

FPGAtors

SystemVerilog and Vivado Basics

An overview of SystemVerilog (HDL) and Vivado (design suite)

Lathan Price • 9/11/2026 • LAR 234

WORKSHOP

Today’s roadmap

An overview of SystemVerilog design and verification and Vivado simulation and synthesis

01 Overview	SystemVerilog, Vivado, 4-bit ALU overview
02 Verification Basics	What is verification and why is it important?
03 Simulation & Synthesis	Why are simulation and synthesis important in our designs?
04 Conclusion	Summary, Q&A, FPGAtors future events

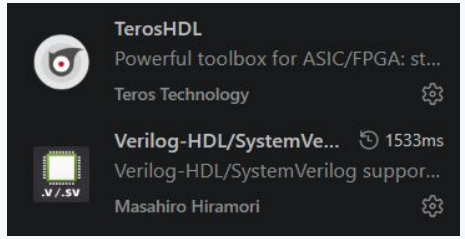
Tools Overview

What tools do you need to get started?

Tools used in this workshop:

- **Visual Studio Code**: Download Visual Studio Code - Free AI Code Editor for Mac, Linux, Windows

  Recommended Extensions:



- **Vivado**: Adaptive SoCs & FPGA Design Tools Downloads - 2026.1

***This will NOT be a code-along demo, but rather a walkthrough of existing code and simulations that have been run. alu.sv and alu_tb.sv files will be available after the workshop if you want to test the design for yourself using the tools linked above.**

SystemVerilog Overview

Software vs. SystemVerilog (Hardware Description Language)

Software that you may know

- Runs one line at a time
- CPU fetches, decodes, and executes instructions
- Order within the program matters
- The hardware is fixed, but software is able to be changed

SystemVerilog (HDL)

- Describes physical circuits
- Everything runs at once in parallel (there are some exceptions, for example inside an ‘initial’ or ‘always’ block)
- Wires and gates are being described, not ordered steps
- The FPGA physically reconfigures itself to become a design

SystemVerilog Overview

Code structure overview

Things you should know within the code

• **Module … endmodule** - A named block of hardware with a list of ports (inputs and outputs). For example, in our case, an ALU would be one module.
• **logic [3:0] name** - a 4-bit bus. [3:0] means bits 3 down to 0. A single bit is represented as **logic name**.
• **input / output** - The direction of a port signal – signals coming in vs. results going out
• **always_comb** - “When any input changes, recompute”. Describes combinational logic (pure gates, no memory, not clock-driven)
• **case (opcode)** - Pick one branch based on the value of opcode. For example, in the case of opcode being “00”, addition will occur. When the opcode is “01”, subtraction will occur, and so on.

Vivado Overview

Why do we use Vivado?

- Vivado is AMD/Xilinx's design suite
- Allows designs to be mapped onto AMD/Xilinx boards
- Other features of Vivado include:
    - Code editing
    - Simulation
    - Synthesis
    - Much more (we will not cover today)

CONCEPT

4-Bit ALU

Block diagram



CONCEPT

4-Bit ALU

Opcode and operation overview

Opcode	Operation	What it does
2’b00	Add	in1 + in2 + carry_in
2’b01	Subtract	in1 - in2 - carry_in
2’b10	Logical shift left	shift in1 left by 1 bit
2’b11	NOT	bitwise invert of in1

SystemVerilog 4-Bit ALU Walkthrough

CONCEPT

Verification Basics

Verification is the process of proving that your design does what it is intended to do before it reaches hardware

Why verification is important:

- Hardware bugs are expensive
- Simulation catches mistakes quickly instead of spending hours debugging synthesis or real hardware
- You can be sure that every case is tested, not just cases that you decide to try (more on this in the testbench walkthrough)
- In industry, verification is often a bigger effort than design

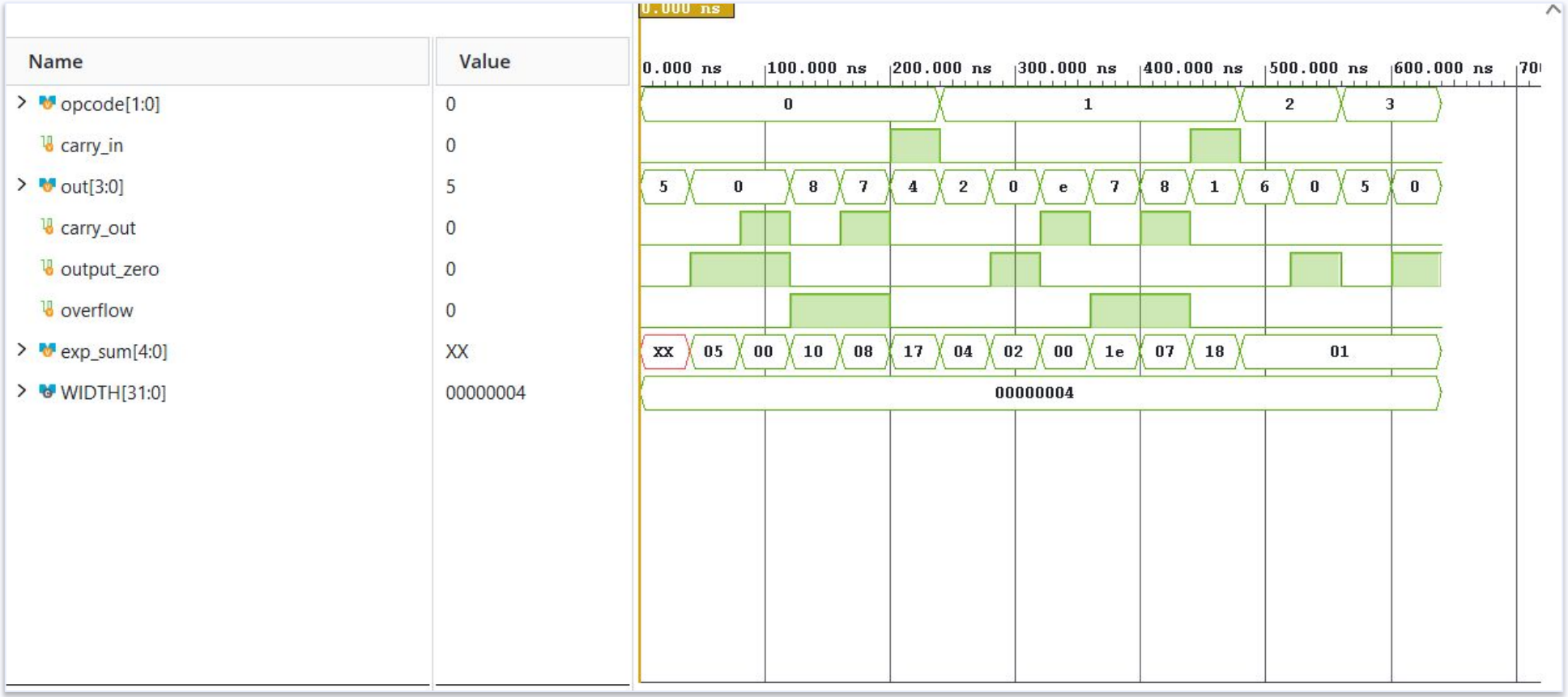
Testbenches:

- How we test designs
- Feeds the design inputs and checks the outputs against expected outputs
- Testbenches, like designs, can be designed in different ways depending on the purpose

SystemVerilog 4-Bit ALU Testbench Walkthrough

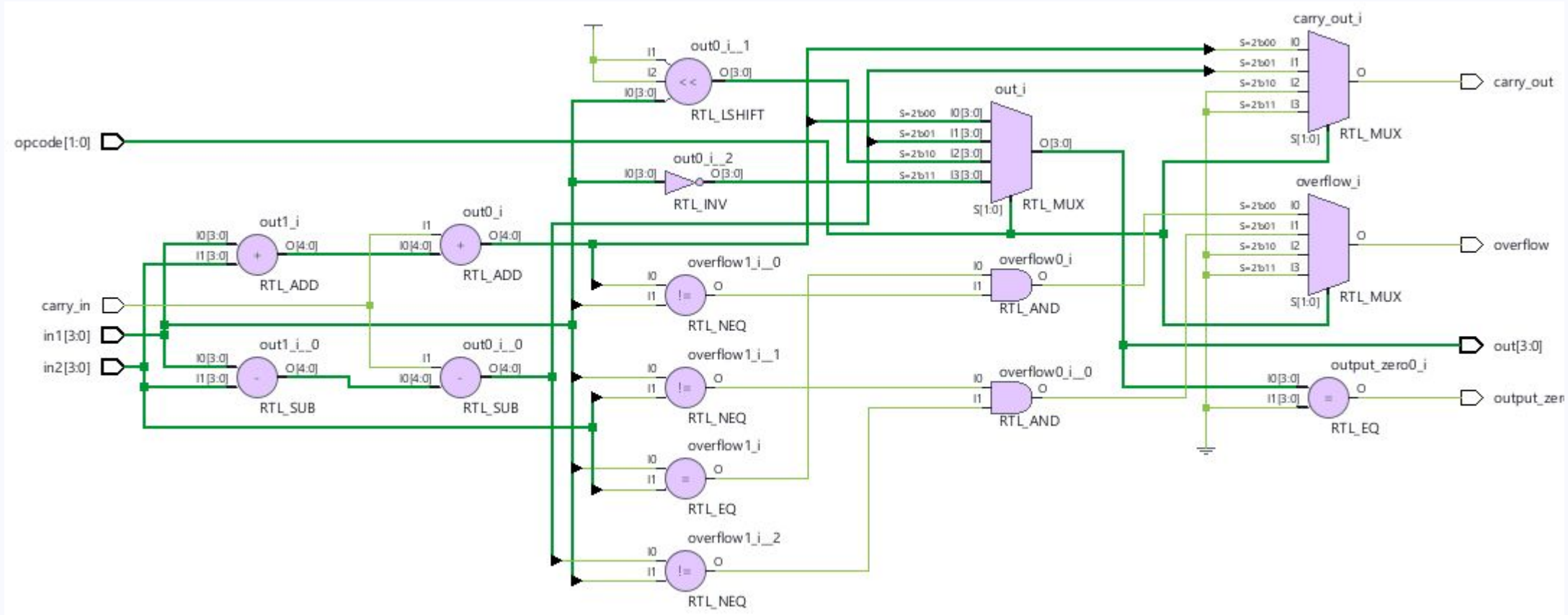
Vivado: Simulation

Simulation allows the designer to see what signals are changing and when



Vivado: Synthesis

Synthesizing designs allows the designer to see the design at the gate-level



Vivado Simulation and Synthesis Walkthrough

Summary

What was covered in this workshop?

- SystemVerilog is a HDL used to design digital circuits
- Vivado is a design suite used to simulate, synthesize, and test designs
- Verification is an important part of the design process; your design must work properly!
- Simulating designs allows the designer to see what signals are changing and when
- Synthesizing designs allows the designer to see the design at the gate-level

Conclusion

Please sign in if you haven't already!

Thank you for attending!

The workshops the rest of the semester will expand on the basic knowledge from the last two workshops and go into a variety of higher-level FPGA concepts

Questions?

