University of Illinois at Urbana-Champaign
Dept. of Electrical and Computer Engineering

ECE 120: Introduction to Computing

The Clock Abstraction

Latches Can be Used Directly for Sequential Systems

Many high-speed designs are based on latches.

A set of latches serves as input to combinational logic.

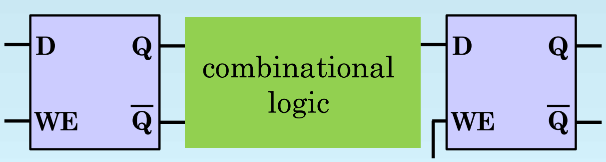
The outputs are stored in latches.

And so forth. (Eventually making a loop.)

Complemented Inputs are Usually "Free"

Now you can understand why **complemented inputs are usually "free"** (do not require inverters to generate).

If inputs come from latches, we can simply connect wires to the **Q** or to the **Q'** outputs.



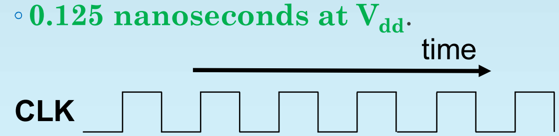
A Clock Signal is Idealized as a Square Wave

A **clock signal** is used to drive the **WE** inputs of the latches.

The clock is (ideally) a **square wave**.

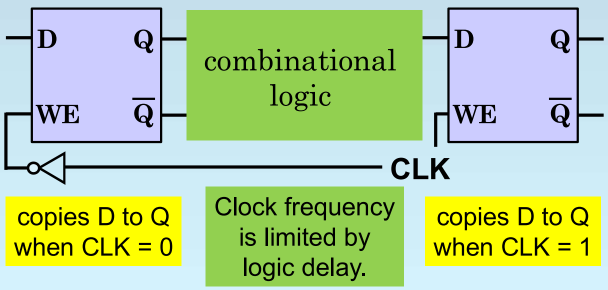
So a **4 GHz** clock repeats:

- **0.125 nanoseconds at 0V**,
- **0.125 nanoseconds at V_{dd}**.



Sets of Latches Alternate Write Enable Sense

Alternating sets of latches accept input in alternating clock phases (low and high).



Reality is Substantially More Challenging

Ideally, the clock
- is a square wave, and
- edges arrive at all latches at the same time.

In the real world,
- there are **no square waves**, and
- "at the **same time**" is **not meaningful** (an effect of special relativity).

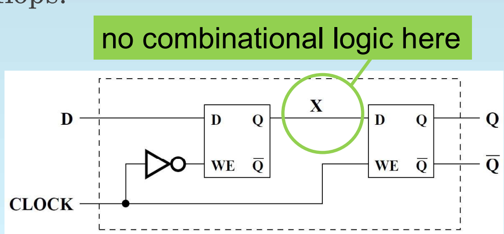
We will use a simpler abstraction.

And leave the problem of **clock skew** (timing of clock edges) to the circuit designers.

Assume Flip-Flops and a Common Clock in Our Class

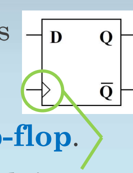
In particular, we combine consecutive sets of latches into "flip-flops" (as shown below),

and only allow combinational logic between flip-flops.

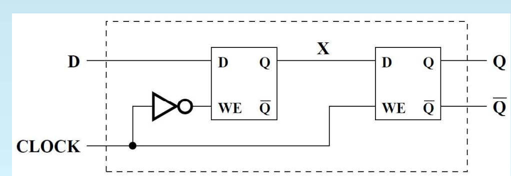


Our Class Uses only One Type of Flip-Flop

Design below (symbol to right) shows
- a **master-slave implementation** (using two gated D latches) of
- a **positive-edge-triggered D flip-flop**.



Note the use of a triangle for the clock input.



What Does the Name Mean?

A “**flip-flop**” stores one bit, and changes value **once each clock cycle**.

A “**D flip-flop**” accepts the bit to store **using a D(ata) input**.

“**Positive-edge-triggered**” means that the flip-flop’s value **changes on the rising edge** (low to high) of the clock signal.

Our Simplifying Assumptions Imply Discrete Time

So what does our use of flip-flops and ignoring clock skew imply?

Discrete time!

Time is an integer.

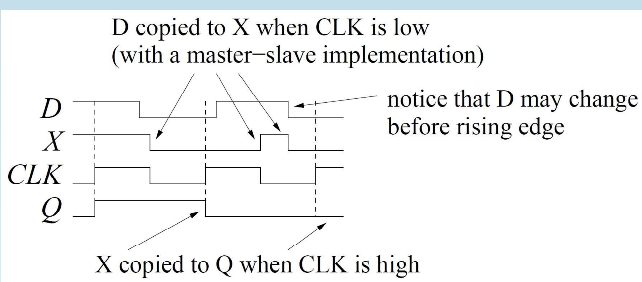
Each clock cycle is one unit of time.

Flip-flops copy their **D** inputs to their Q outputs on the rising edge of the clock.

Between integer values of time, **we assume that nothing changes**.

Our Flip-Flop Stores a New Bit on Each Rising Clock Edge

Let’s see how our flip-flop works internally. Remember that **X** is between the two latches.



Our Flip-Flop Stores a New Bit on Each Rising Clock Edge

In our class, all of your designs for sequential systems will be **clocked synchronous sequential circuits**. These assume use of

- flip-flops (for us, positive-edge-triggered D flip-flops) and
- a common (synchronous) clock signal.

Components such as latches and flip-flops are examples of **sequential feedback circuits**. You should understand how they work, but we don’t expect you to design any.