CPU Architecture

LAB2 preparation report

VHDL part2

Sequential code and Behavioral modeling

Hanan Ribo

3/04/2025

Table of Contents

1. Aim of the Laboratory 3
2. System Design definition 3
3. Useful names extension 5
4. Test and Timing: 5
5. Requirements 5
6. Grading Policy 7

1. Aim of the Laboratory

- Obtaining skills in VHDL part 2 code, which contains Sequential code and Behavioral modeling.
- Obtaining basic skills in ModelSim (multi-language HDL simulation environment).
- knowledge of digital systems design.
- Proper analysis and understanding of architectural design.

2. System Design definition

In this laboratory, you will design a synchronous digital system that implements a dynamically growing N-modulo counter where N increases dynamically each counting round until it reaches the ***UpperBound*** value, where the repeat input equals 0, at which point it resets to the beginning, where the repeat input equals 1. The system module is depicted in Figure 1. The system block diagram is shown in Figure 2. You are required to design the system and create a test bench for testing.

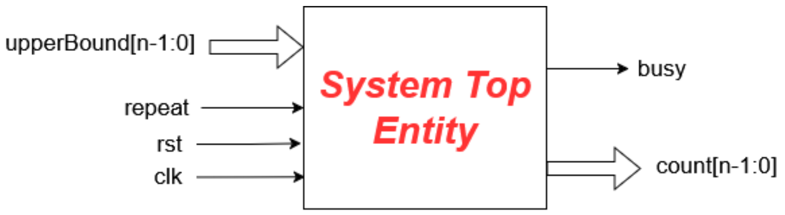

Figure 1: System module inputs and outputs

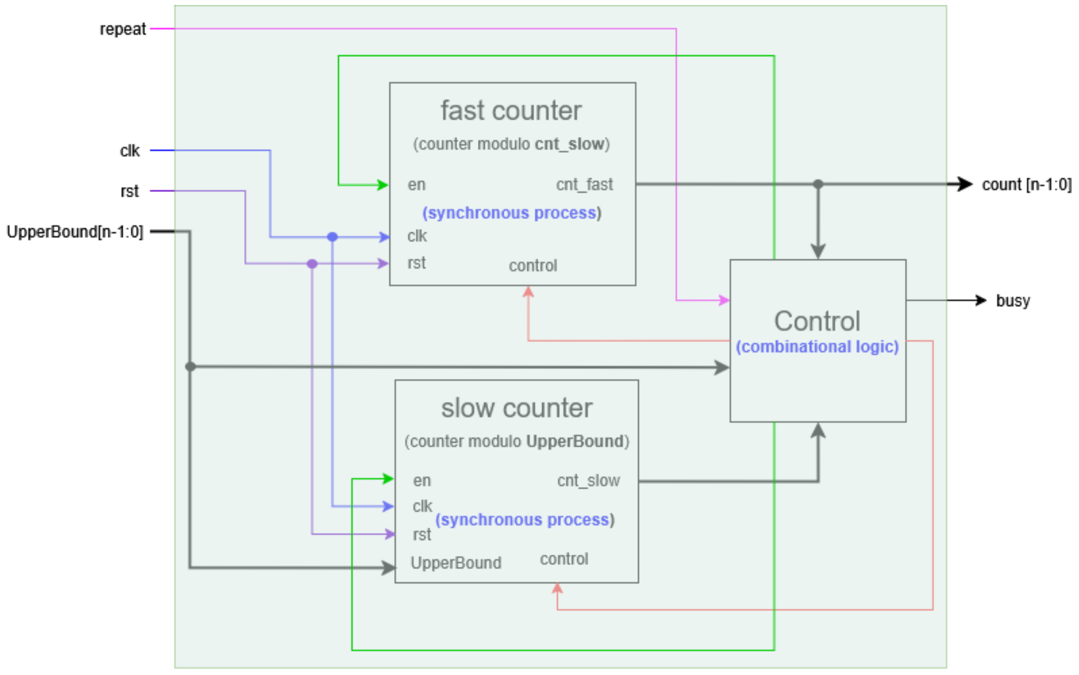

Figure 2: System module block diagram

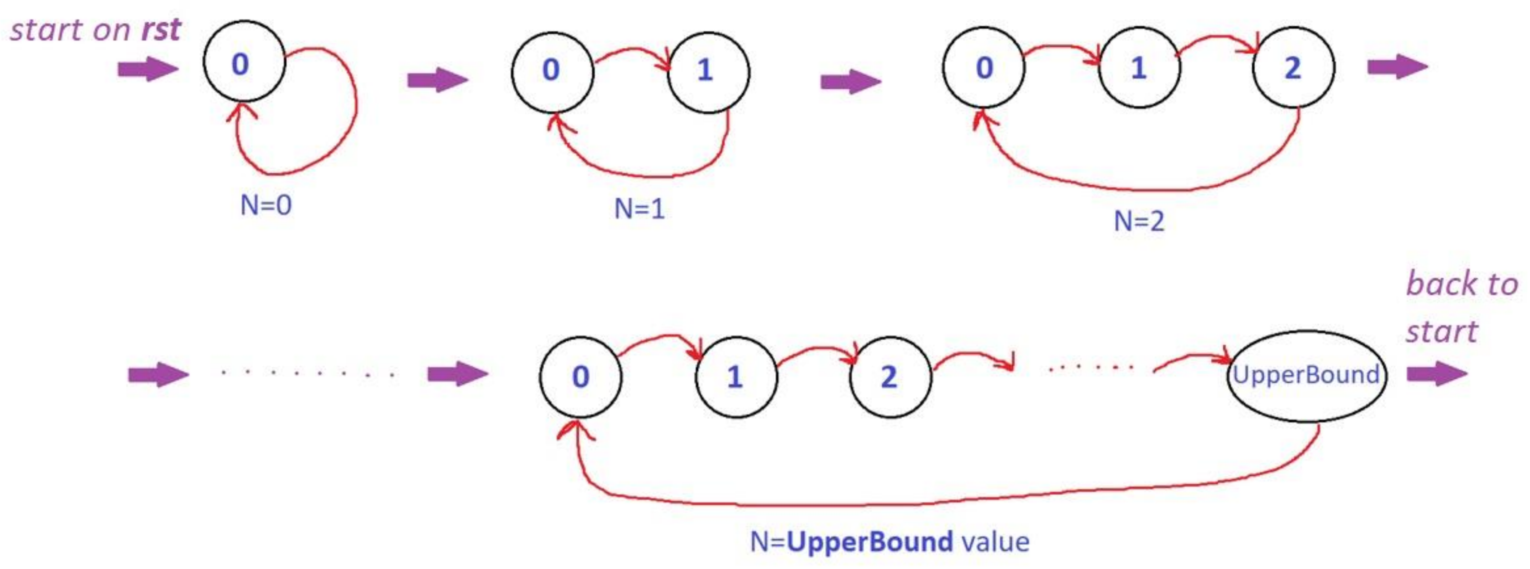

Figure 3: System structure and transitions diagram

- The Top-Level design modeling should be Behavioral.
- You are given the above two files <u>that you must use ***only*** in your project</u>: **top.vhd**, **aux_package.vhd (**you must use the given Entity definitions and can only add your code to these files, ***you are not allowed to erase anything*****).** The given **top.vhd** contains two PROCESS' templates that you can use ***<u>only</u>*** *(i.e., you cannot add more PROCESS but concurrent code)*.
- The generic n value must be verified for 8,16,32 (set from the **tb.vhd** file).
- The submitted project must be compiled using the given **tb.vhd** file (otherwise, the submitted project will be considered a failure).
- The submitted assignments pass through the copy checking machine; in this case, both sides' assignments will be disqualified.
- **<u>Three Examples</u>:**

  Example tb_1 - <u>where the ***UpperBound*** value remains unchanged.</u>

  Example tb_2 - <u>where the ***UpperBound*** value is changed throughout time.</u>

  Example tb_3 – <u>where the **rst** and **UpperBound** values are changed throughout time.</u>

Figure 4: Examples using waveform images, and list files

3. Useful names extension

- The very recommended use of Entity ports names is as follows:
  i. The names of input ports have i extension, that is, **port_name_i**
  ii. The names of output ports have o extension, that is, **port_name_o**

- The very recommended use of SIGNALS names is as follows:
  i. The name of signal used as wire (connection only) has w extension, that is, **signal_name_w**
  ii. The name of the signal used as a synchronous register (has a synchronous memory) has q extension, that is, **signal_name_q**
  iii. The name of the signal used as an asynchronous/combinational register (has an asynchronous memory) has r extension, that is, **signal_name_r**

- The very recommended use of VARIABLE names is as follows:
  The name of the variable has v extension, that is, **variable_name_v**

4. Test and Timing:

- Design a test bench which tests all the system.
- Analyze the results by zooming on the important transactions in the waveforms. explain these (input/output/internal signals of the system).
- You are welcome to use the Internet also as reference.
- The timing of the system will be ideal (means a functional simulation).

5. Requirements

a. The lab assignment is in pairs (as shown in the inlay file).
b. The design must be well commented.
c. **Important:** For each of two submodules:
   - Graphical description (a square with ports going in and out) and short descriptions.

d. Elaborated analysis and wave forms:
   - Remove irrelevant signals.
   - Zoom on regions of interest.
   - Draw clouds on the waveform with explanations of what is happening (Figure 4).

- Change the waveform colors in ModelSim for clear documentation

  (Tools->Edit Preferences->Wave Windows).

e. A ZIP file in the form of **id1_id2.zip** (where id1 and id2 are the identification number of the submitters, and id1 < id2) *must be upload to Moodle only by student with id1* (any of these rule's violation disqualifies the task submission).

f. The **ZIP** file will contain (***only the exact next sub folders***):

Directory	**Contains**	**Comments**
DUT	Project VHDL files	**Only VHDL files of DUT**, excluding test bench **Note: your project files must be well compiled without errors as a basic condition before submission**
TB	VHDL files that are used for a test bench	A single representative ***tb.vhd*** (your version) of the system top verification
SIM	DO files of wave and list forms	To pairs of DO file (one for wave and one for list)
DOC	Project documentation	• ***readme.txt*** (list of the DUT **.vhd* files with their brief functional description) • ***pre1.pdf*** (report file that includes brief explanation of the top module with its wave diagrams)

Table 1: Directory Structure

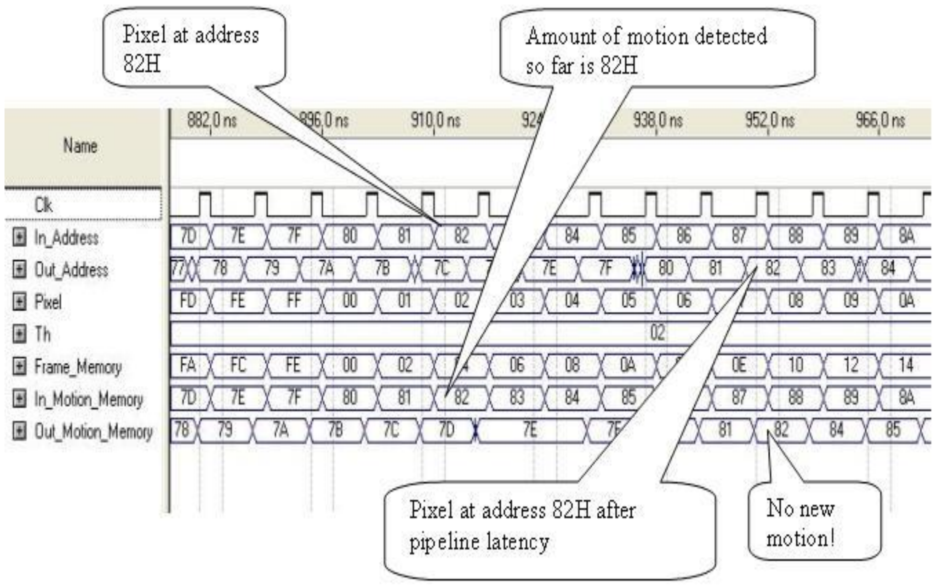

Figure 5: Clouds over the waveform example

6. Grading Policy

Weight	Task	Description
3%	Correct submission	The required four folders are the immediate subfolders under the submission id1_id2.zip folder
7%	Documentation	The "clear" way in which you presented the requirements and the analysis, and conclusions on the work you've done
90%	Analysis and Test	The correct analysis of the system (under the requirements)

Table 2: Grading

Under the above policies, you'll also be evaluated using common sense:

- Your files will be compiled and checked; the system must work.
- Your design and architecture must be intelligent, minimal, effective, and well organized.