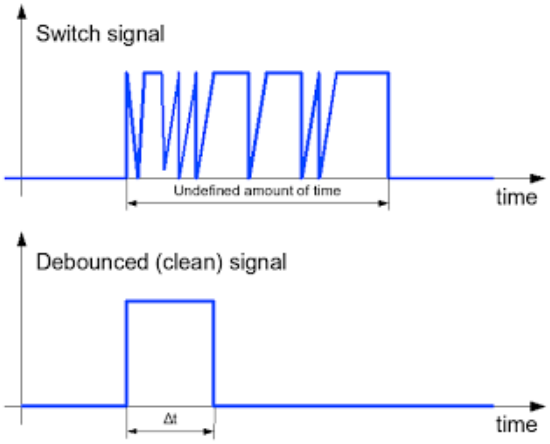
SWITCH DEBOUNCER

When a switch or pushbutton is closed the metal contacts bounce before coming to rest, effectively opening and closing the switch many times. This is of no importance for many applications (e.g. a light bulb), but when you want the switch to trigger a single event (e.g. a keyboard key), bouncing is problematic since the switch would trigger many events.

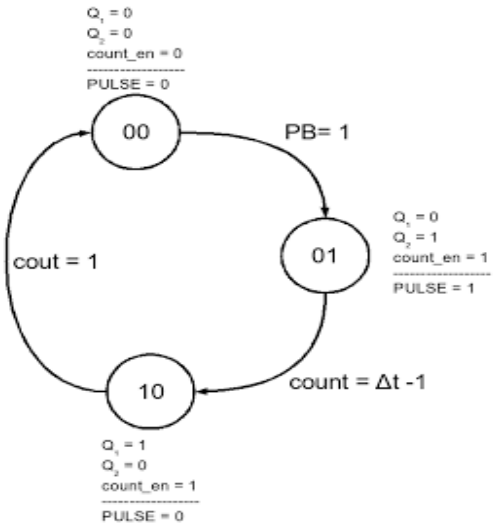
A switch debouncer is a small circuit that generates a single “clean” pulse when a physical switch or pushbutton closes.

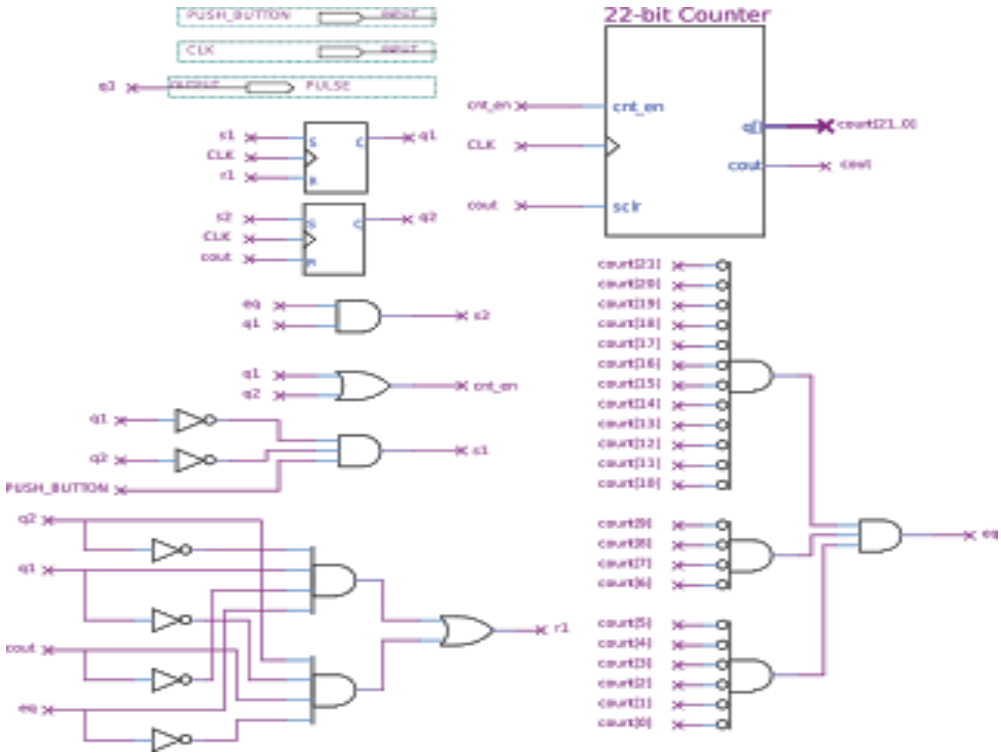


We implemented a debouncer that produces a single (or as many as you want) clock-cycle-wide pulse when a pushbutton is pushed. Also, it doesn’t allow the creation of another pulse for the next 160 ms.

Here is a state transition diagram for the debouncer followed by a nice schematic.

Note: count_en, count and cout (carry out) refer to the inputs and outputs of the counter on the debouncer schematics (see below), Δt stands for the number of clock cycles the pulse should last, PB = 1 means that the pushbutton has been pushed, and Q_1 and Q_2 refer to the output of the S-R latches.





This implementation is nice in theory (and for FPGA boards) but in reality, 22-bit counters are impossible to find and using a mix of AND and OR gates is not very practical. This is not really a problem since the combinational logic can be easily mapped into NAND-NAND logic and the 22-bit counter can be replaced by a more standard 32-bit counter for instance (in that case cout should be generated using more combinational logic).

Note: The AND gates on the right-hand side (the ones that have NOT ed inputs) are used to compare the counter count with the number of clock periods the pulse should last. By correctly setting their inputs, the pulse width can be changed to any number of clock cycles. Similarly, the 160 ms wait time is also easily customizable.

Source: http://www.carlitoscontraptions.com/2007/03/switch-debouncer/