[CS M51A F14] Homework 5

Due: 12/05/14

TA: Yang Lu (yangluphil@gmail.com)

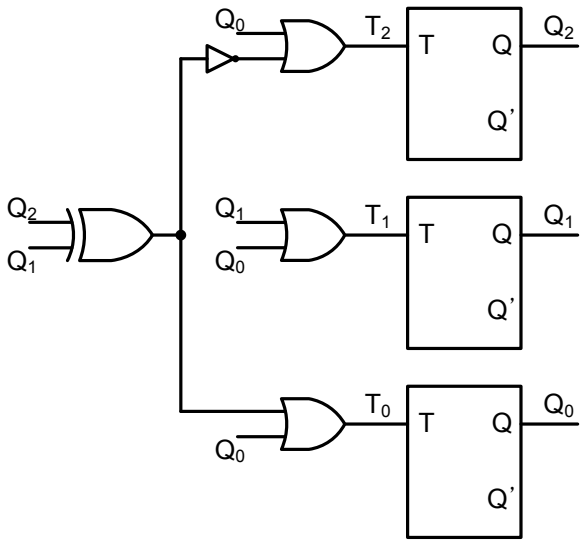
Problems (40 points total)

Problem 1 (6 points)

1. Create a D flip-flop using a JK flip-flop. Use the excitation tables on page 221 of the textbook.
2. Create a T flip-flop using an SR flip-flop.
3. Create a JK flip-flop using a T flip-flop.

Problem 2 (6 points)

We would like to analyze the sequential system shown below. It is an autonomous counter which outputs a fixed string of numbers. The output changes at every clock cycle.



1. Write the expressions for T_2, T_1, and T_0.
2. Write the table of T_2, T_1 and T_0. Using this table, show the state transition table.
3. Draw the state transition diagram of the system.

Problem 3 (8 points)

We would like to design a pattern recognizer with a binary stream as input. The system has a binary output bit, which is 1 whenever $x(t-3,t) = 0010$ or 0001 and 0 otherwise.

1. Draw the state transition diagram. Try to minimize the number of states. (*Hint: The system can be designed using only 5 states.*)

2. Encode the states into binary bits. From the encoding, write the state transition table and table of JK flip-flop inputs.

3. Using K-maps, derive the switching expressions for each flip-flop input J_i, K_i, and the output bit z.

4. Draw the sequential flip-flop network.

Problem 4 (6 points)

We would like to put together a 12-input decoder using 4-input decoders. Let us consider the following two methods.

1. Using a tree structure, how many levels would we need? How many 4-input decoders are used in total?

2. The textbook has an example of using a coincident structure which divides the n-inputs into two groups using $\frac{n}{2}$-input decoders. This can be expanded to the case where we divide the n inputs into k groups, using r-input decoders. In this case, what would be the value of k in terms of n and r? How many AND gates would we need?

3. Now, for our coincident 12-input decoder, how many 4-input decoders do we need? How many AND gates? How many inputs do we need for the AND gates?

Problem 5 (4 points)

We want to use multiplexers to implement the function $f(a, b, c, d) = \sum m(1, 2, 4, 7, 10, 11, 14)$ using the following methods.

For both cases, show your work, either the minterm expressions or K-map implementations. Assume that complements to inputs are available.

1. Implement the function using 8-to-1 multiplexers. Use a, b, c as the selection bits.

2. Repeat the same process using 4-to-1 multiplexers. Use a, b as the selection bits, and for the additional logic needed, only use XOR gates and NOR gates.

LogiSim Design Problem (10 points)

1 Introduction

For this assignment, you will be designing the combinational circuits for a pattern detector using two different flip-flop types. The detector will output a 1 whenever "1001", "110", or "011" pattern is detected from a given input stream.

Input: $x(t) \in \{0, 1\}$
$\text{reset}(t) \in \{0, 1\}$

Output: $z(t) \in \{0, 1\}$

Function: $z(t) = \begin{cases} 1 & \text{if reset}(t) = 0 \text{ and if } x(t-3, t) = 1001 \text{ or } x(t-2, t) = 110 \text{ or } x(t-2, t) = 011 \\ 0 & \text{otherwise} \end{cases}$

1.1 Design Guidelines

You are given two skeleton files, one using D flip-flops and one using JK. You must adhere to pin (input/output) names and implement the missing logic for given modules.

The top-level modules "TOP_patternD" and "TOP_patternJK" are completed and you do not have to add anything here. They have two one-bit inputs x and reset, along with a clock input. When reset is set to 1, the state machine will immediately revert to the initial state. The main output is a single bit value z, which is 1 when one of the three strings that we are looking for is found, and 0 otherwise. The output signal curState is mainly for reference and debugging purposes.

The sub-module "state_comb" will include all combinational logic for determining the next state and output. This will be the parts you need to complete. In relation to the flip-flops used in the TOP module, the logic for deriving the next state will be significantly different from each other. For the process of deriving each logic, you can follow the process that we looked at in chapter 8.

1.2 Allowed Components

In completing the pattern detector, you are only allowed to use the gates NOT, AND, OR, NAND, NOR, XOR, XNOR, and the **Tunnel** and **Constant** components under the Wiring category. Refrain from using long wires to connect everything, instead utilize the tunnel component to improve readability of the circuit.

1.3 Bit vectors

All bit vectors used in the design are indexed in little-endian form; the most significant bit has the largest index, all the way down to the least significant bit, which is always index 0.

1.4 Skeleton module

DO NOT change the location of the input/output pins in each module or add any new pins to the skeleton module. Moving the positions of the pins can cause the signals to connect to different pins from the original setup. Points will be deducted for any errors caused by this.

1.5 Submission

Please print out your design and attach it to your homework submission.