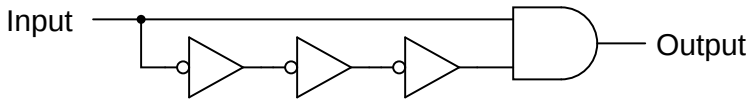

Sequential Logic: Flipflops

Questions

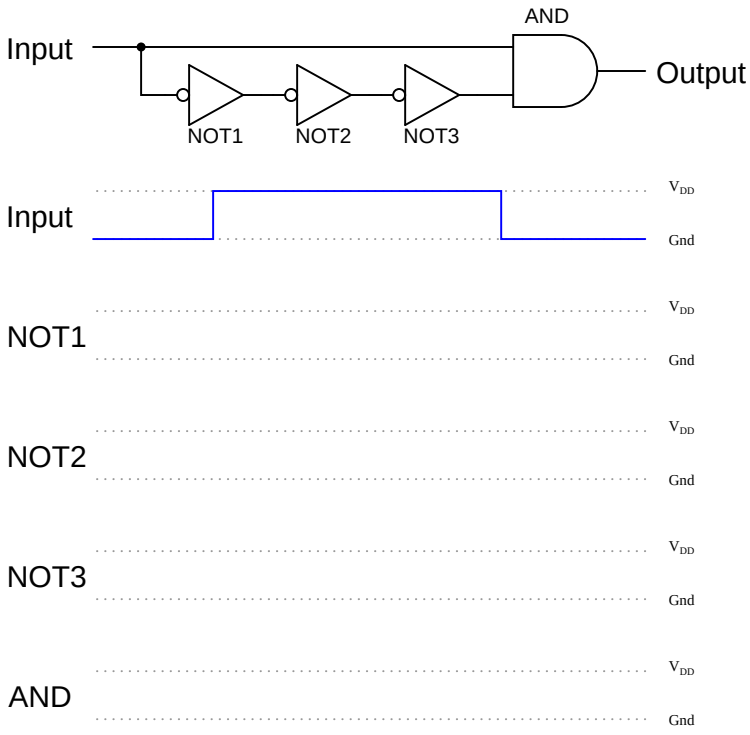
Question 1

Usually, propagation delay is considered an undesirable characteristic of logic gates, which we simply have to live with. Other times, it is a useful, even necessary, trait. Take for example this circuit:



If the gates constituting this circuit had zero propagation delay, it would perform no useful function at all. To verify this sad fact, analyze its steady-state response to a "low" input signal, then to a "high" input signal. What state is the AND gate's output always in?

Now, consider propagation delay in your analysis by completing a timing diagram for each gate's output, as the input signal transitions from low to high, then from high to low:

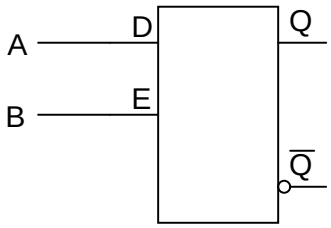


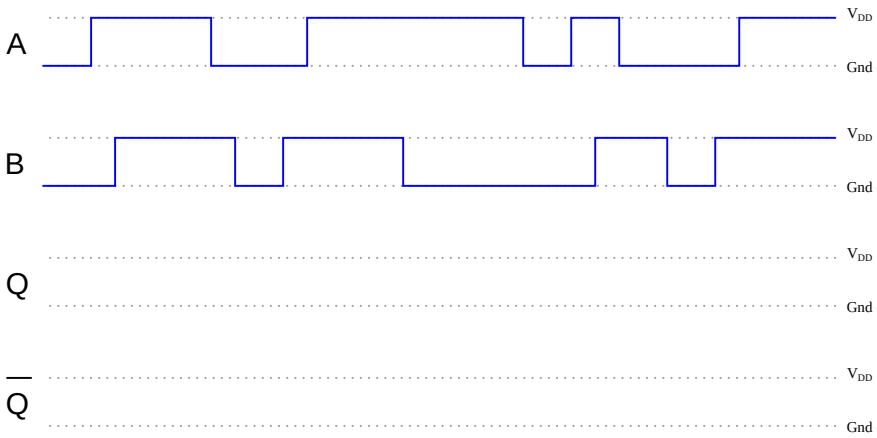
What do you notice about the state of the AND gate's output now?

file 01362

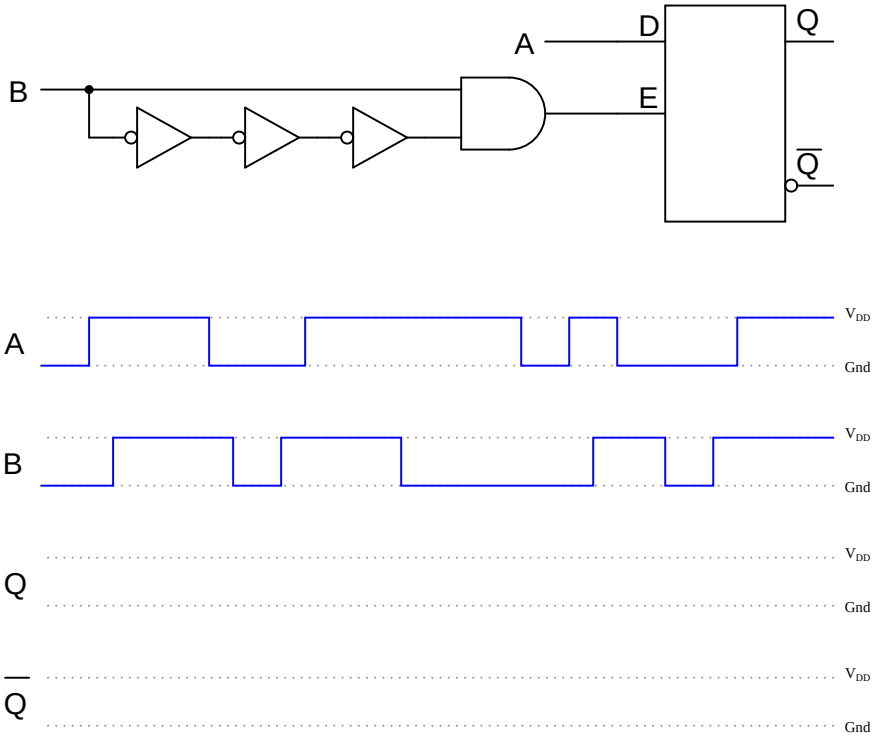
Question 2

Determine the Q and $\overline{Q}$ output states of this D-type gated latch, given the following input conditions:





Now, suppose we add a propagation-delay-based one-shot circuit to the Enable line of this D-type gated latch. Re-analyze the output of the circuit, given the same input conditions:

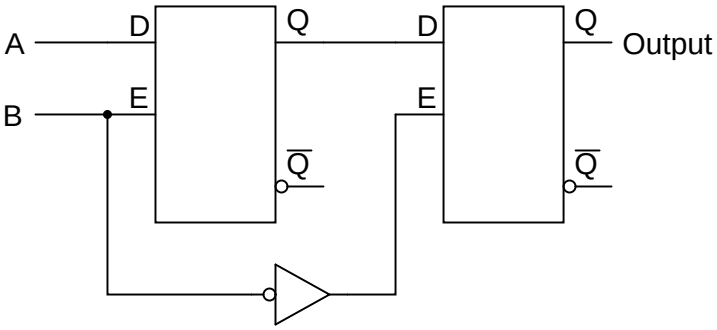


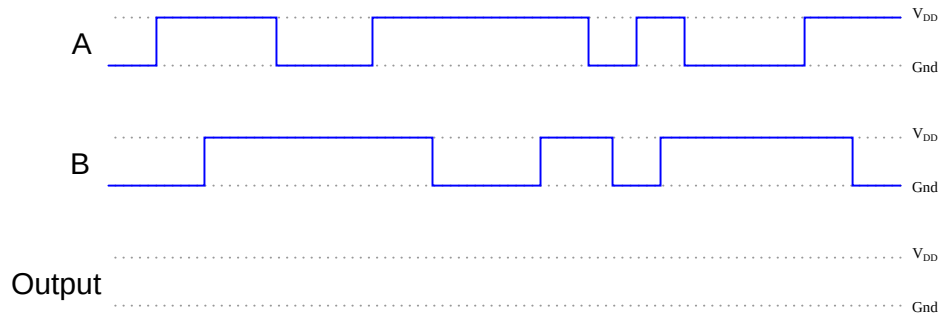
Comment on the differences between these two circuits' responses, especially with reference to the enabling input signal (B).

file 01364

Question 3

Determine the final output states over time for the following circuit, built from D-type gated latches:



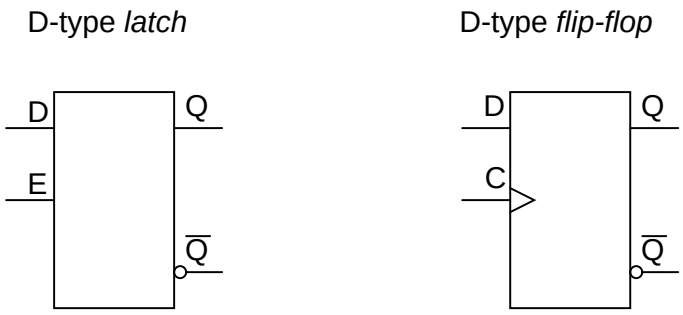


At what specific times in the pulse diagram does the final output assume the input's state? How does this behavior differ from the normal response of a D-type latch?

file 01363

Question 4

Shown here are two digital components: a D-type *latch* and a D-type *flip-flop*:

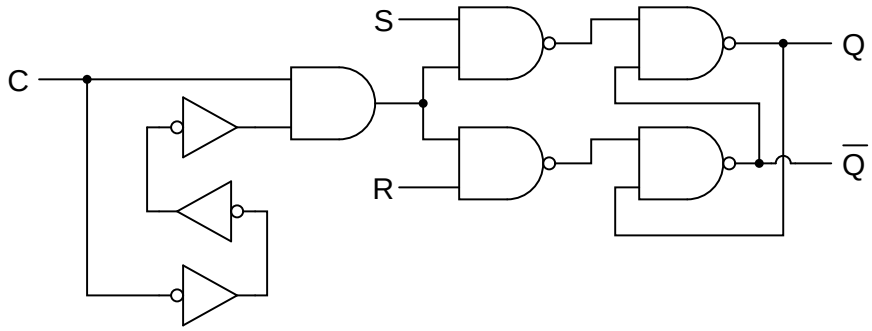


Other than the silly name, what distinguishes a "flip-flop" from a latch? How do the two circuits differ in function?

file 01365

Question 5

Explain how the addition of a propagation-delay-based one-shot circuit to the enable input of an S-R latch changes its behavior:

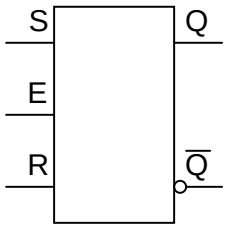


Specifically, reference your answer to a truth table for this circuit.
file 01366

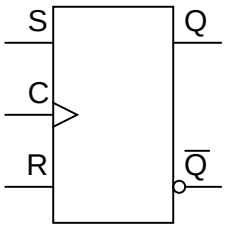
Question 6

Plain S-R latch circuits are "set" by activating the S input and de-activating the R input. Conversely, they are "reset" by activating the R input and de-activating the S input. Gated latches and flip-flops, however, are a little more complex:

S-R gated latch



S-R flip-flop



Describe what input conditions have to be present to force each of these circuits to *set* and to *reset*.

For the S-R gated latch:

- Set by . . .
- Reset by . . .

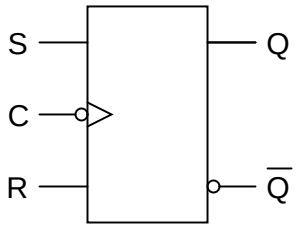
For the S-R flip-flop:

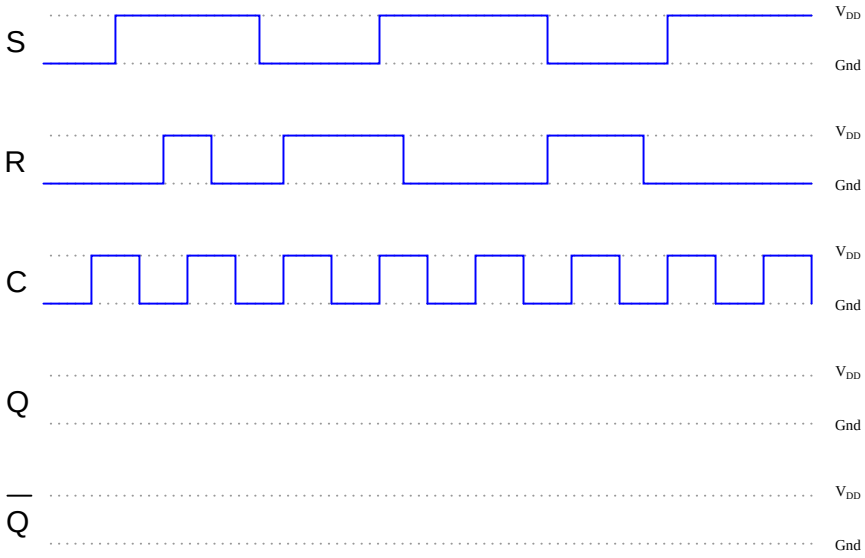
- Set by . . .
- Reset by . . .

file 02935a

Question 7

Determine the output states for this S-R flip-flop, given the pulse inputs shown:

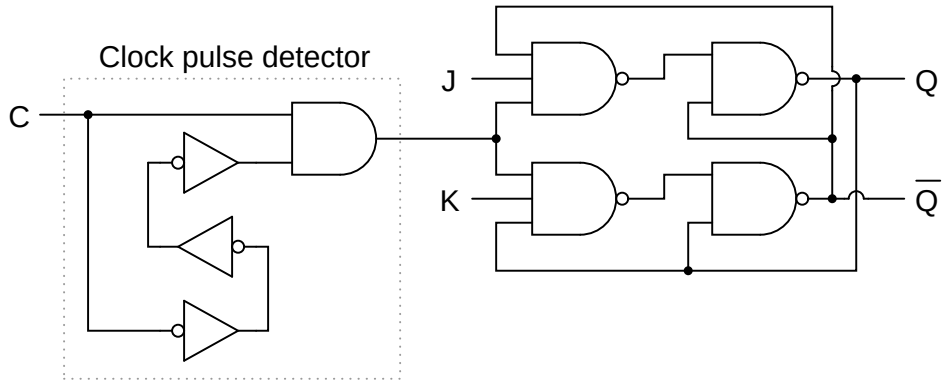




file 01367

Question 8

An extremely popular variation on the theme of an S-R flip-flop is the so-called *J-K flip-flop* circuit shown here:



Note that an S-R flip-flop becomes a J-K flip-flop by adding another layer of feedback from the outputs back to the enabling NAND gates (which are now three-input, instead of two-input). What does this added feedback accomplish? Express your answer in the form of a truth table.

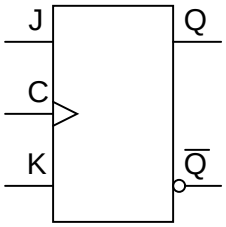
One way to consider the feedback lines going back to the first NAND gates is to regard them as extra *enable* lines, with the Q and $\overline{Q}$ outputs selectively enabling just one of those NAND gates at a time.

file 01368

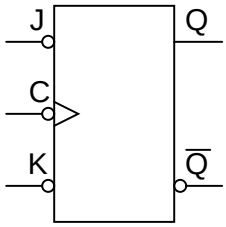
Question 9

Determine what input conditions are necessary to set, reset, and toggle these two J-K flip-flops:

Active-high inputs



Active-low inputs



For the J-K flip-flop with active-high inputs:

- Set by . . .
- Reset by . . .
- Toggle by . . .

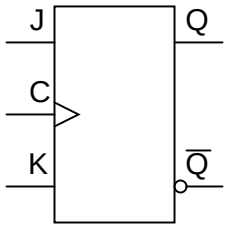
For the J-K flip-flop with active-low inputs:

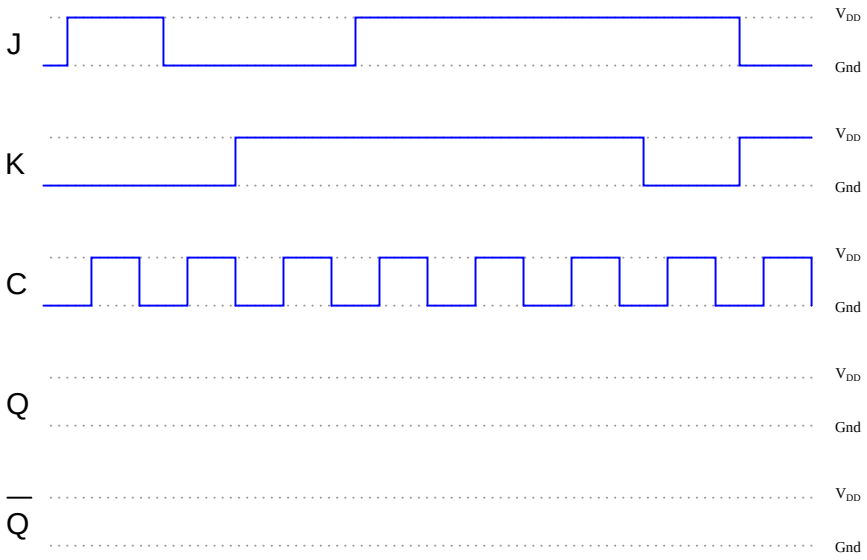
- Set by . . .
- Reset by . . .
- Toggle by . . .

file 02936

Question 10

Determine the output states for this J-K flip-flop, given the pulse inputs shown:

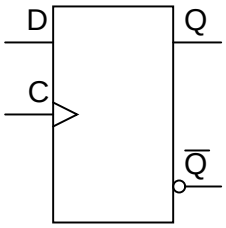


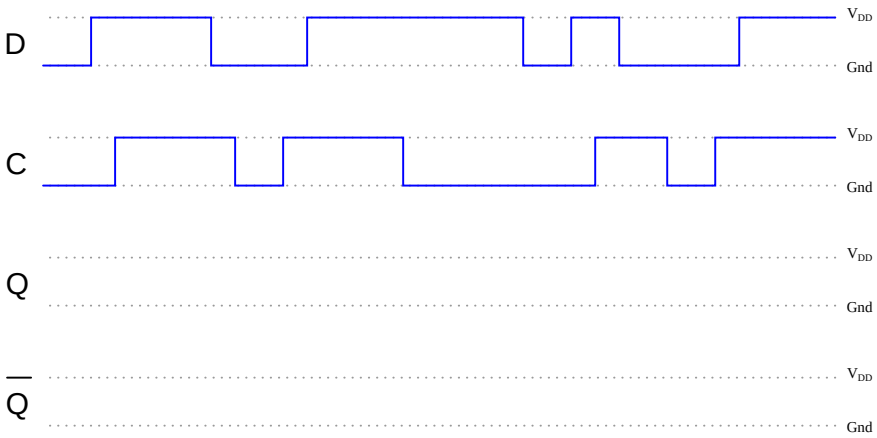


file 02934

Question 11

Determine the output states for this D flip-flop, given the pulse inputs shown:

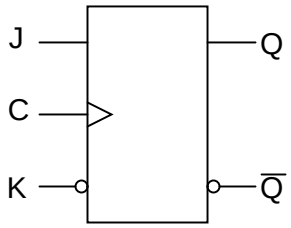


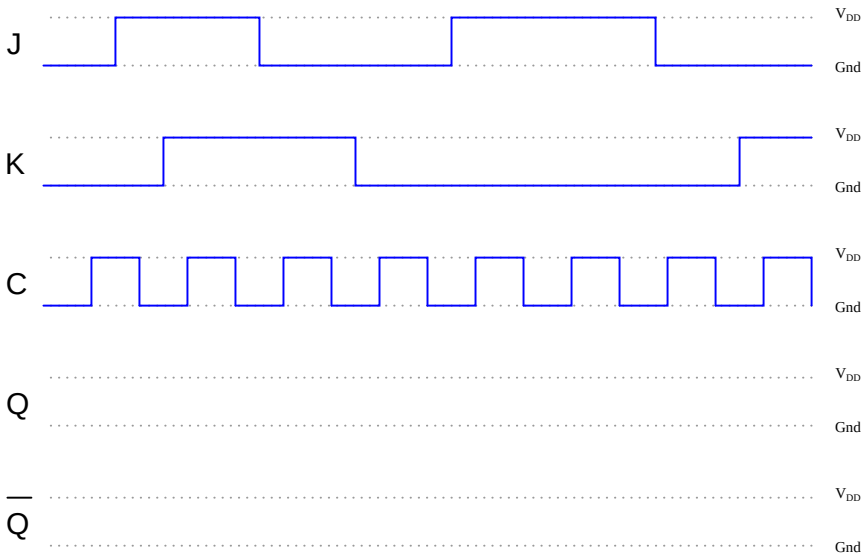


file 02940

Question 12

Determine the output states for this J-K flip-flop, given the pulse inputs shown:

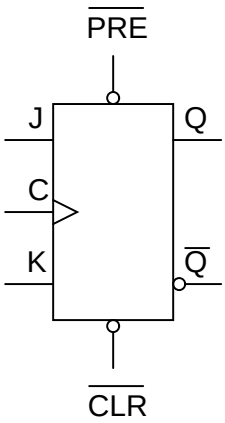




file 02939

Question 13

Flip-flops often come equipped with *asynchronous input lines* as well as synchronous input lines. This J-K flip-flop, for example, has both "preset" and "clear" asynchronous inputs:



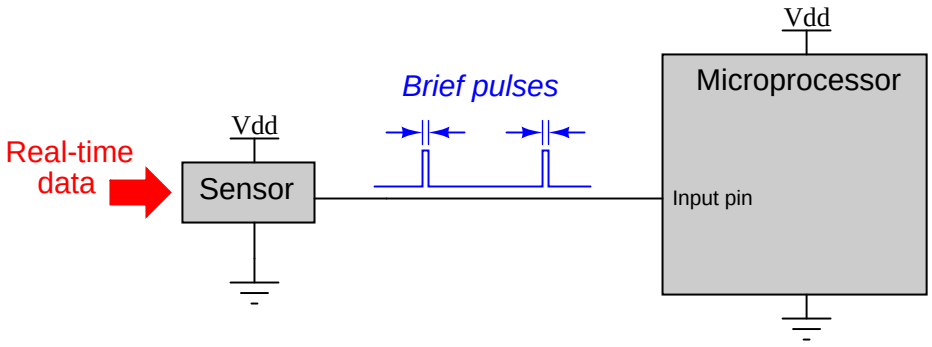
Describe the functions of these inputs. Why would we ever want to use them in a circuit? Explain what the "synchronous" inputs are, and why they are designated by that term.

Also, note that both of the asynchronous inputs are *active-low*. As a rule, asynchronous inputs are almost always active-low rather than active-high, even if all the other inputs on the flip-flop are active-high. Why do you suppose this is?

file 01370

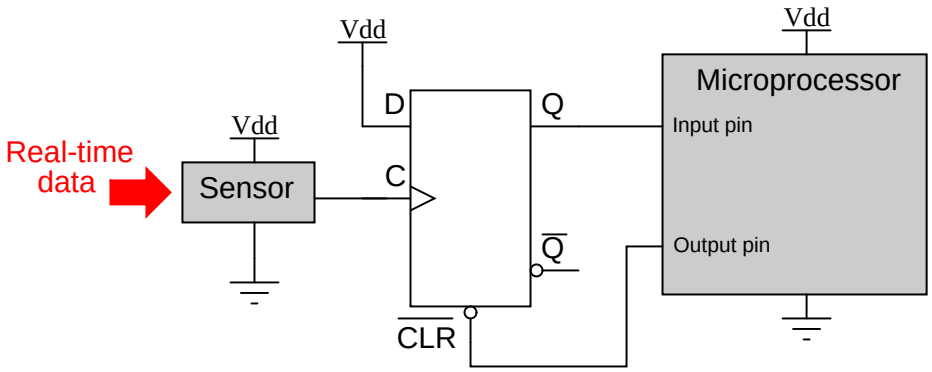
Question 14

A scientist is using a microprocessor system to monitor the boolean ("high" or "low") status of a particle sensor in her high-speed nuclear experiment. The problem is, the nuclear events detected by the sensor come and go much faster than the microprocessor is able to sample them. Simply put, the pulses output by the sensor are too brief to be "caught" by the microprocessor every time:



She asks several technicians to try and fix the problem. One tries altering the microprocessor's program to achieve a faster sampling rater, to no avail. Another recalibrates the particle sensor to react slower, but this only results in missed data (because the real world data does not slow down accordingly!). No solution tried so far works, because the fundamental problem is that the microprocessor is just too slow to "catch" the extremely short pulse events coming from the particle sensor. What is required is some kind of external circuit to "read" the sensor's state at the leading edge of a sample pulse, and then hold that digital state long enough for the microprocessor to reliably register it.

Finally, another electronics technician comes along and proposes this solution, but then goes on vacation, leaving you to implement it:

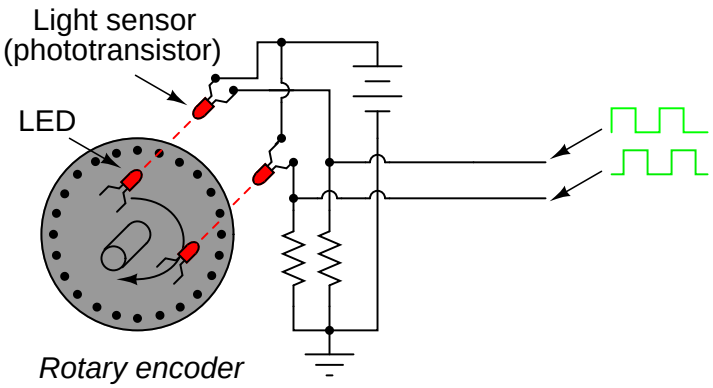


Explain how this D-type flip-flop works to solve the problem, and what action the microprocessor has to take on the output pin to make the flip-flop function as a detector for multiple pulses.

file 01464

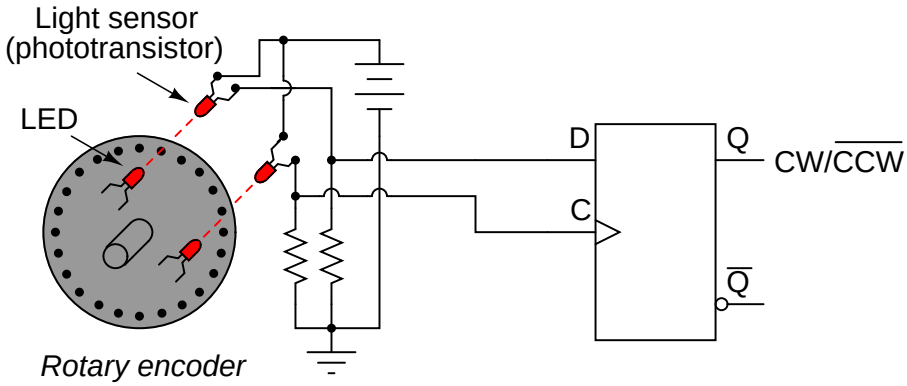
Question 15

A common type of rotary encoder is one built to produce a *quadrature* output:



The two LED/phototransistor pairs are arranged in such a way that their pulse outputs are always 90^o out of phase with each other. Quadrature output encoders are useful because they allow us to determine direction of motion as well as incremental position.

Building a quadrature direction detector circuit is easy, if you use a D-type flip-flop:

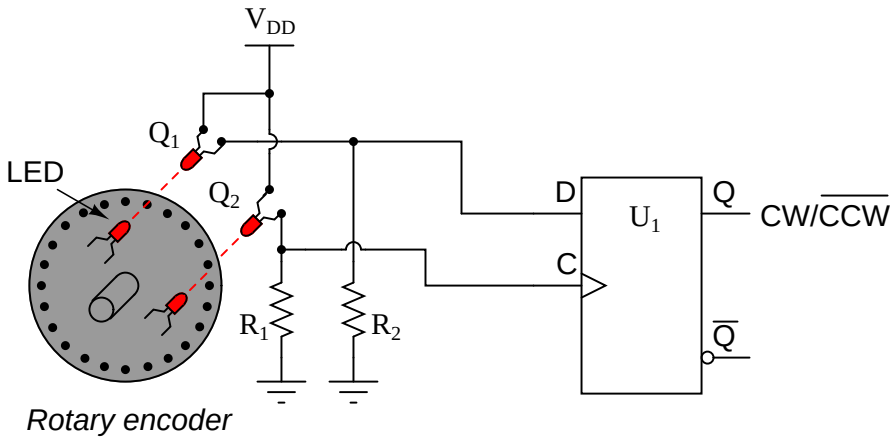


Analyze this circuit, and explain how it works.

file 01384

Question 16

Identify at least one component fault that would cause the flip-flop to indicate "clockwise" all the time, regardless of encoder motion:

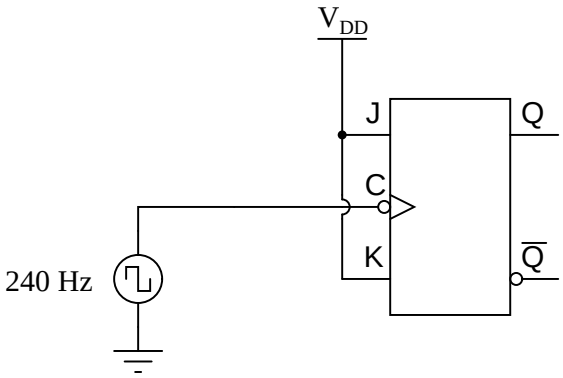


For each of your proposed faults, explain *why* it will cause the described problem.

file 03895

Question 17

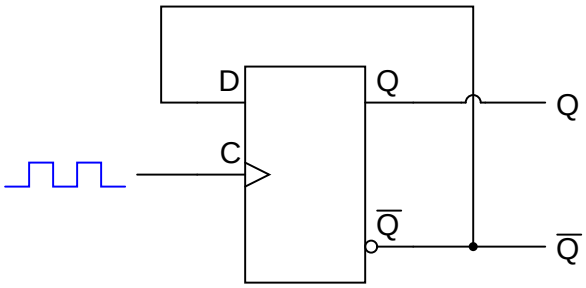
If the clock frequency driving this flip-flop is 240 Hz, what is the frequency of the flip-flop's output signals (either Q or $\overline{Q}$)?



file 01372

Question 18

Although the *toggle* function of the J-K flip-flop is one of its most popular uses, this is not the only type of flip-flop capable of performing a toggle function. Behold the surprisingly versatile D-type flip-flop configured to do the same thing:



Explain how this circuit performs the "toggle" function more commonly associated with J-K flip-flops.
file 03453