

EXAMPLES ON DESIGN OF FSMs:

Ⓐ Multiples:

A.I.: Design a FSM that reads in 1 bit at a time and outputs a 1 if the ~~sum~~ number of ones it has seen so far is a multiple of 3, and a 0 otherwise.

Implement a Moore machine.

Solution

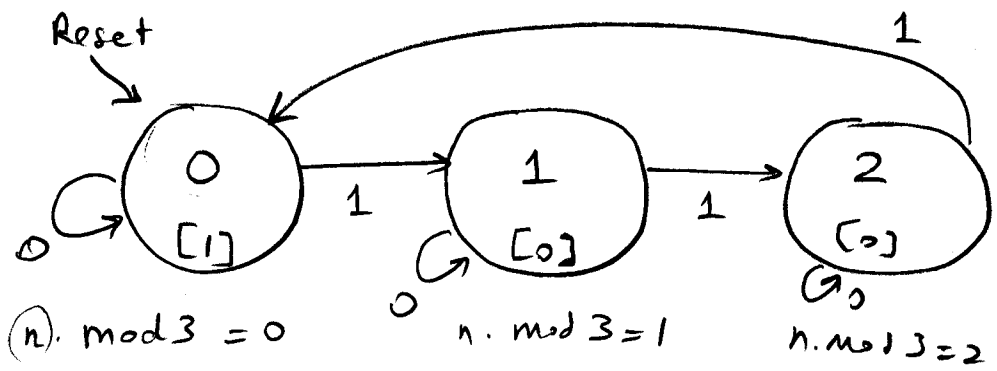
Example operation:

x	1	0	1	1	1	0	0	1	1	0	0	...
y	1	0	0	0	1	0	0	0	0	1	1	...

Since zero (1's) is a multiple of 3.

State diagram:

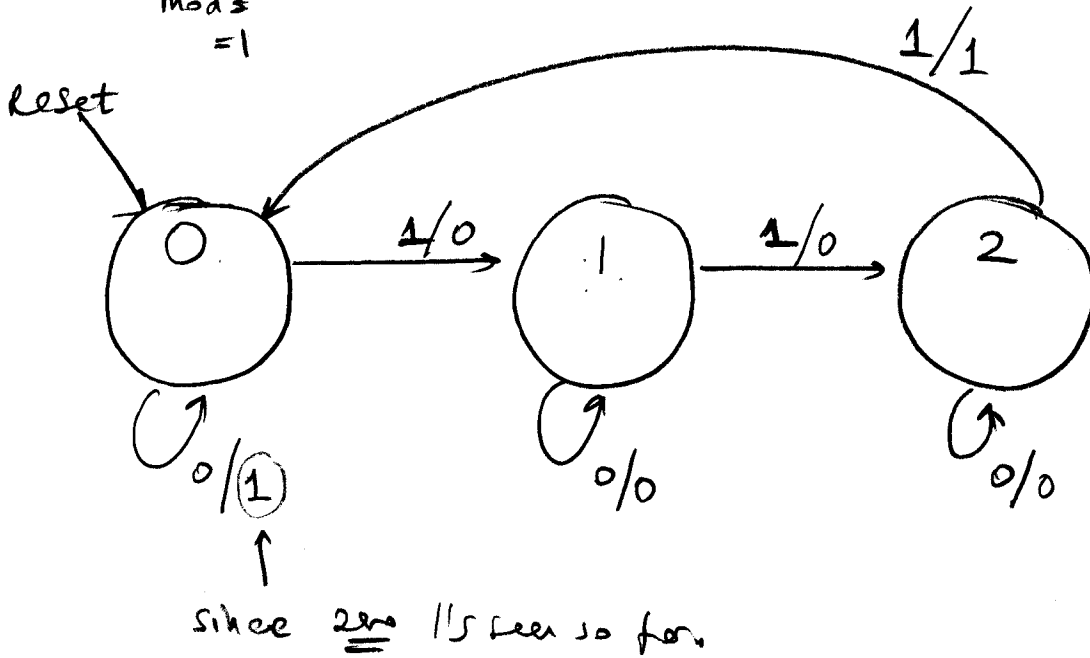
- Advice: Draw the start state and ^{First} successⁿ paths.



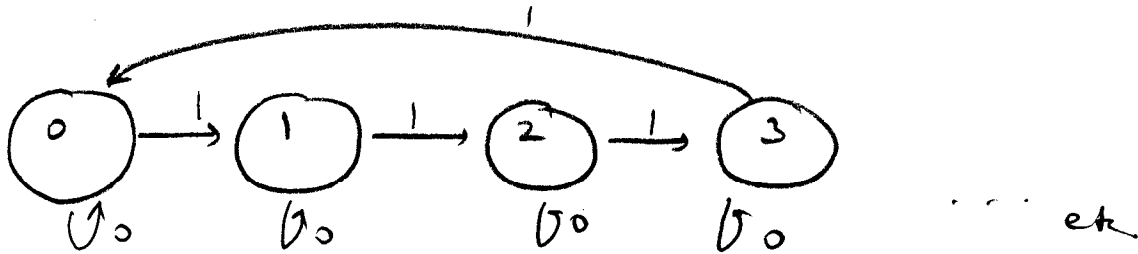
Implement it as a Mealy machine:

	1	1	2	3	4	4	4	5	6	6	6
X	1	0	1	1	1	0	0	1	1	0	0
Y	0	0	0	1	0	0	0	0	1	1	1

mod 3
= 1

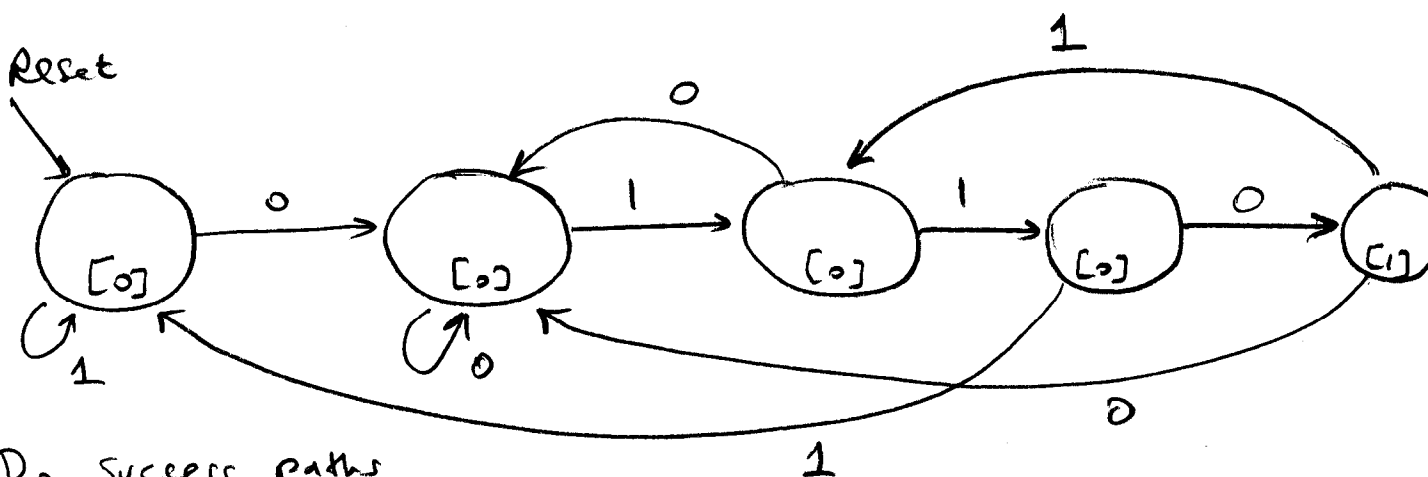


A.2 : similarly: ... multiple of 4:



③ Recognizing patterns:

- B.1** - observe a bitstream, one bit at a time.
- Report it (output a 1) whenever the last 4 bits you have observed are the pattern. 0110.
- (overlapping patterns are allowed.)



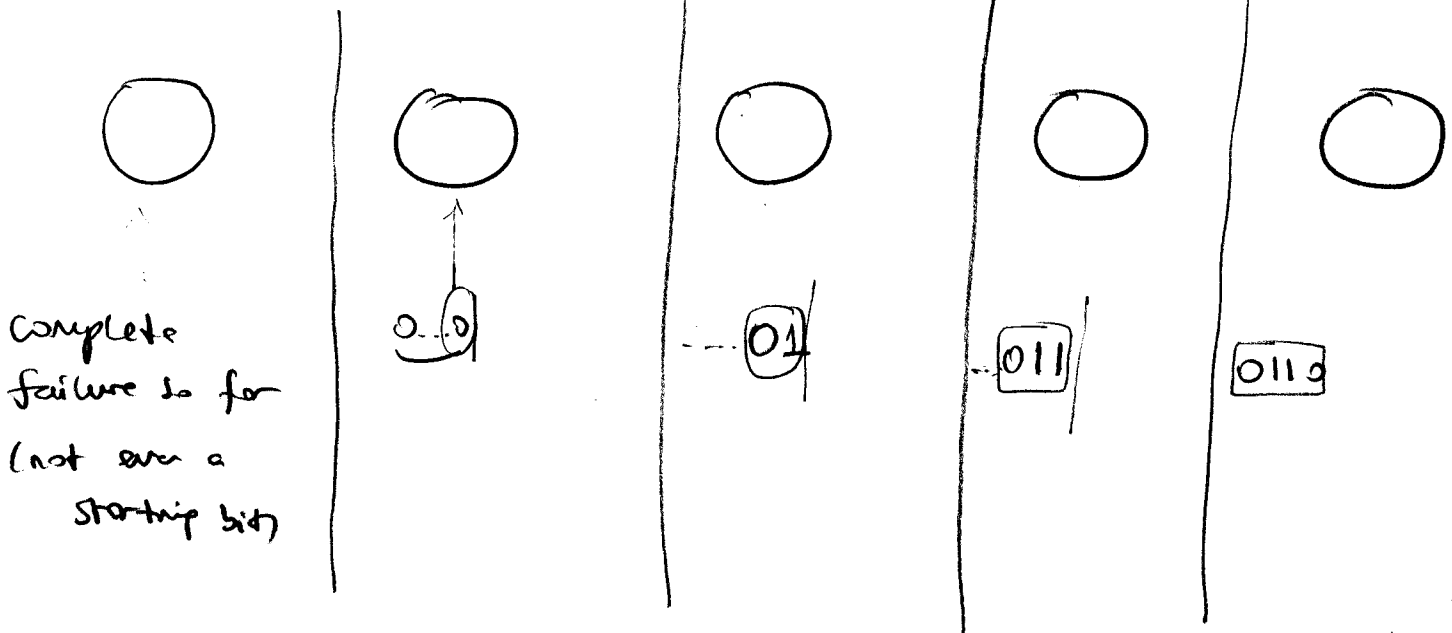
- Do success paths
- ✓ 0110110
 - ✓ 01100110
 - ✓ 0110000110
- Cover / draw remaining ones:

11...10110

010

0111

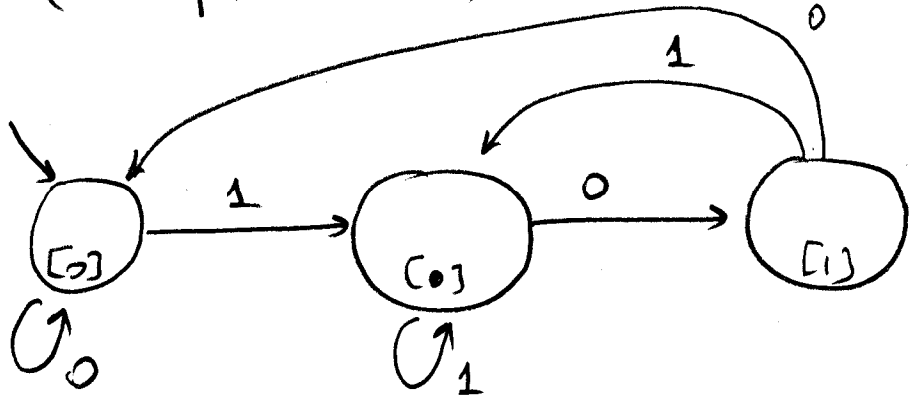
State interpretable:



- 5 states
- seems irreducible.
- can do state assignment. (3 bits per state)

B.2 - Detect the pattern: 10

-(overlaps allowed.)



↓
accumulated
nothing
so far

↓
accumulated
...||

101 ↓
accumulate
10

101

100

B.3 - Detect the pattern: 011 OR 101
 - (overlaps are allowed)

