

# **Introduction to hardware design tools and methods**

Digital Circuits Lab (CS-220)

Spring '26

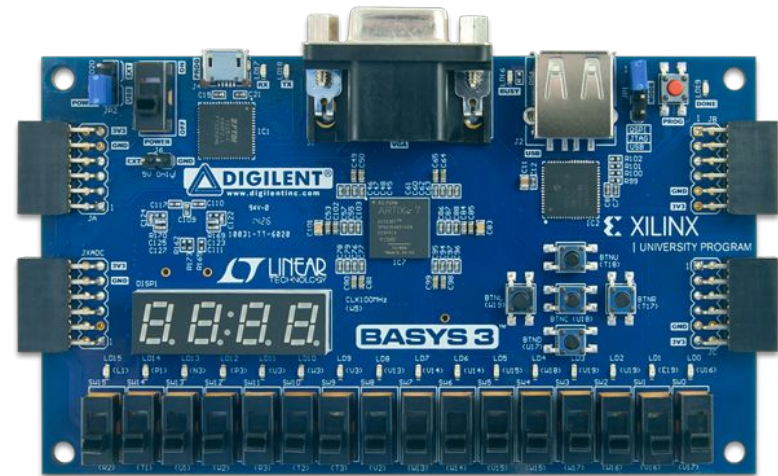
Sotiris Totomis & Konstantinos Sarioglou

# Overview

- Field Programmable Gate Arrays (FPGAs)
- Simulation
- Synthesis
- Implementation
- Tools installation
- Vivado tool flow & walkthrough
  - Simulation & Synthesis steps
- Verilator
  - Simulation steps
- Hands-on SystemVerilog demo
  - How to develop
  - How to debug

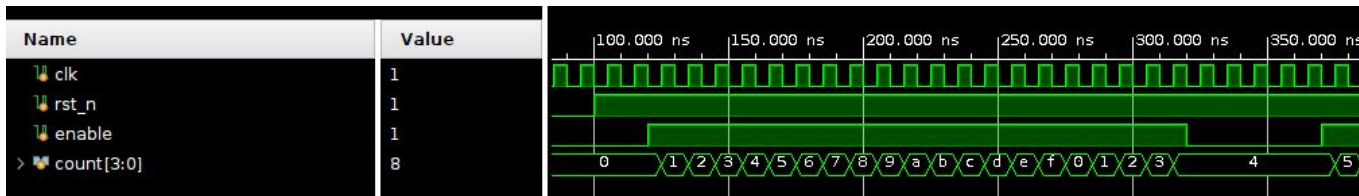
# Field Programmable Gate Arrays (FPGAs)

- FPGAs are programmable integrated circuits
- Consist of configurable logic blocks (CLBs)
  - Look-Up Tables (LUTs) for combinational logic
  - Flip-Flops (FFs) for sequential logic
  - Multiplexers (MUXs) for routing
- Advantages
  - Flexibility to reprogram for different applications
  - Lower development costs compared to ASICs
- They are used in various fields
  - Signal processing, aerospace, AI/ML
- FPGA vs. ASIC
  - ASIC is application specific (duh!)
  - FPGAs are better options for prototyping
- Some well-known vendors
  - **Digilent**
  - **Xilinx (AMD)**
  - Altera (Intel)



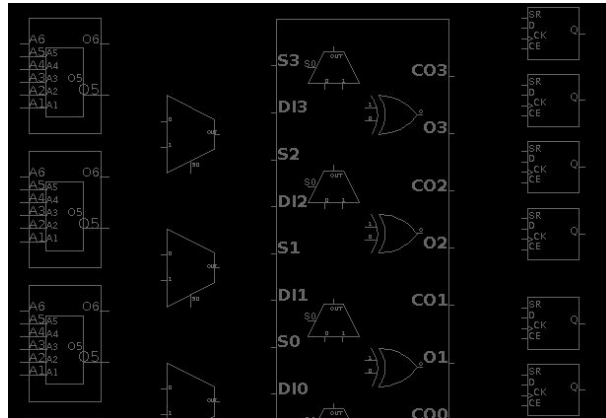
# Simulation

- The process of modeling and analyzing the behavior of digital circuits using software tools
  - A critical step in the design and verification before the hardware implementation (FPGA)
  - Register Transfer Level (RTL) simulation tests the code written in SystemVerilog
  - Simulation waveforms are our main debugging tool!
- Simulator alternatives
  - Vivado Design Suite includes a built-in simulator
    - Provides a graphical waveform for signal inspection
  - Verilator is an open-source simulator
    - Produces waveforms (vcd files), it requires a viewer to open them (GTKWave)
- Why do we need a testbench (TB) for simulation? TB is not synthesizable!



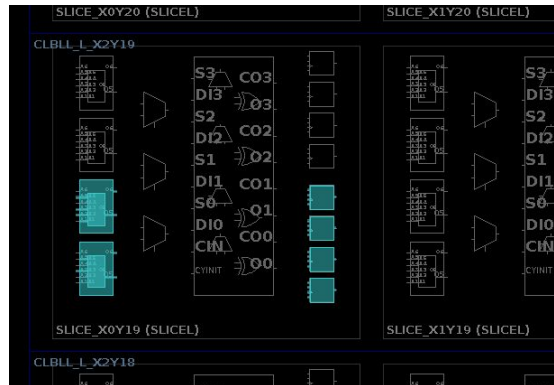
# Synthesis

- The process of converting high-level Hardware Description Language (HDL) code into gate-level netlist (logic gates, FFs, etc.)
- Reports
  - Space utilization
  - Timing & critical path
- What do we need to synthesize our code?
  - A top module, acts like a wrapper for our synthesizable code
    - It instantiates the synthesizable modules
    - It connects them to the FPGA's I/O pins via constraint files (XDC)



# Implementation

- The process of converting synthesized design into a physical layout that can be finally programmed onto an FPGA
- Places the design and routes the connections, meeting timing and resource constraints
- Timing, and congestion challenges - but not our problem for now!



# Tools installation

- Simulation options
  - Vivado, but it has high disk space requirements (~70 GB)
  - Alternatives for Linux/MacOS
    - Verilator + GTKWave (follow the site's instructions)
    - Icarus Verilog + GTKWave (follow the site's instructions)
  - You are able to use the department's machines for simulations
    - Don't forget `ssh -X` for display forwarding!
- For the FPGA flow (synthesis & implementation) we will use only Vivado during lab hours

# Vivado installation - Version/OS/Account

## Version

2024.2

2024.1

2023.2

Vivado Archive

ISE Archive

CAE Vendor Libraries  
Archive

AMD Unified Installer for FPGAs & Adaptive SoCs 2024.2: Windows Self Extracting Web Installer (EXE - 222.91 MB)

MD5 SUM Value : 84ee3816d8cbd0e0e38a1f9ac7eaa780

Download Verification 

Digests

Signature

Public Key

AMD Unified Installer for FPGAs & Adaptive SoCs 2024.2: Linux Self Extracting Web Installer (BIN - 303.93 MB)

MD5 SUM Value : 20c806793b3ea8d79273d5138fbd195f

Download Verification 

Digests

Signature

Public Key



## Sign-In

E-mail Address

sototomis@gmail.com

Password

..... 

Sign in

OR

Create Account

# Vivado installation - Initial setup

AMD Unified Installer for FPGAs & Adaptive SoCs 2024.2 - Select Install Type

Select Install Type

Please select install type and provide your AMD.com E-mail Address and password for authentication.

User Authentication

Please provide your AMD user account credentials to download the required files. If you don't have an account, [please create one](#). If you forgot your password, you can [reset it here](#).

E-mail Address

Password

Download and Install Now

Select your desired device and tool installation options and the installer will download and install just what is required.

Download Image (Install Separately)

The installer will download an image containing all devices and tool options for later installation. Use this option if you wish to install a full image on a network drive or allow different users maximum flexibility when installing.

Copyright © 1986-2022 Xilinx, Inc. All rights reserved.  
Copyright © 2022-2025 Advanced Micro Devices, Inc. All rights reserved.

< Back Next > Cancel

AMD Unified Installer for FPGAs & Adaptive SoCs 2024.2 - Select Product to Install

Select Product to Install

Select a product to continue installation. You will be able to customize the content in the next page.

Vitis

Installs Vitis Core Development Kit for embedded software and application acceleration development on AMD platforms. Vitis installation includes Vivado Design Suite. Users can also install Vitis Model Composer to design for AI Engines and Programmable Logic in MATLAB and Simulink. There is an option to install Power Design Manager for power estimation of Versal, UltraScale+, and Kria products.

Vivado

Includes the full complement of Vivado Design Suite tools for design, including C-based design with Vitis High-Level Synthesis, implementation, verification and device programming. Complete device support, cable driver, and Document Navigator included. Users can also install Vitis Model Composer to design for AI Engines and Programmable Logic in MATLAB and Simulink. Users can select to install the Vitis Embedded Development which is an embedded software development package. There is an option to install Power Design Manager for power estimation of Versal, UltraScale+, and Kria products.

AMD Unified Installer for FPGAs & Adaptive SoCs 2024.2 - Select Edition to Install

Select Edition to Install

Select an edition to continue installation. You will be able to customize the content in the next page.

Vivado ML Standard

Vivado ML Standard Edition is the no-cost, device limited version of the Vivado ML Enterprise edition. Users can add Vitis Model Composer which is an AMD toolbox for MATLAB and Simulink to design for AI Engines and Programmable Logic. Users can select to install the Vitis Embedded Development which is an embedded software development package. If you have been using AMD System Generator for DSP, you can continue development using Vitis Model Composer. There is an option to install Power Design Manager for power estimation of Versal, UltraScale+, and Kria products.

Vivado ML Enterprise

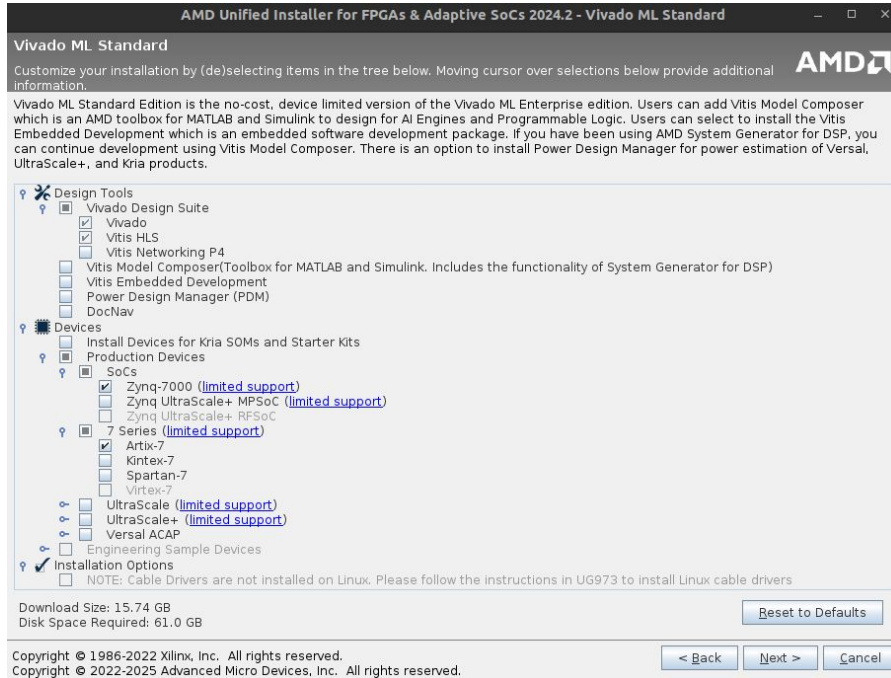
Vivado ML Enterprise Edition includes the full complement of Vivado Design Suite tools for design, including C-based design with HLS, implementation, verification, and device programming. Complete device support, cable drivers, and documentation Navigator are included. Users can add Vitis Model Composer which is an AMD toolbox for MATLAB and Simulink to design for AI Engines and Programmable Logic. Users can select to install the Vitis Embedded Development which is an embedded software development package. If you have been using AMD System Generator for DSP, you can continue development using Vitis Model Composer. There is an option to install Power Design Manager for power estimation of Versal, UltraScale+, and Kria products.

Copyright © 1986-2022 Xilinx, Inc. All rights reserved.  
Copyright © 2022-2025 Advanced Micro Devices, Inc. All rights reserved.

< Back Next > Cancel

*Linux: If you install Vivado in a system's directory, you have to execute the installer binary with sudo permissions*

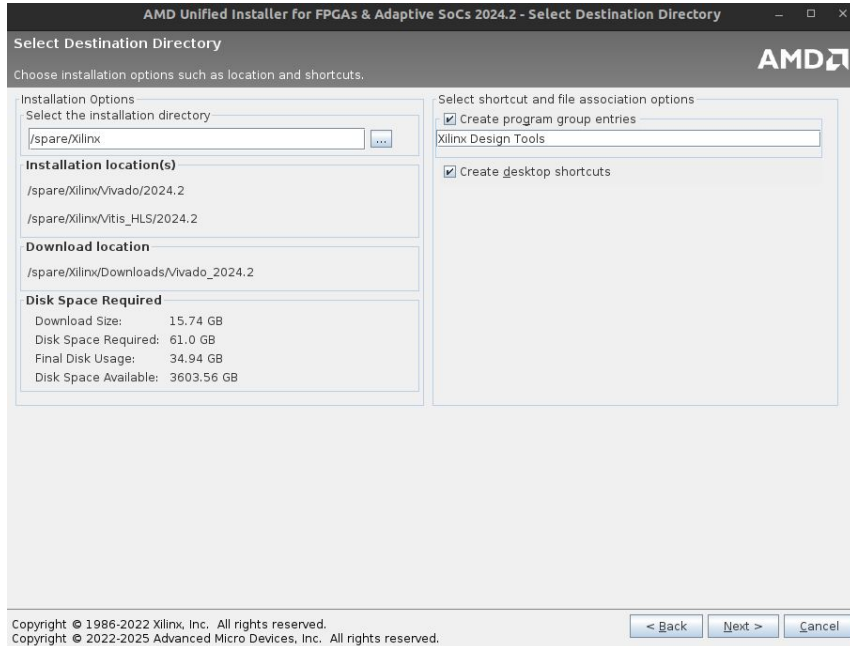
# Vivado installation - Devices & Basys 3 board files



*If you would like to test synthesis and implementation locally, you have to copy to your vivado directory the associated board files. Do this step **after** Vivado installation completes.*

<https://digilent.com/reference/software/vivado/board-files>

# Vivado installation - Final step, wait & run



```
sototo@rvila:/spare/Xilinx$ pwd
/spare/Xilinx
sototo@rvila:/spare/Xilinx$ cd Vivado/
sototo@rvila:/spare/Xilinx/Vivado$ ls
2024.2
sototo@rvila:/spare/Xilinx/Vivado$ cd 2024.2/
sototo@rvila:/spare/Xilinx/Vivado/2024.2$ source settings64.sh
sototo@rvila:/spare/Xilinx/Vivado/2024.2$ cd bin/
sototo@rvila:/spare/Xilinx/Vivado/2024.2/bin$ sudo ./vivado
```

# Vivado - Project creation

**AMD**  
Vivado  
ML Edition

Quick Start  
Create Project >  
Open Project >  
Open Example Project >

Tasks  
Manage IP >  
Open Hardware Manager >  
Vivado Store >

Learning Center  
Documentation and Tutorials >  
Quick Take Videos >  
What's New in 2024.2 >

**AMD**  
Vivado  
ML Edition

Create a New Vivado Project

This wizard will guide you through the creation of a new project.

To create a Vivado project you will need to provide a name and a location for your project files. Next, you will specify the type of flow you'll be working with. Finally, you will specify your project sources and choose a default part.

Back Next Finish Cancel

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

Project name: project\_1

Project location: /spare/Xilinx/Vvado/2024.2/bin

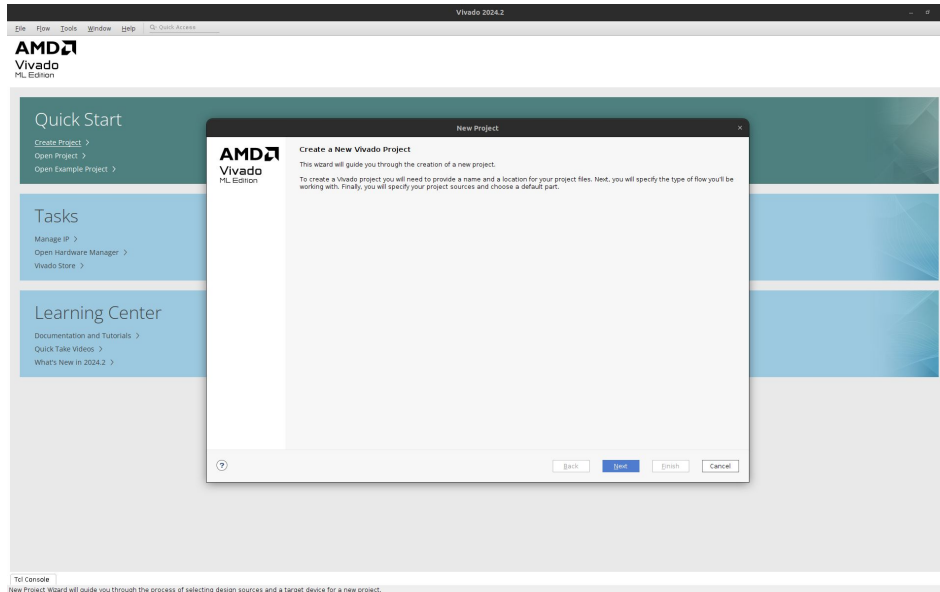
Create project subdirectory

Project will be created at: /spare/Xilinx/Vvado/2024.2/bin/project\_1

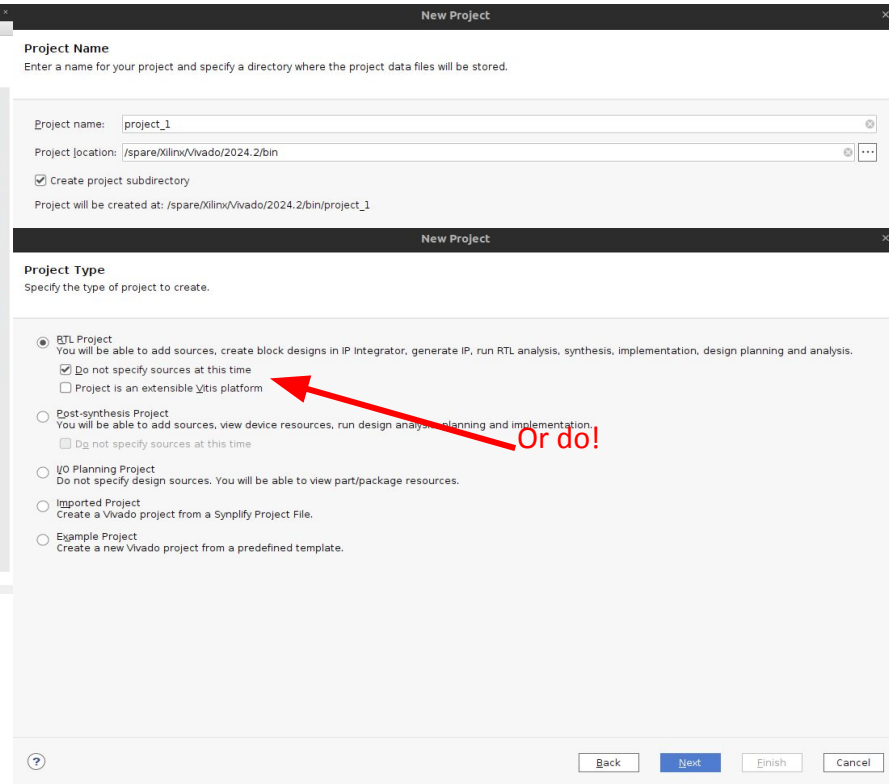
Back Next Finish Cancel

Tcl Console  
New Project Wizard will guide you through the process of selecting design sources and a target device for a new project.

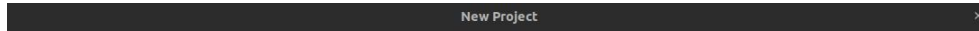
# Vivado - Project creation



Tcl Console  
New Project Wizard will guide you through the process of selecting design sources and a target device for a new project.



# Vivado - Import files & constraints



## 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.

	+	-	↑	↓					
					Index	Name	Library	HDL Source For	Location
					1	counter.sv	xil_defaultlib	Synthesis & Simulation	/home/sototo/Downloads/lab0_code
					2	lab0_tb.sv	xil_defaultlib	Simulation only	/home/sototo/Downloads/lab0_code
					3	lab0_top.sv	xil_defaultlib	Synthesis & Simulation	/home/sototo/Downloads/lab0_code



## Add Constraints (optional)

Specify or create constraint files for physical and timing constraints.

Constraint File	Location
lab0.xdc	/home/sototo/Downloads/lab0_code

**These are provided by us**

Copy constraints files into project

Scope Sources

- Design Sources (1)
  - top (top.sv) (1)
    - u\_counter : counter\_4bit (counter.sv)
- Constraints
- Simulation Sources (1)
  - sim\_1 (1)
    - counter\_tb (counter\_tb.sv) (1)
      - top\_instance : top (top.sv) (1)
        - u\_counter : counter\_4bit (counter.sv)
- Utility Sources

# Vivado - Board selection

New Project

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

Parts | Boards

Reset All Filters

Vendor: All Name: All Board Rev: Latest

Search: Q-

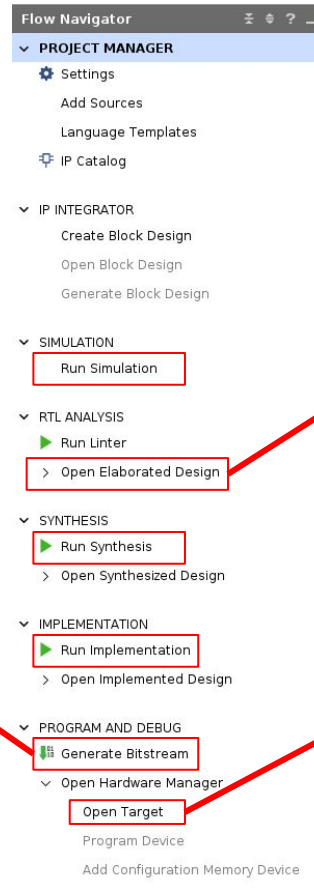
Display Name	Preview	Status	Vendor	File Version	Part	I/O Pin Count	Board Rev	Availab
Arty		Installed	digilentinc.com	1.1	xc7a35t1csg324-1L	324	C.0	210
Basys3		Installed	digilentinc.com	1.2	xc7a35t1cpg236-1	236	C.0	106
Cmod A7-15t		Installed	digilentinc.com	1.2	xc7a15t1cpg236-1	236	B.0	106
Cmod A7-35t		Installed	digilentinc.com	1.2	xc7a35t1cpg236-1	236	B.0	106
Cora Z7-07S		Installed	digilentinc.com	1.1	xc7z007sclg400-1	400	B.0	100

Refresh

Back Next Finish Cancel

<https://digilent.com/reference/software/vivado/board-files>

# Vivado - Flow Navigator



Displays an abstract schematic of your design


Generates the binary file that contains the configuration data to program an FPGA

Opens the connected FPGA device to program it. Don't forget to switch on the FPGA!

# Verilator

- ssh -X [csd\_host]
- Follow the site instructions
- Modify the Makefile prototype for each new simulation
- Must check simulator flags in source files!

```
SRCS=lab0_tb.sv lab0_top.sv counter.sv
TOP_MODULE=lab0_tb
#TOP_MODULE=example
HY220_TOOLS_DIR=~hy220/tools
include $(HY220_TOOLS_DIR)/common/Makefile.include
```



```
// increase with timer
counter #(
    .N(27),
    `ifdef XILINX_SIMULATOR
        .MAX(9)
    `else
        .MAX(99999999)
    `endif
)
timescale 1ns / 1ps
```

```
// increase with timer
counter #(
    .N(27),
    `ifdef XILINX_SIMULATOR
        .MAX(9)
    `else
        .MAX(9)
    `endif
)
timescale 1ns / 1ps
```

# Switch to demo

- Various module examples
- Hierarchy of files for design and simulation
- `always_comb` vs `always_ff`
- synchronous vs. asynchronous reset
- Debugging and waveforms
- Synthesis & implementation steps