**Experiment 9**

**BCD Seven Segment Display**

**Introduction:**

A seven-segment display is a form of electronic display device for displaying decimal numerals that is an alternative to the more complex dot matrix displays. Seven-segment displays are widely used in digital clocks, electronic meters, basic calculators, and other electronic devices that display numerical information.To display numbers ranging from 0-9 we use BCD seven segment display. BCD displays are of two types. First is common anode and second is common cathode. In common anode, the anodes of all seven diodes in BCD are connected to the connected to +Vcc or to +5V and cathodes are provided to the user. While in common cathode, the cathodes of all the diodes are connected to the ground and the anodes are provided to the user.

**Objective:**

To learn interfacing of BCD seven segment display with 8051 microcontroller and displaying numbers using the BCD display.

**Procedure:**

We started by taking an input from the user on port 3. The number was masked and lower and upper nibbles were separated. They were treated as two separate numbers. The numbers were added or multiplied depending on the value of R0 which was hard coded. If value of r0 was 0, the two numbers were added and if it’s value was 1, the numbers were multiplied. The result was finally displayed on a seven segment BCD display.

**Applications:**

Applications of Seven Segment Displays are these displays are commonly used in timers, clock radios, digital clocks, calculators and wristwatches. These devices can also be found in speedometers, motor-vehicle odometers, and radio frequency indicators..

**Issues:**

No issues faced while performing the experiment.

**Conclusions:**

Thus we conclude that we can interface 80C51 with BCD seven segment display. So we can display numbers ranging from 0-9 using BCD seven segment display. The task we perform is displaying a 4-digit number on an output device using the 89C52 micro-controller.

**Post Lab:**

org 0x0

mov dptr,#list

back:call path

mov p0,a

call path

mov p1,a

call path

mov p2,a

call path

mov p3,a

sjmp back

path:

clr a

movc a,@a+dptr

clr cy

subb a,#30h

inc dptr

ret

list: db "1234","5678","9513","7412",0

end