

# Digital Design & Computer Arch.

## Lab 6 Supplement: Testing the ALU

Prof. Onur Mutlu

ETH Zurich

Spring 2020

3 April 2020

# What Will We Learn?

---

- In Lab 6, you **learn** how to:
  - Verify the functionality of your designs using testbenches.
  - Find and resolve bugs in your design.
  
- You will:
  - Write a testbench that verifies the correctness of your ALU from Lab 5.
  - Use the same testbench to find and fix bugs in a buggy ALU that we provide.

# Preparation

---

- **You are expected to finish Lab 5 before continuing,** because we will be testing the ALU from Lab 5.
  
- Download the material for Lab 6, which includes:
  - A template **testbench file**;
  - A template for the **test-vectors**;
  - A Verilog description of an ALU, which **contains some bugs**.

# Part 1: Expected Results

---

- Before writing our testbench, we need to prepare a set of inputs for which the **expected results** are known.
- You will be given **a set of inputs** for the ALU you designed in Lab 5.
- Determine the **correct result** for each set. Then, specify them in the file **testvectors\_hex.txt** that we provide.
- **For output 'zero'**: directly set its expected value within the testbench

# Part 2: Preparing the Testbench

---

- Create a project with your ALU from Lab 5 and the testbench template we provided you with.
- Make the necessary modifications to the testbench.
- After this, you will have a testbench that will
  - Apply the vectors in the **testvectors\_hex.txt** file;
  - Check the **actual outputs** of our ALU against **what we expect**.

# Part 3: Simulating the ALU

- Run behavioral simulation using Vivado's built-in simulator.

The screenshot displays the Vivado 2019.2.1 interface during a behavioral simulation of an ALU. The left sidebar shows the Project Manager with sections for Settings, IP Integrator, Synthesis, and Implementation. The main workspace is divided into several panels:

- Scope:** A table listing design units and block types.
- Objects:** A table listing simulation objects and their values.
- Waveform:** A signal viewer showing the timing of various signals.
- Tcl Console:** A window showing simulation messages and timing information.

Name	Design Unit	Block Type
ALU_test	ALU_test	Verilog Module
uut	ALU	Verilog Module
glbl	glbl	Verilog Module

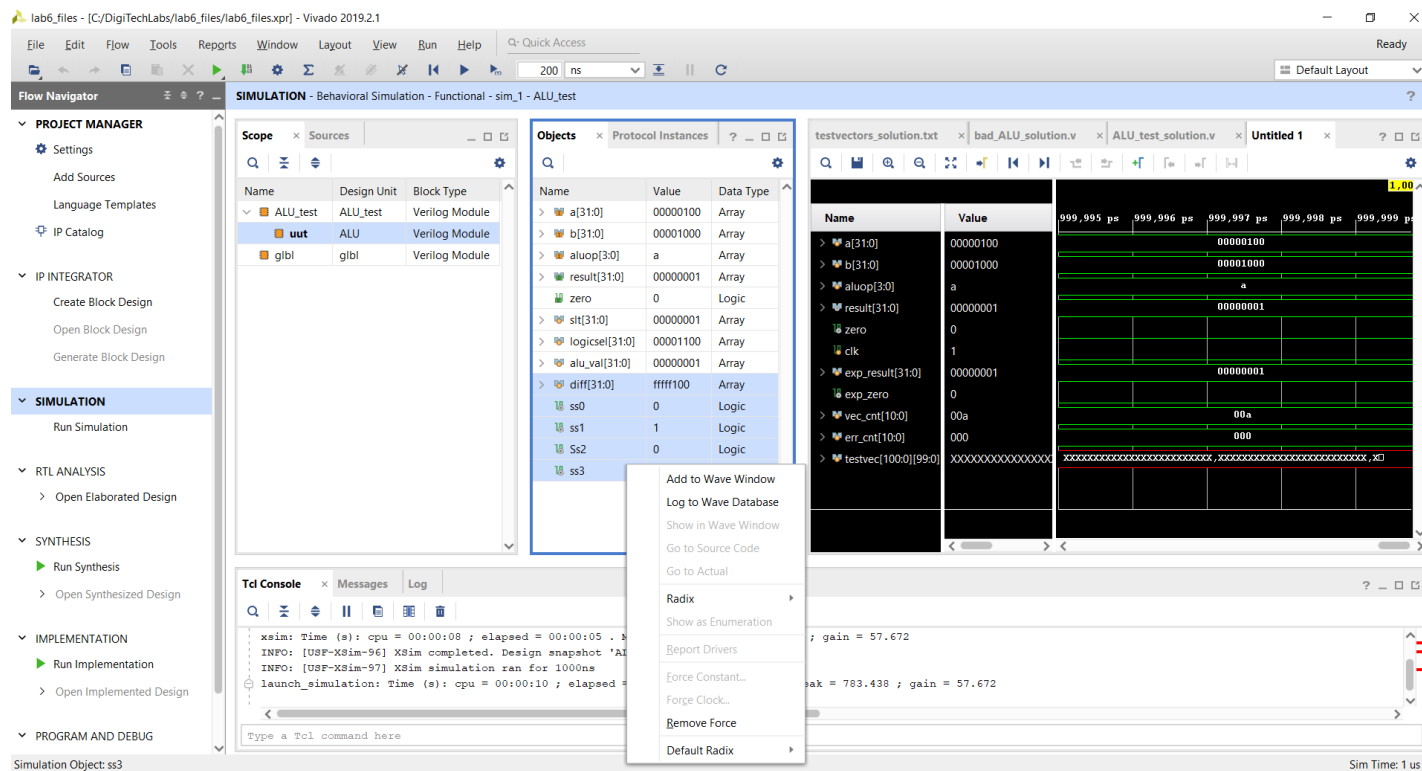
Name	Value	Data Type
a[31:0]	00000100	Array
b[31:0]	00001000	Array
aluop[3:0]	a	Array
result[31:0]	00000001	Array
zero	0	Logic
clk	1	Logic
exp_result[31:0]	00000001	Array
exp_zero	0	Logic
vec_cnt[10:0]	00a	Array
err_cnt[10:0]	000	Array
testvec[100:0][5]	XXXXXXXXXX	Array

```
xsim: Time (s): cpu = 00:00:08 ; elapsed = 00:00:05 . Memory (MB): peak = 783.438 ; gain = 57.672
INFO: [USP-XSim-96] XSim completed. Design snapshot 'ALU_test_behav' loaded.
INFO: [USP-XSim-97] XSim simulation ran for 1000ns
launch_simulation: Time (s): cpu = 00:00:10 ; elapsed = 00:00:17 . Memory (MB): peak = 783.438 ; gain = 57.672
```

1. Errors
2. Display messages

# Part 4: Debugging the Problem

- Using a simulator can help you locate the problems in your circuits.
- You can not only observe the outputs but the state of all internal variables as well.



# Last Words

---

- In Lab 6, you learn how to
  - write testbenches in Verilog to verify the functionality of the design.
  - Find and resolve bugs in your design
- Write a testbench that verifies the correctness of your ALU from Lab 5.
- Use the same testbench to find and solve bugs in a buggy ALU that we provide.
- In the report, you will design a testbench for your FSM from Lab 4.



# Report Deadline

---

**23:59, 1 May 2020**

# Digital Design & Computer Arch.

## Lab 6 Supplement: Testing the ALU

Prof. Onur Mutlu

ETH Zurich

Spring 2020

3 April 2020