

ICS Lab4 Report

PB20000096 潘廷岳

Pro_1

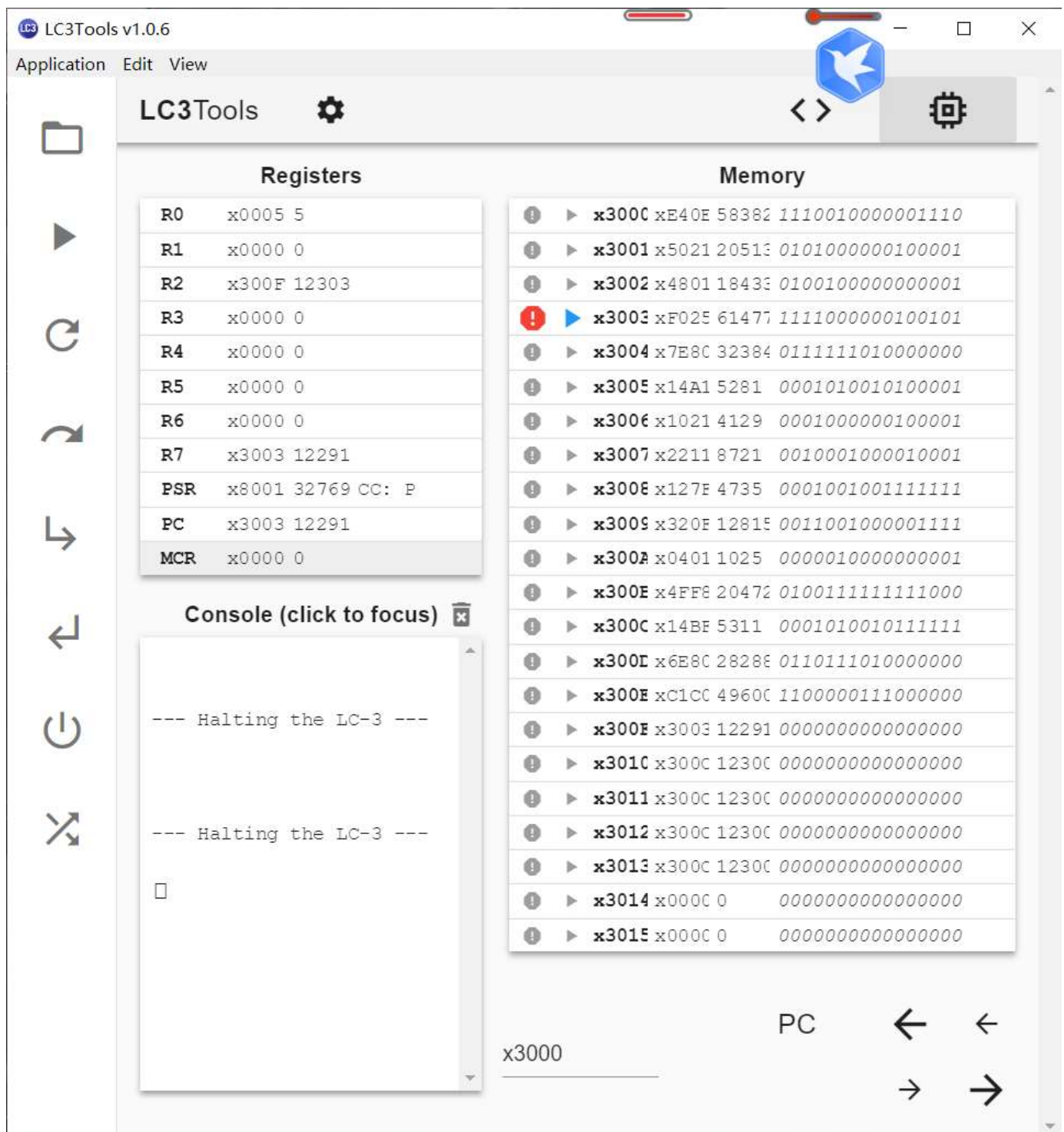
通过阅读，按照逻辑可以补全程序。这个程序貌似没啥作用。补全过程如下所示：

```
;R0~7=0

;R0 = 5, R1 = 0, R2 = 300f, R3 = 0
;R4 = 0, R5 = 0, R6 = 0, R7 = 3003

;x3000;1110 010 000001110 //LEA, R2<-x300F
;0101 000 000 1 00000 //AND, R0<-0
;;0100 1 0000000000x // JSR x3004 (R7=x3003)
;1111 0000 00100101 // trap x25
;0111 111 010 000000 //STR, R7 -> mem[R2=x300f]
;;x3005;0001 010 010 1 0x001//ADD R2 <- R2+1
;0001 000 000 1 00001//ADD R0<-R0+1
;0010 001 000010001//LD R1<-mem[x3019]
;;0001 001 x01 1 11111//ADD R1 <- R1-1;
;0011 001 000001111//ST R1->mem[x3019]
;x300A;0000 010 000000001//BRz to x300C
;0100 1 1111111000//JSR to x3004
;0001 010 010 1 11111//ADD R2 <- R2-1
;;01x0 111 010 000000//LDR R7 <- mem[x300f = R2]
;1100 000 111 000000//RET PC=R7
;x300F;0000 000000000000
;0000 000000000000
;0000 000000000000
;0000 000000000000
;0000 000000000000
;0000 000000000000
;x3015;0000 000000000000
;0000 000000000000
;0000 000000000000
;0000 000000000000
;0000 000 000000101
;x301A
```

执行结果如图所示：



Pro_2

这里利用迭代求解一个数的模7余数，并存入R1中。

证明：

设 $x=8k+b, 0<b<8$, 则 $x=7k+(k+b)$
 则 $x\equiv k+b \pmod{7}$
 迭代求解, 最终必得到 $\text{mod } 7$ 结果。

补全过程如下所示:

```
0011 0000 0000 0000
0010 001 000010101 ; x3000 LD, R1<= mem[x3016]
0100 1 00000001000 ; JSR to x300A,R7<=x3002
0101 010 001 1 00111 ; AND, R2=R1 mod 7
0001 001 010 000 100 ;ADD R1=R2+R4
0001 000 001 1 11001; ADD R0=R1-7
0000 001 111111011 ; x3005 BRp to x3001
0001 000 001 1 11001 ; ADD, R0=R1-7
0000 100 000000001 ; BRn to x3009
0001 001 001 1 11001 ; ADD R1=R1-7
1111 0000 0010 0101 ; TRAP x25
0101 010 010 1 00000 ; x300A AND ,R2<=0
0101 011 011 1 00000 ; AND R3<=0
0101 100 100 1 00000 ; AND R4<=0
0001 010 010 1 00001 ;ADD R2=R2+1
0001 011 011 1 01000 ; ADD R3=R3+8
0101 101 011 000 001 ;x300f AND R5,R3,R1;
0000 010 000000001 ; BRz to x3012
0001 100 010 000 100 ; ADD R4,R2,R4
0001 010 010 000 010 ;ADD R2,R2,R2
0001 011 011 000 011 ; ADD R3,R3,R3
0000 001 111111010 ;BRp to x300f
1100 000111000000 ;x3015 RET
0000 0001 0010 0000 ; x3016 .FILL #288
```

执行结果如图所示：

R0	xFFFA	65530
R1	x0001	1
R2	x0000	0
R3	x8000	32768
R4	x0001	1
R5	x0000	0
R6	x0000	0
R7	x3002	12290
PSR	x8004	32772 CC: N
PC	x3009	12297
MCR	x0000	0

Console (click to focus)

!	▶	x3000	x2215	8725	0010001000010101
!	▶	x3001	x4808	18440	0100100000001000
!	▶	x3002	x5467	21607	0101010001100111
!	▶	x3003	x1284	4740	0001001010000100
!	▶	x3004	x1079	4217	0001000001111001
!	▶	x3005	x03FB	1019	0000001111111011
!	▶	x3006	x1079	4217	0001000001111001
!	▶	x3007	x0801	2049	0000100000000001
!	▶	x3008	x1279	4729	0001001001111001
!	▶	x3009	xF025	61477	1111000000100101
!	▶	x300A	x54A0	21664	0101010010100000
!	▶	x300B	x56E0	22240	0101011011100000
!	▶	x300C	x5920	22816	0101100100100000
!	▶	x300D	x14A1	5281	0001010010100001
!	▶	x300E	x16E8	5864	0001011011101000
!	▶	x300F	x5AC1	23233	0101101011000001
!	▶	x3010	x0401	1025	0000010000000001
!	▶	x3011	x1884	6276	0001100010000100
!	▶	x3012	x1482	5250	0001010010000010
!	▶	x3013	x16C3	5827	0001011011000011
!	▶	x3014	x0377	1010	000001111111010