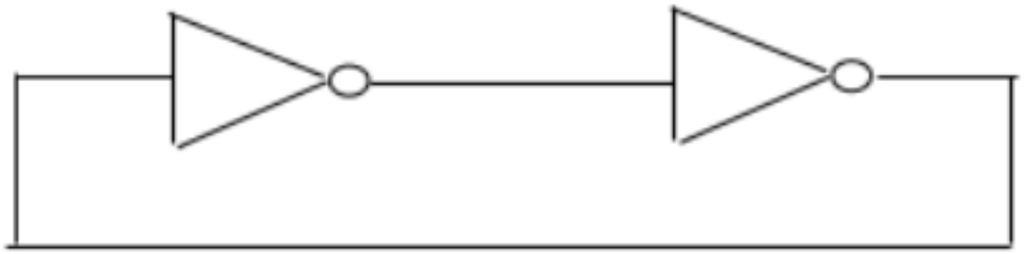
Australian National University

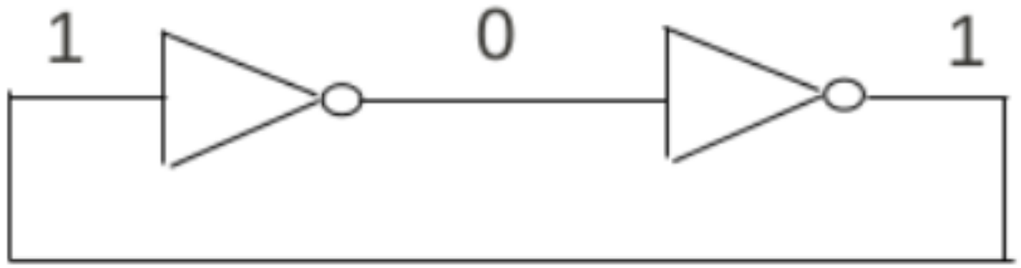
Registers and Finite State Machines

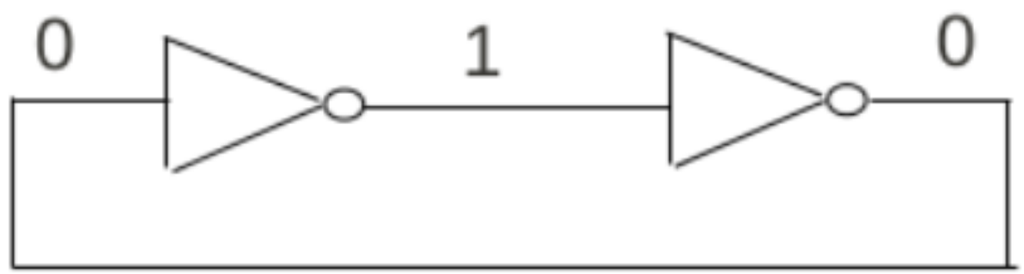
Eric McCreath

Memory

- State information can be maintained by connecting the output back into the input.



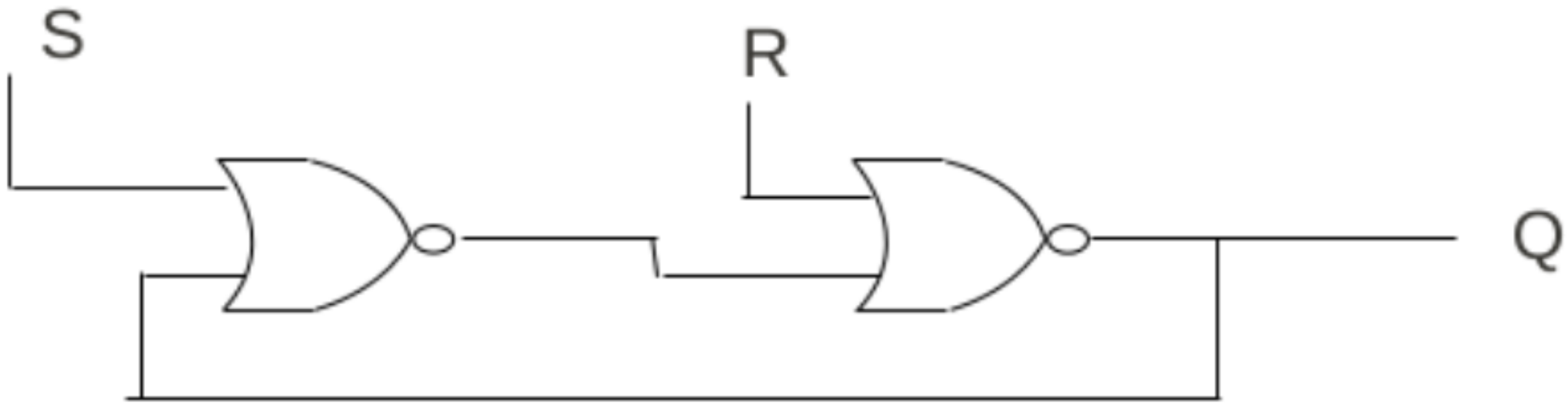




- This circuit can store one bit of information.
- However, there is no way of changing the state of memory within this circuit.

State

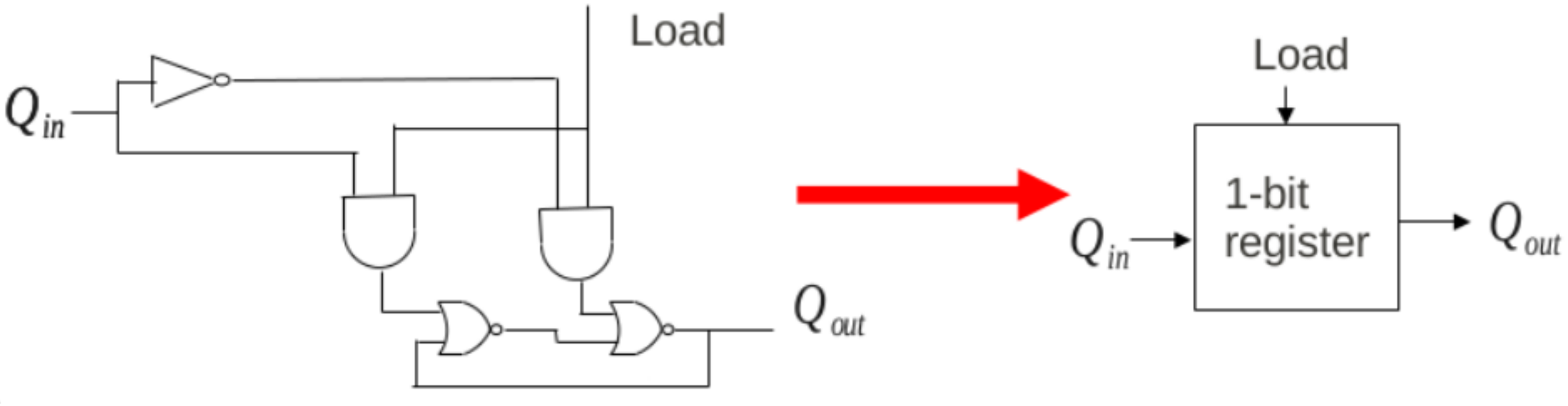
- By using two 'nor' gates, in much the same way as the two 'not' gates, we can set and reset the one bit of memory. This is known as a flip-flop.



- If S=0 and R=0 then whatever is in memory will be maintained. If S=1 then Q=1. If R=1 then Q=0.

Registers

- Usually some more gates are added to the flip-flop to make it easier to control.



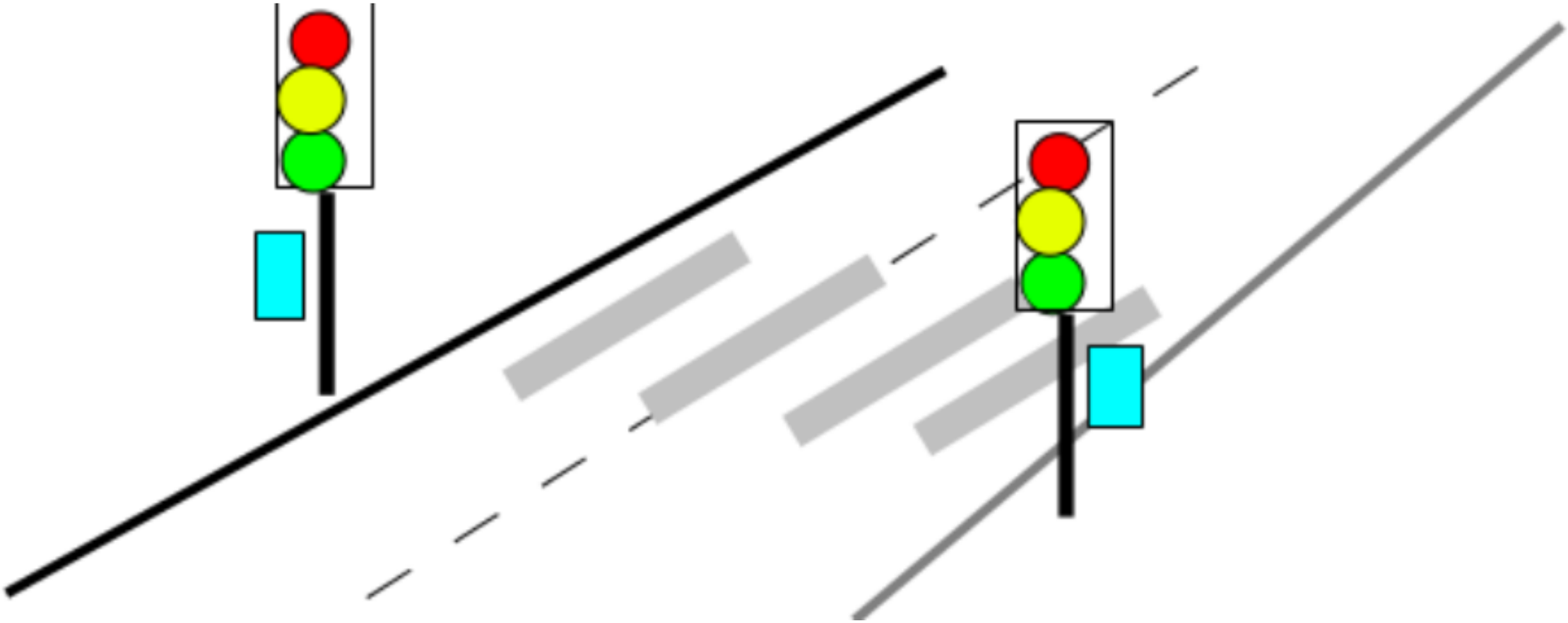
- We can also combine k of these 1 bit registers together to form a k-bit register.

Finite State Machine

- Most of the components that we have looked at so far are purely functional. However, in the design of a computer we require a component that can control and sequence events in time. These components are know as finite state machines.
- A finite state machine maintains its current states. Also it is given a number of inputs are returns a number of outputs.
- A clock, which is a signal that regularly changes between 1 and 0, is used to move the finite state machine from state to state.

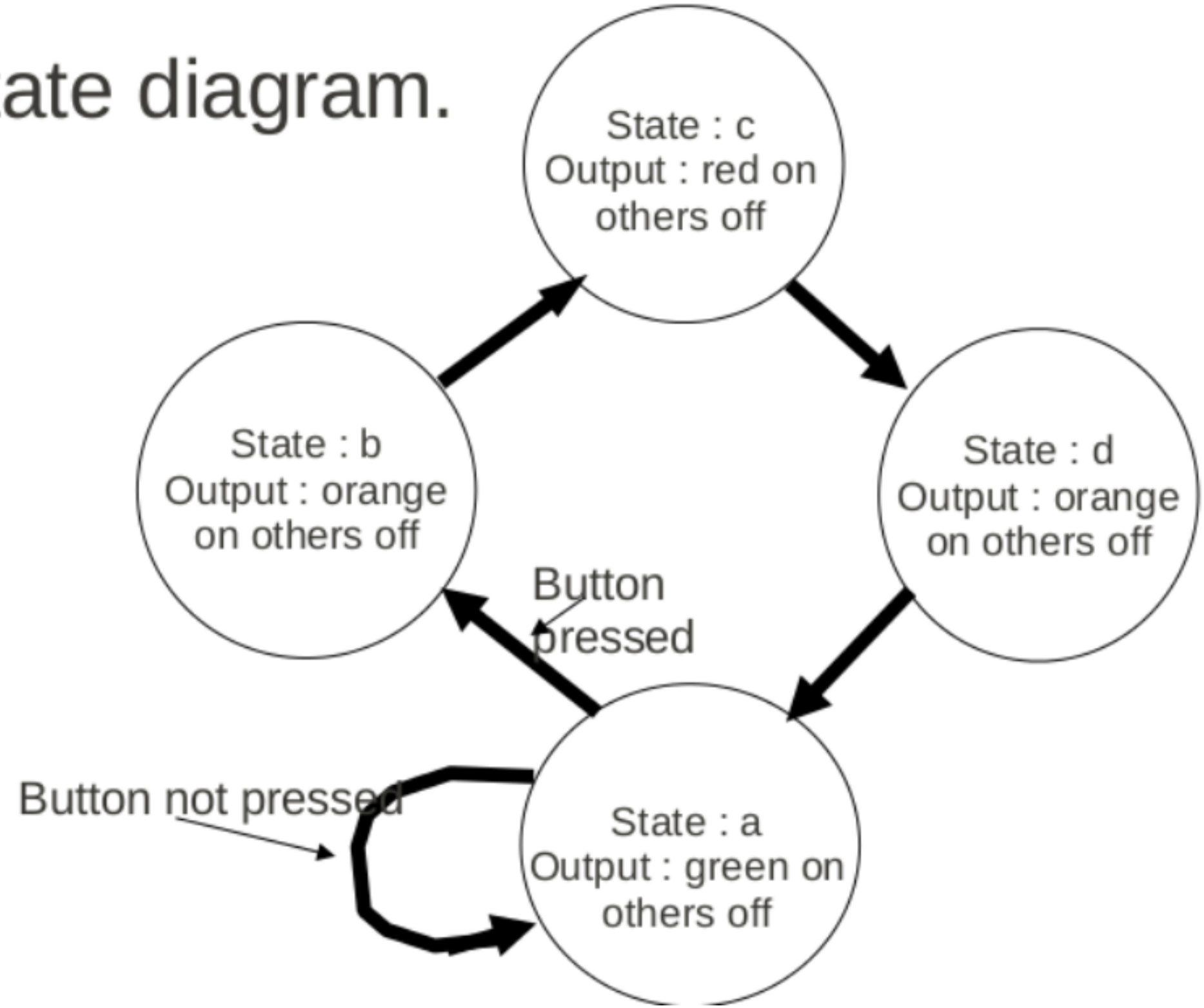
Finite State Machine - example

- Suppose we wish to construct a device that will control a traffic light at a pedestrian crossing.
- The device will have three outputs. The first turns the red light on, the second turns the orange light on, and the third turn the green light on.
- It will also have an input from the pedestrian's button.



Finite State Machine

- The state diagram.



Finite State Machine

- The current state can be stored in a two bit register.
- This can be used in conjunction with the input to work out the next state and the outputs.