**Experiment 7**

**DEBOUNCING OF MECHANICAL SWITCHES**

**4-BIT SIMPLE CALCULATOR**

**Introduction:**

Push-button switches, toggle switches, and electro-mechanical relays all have one thing in common: contacts. It's the metal contacts that make and break the circuit and carry the current in switches and relays. Because they are metal, contacts have mass. And since at least one of the contacts is on a movable strip of metal, it has springiness. Since contacts are designed to open and close quickly, there is little resistance (damping) to their movement. A simple hardware debounce circuit for a momentary N.O. push-button switch is show in As you can see, it uses an RC time constant to swamp out the bounce. If you multiply the resistance value by the capacitance value you get the RC time constant. You pick R and C so that RC is longer than the expected bounce time. An RC value of about 0.1 seconds is typical. Note the use of a buffer after the switch to produce a sharp high-to-low transition. And remember that the time delay also means that you have to wait before you push the switch again. If you press it again too soon it will not generate another signal.If you're the one developing the digital "box", then you can debounce in software. Usually, the switch or relay connected to the computer will generate an interrupt when the contacts are activated. The interrupt will cause a subroutine (interrupt service routine) to be called. A typical debounce routine is given below in a sort of generic assembly language.The idea is that as soon as the switch is activated the Debounce Routine (DR) is called. The DR calls another subroutine called DELAY which just kills time long enough to allow the contacts to stop bouncing. At that point the DR checks to see if the contacts are still activated If so, the DR waits for the contacts to clear. If the contacts are clear, DR calls DELAY one more time to allow for bounce on contact-release before finishing.A debounce routine must be tuned to your application; the one above may not work for everything. Also, the programmer should be aware that switches and relays can lose some of their springiness as they age. That can cause the time it takes for contacts to stop bouncing to increase with time. So the debounce code that worked fine when the keyboard was new might not work a year or two later. Consult the switch manufacturer for data on worst-case bounce times

**Objective:**

● To understand the phenomena of bouncing in mechanical switches and its solution

● To be able to use mechanical switches in making a simple application

**Procedure:**

HARDWARE SETUP

● There are 8-SPDT switches on the trainer. Make initial state of all switches at 0V.

● Input is given through these SPDT switches. There are 4 data switches and 1 control,

acting like an enter key, connected at P1.

● User prepares a 4-bit data over the binary switches, and then sets the control switch to

logic 1 and then back to logic 0 (0-1-0).

● After this action controller reads the data at input port P1, performs desired operation

and outputs the result to port P2.

SOFTWARE

Your program must be capable of doing the following jobs:

● Configure port P1 as input port

● Write a sub routine to debounce the switch connected at P1.4

● Read data at P1 when SPDT at P1.4 is made 1 from zero and then back to zero

● Repeat the previous step three times, twice for the operands and once for the operator

● Decode the operator, perform desired operation and show result at LEDs connected at

P2

● Get ready to take input again

**Applications:**

Latches are used to keep the conditions of the bits to encode binary numbers.Feeding the signal into a logic gate or a microcontroller sends multiple key press signals which is not what you want so you have to ignore the bouncing signal - this is known as debouncing the switch.

**Issues:**

No issues faced.

**Conclusions:**

In this lab we handle bouncing by software and harware.we also learn how to interface switch by use of microcontroller 80C52.

**Post Lab:**

As f=100 mHz then T=1/f so T=10s

Ton+Toff=10 --1

From deuty cycle we have

Duty cycle =(Ton/Ton+Toff)\*100

If duty cycle is 10% then

0.1=Ton/(Ton+Toff)

0.1Ton+0.1Toff=Ton

0.1Toff=0.9Ton

0.9Ton-0.1Toff --2

Solving eq 1 and 2

Toff=9s and Ton=1s

org 0X0

main:

clr p3.3

jb p2.4,$

call debounce

mov a,p2

anl a,#0fh

mov r1,a

setb p3.3

call delay2

clr p3.3

call delay3

jmp main

debounce:

call delay

jnb p2.4,$

call delay

ret

delay2:

mov a,#0

add a,r1

mov b,#7

mul ab

jz en

mov r3,a

lp1: mov r4,#255

lp2: mov r5,#255

lp3: djnz r5,lp3

djnz r4,lp2

djnz r3,lp1

en:

ret

delay3:

mov a,#10

clr cy

subb a,r1

mov b,#7

mul ab

jz ed

mov r0,a

lp7: mov r6,#255

lp8: mov r7,#255

lp9: djnz r7,lp9

djnz r6,lp8

djnz r0,lp7

ed:

ret

delay:

mov r4,#4

lp5: mov r5,#255

lp6: djnz r5,lp6

djnz r4,lp5

ret

end



