Name: ______________________________ ID# ____________________

Date Submitted: __________________ Lab Section # ____________________

CSE 2441 – Introduction to Digital Logic Fall Semester 2020

Lab Number 9 – Digital Lock Controller

(Due – November 1, 2020, 11:59 PM)

This is a BitBoard Lab.

CSE 2441 LABORATORY ASSIGNMENT 9 FALL 2020

DIGITAL LOCK CONTROLLER

(100 POINTS)

PURPOSE/OUTCOMES

To design, implement on The BitBoard, and test a Moore-type sequential circuit that functions as the controller for a digital lock. Once you complete this assignment, you will have demonstrated an ability to design a sequential circuit that meets specified requirements and to implement the design using standard logic gates and flip-flops.

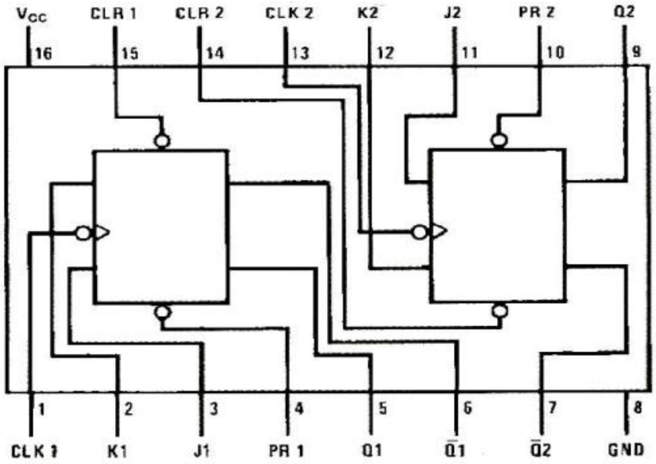
BACKGROUND

Sequential circuits can be used as sequence detectors and, hence, can be designed as the controller for a digital lock. Sequences can be considered as block or non-block codes. Block codes have a fixed length and are not overlapping for detection purposes. A non-block code has no fixed length and may contain overlapping segments. For example consider the one-bit code 1-0-1-1. The sequence 1-0-1-1-0-1-1-0 would produce output sequences 0-0-0-1-0-0-0-0 and 0-0-0-1-0-0-1-0 for block and non-block detectors, respectively.

For block codes, the number of possible code sequences of length n is m^n where m is the number of symbols in the code. For the above example, *m=2* since 0 and 1 are the code characters and n=4. So the number of code sequences is $2^4 = 16$.

Sequential circuits can be designed as Mealy machines or Moore machines. The current output of a Mealy machine is a function of the current input and current state of the machine. Consequently, unwanted spurious outputs, or glitches, may occur if inputs change between state changes. Spurious outputs cannot occur in Moore machines since the current output is a function of only the current state. Mealy machines usually require fewer states than an equivalent Moore machine but that is preferred when spurious outputs must be avoided.

An SN74LS112, dual JK flip-flop, integrated circuit will be used in this laboratory. Its pin-out diagram is shown below. Note that the 74112 is a negative-edge triggered device and that the preset (PR) and clear (CLR) inputs are active low. It's advisable to connect preset and/or clear signals to Vcc if they are not being used. Otherwise, the flip-flop will not operate as expected.



DESIGN REQUIREMENTS

Your assignment is to design, implement, and test a controller for a digital lock as shown in figure 1 that meets the requirements specified below.

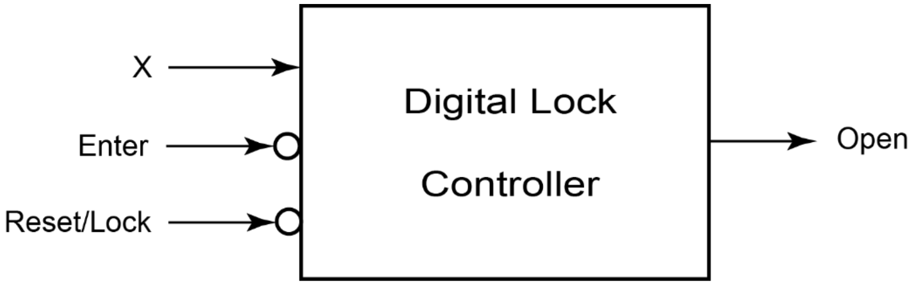


Figure 1. Digital Lock Controller Block Diagram

1. *X is* used to input the binary code sequences for unlocking the lock.
2. *Enter* is used to tell the controller to read the current input value.
3. *Reset/Lock* is used to restart the code entry process or to lock the lock after it's opened.
4. The lock will be opened (*Open = 1*) **if and only if** code sequence *X:1-0-0-1* is entered.
5. The controller must then stay in the Open = 1 state until locked (Open = 0).

DESIGN ASSIGNMENT (30 points)

1. Develop the state diagram and state table for a Moore machine that meets the design requirements.
2. Make a state assignment and generate state and output transition tables.
3. Derive flip-flop excitation maps. Use JK flip-flops (SN74112) for memory elements.
4. Derive circuit output maps.
5. Produce logic equations for flip-flop inputs and circuit outputs.
6. Use gates provided in The BitBox chip set to realize the combinational logic circuits. Try to minimize the number of chips but keep wiring complexity in mind.
7. Capture and verify your design using Quartus II.

LAB ASSIGNMENT (70 points)

1. Implement your design on The BitBoard using the following pin assignments.

   X = SW0
   Enter = KEY1
   Reset/Lock = KEY0
   Open = LEDR0

2. Test your implementation and record your results.

REPORT REQUIREMENT

1. Cover sheet (as shown on this assignment)
2. Lab purpose
3. State diagram of your controller
4. Circuit diagram of your realization (screen shot)
5. Simulation results (waveforms) showing the verification of your design
6. Picture of your wired circuit on The BitBoard wiring
7. Test results table for part 2
8. Pictures of your circuits under test