Embedded System Lab: Session First - Microcontroller

LAB – 1: Familiarization with 8051/8052 Microcontroller

Objectives

To enable us to write assembly language code for the 8051/8052 micro-controller capable of:

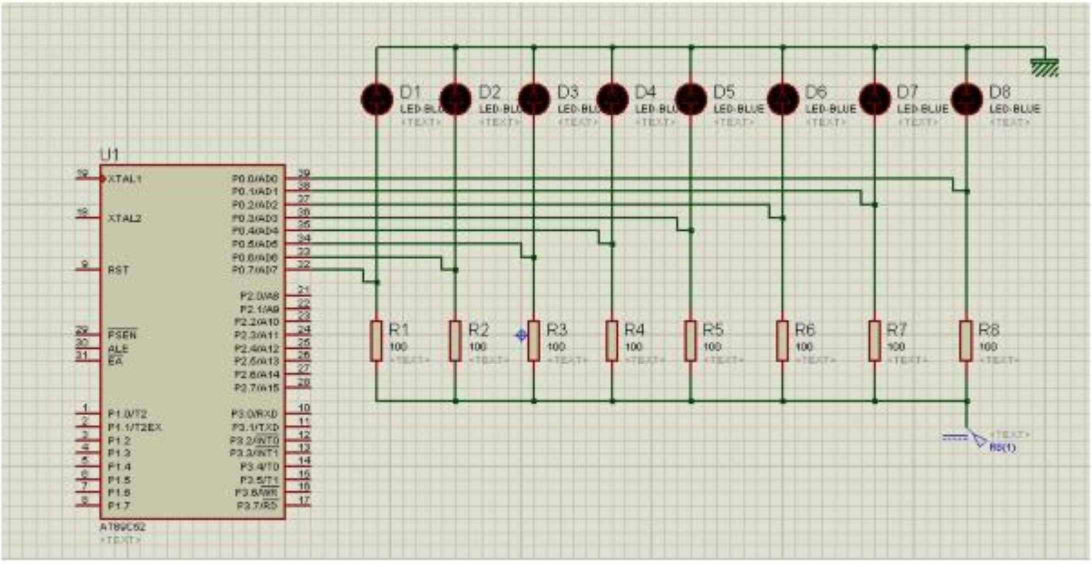
- Data Manipulation
- Arithmetic and logic operations
- Looping and branching techniques
- Subroutine calls

Equipment Required

- Hardware: 8051 or 8052 micro-controller development board, Jumper cables
- Simulation Software: KEIL, Vision-Embedded development tool, Proteus Design Suite – Professional PCB layout, circuit design and simulation tool
- In-System Programming (ISP) Software: ProgISP – An in-system-programmable tool to load HEX files in to micro-controller
- Device Drivers: LibUSB – Application controlling data transfer to/from USB devices

Problems

The circuit diagram, consisting of micro-controller AT89C52, LEDs connected to Port 0 and Pull-up resistors, used to confirm the code written in assembly language and c language is shown below:



1. Write code to add the numbers 897F9AH and 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 micro-controller to interface with LEDs.
2. Implement a subroutine that replaces the SWAP instruction using rotate right instructions. Test your program on the contents of the accumulator when it contains the number 6BH.
3. Multiply, by using looping and successive addition technique, 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.
4. Divide, by using looping and successive subtraction technique, the data in RAM location 3EH by the number 12H; put the quotient in R4 and remainder in R5. Data in 3EH should be AFH.
5. 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. Implement a subroutine that extracts both the smallest and largest numbers from the stored numbers.
6. 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. Implement a subroutine that orders the numbers in ascending order using bubble or any other sort algorithm and implement s subroutine that order the numbers in descending order using selection sort algorithm.
7. Store numbers from 00H to 20H in internal RAM starting from memory location 40H. Implement a subroutine that extracts only the prime numbers.
8. Find the factorial of a number stored in R3. The value in R3 could be any number in the range from 00H to 05H. Implement a subroutine that calculates the factorial. The factorial needs to be represented in both hexadecimal and decimal formats.