

Experiment 5

Building a counter

This experiment explicitly involves several stages of a single project, from a conceptual design, to computer-based simulation, to physical implementation and debugging. The project: a decimal counter. Note that some of the necessary data is not provided on the diagrams of this experiment; it is essential that you consult the appropriate datasheets directly.

5.1 Building a device, stage by stage

Counting physical events is a common requirement that illustrates well many of the essential elements of a successful real-life application of electronic circuits. It involves:

the physical stage: the closing of a pair of mechanical contacts, a change in the illumination of a photosensor, a voltage drop that activates a solid-state relay or a transistor. Here, the issues of imperfections, of efficient signal transduction (from the physical into the electronic realm), and of signal conditioning and noise resilience play an important role;

the signal processing stage, where the now electronic (logical) signal is processed, tallied up, analyzed or compared with a desired reference condition;

the display and feedback stage, where the results are either reported to the user, recorded for later analysis, or used to provide signals to actuators and control devices of a physical system. At this stage the feedback loop back to the physical realm is closed.

Pressing a push-button switch, counting these contact closures, and displaying the resulting tally on a seven-segment display are the three stages of this lab experiment, but this general three-stage structure is shared by virtually all electronic devices involved in the interactions with the physical world.

- ? Sketch a block diagram, representing the way an electronic system would gather the information about the physical world, process it electronically, and implement the resulting control decision. Briefly describe an example, different from this experiment, that your diagram may represent.
- ? For your example experiment, indicate the approximate bandwidth requirements, or the amount of information per unit time, that you expect to be communicated between the various blocks on your diagram. Where do you expect the bottlenecks to occur?

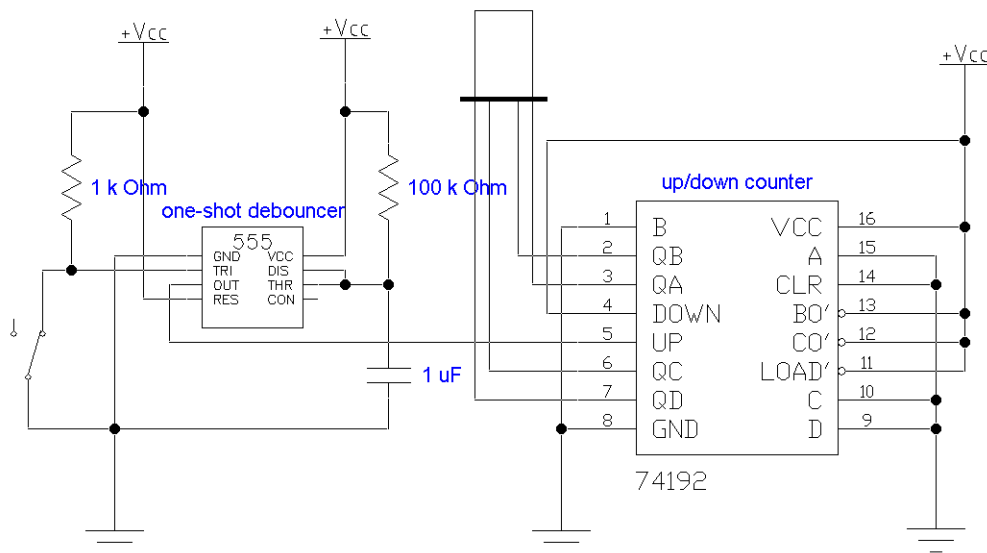


Figure 5.1: EWB circuit diagram

5.2 A virtual prototype

A 74192 IC is a convenient decimal counter, capable of reset to zero, preloading with a specified value, and of counting both up and down, as well as generating carry and borrow signals that allow one to construct multidigit counters. We will explore only a very limited subset of its capabilities.

- ❗ Use *Electronics Workbench* to simulate the circuit shown in Fig. 5.1. Initially, connect a 74192 and a four-input BCD-decoded seven-segment display, and connect a switch that will alternately toggle between a V_{cc} and the ground directly to the “Up” pin on the IC. A virtual switch toggles cleanly and instantly and will require no de-bouncing.
- ❗ Verify the operation of the counter. You may want to add additional elements such as logic probes on the outputs of the 74192, to make this task easier. See if using the switch to toggle the “Down” input of the 74192, while holding its “Up” input high produces the expected down-counting.
- ❓ Download from the web, or look up in a data book available in the lab, the datasheet of a 74192. How is the counting by 10 realized inside the IC out of four flip-flops connected in series? What is this technique called?
- ❗ Now complete the full circuit of Fig. 5.1 by connecting the 555-based debouncer. Choose external RC values so that the width of a pulse generated by the monostable is longer than about 50ms, a typical settling time of a real micromechanical switch. Use the virtual oscilloscope to verify that your calculations are correct and the pulse width is as expected.

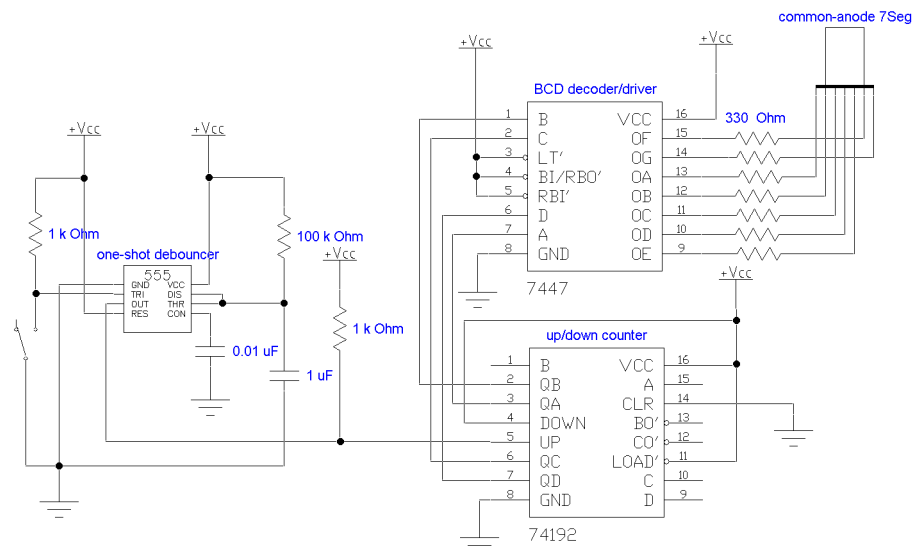


Figure 5.2: A real circuit involves additional details

5.3 Circuit realization

Additional components required

- one 555 timer IC
- one 74192 decimal counter IC
- one 7447 BCD-to-seven-segment decoder/driver
- one seven-segment LED, common anode
- miscellaneous capacitors and resistors, momentary switches

You are now ready to implement the real circuit. In addition to the de-bouncing of the real mechanical switches, the magic “BCD-encoded seven-segment display” has to be made out of separate components: a 7447 encoder/driver IC, current-limiting resistors, and a common-anode seven-segment LED display, as illustrated in Fig. 5.2.¹

- ❗ Start building the counter circuit, starting with the final stage, the LED display. Make sure you have selected a common-anode LED display, and determine its pinout. The common-anode requirement is essential, as the 7447 decoder uses negative logic: it sets those segment lines it wants to activate into the low state. Do not forget to connect the common anode pin to the V_{cc} (not shown on the diagram).
- ❓ The suggested resistor values may not be optimal. Consult the 7447 datasheet, and that of a typical LED display device, to determine the optimal current that would make the LED display as bright as possible yet will not overload the 7447 driver, either by exceeding the current-carrying capabilities of a single driver line, or by exceeding the total power dissipation limits for the worst-case scenario of all seven segments being lit at once (the digit “8”).
- ❗ Add the 74192 counter. Use a TTL square wave from the function generator set to a frequency of 0.5–5 Hz as the counting input.

¹Note that our version of *Electronics Workbench* does not have a common-anode seven-segment display, only a common-cathode one, so if you were to simulate the circuit in Fig. 5.2, the display would show patterns complementary to those you would expect for the decimal digits. In the lab, the real common-anode displays *are* available, so the circuit should work as expected.

- ❓ Which is the active edge of the counting input? Sketch a timing diagram for a sequence of several cycles of the counter, starting with the state of the outputs that represents the decimal “9”, and including all relevant input and output lines.
- ⚠ Complete the circuit by adding a push-button switch, debounced with the help of a 555 monostable. Verify a reliable counter operation. Choose the RC values so that the fastest sequence of pushes of the switch that you can produce does not miss any and does not have any extra ones.
- ❓ Vary the RC until the time constant is too short to provide effective debouncing and you just start seeing multiple counts for a single pressing of the push button. What is the minimum pulse width that would provide reliable debouncing?
- ⚠ Time permitting, explore other modes of operation of the 74192. A second switch, debounced with another 555-based one-shot, can be connected to the “Down” input, providing two-button up/down counter. A combination of four bits could be preloaded into the counter by pulsing on a “Load” input (active low). This is a decimal counter, so normally it shows only values from 0 to 9; what would happen if you pre-loaded a binary-coded four-bit value that is greater than 9 ?