Homework 2

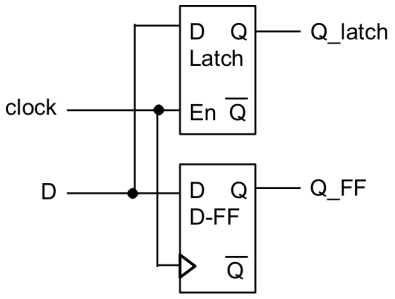
Out: 9.29.21
Due: 10.6.21

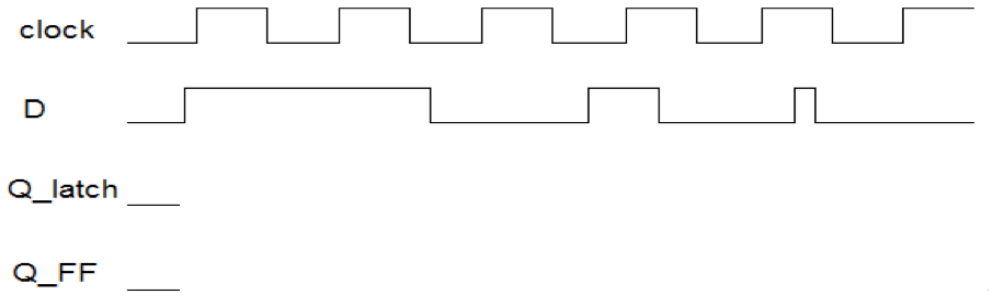
1. [Combinational Logic]
   Design a combinational circuit with three inputs, x, y, and z, and three outputs, A, B, and C. When the decimal value of the input is 0, 1, 2, or 3, the output = input + 3, and when the decimal value of the input is 4, 5, 6, or 7, the output = input – 1. Show your truth table, K-maps, and minimized logic expressions.

2. [Adders]
   A half adder has two inputs, A and B, and two outputs, AB and A ⊕ B. Given 3 half adders in total (for all expressions), and no additional gates, implement the following four Boolean expressions:
   D = A ⊕ B ⊕ C
   E = A'BC + AB'C
   F = ABC' + (A'+B')C
   G = ABC

3. [Comparators]
   Assume that X consists of 3 bits, x2, x1, x0, and Y consist of 3 bits y2, y1, y0. Write logic functions that are true if and only if:
   a) X < Y, where X and Y are unsigned binary numbers
   b) X < Y, where X and Y are signed (two's complement) binary numbers
   c) X = Y
   In your solution, use a hierarchical approach that can be extended to larger numbers of bits.

4. [Latches and FFs]
   Fill in the output waveforms for the outputs of a D latch and positive edge triggered D Flip-Flop below.





5. [Division]
Perform the binary division of 110111 divided by 101 (both are unsigned numbers). For each step (states are numbered by which bit of the divisor is examined), show the contents of the remainder register and the answer register (both of which are 6 bit unsigned numbers):

State	Remainder	Answer
Start	000000	000000
5		
4		
3		
2		
1		
0		

6. [FSM Design]
A vending machine accepts nickels (5 cents) and dimes (10 cents) only. Some of the snacks, which this vending machine dispenses, cost 25 cents, while others cost 40 cents. Your task is to draw the finite state machine diagram for the control logic of this vending machine. This should be a Moore-type FSM (the output is only a function of the state). The control logic accepts the following inputs:

inNickel – the user inserted a nickel
inDime – the user inserted a dime
vend25 – the user requested a 25 cent snack
vend40 – the user requested a 40 cent snack

The control logic generates the following outputs:

outVend – dispense the requested snack
outNickel – Give back a nickel to the user
outDime – Give back a dime to the user

Your vending machine controller must meet the following requirements:

- It only asserts *outVend* in response to *vend25* or *vend40* if sufficient money has already been paid to satisfy the request (and ignores vending requests otherwise).
- After asserting *outVend*, it must return any money entered beyond the price to the user via *outDime* and/or *outNickel* (for example, if the user puts in 40 cents, and requests a 25 cent snack, 15 cents must be returned in change).
- Only one of *outVend*, *outDime* or *outNickel* may be asserted at a time.

You may assume the following (to simplify your design):

- At most one input will be asserted at a time.
- If the user enters more money than the maximum snack price, you should return some or all of the money over that price immediately.
- To keep your diagram simple, you should omit any self-loops on no input (that is, if you do not draw a self loop on a state, we will assume that on no inputs, you stay in that state).

For this question, you should submit a drawing of the state diagram, and a clear description of your output function.