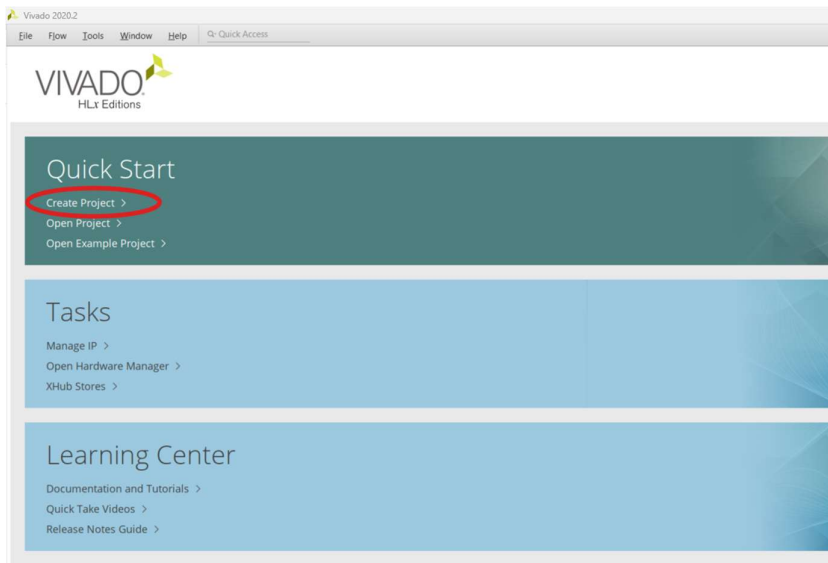



# **Getting started with Vivado design suite for EISLER Artix-7 FPGA Board**



 **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: D:/project\_EISLER\_TEST

**Project Type**

Specify the type of project to create.

- ☒ **RTL Project**  
You will be able to add sources, create block designs in the IP Integrator, generate IP, run RTL analysis, synthesis, implementation, design.
- ☐ Do not specify sources at this time
- ☐ Project is an extensible Vitis platform
- ☐ **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.

**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 Directory

Add Files

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

Target language: VHDL

Simulator language: Mixed

**Add Source Files**

Look in: TEST.SRCS

- ☒ hex\_seven\_seg.vhd
- ☒ SEVEN\_SIGMENT.vhd

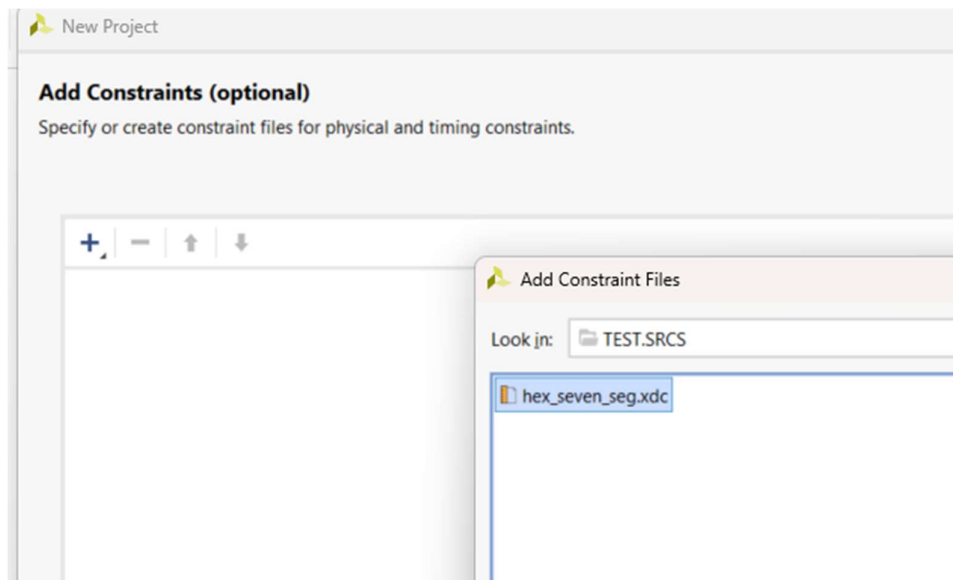
File name: "hex\_seven\_seg.vhd" "SEVEN\_SIGMENT.vhd"

Files of type: Design Source Files (.vhd, vhdI, vhf, vhdP, vho, v, vf, verilog, vr, vg, vb, tf, t

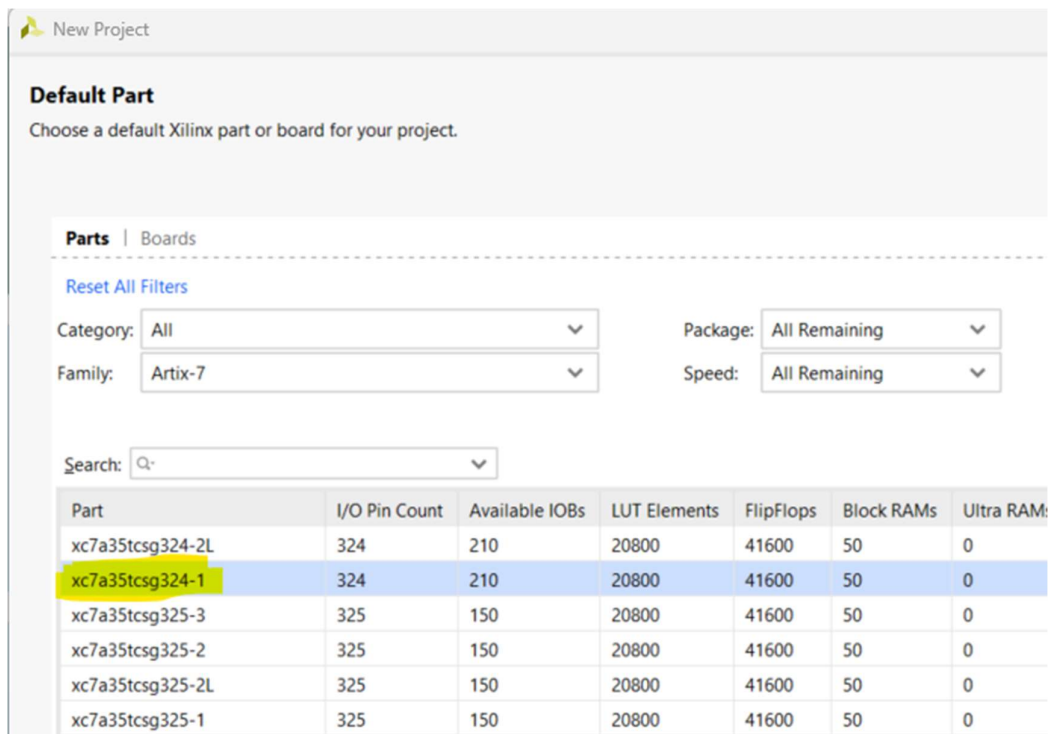
< Back

Next >

Finish



## FOR EISLER ARTIX 7 FPGA BOARD



## FOR PINE SPARTAN7 FPGA BOARD

[Reset All Filters](#)

Category:

Family:

Package:

Speed:

Search:  (3 matches)

Part	I/O Pin Count	Available IOBs	LUT Elements	FlipFlops
xc7s15ftgb196-1	196	100	8000	16000
xc7s15ftgb196-1IL	196	100	8000	16000
xc7s15ftgb196-1Q	196	100	8000	16000

File Edit Flow Tools Reports Window Layout View Help Q Quick Access

Flow Navigator PROJECT MANAGER - project\_2

PROJECT MANAGER

- Settings
- Add Sources
- Language Templates
- IP Catalog

IP INTEGRATOR

- Create Block Design
- Open Block Design
- Generate Block Design

SIMULATION

- Run Simulation

Sources

- Design Sources (1)
  - hex\_seven\_seg(BEHAVIORAL) (hex\_seven\_seg.vhd) (1)
    - INST\_SEVEN\_SEGMENT : SEVEN\_SEGMENT(BEHAVE) (SEVEN\_SIGMENT.vhd)
- Constraints (1)
  - constrs\_1 (1)
    - hex\_seven\_seg.xdc
- Simulation Sources (1)
- Utility Sources

Hierarchy Libraries Compile Order

The screenshot displays the Xilinx Vivado IDE interface. On the left, the **PROJECT MANAGER** pane shows a tree view with categories like Settings, IP Catalog, IP INTEGRATOR, SIMULATION, RTL ANALYSIS, SYNTHESIS, IMPLEMENTATION, and PROGRAM AND DEBUG. The **PROGRAM AND DEBUG** category is expanded, and the **Generate Bitstream** option is highlighted with a red circle.

The **Sources** window shows the project hierarchy: Design Sources (1) containing **hex\_seven\_seg(BEHAVIORAL) (hex\_seven\_seg.vhd) (1)**, Constraints (1) containing **constrs\_1 (1)**, Simulation Sources (1), and Utility Sources.

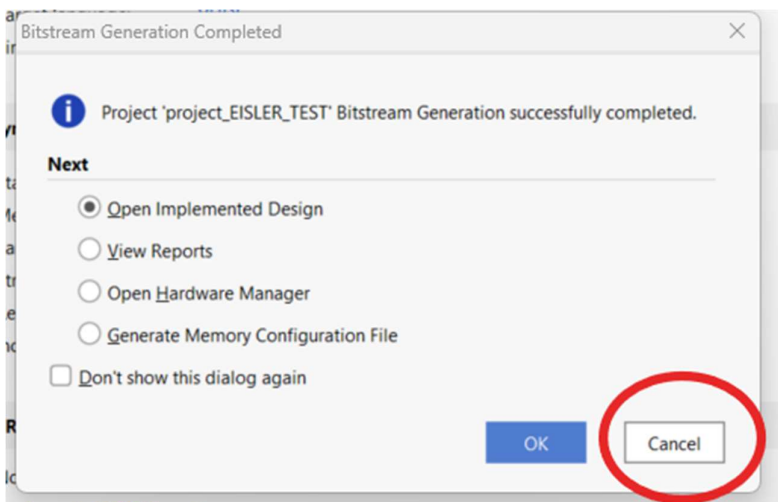
The **Source File Properties** window for **hex\_seven\_seg.vhd** shows the following details:

- Location:** D:/project\_2/project\_2\_srcs/sources\_1/imports/TEST.SRCS
- Type:** VHDL
- Library:** xil\_defaultlib
- Size:** 1.9 KB
- Modified:** Today at 13:17:49 PM

The **hex\_seven\_seg.vhd** code is displayed in the editor, showing a VHDL entity with ports and signals.

The **Design Runs** table at the bottom shows the status of the synthesis and implementation runs:

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAM	URAM	DSP	Start	Elapsed	Run Strategy
synth_1	constrs_1	Not started															Vivado Synthesis
impl_1	constrs_1	Not started															Vivado Implementation



Open Hardware Manager

**Make sure board is powered on & USB connector is connected to computer properly**

- ▼ IMPLEMENTATION
  - ▶ Run Implementation
  - > Open Implemented Design

- ▼ PROGRAM AND DEBUG
  - ⚙ Generate Bitstream
  - ▼ Open Hardware Manager
    - Open Target**
    - Program Device
    - Add Configuration Memory Device

Library: xil\_defaultlib ...

Size: 1.9 KB

Modified: Today at 13:17:49 PM

Copied to: D:/project\_FISLER\_TEST

General Properties

Tcl Console Messages Log Rep			
[Icons]			
Name	Constraints	Status	
✓ synth_1	constrs_1	synth_de	
✓ impl_1	constrs_1	write_bit	

File Edit Flow Tools Reports Window Layout View Help

[Icons]

Dashboard

Flow Navigator

▼ PROJECT MANAGER

- ⚙ Settings
- Add Sources
- Language Templates
- 🔍 IP Catalog

▼ IP INTEGRATOR

- Create Block Design
- Open Block Design
- Generate Block Design

▼ SIMULATION

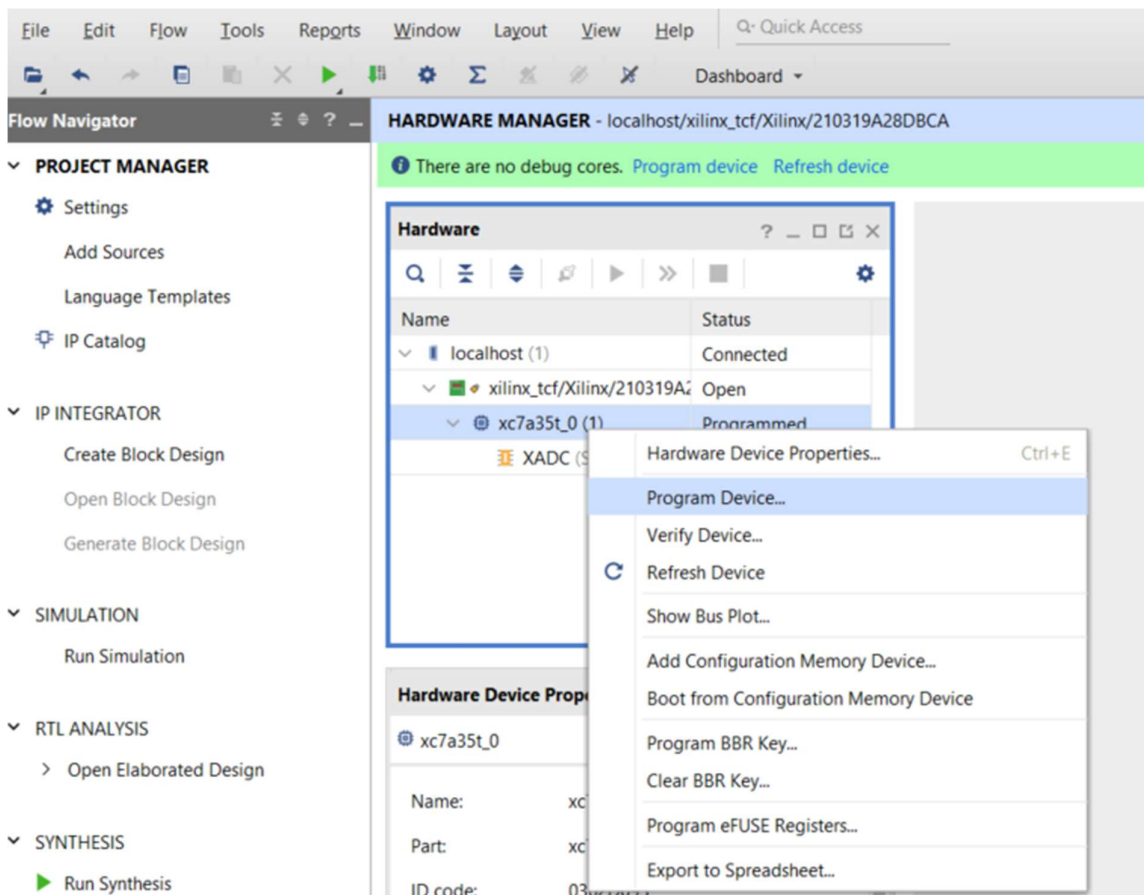
HARDWARE MANAGER - localhost/xilinx\_tcf/Xilinx/210319A28DBCA

There are no debug cores. Program device Refresh device

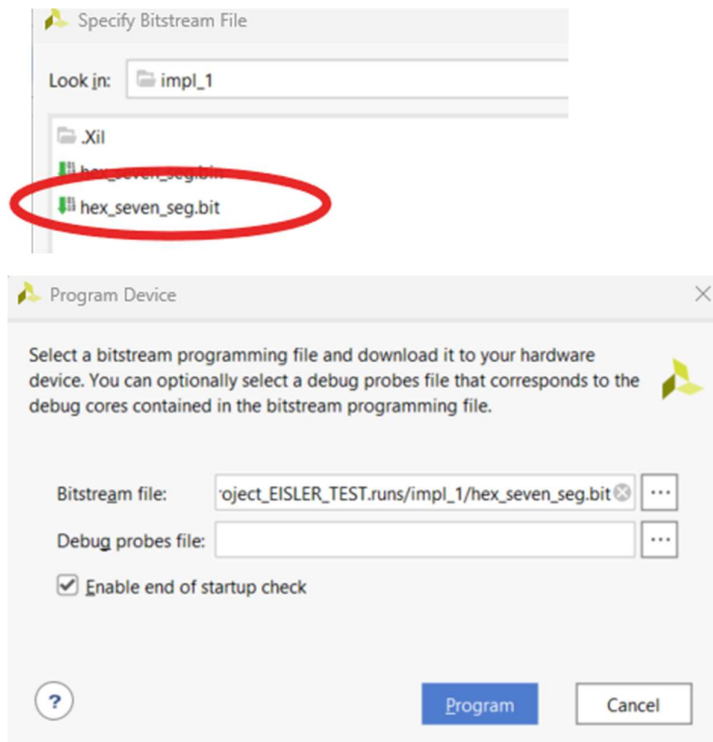
Hardware

[Icons]

Name	Status
localhost (1)	Connected
▼ xilinx_tcf/Xilinx/210319A28DBCA	Open
▼ xc7a35t_0 (1)	Programmed
XADC (System Monit	



In runs/impl/ folder



Now check output on board



## How To program flash attached to FPGA

The screenshot displays the Xilinx IDE interface. In the top-left pane, the 'PROGRAM AND DEPLOY' section is expanded, and 'Generate Bitstream' is selected. A yellow box highlights the 'Bitstream Settings...' button. The top-right pane shows the 'General' properties of the bitstream, including 'Library: xil\_defaultlib', 'Size: 1.9 KB', and 'Modified: Today at 13:17:49 PM'. The bottom pane shows the 'Bitstream' settings window. In the left sidebar, 'Project Settings' is expanded, and 'Bitstream' is selected, circled in red. The main area of the 'Bitstream' window shows a table of settings for 'Write Bitstream (write\_bitstream)'. The '-bin\_file' option is selected with a checkmark and circled in red. Below the table, the '-bin\_file' description is shown: 'Write a binary bit file without header (.bin)'.

**Bitstream Settings**

Specify various settings related to writing Bitstream

Note: Additional bitstream settings will be available once you open an implemented design.

Write Bitstream (write_bitstream)	
tcl.pre	...
tcl.post	...
-raw_bitfile	<input type="checkbox"/>
-mask_file	<input type="checkbox"/>
-no_binary_bitfile	<input type="checkbox"/>
<b>-bin_file</b>	<input checked="" type="checkbox"/>
-readback_file	<input type="checkbox"/>
-logic_location_file	<input type="checkbox"/>
-verbose	<input type="checkbox"/>
More Options	

**-bin\_file**  
Write a binary bit file without header (.bin).

