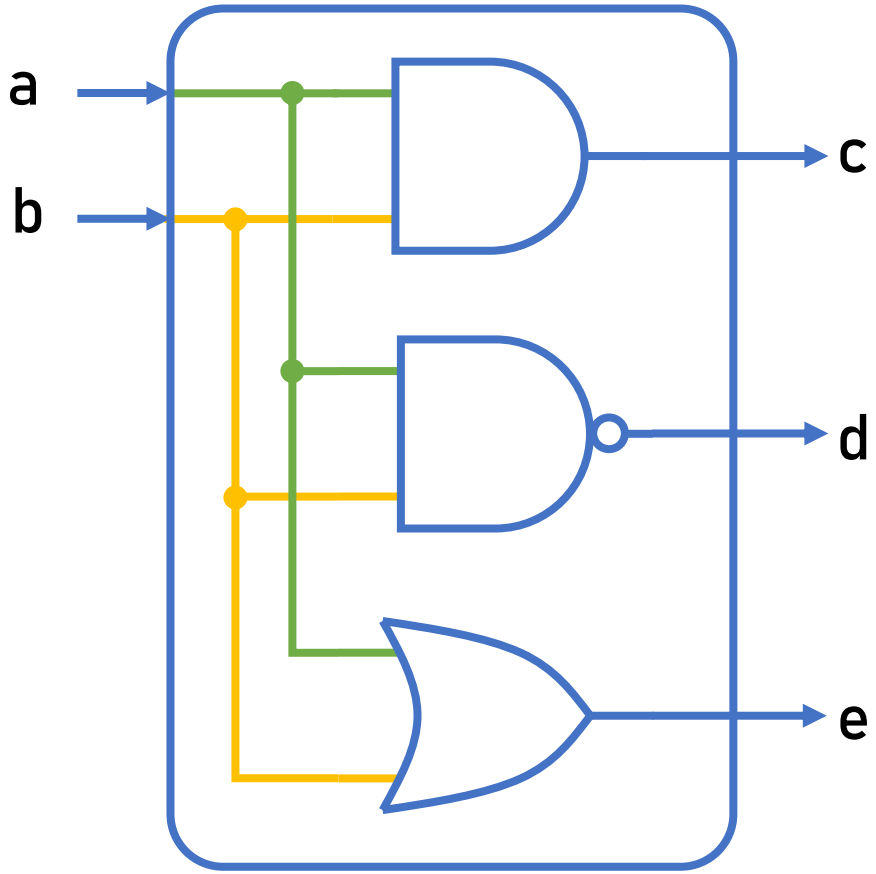
```
c(3 downto 0) <= "0111";
```

Signal assignment: example




```
entity gates is
  port(
     a: in  std_logic;
     b: in  std_logic;
     c: out std_logic;
     d: out std_logic;
     e: out std_logic
  );
end gates;
```

Signal assignment: example




```
architecture arch of gates is
begin
   -- basic logical operators:
   -- and, nand, or, nor, xor,
   -- xnor, not
   c <= a and  b;
   d <= a nand b;
   e <= a or   b;
end arch;
```

Conditional signal assignment

- Syntax

```
signal <= expression_1 when condition_1 else expression_2;

signal <= expression_1 when condition_1 else
          expression_2 when condition_2 else
          expression_3;
```

- Comparison operators:
  - equality, inequality `=, /=`
  - greater than, less than, … `>, <, >=, <=`

"select" statement

- Syntax

```
with selector select
    output <= expression_1 when value_1,
              expression_2 when value_2,
              ...
              expression_dfl;
```

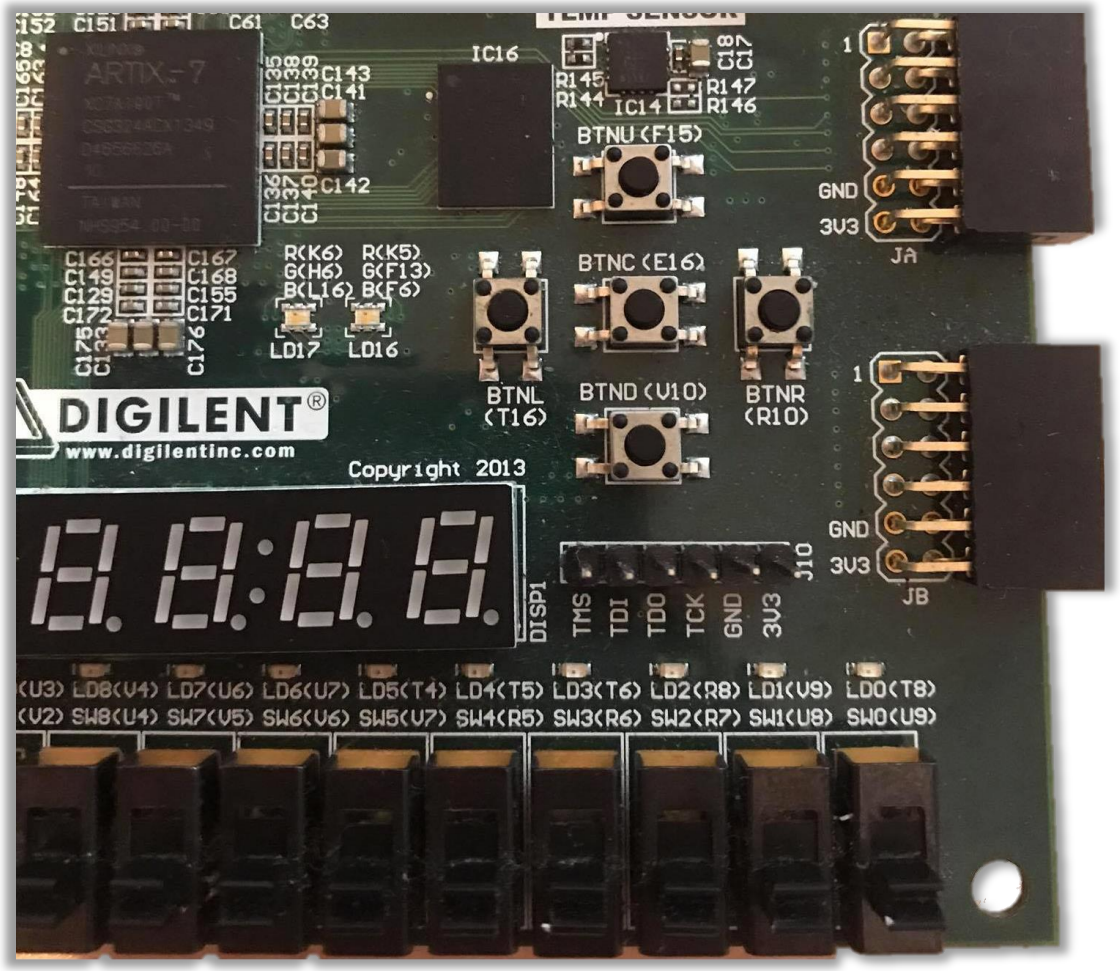
Defining constraints: mapping of external signals to pins

- We need to define physical pins on which external signals of our design are mapped.
- Example: let a and b be switches, c, d, and e LEDs
- Definitions go to XDC file (Xilinx Design Constraints)
- Syntax

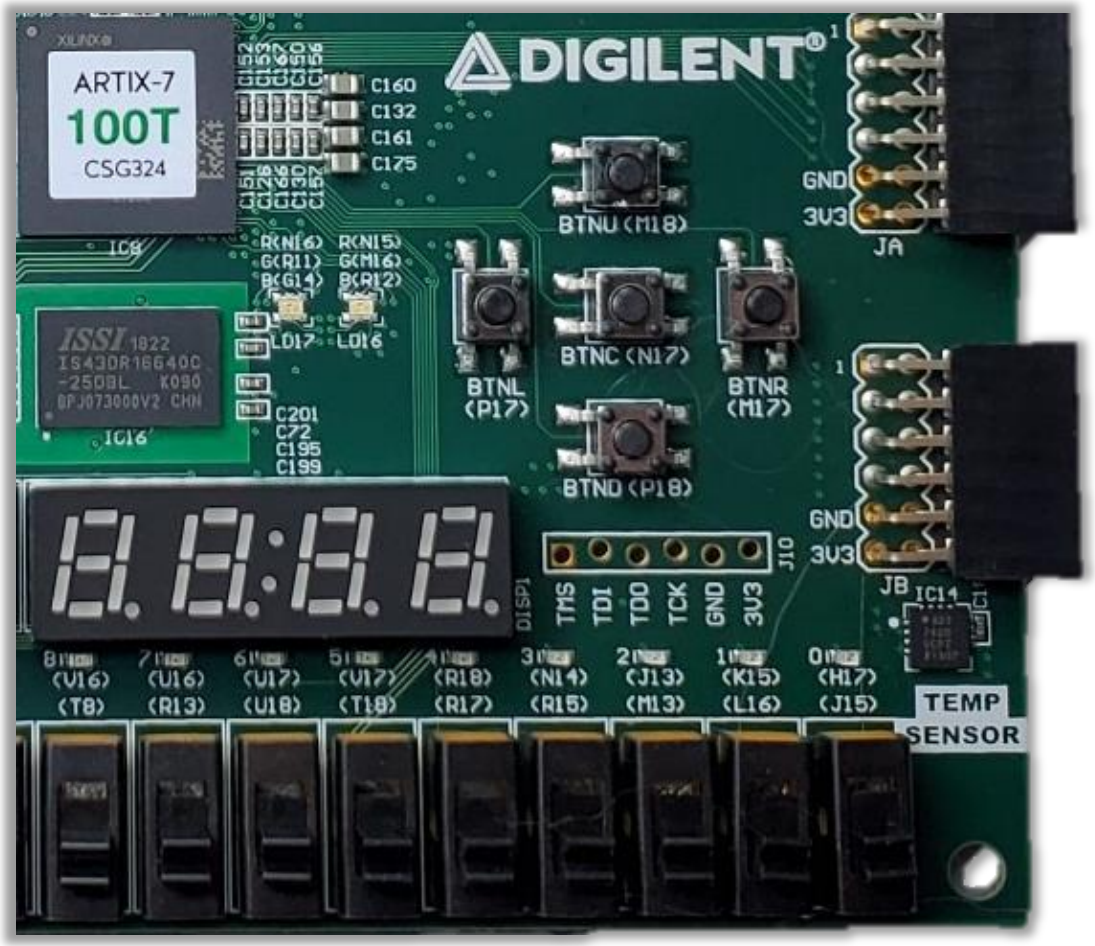
```
set_property -dict {PACKAGE_PIN J15 IOSTANDARD LVCMOS33} [get_ports {a}];
# vektor
set_property -dict {PACKAGE_PIN R18 IOSTANDARD LVCMOS33} [get_ports {c[0]}];
set_property -dict {PACKAGE_PIN V17 IOSTANDARD LVCMOS33} [get_ports {c[1]}];
```

- Oznako pina najdete v dokumentaciji razvojne plošče ali neposredno na plošči.

Pin labels



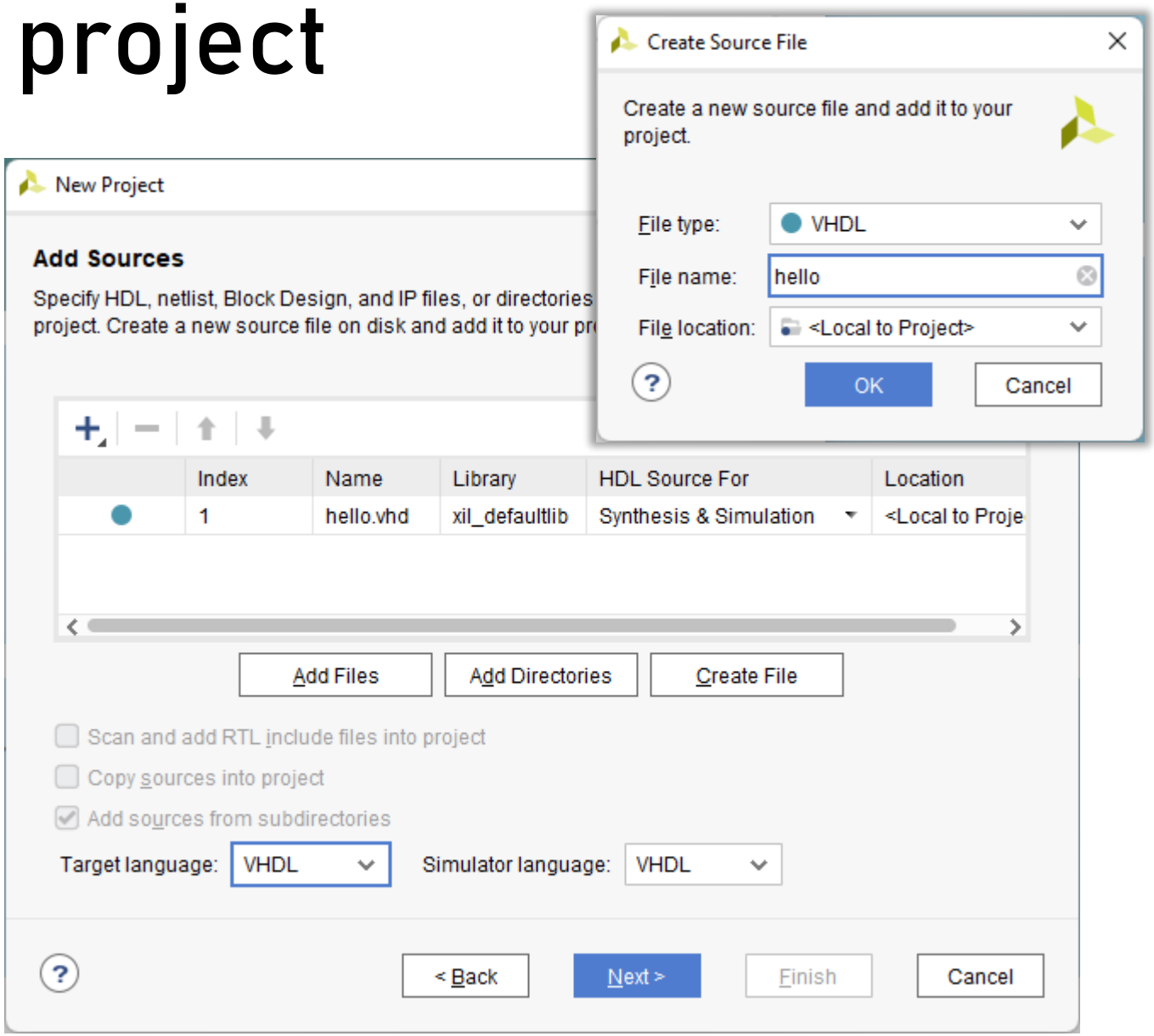
Nexys4



Nexys4 DDR / Nexys A7

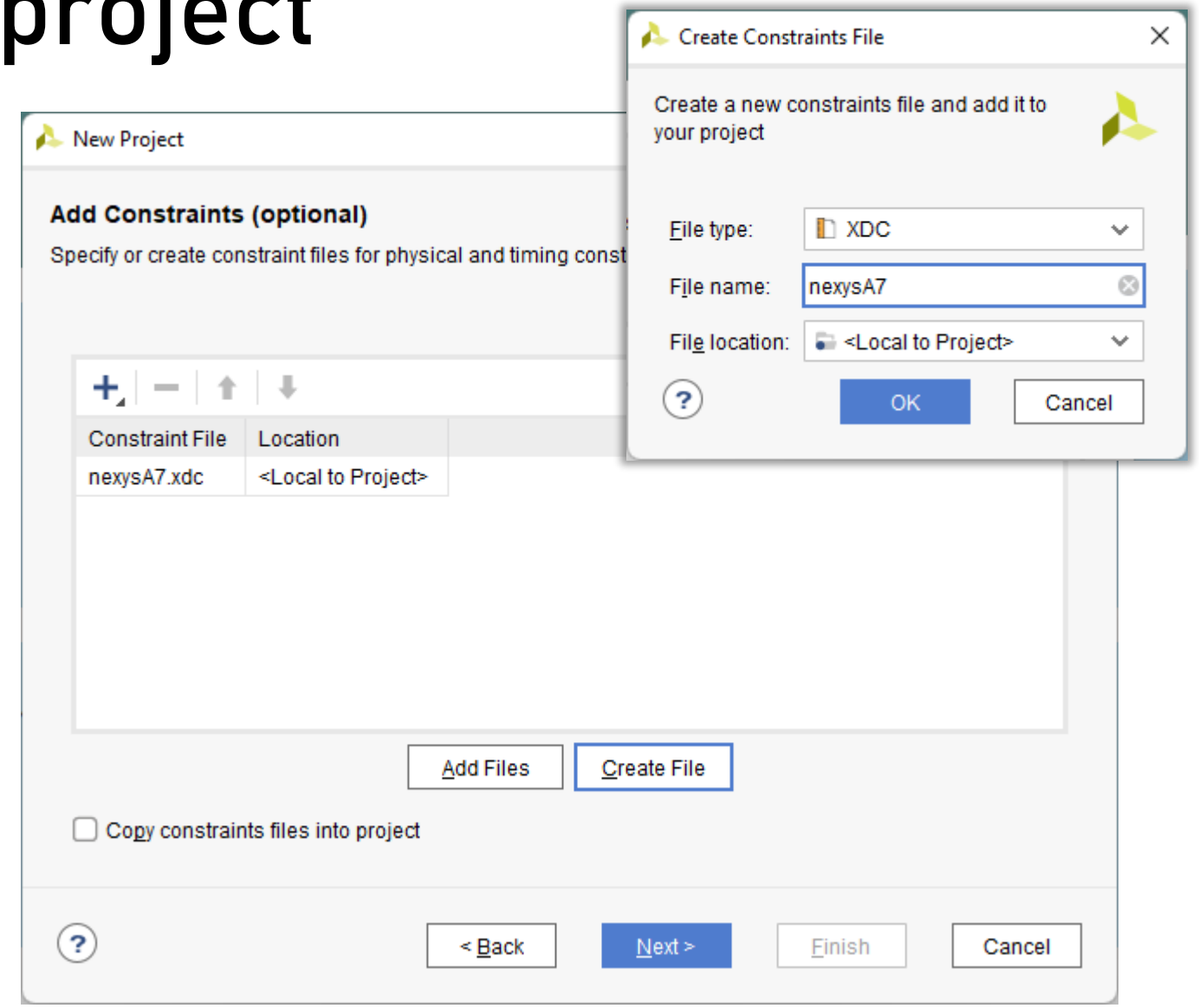
Xilinx Vivado: new project

1. File → Project → New…
2. Define the project name and location
3. Select "RTL Project"
4. Add/create source files (select VHDL as a target and simulator language)



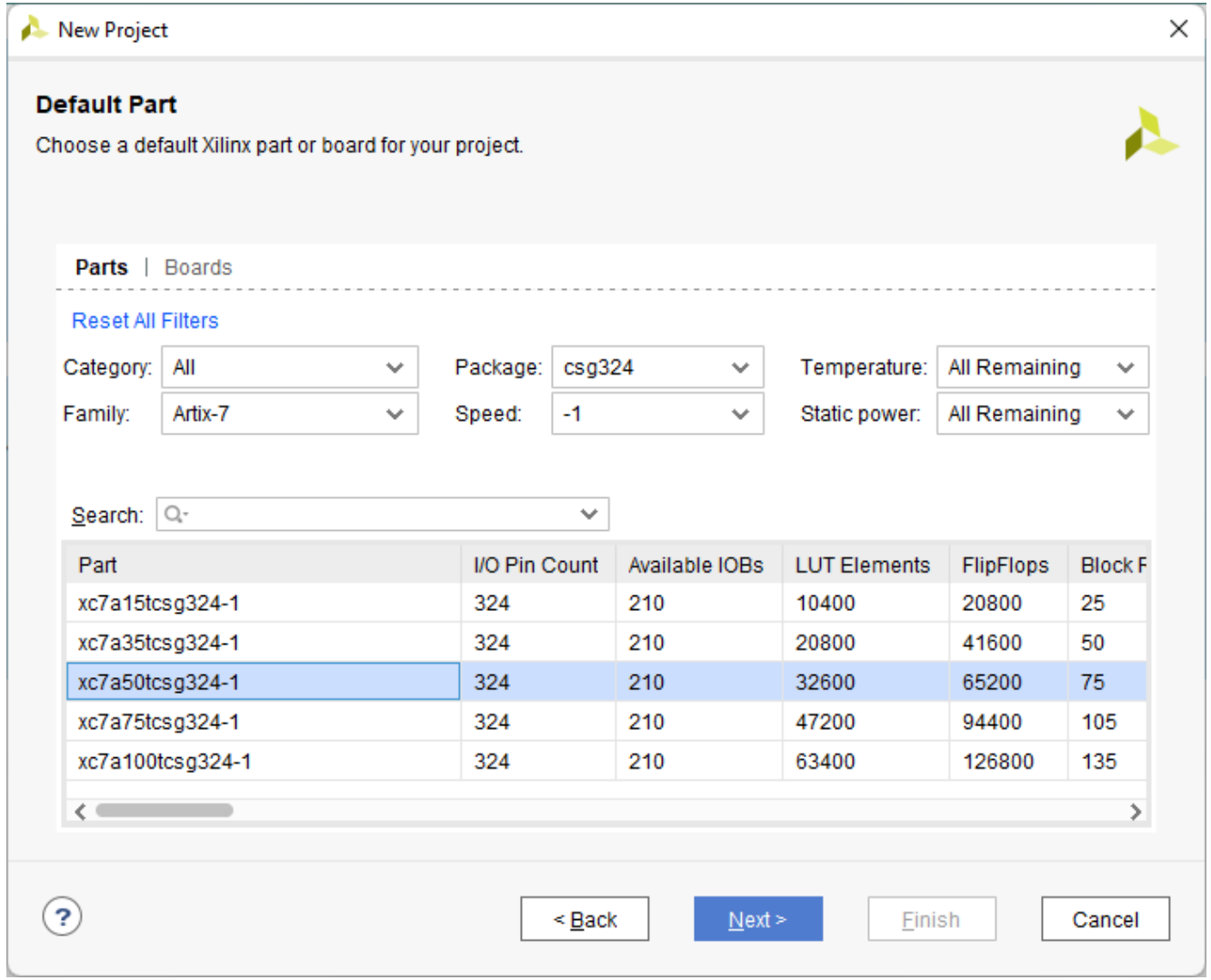
Xilinx Vivado: new project

5. Add/create XDC file for constraints definition
   - see Moodle for links to templates curated by Digilent



Xilinx Vivado: new project

6. Choose a FPGA chip
   - Nexys4,
     Nexys4 DDR and
     Nexys A7 100T:
     `xc7a100tcsg324-1`
   - Nexys A7 50T:
     `xc7a50tcsg324-1`
7. Next
8. Finish



Challenge

1. Create your first project. Describe a modul in VHDL that will turn a LED on or off with respect to a switch.

2. Design a comparator of two four-bits numbers
   - each number is represented by four switches
   - the result of a comparison should be displayed using LEDs:
     - `output = 2 =` 10_2, when the first number is greater than the second
     - `output = 1 =` 01_2, when the first number is less than the second
     - `output = 0 =` 00_2, when the numbers are equal