

LAB 05: Interrupt a Running Program

The purpose of this assignment is to show how interrupt-driven Input/Output can interrupt a program that is running, execute the interrupt service routine, and return to the interrupted program, picking up exactly where it left off (just as if nothing had happened). In this assignment, we will use the **Keyboard** as the input device for interrupting the running program.

Your Mission:

1. Write the user program described below.
2. Write the keyboard interrupt service routine described below.

Details:

The assignment consists of three parts, but **you only need to do the first two parts.**

A. The user program

Your user program, which starts at **x3000**, will continually (i.e. in an infinite loop) print the word “ICS2020” like:

ICS2020 ICS2020 ICS2020 ICS2020 ICS2020 ICS2020

To ensure the output on the screen is not too fast to be seen by the naked eye, the user program should include a piece of code that will count down from 2500 (or any other numbers) **after each word is output on the screen.**

A simple way to do this is with the following subroutine DELAY:

<i>DELAY</i>	<i>ST R1, SaveR1</i>
	<i>LD R1, COUNT</i>
<i>REP</i>	<i>ADD R1, R1, #-1</i>
	<i>BRp REP</i>
	<i>LD R1, SaveR1</i>
	<i>RET</i>
<i>COUNT</i>	<i>.FILL x7FFF</i>
<i>SaveR1</i>	<i>.BLKW #1</i>

B. The keyboard interrupt service routine

The keyboard interrupt service routine, which starts at **x1000**, will examine the key typed to see if it is a **decimal digit**.

If the character typed is **NOT** a decimal digit the interrupt service routine will, starting on a new line on the screen, print "<the input character> is not a decimal digit." For example, if the input key is '#', the interrupt service routine will print:

is not a decimal digit.

The service routine would then print a line feed (x0A) to the screen, and finally terminate with an RTI.

If the character typed **IS** a decimal digit, the interrupt service routine will, starting on a new line on the screen, print "<the input character> is a decimal digit.". If the input key is '4', the interrupt service routine will print:

4 is a decimal digit

The service routine would then print a line feed (x0A) to the screen, and finally terminate with an RTI.

Hint: Don't forget to save and restore any registers that you use in the interrupt service routine.

C. The operating system enabling code

Unfortunately, we have not installed Windows or Linux on the LC-3, so we provide you with [STARTER CODE](#) (in attachment) that enables interrupts. **You MUST use the starter code for this assignment.** The locations to write the user program and interrupt service routine are marked with comments.

The starter code does the following:

1. Initializes the interrupt vector table with the starting address of the interrupt service routine. The keyboard interrupt vector is **x80** and the interrupt vector table begins at memory location **x0100**. The keyboard interrupt service routine begins at **x1000**. Therefore, we must initialize

memory location **x0180** with the value **x1000**.

2. Sets bit 14 of the KBSR to enable interrupts.
3. Pushes a PSR and PC to the system stack so that it can jump to the user program at **x3000** using an RTI instruction.

Example:

ICS2020 ICS2020 ICS2020 ICS2020

h is not a decimal digit.

//Input character 'h'

ICS2020 ICS2020 ICS2020 ICS2020 ICS2020

4 is a decimal digit

//Input character '4'

ICS2020 ICS2020 ICS2020 ICS2020 ICS2020 ICS2020...

Notes and Suggestions:

1. Since the interrupt can be triggered at any point, the output of the interrupt service routine may show up anywhere.
2. Since your user program contains an infinite loop, you will have to press the "**Pause**" button in the simulator if you wish to stop the program.
3. Unlike previous labs, the PC will be initialized to **x800** for this assignment because the first code that is executed will be in the operating system.
4. **Please make sure that the "Ignore privileged mode" switch is OFF. (Default configuration is OFF in LC-3 simulator)**

Additional Requirements:

If you don't comply with these requirements, the lab may be counted as an invalid work.

1. The report shall contain at least 3 parts: How do you work out the algorithm? How do you write the program? And how do you design your own test cases to ensure the program works fine?
2. Save your report in pdf format and name it such as **report.pdf**.
3. Your program should be renamed to **interrupt.asm**.
4. Put all above in a directory named after your student number and pack it

(e.g. PB07210340_张海博_LAB05.zip).