LAB6:FACTORIAL

郑悟强 PB22051082

2023.12.23

1 Purpose

• 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.

• After receiving a keyboard interrupt, you need to echo the input character and determine whether it is a decimal number. If the input is indeed a decimal number, you should store its value in memory and pass it as the variable N to the Factorial subroutine for calculating the result.

2 Principles

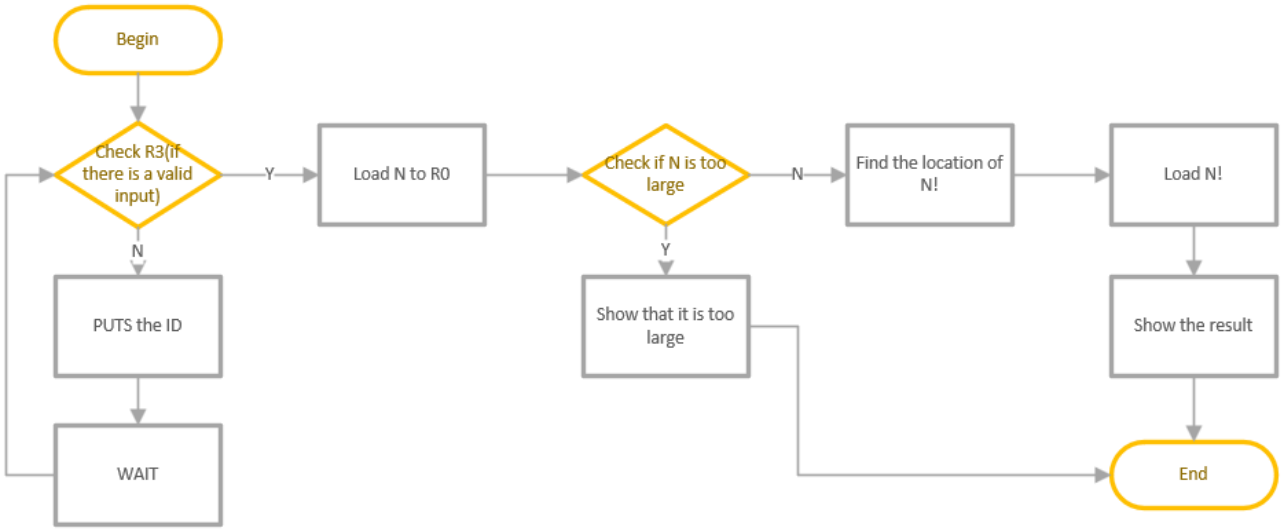
2.1 user program

user program's task:

Print my ID again and again, until being interrupted by keyboard input.

If the keyboard has inputted a valid character, stop printing and output the factorial.

flow chart:

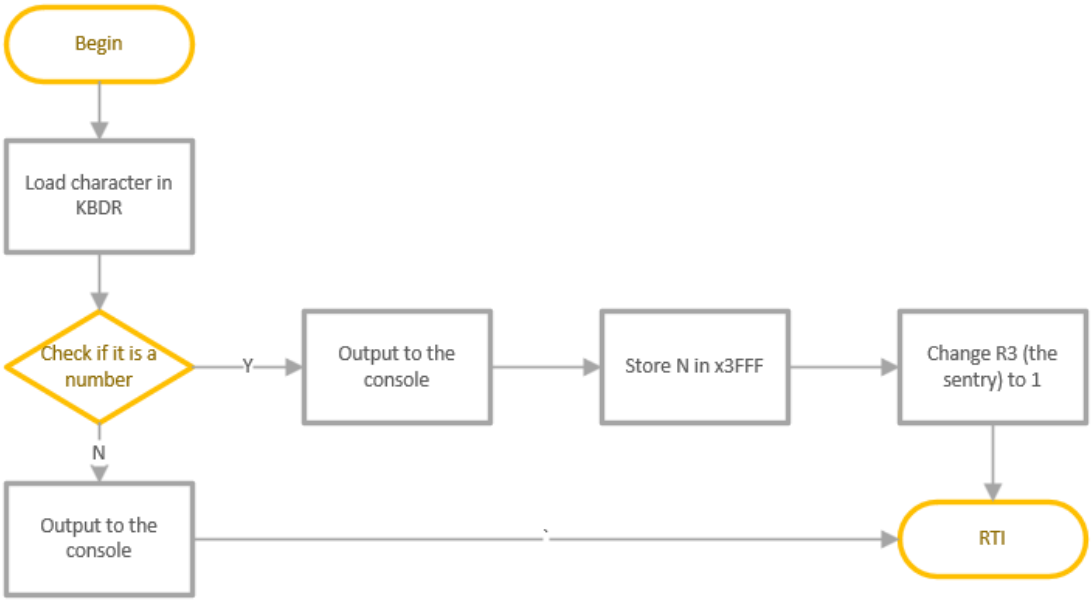


2.2 interrupt service

interrupt service's task:

Store the character, and check if it is a number, then stores it in *x3FFF*

flow chart:



2.3 lookup table to calculate N!

lookup table:

```
            .ORIG   x3FFF
            ; *** Begin factorial data here ***
FACT_N      .FILL   xFFFF           ;the location to store N
            .FILL   x4008   ;0!
            .FILL   x400A   ;1!
            .FILL   x400C   ;2!
            .FILL   x400E   ;3!
            .FILL   x4010   ;4!
            .FILL   x4013   ;5!
            .FILL   x4017   ;6!
            .FILL   x401B   ;7!
LOCA_0      .STRINGZ    "1"
LOCA_1      .STRINGZ    "1"
LOCA_2      .STRINGZ    "2"
LOCA_3      .STRINGZ    "6"
LOCA_4      .STRINGZ    "24"
LOCA_5      .STRINGZ    "120"
LOCA_6      .STRINGZ    "720"
LOCA_7      .STRINGZ    "5040"
            ; *** End factorial data here ***
            .END
```

how to use this:

```
LD R1, N_LOCA
LDR R0, R1, #0    ;store N in R0
ADD R1, R0, R1    ;R1 stores the address of the address of (N-1)!
LDR R2, R1, #1    ;R2 stores the address of N!
```

3 Procedure

3.1 store the parameters

- **Question: how to store some data between user program and interrupt service**

solution1: use the stack

For the first time, I tried to use the stack to store the data in registers. But the thing is, the two program are in different area to use different stacks, (user program uses *user stack* and interrupt service uses *supervisor stack*), so R6 will represents different top pointer. It will easily lcover the data or store the data in a wrong place. Besides, RTI instruction will also pop the stack for two times, the location PC comes back will also make a mistake.

solution2: store in the memory

Without the stack, I choose to use the memory to store the parameters. Using the lookup table, I can just store the N we input and easily find N! in user program. It will cost a little memory location but save a lot of time and the code will be much more stable.

3.2 the gap between number and its ASCII code

- **Question: there is a gap between the number's data and its ASCII code**

When we need to output N to the console, we need R0 to store N's ASCII code. However, when we need to calculate N (*e.g compare if N is too large, use N to find the N! in the lookup table*), we just need the data of the number.

solution: We can just store the gap *#-48* before. If we need to output N, just use TRAP instruction. When we need to calculate , R0 <- N + ASCIIgap.

4 Results

When we start executing, the program will output the student ID again and again.

Console (click to focus)

```
PB22051082 PB22051082 PB22051082 PB22051082 PB22051082
PB22051082 PB22051082 PB22051082 PB22051082 PB22051082
PB22051082 PB22051082 PB22051082 PB22051082 PB22051082
PB22051082 PB22051082 PB22051082 PB22051082 PB22051082
PB22051082 PB22051082 PB22051082 PB22051082 PB22051082
PB22051082 PB22051082 PB22051082 PB22051082 PB22051082
PB22051082 PB22051082 PB22051082 PB22051082 PB22051082
PB22051082 PB22051082 PB22051082 PB22051082 PB22051082
PB22051082 PB22051082 PB22051082 PB22051082 PB22051082
PB22051082 PB22051082 PB22051082 PB22051082 PB22051082
PB22051082 PB22051082 PB22051082 PB22051082 PB22051082
PB22051082 PB22051082 PB22051082 PB22051082 PB22051082
PB22051082 PB22051082 PB22051082 PB22051082 PB22051082
PB22051082 PB22051082 PB22051082 PB22051082 PB22051082
PB22051082 PB22051082 PB22051082
```

If we kick the keyboard and input a character but not a number.

Console (click to focus)

```
PB22051082 PB22051082 PB22051082 PB22051082 PB22051082
PB22051082 PB22051082 PB22051082 PB22051082 PB22051082
PB22051082 PB22051082
w is not a decimal digit.
 PB22051082 PB22051082 PB22051082 PB22051082 PB22051082
PB22051082 PB22051082 PB22051082 PB22051082 PB22051082
PB22051082 PB22051082 PB22051082 PB22051082 PB22051082
PB22051082 PB22051082 PB22051082 PB22051082 PB22051082
PB22051082 PB22051082 PB22051082 PB22051082 PB22051082
PB22051082 PB22051082 PB22051082 PB22051082 PB22051082
PB22051082 PB22051082 PB22051082 PB22051082 PB22051082
PB22051082 PB22051082 PB22051082 PB22051082 PB22051082
PB22051082
T is not a decimal digit.
 PB22051082 PB22051082 PB22051082 PB22051082 PB22051082
```

If we kick the keyboard and input a number but larger than 7.

Console (click to focus)

```
PB22051082 PB22051082 PB22051082 PB22051082 PB22051082
PB22051082 PB22051082 PB22051082 PB22051082 PB22051082
PB22051082 PB22051082 PB22051082 PB22051082 PB22051082
PB22051082 PB22051082 PB22051082 PB22051082 PB22051082
PB22051082 PB22051082 PB22051082 PB22051082 PB22051082
PB22051082 PB22051082 PB22051082 PB22051082 PB22051082
PB22051082 PB22051082 PB22051082 PB22051082 PB22051082
PB22051082 PB22051082 PB22051082 PB22051082 PB22051082
PB22051082
8 is a decimal digit.
8 is too large for LC3

--- Halting the LC-3 ---
```

If we input a number less than or euqal to 7.

```
Console (click to focus)
PB22051082 PB22051082 PB22051082 PB22051082 PB22051082
PB22051082 PB22051082 PB22051082 PB22051082 PB22051082
PB22051082 PB22051082 PB22051082 PB22051082 PB22051082
PB22051082 PB22051082 PB22051082 PB22051082 PB22051082
PB22051082 PB22051082 PB22051082 PB22051082 PB22051082
PB22051082 PB22051082 PB22051082 PB22051082 PB22051082
PB22051082 PB22051082 PB22051082 PB22051082 PB22051082
PB22051082 PB22051082 PB22051082 PB22051082 PB22051082
PB220510
6 is a decimal digit.
6! = 720

--- Halting the LC-3 ---
```