A_3

Assignment 3

CS 3482; Professor Tang

Connor Taffe. T no. 3742

November 3, 2015

1. To test the circuits, we will assure that the Q only changes when G is 1 and D is changed. Q should reflect the value of D while G is on (1), and then lock when G is off (0). G should serve as the *latch* in the D latch circuit.

   The D flip flop is edge triggered rather than level triggered as the D latch is. This means that the D flip flop should only change on rising edges of G. Q only reflects D if it is at the rising edge of G, otherwise it is locked.

2.

3. Each row of output D bits is the input Q bits of the last state plus 1. The sum of minterms obtained from the Karnaugh maps of each of the four Ds are as follows:

$$D_0 = Q_0'$$

$$D_1 = Q_1'Q_0 + Q_1Q_0'$$

$$\begin{aligned} D_2 &= Q_3'Q_2'Q_1Q_0 + Q_3'Q_2Q_1' + Q_3'Q_2Q_1Q_0' + Q_3Q_2Q_1Q_0 + Q_3Q_2'Q_1' + Q_3Q_2'Q_1Q_0' \\ &= Q_3'(Q_2'Q_1Q_0 + Q_2Q_1' + Q_2Q_1Q_0') + Q_3(Q_2'Q_1Q_0 + Q_2Q_1' + Q_2Q_1Q_0') \\ &= (Q_3' + Q_3)(Q_2'Q_1Q_0 + Q_2Q_1' + Q_2Q_1Q_0') \\ &= \top(Q_2'Q_1Q_0 + Q_2Q_1' + Q_2Q_1Q_0') \\ &= Q_2'Q_1Q_0 + Q_2Q_1' + Q_2Q_1Q_0' \end{aligned}$$

$$D_3 = Q_3Q_2' + Q_3Q_2Q_1' + Q_3Q_2Q_1Q_0' + Q_3'Q_2Q_1Q_0$$

   Note I did some quick extra algebra to save myself some duplicate gates. As the Q_3s factor out and become a tautology because a thing or it's inverse must be true.

   For the resulting circuit, I've built one with G and CLR inputs, and D and Q outputs, as per your instructions that they must be testable. The D output in this circuit is unrelated to the D inputs in the next question.

   For each clock cycle you can see that on the rising edge of the input clock G, the counter increments it's four bit internal state. The finite state machine chugs along preserving its state in the D flip flop memory units. When the previous state is hexadecimal F or 16, all the bits are used up and the state returns to 0 as per the truth table I originally came up with the sum of minterms specified with.

4. This version of the circuit adds a quadruple two to one muliplexer which exposes a *LD* switch to the circuit. When on, this switches the current state from the calculated next state from the state machine logic, to one provided by a user, allowing the counter to be "set."

   Testing reaveals a fully functioning implementation which can take a user provided number and count from it, properly wrapping as it would with its own counter.