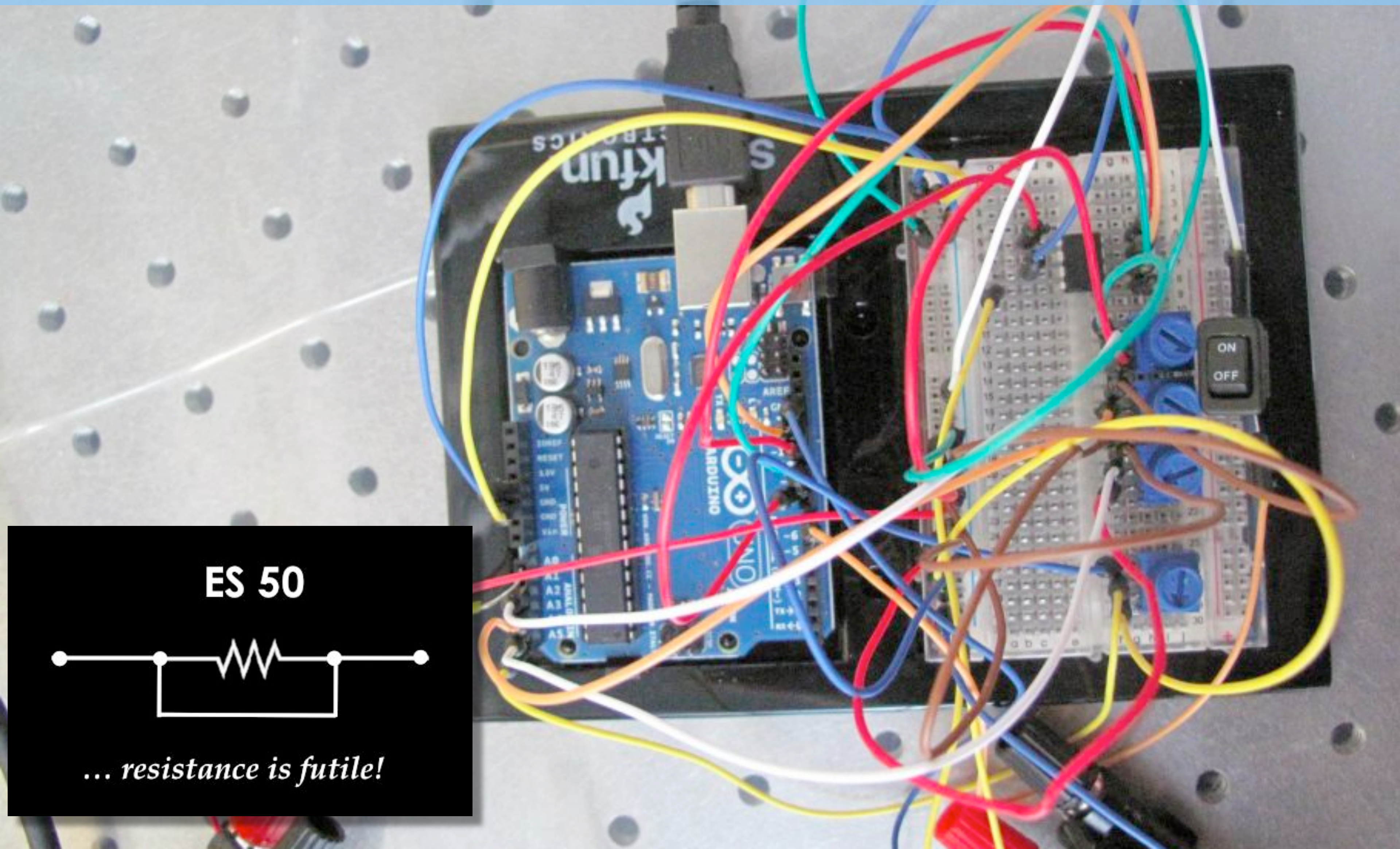
ES 50: Introduction to Electrical Eng.

Explore EE, satisfy a Gen Ed requirement & have fun



Lecture 13: Finite State Machines

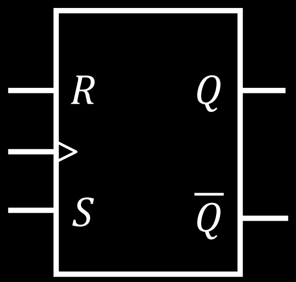
October 28, 2015

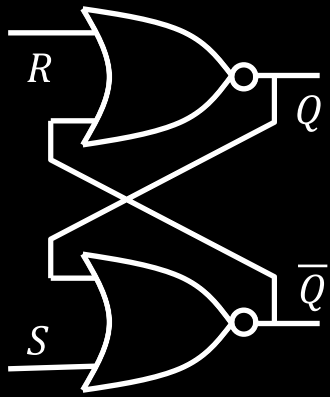
HARVARD
School of Engineering and Applied Sciences

Last Time

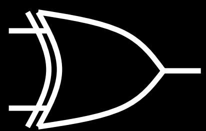
- Sequential Logic
- SR Latch
- Synchronous vs. Asynchronous Circuits
- Basic Timing Diagram

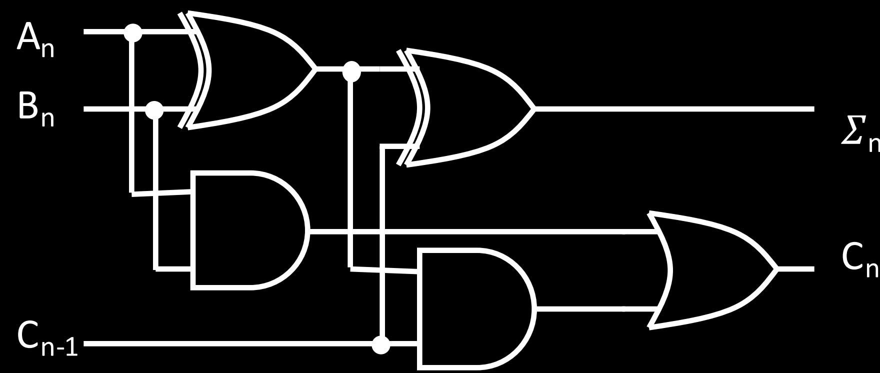
S	R	Q(t+1)
0	0	Q(t) no change
0	1	0
1	0	1
1	1	undefined

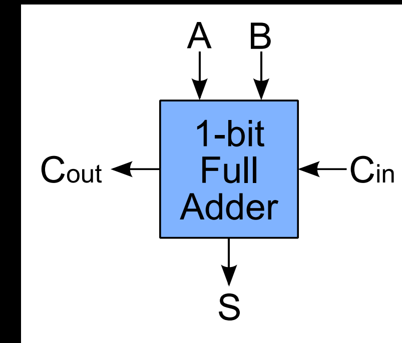


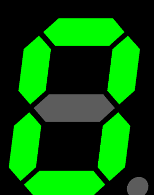


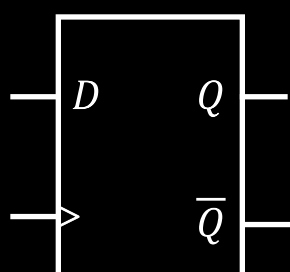
Digital Logic So Far

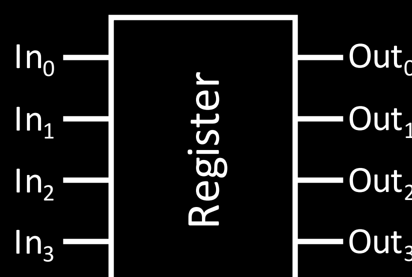


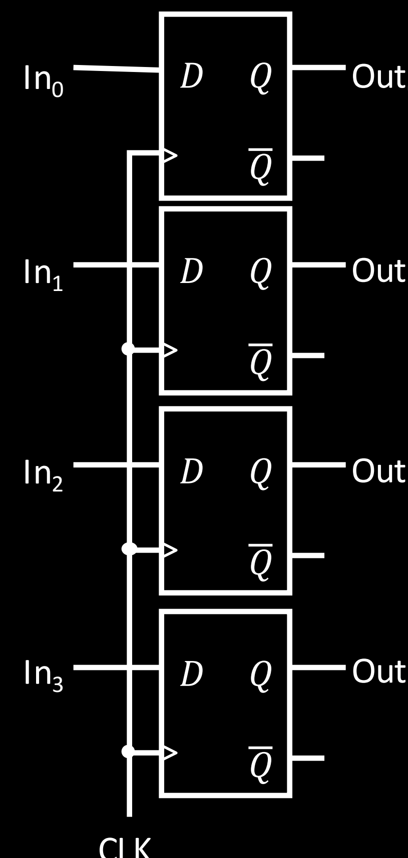












What else can we do with small simple logic circuits?

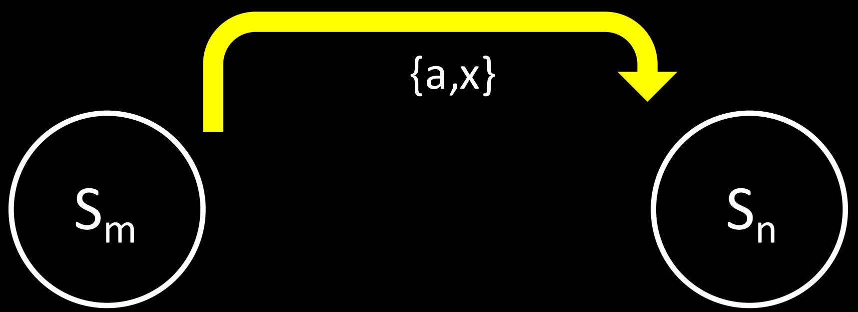
Finite State Machine (FSM)

An abstraction used to design digital logic (computer programs). It is composed of a finite number of states, transitions, outputs, & inputs.

- Examples: vending machines, ATMs, counters, alarm systems…

FSMs can be implemented use sequential logic

- State diagram is used to represent the operation of a FSM



States: S_m, S_n
Inputs: a
Outputs: x

Each state is a result of all previous inputs

- FSM has memory so flip-flops are needed to represent states

Output of FSM depends on current state & current inputs

Turnstile

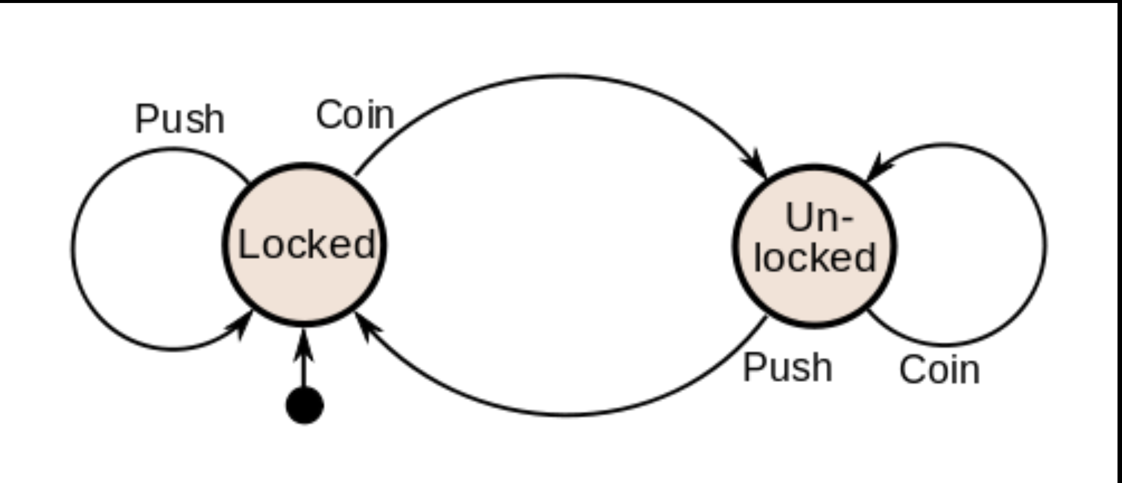
Used to control entry of exit into some location

Possible States

- Locked & Unlocked

Possible Actions

- Insert Coin/Payment
- Push turnstile

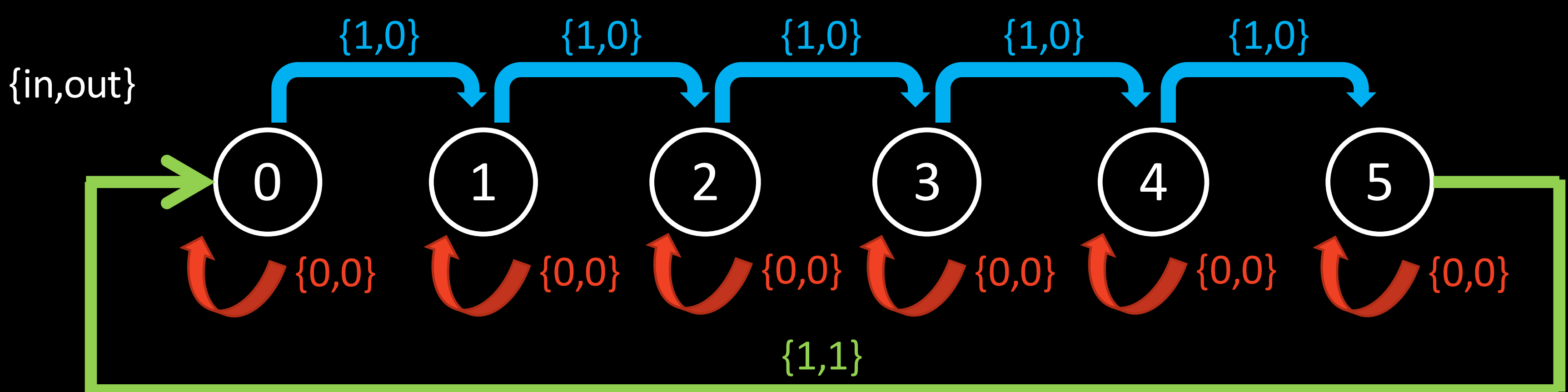


Procedure for Synthesis of FSM

1. Name all possible states
2. Consider all possible inputs, for every state
3. Label input & outputs for each transition between states
4. Draw the diagram: you need to consider every possible input in each state
    - Otherwise transitions of the FSM may be undefined for certain input aka machine freezes or blue screen of death

Example: Mod 6 counter

- Counter counts up when a button is pressed. When it counts up to 6, light bulb lights up, and counter is reset to state 0.
- Inputs: {button not pressed, button pressed} ⇔ {0,1}
- Outputs: {light off, light on} ⇔ {0, 1}
- States: {0, 1, 2, 3, 4, 5}



Example: 2011 Quiz

The elevator in Maxwell-Dworkin building operates between the ground floor (G) and any of the three floors (1, 2, 3). The elevator has 4 buttons (G, 1, 2, 3), two LED indicators (UP & DOWN), & a sound indicator (SOUND). This elevator can be realized as a finite state machine that operates according to the following rules:

- A student pushes one of the four buttons to select the desired destination floor. Once one of the floors is selected, additional selections are ignored.
- If the destination is higher than the origin, the elevator goes up to the desired floor & UP arrow is lit.
- If the destination is lower than the origin, the elevator goes down to the desired floor, & DOWN arrow is lit.
- If the destination is the same as the origin, the elevator does not move, & SOUND is activated to remind a sleepy student that she/he is already at the desired floor.
  - You do not have to worry about sending signals to the elevator's motor to tell it to start or stop moving. Also, you don't have to consider the control panels at each floor.

A. What are the states, inputs and outputs of this finite state machine?

B. Draw the state diagram of the FSM.

C. How many flip-flops would be needed to implement this FSM? How many input and output wires would you need, assuming that both inputs and outputs are encoded.

MD Elevator Finite State Machine

States: {G,1,2,3}

Inputs: {G,1,2,3}, but we can also do nothing → {nothing,G,1,2,3}

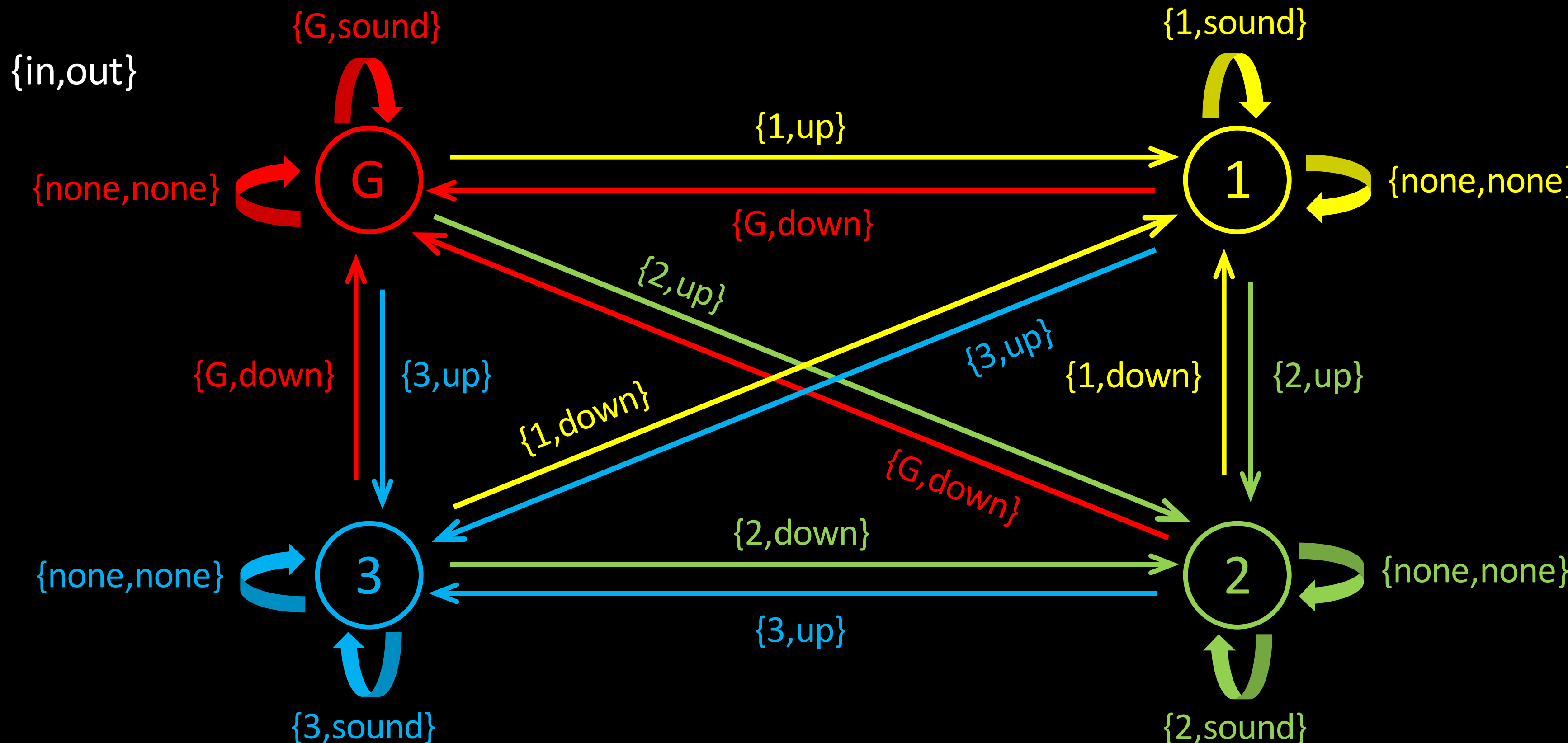
Outputs: {up,down,sound,nothing}

Flip-Flops: 2 (4 states)

Input wires: 3 (5 inputs)

- If we are clever, this be reduced to two by using the CLK as an input

Output wires: 2 (4 outputs)



Next Time

Images as 2D Signals!