DESIGN, TEST AND RELIABILITY OF ELECTRONIC SYSTEMS

1st Project

2019/2020

1. Specifications

The goal of this project is to develop a controller that fulfills the specifications presented below.

The controller must be described in **synthesizable Verilog**. The *Xilinx Vivado HL WebPACK Edition* (available for download at www.xilinx.com) can be used to check whether the Verilog is synthesizable or not.

The controller must synthesize with the *Vivado* design suite **without any errors or warnings**. To check for these errors and warnings, choose the option *SYNTHESIS → Run Synthesis* that is available in the left pane.

To validate the behavior of the controller, the simulator included in the *Vivado* design suite can be used, the easiest to use due the integration in the design flow. To run the simulator use the option *SIMULATION → Run Simulation → Run Behavioral Simulation* that is available in the left pane.

To validate the quality of the testbench, the code coverage must be evaluated using the newest Cadence simulator (*Xcelium*) available in a remote Linux machine. The goal should be a code coverage of 100% for the *Block* and *FSM* categories. The instructions on the setup to obtain the code coverage are provided in the laboratory.

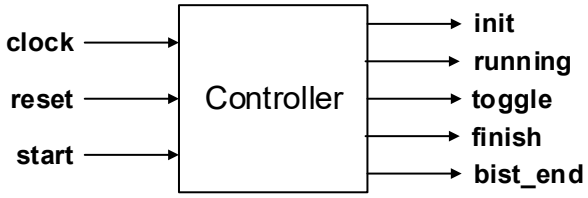

Figure 1: Controller interface

The names of the signals shown in Figure 1, as well as the sensitivity to the edges and logic levels mentioned below, **must be preserved**.

Figure 2 illustrates the behavior of the input and output signals. That information is complemented with a more detailed explanation of the controller inputs and outputs.

Controller inputs

clock: The controller must be sensitive to the rising edges of the clock.

start: After a 0→1 transition in this signal, a new sequence must be initiated (see Figure 2). The **start** value is captured on the rising edge of the clock. While the full sequence is not completed (**bist_end** signal set to HIGH), further transitions in the **start** signal must be ignored.

reset: The **reset** must be synchronous and active at the logic level '1'. This signal is used to restart the state machine and the counters, preparing the controller to start a new sequence. When the **reset** signal goes LOW and the **start** signal is HIGH, a new sequence **must not** be restarted. The controller must wait for a 0→1 transition in the **start** signal to start a new sequence.

Note: When the sequence is complete, a 0→1 transition in the **start** signal is enough to restart a new sequence. The activation of the **reset** signal is not necessary in this case.

Controller outputs

init: This signal is a pulse with a duration of one clock cycle, indicating the start of a new sequence.

running: This signal must be HIGH for **N** clock cycles. If the **reset** signal goes HIGH, then this signal must go LOW.

toggle: This signal must toggle, while the **running** signal is HIGH.

finish: This signal is a pulse with a duration of one clock cycle, indicating the end of a sequence.

bist_end: This signal must go HIGH when the sequence is completed. The **bist_end** signal must go LOW when the **reset** signal is activated or the **start** signal has a 0→1 transition.

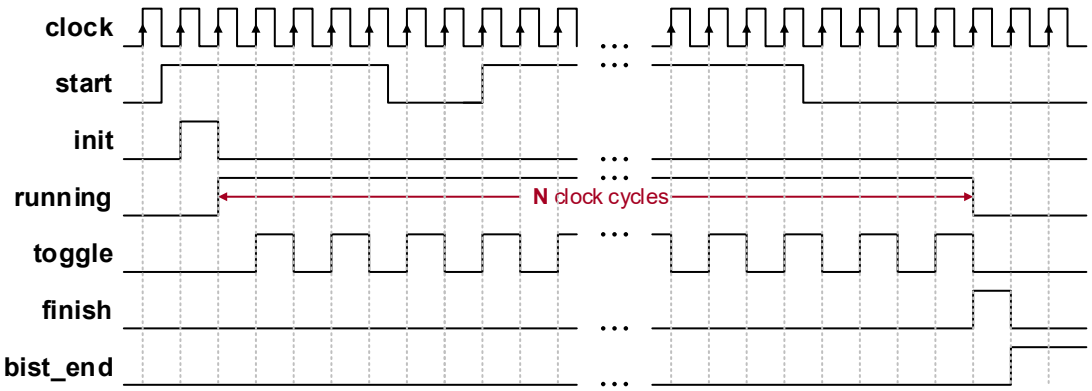

Figure 2: Waveforms

NOTES: The number of clock cycles, **N**, is defined group by group. However, these values **must be defined as parameters** in your Verilog code, to be easily adjustable in the future.

The behavior of the **start** signal is just an example to indicate that all transitions of this signal during the generation of the sequence are discarded.

2. Report

The report must include, at least, the following items:

- State diagram of the state machine
- Textual description of each state of the state machine
- Description of the test scenarios that are included in your *testbench* (activation of the **reset** while the sequence is running, reactivation of the **start** while the sequence is running, etc.).

3. Delivery

The delivery package must be submitted by email and must include, at the least, the following items:

- Report
- Verilog code
- Testbench
- Synthesis report generated by the *Vivado* design suite
- Code coverage report generated by the *Xcelium* simulator