Krestin Ebraheem, U#50717721
Joseph Daniel Moreno, U#60156204

FPGA Design: Final Project

- "FPGA_FinalProject" folder's contents
  - pattern_recognizer.vhd
  - pattern_recognizer_tb.vhd
  - ram.vhd
  - Provided test files: t1.hex, t2.hex, t3.hex, t4.hex
  - Test files created by us (using the test generator script): test0.hex, test1.hex, test2.hex, test3.hex, test4.hex
  - Was able to generate a bitstream.

- Regarding "pattern_recognizer.vhd"…
  - It is the top module of our project.
  - 1 FSM with 9 states (each state described as a comment).
  - pat0 through pat20 refer to the patterns in "patterns.txt" from top to bottom.
  - Design uses counters to keep track of active pattern instances.
  - Most of the logic for the design is in state 1 (s1).
  - RAM entity created and connected (using "ram.vhd").

- Regarding "pattern_recognizer_tb.vhd"…
  - Simulation duration set to 5000 ns.
  - clock period = 10 ns
  - reset = '0' for the whole simulation and go = '1' for the first 20 ns.

- Regarding "ram.vhd"…
  - Uses some of the BRAM initialization from file code from the "vhdl-bram" lecture slides.
  - File currently set to "t1.hex"

- Regarding the test files created by us…
  - test0.hex has 1 completed instance of each pattern.
  - test1.hex has 1 completed instance of pat15, 1 completed instance of pat16, and 2 completed instances of pat17.
  - test2.hex has 1 completed instance of pat2, 1 completed instance of pat5, 1 completed instances of pat8, 1 completed instance of pat11, and 1 completed instance of pat20.
  - test3.hex has 2 completed instances of pat16, 3 completed instances of pat17, 1 completed instance of pat19, and 4 completed instances of pat20.
  - test4.hex has the same input as test3.hex except for line 64; inserted an 18 (0x12) on line 64 to trigger the violation flag.

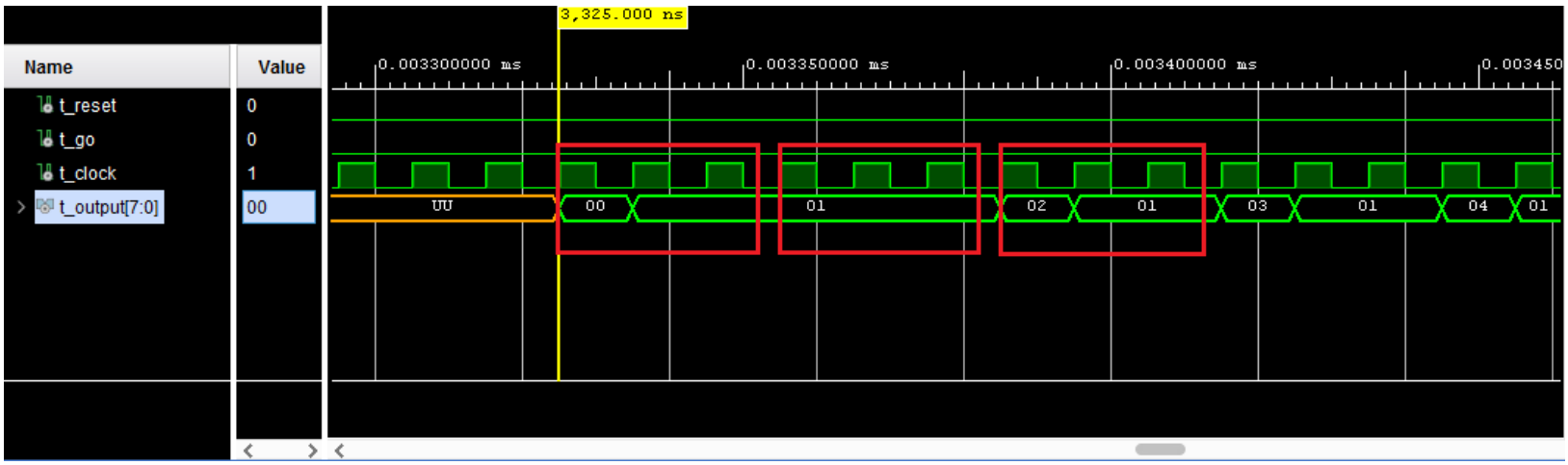
Krestin Ebraheem, U#50717721
Joseph Daniel Moreno, U#60156204

Simulation Explanation

The pattern recognizer starts reading the input once go = ‘1’.

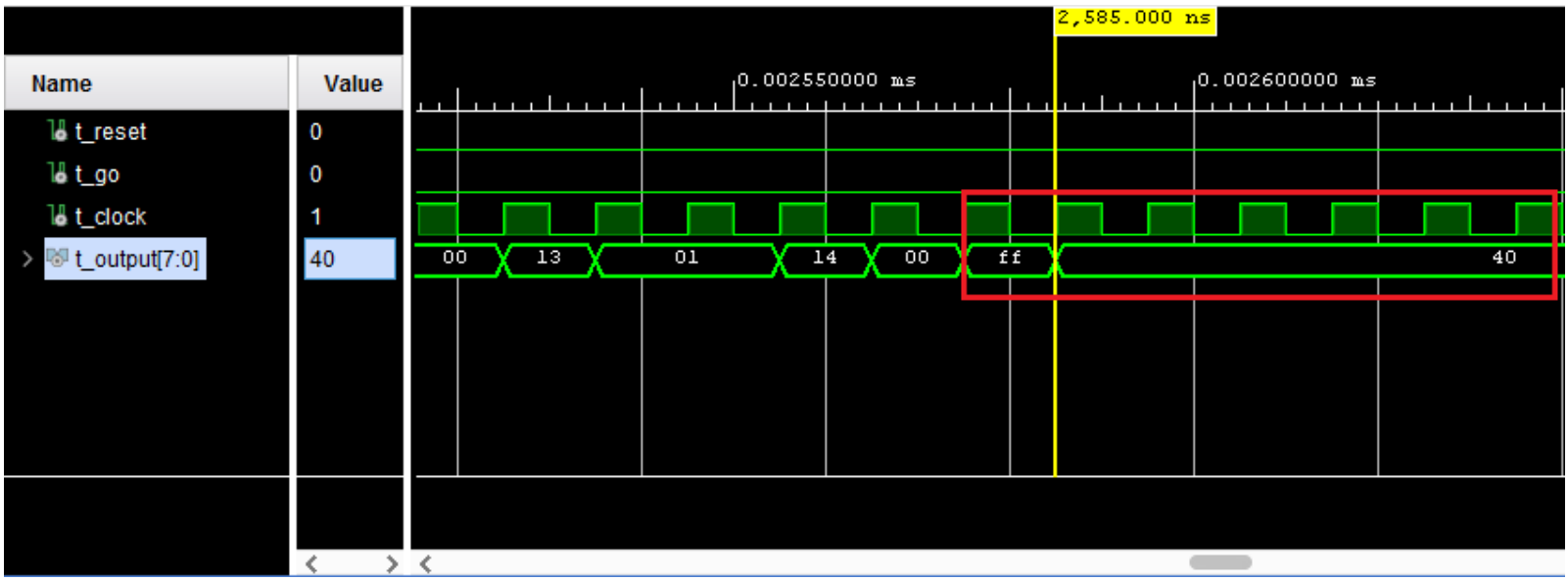
The output starts once known numbers start showing up.
The output format is as follows: pattern # **for 1 cycle** and its corresponding number of completed instances **for 2 cycles**.

Using t1.hex as an example, pattern 0 has 1 completed instance, pattern 1 has 1 completed instance, pattern 2 has 1 completed sequence, and so on.



So the output will show x”00” through x”14” to represent the pattern “ID” and show their corresponding count for completed instances.

Using test4.hex as an example, when a violation occurs, the output will show x”ff” then output the address/index at which the violation occurred from the input.



Krestin Ebraheem, U#50717721
Joseph Daniel Moreno, U#60156204

When a violation occurs, the x”ff” output interrupts pattern 20’s output of the number of completed instances, so it shows for only 1 cycle.

For t1.hex through t4.hex, our design recognized 1 completed instance for each pattern. For test files test0.hex through test3.hex, our design recognized the correct number of completed pattern instances. For test4.hex, our design correctly detected a violation.