1. 实验题目: BINARY DIVISOR

2. 实验思路&&代码讲解

。 初始化:

```
R1<-2
R2<-1
R3<-0
```

```
0011000000000000 ;.ORIG x3000
0101001001100000 ; initialization
0001001001100010 ; R1 start from 2
010101010100000 ; follow
0001010010100001 ; R2 start from 1
0101011011100000 ; R3 store the result
```

• 循环

```
R3负责存右移后的结果
R1负责和R0与,如果不是0,说明R3下一位要置为1,即R3+=R2
每轮循环最后要R2<<1,R1<<1

在这里,我发现循环中省1条BR指令的方法,即:
0000101111111010 ; BRnp LOOP
利用R1最后一层循环会变成0来作判断。
```

```
; LOOP

010110000100000000; R4 check the bit

0000010000000001; BRZ ADDD

0001011011000010

0001010010000001 ; ADDD

000101111111010 ; BRnp LOOP

0101100010000000

00000100000000001 ; BRZ SAVE

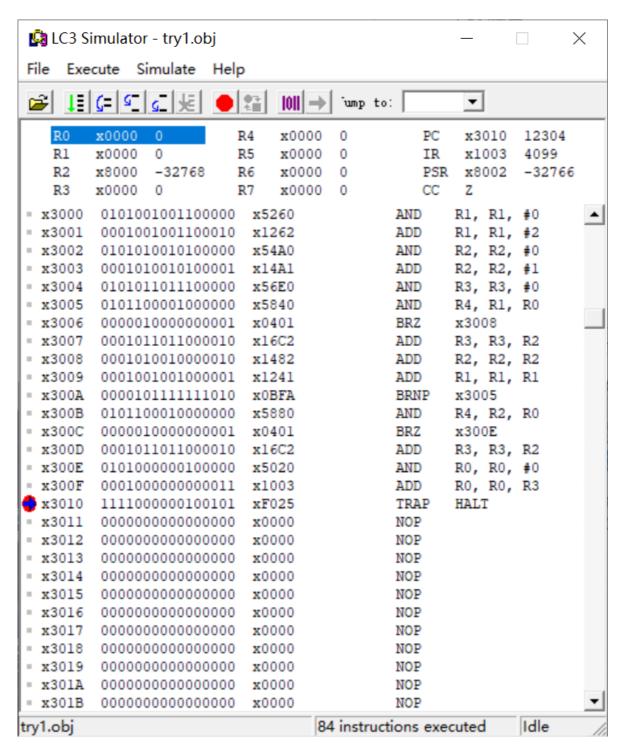
0001011011000010
```

。 最后R3中的结果放回到R0

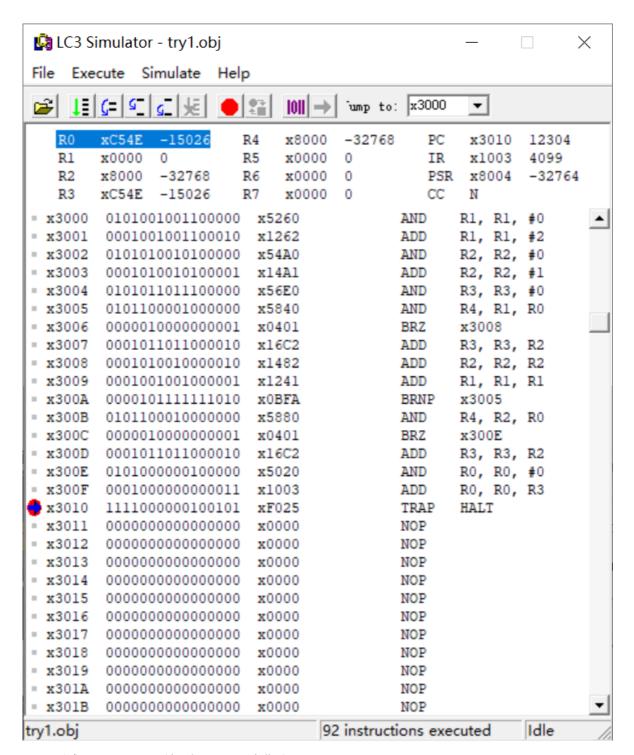
```
010100000100000 ; SAVE
000100000000011
1111000000100101 ; HALT
```

3. 运行结果

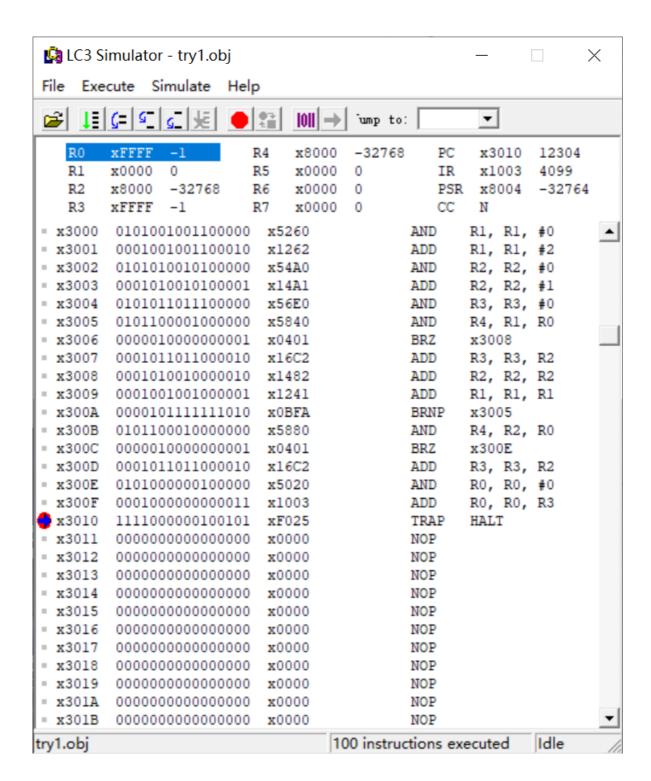
• 测试1: RO<-x0000 结果如图, 84条指令:



• 测试2: RO<-x8A9C 结果如图, 92条指令:



• 测试3: RO<-xFFFF 结果如图, 100条指令:



4. 源代码

0101100010000000

0000010000000001 ; BRZ SAVE

0001011011000010

0101000000100000 ; SAVE

0001000000000011

1111000000100101 ; HALT