Time

Until now: we have essentially ignored the issue of time

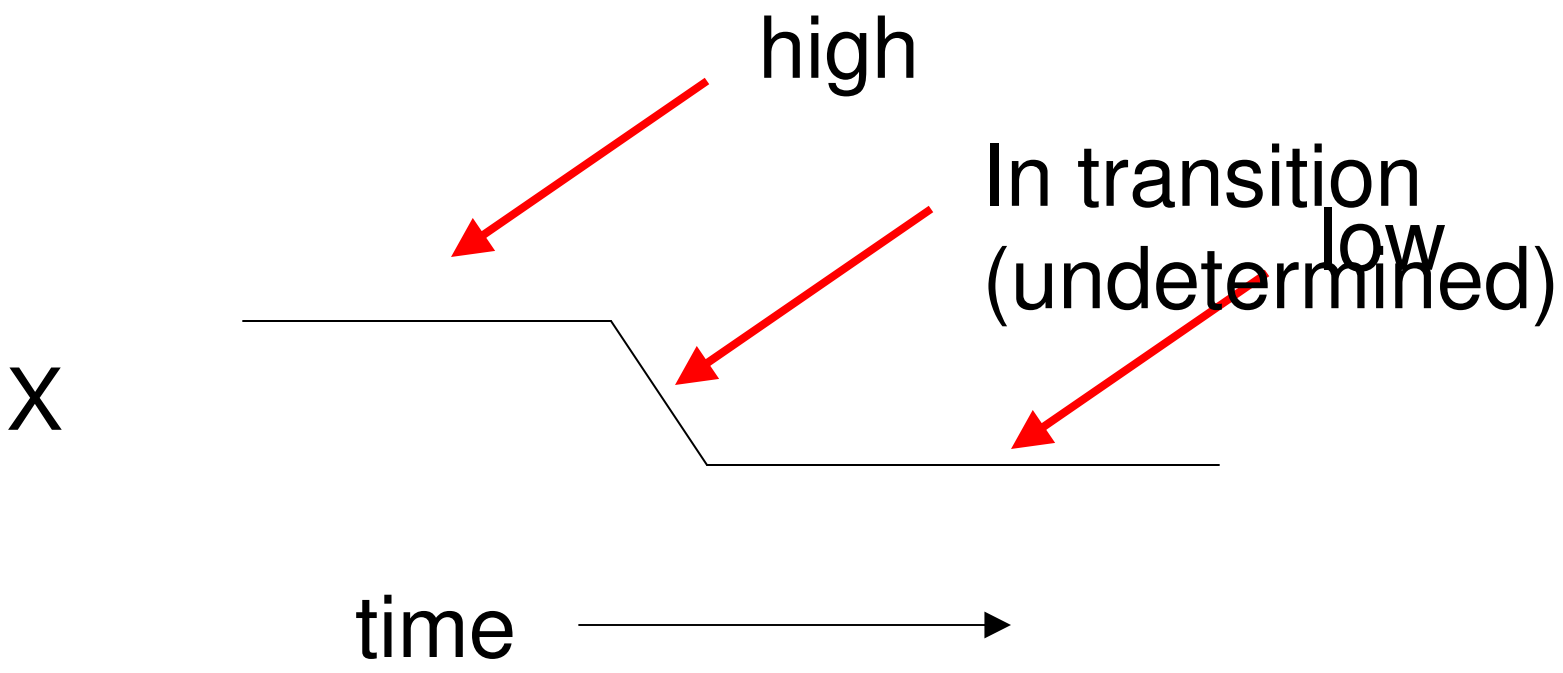
- We have assumed that our digital logic circuits perform their computations instantaneously
- Our digital logic circuits have been "stateless"
  - Once you present a new input, they forget everything about previous inputs
  - We call this type of digital system **combinatorial logic**

Time

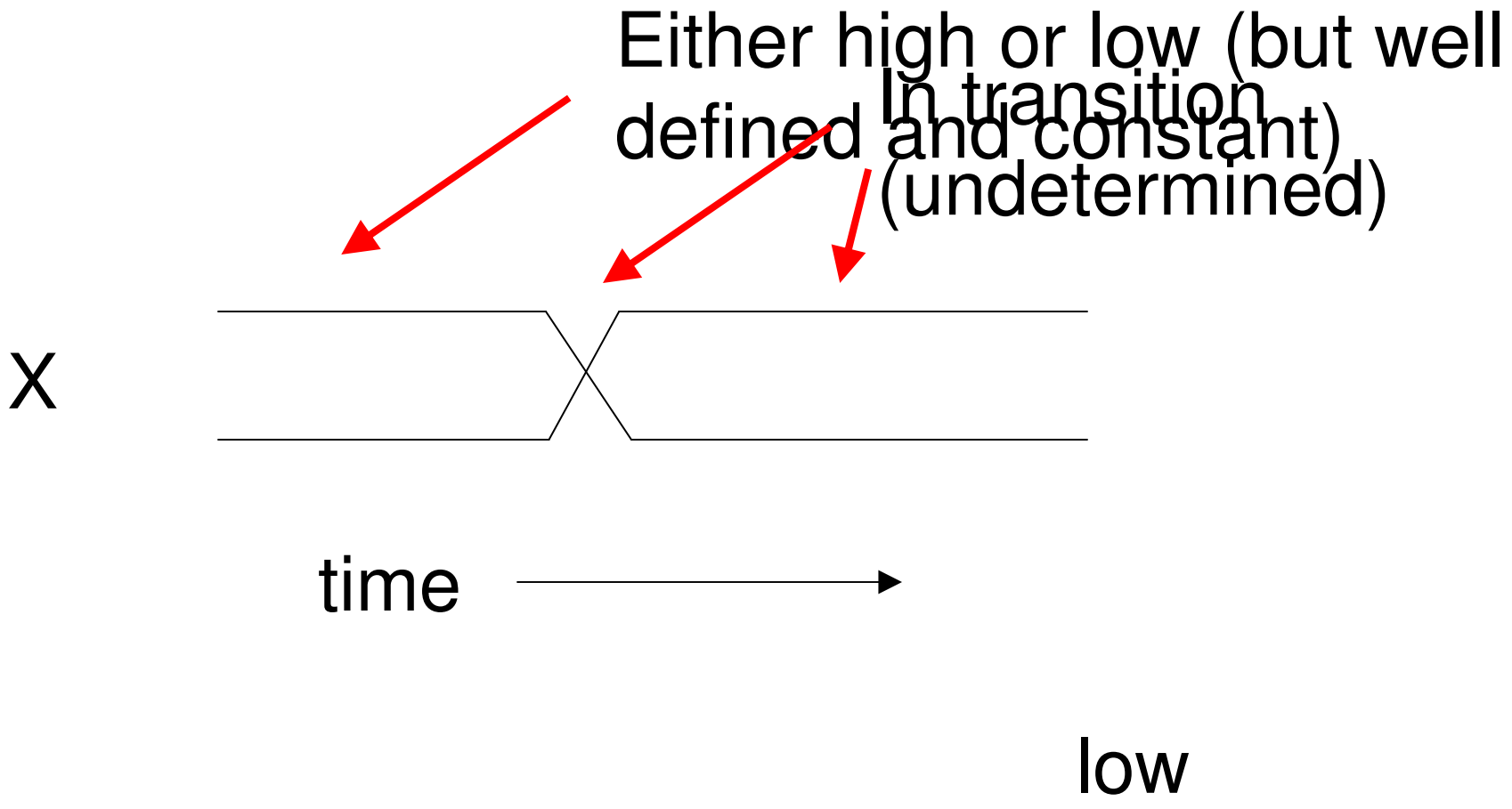
In reality, time is an important issue:

- Even our logic gates induce a small amount of delay (on the order of a few nanoseconds)
- For much of what we do – we actually want our circuits to have some form of memory

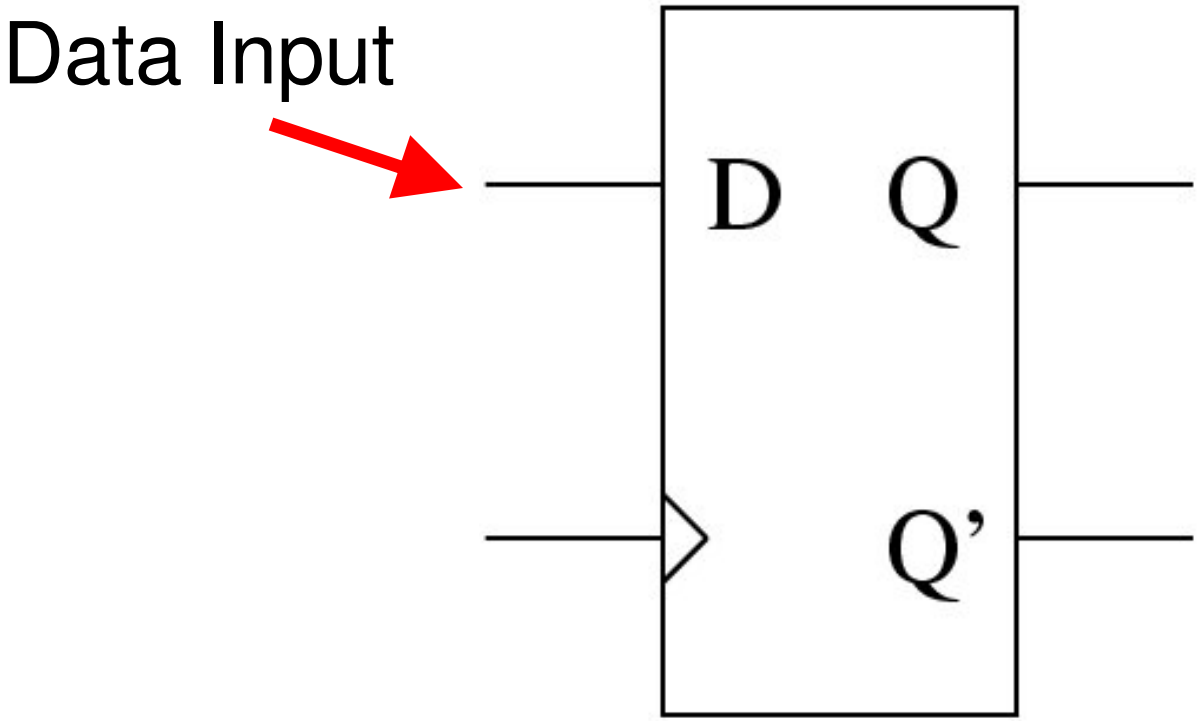
Timing Notation



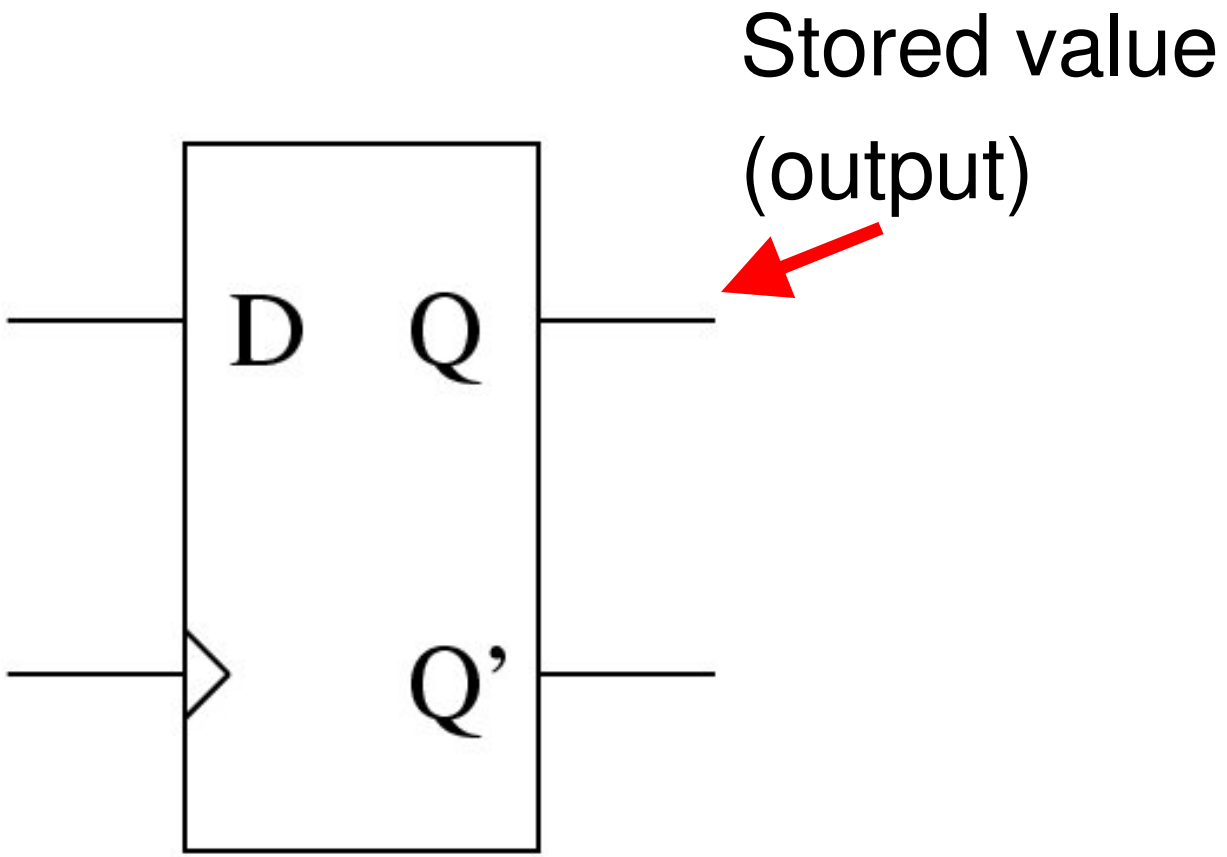
Timing Notation



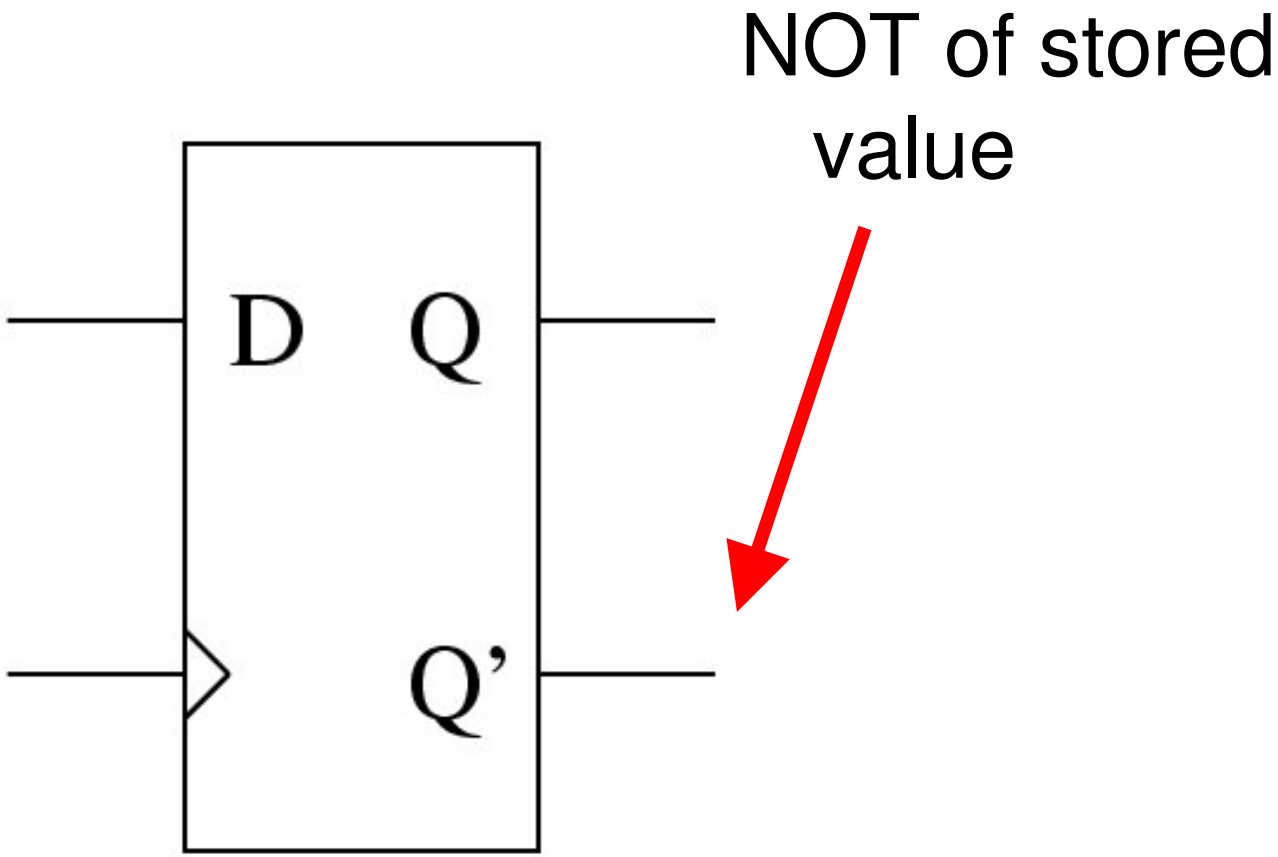
D Flip Flops



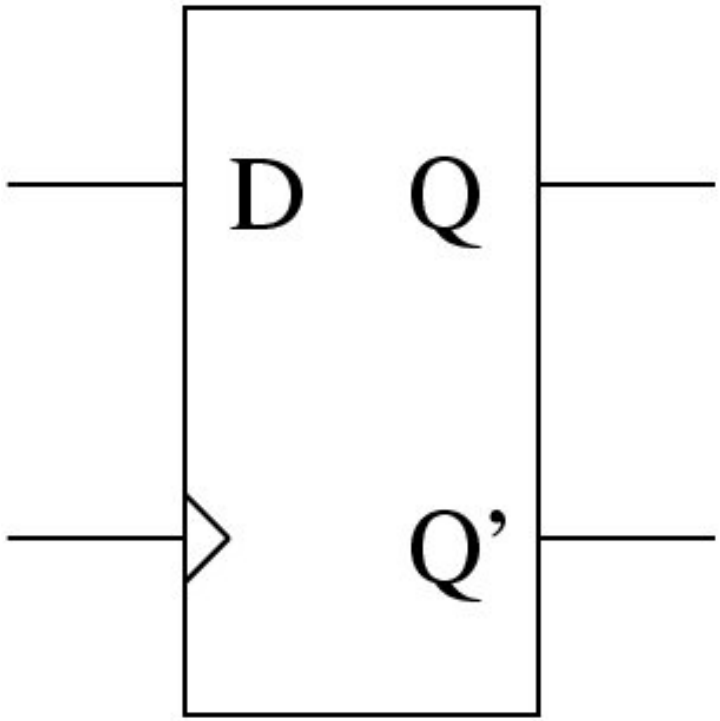
D Flip Flops



D Flip Flops

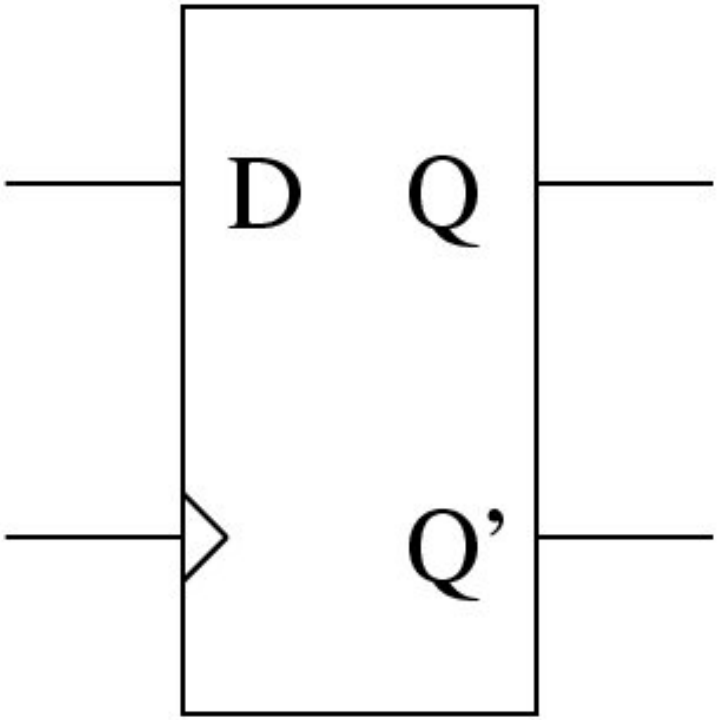


D Flip Flops



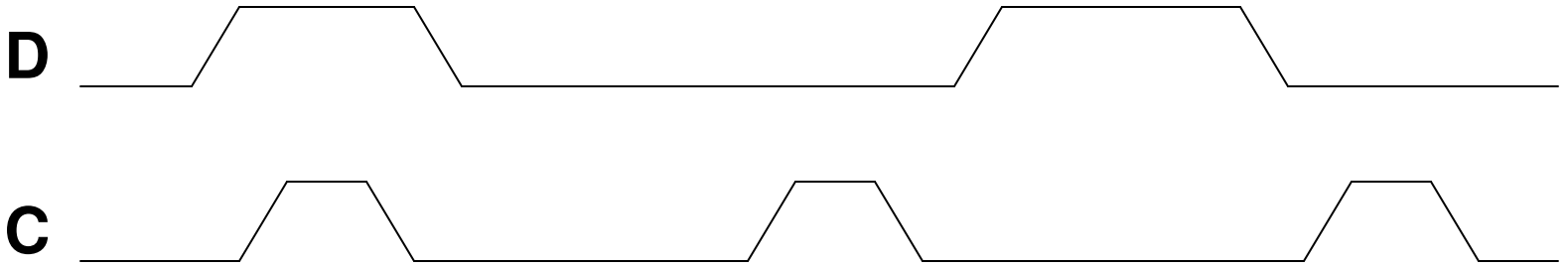
Clock input

D Flip Flops



When the clock transitions from high to low:
the value of D is stored

D Flip Flop

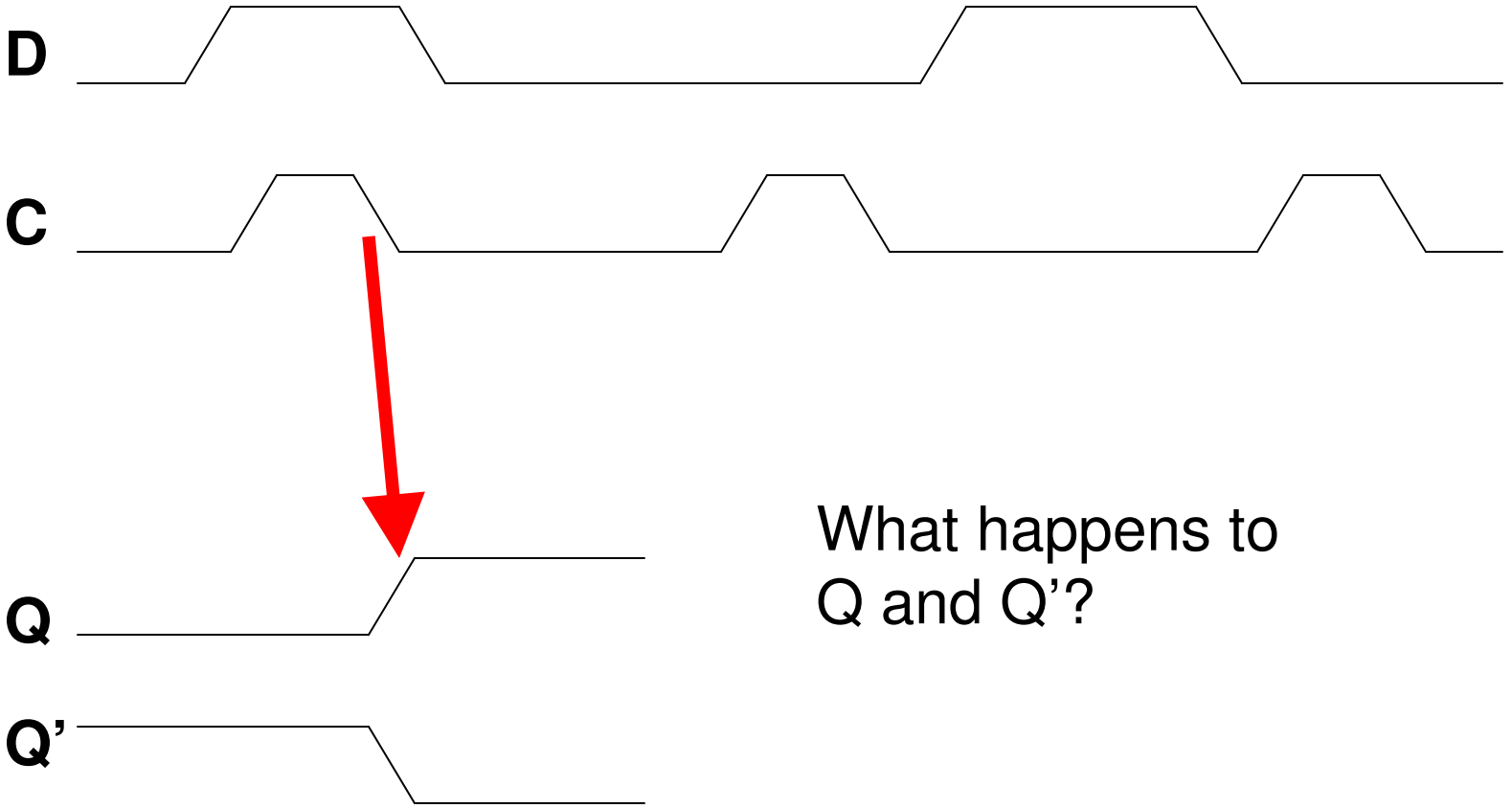


Q

Q'

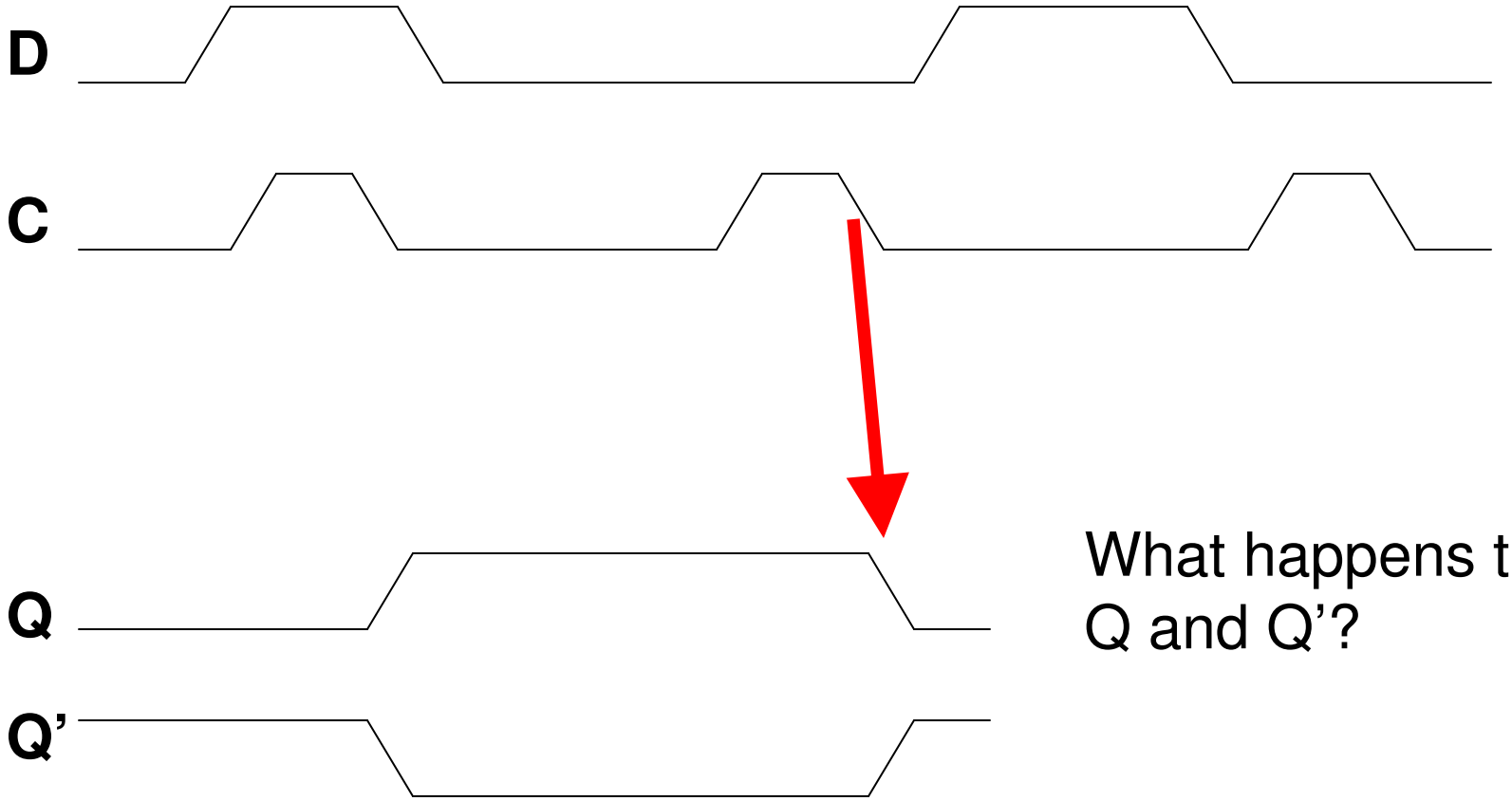
What happens to
Q and Q'?

D Flip Flop



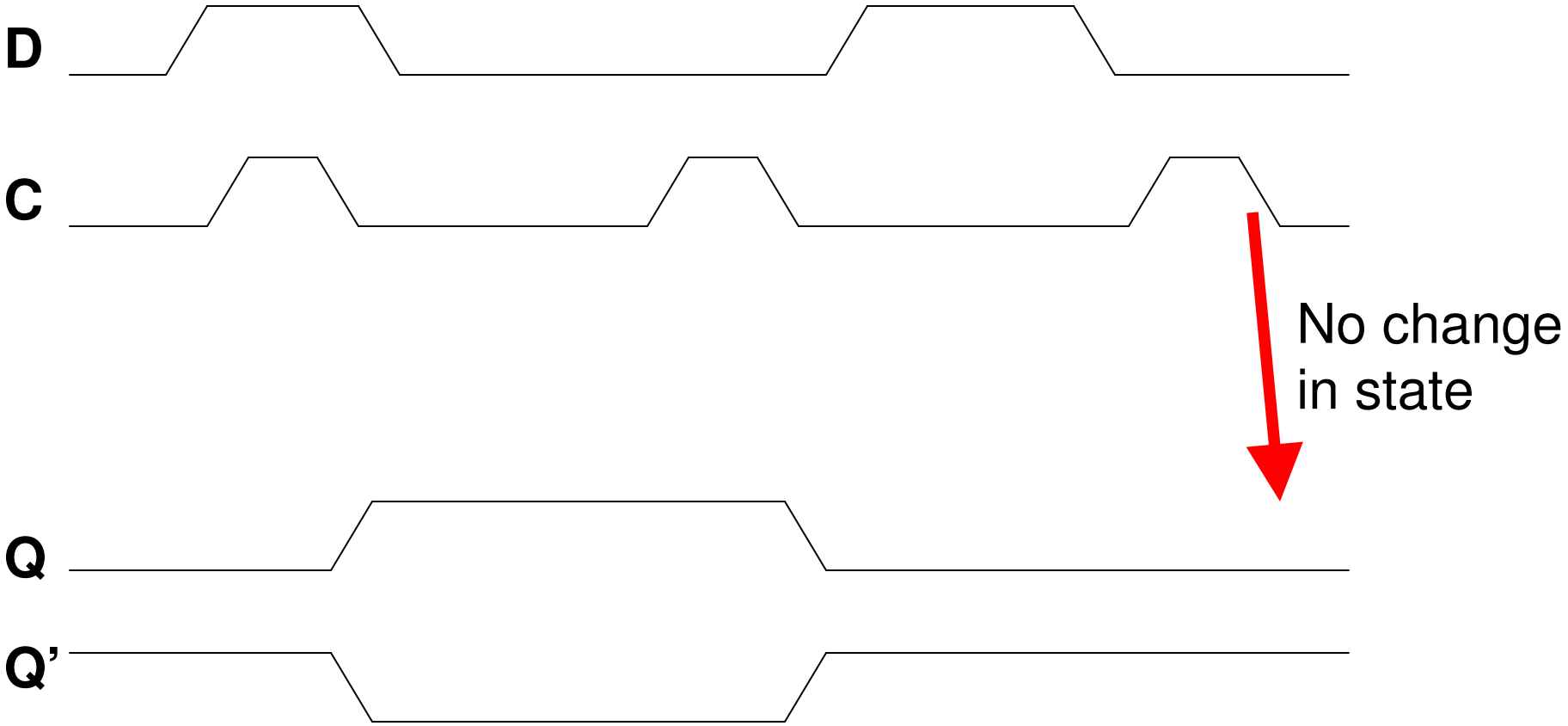
What happens to
Q and Q'?

D Flip Flop



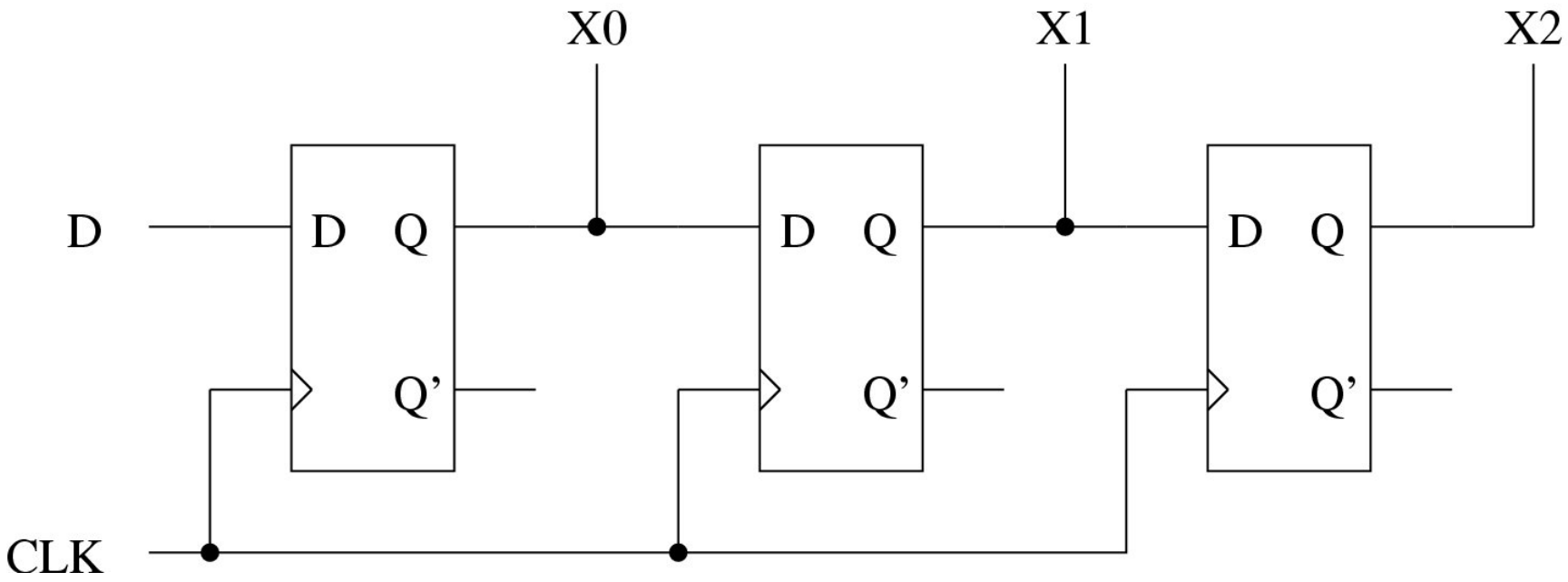
What happens to Q and Q'?

D Flip Flop



An Application of D Flip Flops

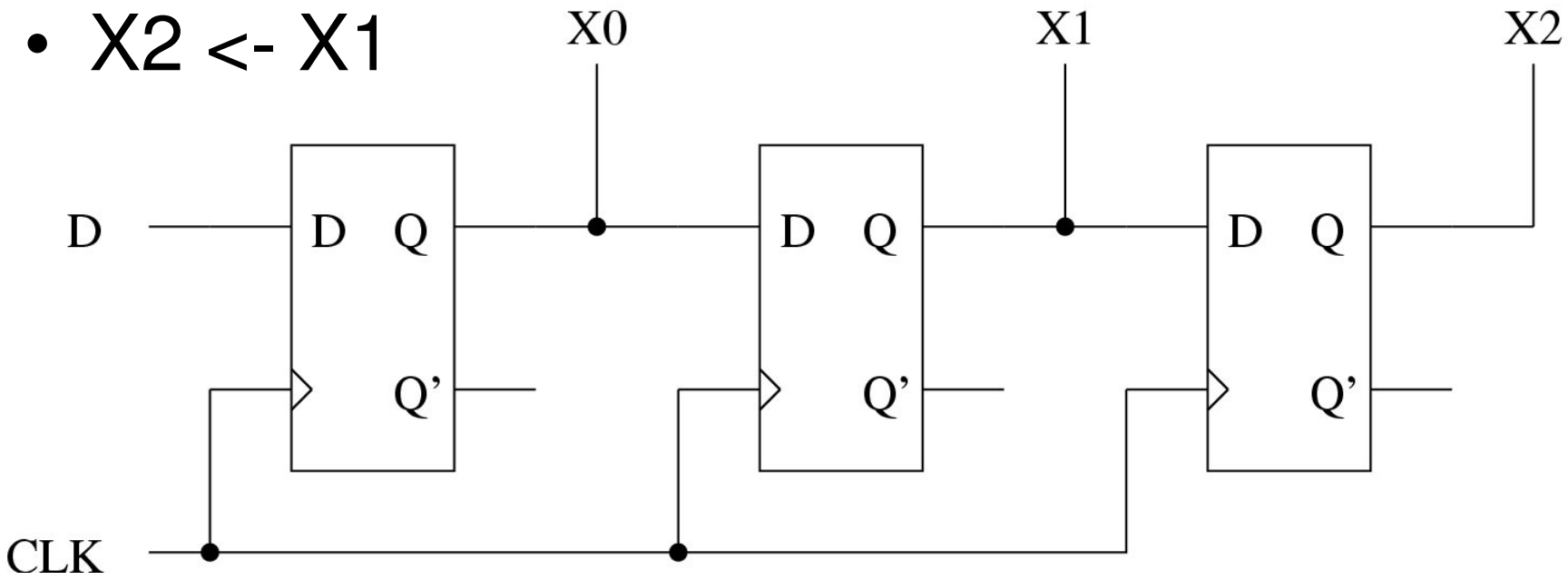
What does this circuit do?



Shift Register

On each clock transition from high to low:

- X0 takes on the current value of D
- X1 <- X0
- X2 <- X1



Last Time

- Boolean Algebra and simplification of circuits
- The need to introduce memory
  - Timing notation
  - D flip flops
  - Shift registers

Today: Sequential Logic

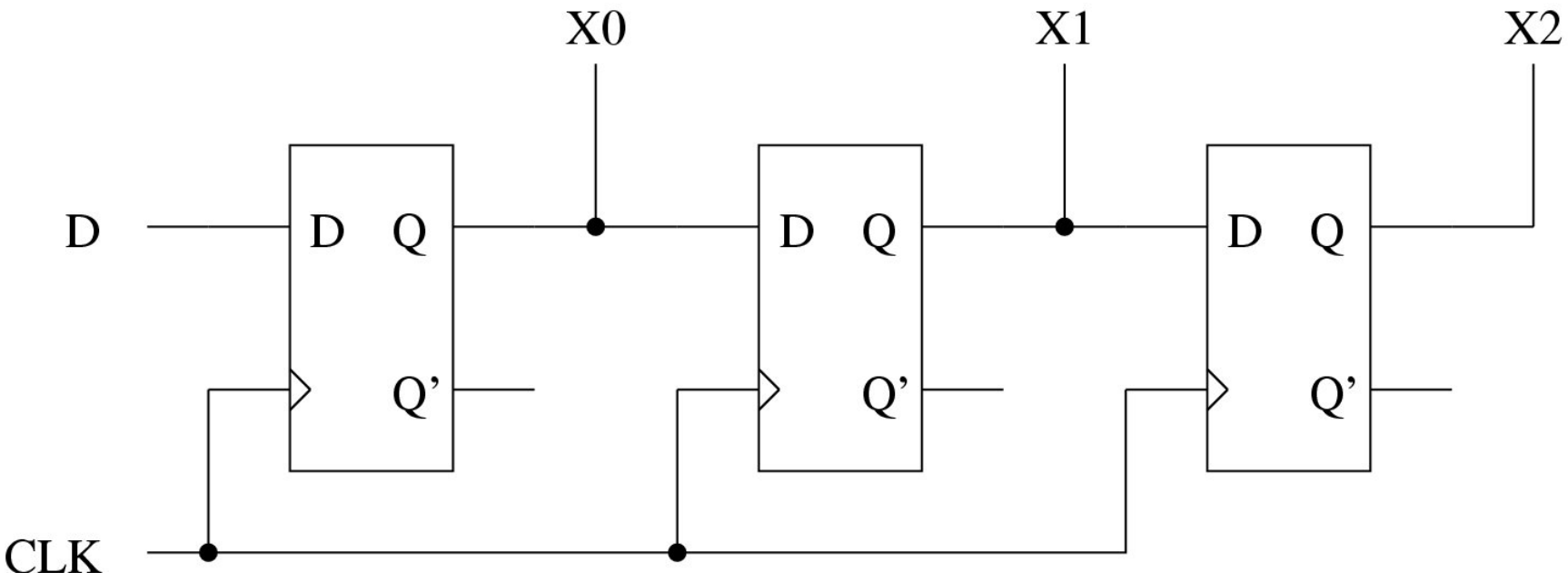
- More on the shift register
- Counting
- Binary numbers
- Sequential logic circuit design

Administrivia

- Homework 1
  - Due today at 5:00
- Homework 2
  - Has been posted
  - Due: February 15th

An Application of D Flip Flops

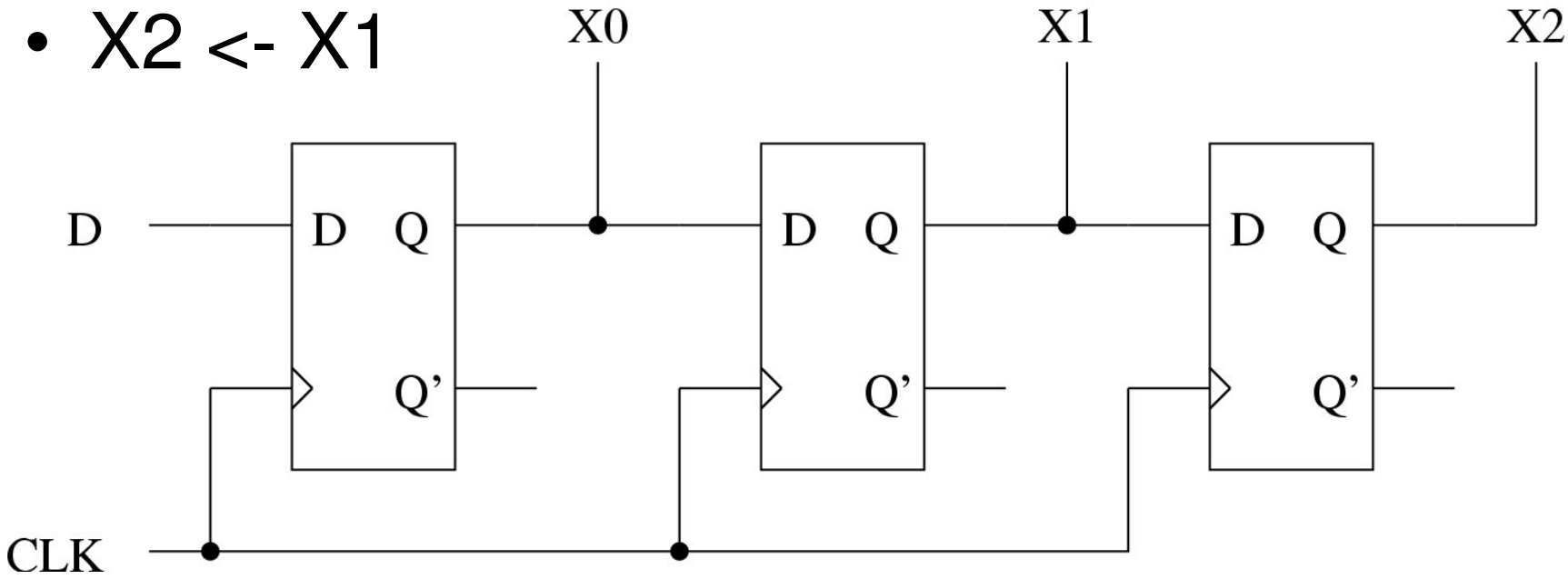
What does this circuit do?



Shift Register

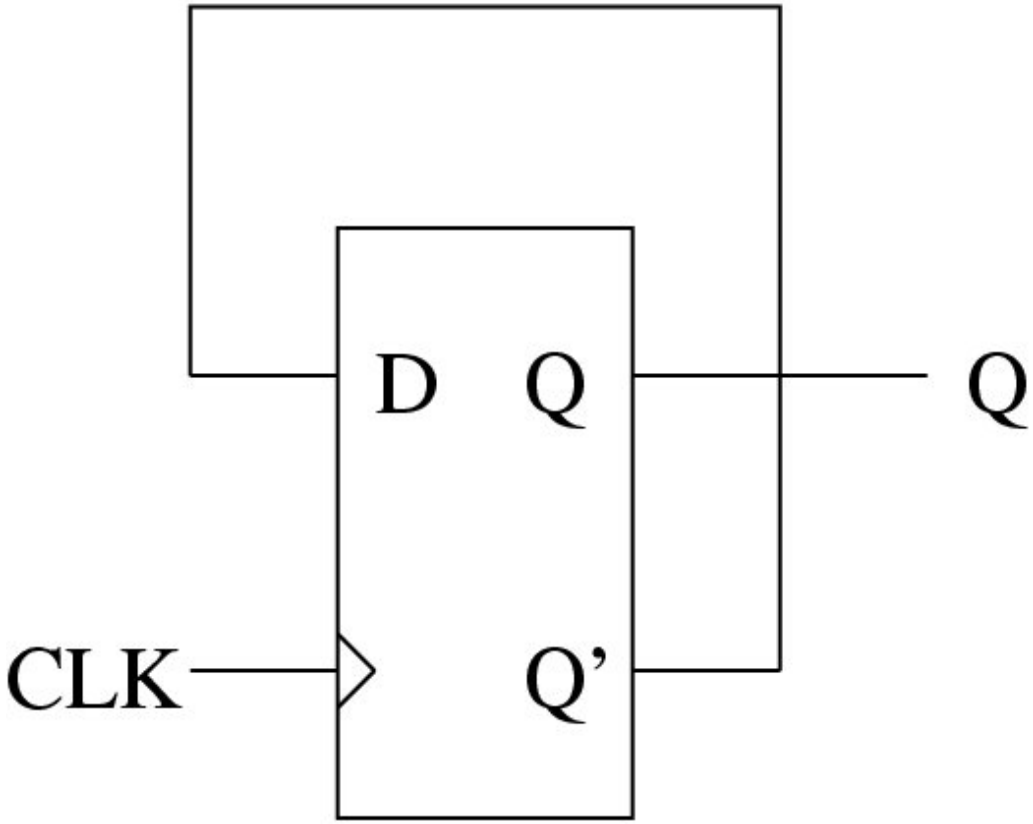
On each clock transition from high to low:

- X0 takes on the current value of D
- X1 <- X0
- X2 <- X1



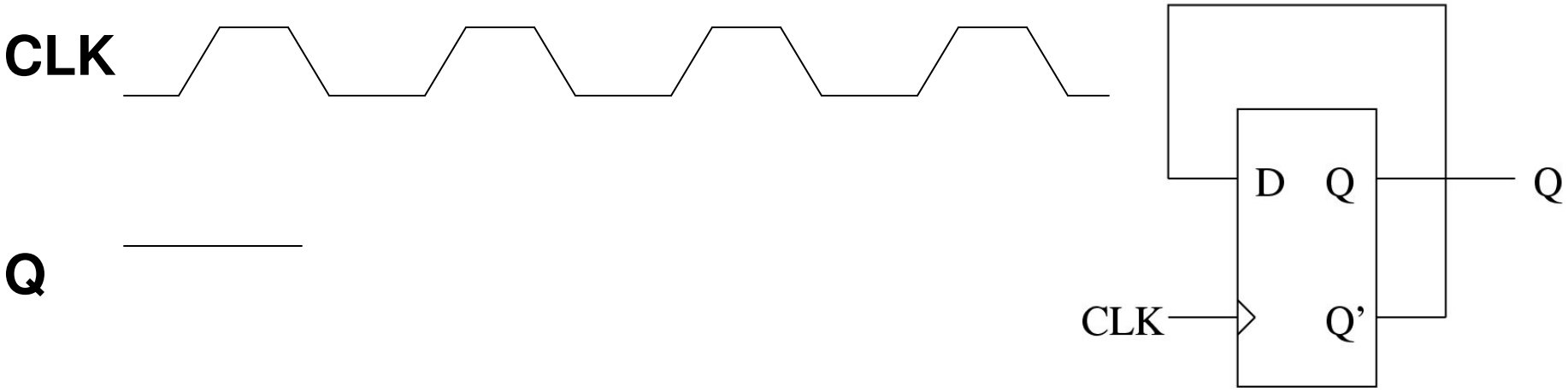
Another D Flip Flop Circuit

How does this circuit behave?



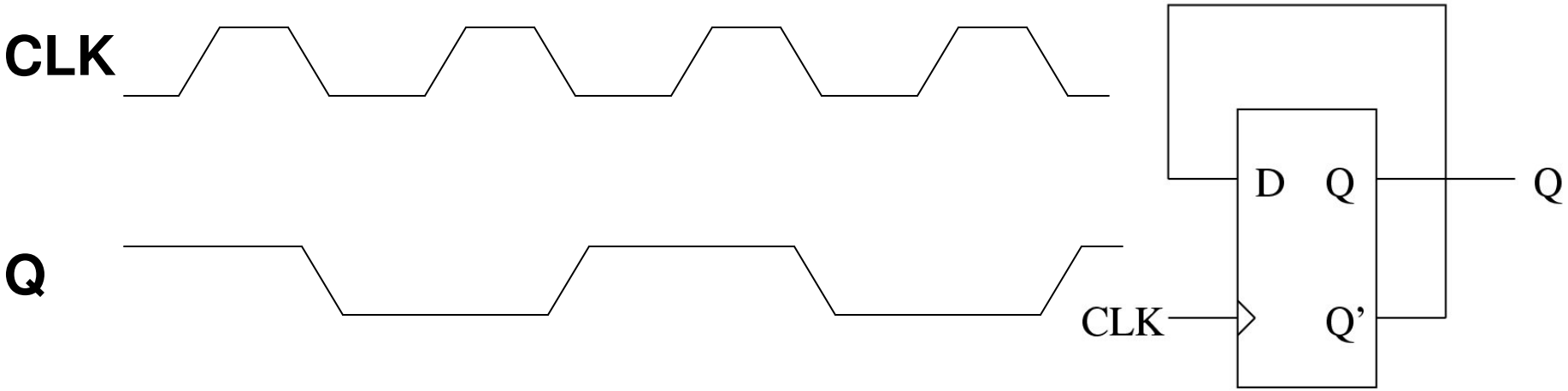
Frequency Divider

How does this circuit behave?



Frequency Divider

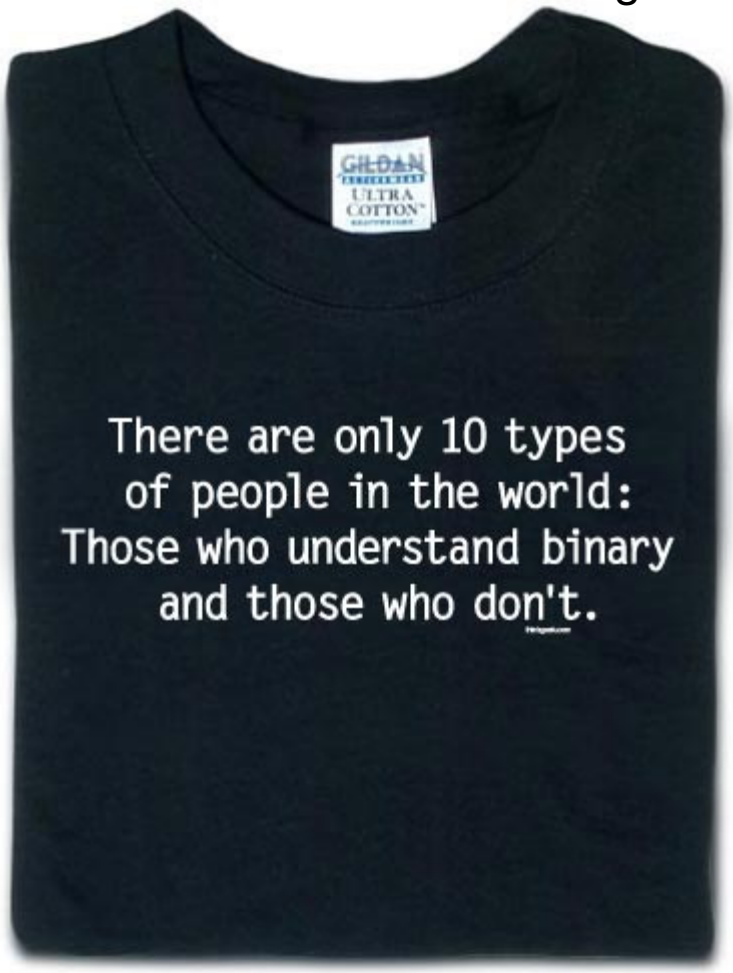
Q flips state on every downward edge of the clock



A Bit About Binary Encoding

www.thinkgeek.com

If a boolean variable can only encode two different values, how do we represent a larger number of values?



Binary Encoding

How do we represent a larger number of values?

Binary Encoding

How do we represent a larger number of values?

- As with our decimal number system: we concatenate binary digits (or “bits”) into strings

Binary Encoding

- The first (rightmost) bit is the 1's digit
- The second bit is the 2's digit
- The ith bit is the 2^{i-1} 's digit

Binary Encoding

How do we convert from binary to decimal in general?

B2	B1	B0		decimal
0	0	0		0
0	0	1		1
0	1	0		2
0	1	1		3
1	0	0		4
1	0	1		5
1	1	0		6
1	1	1		7

Binary to Decimal Conversion

$$value = B_0 + B_1 * 2^1 + B_2 * 2^2 + B_3 * 2^3 + \ldots$$

$$value = \sum_{i=0}^{N-1} B_i * 2^i$$

How do we convert from decimal to binary?

Decimal to Binary Conversion

$$\forall i : B_i \leftarrow 0$$

$$while(value \neq 0)$$

$$\{$$

$$Find\ i\ suchthat\ 2^{i+1} > value \geq 2^i$$

$$B_i \leftarrow 1$$

$$value \leftarrow value - 2^i$$

$$\}$$

Binary Counter

How would we build a circuit that counts the number of clock ticks that have gone by?

B2	B1	B0
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

Binary Counter

How would we build a circuit that counts the number of clock ticks that have gone by?

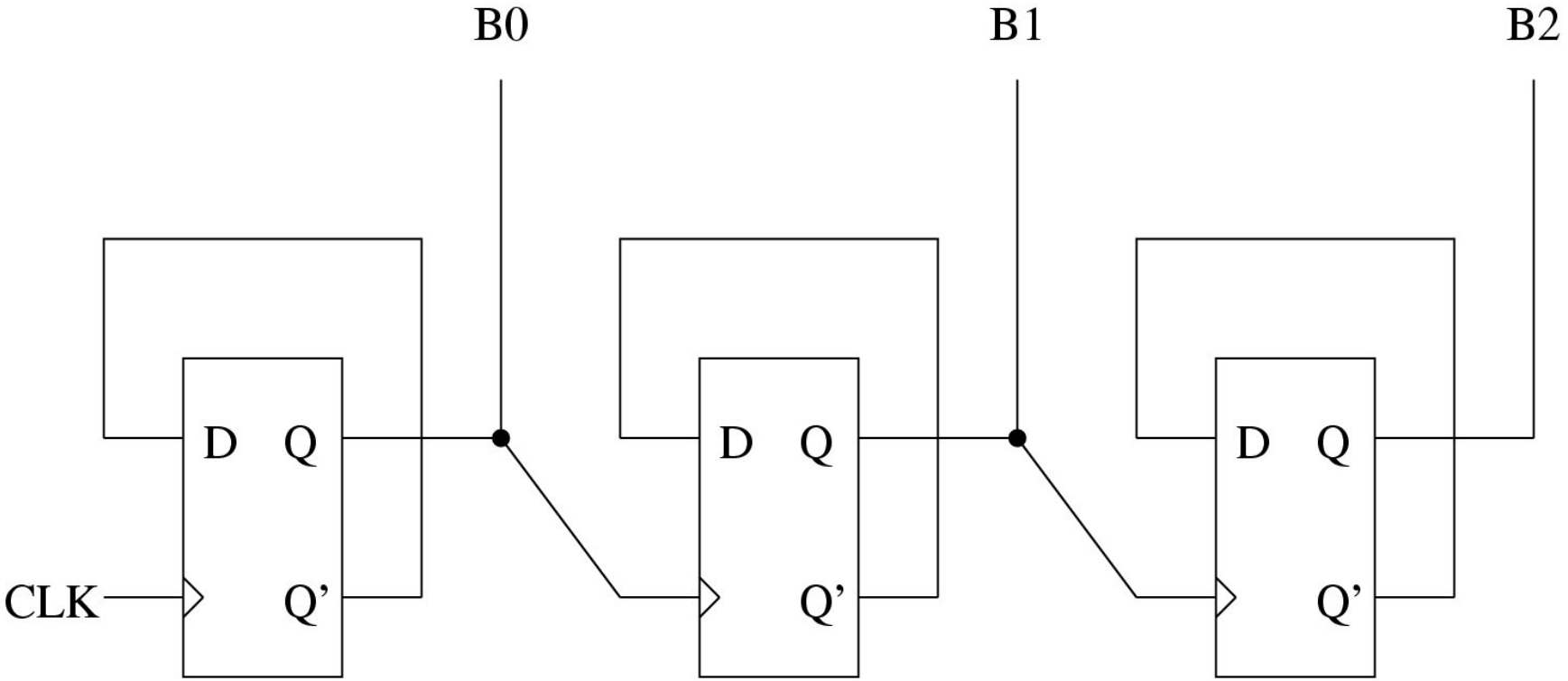
Insight:

- B1 changes state at half the frequency that B0 does
- B2 changes state at half the frequency of B1

B2	B1	B0
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

Ripple Counter

The carry “ripples” down the chain …



Ripple Counter

- Problem: the bits do not change state at the same time
- How would we go about designing a circuit to fix this?

Synchronous Counter

All flip-flops must change state at the same time. How do we ensure this?

Synchronous Counter

All flip-flops must change state at the same time. How do we ensure this?

- The same clock signal must be used for all flip-flops

Next Time

- A bit more on sequential circuit design
- Processor components
  - Registers
  - Memory
  - Arithmetic logical units

Last Time

- Flip-flops
- Sequential logic
  - Definition
  - Design
- Counters:
  - Ripple counter
  - Synchronous counter

Today

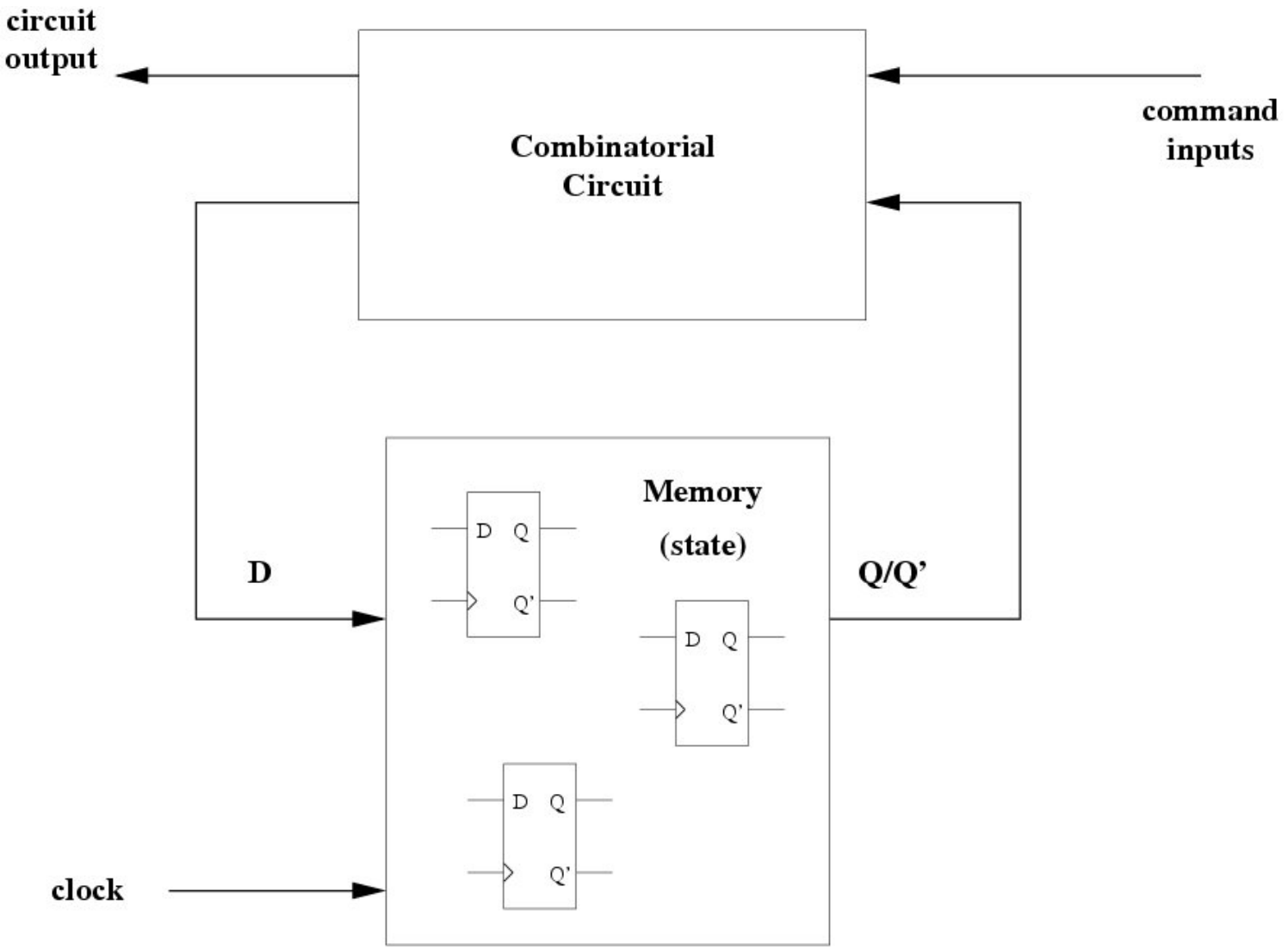
- Check our synchronous counter design
- Sequential logic with external inputs
  - Example: up/down counter
- Microprocessor components
  - Memory
  - Registers

Sequential Circuit Design

A sequential circuit combines:

- A set of memory elements
  - In our case, D flip-flops
- A combinatorial logic circuit
  - Input:
    - Current state of the flip-flops (the Q's)
    - Any other circuit inputs (e.g., command signals)
  - Output:
    - The next states (the D inputs to the flip-flops)
    - Any other outputs from the circuit

Sequential Circuit Outline



Sequential Circuit Design Process

1. Determine how many flip-flops that you will need; label them uniquely (e.g., 0, 1, 2…)
   - The different combinations of flip-flop states (0 or 1) correspond to the possible circuit states
2. Write out the truth table
   - For each circuit state (the Q's) and command input, determine what the next circuit state should be (call these D's)

Sequential Circuit Design Process

3. For each next state output bit:
   - Write the algebraic expression in minterm form
   - Simplify
   - Design the corresponding combinatorial circuit

Sequential Circuit Design Process

4. Design the full circuit:
   - Draw the flip-flops and command inputs
   - Use the Q's and the command inputs as inputs into your combinatorial circuit(s)
   - Connect the outputs of your combinatorial circuit(s) to the input side of your D flip-flops (i.e., the D input)
   - Connect an external clock signal to all of your D flip-flop clock inputs

Sequential Circuit Design Notes

- In all of our examples to date, the “circuit outputs” have been the same as our current flip-flop states (our Q’s)
  - More generally, they can be any function of these Q’s and the command input

Synchronous Counter

All flip-flops must change state at the same time. How do we ensure this?

- The same clock signal must be used for all flip-flops

Up/Down Synchronous Counter

For our example

- 2 bit counter (so 2 flip-flops)
- 1 control line:
  - C=0 -> count up on the next clock
  - C=1 -> count down on the next clock