Familiarization with 8051/8052 Microcontroller

1) Objectives

To enable students to write assembly language code for the 8051/8052 microcontroller capable of:
(a) Data manipulation
(b) Arithmetic and logic operations
(c) Looping and branching techniques
(d) Subroutine calls

2) Equipment Required

(a) Hardware:
  (i) 8051 or 8052 microcontroller development board
  (ii) Jumper cables

(b) Simulation Software:
  (i) KEIL µVision – Embedded development tool
  (ii) Proteus Design Suite – Professional PCB layout, circuit design and simulation tool

(c) In-System Programming (ISP) Software:
  (i) ProgISP – An in-system-programmable tool to load HEX files in to microcontroller

(d) Device Drivers:
  (i) LibUSB – Application controlling data transfer to/from USB devices

3) Lab Exercises

(A) Write code to add the numbers 897F9AH to 34BC48H and save the result in internal RAM starting at 40H.
- ➢ The result should be displayed continuously on the LEDs of the development board starting from least significant byte with an appropriate timing interval between each byte. Use port zero (P0) of the microcontroller to interface with LEDs.

(B) Implement a subroutine that replaces the SWAP instruction using rotate right instructions.

(i) Test your program on the contents of the accumulator when it contains the number 6BH.
- ➢ The original number and the result should be displayed continuously on the LEDs of the development board one-by-one with an appropriate timing interval between them. Use port zero (P0) of the microcontroller to interface with LEDs.

(C) Multiply the data in RAM location 22H by the data in RAM location 15H and put the result in RAM locations 19H (low byte) and 1AH (high byte). Data in 22H should be FFH and data in 15H should be DEH.

(i) Use **looping** and **successive addition** technique
- ➢ The product (high byte and low byte) should be displayed continuously on the LEDs of the development board one-by-one with an appropriate timing interval between them. Use port zero (P0) of the microcontroller to interface with LEDs.

(D) Divide the data in RAM location 3EH by the number 12H; put the quotient in R4 and the remainder in R5. Data in 3EH should be AFH.

(i) Use **looping** and **successive subtraction** technique
- ➢ The quotient and remainder should be displayed continuously on the LEDs of the development board one-by-one with an appropriate timing interval between them. Use port zero (P0) of the microcontroller to interface with LEDs.

Familiarization with 8051/8052 Microcontroller

(E) Store ten hexadecimal numbers in internal RAM starting from memory location 50H. The list of numbers to be used is: D6H, F2H, E4H, A8H, CEH, B9H, FAH, AEH, BAH, CCH.

(i) Implement a **subroutine** that extracts both the **smallest** and **largest** numbers from the stored numbers.
- The smallest and largest numbers should be displayed continuously on the LEDs of the development board one-by-one with an appropriate timing interval between them. Use port zero (P0) of the microcontroller to interface with LEDs.

(F) Store ten hexadecimal numbers in internal RAM starting from memory location 60H. The list of numbers to be used is: A5H, FDH, 67H, 42H, DFH, 9AH, 84H, 1BH, C7H, 31H.

(i) Implement a **subroutine** that orders the numbers in **ascending** order using **bubble sort algorithm**.
- The sorted list of numbers should be displayed continuously on the LEDs of the development board one-by-one with an appropriate timing interval between them. Use port zero (P0) of the microcontroller to interface with LEDs.

(ii) Implement a **subroutine** that orders the numbers in **descending** order using **selection sort algorithm**.
- The sorted list of numbers should be displayed continuously on the LEDs of the development board one-by-one with an appropriate timing interval between them. Use port zero (P0) of the microcontroller to interface with LEDs.

(G) Store numbers from 00H to 20H in internal RAM starting from memory location 40H.

(i) Implement a **subroutine** that extracts only the **prime numbers**
- The prime numbers should be displayed continuously on the LEDs of the development board one-by-one with an appropriate timing interval between them. Use port zero (P0) of the microcontroller to interface with LEDs.

(H) Find the **factorial** of a number stored in R3.

(i) The value in R3 could be any number in the range from 00H to 05H.
(ii) Implement a **subroutine** that calculates the factorial
(iii) The factorial needs to be represented in both **hexadecimal** and **decimal** formats
- The factorial in hexadecimal and decimal formats should be displayed continuously on the LEDs of the development board one-by-one with an appropriate timing interval between them. Use port zero (P0) of the microcontroller to interface with LEDs.

(I) Store ten hexadecimal numbers in internal RAM starting from memory location 55H. The list of numbers to be used is: 25H, 3DH, F7H, 13H, C9H, 4EH, 62H, 77H, B8H, EBH.

(i) Implement a subroutine that searches for the occurrence of the binary sequence $(11)_2$ within a number amongst the given list of numbers.
(ii) Only those numbers that contain the specified binary sequence should be displayed continuously on the LEDs of the development board one-by-one with an appropriate timing interval between them. Use port zero (P0) of the microcontroller to interface with LEDs.