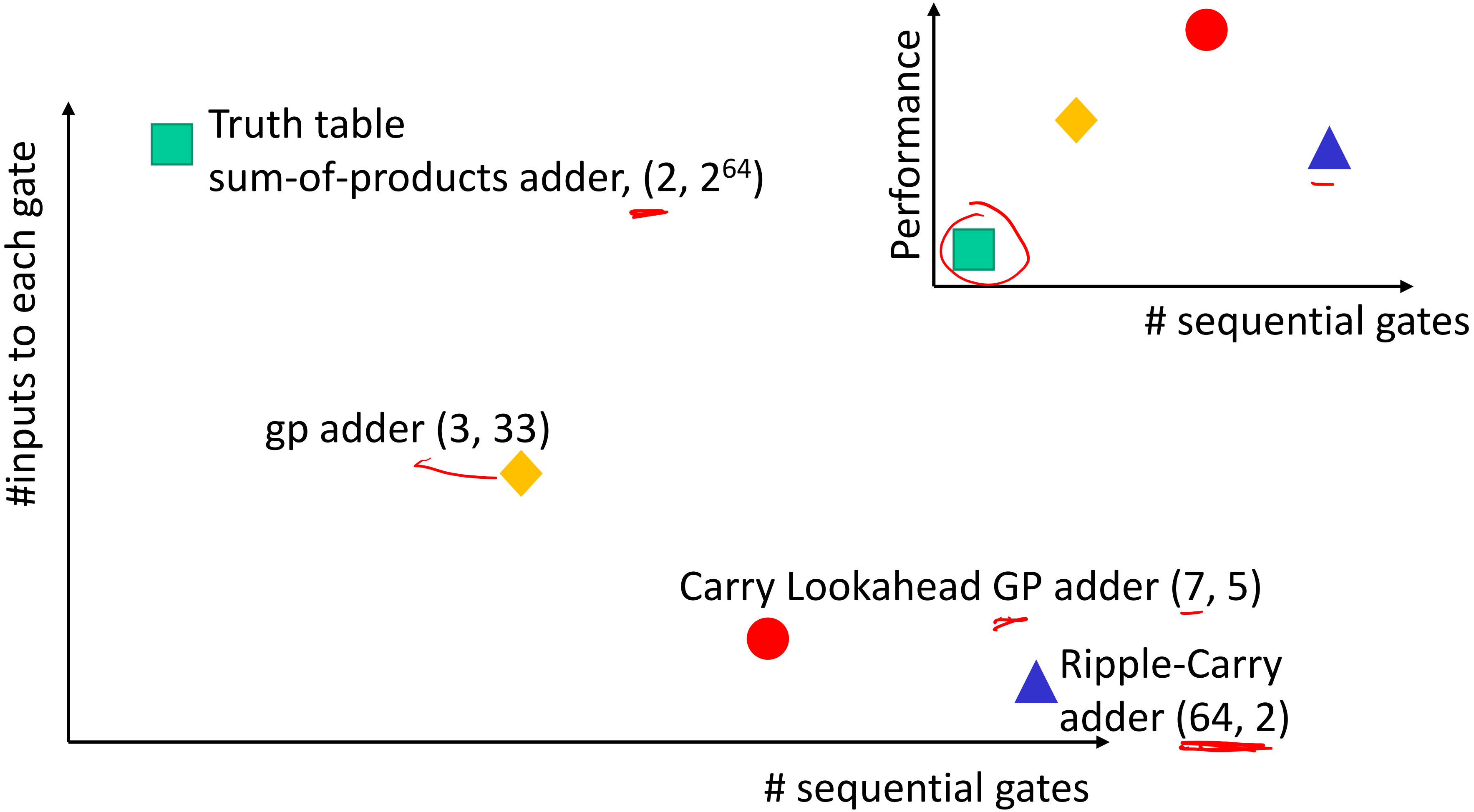
Lecture 14: Sequential Circuits, FSM

.align 2

- Today's topics:
    - Sequential circuits
    - Finite state machines

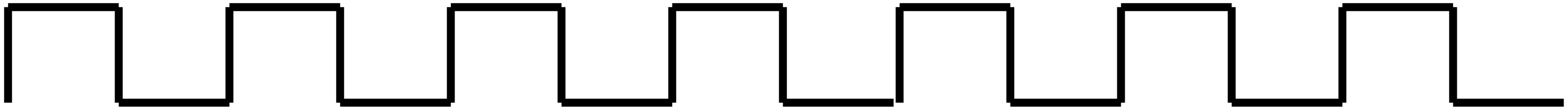
Trade-Off Curve



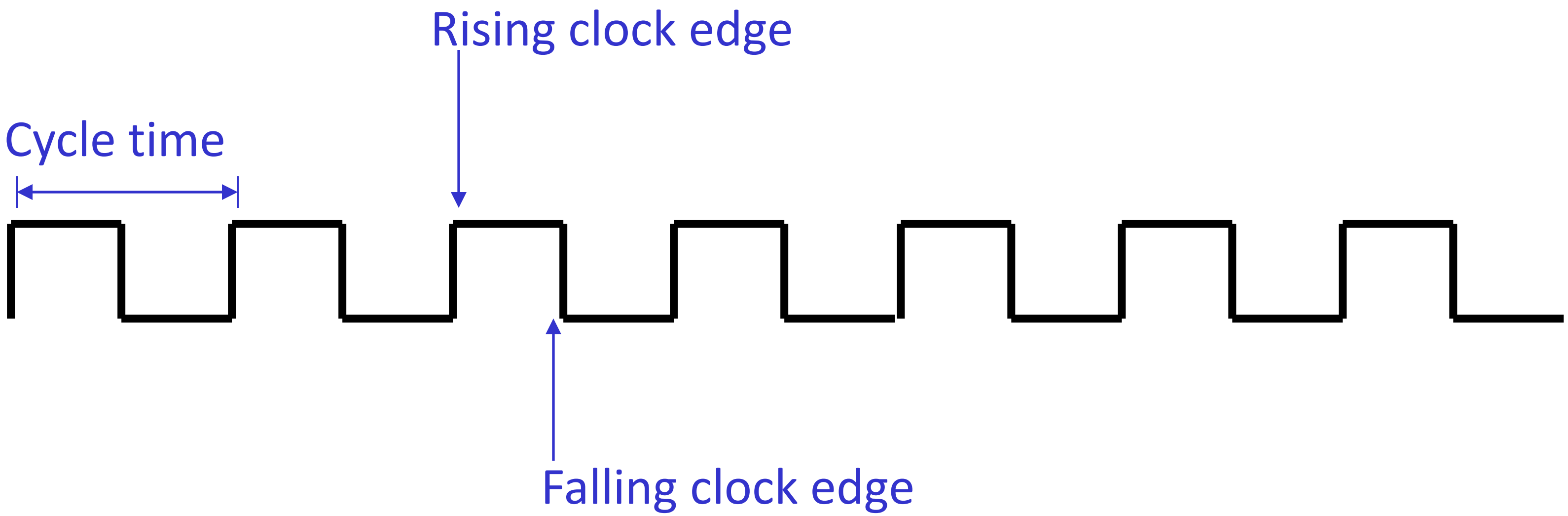
Comb   Seq (have state)

Clocks

- A microprocessor is composed of many different circuits that are operating simultaneously – if each circuit X takes in inputs at time TI_X, takes time TE_X to execute the logic, and produces outputs at time TO_X, imagine the complications in co-ordinating the tasks of every circuit
- A major school of thought (used in most processors built today): all circuits on the chip share a clock signal (a square wave) that tells every circuit when to accept inputs, how much time they have to execute the logic, and when they must produce outputs



Clock Terminology

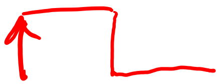

Cycle time

Rising clock edge

Falling clock edge

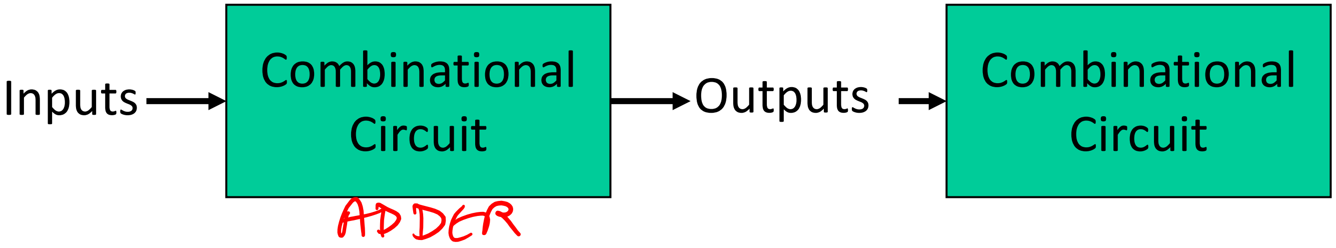
$$4\text{ GHz} = \text{clock speed} = \frac{1}{\text{cycle time}} = \frac{1}{250\text{ ps}}.$$

Sequential Circuits

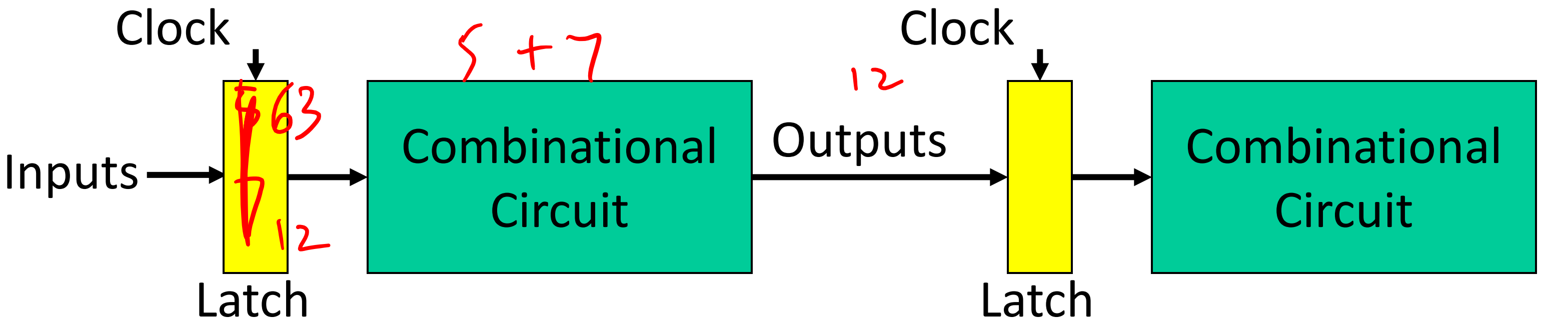


5+7

- Until now, circuits were combinational – when inputs change, the outputs change after a while (time = logic delay thru circuit)



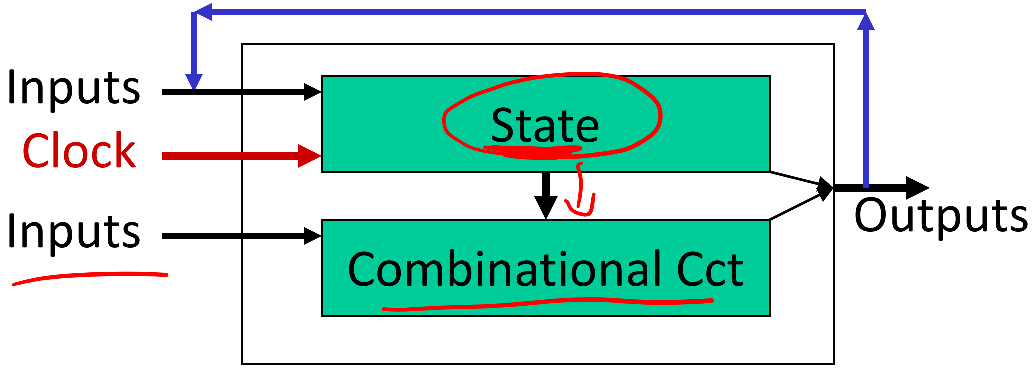
- We want the clock to act like a start and stop signal – a “latch” is a storage device that separates these circuits – it ensures that the inputs to the circuit do not change during a clock cycle



Sequential Circuits

- Sequential circuit: consists of combinational circuit and a storage element

- At the start of the clock cycle, the rising edge causes the "state" storage to store some input values



- This state will not change for an entire cycle (until next rising edge)

- The combinational circuit has some time to accept the value of "state" and "inputs" and produce "outputs"

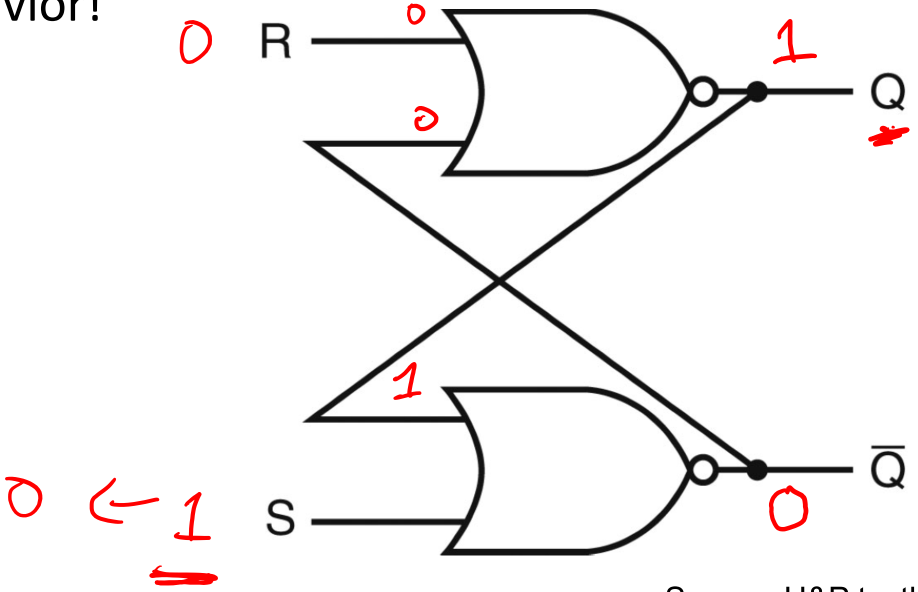
- Some of the outputs (for example, the value of next "state") may feed back (but through the latch so they're only seen in the next cycle)

Designing a Latch

1 0 ω

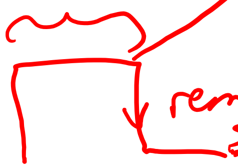
- An S-R latch: set-reset latch
  - When Set is high, a 1 is stored
  - When Reset is high, a 0 is stored
  - When both are low, the previous state is preserved (hence, known as a storage or memory element)
  - Both are high – this set of inputs is not allowed

Verify the above behavior!



Source: H&P textbook

D Latch

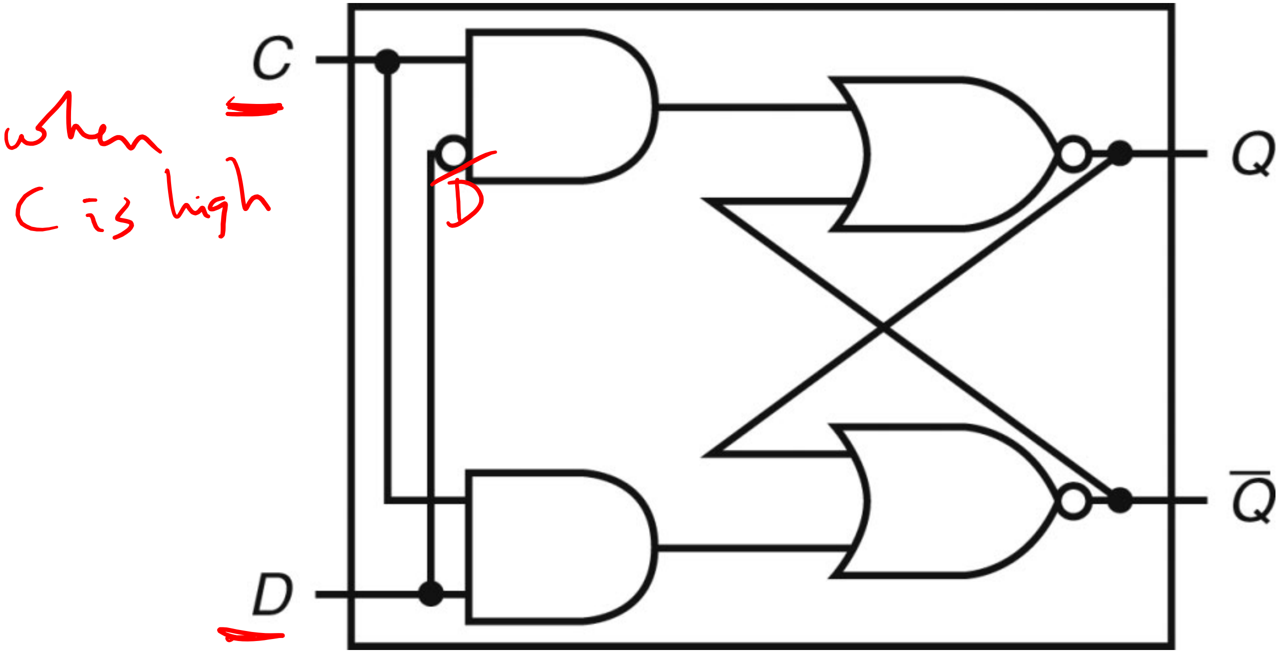


accepting input D → last D is relevant

remember state

when C = 0, I'm in remember state ⇒ R=S=0

- Incorporates a clock
- The value of the input D signal (data) is stored only when the clock is high – the previous state is preserved when the clock is low

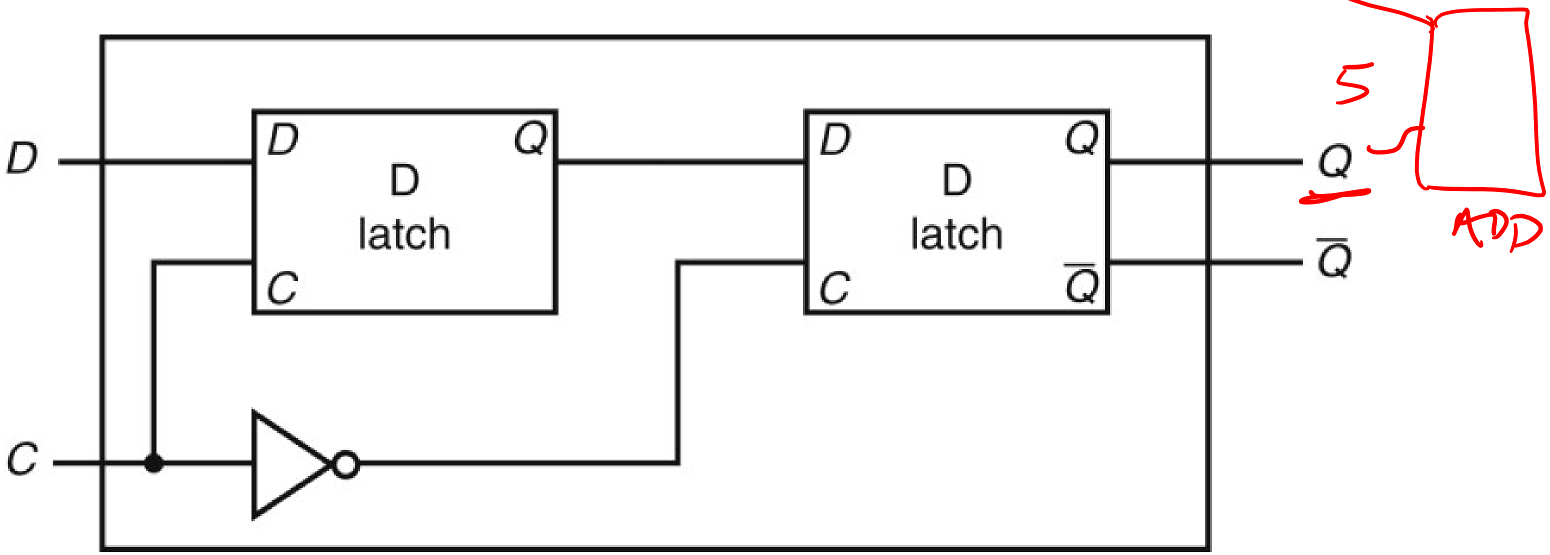


Source: H&P textbook

D Flip Flop

- Terminology:
  Latch: outputs can change any time the clock is high (asserted)
  Flip flop: outputs can change only on a clock edge

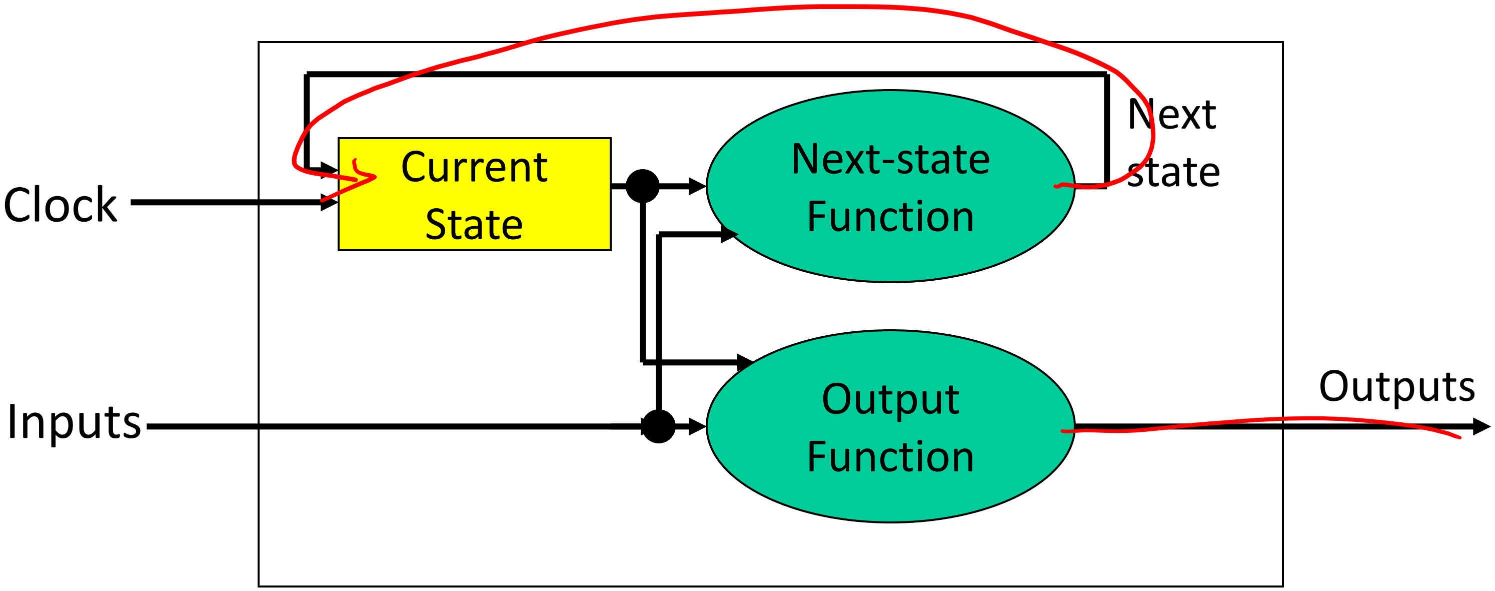
- Two D latches in series – ensures that a value is stored only on the falling edge of the clock



Source: H&P textbook

Finite State Machine

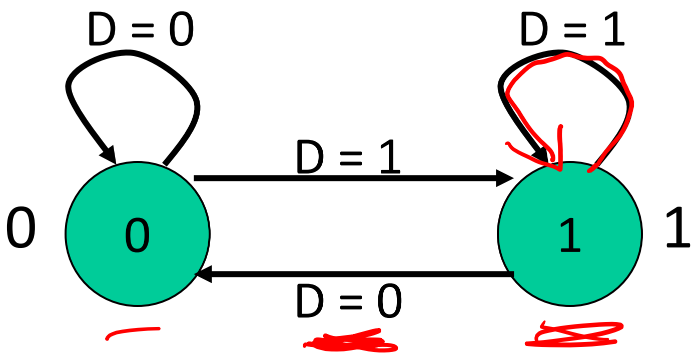
- A sequential circuit is described by a variation of a truth table – a finite state diagram (hence, the circuit is also called a finite state machine)
- Note that state is updated only on a clock edge



State Diagrams

- Each state is shown with a circle, labeled with the state value – the contents of the circle are the outputs

- An arc represents a transition to a different state, with the inputs indicated on the label



This is a state diagram for ___?

3-Bit Counter

- Consider a circuit that stores a number and increments the value on every clock edge – on reaching the largest value, it starts again from 0

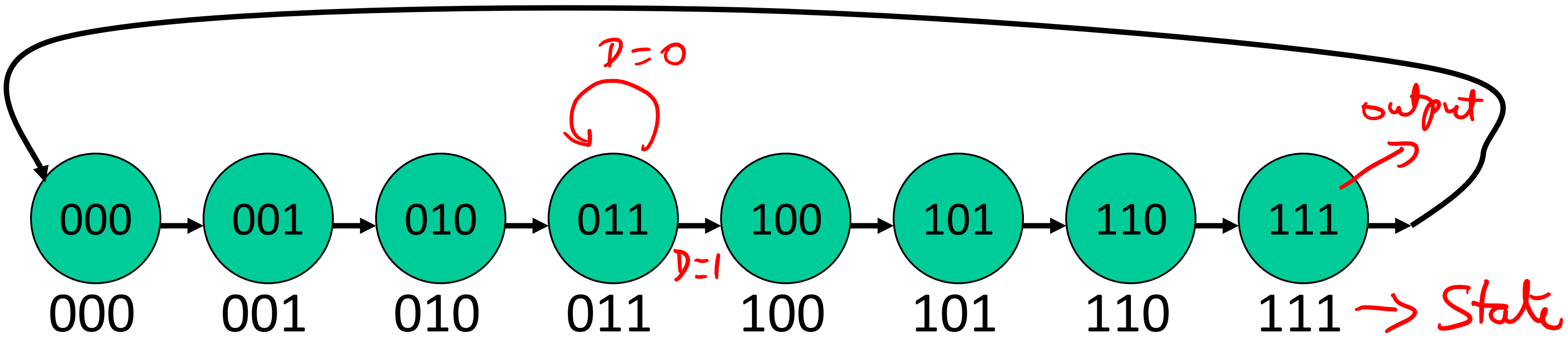
  Draw the state diagram:
  - How many states? 8 (0 - 7)
  - How many inputs?

    no inputs
    (just the clk)

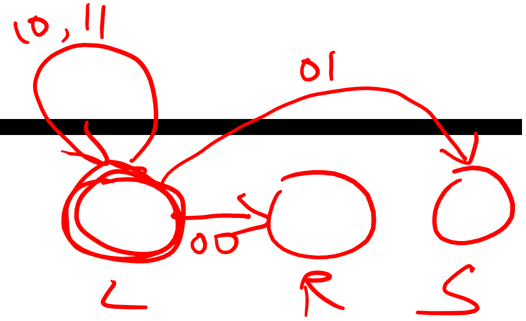
3-Bit Counter

- Consider a circuit that stores a number and increments the value on every clock edge – on reaching the largest value, it starts again from 0

  Draw the state diagram:
  - How many states?
  - How many inputs?



Tackling FSM Problems



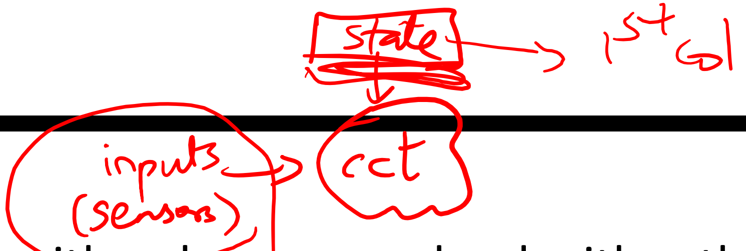
- Three questions worth asking: L R S
  - What are the possible output states? Draw a bubble for each.
  - What are inputs? What values can those inputs take?
  - For each state, what do I do for each possible input value? Draw an arc out of every bubble for every input value.

W < 0, 1

RW < 0, 1

LW < 0, 1

Traffic Light Controller



- Problem description: A traffic light with only green and red; either the North-South road has green or the East-West road has green (both can't be red); there are detectors on the roads to indicate if a car is on the road; the lights are updated every 30 seconds; a light need change only if a car is waiting on the other road

2nd col

State Transition Table:

How many states?

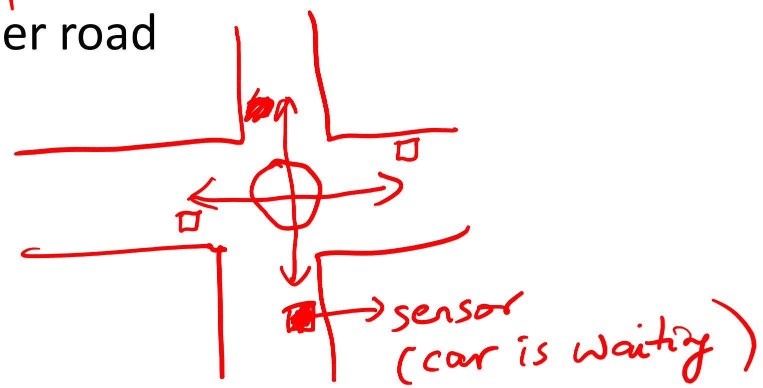
How many inputs?

How many outputs?

NS sensors ← 1, 0

EW sensors ← 1, 0

3rd col



NS/EW	NS/EW sensors	out

NS Green

EW Green

State Transition Table

- Problem description: A traffic light with only green and red; either the North-South road has green or the East-West road has green (both can’t be red); there are detectors on the roads to indicate if a car is on the road; the lights are updated every 30 seconds; a light must change only if a car is waiting on the other road

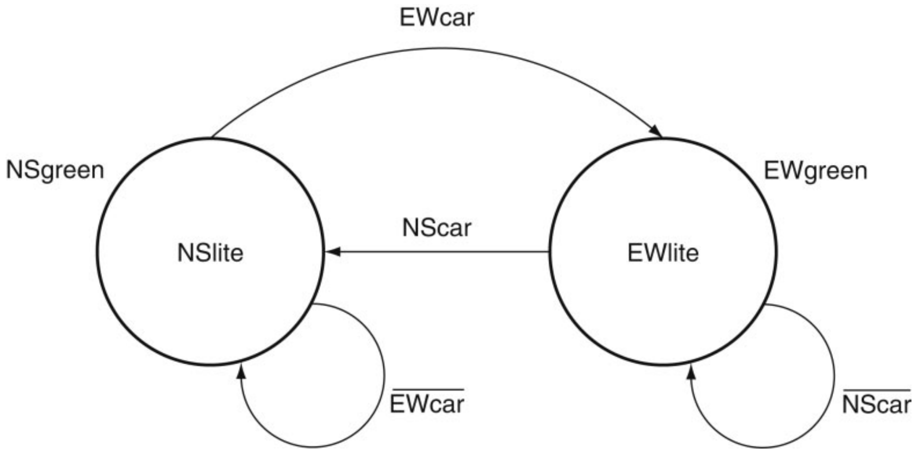
State Transition Table:

CurrState	InputEW	InputNS	NextState=Output
N	0	0	N
N	0	1	N
N	1	0	E
N	1	1	E
E	0	0	E
E	0	1	N
E	1	0	E
E	1	1	N

State Diagram

State Transition Table:

CurrState	InputEW	InputNS	NextState=Output
N	0	0	N
N	0	1	N
N	1	0	E
N	1	1	E
E	0	0	E
E	0	1	N
E	1	0	E
E	1	1	N



Source: H&P textbook

Tackling FSM Problems

- Three questions worth asking:
    - What are the possible output states? Draw a bubble for each.
    - What are inputs? What values can those inputs take?
    - For each state, what do I do for each possible input value? Draw an arc out of every bubble for every input value.

Example – Residential Thermostat

Outputs : AC Ht Off

- Two temp sensors: internal and external
- If internal temp is within 1 degree of desired, don’t change setting
- If internal temp is > 1 degree higher than desired, turn AC on; if internal temp is < 1 degree lower than desired, turn heater on
- If external temp and desired temp are within 5 degrees, disregard the internal temp, and turn both AC and heater off

Inputs : Ext T Int T

2 values (desrange) (outofrange)

69-71 > 71 < 69

3 values

3 values

< 65 65-75 > 75

3 output/input states

18 rows $= 3 \times 3 \times 2$

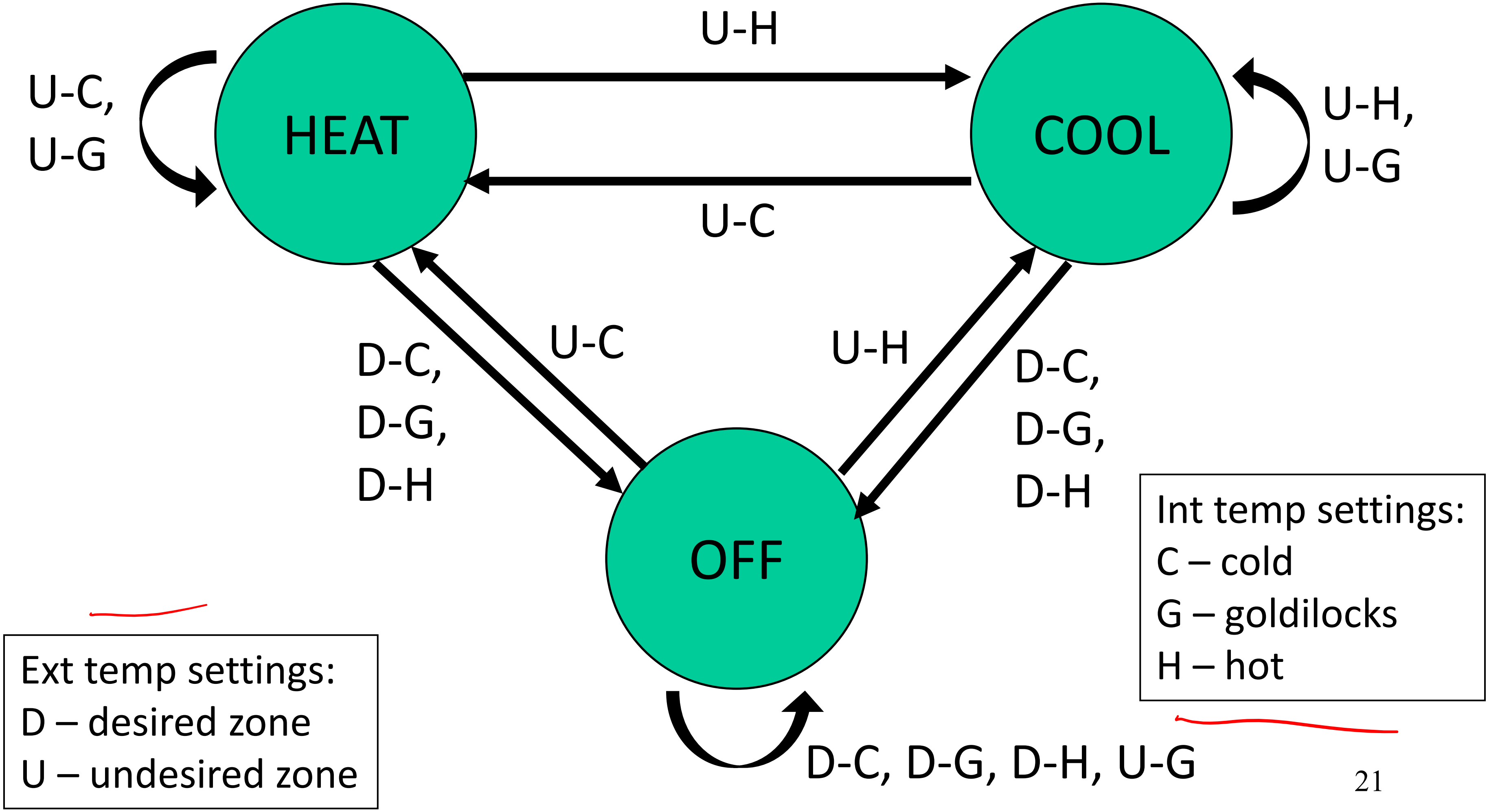
3 values for Int T

2 values for Ext T

Finite State Machine Table

Current State	Input E	Input I	Output State
HEAT	D	C	OFF
HEAT	D	G — Goldilocks	OFF
HEAT	D → Desired	H — Too hot	OFF
HEAT	U → Under	C — Too cold	HEAT
HEAT	U	G	HEAT
HEAT	U	H	COOL
COOL	D	C	OFF
COOL	D	G	OFF
COOL	D	H	OFF
COOL	U	C	HEAT
COOL	U	G	COOL
COOL	U	H	COOL
OFF	D	C	OFF
OFF	D	G	OFF
OFF	D	H	OFF
OFF	U	C	HEAT
OFF	U	G	OFF
OFF	U	H	COOL

Finite State Diagram



Int temp settings:
C – cold
G – goldilocks
H – hot

Ext temp settings:
D – desired zone
U – undesired zone

Latch vs. Flip-Flop

- Recall that we want a circuit to have stable inputs for an entire cycle – so I want my new inputs to arrive at the start of a cycle and be fixed for an entire cycle

- A flip-flop provides the above semantics (a door that swings open and shut at the start of a cycle)

- But a flip-flop needs two back-to-back D-latches, i.e., more transistors, delay, power

- You can reduce these overheads with just a single D-latch (a door that is open for half a cycle) as long as you can tolerate stable inputs for just half a cycle