Lab Report - 05

PB20111686 Ruixuan Huang

Lab Goal

Realize a given C++ program by assembly language.

Solution

Before presenting my solution, I'll talk about my idea briefly.

- C++ language is different from assembly. Before carrying the function, we should do some neccessary initial works.
- LC-3 ISA doesn't include mutiply instruction. So I use instructions realized in lab1.
- I use muti-subtraction to realize the `%` operator. If during muti-subtraction there appears zero, that means this number is divisible. If negative appears, that means not.
- Using `RET` to store where the `HALT` instruction locates.

```
.ORIG x3000
INIT   AND R1, R1, #0
       AND R2, R2, #0
       ADD R1, R1, #1  ;R1 = 1
       ADD R2, R2, #2  ;R2 = 2;[R2 = i]
       JSR TIMES
       HALT
TIMES  AND R3, R3, #0  ;R3 = 0
       AND R4, R4, #0  ;R4 = 0
       ADD R4, R4, R2  ;R4 = i
TLOOP  ADD R3, R3, R2  ;R3 += i
       ADD R4, R4, #-1 ;R4--
       BRp TLOOP       ;if(R4 > 0) goto TLOOP, if not, R3 is i*i
       NOT R3, R3
       ADD R3, R3, #1
       ADD R3, R0, R3  ;R3 = R0 - i*i
       BRzp MOD
       RET
MOD    AND R5, R5, #0  ;R5 = 0
       AND R6, R6, #0  ;R6 = 0
       ADD R5, R5, R0  ;R5 = R0
       NOT R6, R2
       ADD R6, R6, #1  ;R6 = -i
LOOP   ADD R5, R5, R6  ;R5 = R0 - i
       BRp LOOP
       BRn IFN
       BRz IFZ
IFN    ADD R2, R2, #1
       BRnzp TIMES
IFZ    AND R1, R1, #0
       RET
.END
```