Name: Solution **Documentation**:

1. We wish to recognize the sequence "110" using a Moore machine implementation.

(a) Determine the required number of states and their encoded values

4 states

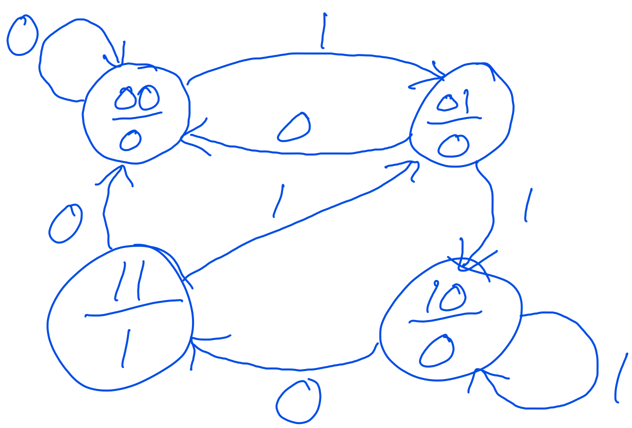
S∅: ∅ bits of sequence detected — S∅: 00

S1: 1 bit of sequence detected — S1: 01

S2: 2 bits " " " — S2: 10

S3: 3 " " " " — S3: 11

(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	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	1	1	0
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	0	1	1

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

*Fewer 1's than 0's ⇒ SOP

$$q_1^+ = \bar{q}_1 q_0 x + \underbrace{q_1 \bar{q}_0 \bar{x} + q_1 \bar{q}_0 x}_{\text{simplifies to } q_1 \bar{q}_0}$$

$$\boxed{q_1^+ = \bar{q}_1 q_0 x + q_1 \bar{q}_0}$$

$$\boxed{q_0^+ = \bar{q}_1 \bar{q}_0 x + q_1 \bar{q}_0 \bar{x} + q_1 q_0 x}$$

Not required, but...

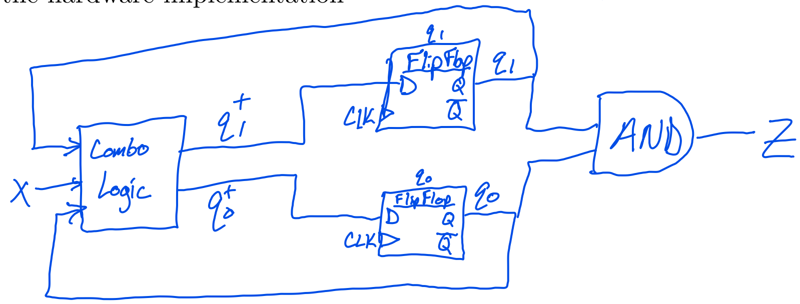
$$Z = q_1 q_0 \bar{x} + q_1 q_0 x$$

$$= q_1 q_0 (\cancelto{1}{\bar{x} + x})$$

$$\boxed{Z = q_1 q_0}$$

(e) Sketch the hardware implementation

\# of flip flops = # of state bits = 2



*Since $Z = q_1 q_0$, I just used the AND gate instead of a generic "Combo Logic" box. Either way is correct.

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.

State	Encoded	#bits in Unlock sequence
S0	00	0
S1	10	1
S2	01	2
S3	11	3

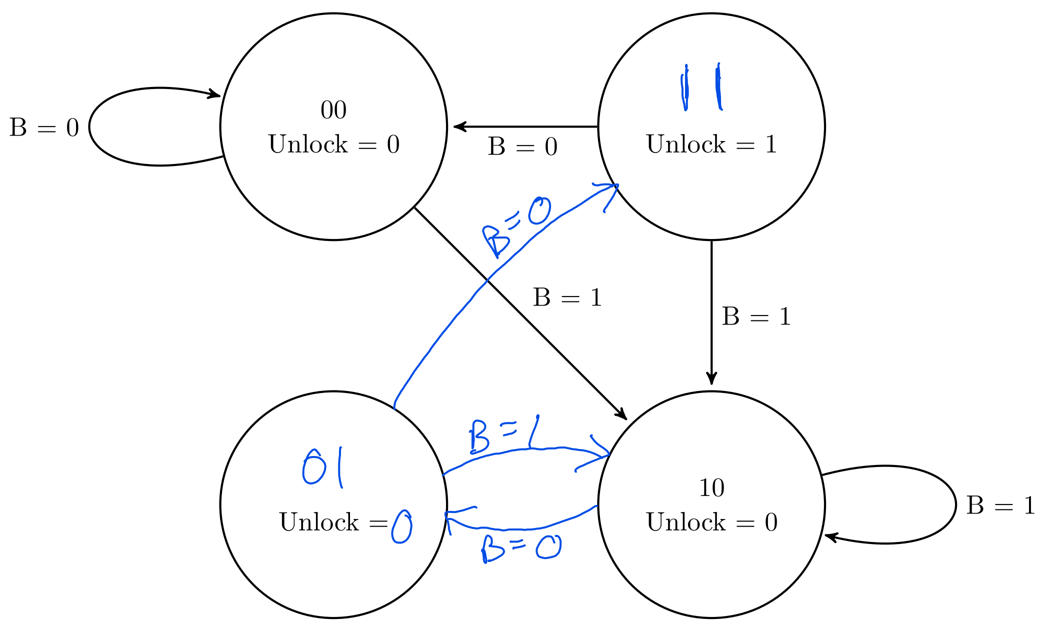
- There are 4 states which must all be encoded with a 2-bit name
- Note, the state names are not at all derived from the lock combination!
- The input button push "B" can be 0 or 1.
- There are three numbers in the unlock code.

(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.

- Use the info in the transition diagram to complete the table and the info in the table to finish the diagram.

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

- Moore machine since output Unlock inside state in diagram; easy to complete Unlock column in table!



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

$$Unlock = q_1 q_0 \overline{B} + q_1 q_0 B = q_1 q_0 (\overline{B} + B) = q_1 q_0$$

(makes sense, since this a Moore machine and Unlock only equals 1 in state S3 (11).)

UNLOCK = $q_1 q_0$

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

3 0's in q_1^+ column

$$q_1^+ = (q_1 + q_0 + B)(\overline{q_1} + q_0 + B)(\overline{q_1} + \overline{q_0} + B)$$

q_1^+ POS = $(q_1 + q_0 + B)(\overline{q_1} + q_0 + B)(\overline{q_1} + \overline{q_0} + B)$

(d) What is the combination for the lock?

- Follow the input sequence needed to get to state S3 (11).

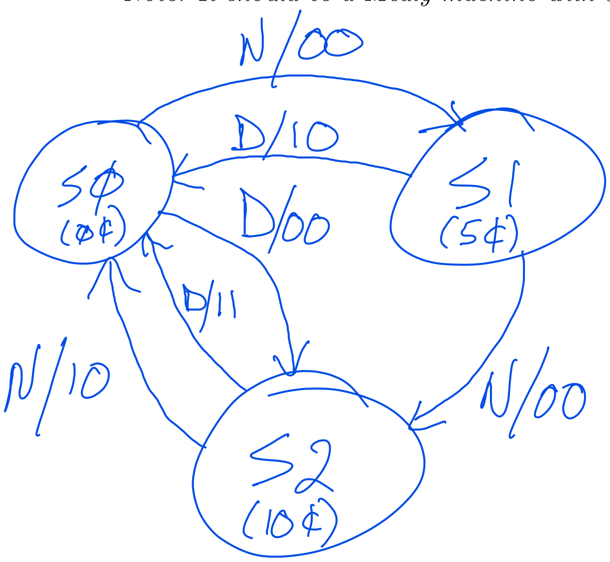
Combination = 100

Name: **Documentation**:

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. ← Derived requirement: can only give change if also vending.

   (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



Outputs	Meaning
00	No vend, no change
01	(unused)
10	vend, no change
11	vend, change

* Mealy machine use INPUT/OUTPUT notation with each transition arrow

   (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.

Inputs	Encoded
N	0
D	1

State	Encoded	Meaning
S0	00	no money
S1	01	5¢
S2	10	10¢

   (c) Create the state transition table.

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

Use DON'T CARES for 4th unused state

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

* Fewer 1's than Ø's ⇒ SOP

$$\boxed{\begin{aligned} q_1^+ &= \bar{q}_1 \bar{q}_0 X + \bar{q}_1 q_0 \bar{X} \\ q_0^+ &= \bar{q}_1 \bar{q}_0 \bar{X} \end{aligned}}$$

$$Z_1 = \bar{q}_1 q_0 X + \underbrace{q_1 \bar{q}_0 \bar{X} + q_1 \bar{q}_0 X}_{\text{simplifies}}$$

$$\boxed{Z_1 = \bar{q}_1 q_0 X + q_1 \bar{q}_0}$$

$$\boxed{Z_2 = q_1 \bar{q}_0 X}$$

(e) Implement the design in hardware.

\# Flip Flops = # of bits for states = 2