How to de-bounce a switch using CMOS & TTL

It has come to my attention that there is a definite lack of understanding on how this simple procedure is achieved. Especially for low numbers of switches in non-microprocessor controlled devices. If the number of buttons you require on your front panel numbers less than 16 then this is how you get a clean button press.

You can use CMOS or TTL but the component values change. I'll show the component values for CMOS but it should be noted that the values should also be sweetened to taste. That is to say that you can Taylor the reaction time of the switch depending on your application's requirement and the feel you'd like to give it.

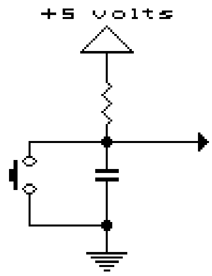
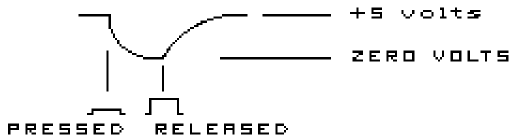
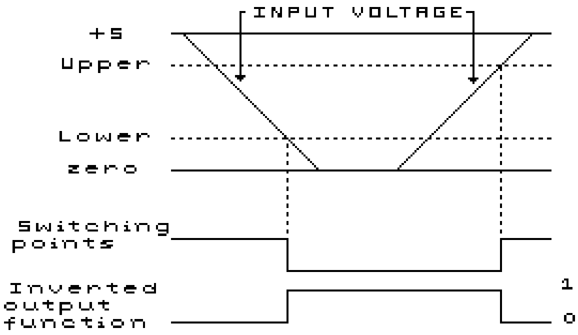


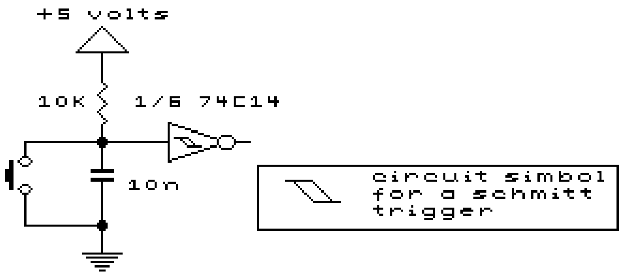
Figure 1 shows the basic arrangement. A resistor pulls a capacitor up to 5 volts (or what ever your positive supply voltage is if you're using CMOS. Up to 15 volts) Because of the time coefficient of the capacitor and resistor, this takes time to occur. The bigger the capacitor or resistor, the longer it takes. Placed at the junction of the resitor and capacitor, the momentary press button pulls the capacitor to ground. It doesn't short out the power supply because of the resistor. When the switch is made, the voltage on the capacitor falls away very rapidly to nothing. or near nothing depending on the switch. (That's another story and I'll get to that) When the switch is released the potential across the capacitor is charged up again slowly by the resistor. The charge/discharge rate is a smooth logarithmic curve. One faster than the other.



However CMOS and TTL don't like smooth transitions. Even TTL has a small linear region and will tend to get into undecided states. CMOS is even worse because it can easily run linear and because of it's high impedance nature can flollop round like an old mattress. A device called a Schmitt Trigger is needed. A Schmitt trigger is a device that will not change state till a certain threshold voltage has been reached. Called a hystereesis level. Actually there are two. An upper hysteresis and a lower. Assume the output is at 'zero'. In order to change this state to a one, the input voltage must attain or exceed the upper hysteresis level. Likewise if the output is a 'one' the input voltage must fall bellow the lower hysteresis level before it will switch back to 'zero'. Shown bellow.



There are a number of parts in both CMOS and TTL. The most common of which is the 7414 hex inverting schmitt trigger. This is basically an invertor with a preset hysteresis function. It is also available in CMOS called a 74C14. Also in CMOS is the 4093 which is a NAND gate schmitt trigger in a quad package. These are commonly used to clean up nasties in data streams. Referred to as "squaring up" since it can take a rough waveform and put square edges back on it. Also useful for resetting microprocessors which is similar to the switch debounce. But we're getting ahead of ourselves.



Above is the full debounce circuit using one sixth of a 74C14 package. Remember the output is inverted. If the input is high the output is low and visa versa. So If the resistor has pulled the capacitor high then the output of the schmitt trigger will be low. When the switch is engaged it pulls the voltage on the capacitor to near zero. As it crosses the lower hysteresis threshold the output of the schmitt trigger flys high until the switch is released. At which time the capacitor begins charging again and when it charges enough to cross the upper hysteresis level, the output flys low again. The speed at which one can press the switch is determined by the time coefficient of the resitor and capacitor. If the button is pressed again before the capacitor has charged up sufficiently, the output will remain high. It is also possible to give the switch on a time coefficient as well by placing a resistor in series with the switch to ground. However it should be noted that this must be significantly smaller than the value of the pull-up resistor otherwise it will merely equalize somewhere in the middle. No switching will take place since nether threshold was obtained.

Which brings me neatly to a word of caution about momentary pressbutton switches. Most switches have a moment that pressed two bits of metal together shorting the two switch terminals together. This kind of switch has very little resistance. So little as to make no odds as far as our switch is concerned. However it is common these days to find switches made from rubber actuators with conductive surfaces. These typically have a resistance of 50 ohms or so. But can be as high as 200 ohms. This means that when the switch makes, it only pulls down by the resistance across the pad. As if you had placed a resistor in series with the switch. In which case it will have a slight switch-on time depending on the value of the other resistor. This is particularly important to realize with TTL. Since a typical pull-up resistor of 4K7 with a pull down of around 200 ohms probably wont reach the lower threshold. Or worse. It will some times but not when you really need it.

The reason for pulling up with a resistor and not down is also typical of TTL Circuits. Though it is possible to turn the whole scheme upside down. As mentioned, TTL inherently pulls it self up. To pull it down may require 200

Ohms or less and this may be difficult to work with. Note that CMOS and it's high impedance nature neither pulls up nor down. and it can be easily used upside down. But it is more general to stick with the conventions of TTL. If you wished to have an output that is stable high and goes low when the button is pressed, run the output through a second invertor instead. Either a normal invertor from another package or a spare one from the Schmitt trigger package.

It should be noted that upon power-up a pulse will be generated from this debounce circuit. This is because the capacitor needs time to charge up initially. The Schmitt trigger's output will rise very quickly on power-up and then fall when the capacitor charges. Ironically this is the simplest way to reset a microprocessor or logic system. If the Microprocessor's reset pin is active high (Meaning it's reset when the pin is high) then the micro will be reset for a time period of the capacitor/resistor. When the cap charges the schmit trigger will go low and the micro will initialize and fetch it's first instruction. Likewise if this initial switching is a problem for further logic that it controls then a reset circuit could be used to inhibit the action of the logic until everything stabilizes. Just make sure that the capacitor/resistor of the reset circuit is something larger than the capacitor/resistor combos of the debounce circuit.

Now I mentioned that this is useful for switches up to 16. After that there are a couple of ICs from Nationa Semiconductor in their CMOS range. Known as MM74C922 and MM74C923. 16 key and 20 key encoders respectively. These single chips solutions can debounce up to 20 switches in a matrix with only a couple of external components. It outputs a 4 or 5 bit code representing the switch pressed and also a signal to notify further logic that a switch has been pressed. It can be configured to work in almost any situation. The DATA on these can be found at National's most excellent Web site. All data sheets are in Adobe's acrobat format.

If a very large array of switches is required then it's time to learn about microprocessors and that's a whole other topic of discussion.