
Sequential Logic: Counters

Questions

Question 1

Count from zero to fifteen, in binary, keeping the bits lined up in vertical columns like this:

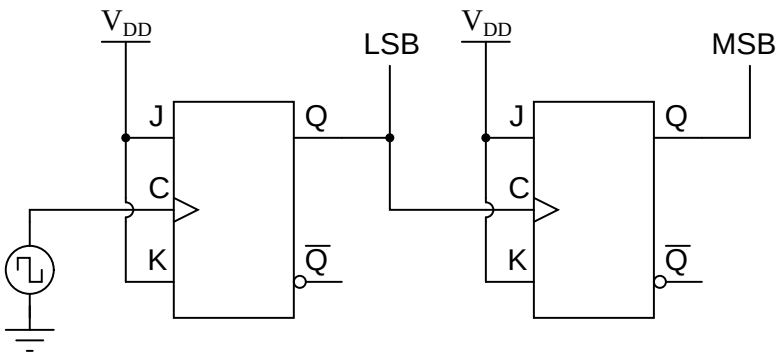
```
0000
0001
0010
. . .
```

Now, reading from top to bottom, notice the alternating patterns of 0's and 1's in each place (i.e. one's place, two's place, four's place, eight's place) of the four-bit binary numbers. Note how the least significant bit alternates more rapidly than the most significant bit. Draw a timing diagram showing the respective bits as waveforms, alternating between "low" and "high" states, and comment on the *frequency* of each of the bits.

file 01373

Question 2

Shown here is a simple two-bit binary counter circuit:



The Q output of the first flip-flop constitutes the least significant bit (LSB), while the second flip-flop's Q output constitutes the most significant bit (MSB).

Based on a timing diagram analysis of this circuit, determine whether it counts in an *up* sequence (00, 01, 10, 11) or a *down* sequence (00, 11, 10, 01). Then, determine what would have to be altered to make it count in the other direction.

file 01374

Question 3

Draw the schematic diagram for a four-bit binary "up" counter circuit, using J-K flip-flops.

file 01375

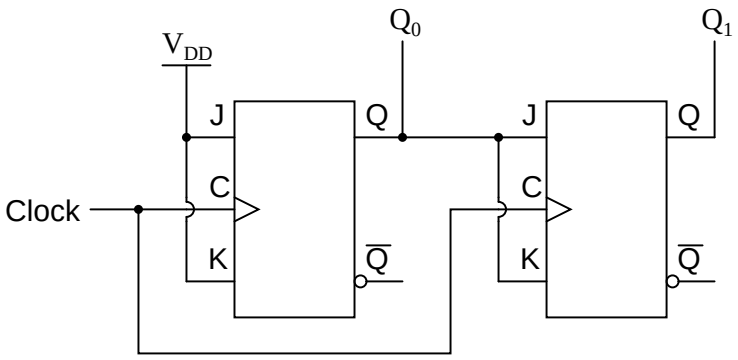
Question 4

Counter circuits built by cascading the output of one flip-flop to the clock input of the next flip-flop are generally referred to as *ripple* counters. Explain why this is so. What happens in such a circuit that earns it the label of "ripple"? Is this effect potentially troublesome in circuit operation, or is it something of little or no consequence?

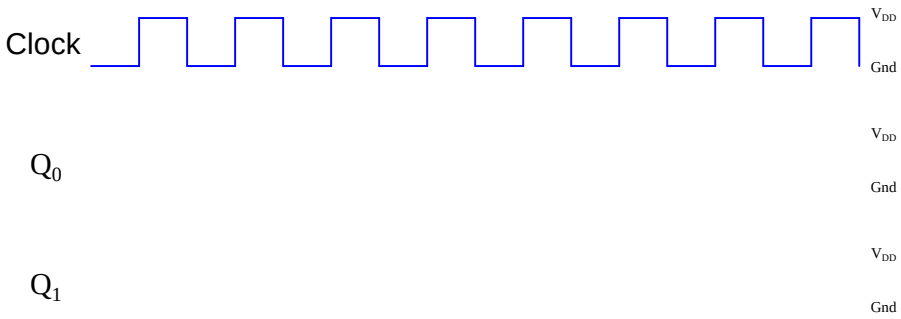
file 01388

Question 5

A style of counter circuit that completely circumvents the "ripple" effect is called the *synchronous* counter:



Complete a timing diagram for this circuit, and explain why this design of counter does not exhibit "ripple" on its output lines:

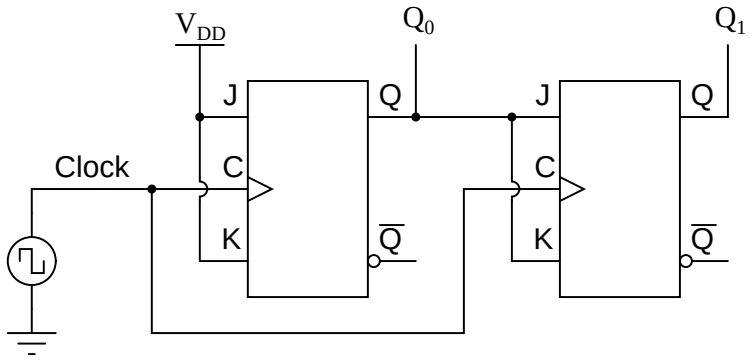


Challenge question: to *really* understand this type of counter circuit well, include propagation delays in your timing diagram.

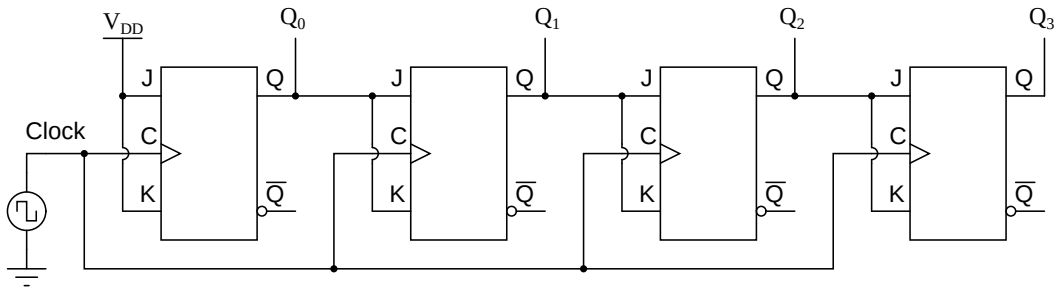
file 01396

Question 6

A student just learned how a two-bit synchronous binary counter works, and he is excited about building his own. He does so, and the circuit works perfectly.



After that success, student tries to expand on their success by adding more flip-flops, following the same pattern as the two original flip-flops:

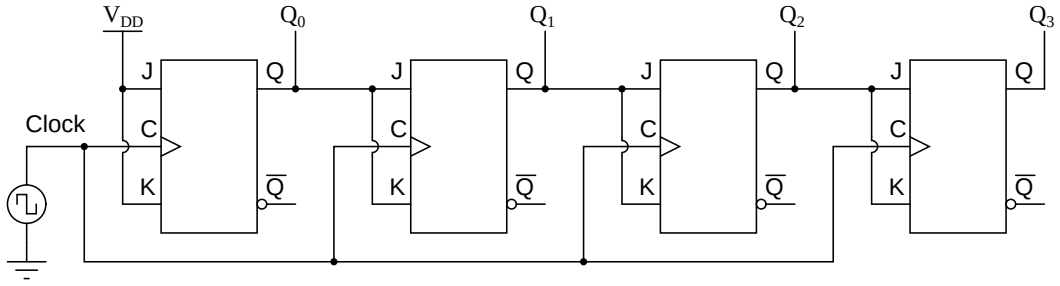


Unfortunately, this circuit didn't work. The sequence it generates is not a binary count. Determine what the counting sequence of this circuit is, and then try to figure out what modifications would be required to make it count in a proper binary sequence.

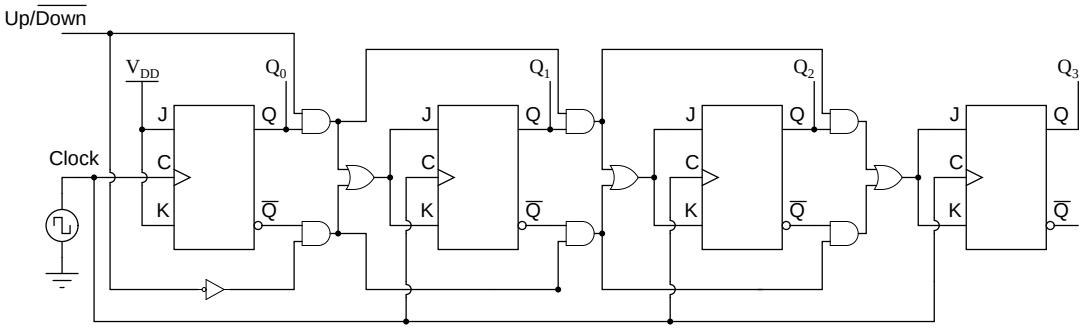
file 01397

Question 7

Synchronous counter circuits tend to confuse students. The circuit shown here is the design that most students think ought to work, but actually doesn't:



Shown here is an up/down synchronous counter design that *does* work:

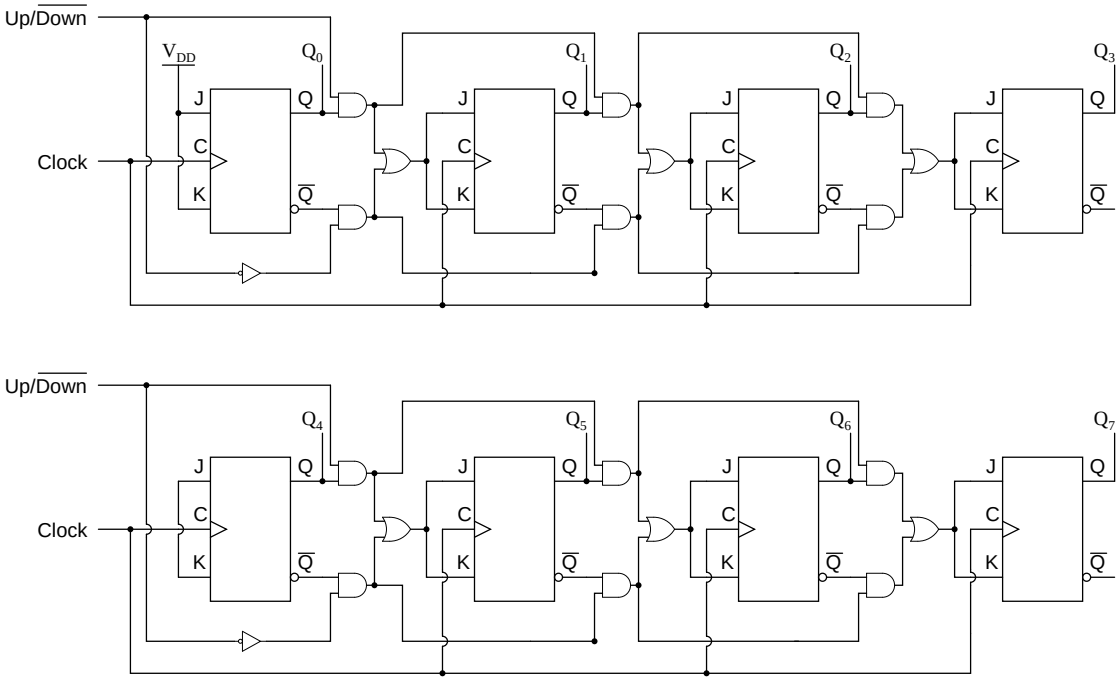


Explain why this circuit is able to function properly (counting in either direction), while the first circuit is not able to count properly at all. What do those "extra" gates do to make the counter circuit function as it should. Hint: to more easily compare the up/down counter to the faulty up counter initially shown, connect the Up/$\overline{\text{Down}}$ control line high, and then disregard any lines and gates that become disabled as a result.

file 01400

Question 8

Suppose we had two four-bit synchronous up/down counter circuits, which we wished to *cascade* to make one eight-bit counter. Draw the necessary connecting wires (and any extra gates) between the two four-bit counters to make this possible:

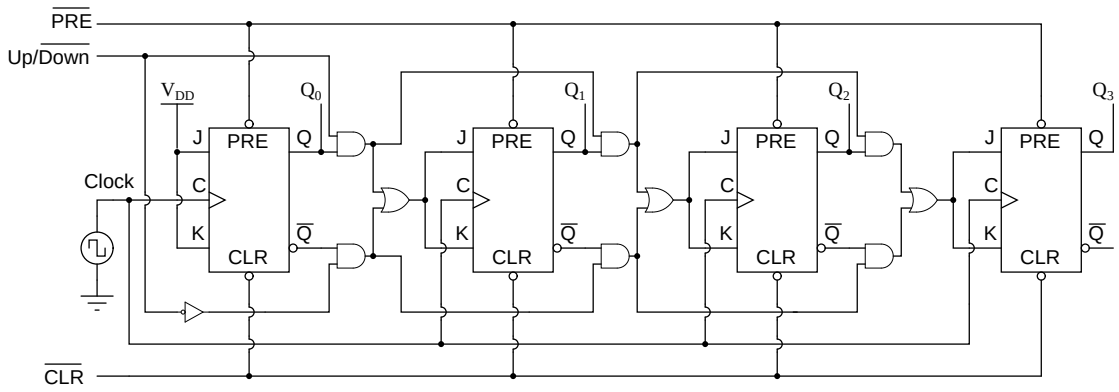


After deciding how to cascade these counters, imagine that you are in charge of building and packaging four-bit counter circuits. The customers who buy your counters might wish to cascade them as you did here, but they won't have the ability to "go inside" the packaging as you did to connect to any of the lines between the various flip-flops. This means you will have to provide any necessary cascading lines as inputs and outputs on your pre-packaged counters. Think carefully about how you would choose to build and package your four-bit "cascadable" counters, and then draw a schematic diagram.

file 01402

Question 9

Supposed we used J-K flip-flops with asynchronous inputs (Preset and Clear) to build a counter:

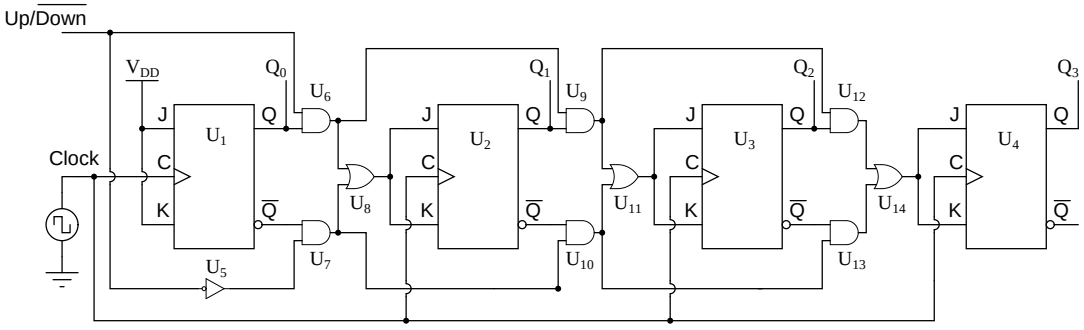


With the asynchronous lines paralleled as such, what are we able to make the counter do now that we weren't before we had asynchronous inputs available to us?

file 01401

Question 10

Identify a single fault that would allow this synchronous counter circuit to count up on demand, but not down:



Explain *why* your proposed fault would cause the problem.

file 03897

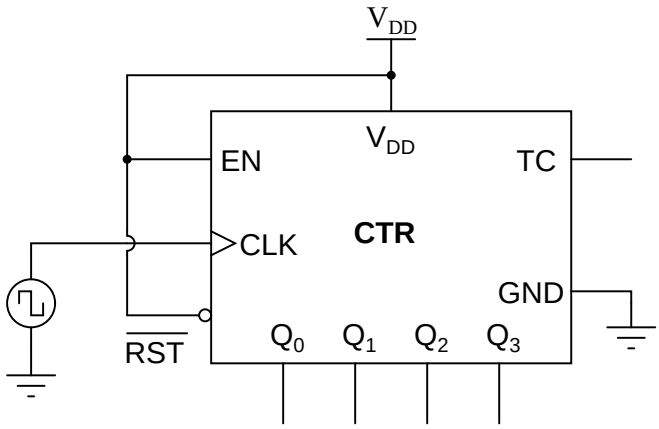
Question 11

Determine the *modulus* (MOD) of a four-bit binary counter. Determine the modulus of two four-bit binary counters cascaded to make an eight-bit binary counter.

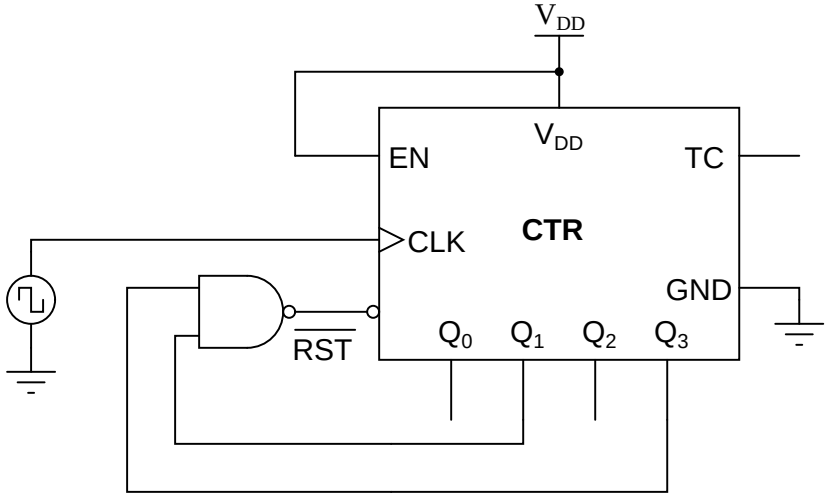
file 01404

Question 12

Consider the following four-bit binary counter integrated circuit (IC). When clocked by the square wave signal generator, it counts from 0000 to 1111 in sixteen steps and then "recycles" back to 0000 again in a single step:



There are many applications, though, where we do not wish the counter circuit to count all the way up to full count (1111), but rather recycle at some lesser terminal count value. Take for instance the application of BCD counting: from 0000 to 1001 and back again. Here is one way to truncate the counting sequence of a binary counter so that it becomes a BCD counter:



Explain how the NAND gate forces this counter to recycle after an output of 1001 instead of counting all the way up to 1111. (Hint: the reset function of this IC is assumed to be *asynchronous*, meaning the counter output resets to 0000 immediately when the $\overline{RST}$ terminal goes low.)

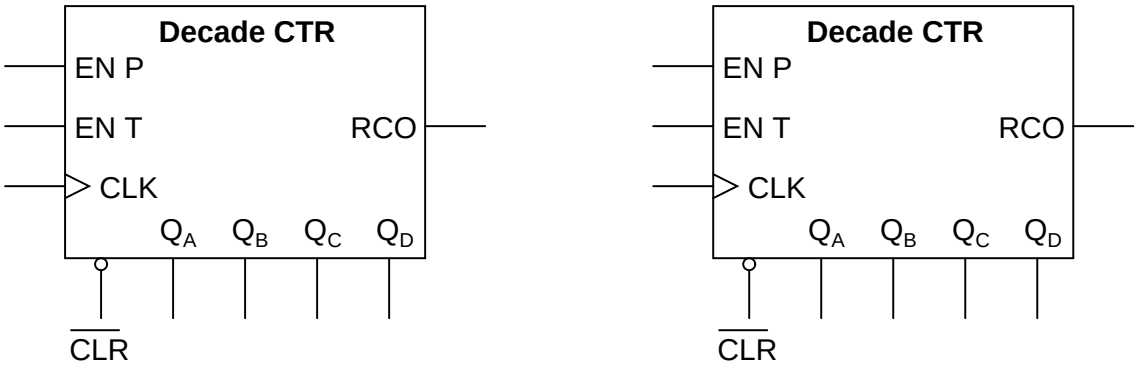
Also, show how you would modify this circuit to do the same count sequence (BCD) assuming the IC has a *synchronous* reset function, meaning the counter resets to 0000 if $\overline{RST}$ is low *and* the clock input sees a pulse.

file 02953

Question 13

Suppose you had an astable multivibrator circuit that output a very precise 1 Hz square-wave signal, but you had an application which requires a pulse once every *minute* rather than once every second. Knowing that there are 60 seconds in a minute, can you think of a way to use digital counters to act as a "frequency divider" so that every 60 multivibrator pulses equates to 1 output pulse?

You don't have a divide-by-60 counter available, but you do have several divide-by-10 ("decade") counters at your disposal. Engineer a solution using these counter units:

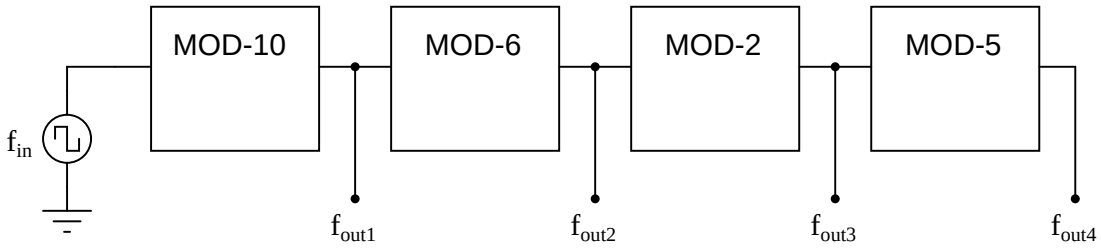


Note: assume these counter ICs have *asynchronous* resets.

file 01405

Question 14

When counters are used as frequency dividers, they are often drawn as simple boxes with one input and one output each, like this:



Calculate the four output frequencies (f_{out1} through f_{out4}) given an input frequency of 1.5 kHz:

- f_{out1} =
- f_{out2} =
- f_{out3} =
- f_{out4} =

file 02956

Question 15

A student builds a four-bit asynchronous counter circuit using CMOS J-K flip-flops. It seems to work . . . most of the time. Every once in a while, the count suddenly and mysteriously "jumps" out of sequence, to a value that is completely wrong. Even stranger than this is the fact that it seems to happen every time the student waves their hand next to the circuit.

What do you suspect the problem to be?

file 01406