Interfacing Mechanical Switches/Buttons

- Mechanical switches and buttons do not make solid, steady contact immediately after being pressed
- For a short few ms time, “bouncing” will ensue
- Need to “denounce” switches with software

What is state?

- The sum of all your previous experiences is what known as **state**
- Your “state” determines you interpretation of sense and thoughts
- In a circuit, ‘state’ refers to all the bits being remembered (registers or memory)
- In software, ‘state’ refers to all the variables that are being used

State Machine Block Diagram

- A system that utilizes state is often referred to as a state machine (a.k.a. Finite state machine/FMS)
- Most state machines can be embodied in the following form
    - Logic examines what’s happening right now (inputs) & int then past (state) to
        - Produce outputs
        - Update the state

State Diagrams

- A state machine can be visualized and represented as a flow chart (state diagram)
    - Circles or boxes represent state
    - Arrows show what input causes a transition
    - Output can be generated whenever you reach a particular state or based on the combination of state + input
- Examples: traffic lights, washing machine

Implementation Tips:

- Continuously loop
- Each iteration
    - Poll inputs
    - Use if statement to decide current state
    - In each state, update state appropriately based on desired input transitions from that state
    - Produce appropriate output from that state
- Ask questions like
    - What do I need to remember to interpret my inputs or produce my outputs?
    - Is there a distinct sequence of “steps” or “modes” that are used?