Discussion 5
SDS & FSM

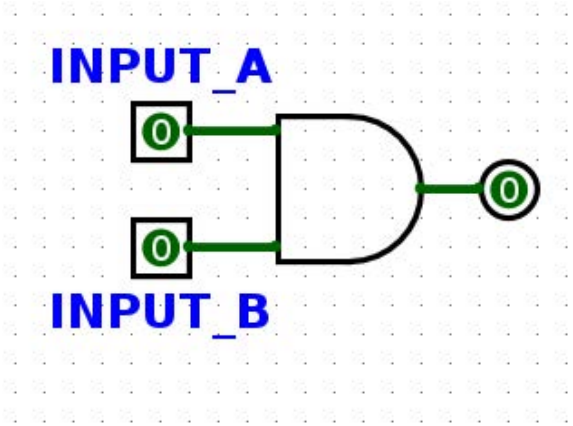
Zilin Si

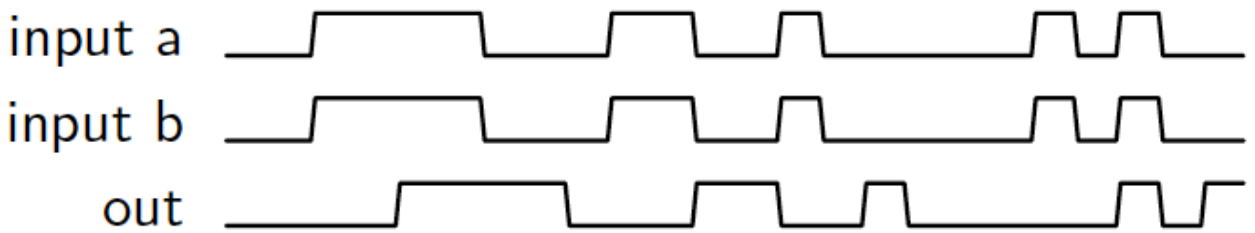
Time schedule reminder

- Next Wednesday(3rd April) is about mid-term 1
- This Monday’ s discussion is about SDS & FSM
- This Wednesday’ s discussion is the review of mid-term
- Next Monday’ s discussion is the review of mid-term
- Next Wednesday’ s discussion is about SDS & FSM
- Feel free to come and listen as usual time
- Half of TAs will join the review discussion

State Intro

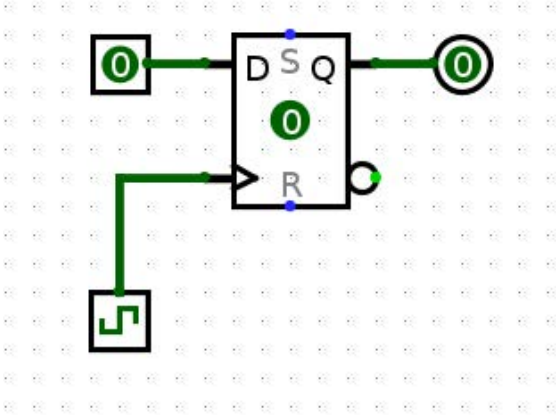
- Combinational logic

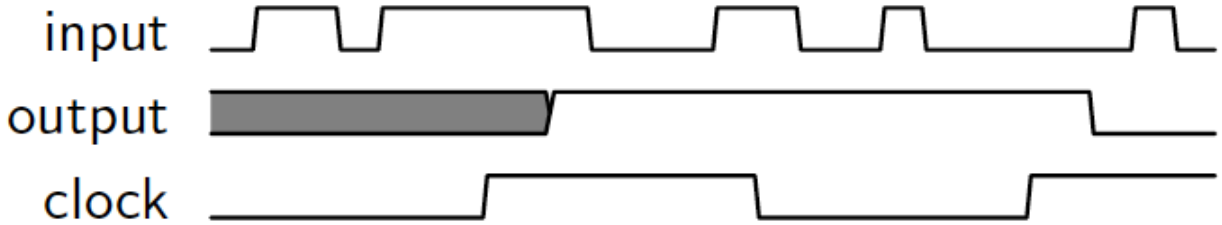




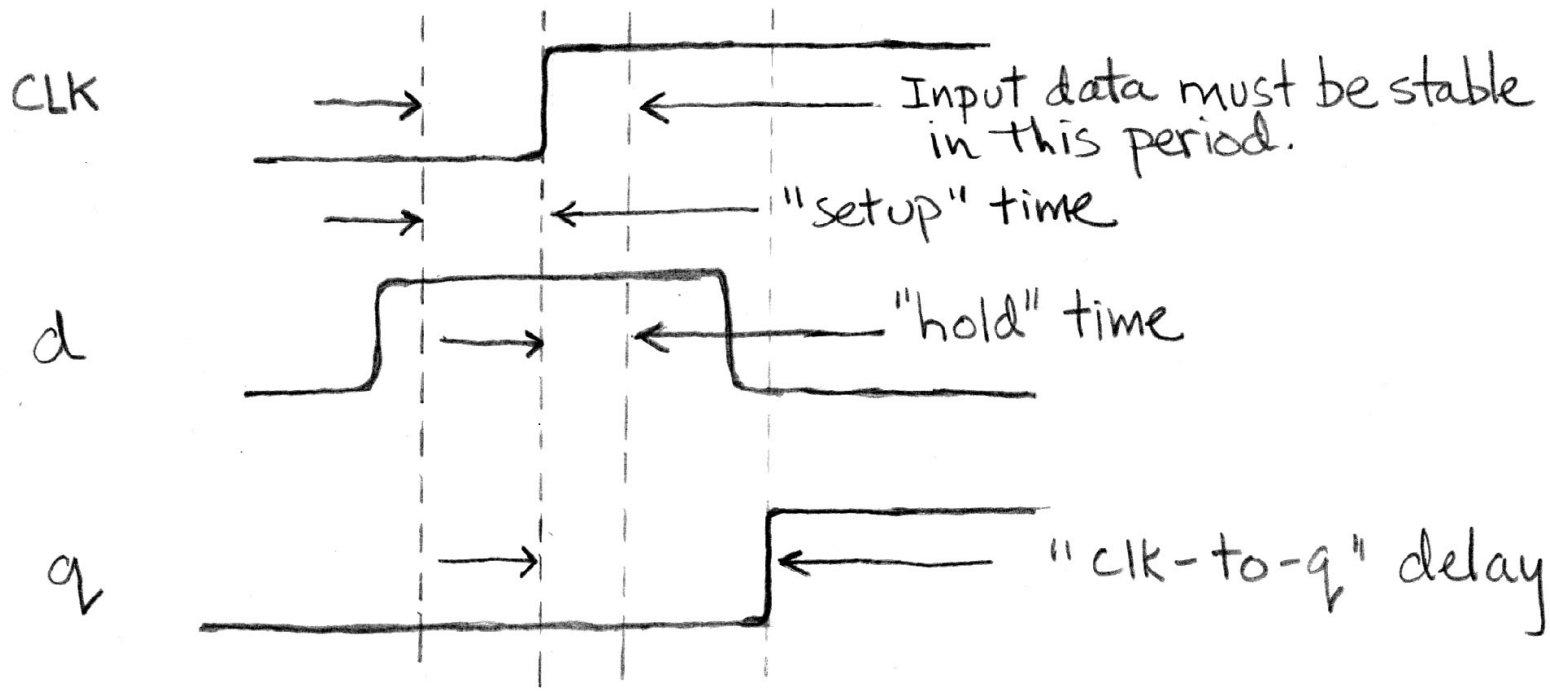
State Intro

- State elements

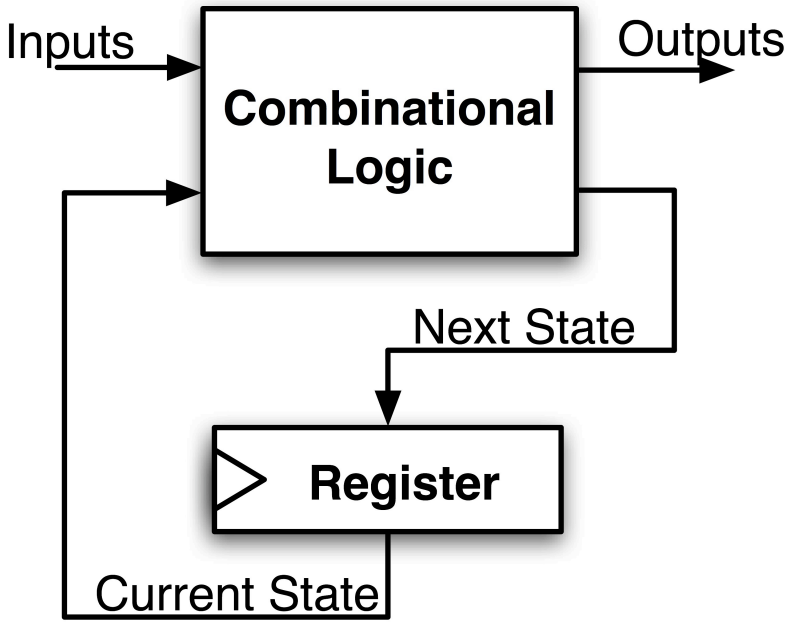




SDS


CLK
d
q
Input data must be stable in this period.
"setup" time
"hold" time
"clk-to-q" delay

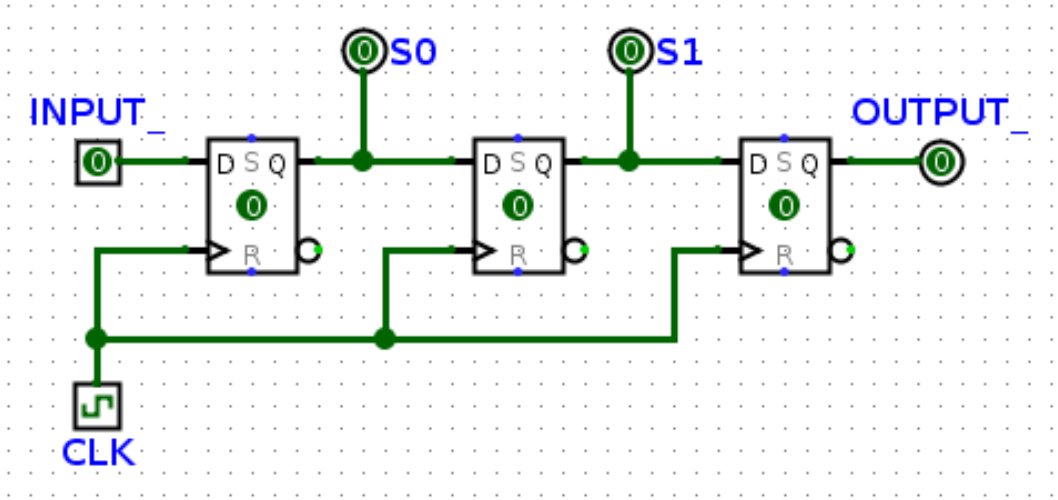
SDS

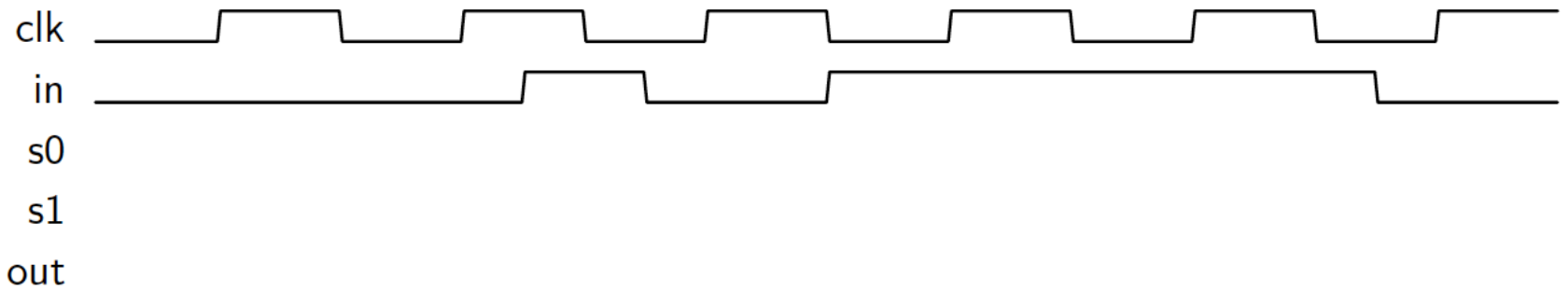


Max Delay = CLK-to-Q Delay
+ CL Delay
+ Setup Time

Example 1

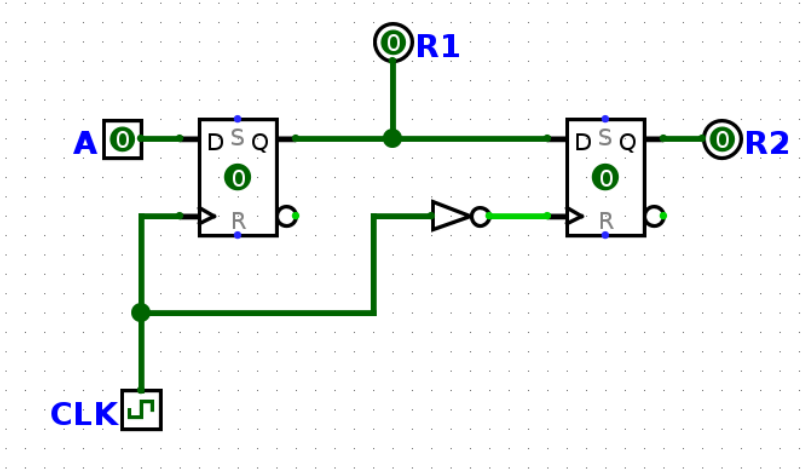
The clock period(rising edge to rising edge) is 10ns. For every register, clk-to-q delay is 1ns and set up/hold time are 0ns. There is no delay associated with a NOT gate.

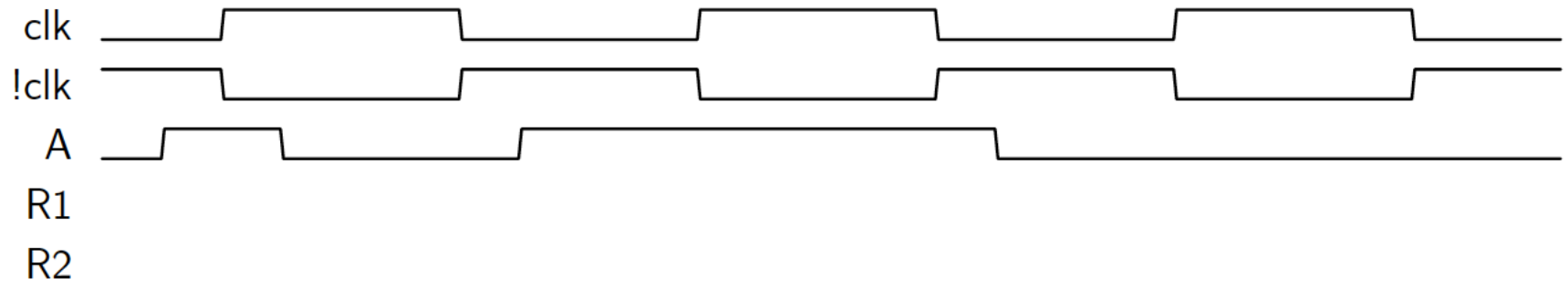




Example 2

The clock period(rising edge to rising edge) is 10ns. For every register, clk-to-q delay is 1ns and set up/hold time are 0ns. There is no delay associated with a NOT gate.

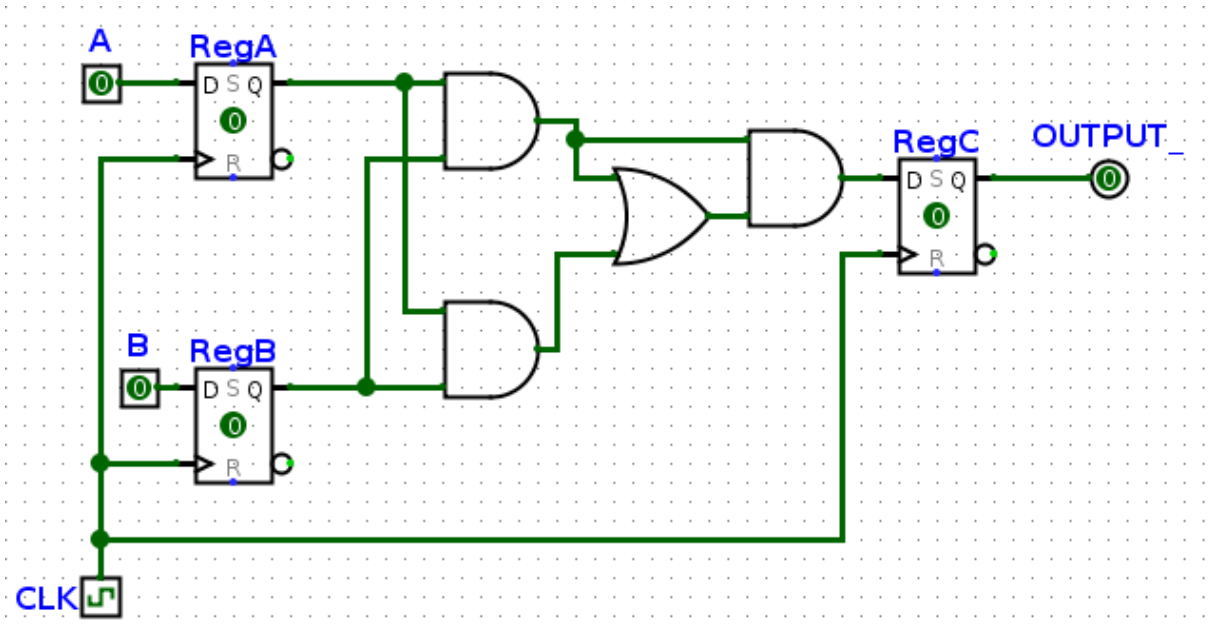




Example 3

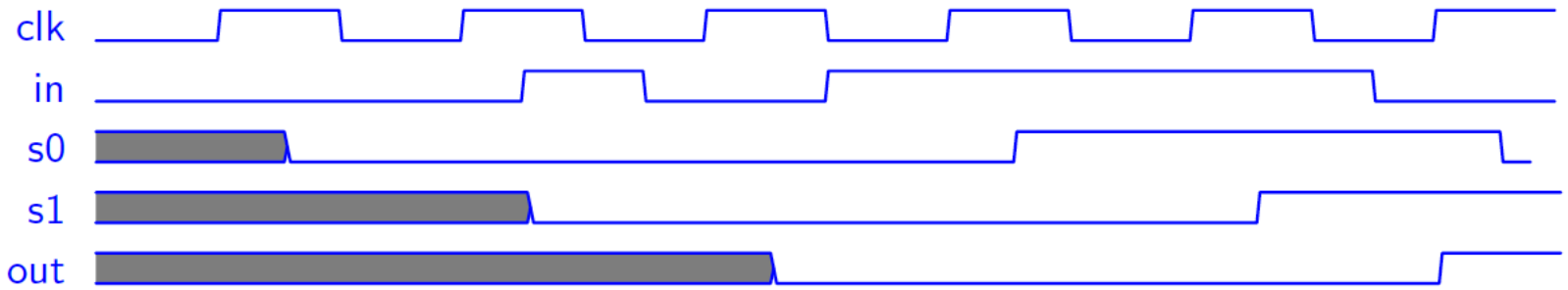
In the circuit below, RegA and RegB have set up, hold, and clk-to-q times of 4ns, all logic gates have a delay of 5ns, and RegC has a set up time of 6ns.
Question:

- What is the maximum allowable hold time for RegC?
- What is the minimum acceptable clock cycle time for this circuit, and clock frequency does it correspond to?

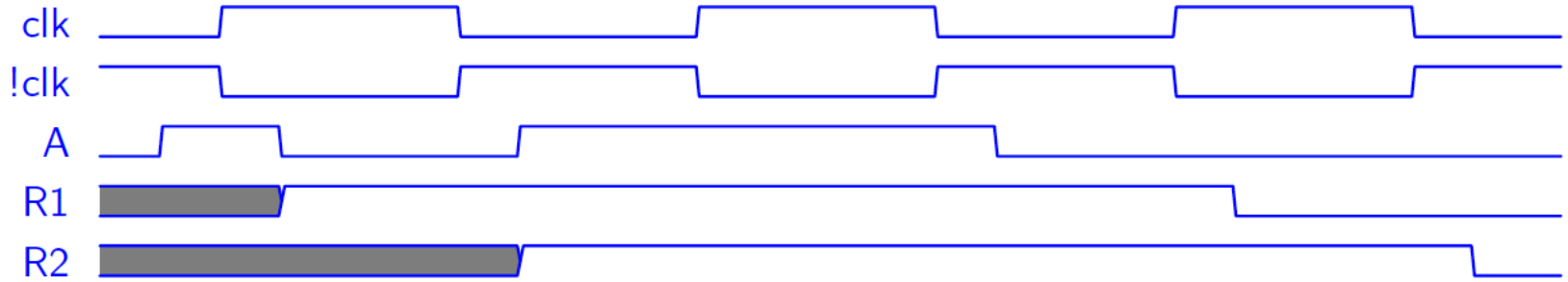


Solutions for SDS

- Example 1



- Example 2



Solutions for SDS

- Example 3
  - The maximum allowable hold time for RegC is how long it takes for RegC’ s input to change, so (clk-to-q of A and B) +CL = 4+(5+5) = 14ns.
  - The minimum acceptable clock cycle time is clk-to-q + longest CL time + set up time = 4+(5+5+5)+6 = 25ns.
  - 25ns corresponds to a clock frequency of (1/(25$*\mathbf{10}^{-9}$)) s^{-1} = 40 MHz/

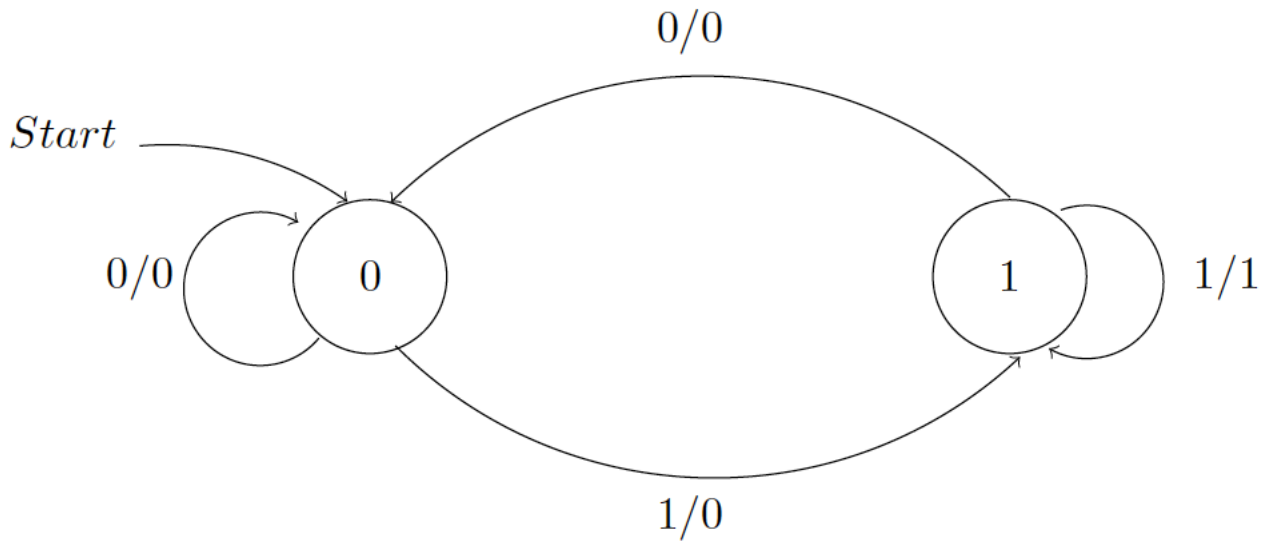
FSM

- Automatons are machines that receive input and use various states to produce output. A finite state machine is a type of simple automaton where the next state and output depend only on the current state and input. Each state is represented by a circle, and every proper finite state machine has a starting state, signified either with the label "Start" or a single arrow leading into it. Each transition between states is labeled [input]/[output].

Example 1

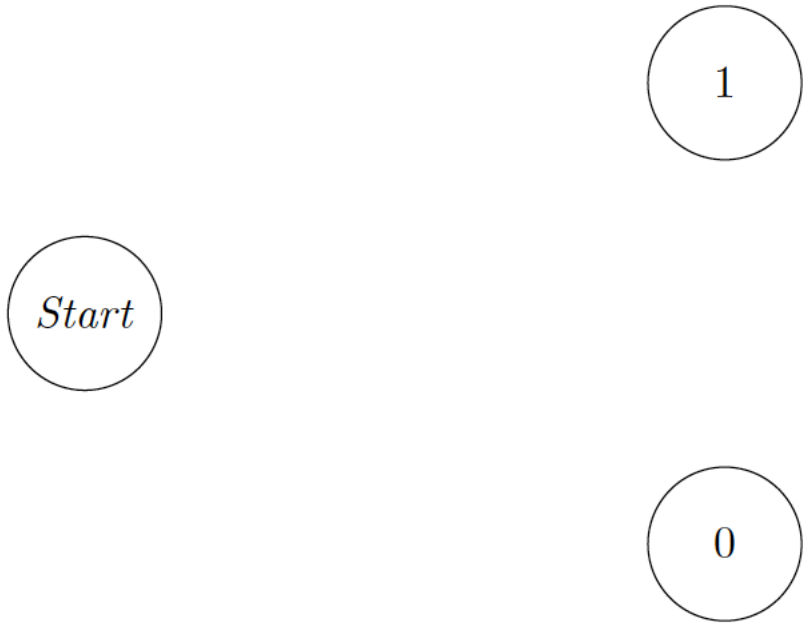
Question:

- What pattern in a bitstring does the FSM below detect?
- What would it output for the input bitstring "011001001110"?



Example 2

Fill in the following FSM for outputting a 1 whenever we have two repeating bits as the most recent bits, and a 0 otherwise. You may not need all states.



Solutions for FSM

- Example 1
  - The FSM outputs a 1 if detects the pattern " 11".
  - The FSM would output "001000000110".

- Example 2