THE UNIVERSITY OF KANSAS

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

EECS 645 – Computer Architecture

Spring 2023

Homework 01

Student Name: Student ID:

Resolving Type Mismatch using Conversion Functions

The purpose of this homework assignment is to master using the conversion functions of IEEE's ***numeric_std*** package (refer to the provided file "VHDL_1164pkg.pdf") to resolve type mismatches in VHDL. For that purpose, we will describe in behavioral VHDL a modulo-m up/down counter with the following interface:

- Generics
  - Modulo base (*m* with default value of 16)
- Inputs
  - Clock (clk → 1 bit)
  - Asynchronous reset (rst → 1 bit)
  - Counting direction (up_down → 1 bit)
    - 1 → Counting up,
    - 0 → Counting down
- Outputs
  - Count value (count → $\lceil \log_2(m) \rceil$ bits)

- The *mod-m up/down counter* should work as follows:
  - When counting up (up_down = '1'), it starts from 0, 1, 2, ..., m-1 (e.g., when m = 27 as in the provided testbench, then it should count 0, 1, 2, ..., 26 and then loops back to 0, 1, and so on)
  - When counting down (up_down = '0'), it starts from m-1, m-2, ..., 2, 1, 0 (e.g., when m = 27 as in the provided testbench, then it should count 26, 25, ..., 2, 1, 0 and then loops back to 26, 25, and so on)

- In Vivado
  - Create a project
  - Add design and simulation source files
  - Run behavioral simulation
  - Your waveform configuration should be identical to the provided waveform snapshot, see Figure 1.
- Steps:
  1) Download the file "HW01_Type_Mismatch.zip" from Canvas and extract its contents.
  2) Rename the folder "HW01_Type_Mismatch" to "HW01_Type_Mismatch_<your last name>", for example "HW01_Type_Mismatch_El-Araby".
  3) Launch Vivado and create a new project, for example "vivado_project", with the default settings under the following directory "\HW01_Type_Mismatch_<your last name>\mod_m_up_dn_count\" resulting in the following project directory "\HW01_Type_Mismatch_<your last name>\mod_m_up_dn_count\vivado_project\"
  4) Add to the project the VHDL design and simulation source files from the folders; "\HW01_Type_Mismatch_<your last name>\mod_m_up_dn_count\design_sources" and "\HW01_Type_Mismatch_<your last name>\mod_m_up_dn_count\simulation_sources" respectively.
  5) Edit the VHDL file in the folder "\HW01_Type_Mismatch_<your last name>\mod_m_up_dn_count\design_sources\" according to your design such that it describes the required *modulo-m* up/down counter.
  6) Set the simulation time to the proper time, e.g., ***1000 ns***, and then launch Vivado Simulator.
  7) Verify the correctness of your design. Your waveform configuration should be identical to the waveform snapshot shown in Figure 1. You may go back to step 5 to correct your code until your design works properly as required.
  8) After you are done, compress the folder "\HW01_Type_Mismatch_<your last name>" to "HW01_Type_Mismatch_<your last name>.zip", for example "HW01_Type_Mismatch_El-Araby.zip" and upload it to Canvas before the due date and time.

Grade Distribution:

- Functional Correctness, i.e., correct source code → 75 / 100
- Proper Setup of Vivado Project → 25 / 100

NOTE:

Homework submission is a "Single Attempt", i.e., carefully review everything that you want to submit before hitting the "submit" button and make sure that you have uploaded all documents you want to submit and have not missed anything.

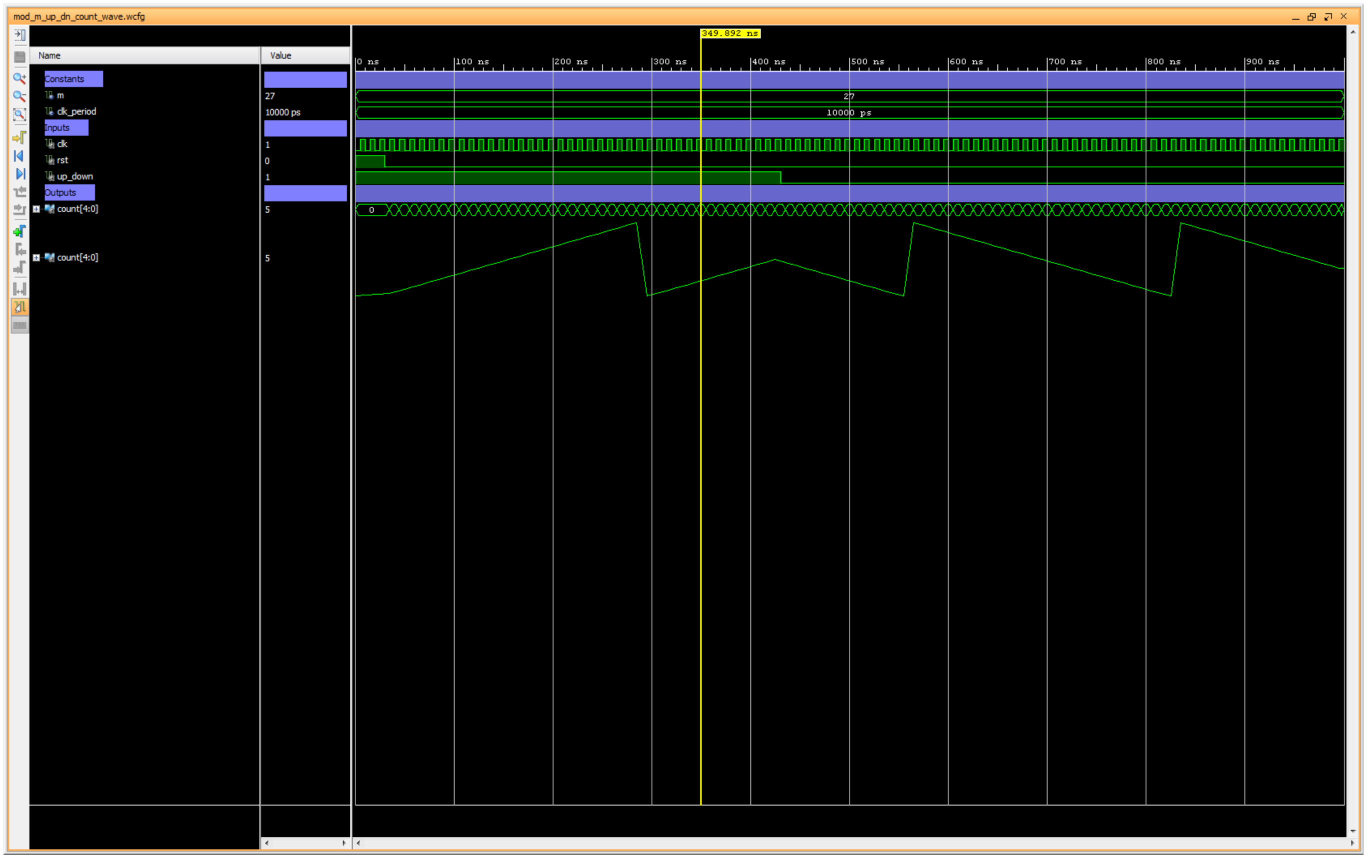

Figure 1. Snapshot of Correct Waveform Configuration