Homework 4: State Machines

Due 12/7/2016 at 9:59 AM – No Late Submissions

Problem 1:

You are to design a state machine to control an old fashioned vending machine that would drop a token for a price of 20 cents. There are two possible inputs at any state: N (for nickel) and D (for dime). Based on the input, the machine is to transition to a new state. When any sequence of coins worth 20 cents or more is input, the machine transitions to the start state and outputs a 1 to indicate that the vending machine is to drop the token. All other transitions are accompanied with an output of 0. If the value of the coins is higher than 20 cents, the extra money is lost.

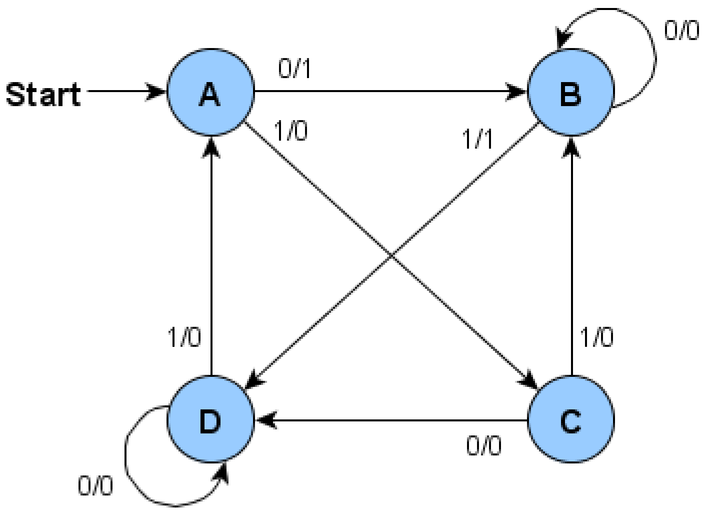
Draw the state diagram for this machine. If you're looking for free software to draw state diagrams, try *yEd* (https://www.yworks.com/products/yed).

Submit your answer as a PDF document to the **H4P1 Diagram** drop box in Canvas.

Problem 2:

Read all the instructions before starting!

In this problem, you will design a circuit that implements the combinational part of the following state machine:



		S1	S0
A	=	0	0
B	=	0	1
C	=	1	0
D	=	1	1

You must use the state encoding shown on the right. Note that *S1* is the most significant bit of the state and *S0* is the least significant bit. The input to the state machine will be labeled *In*. The output of the state machine will be labeled *Out*.

The first step is to complete the state table. Go to the Canvas quiz titled **H4P2 Table**. Complete the state table and submit it. You only have one attempt to get credit. If you didn't get it right, you must correct it before moving on.

Next, implement the state table as a combinational circuit in Logisim starting from the following skeleton file: https://www.cs.colostate.edu/~cs270/.Fall16/assignments/H4P2/src/h4p2.circ

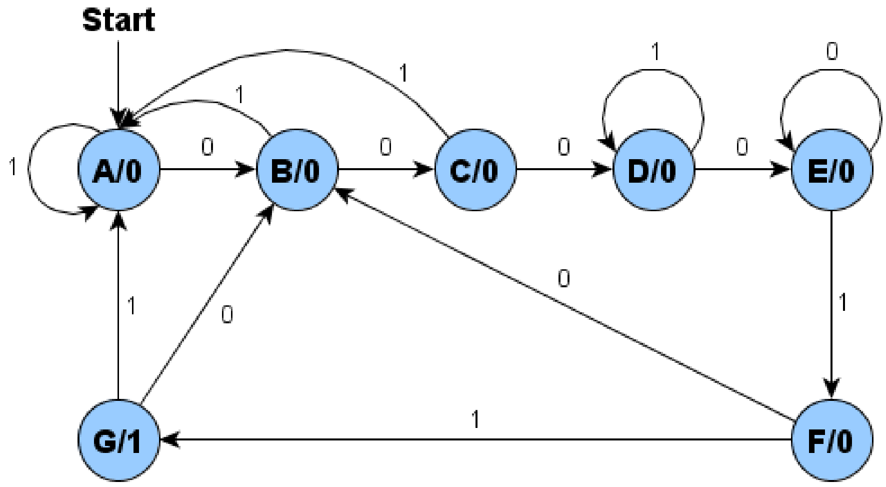
Follow these guidelines:

1. Do not introduce additional input or output pins and do not modify the input and output sections. Work on the main circuit only. Do not create sub-circuits.
2. You should implement the combinational logic part only. Therefore, your circuit should not contain flip flops or clocks. You may use elements from the *Wiring* and *Gates* libraries only.
3. Your circuit will be auto-graded. Submit it to the **H4P2** drop box in the Checkin tab. Make sure you pass preliminary testing which will check that your circuit is suitable for auto-grading. If you don't pass preliminary testing, you will get a 0 for the circuit.

Problem 3:

Read all the instructions before starting!

In this problem, you will design a Logisim circuit that implements the following state machine:



		S2	S1	S0
A	=	0	0	0
B	=	0	0	1
C	=	1	1	0
D	=	1	0	0
E	=	0	1	1
F	=	1	0	1
G	=	0	1	0

You must use the state encoding shown on the right. Note that *S2* is the most significant bit of the state and *S0* is the least significant bit. Also, notice that the state encoding is unusual: it's not in increasing binary order. Be careful about this when building your state table. The input to the state machine will be labeled *In*. The output of the state machine will be labeled *Out*. This is a *Moore state machine* since the output only depends on the current state.

The first step is to build the state table. How many rows should it have? Make sure you check your state table thoroughly to make sure it matches the diagram before moving on to the circuit.

Once you're comfortable with your state table, implement the state machine in Logisim. Start from the following skeleton file: https://www.cs.colostate.edu/~cs270/.Fall16/assignments/H4P3/src/h4p3.circ

Follow these guidelines:

1. Do not introduce additional input or output pins and do not modify the input and output sections. Work on the main circuit only. Do not create sub-circuits. You may use elements from the *Wiring*, *Gates*, and *Memory (only D flip flops)* libraries only.
2. Since the state encoding requires 3 bits to represent a state, you will need 3 D flip flops. Connect the Q output of each flip flop to the corresponding output pin. For example, the Q output of the flip flop that stores S2 should be connected to the S2 output pin. Don't forget the clock in your circuit. You don't need a flip flop for *Out* (*what do you think is the reason?*).
3. Your circuit will be auto-graded. Submit it to the **H4P3** drop box in the Checkin tab. Make sure you pass preliminary testing which will check that your circuit is suitable for auto-grading and will test a state transition. If you don't pass preliminary testing, you will get a 0 for the circuit.

Advanced question (for fun): can you write the regular expression that this state machine recognizes?