Lab3-Report

1 Algorithm

MVC Design Mode

I plan to use MVC, that is Model, View, and Control.

In my program, the user code(start at x3000) is used to View, and the interrupt service routine is used to control and update model.

In the View part, we can simply use two registers, one served as a line counter(the number of characters in one line), and the other containing the 'task counter'. It is convenient to iterate the output module, counting down the line counter. When the counter comes to 0, print `Enter` on the screen, reload it with 40, and continue iterating.

In the Control and Model part, which is the interrupt service routine, we detect what char is input.

The Control function (to detect whether keyboard wants service and interrupt the program) acutually has been accomplished by the Interrupt-Driven I/O itself, so we don't need to take care of it too much.

Updating Model based on input is our main task in the service routine.There are 3 conditions: `Enter`, `0-9`, or the other chars.

- The most simpliest condition is `Enter`, where we only need to judge whether task counter is 0.If not, decrement the counter and end the routine, otherwise end the routine directly.
- Then `0-9`.Since LC-3 only has ADD, NOT, AND operations in ALU, we must operate twice to complete judgement. That is, if `KBDR - '0'` is positive, and `KBDR - '0' - 10` is negative, the input is `0-9`. If so, reset the task counter and end the routine.
- Finally the other characters. Similar to user code, we use iteration to print a single line with 40 input chars.Add `Enter` before and after the line, and end the subroutine.

2 Codes & Comments

```
; Part of OS Booting Code
            LD      R0, INTEN
            STI     R0, KBSR            ; set interrupt enable bit to 1
;
            LD      R0, StartAdd
            STI     R0, INTV            ; set interrupt vector table, x0180 contains
x0800
;
INTEN       .FILL x6000                 ; INT.EN bit MASK
KBSR        .FILL xFE00
StartAdd    .FILL x0800                 ; the start address of the INT service
subroutine
INTV        .FILL x0180                 ; the keyboard INTV
```

```
;
; User Code
            LD      R3, ASCII0
            ADD     R3, R3, #7          ; R3 contains the char, initialize to 7
            LD      R2, MAXCHAR         ; R2 contains the char counter
;
AGAIN       BRz     Newline
            JSR     DELAY
            ADD     R0, R3, #0
            OUT                         ; output the current char to monitor
            ADD     R2, R2, #-1         ; counter--
            BR      Loop
Newline     LD      R0, Enter
            OUT                         ; output '\n' to monitor
            LD      R2, MAXCHAR         ; reset counter to 40
Loop        BR      AGAIN
; Delay is omitted here
Enter       .FILL   x000A
ASCII0      .FILL   #48
MAXCHAR     .FILL   #40
;
; Interrupt Service Routine
            ; PUSH R0,R1,R2 to SS, omitted here
            LDI     R0, KBDR            ; R0 uses for load the char keyboard input
            LD      R1, ASCEnter
            NOT     R1, R1
            ADD     R1, R1, #1          ; R1 <- '\n'
            LD      R2, CHAR0
            NOT     R2, R2
            ADD     R2, R2, #1          ; R2 <- -'0'
;
            ADD     R1, R1, R0
            BRz     DECREASE            ; KBDR is '\n'
            ADD     R1, R2, R0
            BRzp    IfDigit             ; to test if KBDR is '0'~'9'
;
Other       ADD     R1, R0, #0          ; KBDR is neither '\n' nor digit
            LD      R2, MAXNUM
            LD      R0, ASCEnter
            OUT
            ADD     R0, R1, #0
L2          ST      R7, SaveR7          ; Noting: SaveR7
            JSR     Delay2
            LD      R7, SaveR7
            OUT
            ADD     R2, R2, #-1
            BRp     L2
            LD      R0, ASCenter
            OUT
```

```
            BR      EndINT
;
IfDigit     ADD     R1, R1, #-10
            BRzp    Other
            ADD     R3, R0, #0          ; KBDR is digit
            BR      EndINT

DECREASE    ADD     R1, R3, R2          ; test if current task counter is '0'
            BRz     EndINT
            ADD     R3, R3, #-1
            BR      EndINT

EndINT      ; POP R0,R1,R2 to SS, omitted here
            RTI
; Delay is omitted here
;
SaveR7      .BLKW   1
KBDR        .FILL   xFE02
ASCenter    .FILL   x000A
CHAR0       .FILL   #48
MAXNUM      .FILL   #40
```

3 TA's Questions

After running the OS booting code, what value is contained in R6?

In the booting code, R6 is the pointer to SS. After RTI, R6 points to US.

I simply thought that R6 contains xFE00. But the TA told me that Save_USP is randomized, so it's no idea what value is contained in R6.