Detecting a Switch Press

Momentary switches are useful when you want to get the computer's attention. For example, you might have a program that normally displays the current temperature and time, but switches to a setup routine when you press a switch. On a normally open pushbutton (momentary) switch, the contacts close when you press the switch, then open as you release it. In a normally closed momentary switch, the contacts open when you press, and close on release.

Switch bounce occurs because manual presses of mechanical switches tend to be sloppy. When you press a switch, the contacts normally bounce open and closed several times before they close positively, and bounce again as you lift your finger and the contacts open.

The computer has to be able to tell the difference between a bounce and a genuine switch press. Otherwise, each time you press a switch, and again when you release it, the computer will detect several rapid switch presses. One way to handle switch bounce is to ignore keypresses that are less than a certain length, usually around 10-20 milliseconds, with the exact value depending on the switch characteristics. Ignoring switch bounce is called switch debouncing.

Figure 7-3 shows a hardware debouncing circuit that uses a 74HC14 inverter with Schmitt-trigger input. The circuit generates a clean pulse when S1 is pressed, in spite of switch bounce that may occur.

Point B, the inverter's input, is normally low, and point C, the inverter's output, is normally high. When S1 is pressed, C1 discharges slowly through R2 and the switch contacts. If switch bounce occurs, the voltage at the inverter's input can't change rapidly enough to affect the logic state of the input. Only when the switch remains closed for around 50 milliseconds does point B go high enough to cause point C to switch low.

In a similar way, when the switch contacts open, C1 charges slowly though R1 and R2, and pin C goes high again only when the contacts have remained closed for around 50 milliseconds. The Schmitt-trigger input ensures that the output pulse is clean even if the input changes slowly.

As with software debouncing, you can experiment to find the minimum values that prevent unwanted interrupts due to switch bounce. The debounce time increases as you increase the values of C1 and R2.

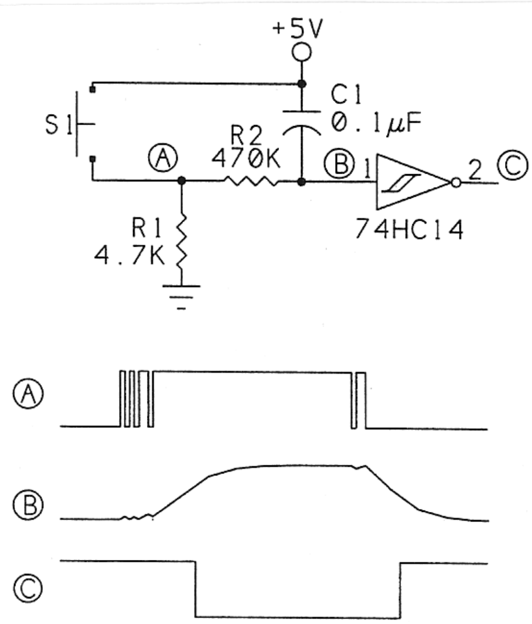

Figure 7-3. A hardware debouncing circuit, using a 74HC14 Schmitt-trigger inverter.