

# Digital Design

## Week 3: Combinational Logic Part III



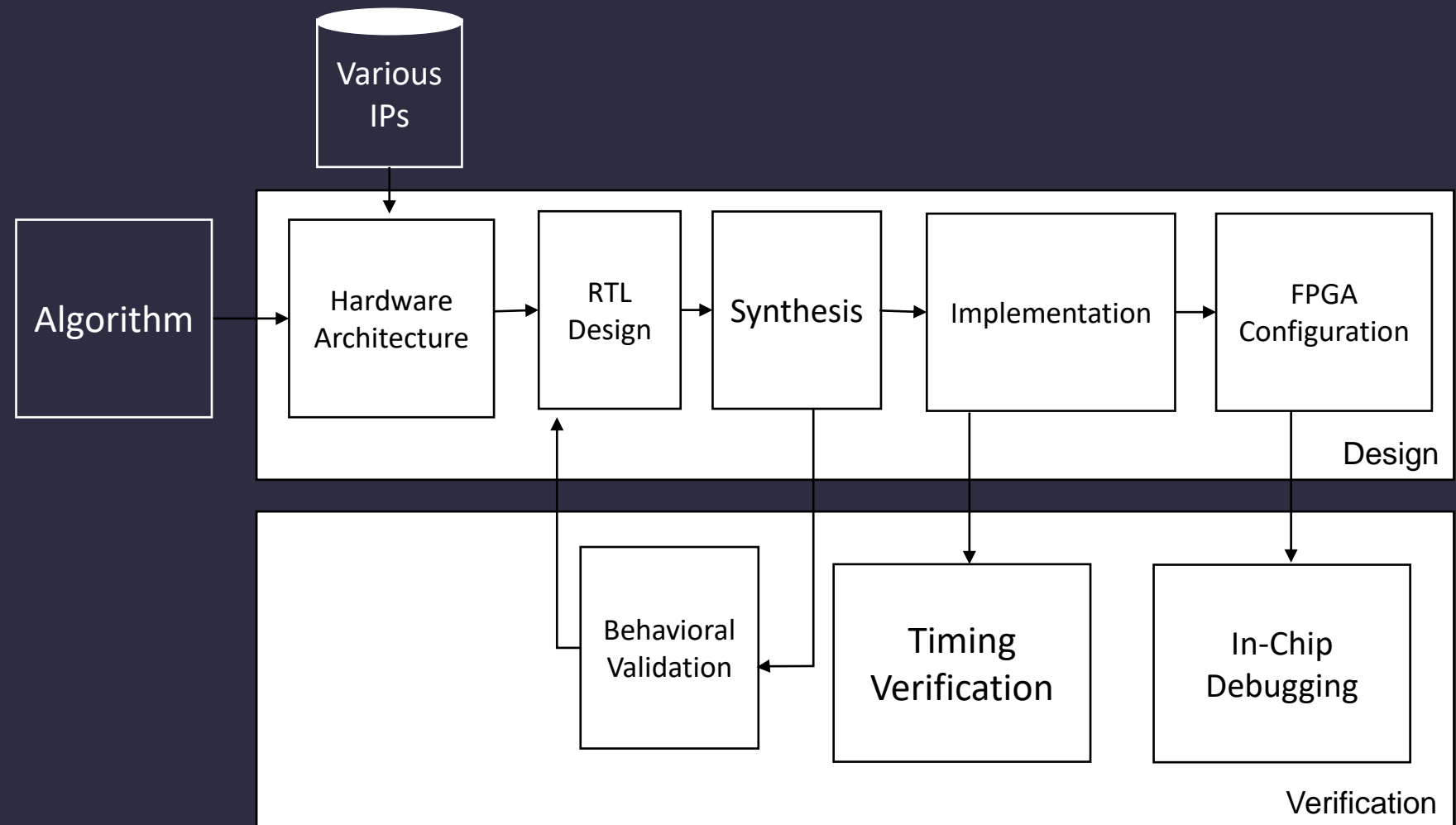
Fenerbahce University

# Combinational Circuits

- Combinational Circuits
  - What is FPGA?
  - Vivado Design Tool

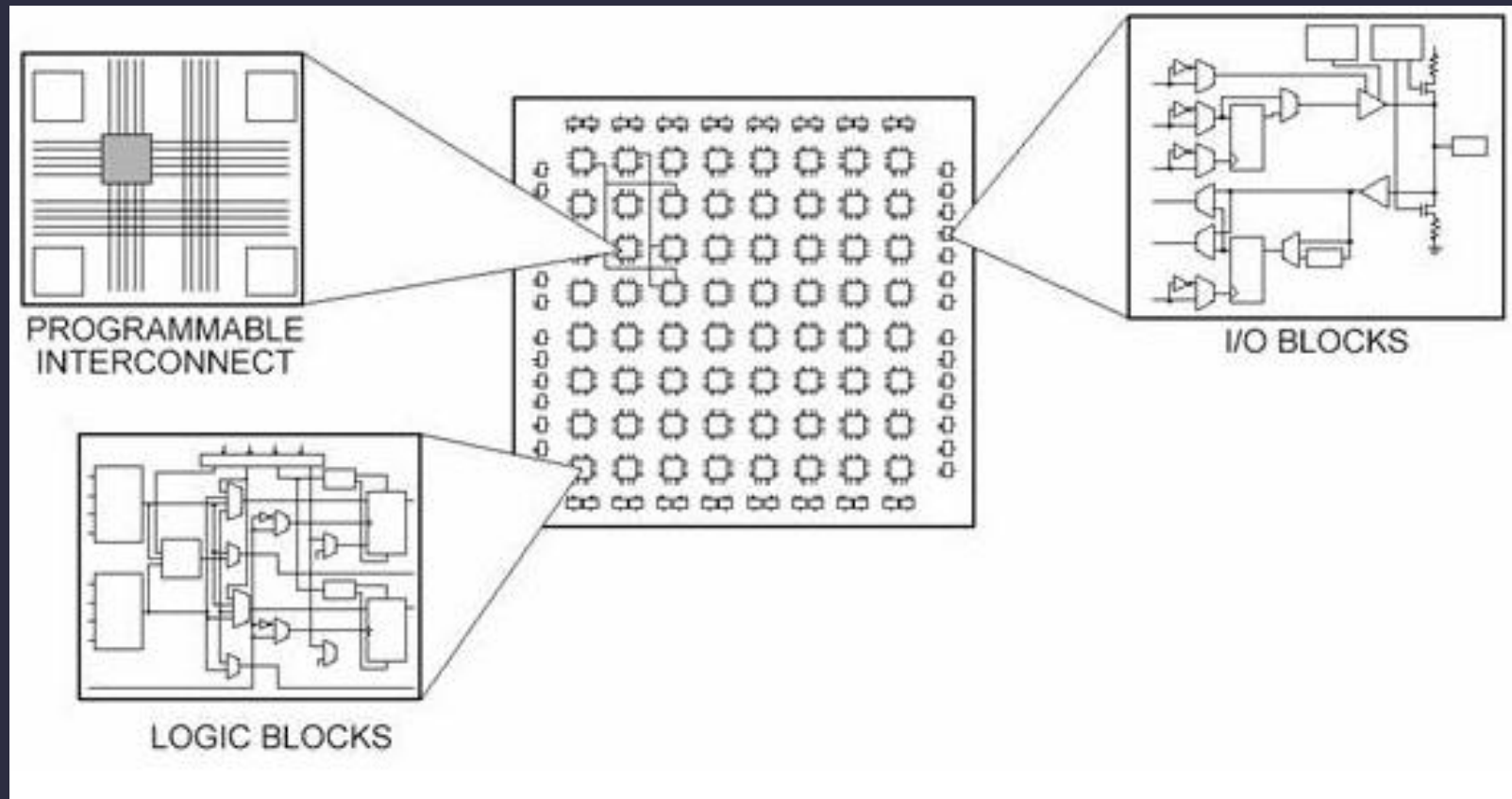
# Vivado Design Tool

- *Design Flow*



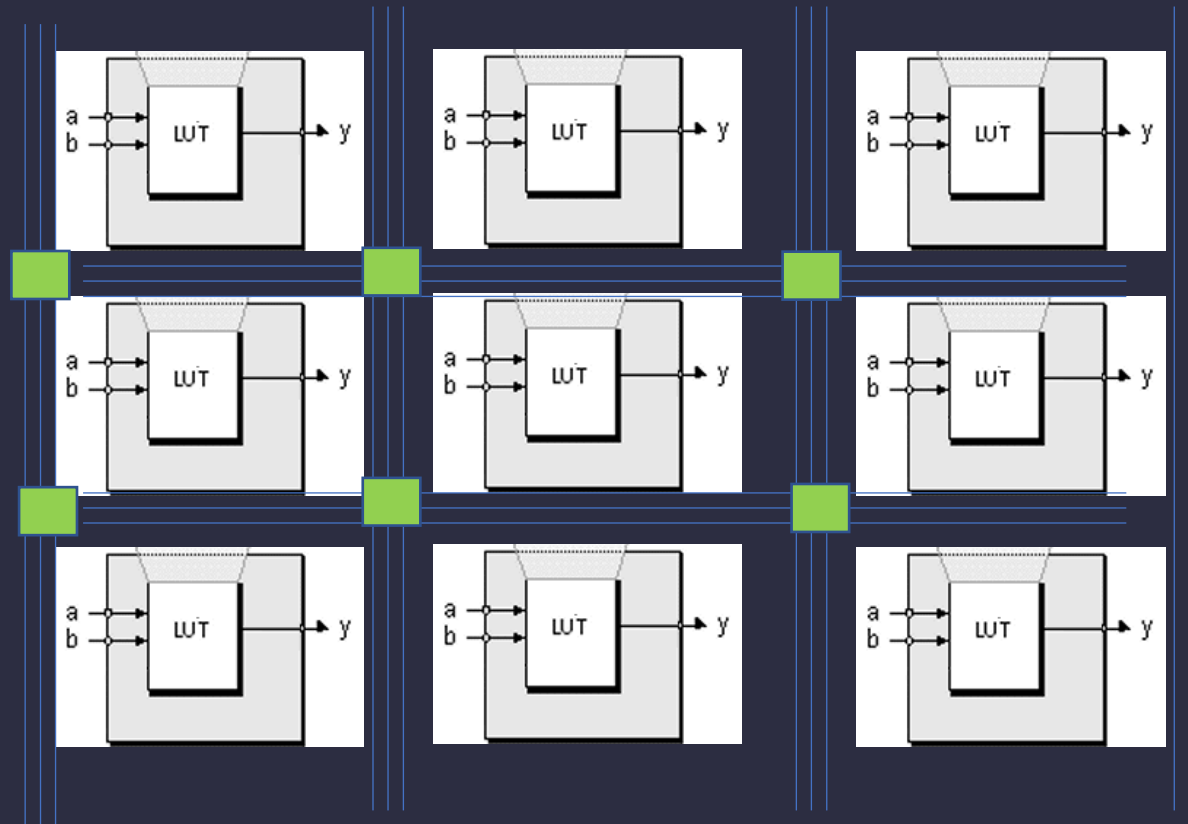
# Verilog – Combinational Circuits

## What is FPGA?



# Verilog – Combinational Circuits

What is FPGA?



# Chip Design Training

## What is FPGA?

- FPGA ( Field programmable Gate Arrays ) is an integrated chip.
- It contains programmable blocks and configurable connections between these blocks.
- By programming these blocks and connections, the desired circuit can be implemented in the FPGA .



Example FPGA Chip

# Chip Design Training

## What is FPGA?

- Some FPGAs can only be programmed once.
- It is called one-time programmable (OTP).

# Chip Design Training

## What is FPGA?

- "Field Programmable" means that it can be programmed in the required application with unlimited count.
- This means that after the FPGA chips are produced in the factory, they can be used in the desired design later on.



# Chip Design Training

## Why is FPGA important?

Wide variety of integrated circuits (IC – Integrated circuits ) are available.

- Memory
- Microprocessors
- Programmable logic devices ( Programmable logic devices )
  - SPLD (Simple)
  - CPLD ( Complex )
- ASIC – Application specific integrated circuit
- And FPGAs ...

# Chip Design Training

## ASICs

- They are useful in the implementation of huge and complex circuits.
- ASICs are integrated circuits built to perform a specific operation.
- ASICs provide very high performance (operation at high frequencies) , ASIC design complexity and design time and costs are quite high.
- And the produced chips cannot be modified . In case of a fault with the chip, all produced chips will be thrown away.

# Chip Design Training

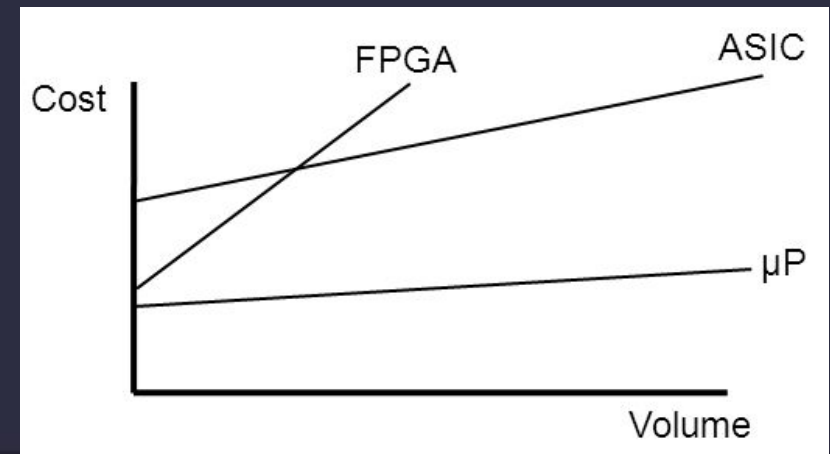
## FPGAs

- FPGAs It is programmable.
- Unlike ASICs ; In the case of an error in the design, the design can be repeatedly modified and tested.

# Chip Design Training

## FPGAs

- FPGAs They are much cheaper than ASICs . ( ASICs are only cheap when millions of units are produced)
- FPGAs is much easier than creating an ASIC design.
- Due to the shorter design time, once a product is produced, the time to market is shorter.
- NRE (Non-recurring engineering) is high when developing a ASIC Design



# Chip Design Training

## FPGA is mainly used

- Telecommunication
- Networking
- Automotive
- Medical
- Various industrial applications
- Prototypes of ASIC designs
- DSP ( Digital signal processor ) applications
- SoC ( System on Chip ), a single IC where all necessary electronics are gathered together

# Chip Design Training

## Reasons for FPGA Use

- Computing Power
- Controlling with nanosecond order

# Chip Design Training

## Reasons for FPGA Use

### Processors

`c[0]=a[0]+b[0];`

`c[1]=a[1]+b[1];`

`c[2]=a[2]+b[2];`

`c[3]=a[3]+b[3];`

`...`

`c[1000]=a[1000]+b[1000];`

### FPGA

`c[0]<=a[0]+b[0];`

`c[1]<=a[1]+b[1];`

`c[2]<=a[2]+b[2];`

`c[3]<=a[3]+b[3];`

`...`

`c[1000]<=a[1000]+b[1000];`

# Chip Design Training

## Reasons for FPGA Use

### Processors

```
c[0]=a[0]+b[0]; ←  
c[1]=a[1]+b[1];  
c[2]=a[2]+b[2];  
c[3]=a[3]+b[3];  
...  
c[1000]=a[1000]+b[1000];
```

### FPGA

```
c[0]<=a[0]+b[0];  
c[1]<=a[1]+b[1];  
c[2]<=a[2]+b[2];  
c[3]<=a[3]+b[3];  
...  
c[1000]<=a[1000]+b[1000];
```



# Chip Design Training

## Reasons for FPGA Use

### Processors

```
c[0]=a[0]+b[0];  
c[1]=a[1]+b[1]; ←  
c[2]=a[2]+b[2];  
c[3]=a[3]+b[3];  
...  
c[1000]=a[1000]+b[1000];
```

### FPGA

```
c[0]<=a[0]+b[0];  
c[1]<=a[1]+b[1];  
c[2]<=a[2]+b[2];  
c[3]<=a[3]+b[3];  
...  
c[1000]<=a[1000]+b[1000];
```

# Chip Design Training

## Reasons for FPGA Use

### Processors

```
c[0]=a[0]+b[0];  
c[1]=a[1]+b[1];  
c[2]=a[2]+b[2]; ←  
c[3]=a[3]+b[3];  
...  
c[1000]=a[1000]+b[1000];
```

### FPGA

```
c[0]<=a[0]+b[0];  
c[1]<=a[1]+b[1];  
c[2]<=a[2]+b[2];  
c[3]<=a[3]+b[3];  
...  
c[1000]<=a[1000]+b[1000];
```

# Chip Design Training

## Reasons for FPGA Use

### Processors

```
c[0]=a[0]+b[0];  
c[1]=a[1]+b[1];  
c[2]=a[2]+b[2];  
c[3]=a[3]+b[3]; ←  
...  
c[1000]=a[1000]+b[1000];
```

### FPGA

```
c[0]<=a[0]+b[0];  
c[1]<=a[1]+b[1];  
c[2]<=a[2]+b[2];  
c[3]<=a[3]+b[3];  
...  
c[1000]<=a[1000]+b[1000];
```

# Chip Design Training

## Reasons for FPGA Use

### Processors


`c[0]=a[0]+b[0];`

`c[1]=a[1]+b[1];`

`c[2]=a[2]+b[2];`

`c[3]=a[3]+b[3];`

...

`c[1000]=a[1000]+b[1000];` 

### FPGA

`c[0]<=a[0]+b[0];`

`c[1]<=a[1]+b[1];`

`c[2]<=a[2]+b[2];`

`c[3]<=a[3]+b[3];`

...

`c[1000]<=a[1000]+b[1000];`

# Chip Design Training

## Reasons for FPGA Use

### Processors

`c[0]=a[0]+b[0];`

`c[1]=a[1]+b[1];`

`c[2]=a[2]+b[2];`

`c[3]=a[3]+b[3];`

...

`c[1000]=a[1000]+b[1000];`

### FPGA

`c[0]<=a[0]+b[0];`



`c[1]<=a[1]+b[1];`



`c[2]<=a[2]+b[2];`



`c[3]<=a[3]+b[3];`



...

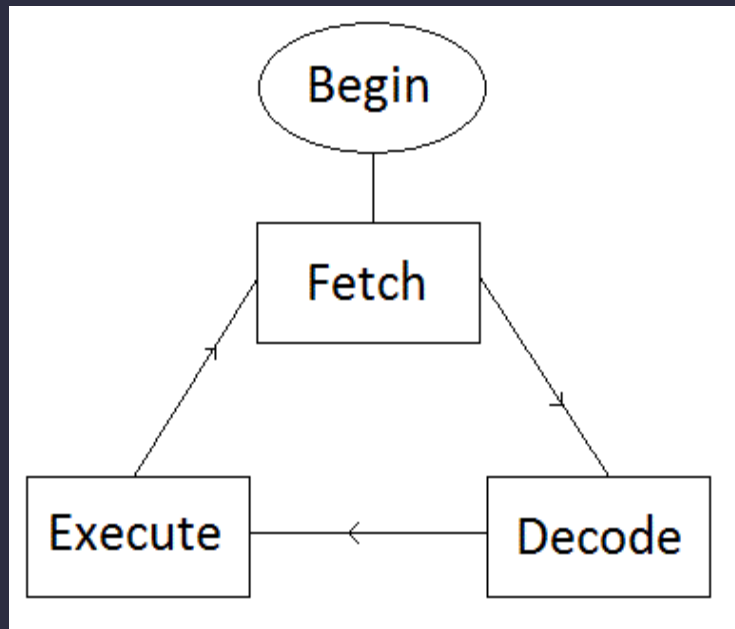
`c[1000]<=a[1000]+b[1000];`



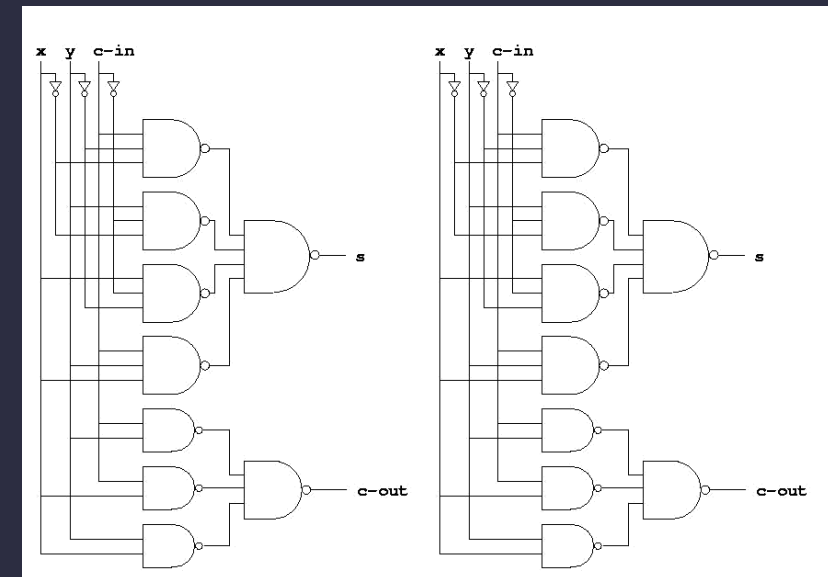
# Chip Design Training

## Reasons for FPGA Use

### Processors



### FPGA



# Chip Design Training

## Reasons for FPGA Use

### CPU

For 1000 transactions (assume each operation takes 1 cycle),

Average Processor frequency: 3 GHz,

Required Time = Number of operations X (1/Frequency)

= 1000 X 1/3 billion

= 333 nanoseconds

### FPGA

For 1000 transactions (assume each operation takes 1 cycle),

Average FPGA frequency: 100mhz

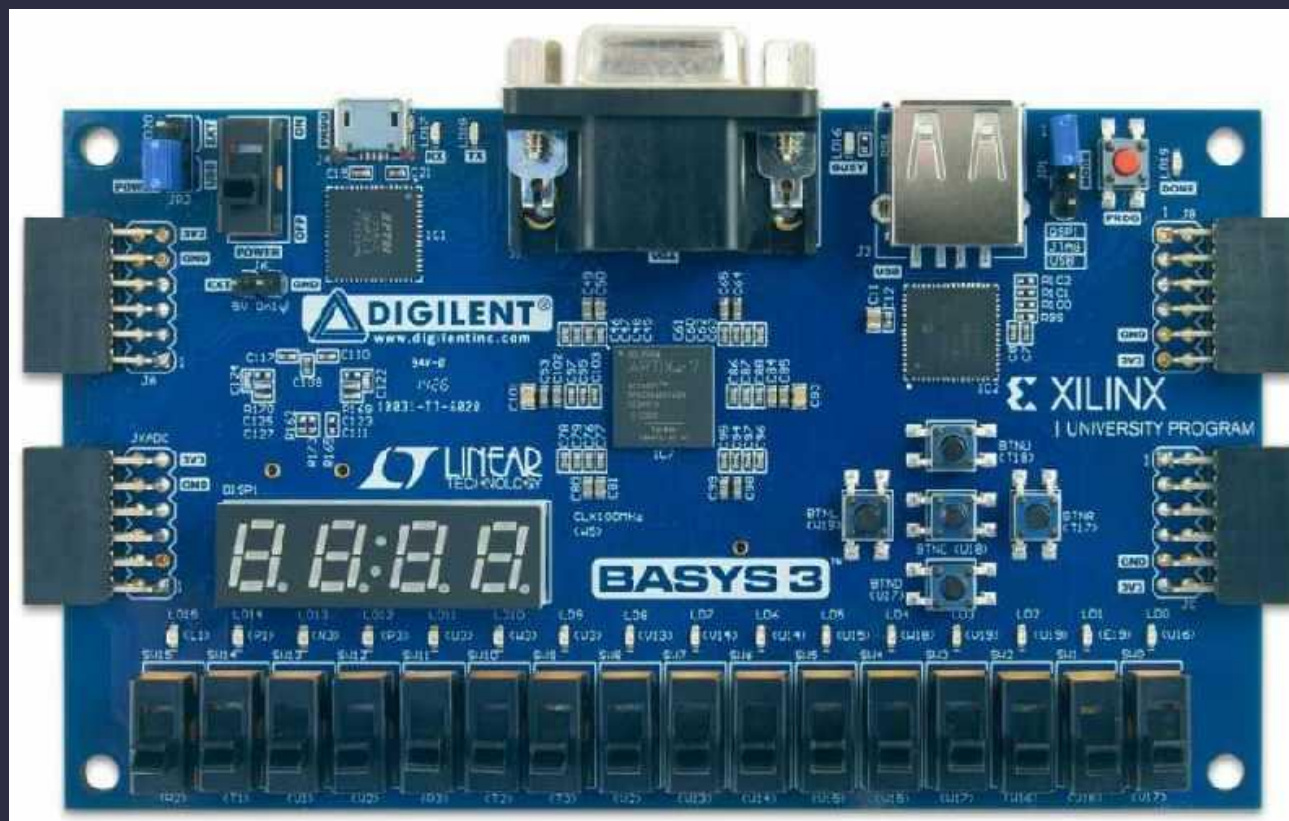
Time required = 1 x (1/Frequency)

= 1 X 1/100m

= 10 nanoseconds

# Verilog – Combinational Circuits

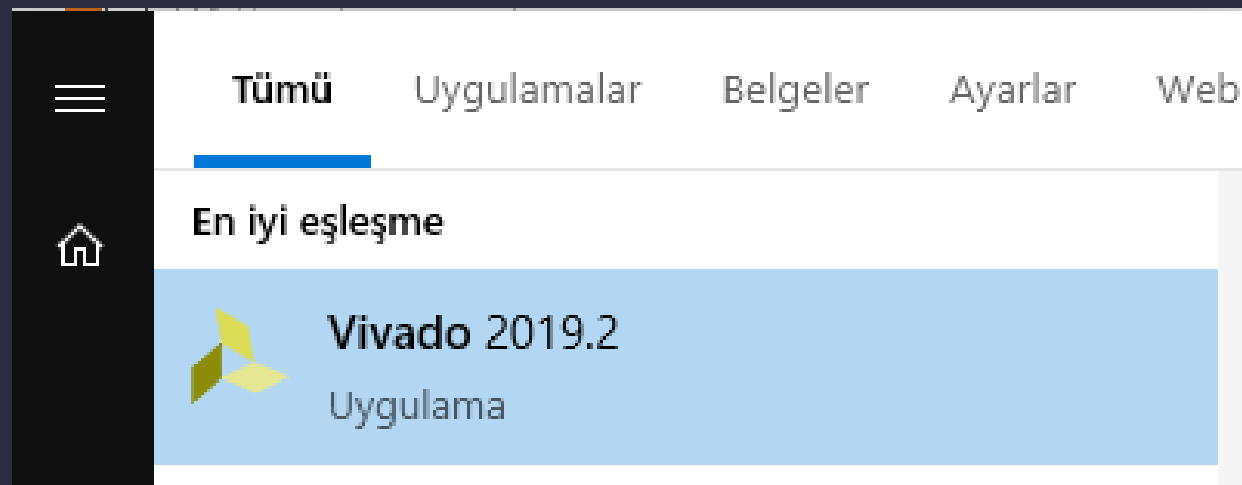
*In LABs Xilinx Basys3 FPGAs with Artix 7 FPGAs will be used.*





# Verilog – Combinational Circuits

Vivado IDE will be used to programming Xilinx Based FPGAs



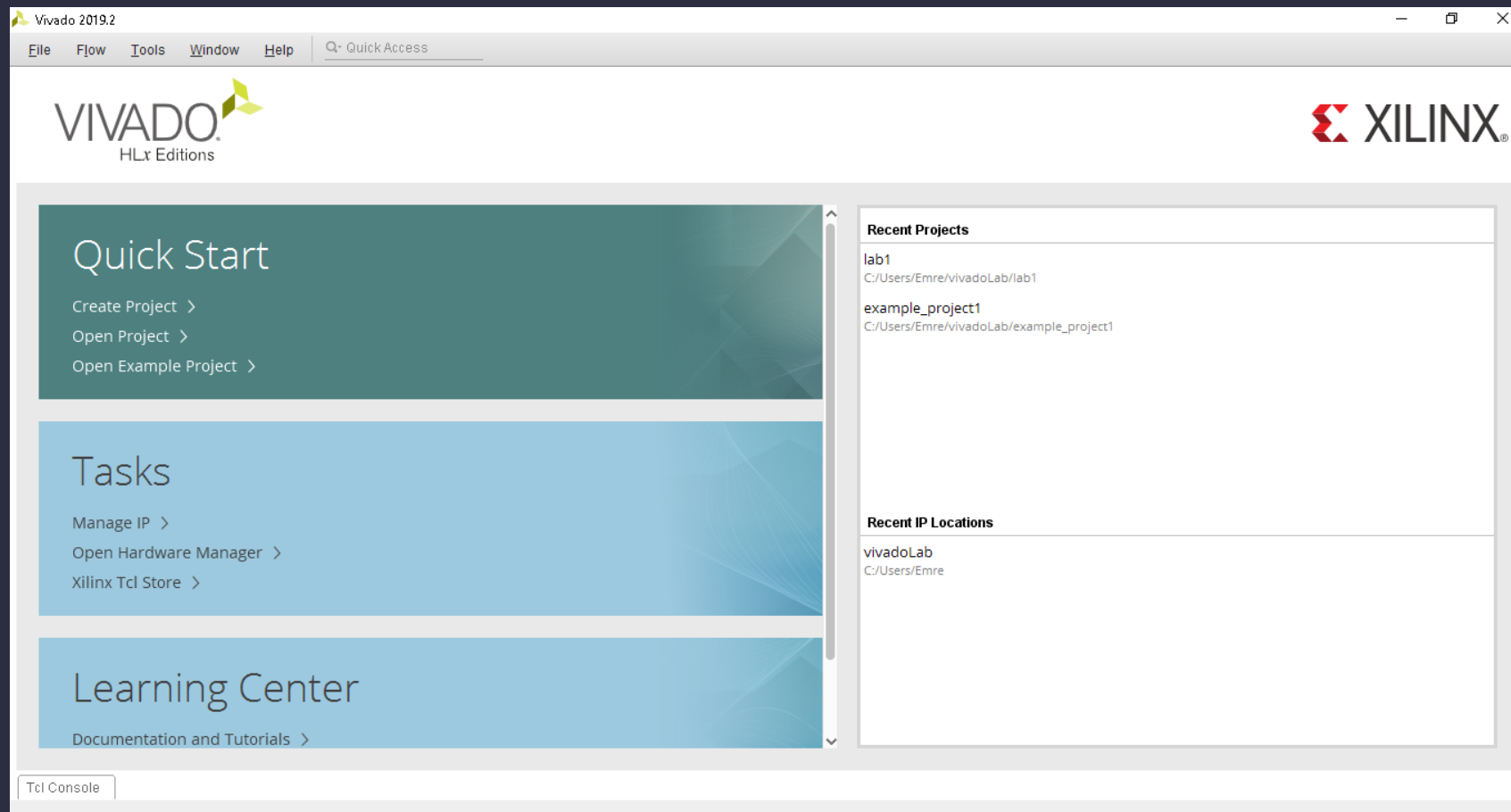
# Verilog – Combinational Circuits

## Vivado Design Tool

- Download Address: <https://www.xilinx.com/support/download.html>
- Installation and Licensing Video : <https://www.youtube.com/watch?v=yW7t28XaVEs>

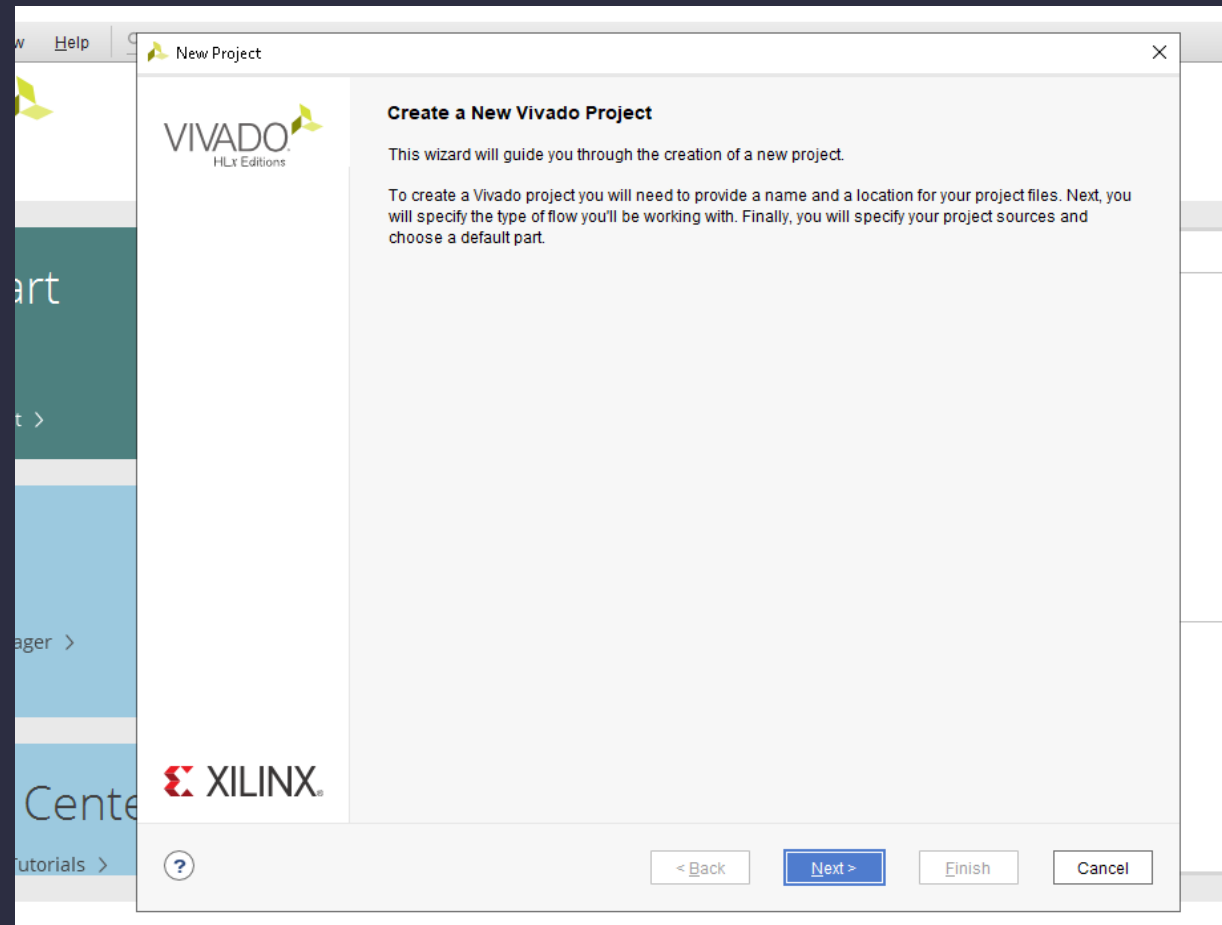
# Verilog – Combinational Circuits

## Vivado Design Tool



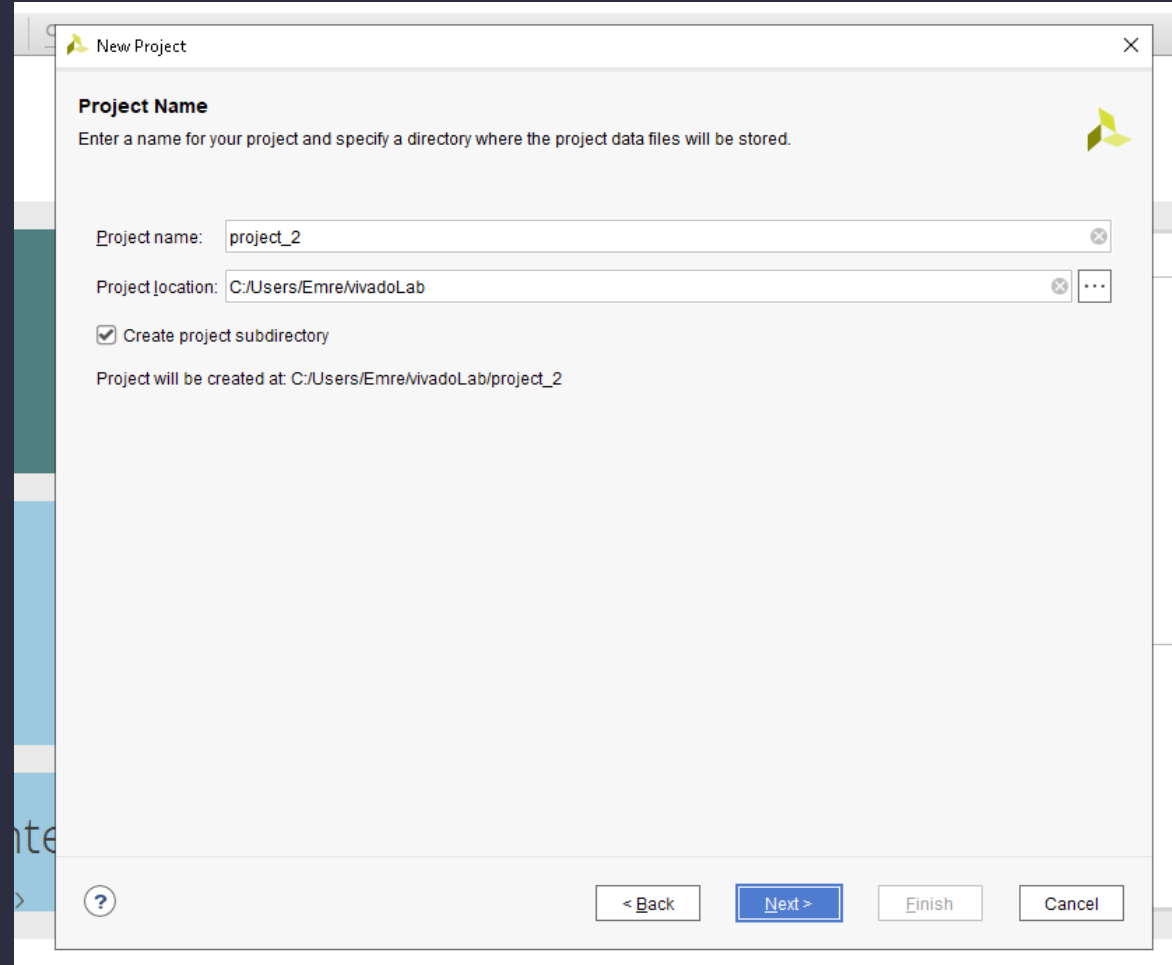
# Verilog – Combinational Circuits

## Vivado Design Tool



# Verilog – Combinational Circuits

## Vivado Design Tool



The image shows the 'New Project' wizard in the Vivado Design Tool. The window title is 'New Project'. The main heading is 'Project Name'. Below it, a instruction says 'Enter a name for your project and specify a directory where the project data files will be stored.' There are two input fields: 'Project name:' with the text 'project\_2' and 'Project location:' with the text 'C:/Users/Emre/MivadoLab'. A checkbox labeled 'Create project subdirectory' is checked. Below the fields, it says 'Project will be created at: C:/Users/Emre/MivadoLab/project\_2'. At the bottom, there are four buttons: '< Back', 'Next >', 'Finish', and 'Cancel'. A help icon (?) is also present at the bottom left.

**New Project**

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

Project name:

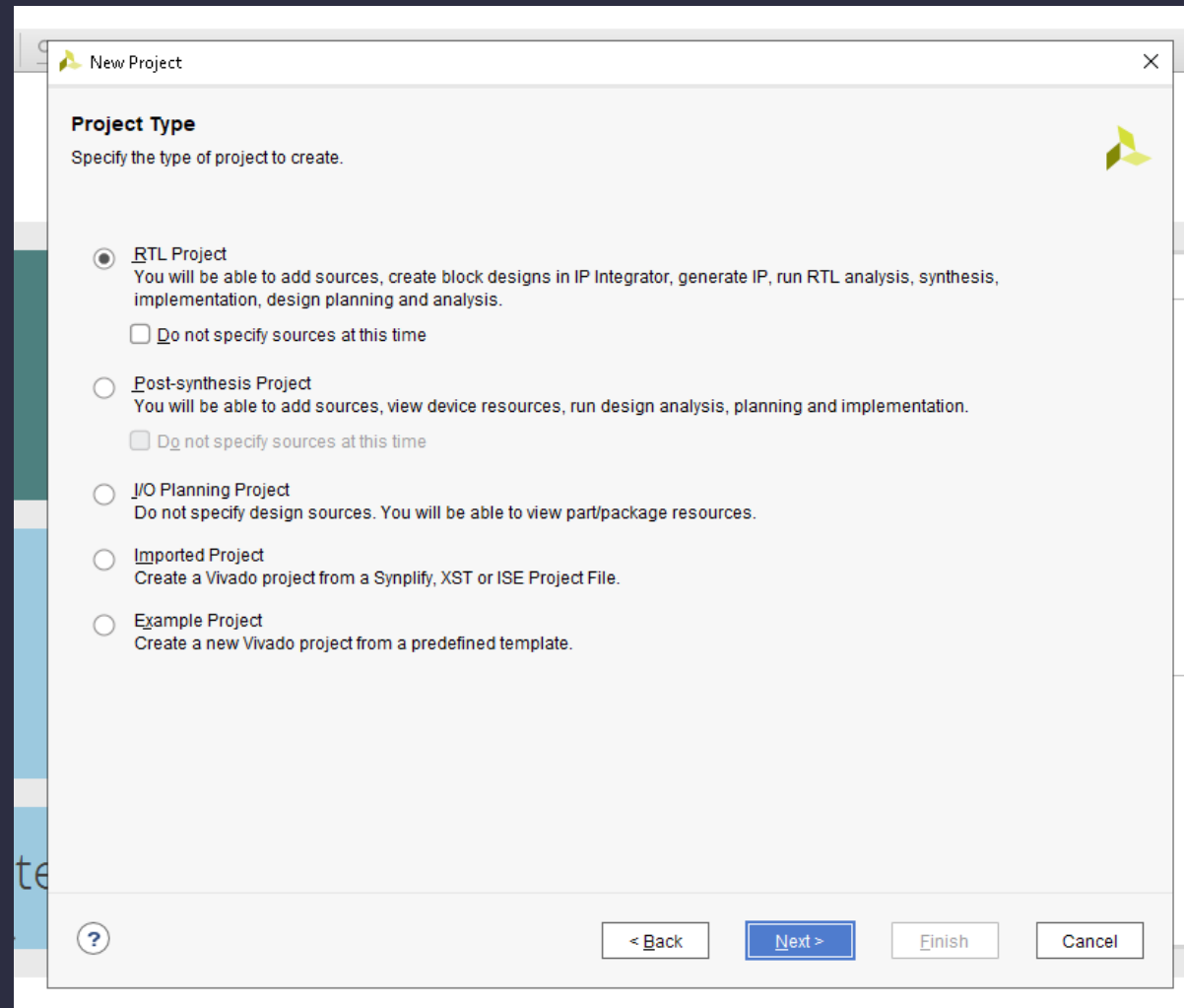
Project location:

☒ Create project subdirectory

Project will be created at: C:/Users/Emre/MivadoLab/project\_2

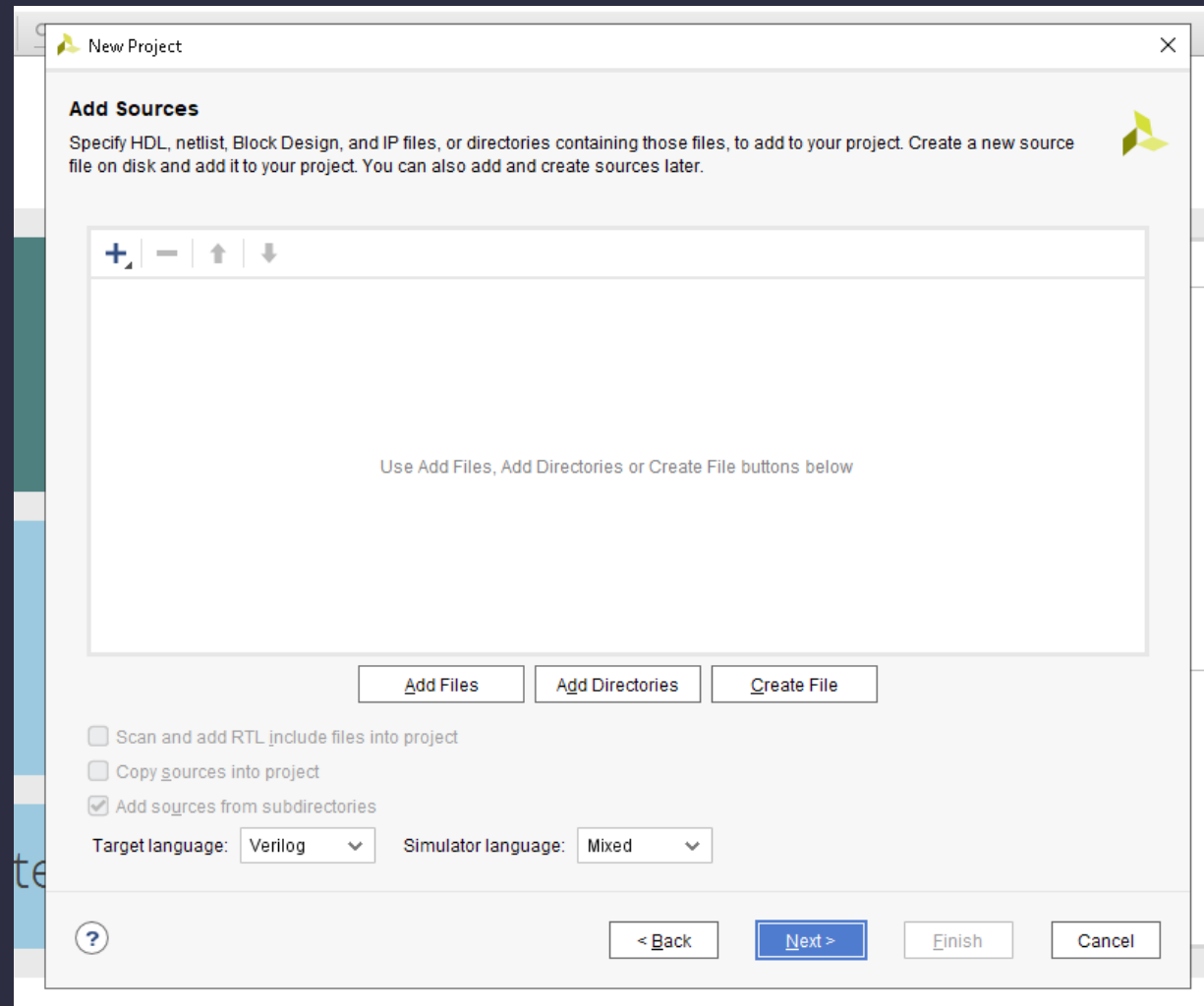
# Verilog – Combinational Circuits

## Vivado Design Tool



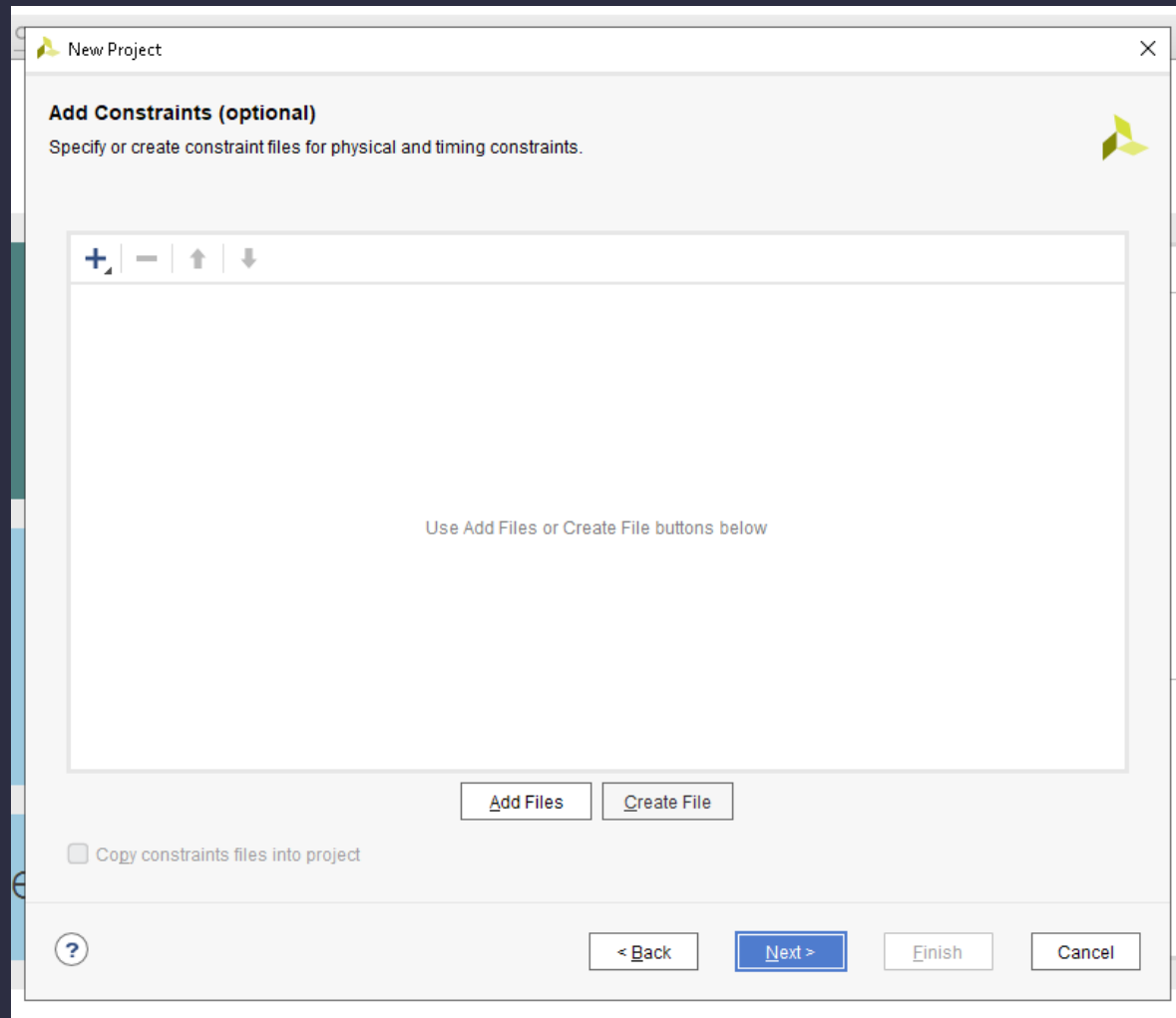
# Verilog – Combinational Circuits

## Vivado Design Tool



# Verilog – Combinational Circuits

## Vivado Design Tool





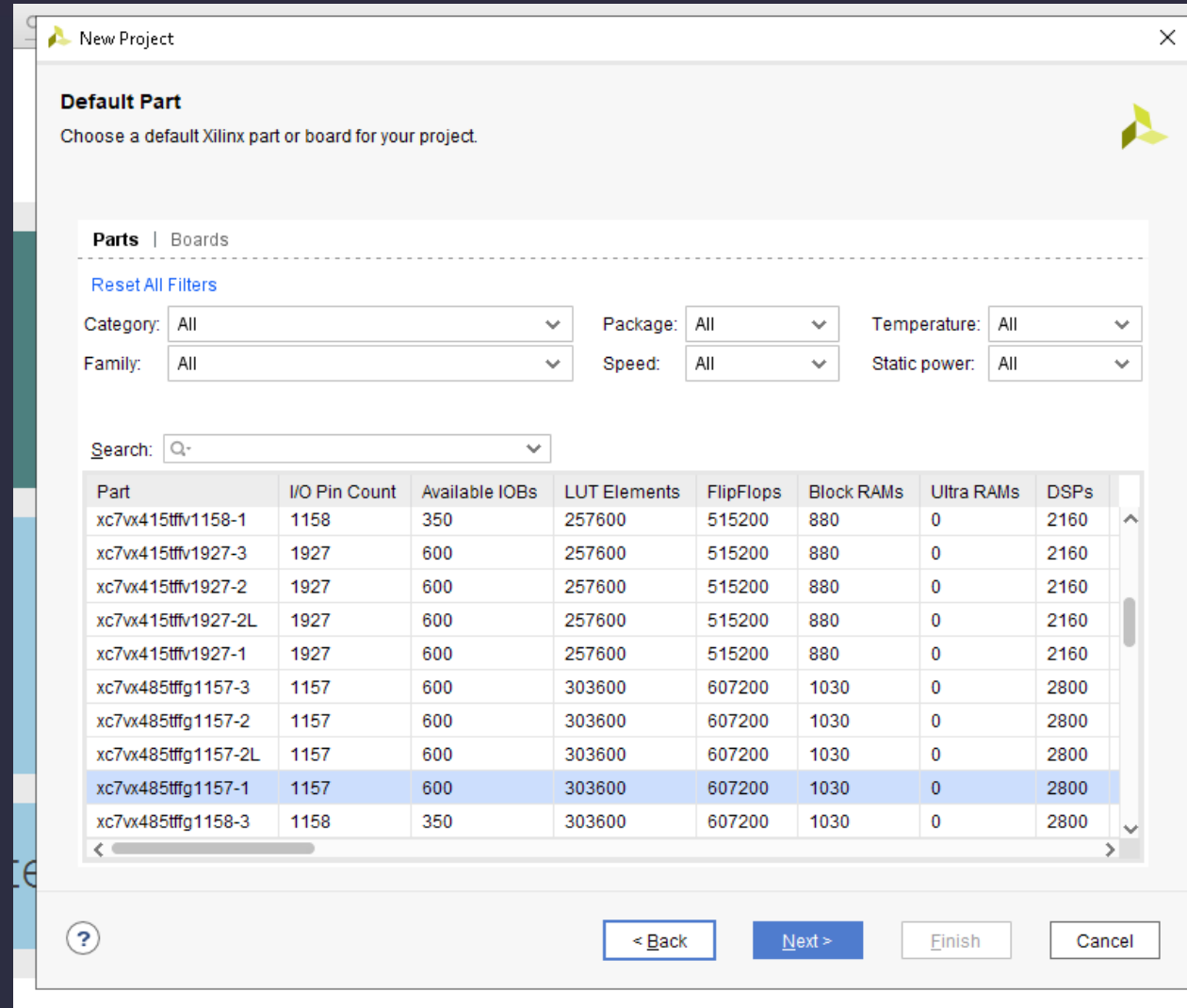
# Verilog – Combinational Circuits

Vivado Design Tool

is the FPGA we will use in LABs

XC7A35Tcpg236-1

The model must be selected.



**New Project**

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

**Parts** | Boards

[Reset All Filters](#)

Category: All Package: All Temperature: All  
Family: All Speed: All Static power: All

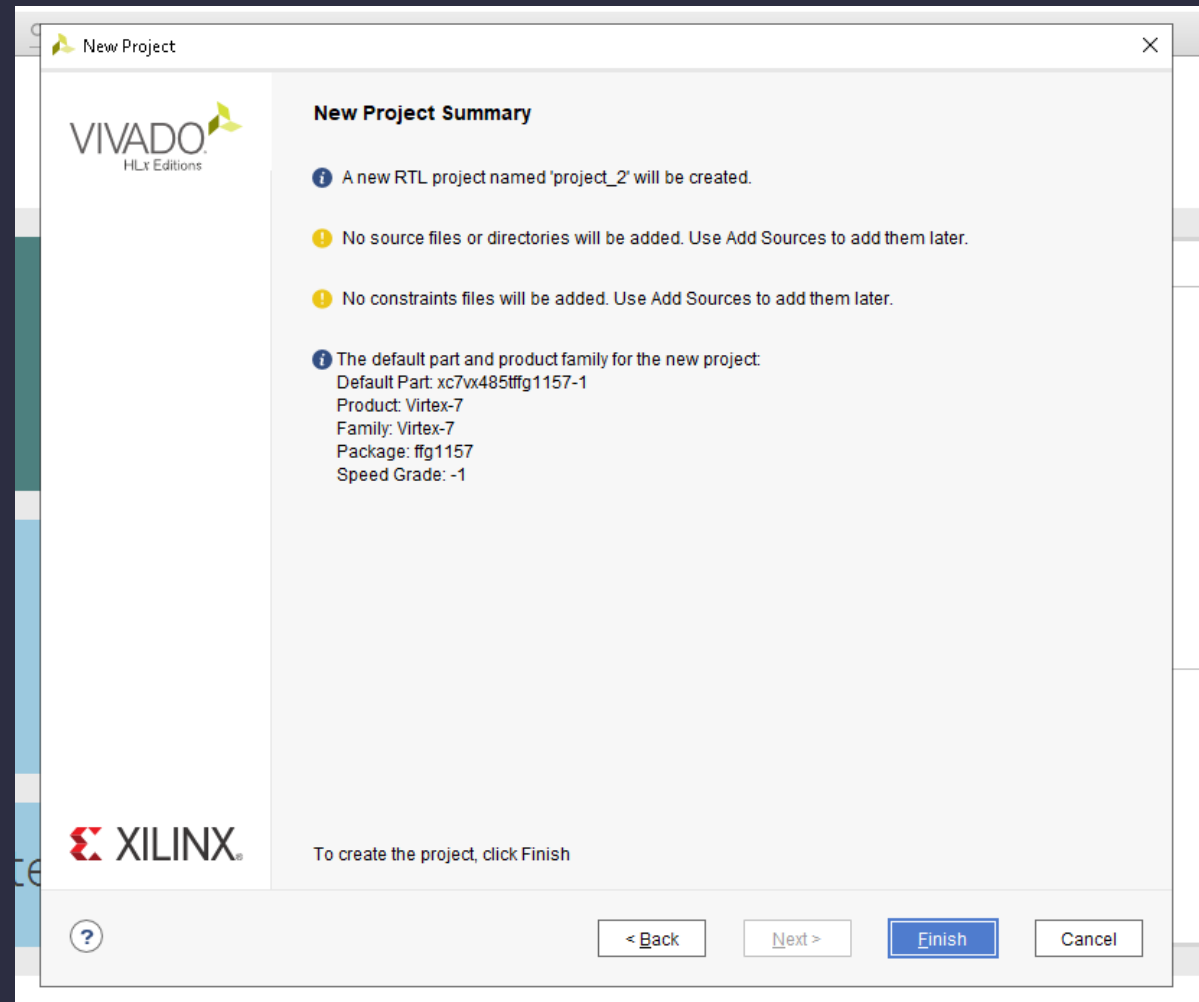
Search: Q-

Part	I/O Pin Count	Available IOBs	LUT Elements	FlipFlops	Block RAMs	Ultra RAMs	DSPs
xc7vx415tffv1158-1	1158	350	257600	515200	880	0	2160
xc7vx415tffv1927-3	1927	600	257600	515200	880	0	2160
xc7vx415tffv1927-2	1927	600	257600	515200	880	0	2160
xc7vx415tffv1927-2L	1927	600	257600	515200	880	0	2160
xc7vx415tffv1927-1	1927	600	257600	515200	880	0	2160
xc7vx485tffg1157-3	1157	600	303600	607200	1030	0	2800
xc7vx485tffg1157-2	1157	600	303600	607200	1030	0	2800
xc7vx485tffg1157-2L	1157	600	303600	607200	1030	0	2800
<b>xc7vx485tffg1157-1</b>	<b>1157</b>	<b>600</b>	<b>303600</b>	<b>607200</b>	<b>1030</b>	<b>0</b>	<b>2800</b>
xc7vx485tffg1158-3	1158	350	303600	607200	1030	0	2800

< Back Next > Finish Cancel

# Verilog – Combinational Circuits

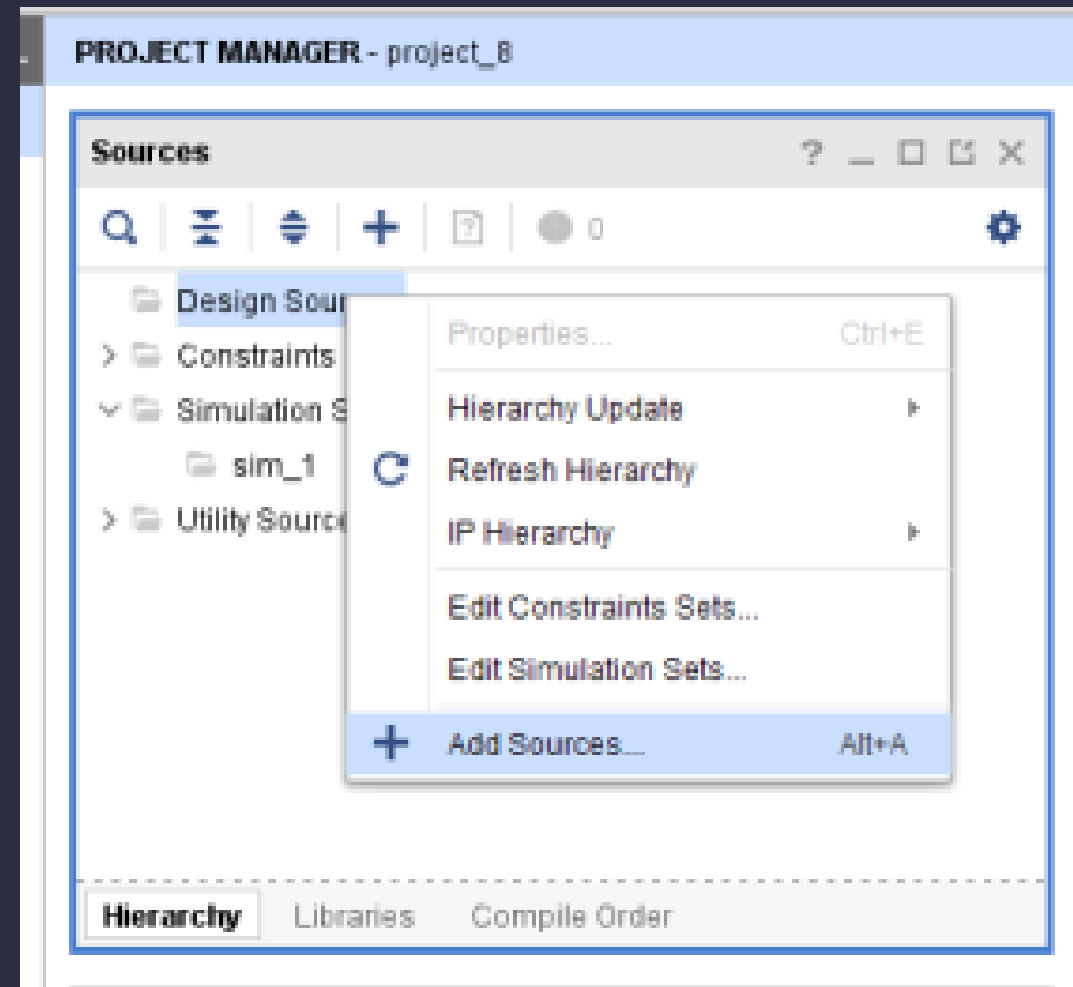
## Vivado Design Tool



# Verilog – Combinational Circuits

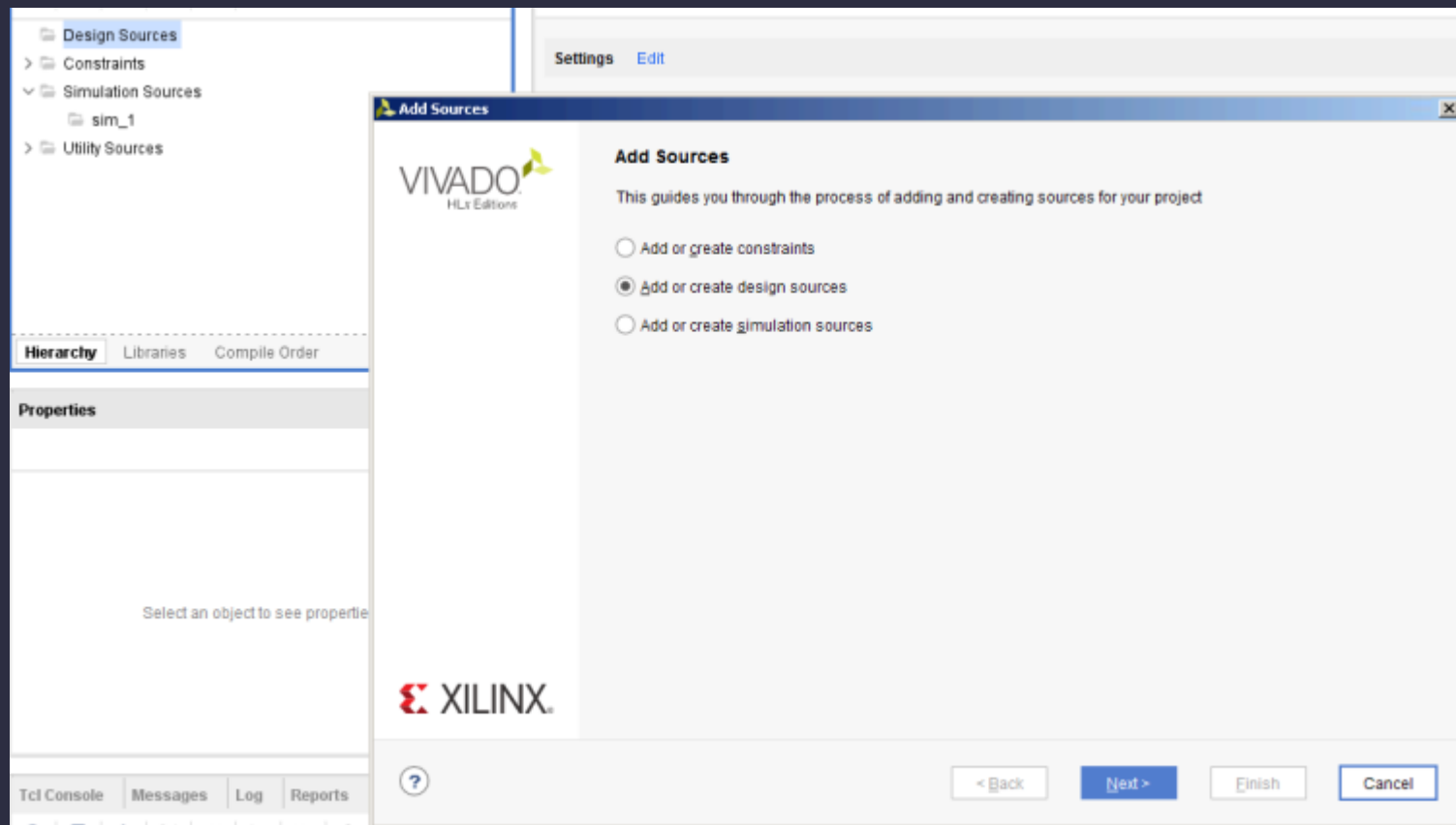
## Vivado Design Tool

Adding a new design resource



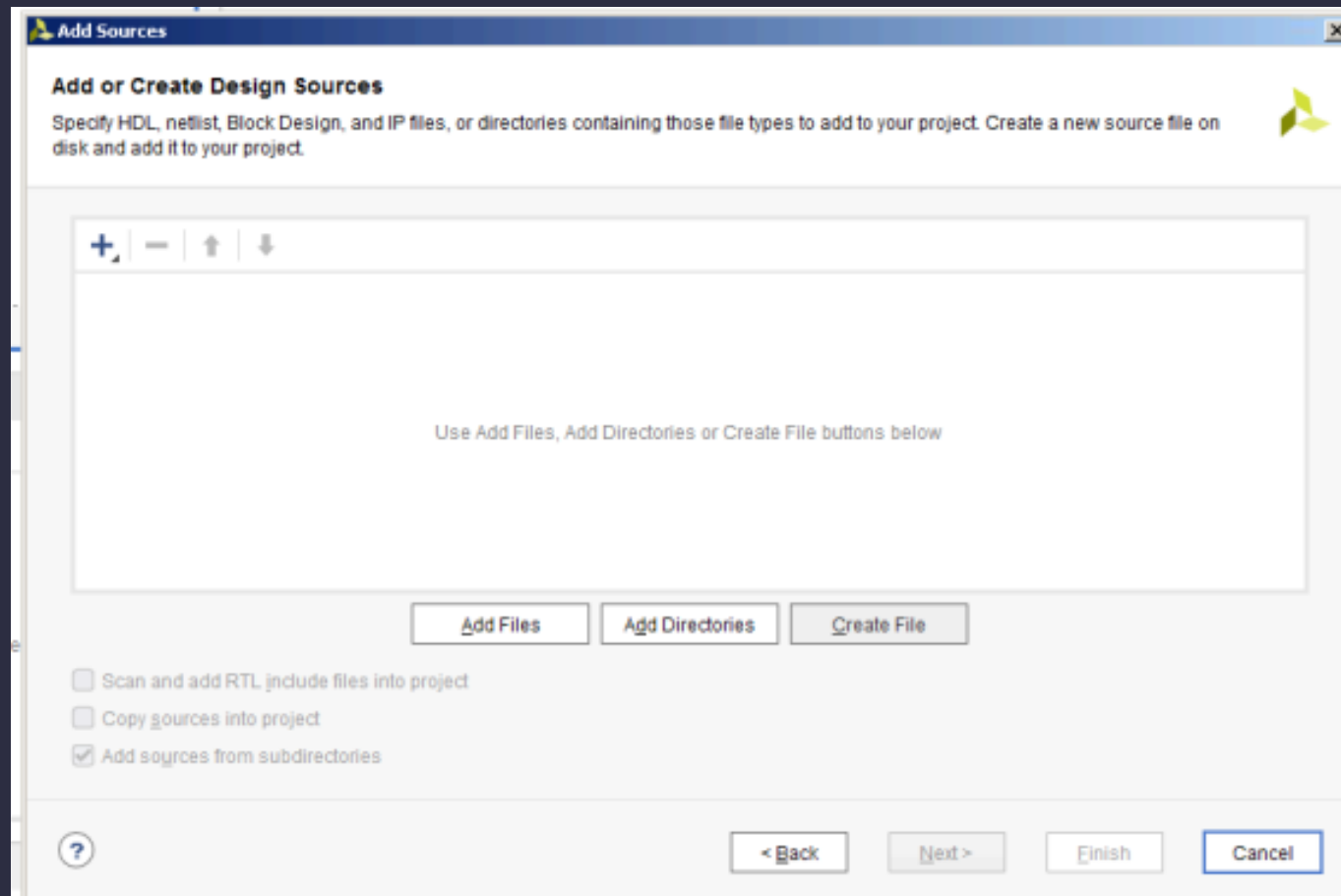
# Verilog – Combinational Circuits

## Vivado Design Tool



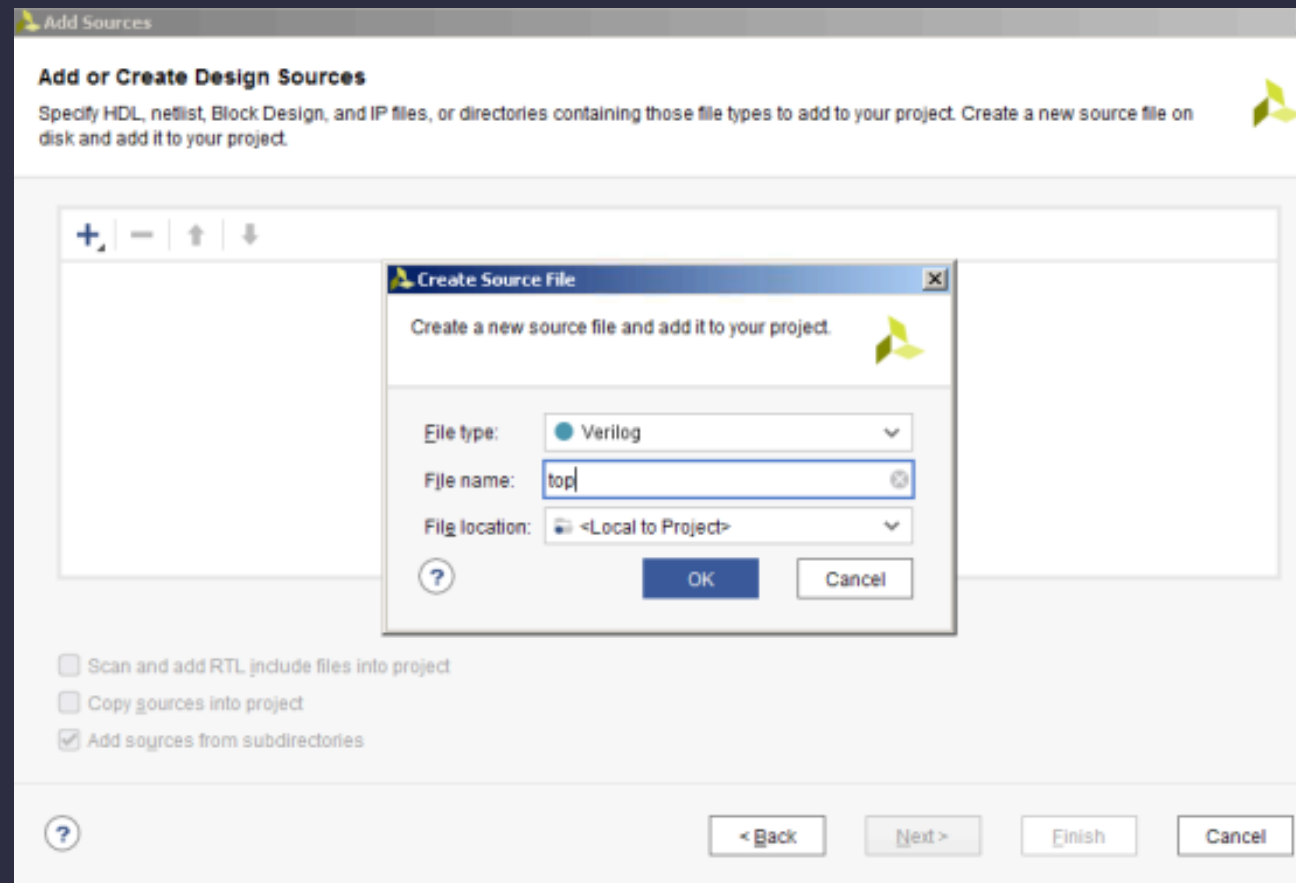
# Verilog – Combinational Circuits

## Vivado Design Tool



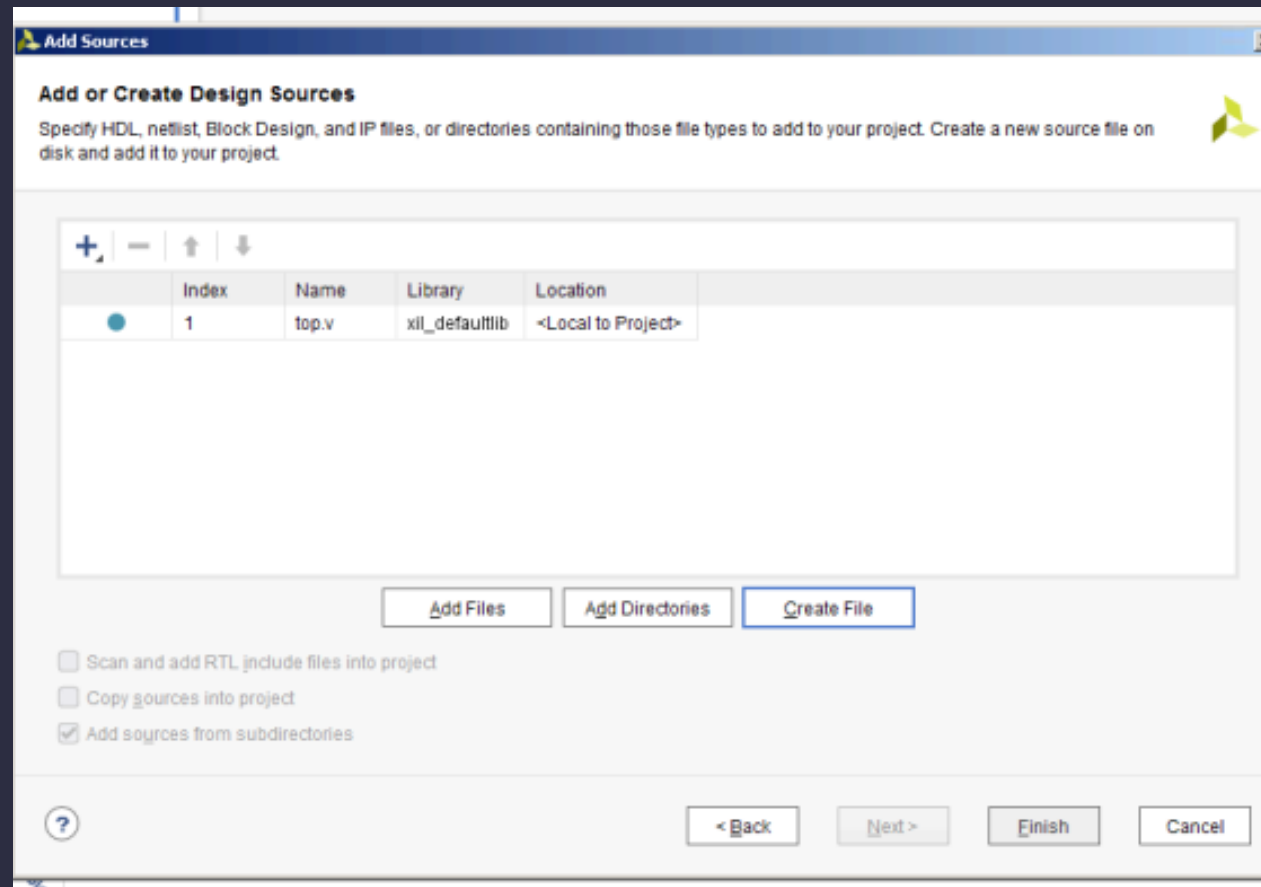
# Verilog – Combinational Circuits

## Vivado Design Tool



# Verilog – Combinational Circuits

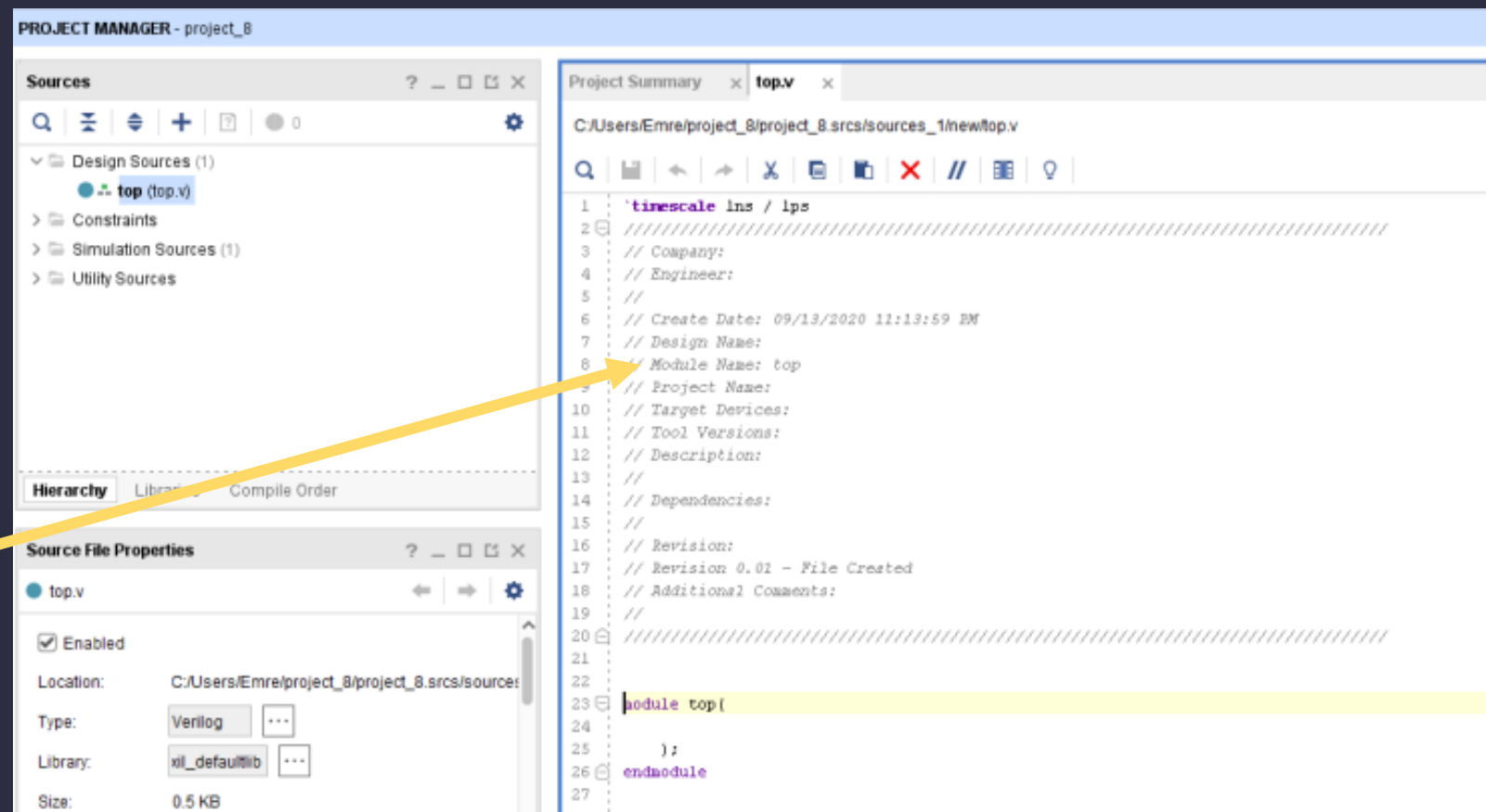
## Vivado Design Tool



# Verilog – Combinational Circuits

## Vivado Design Tool

RTL Design  
File To Do

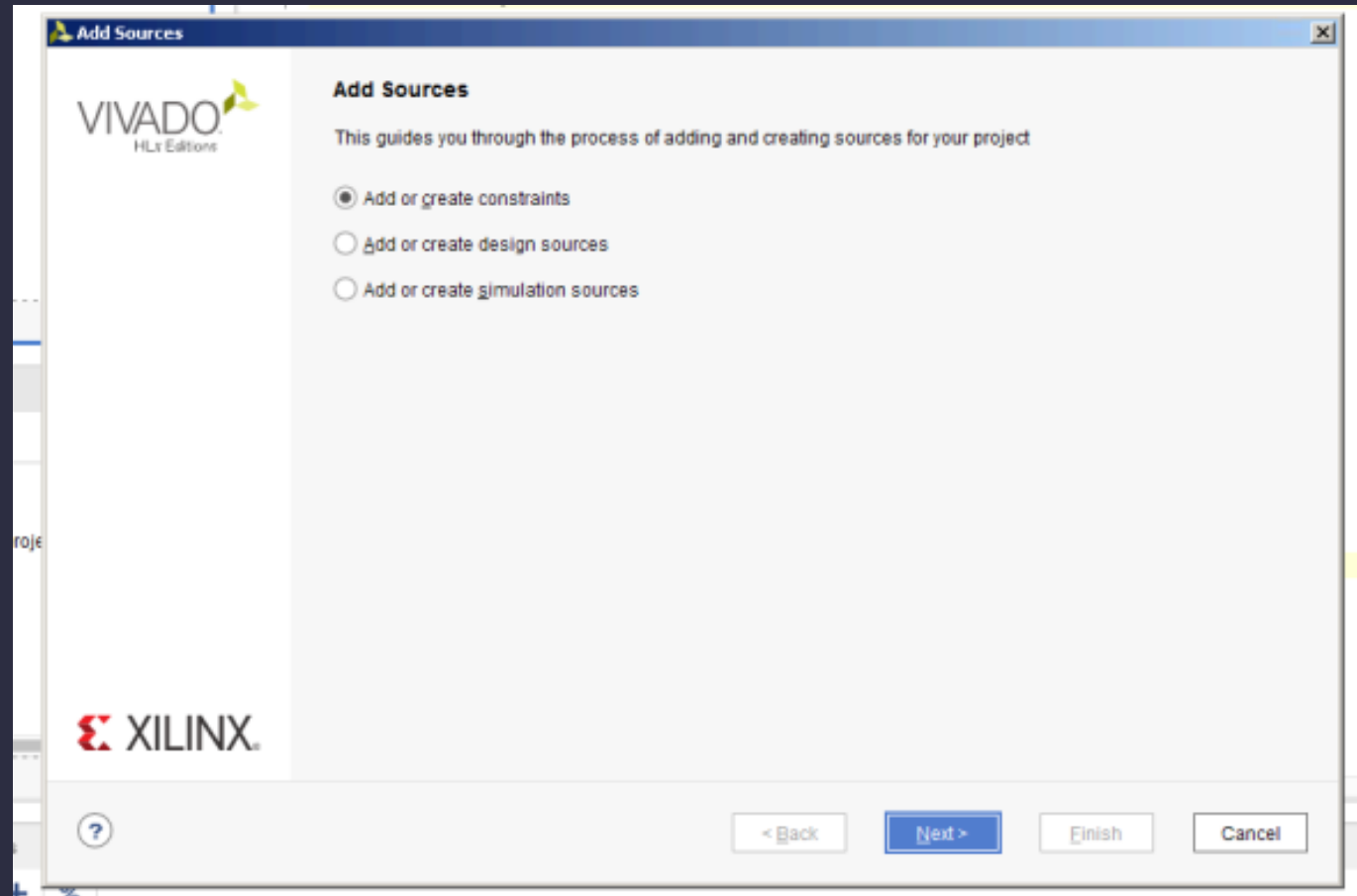




# Verilog – Combinational Circuits

Vivado Design Tool

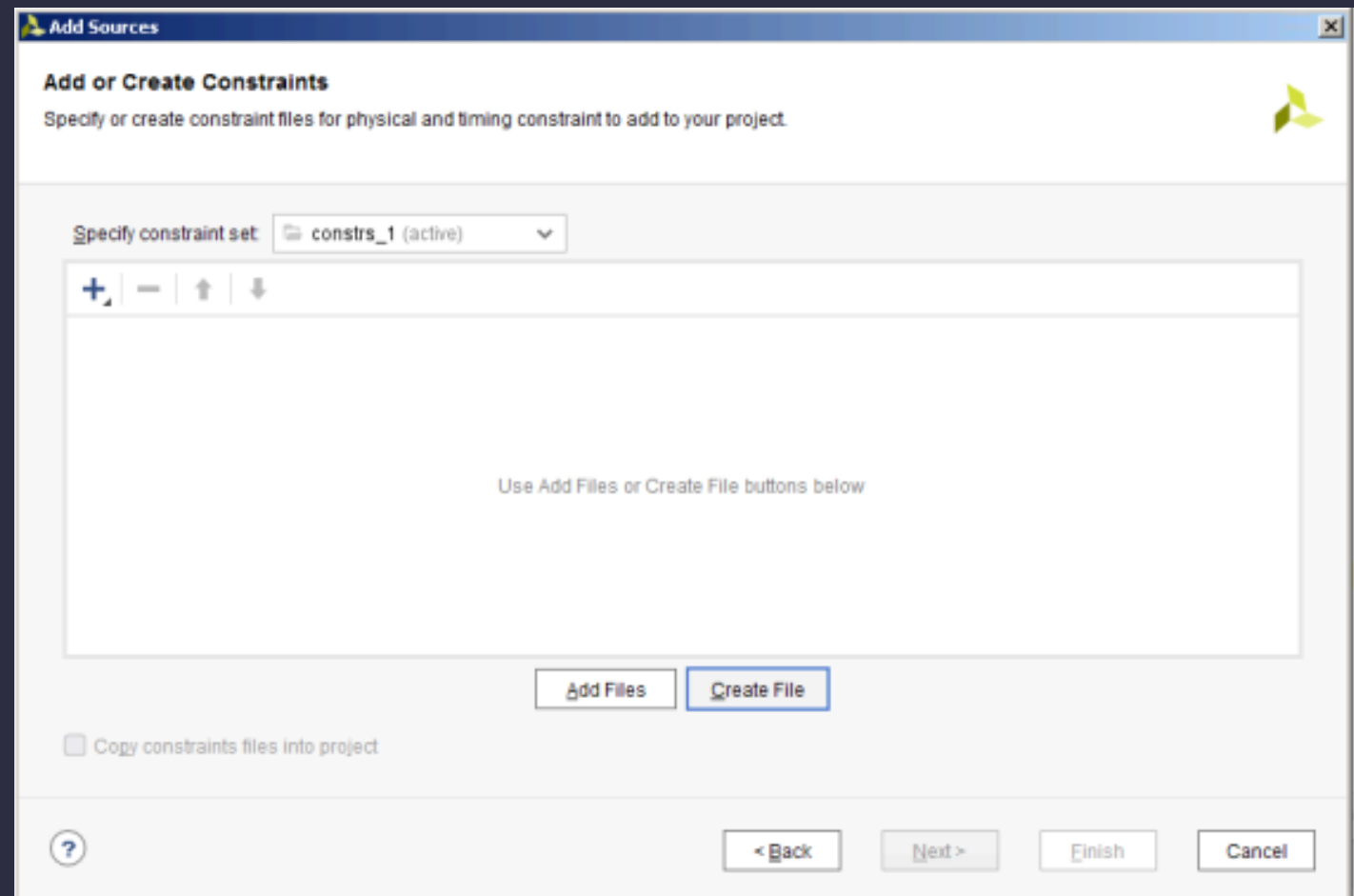
Adding a constraint



# Verilog – Combinational Circuits

## Vivado Design Tool

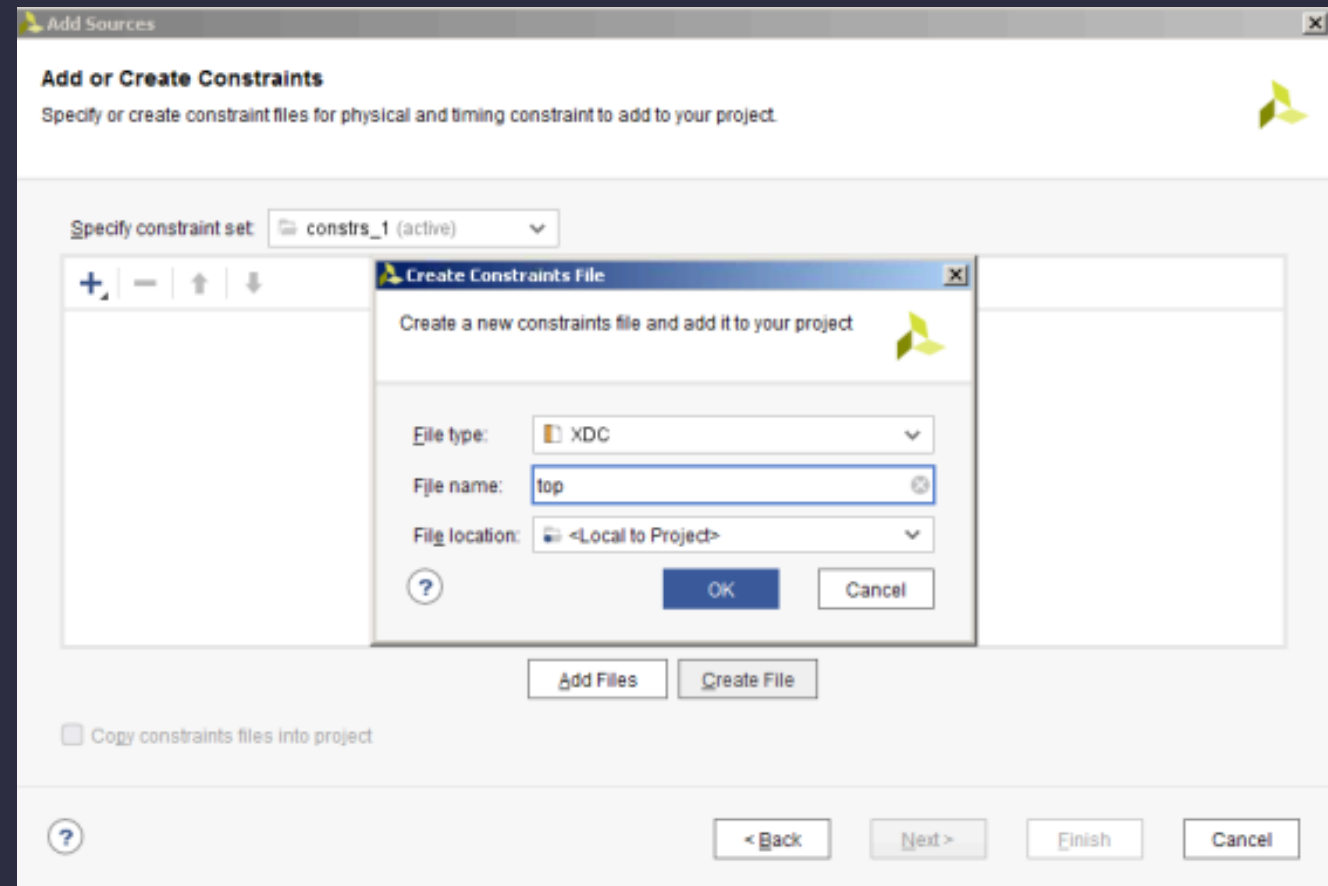
### Adding a constraint



# Verilog – Combinational Circuits

## Vivado Design Tool

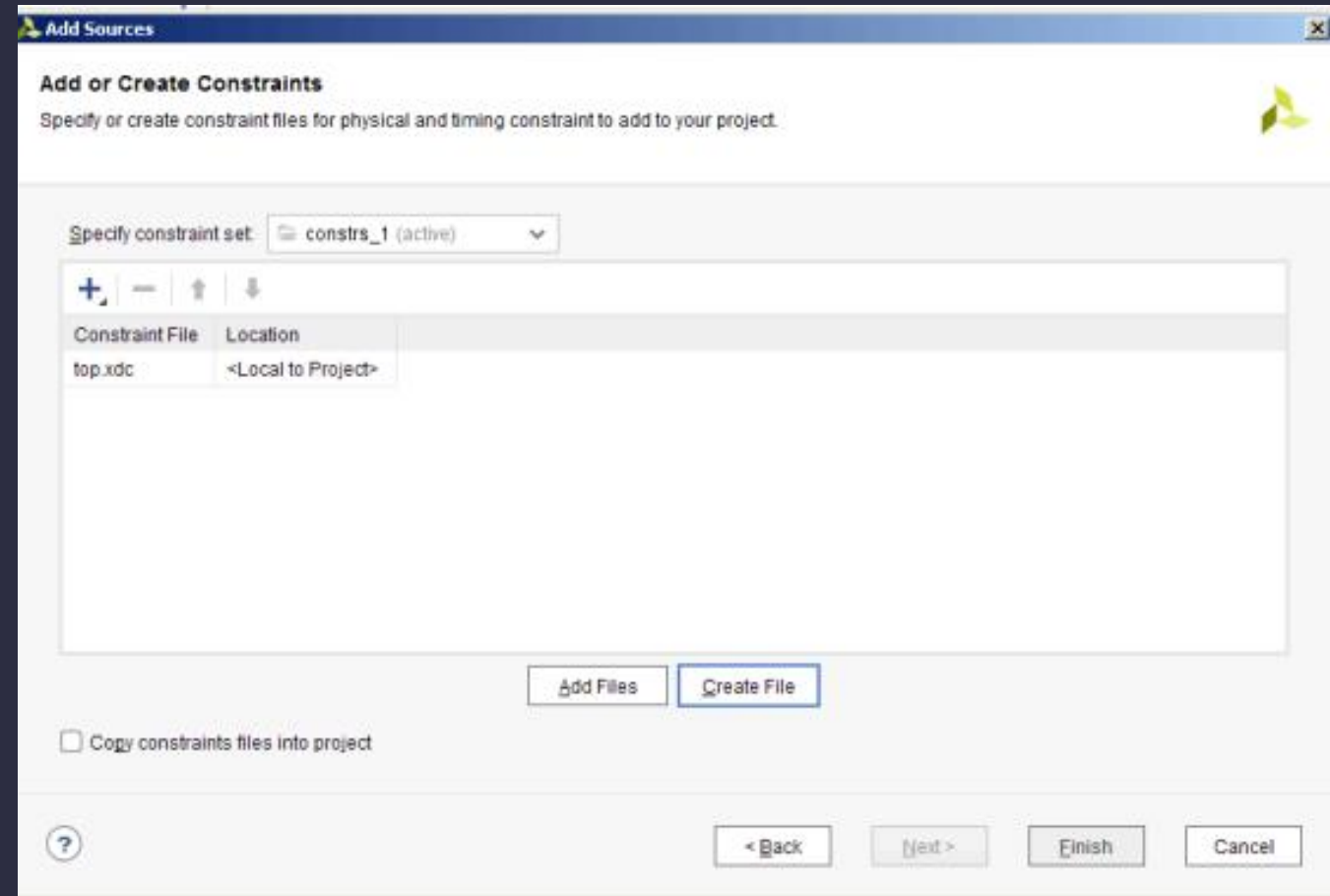
### Adding a constraint



# Verilog – Combinational Circuits

## Vivado Design Tool

### Adding a constraint

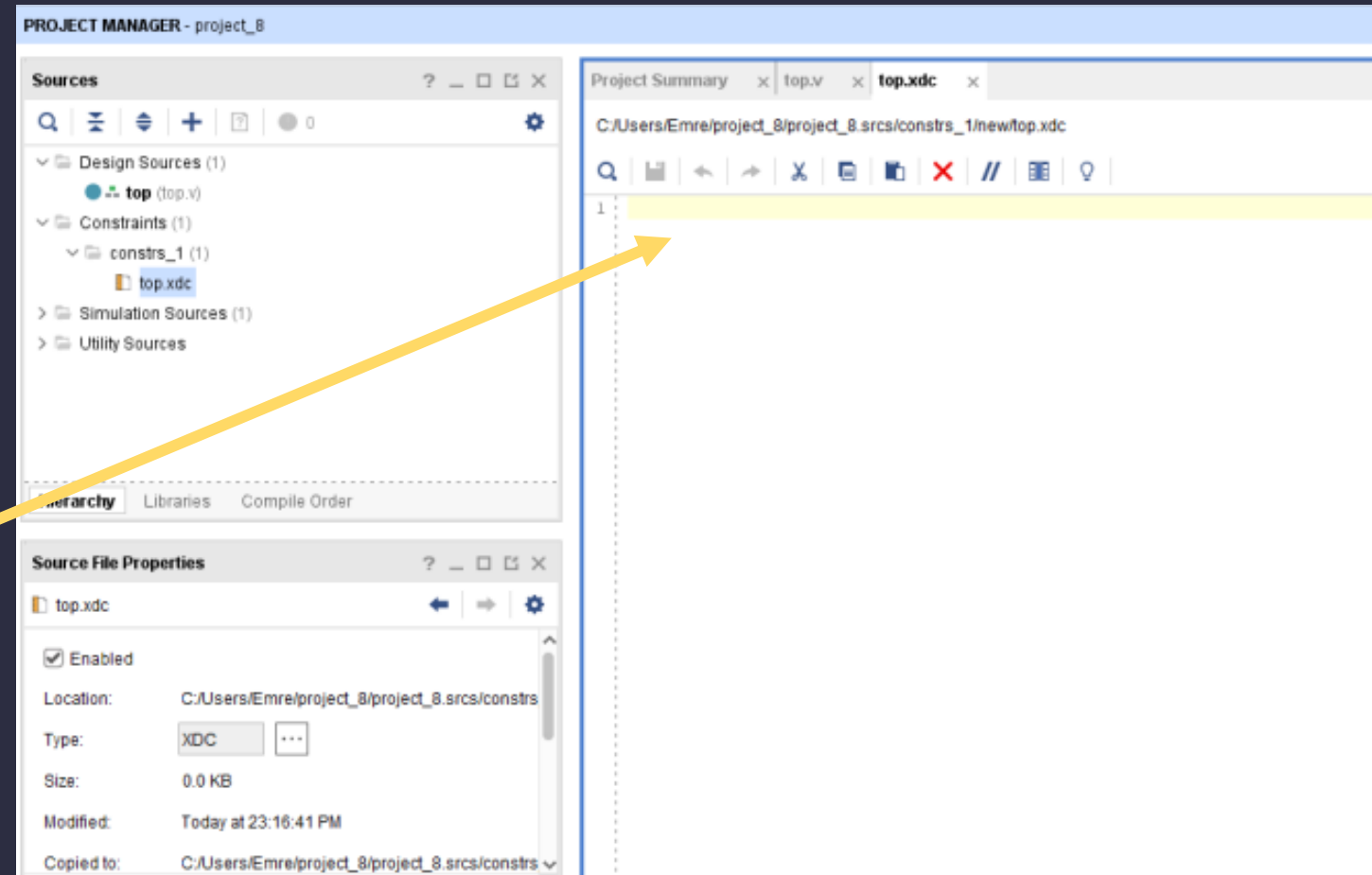


# Verilog – Combinational Circuits

## Vivado Design Tool

### Adding a constraint

### Write constraints



# Verilog – Combinational Circuits

Vivado Design Tool

Bitstream generation

▼ PROGRAM AND DEBUG

 [Generate Bitstream](#)

▼ Open Hardware Manager

Open Target

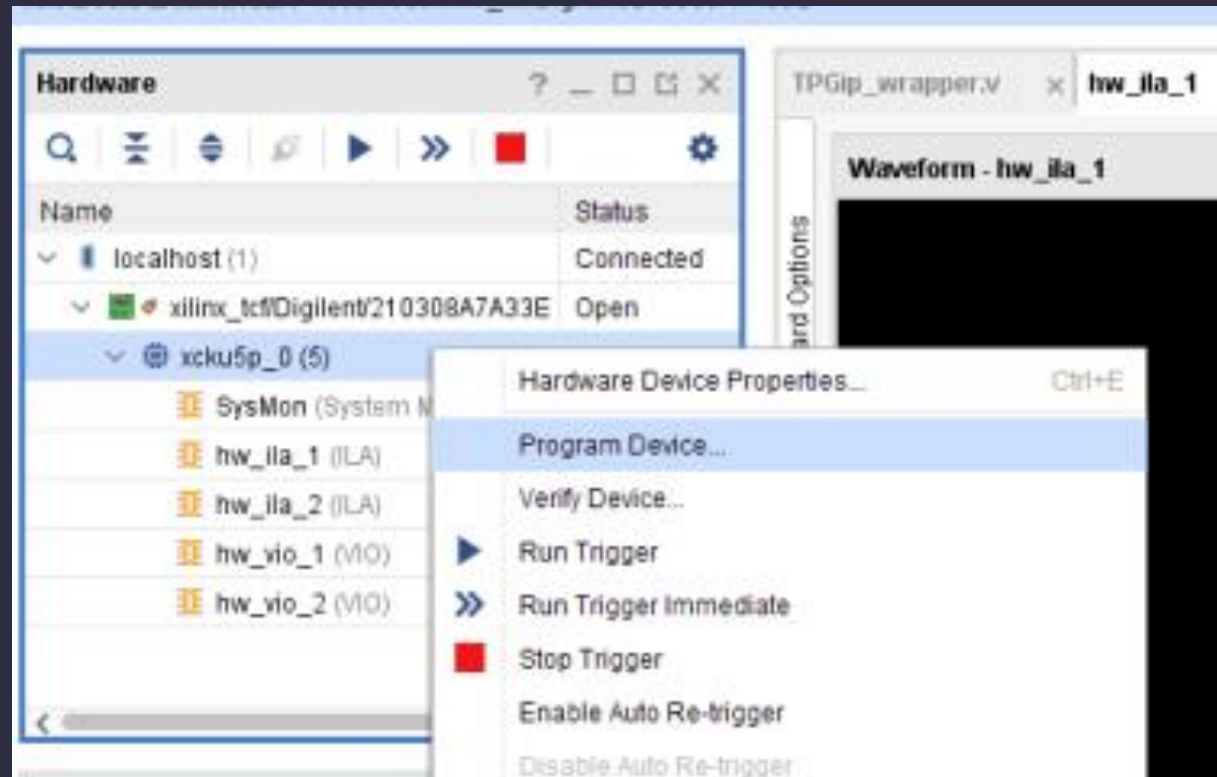
Program Device

Add Configuration Memory Devi

# Verilog – Combinational Circuits

## Vivado Design Tool

## Bitstream generation



# Verilog – Combinational Circuits

## Vivado Design Tool

## Bitstream generation

