Project 2 SEQUENTIAL CIRCUITS

PROBLEM DEFINITION

It is requested to design and implement a synchronous sequential circuit which detects two different 4-bit sequences, A and B. The constraints and requirements are given below.

A) Circuit Properties & Behavior

1. **A** is denoted as the 4-bit binary representation for the **second to last digit of your school number**.
2. **B** is denoted as the 4-bit binary representation for the **last digit in your school number** (If **A=B** choose a different digit).
3. The circuit will have a 1-bit input **X**, a 1-bit output **Y**, a 1-bit **CLK** (clock) input and a 1-bit **RST** (reset) input.
4. Reset behavior will be asynchronous active-1.
5. **A and B may overlap**. Your circuit will detect both, which means your circuit will not ignore detecting one sequence while detecting the other one. Circuit must take all overlaps in consideration, if viable. A overlapping A, A overlapping B, B overlapping B and B overlapping A are all valid situations.
6. **When an A or a B word is detected**, your circuit **sets its** output, Y. Otherwise, the output remains at zero.
7. If either A or B is detected for the second time (not necessarily in a row), **the circuit should go to a lock state, with its output set. The only way to get out of this lock state must be by resetting.** Some example cases are given in Figure-1, at the second page.

B) Design Steps

1. Draw your state diagram and add it to your report. Also, clearly explain your approach on solving the problem at hand.
2. Encode the states of your machine. Discuss how you can encode the states for lesser average power consumption.
3. Make a quick run-through on FSM state reduction techniques. Pick one of those and briefly explain it, then apply it to your state diagram. Show the reduction steps clearly. Redraw your reduced state diagram, if applicable.
4. Gather up timing and resource information from Vivado post-implementation reports. Provide the number of cells used (LUTs and FFs) and the maximum clock frequency for your design.

5. Try out timing constraints and/or Vivado synthesis and implementation strategies (using synthesis and implementation settings menus in Vivado) to improve your design's speed performance. Show all related work in your report, along with final maximum clock frequency and cell usage results.
6. Write a testbench for your design. **Your testing scenario should make the circuit demonstrate all possible use cases (getting locked due to the detection of each pattern, all possible overlap cases, showing that you can only get out of the lock state by resetting, etc.).** Show and explain how each individual case is verified on the waveform **(Do not just add a screenshot of your waveform, show and explain the results clearly).** Your circuit is expected to work correctly in **post-implementation timing simulation**.
7. State the type of your machine (Mealy or Moore, and why so?). Analyze the occurrence of **hazardous outputs** (zararlı hatalı çıkışlar).

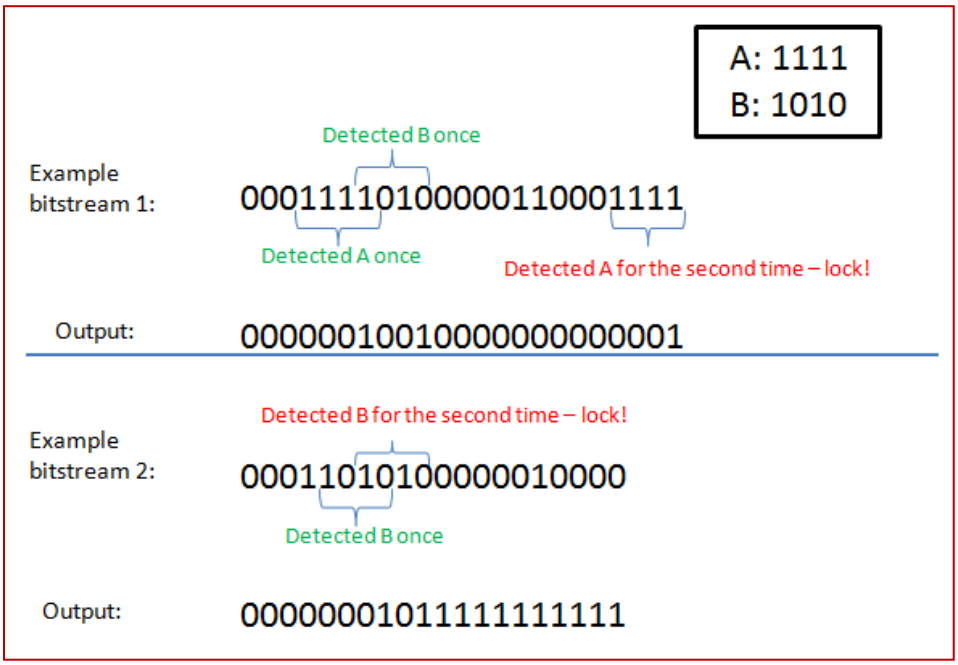

Figure-1: Some examples on the desired circuit behavior. Note that the output may be delayed one cycle depending on the machine type.

NOTE: This WILL NOT be a group project. Prepare your reports individually.

GOOD LUCK!