

Lab4-Report

3210105321 郑浩博

Algorithm Explanation

Firstly we write our main program from x3000. We put our initial-char 'a' and initial-position 5 into x4000 and x4001.:

```
.ORIG x3000
LD R0,Initchar      ;;R0 <- x61
STI R0,Charuser      ;;MEM[x4000] <- R0
LD R1,Initpos        ;;R1 <- x5
STI R1,Posuser       ;;MEM[x4001] <- R1
```

Then we output head-dots and characters and tail-dots:

```
Loop      LDI R1,Posuser      ;;R1 <- MEM[x4001](Position)
          ADD R2,R1,#0        ;;R2 <- R1
          LD R0,Dot           ;;R0 <- ASC DOT(x2e)
Dothead   ADD R2,R2,#-1        ;;R2 <- R2 - 1 (Dot counter --)
          BRn Putchar
          OUT                  ;;OUTPUT: DOT
          BR Dothead
Putchar   LDI R0,Charuser      ;;R0 <- MEM[x4000](Char)
          OUT                  ;;OUTPUT: Char
          OUT
          OUT
          NOT R2,R1
          ADD R2,R2,#9
          ADD R2,R2,#9        ;;R2 <- 17 - R1 (Restore Dot counter)
          LD R0,Dot           ;;R0 <- ASC DOT(x2e)
Dottail   ADD R2,R2,#-1        ;;R2 <- R2 - 1 (Dot counter --)
          BRn Newlineout
          OUT                  ;;OUTPUT: DOT
          BR Dottail
Newlineout LD R0,Newline
          OUT
          LDI R1,Posuser      ;;R1 <- MEM[x4001](Position)
          BRZ Delaycycle
          ADD R1,R1,#-1        ;;R1 <- R1 - 1 (Position --)
          STI R1,Posuser      ;;MEM[x4001] <- R1 (Store position)
```

Then in order to delay the output, decrease the speed, we need to creat a cycle:

```
Delaycycle LD R2,Delaytime    ;;R2 <- x1000
Delay      ADD R2,R2,#-1
          BRp Delay
          BR Loop
```

Then we start our interrupt routine at x2000. We should get input from KBDR, and judge if it is number or character. If it is number, Then we change position in x4001:

```
Changepos  ADD R1,R1,#9      ;;R1 <- R1 + 9
           LDI R0,Pos        ;;R0 <- MEM[x4001]
           ADD R0,R0,R1      ;;R0 <- R0 + R1
           ADD R1,R0,#-16    ;;R1 <- R0 - 16
           BRnz Storepos
           LD R0,max_width   ;;R0 <- 17
Storepos   STI R0,Pos        ;;MEM[x4001] <- R0
```

If it is character, we change character in x4000:

```
STI R0,Char      ;;MEM[x4000] <- R0 (Changechar)
BR ENDING
```

What's more, at x200, we need to Make KBSR[14] be 1, and Make x0180's value is x2000, where our interrupt routine starts:

```
LD R0,KBSR_NUM    ;;R0 <- x4000
STI R0,KBSR        ;;MEM[xFE00] <- R0 (let KBSR[14] be 1)

LD R0,INTV_FC      ;;R0 <- x2000
STI R0,INTV        ;;MEM[x0180] <- R0 (let INTVx0180 be x2000)
RTI
```

Question & Answer

1. Whether we can use TRAP instruction in interrupt routine?

Answer: Yes. it is an instruction, we will naturally get into it.

2. What's your interrupt routine do?

Answer: according to the input, it will change the character position in x4001 or change the character in 4000.