CS 2110 - Lab 04

K-Maps, Sequential Logic, Intro to State Machines

Wednesday, June 1, 2022

Lab Assignment: Canvas Quiz

1. Go to Quizzes on Canvas
2. Select Lab 04, password: **flip-flop**
3. Get 100% to get attendance!
    a) Unlimited attempts
    b) Collaboration is **allowed**!
    c) Ask your TAs for help :)

Homework 1

- Released!
- **Due Thursday, June 2nd at 11:59 PM (24 hr grace period)**
- Files available on Canvas
- Submit on Gradescope
  - **Double check that your grade on Gradescope is your desired grade!**

Homework 2

- Released!
- **Due Monday, June 6th at 11:59 pm (24 hr grace period)**
- Files available on Canvas
- Submit on Gradescope (unlimited submissions)

Homework 3

- Released on Friday, June 3th
- **Due Monday, June 13th at 11:59 pm (24 hr grace period)**
- Files will be available on Canvas
- Submit on Gradescope (unlimited submissions)

Quiz 1

- **Scheduled on Monday, June 6th**
- Quiz opens at your assigned lab time (unless you have already established a different time with your professor)
- Full 75 minutes to take the quiz!
- Closed Notes
- After the quiz, you will join lab for the remaining period.
- A topic list will be posted on Canvas

Timed Lab 1

- **Scheduled on Wednesday, June 8th**
- 75 minutes to complete at the beginning of lab, then lab will be resumed.
- Will cover HW02 material primarily
  - Open notes/book/Internet
  - Unlimited submissions throughout the TL period.

Today's Topics

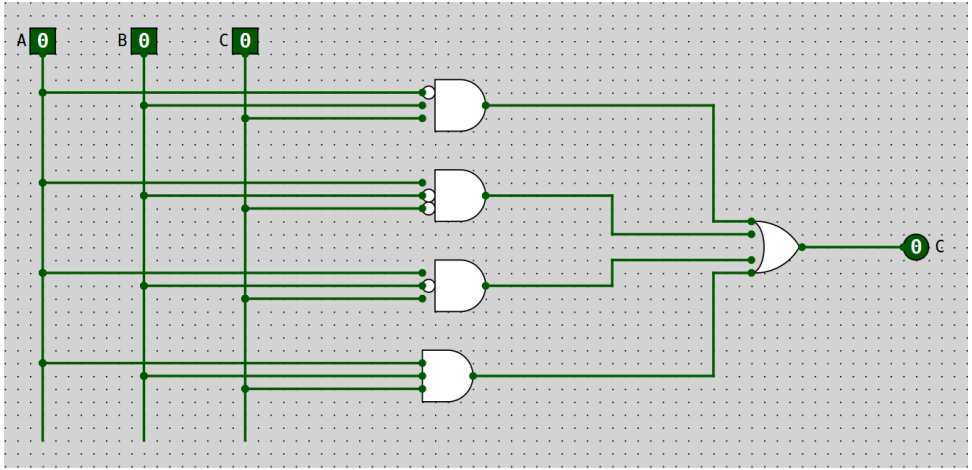
- Karnaugh Maps (K-Maps)
- Sequential Logic
  - What is sequential logic?
  - Level-triggered vs edge-triggered logic
  - RS Latches, Gated D-Latches and D Flip-Flops
- Address Space and Addressability
- Intro to State Machines

Logic Conversion

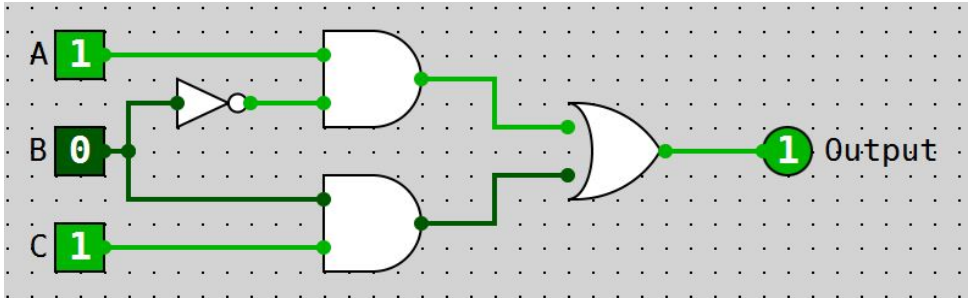
A	B	C	Func (A,B,C)
0	0	0	X
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

Sum of products expressions are long and unwieldy, and they require lots of logic gates.

How can we get from the top circuit to the bottom one?



TO



Karnaugh Maps (K-Maps)

- A method of simplifying Boolean expressions by grouping together related terms
- Results in the simplest sum-of-products expression possible
- Allows for “don’t care” outputs

Step 1: Create the K-Map

A	B	C	Func(A,B,C)
0	0	0	X
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

- Distribute variables across the rows and columns using Gray code order
- Fill in corresponding entries

	AB	AB'	A'B'	A'B
C				
C'				

	AB	AB'	A'B'	A'B
C	1	1	0	1
C'	0	1	X	0

Step 2: Make Groupings

	AB	AB'	A'B'	A'B
C	1	1	0	1
C'	0	1	X	0

	AB	AB'	A'B'	A'B
C	1	1	0	1
C'	0	1	X	0

Rules of K-map grouping:

1. Groups must be rectangular (but they may wrap around edges!)
2. Groups may only contain “1”s or “X”s
3. All “1”s must be contained within at least one group
4. Groups must be as large as possible
5. Have as few groups as possible
6. The size of a group must be a power of 2 – note that you may have a group of size 1

Step 3: Write Simplified Expression

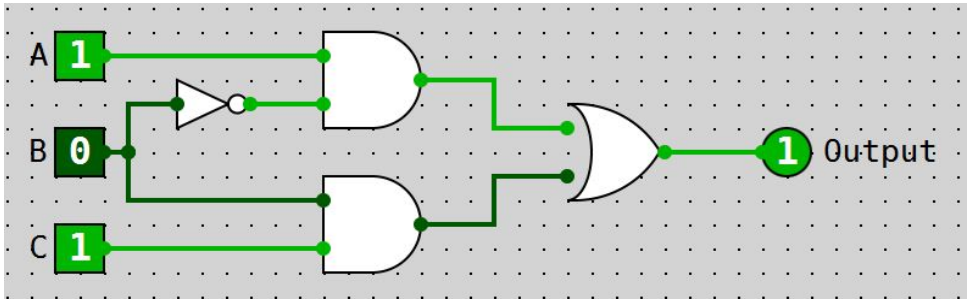
	AB	AB'	A'B'	A'B
C	1	1	0	1
C'	0	1	X	0

	AB	AB'	A'B'	A'B
C	1	1	0	1
C'	0	1	X	0

Pull out the simplified expression based on the K-map groupings

(ABC + A'BC) + (AB'C + AB'C')

BC + AB'



K-Map Reducing Practice

Q1	AB	AB'	A'B'	A'B
C	1	0	1	1
C'	X	0	0	1

Q2	AB	AB'	A'B'	A'B
C	1	1	0	0
C'	0	0	1	1

Q3	AB	AB'	A'B'	A'B
C	0	1	0	1
C'	1	0	1	0

Q4	AB	AB'	A'B'	A'B
C	1	1	1	1
C'	1	1	0	0

K-Map Reducing Practice

Q1	AB	AB'	A'B'	A'B
C	1	0	1	1
C'	X	0	0	1

overlap

Q2	AB	AB'	A'B'	A'B
C	1	1	0	0
C'	0	0	1	1

Q3	AB	AB'	A'B'	A'B
C	0	1	0	1
C'	1	0	1	0

overlap

Q4	AB	AB'	A'B'	A'B
C	1	1	1	1
C'	1	1	0	0

K-Map Reducing Practice

Q1	AB	AB'	A'B'	A'B
C	1	0	1	1
C'	X	0	0	1

B + A'C

Q2	AB	AB'	A'B'	A'B
C	1	1	0	0
C'	0	0	1	1

AC + A'C'

Q3	AB	AB'	A'B'	A'B
C	0	1	0	1
C'	1	0	1	0

ABC'+AB'C+A'B'C'+A'BC
Impossible to simplify using sum-of-products!

Q4	AB	AB'	A'B'	A'B
C	1	1	1	1
C'	1	1	0	0

C + A

K-Map Reducing Practice

A	B	C	Func(A,B,C)
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	X
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

	A’B’	A’B	AB	AB’
C’	1	1	1	1
C	0	X	0	0

	BC	B’C	B’C’	BC’
A	0	1	1	1
A'	X	0	1	1

	A’B’	A’B	AB’	AB
C’	0	1	1	1
C	0	X	1	0

	BC	B’C	B’C’	BC’
A	0	0	1	1
A’	X	0	1	1

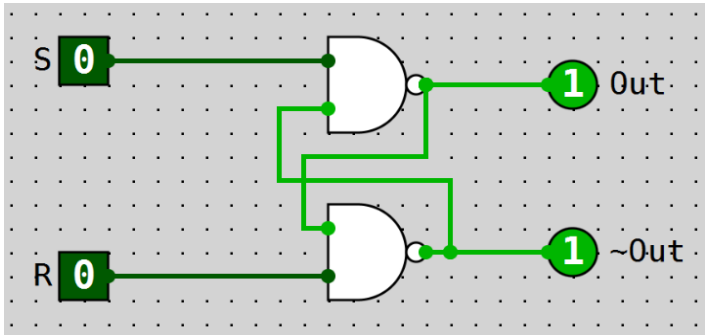
Combinational vs Sequential Logic

Combinational Logic: the outputs of the circuit depend only on the **current** combination of inputs. Examples?

Sequential Logic: output of the circuit depends on both the **current** inputs and **previous** values of the inputs. We will see some examples now.

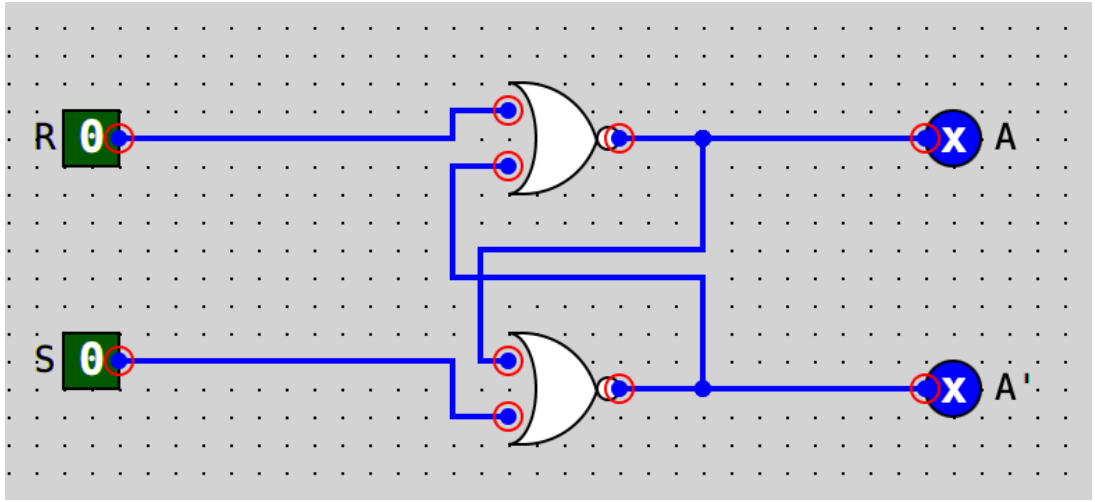
RS Latch

- The Basic Circuit of Sequential Logic
- Stores 1 bit
- 3 valid states:
  - R = 1, S = 1: output at this state depends on previous state
  - R = 1, S = 0: what is the output?
  - R = 0, S = 1: what is the output?



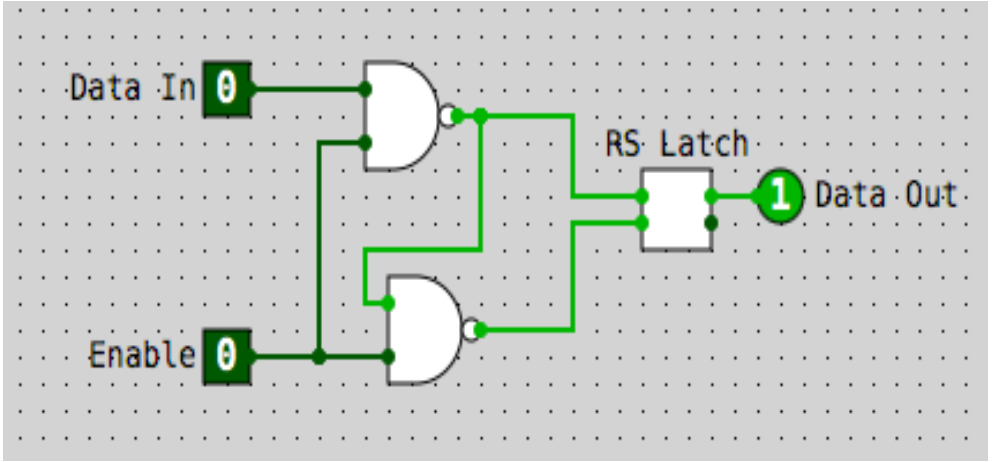
RS Latch

- R = 0, S = 0:
- R = 0, S = 1:
- R = 1, S = 0:
- R = 1, S = 1:



Gated D-Latch

- Essentially just an RS Latch with extra control
- Can choose when to save the output and what to save it as



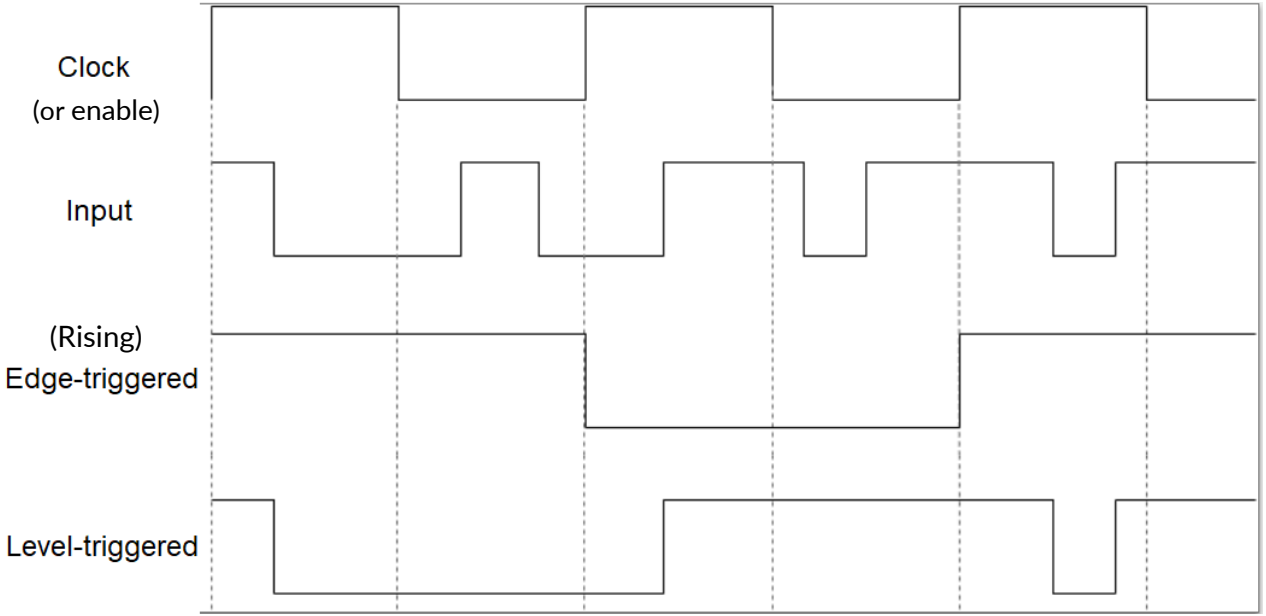
Level-Triggered Logic

- The previous two circuits are examples of level-triggered circuits.
- In level-triggered logic, the output can only change when the enable bit is 1
- When the enable bit is set to 0, then the output is unaffected by changes in the input

Edge-Triggered Logic

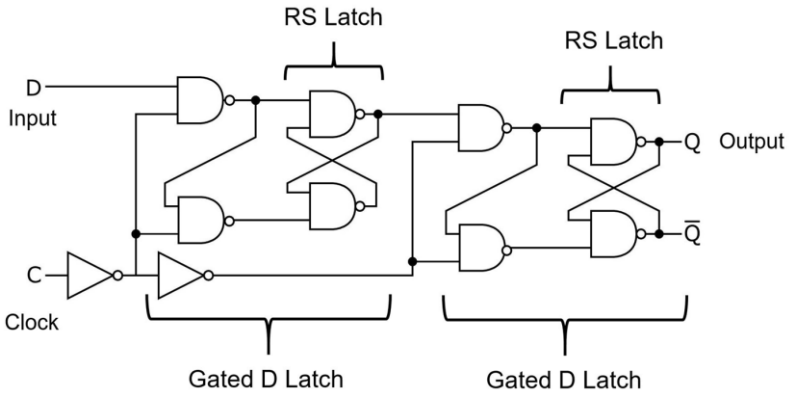
- Many sequential logic circuits are based on clocks instead of enable bits
- Rising-edge triggered logic: The output can only change when the clock changes from 0 to 1
- Falling-edge triggered logic: The output can only change when the clock changes from 1 to 0

Level Triggered vs Edge-Triggered Logic



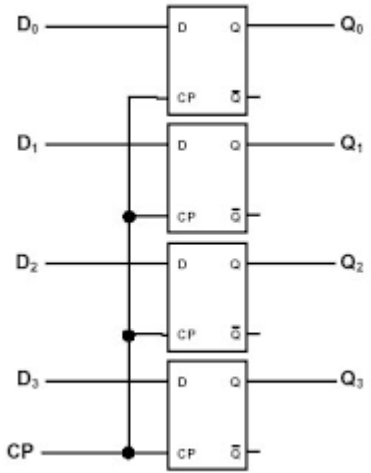
D Flip-Flops

Rising edge-triggered! The left D-latch updates when the clock is 0, and the right D-latch updates when it is 1.



Registers

- We can combine multiple latches to create a *register* to store multiple bits
- Essentially the same as a Gated D-Latch or a D Flip-Flop, but there are *n*-bit input or output lines instead of just one



Combinational Logic Components

- ALU
- Decoder
- Adder
- Multiplexer (Mux)

Sequential Logic Components

- RS Latch
- Gated D Latch
- Registers
- State machines

Address Space & Addressability

- Computer memory is made up of distinct *memory addresses*, each containing some amount of data
- Addressability: the amount of data stored at any given memory address
- Address space: the number of memory addresses that exist
  - An n-bit address line can represent 2^n memory addresses

Address Space & Addressability

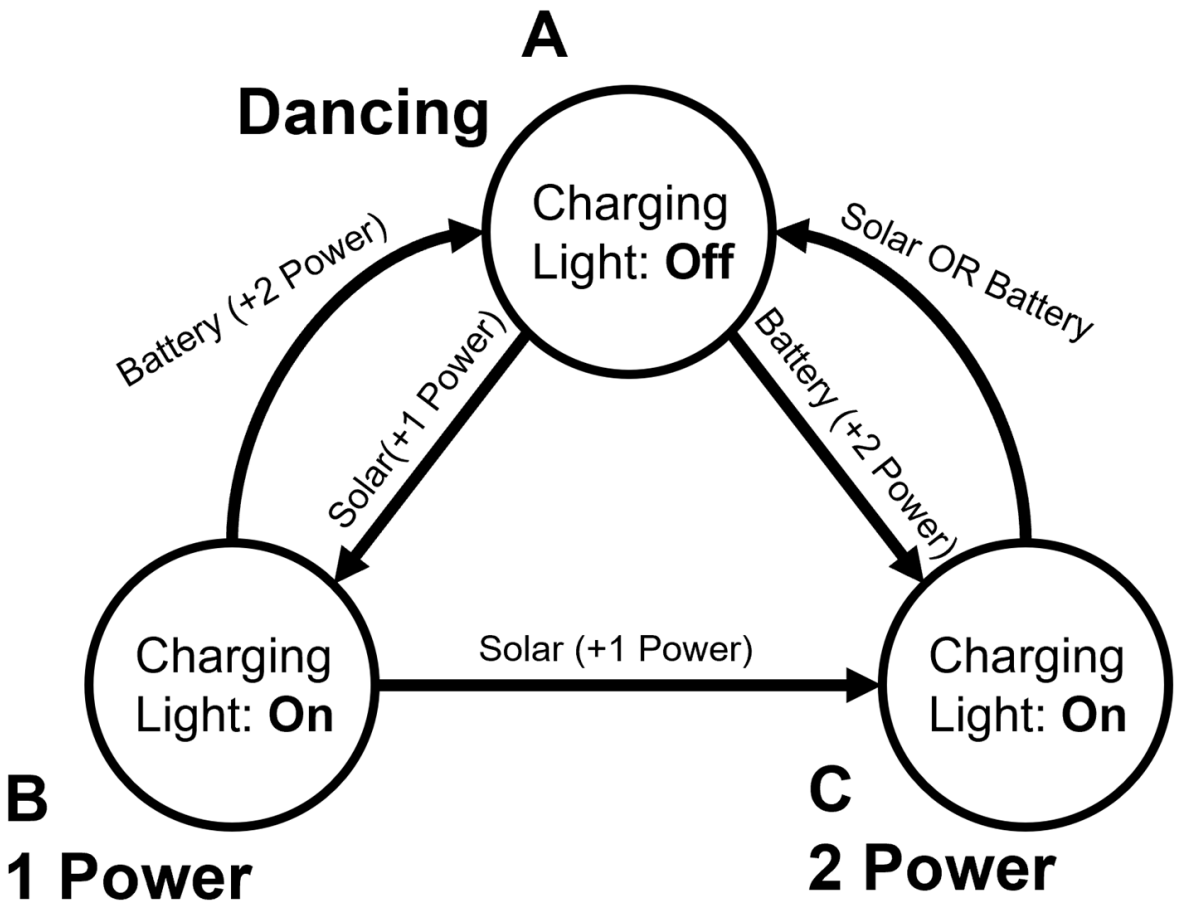
- If a computer uses 8-bit lines to represent its memory addresses, what is its address space?

- What is the addressability of a system with 16 KiB of memory, composed of 256 memory addresses?

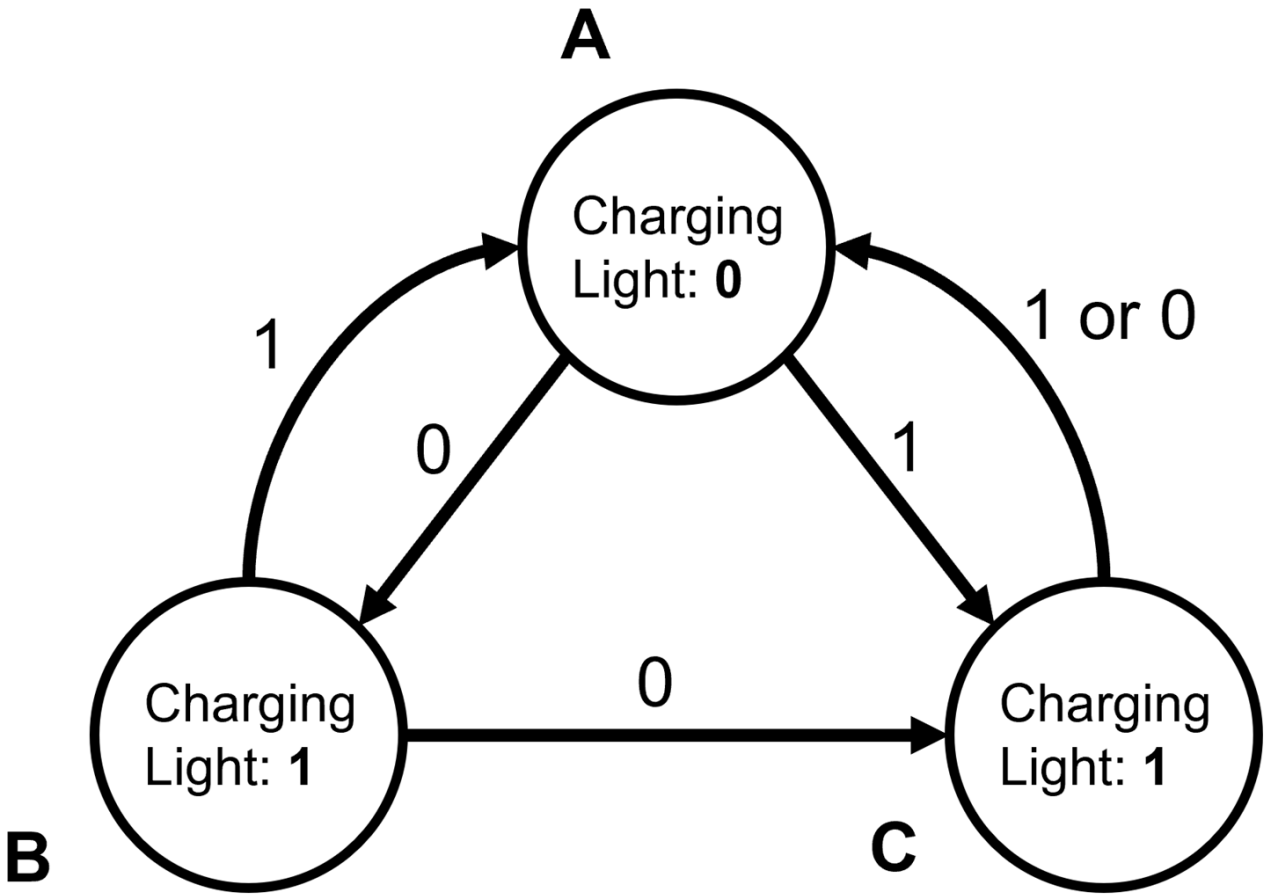
State Machines!

- Example: Dave, the Funky Dancing Robot Crab
  - Dave is a dancing robot crab. Before it can dance it must fully recharge (3 charges). It can recharge slow by solar energy (1 charge) or fast with a battery (2 charges).
  - Dave has 3 possible states:
    1. Dancing (requires 3 charges)
    2. Charging with 1 charge
    3. Charging with 2 charges

Dave's State Transition Diagram



State Transition Diagram - Representation



A word of caution...

- State machines are NOT event-driven; they are locked to the clock
- The state is always updated once per clock cycle, even if you transition into the same state
- This is true of all sequential logic in modern computers