Warmup

Which circuit will be created by the SystemVerilog below?

always - comb

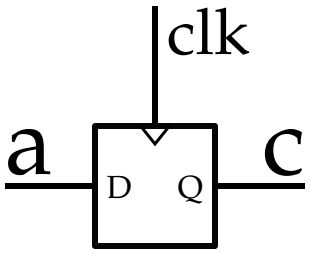
```
always @(a, b)
begin
   b <= a;
   c <= b;
end
```

better to do

b = a

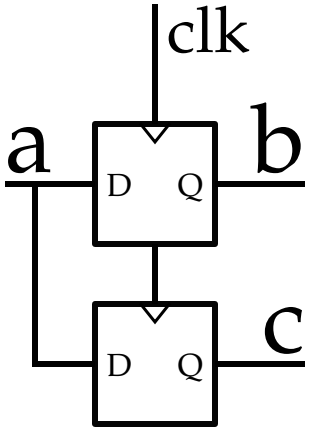
c = b

(A)

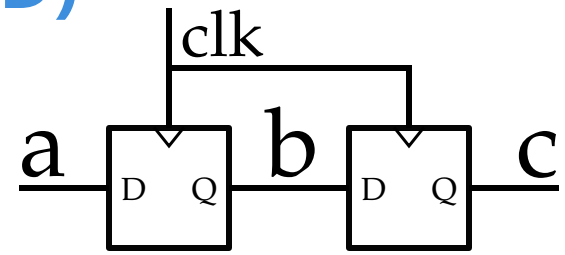


(B)

(C)



(D)



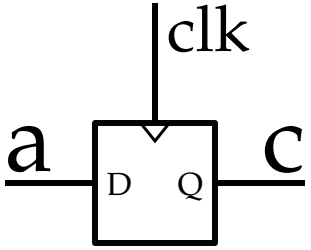
Warmup

Which circuit will be created by the ~~VHDL~~ SV below?

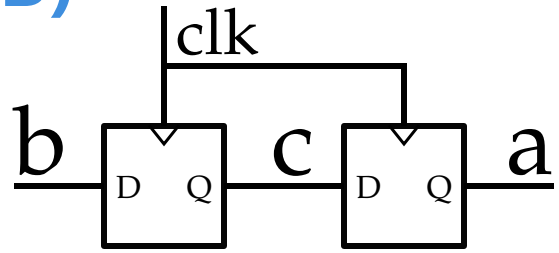
always_ff

```
always @(posedge clk)
begin
    c <= b;
    b <= a;
end
```

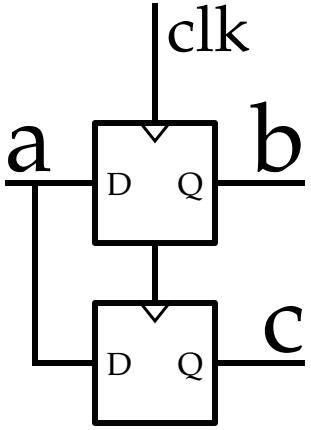
(A)



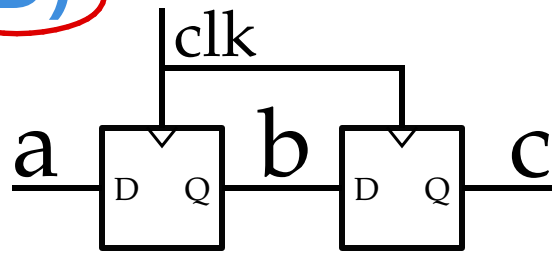
(B)



(C)



(D)

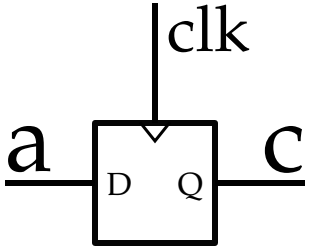


Warmup

Which circuit will be created by the VHDL below?

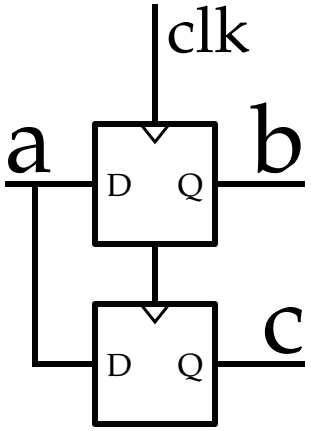
```
always @(posedge clk)
begin
    b <= a;
    c <= b;
end
```

(A)

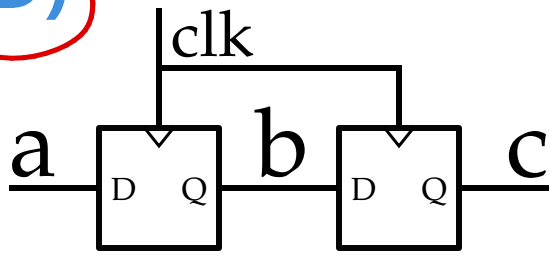


(B)

(C)



(D)



Warmup

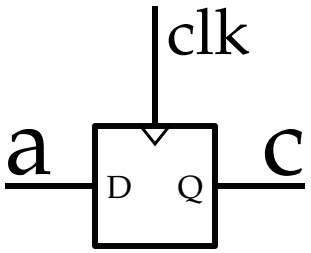
Which circuit will be created by the ~~VHDL~~ SV below?

always_ff

```
always @(posedge clk)
begin
    b = a;
    c = b;
end
```

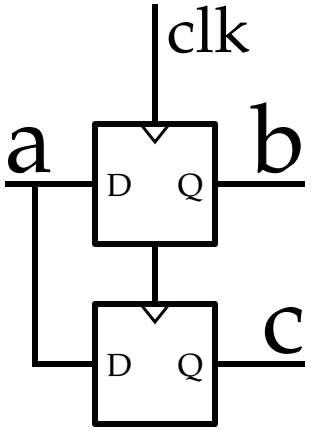
Better to use non-blocking!

(A)

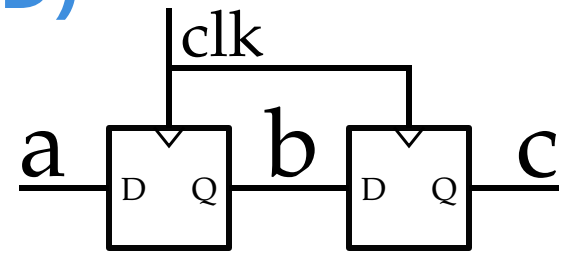


(B)

(C)



(D)



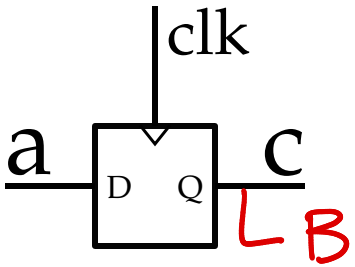
Warmup

Which circuit will be created by the ~~VHDL~~ SV below?

```
assign c = b;
always @(posedge clk)
begin
    b <= a;
end
```

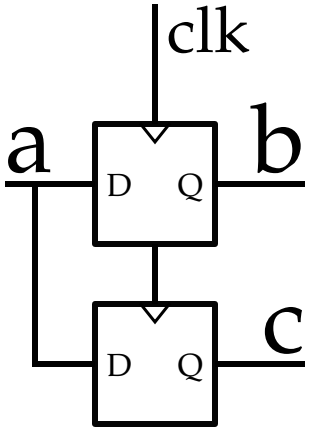
b <= a;
c <= a;

(A)



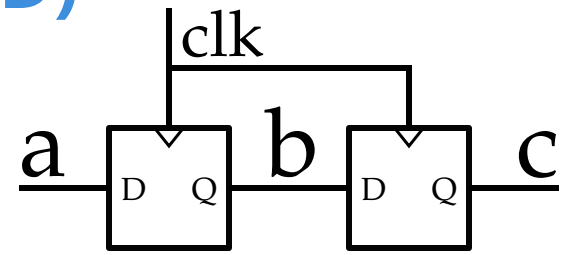
(B)

(C)



same behavior

(D)



EE 201: State machines, part 1

Steven Bell

29 February 2024

Tufts
UNIVERSITY

Objectives

- Explain what a state machine is
- Explain the difference between a Mealy and Moore state machine
- Given an English description of a system:
  - Draw the state diagram
  - Determine how many bits of state are necessary, and choose state encodings
  - Write logic equations for the state transitions and outputs
  - Draw the complete logic circuit diagram for the FSM

What good are state machines?

State machines are a *way of thinking* about digital logic problems

State machines let us build circuits that can perform sequences of actions and make decisions.

Complex behavior can be decomposed into discrete states, and possible actions in each state.

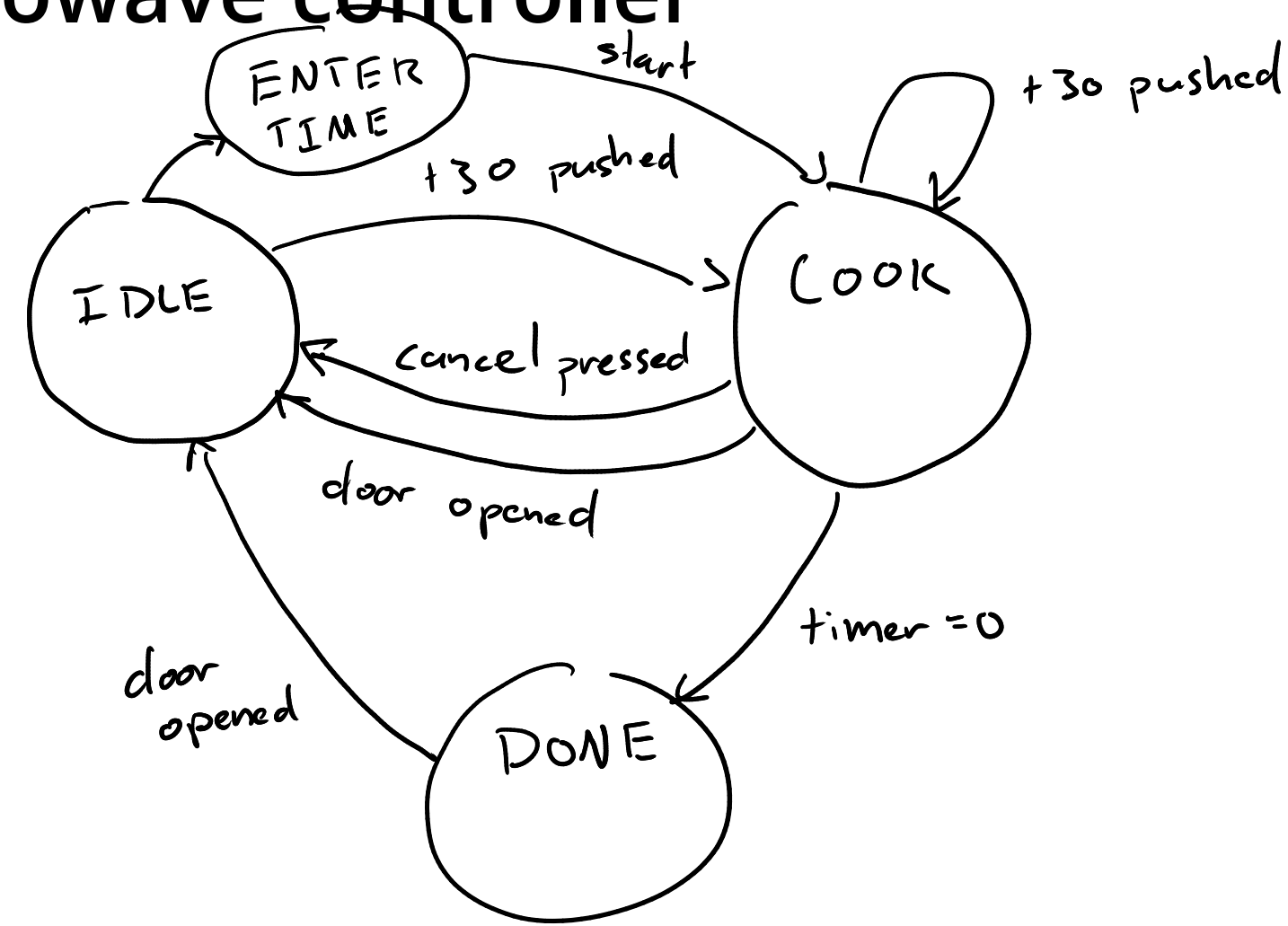
Where do we use FSMs?

Toy examples to torment digital logic students

Creating decision-making or sequential behavior

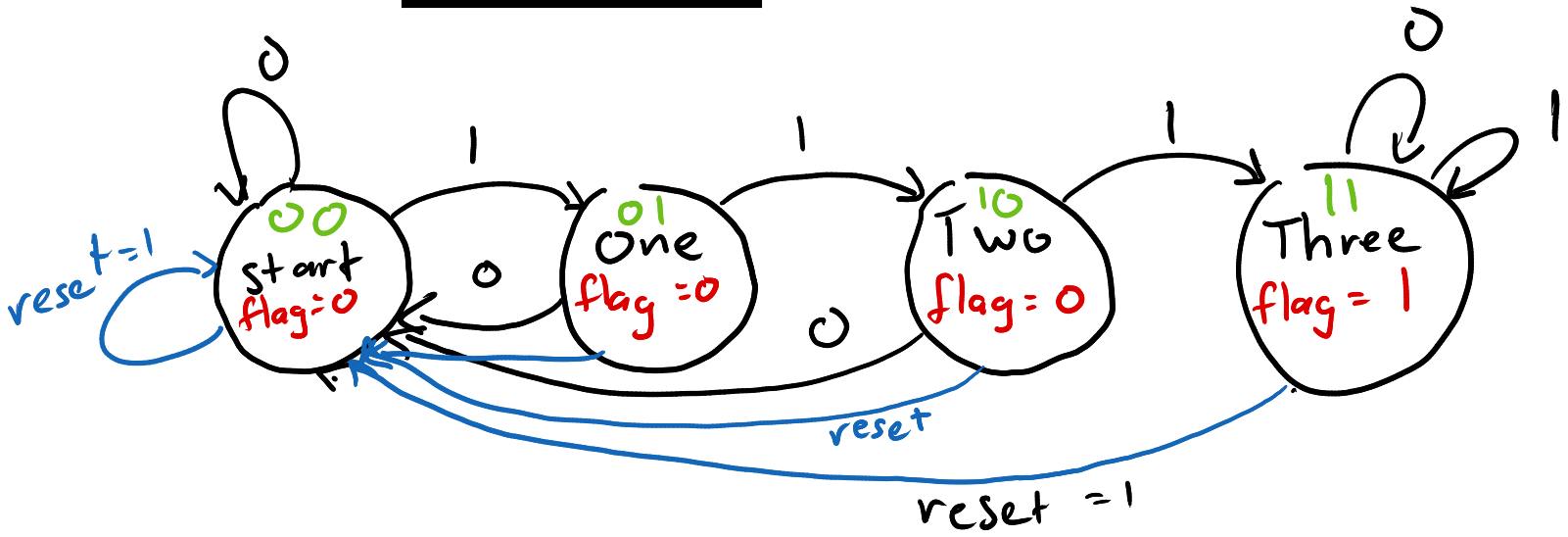
Detecting patterns (e.g., DFAs for parsing)

A microwave controller



Defining the problem

Design a circuit which sets a flag high when the "data" input has been high for 3 clock cycles in a row. The flag should stay high until the reset signal is asserted (i.e., = 1)



Side note: Graphviz is awesome

Graphviz provides a language for describing graphs (like FSMs) and can draw them automatically.

Experiment at edotor.net

It's also installed on the Halligan servers (run `dot --help`)

```
dot -Tpng -oFILENAME INPUTFILE.GV
```

Putting it all together

	S_1	S_0	reset	data	Next state	
start	0	0	0	0	0	0
	0	0	0	1	0	1
	0	0	1	0	0	0
	0	0	1	1	0	0
one	0	1	0	0	0	0
	0	1	0	1	1	0
	0	1	1	0	0	0
	0	1	1	1	0	0
two	1	0	0	0	0	0
	1	0	0	1	1	1
	1	0	1	0	0	0
	1	0	1	1	0	0
three	1	1	0	0	1	1
	1	1	0	1	1	1
	1	1	1	0	0	0
	1	1	1	1	0	0

S_0:

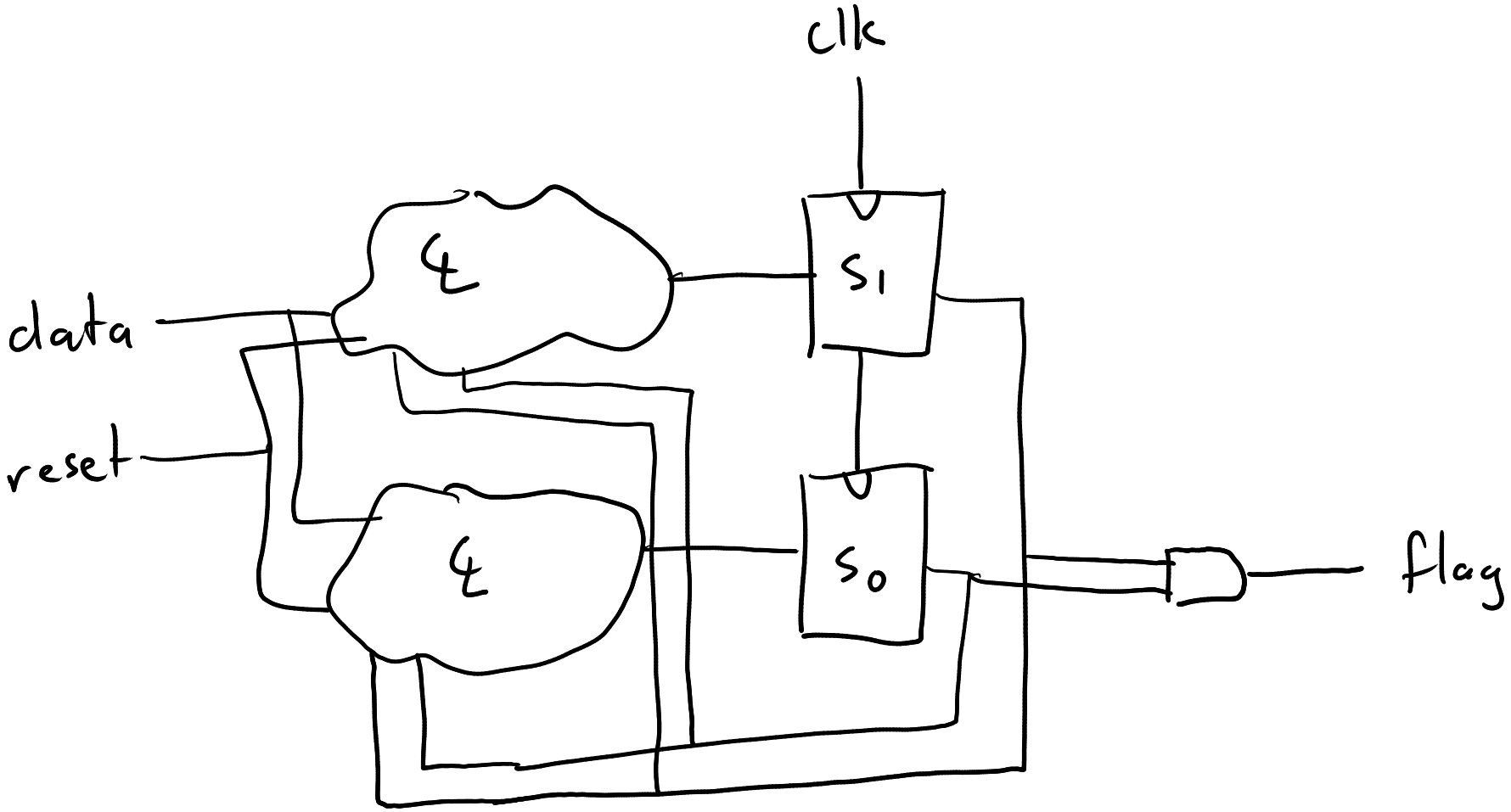
reset, data \ $S_1 S_0$	00	01	11	10
00	0	0	1	0
01	1	0	1	1
11	0	0	0	0
10	0	0	0	0

$\overline{reset}\ data\ \overline{S_0} + \overline{reset}\ S_1 S_0$

S_1	S_0	flag
0	0	0
0	1	0
1	0	0
1	1	1

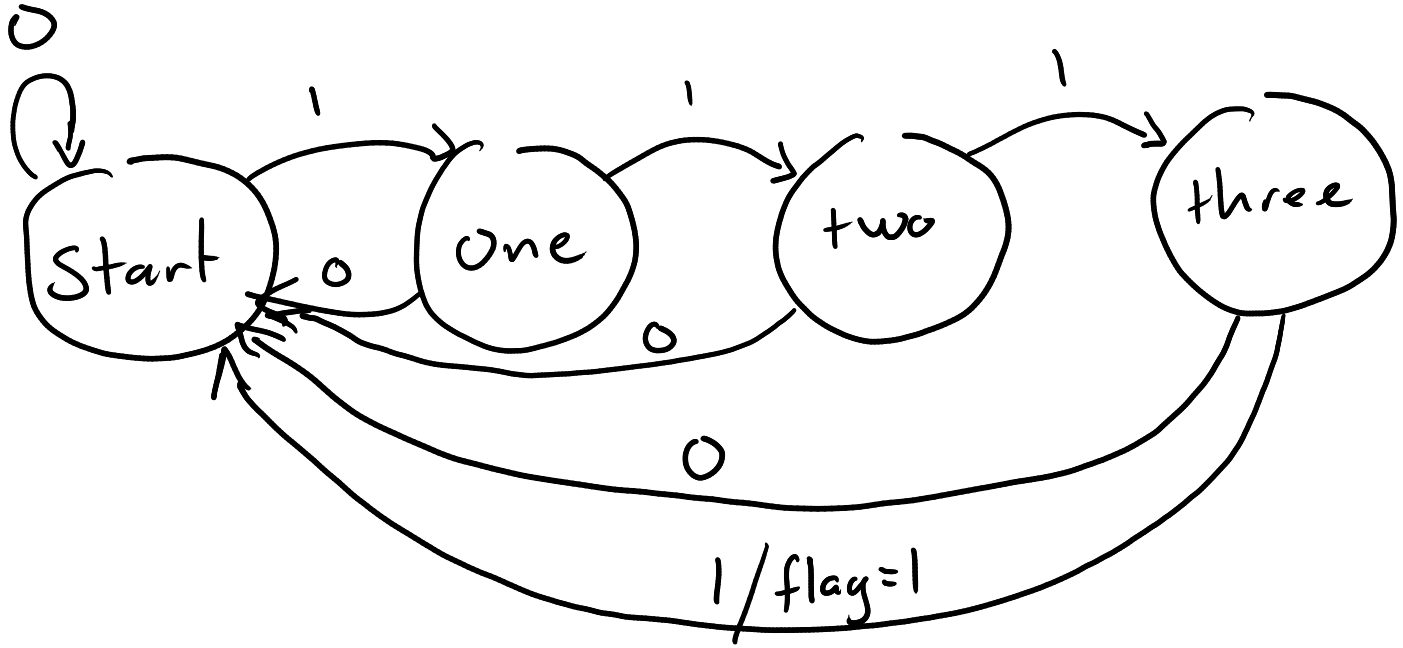
$flag = S_1 S_0$

Putting it all together



Moore vs Mealy

Tweak our previous example: detect four 1s in sequence, and go back to the reset state automatically.



Moore vs Mealy

Book example: Alyssa's robot snail smiles when it detects "01"

What else is a state machine?

Every sequential circuit can be considered a state machine.
Sometimes this is helpful, sometimes not.

For next time

1. No new reading!

2. Paper HW #2 posted, due Thursday 3/7