[CS M51A F14] Assignment 4

Due: 12/05/14

TA: Hyunjin Kim (roykim78@cs.ucla.edu)

Homework Problems (100 points total)

Problem 1 (5 points)

A system is described by the following table. Minimize the number of states. Show the corresponding minimal table.

	Input	
PS	$x = 0$	$x = 1$
A	F,0	D,1
B	F,0	C,1
C	B,0	F,0
D	G,0	A,0
E	I,0	H,1
F	A,0	C,1
G	A,0	D,1
H	A,1	C,2
I	E,0	H,1

Problem 2 (20 points)

Design a binary string detector which takes as input a string of binary values, and outputs a 1 when it detects a string of 1011 or 1100.

1. The input is $x(t)$, and the output is $z(t)$. Write a state description of the string detector by specifying the input and output sets, and write the output function.
2. Fill in the state transition table for a Mealy machine of the string detector. Do not optimize the number of states yet.

PS	$x=0$	$x=1$
S_{init}	$S_0, 0$	$S_1, 0$
S_0		
S_1		
S_{00}		
S_{01}		
S_{10}		
S_{11}		
S_{000}		
S_{001}		
S_{010}		
S_{011}		
S_{100}		
S_{101}		
S_{110}		
S_{111}		
	NS, z	

3. Minimize the number of states of the transition table, and show the final minimized table.

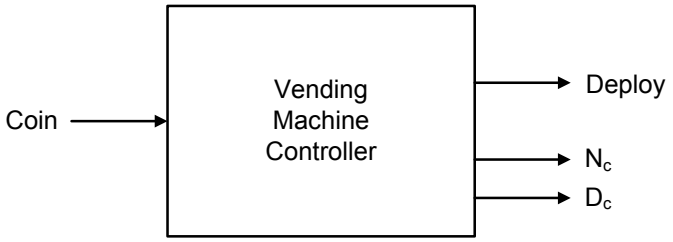
4. Draw the minimized table as a state diagram.

Problem 3 (20 points)

Our goal is to design a vending machine which sells stamps. The price of a stamp is 35 cents (for the sake of the problem). The machine accepts only nickels (5 cents), dimes (10 cents), and quarters (25 cents).

When the total value of coins is equal to or larger than the price of the stamp, the machine deploys the stamp, and returns any change as necessary. The change is given in nickels or dimes, and is returned in a way such that the total number of coins returned is the smallest possible. For example, if the amount of change is 30 cents, the machine returns 3 dimes, and not a mixture of dimes and nickels which will result in a higher coin count.

The machine's control module looks like the diagram below. It has an input *Coin* which denotes the type of coin deposited, and has three outputs: *Deploy*, which is 1 when the machine needs to deploy a stamp and 0 otherwise, N_c, number of nickels to return as change, and D_c, number of dimes to return as change.



1. What is the minimum number of states necessary for the control module? What would each state represent? (*Hint: To find the minimum number of states, first write any state machine which has the functionality that you want, and try to reduce the number of states afterwards.*)

2. Show the state transition table. The output should be written as a three-digit number, where each digit corresponds to the value of *Deploy*, N_c and D_c, in that order.

Problem 4 (15 points)

We are given a sequential system as shown below.

Inputs:	$x \in \{a, b, c\}$
Outputs:	$z \in \{0, 1\}$
Function:	$z = 1$ if $x(t-3, t) = abca$ and the number of a's in $x(0, t)$ is even

Obtain a "loose" state description of the system by creating a functionally valid state machine without worrying about it having the minimum number of states, and afterwards minimize the number of states.

LogiSim Design Problem (40 points)

1 A Pattern Detector

For this assignment, you will be design a pattern detector circuit. Your circuit will output a '1' whenever a "1101" pattern is detected from a given input stream and '0' otherwise.

$$\begin{aligned} \text{Input:} \quad & x(t) \in \{0,1\} \\ & reset(t) \in \{0,1\} \\ \text{Output:} \quad & z(t) \in \{0,1\} \\ \text{Function:} \quad & z(t) = \begin{cases} 1 & \text{if } x(t-3,t) = 1101 \\ 0 & \text{otherwise, or if reset (t)} = 1 \end{cases} \end{aligned}$$

1.1 Provide implementation details of your circuit by filling in the following table with the minimum number (4) of states.

	Input	
PS	$x=0$	$x=1$
$S0$		
$S1$		
$S2$		
$S3$		
	NS,Output	

1.2 Write the expressions of the next state and the output from the transition table

1.3 Logisim Design

You are given a skeleton file. You must adhere to pin (input/output) names and implement the missing logic for given modules. You are allowed to use basic logic gates (e.g., AND, OR, NOT, ..., etc.), wire splitters, as well as **registers** and **multiplexers** required for implementing your sequential circuit. A **clock** module is also provided.

Your top-level module "pattern" will have two primary inputs (x, reset) of size 1-bit. When the reset switch is set to '1', your state machine should revert back to an initial state (e.g., S0). **Print out your logisim design and attach it to your homework submission.**