**Computer Organization Fall 2024**

**HKBU-BNU United International College**

**Lab 9: Input and Output Programming**

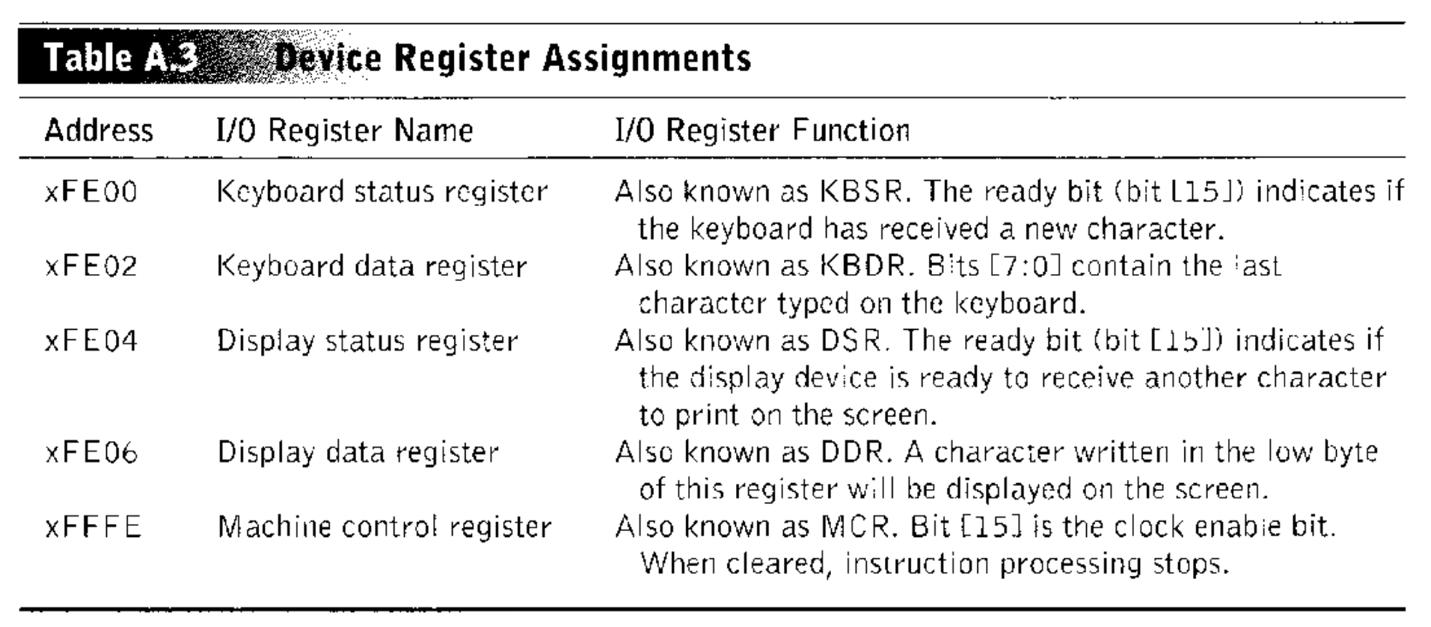
# Lab Objective

To learn how to input and output information in the LC-3 assembly language.

**Introduction**

The LC-3 ISA does not have special I/O instructions. Instead, it uses **memory-mapped I/O** so that data movement instructions can be used for I/O operations. For example, a Load instruction (LD/LDR/LDI/LEA), in which the source address is that of an input device register, is an input instruction. Similarly, a Store instruction (ST/STR/STI) in which the destination address is that of an output device register is an output instruction.

In memory-mapped I/O, the I/O device registers are mapped to a set of memory addresses that are allocated to I/O device registers. In the LC-3, address x0000 to xFDFF are allocated to memory locations. Address **xFE00 to xFFFF** are reserved for input/output device registers as listed in Table A.3.

****

The CPU is usually much faster than the I/O devices. In order to coordinate the interactions between the fast CPU and the slow I/O devices, we need to at least have a 1-bit status register (KBSR or DSR).

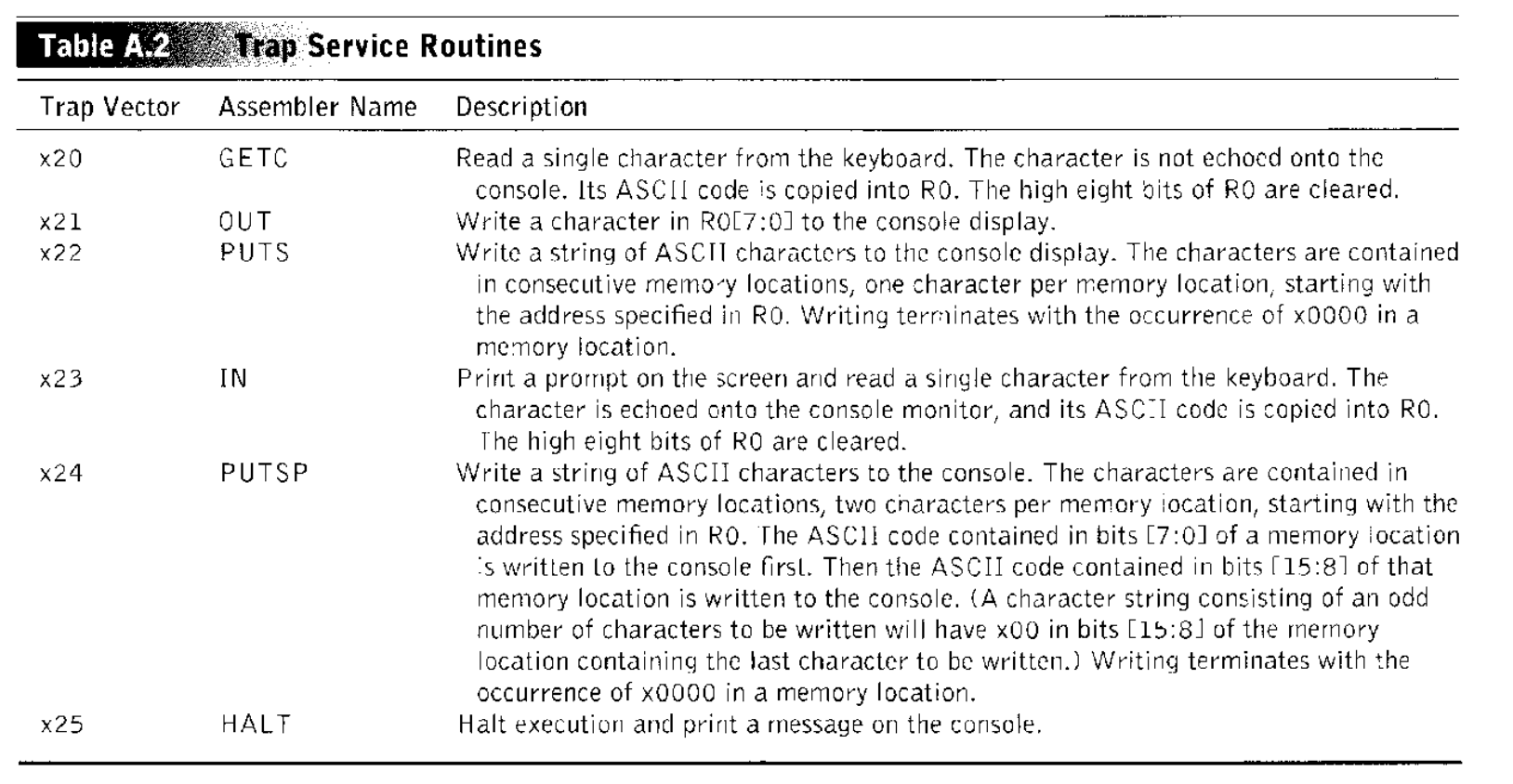
# Based on who is controlling the information transfer between the CPU and the I/O devices, there are two methods of I/O mechanisms: interrupt-driven I/O and polling-based I/O. In the textbook, section 8.2.2 describes how the polling method works and section 8.5 explains how interrupt-driven I/O works.

# When we type at the keyboard, it is helpful to know exactly what characters we have typed. We can get this echo capability easily (without any sophisticated electronics) by the following code in Figure 1. The key typed at the keyboard is displayed on the monitor.

# The LC-3 has also provided a TRAP instruction for I/O operations. The trap vector identifies the service routine the user program wants the operating system to perform. In the following example, the trap vector is x23.

# 

The LC-3 was designed to have up to 256 service routines. But currently it has only defined totally 6 operating system service routines as shown in Table A.2



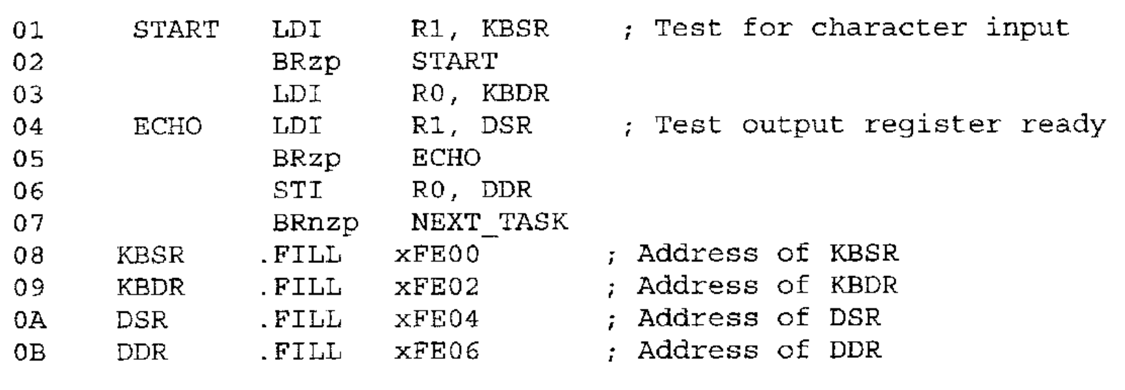
The LC-3 OS has maintained a **Trap Vector Table** that is stored in memory locations x0000 to x00FF. The table is also referred to as the System Control Block. It contains the starting addresses of these 256 service routines. So, you may say that it is **a table of memory addresses**.

The trap routine is invoked by the **TRAP** instruction. At the end of the trap service routine, the **RET (JMP R7)** instruction provides a return to the user program.

# Lab Instruction

**Step 1: Download the LC-3 Simulator for Windows and unzip it to your local disk.**

**Step 2: Convert the LC-3 assembly code fragment in Figure 1 to a complete assembly language program, assemble it and run it in the simulator to verify its function.**



**Figure 1**

# Lab Exercise

Develop a new TRAP Service Routine, TRAP x26, to print “Hello, world!” on the screen. Write a program to call the trap routine defined by you.

# Submission

Zip and Upload your source code (.asm) and a word file describing how you define the trap routine. Show the entry of Trap 26 in the Trap Vector Table and the code in the memory. Name the zip file with your student ID.