Microprocessor based System Design Laboratory (Gy)

Department of Computer Science and Technology, IIEST Shibpur

Date: August 23, 2024

Experiment No. 10: 8085 Assembly Language Programming – Writing Interrupt Service Routines for hardware Interrupts (along with using Monitor Routines for IO Operation)

Prerequisite: Understanding of the last laboratory session [August 23, 2024]
Objective: The SDK being used in the laboratory has used the hardware interrupt RST 5.5 of its 8085 processor for interfacing its keypad. That is why in the last laboratory class you had to unmask RST 5.5 (using SIM instruction) and enable the interrupts (using EI instruction) before calling monitor functions which need key-press (like, **RDKB**, etc). **In todays laboratory session you have to work with the other hardware interrupt lines of the 8085 processor of the SDK.**

Each of you have to write 8085 programs as specified below **in the format shown in the earlier classes.** For each program, you write the assembly code first, and then translate it to machine language using the Instruction Set Table provided to you. You can now load, execute and test the program.

Get the program, that you have written in your notebook, signed by your teacher and upload it against the corresponding assignment in the Google Classroom.

Assignments

[Consult the **2-page Instruction Set** and **16-page Instruction Set Reference Encyclopedia** for choosing appropriate Instructions while writing programs]

You will require the monitor functions documented in the **"Useful software routines" section (Page 46 to 58) in the user manual of the kit.**

1. Read the user manual of the SDK to identify whether you can use the hardware interrupt lines RST 6.5 and RST 7.5 of the 8085 processor of your SDK. Using such an interrupt lines involves (i)"generating interrupts" at those pins of the processor and (ii) writing interrupt service routines at the corresponding locations of the main memory (8 x 6.5 = **0034**H for RST 6.5 and 8 x 7.5 = **003C**H for RST 7.5. Please identify whether the above are feasible in your SDK (without damaging it). Write a small report on this and upload in your Google Classroom.
2. Make arrangement such that the count of RST 6.5 interrupt (that is, how many times RST 6.5 interrupt has reached the 8085 processor) is displayed in the data part of the display section.
3. Make arrangement such that every time RST 7.5 interrupt reaches the 8085 processor, the last two digits of the Examination Roll Number of the next member of your group is displayed in the data part of the display section. That is, if your group has 3 members, the last 2 digits of their Examination Roll Number will be displayed in rotation, as RST 7.5 is generated again and again.