1. We wish to recognize the sequence "110" using a Moore machine implementation.
   (a) Determine the required number of states and their encoded values

   (b) Draw the state transition diagram making sure you consider both input possibilities for each state

   (c) Complete the state transition table

Present State		Input	Next State		Output
q_1	q_0	**x**	q_1^+	q_0^+	**z**
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

   (d) Derive the Boolean expressions for the next state variables

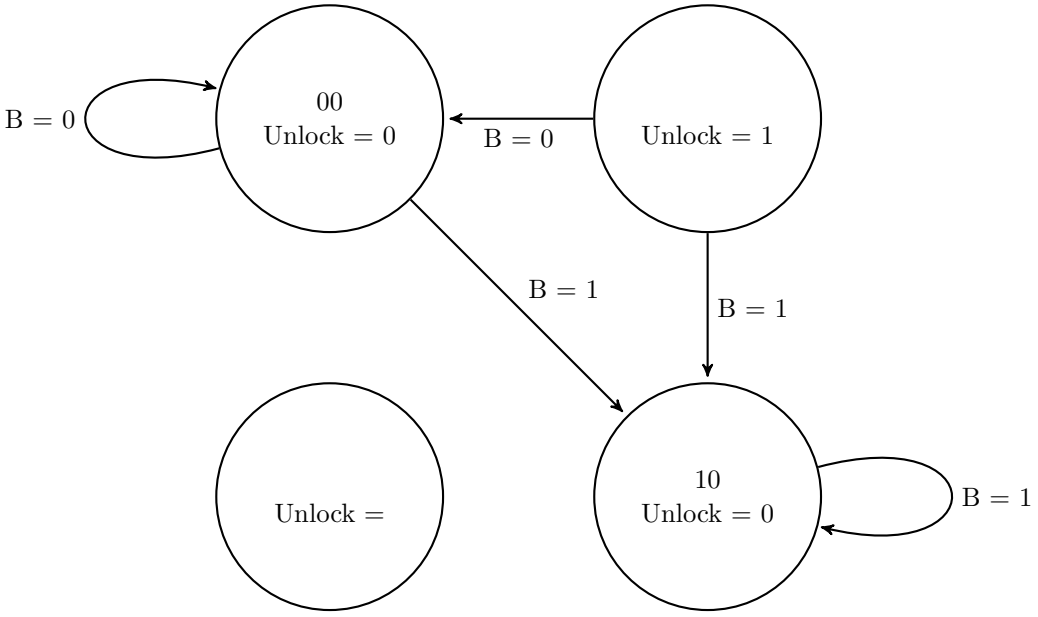
(e) Sketch the hardware implementation

2. We are confronted with a bit of a puzzle! The following digital padlock design has only been partially documented in terms of the state transition diagram and table.

- There are 4 states which must all be encoded with a 2-bit name
- Note, the state names are not at all derived from the lock combination!
- The input button push "B" can be 0 or 1.
- There are three numbers in the unlock code.

(a) Complete the drawing of the state transition diagram (*Note: there are 3 transitions missing, 2 states that need to be assigned, and 1 output value missing*). Then fill in the remaining entries of the state transition table below.

q_1	q_0	**B**	q_1^+	q_0^+	**Unlock**
0	0	0	0	0	
0	0	1	1	0	
0	1	0	1	1	0
0	1	1			0
1	0	0	0	1	
1	0	1			
1	1	0			1
1	1	1			1



(b) Specify the UNLOCK output from the table using a SOP Boolean expression

UNLOCK =

(c) Specify the q_1^+ next state using a POS Boolean expression

q_1^+ POS =

(d) What is the combination for the lock?

Combination =

1. A vending machine requires 15 cents for a tasty cup of coffee. It does not take pennies or quarters. It can provide a nickel in change.

   (a) Draw the state transition diagram.
   Note: It should be a Mealy machine with 3 states, you should return to zero money when you vend

   (b) Encode the states and the inputs. The inputs should be N: nickel or D: dime. You can only input one coin at a time.

   (c) Create the state transition table.

q_1	q_0	**x**	q_1^+	q_0^+	z_1	z_2
0	0	0				
0	0	1				
0	1	0				
0	1	1				
1	0	0				
1	0	1				
1	1	0				
1	1	1				

(d) Write the next state and output logic expressions.

(e) Implement the design in hardware.