

### 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  
0101010010100000 ; 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  
0101100001000000 ; R4 check the bit  
0000010000000001 ; BRZ ADDD  
0001011011000010  
0001010010000010 ; ADDD  
0001001001000001  
0000101111111010 ; BRnp LOOP  
0101100010000000  
0000010000000001 ; BRZ SAVE  
0001011011000010
```

#### ◦ 最后R3中的结果放回到R0

```
0101000000100000 ; SAVE  
0001000000000011  
1111000000100101 ; HALT
```

### 3. 运行结果

- 测试1：R0<-x0000 结果如图，84条指令：

LC3 Simulator - try1.obj

File Execute Simulate Help

Jump to:

R0	x0000	0	R4	x0000	0	PC	x3010	12304
R1	x0000	0	R5	x0000	0	IR	x1003	4099
R2	x8000	-32768	R6	x0000	0	PSR	x8002	-32766
R3	x0000	0	R7	x0000	0	CC	Z	

  

Address	Hex	Binary	Hex	Instruction	Assembly
x3000	0101001001100000	x5260	AND	R1, R1, #0	
x3001	0001001001100010	x1262	ADD	R1, R1, #2	
x3002	0101010010100000	x54A0	AND	R2, R2, #0	
x3003	0001010010100001	x14A1	ADD	R2, R2, #1	
x3004	0101011011100000	x56E0	AND	R3, R3, #0	
x3005	0101100001000000	x5840	AND	R4, R1, R0	
x3006	0000010000000001	x0401	BRZ	x3008	
x3007	0001011011000010	x16C2	ADD	R3, R3, R2	
x3008	0001010010000010	x1482	ADD	R2, R2, R2	
x3009	0001001001000001	x1241	ADD	R1, R1, R1	
x300A	0000101111111010	x0BFA	BRNP	x3005	
x300B	0101100010000000	x5880	AND	R4, R2, R0	
x300C	0000010000000001	x0401	BRZ	x300E	
x300D	0001011011000010	x16C2	ADD	R3, R3, R2	
x300E	0101000000100000	x5020	AND	R0, R0, #0	
x300F	0001000000000011	x1003	ADD	R0, R0, R3	
x3010	1111000000100101	xF025	TRAP	HALT	
x3011	0000000000000000	x0000	NOP		
x3012	0000000000000000	x0000	NOP		
x3013	0000000000000000	x0000	NOP		
x3014	0000000000000000	x0000	NOP		
x3015	0000000000000000	x0000	NOP		
x3016	0000000000000000	x0000	NOP		
x3017	0000000000000000	x0000	NOP		
x3018	0000000000000000	x0000	NOP		
x3019	0000000000000000	x0000	NOP		
x301A	0000000000000000	x0000	NOP		
x301B	0000000000000000	x0000	NOP		

try1.obj 84 instructions executed Idle

- 测试2: R0<-x8A9C 结果如图, 92条指令:

LC3 Simulator - try1.obj

File Execute Simulate Help

Jump to: x3000

R0	x3000	0101001001100000	x5260	AND	R1, R1, #0
R1	x0000	0	x1262	ADD	R1, R1, #2
R2	x8000	-32768	x54A0	AND	R2, R2, #0
R3	x3000	-15026	x14A1	ADD	R2, R2, #1
R4	x8000	-32768	x56E0	AND	R3, R3, #0
R5	x0000	0	x5840	AND	R4, R1, R0
R6	x0000	0	x0401	BRZ	x3008
R7	x0000	0	x16C2	ADD	R3, R3, R2
PC	x3010	12304	x1482	ADD	R2, R2, R2
IR	x1003	4099	x1241	ADD	R1, R1, R1
PSR	x8004	-32764	x0BFA	BRNP	x3005
CC	N		x5880	AND	R4, R2, R0
			x0401	BRZ	x300E
			x16C2	ADD	R3, R3, R2
			x5020	AND	R0, R0, #0
			x1003	ADD	R0, R0, R3
			x3010	TRAP	HALT
			x0000	NOP	
			x0000	NOP	
			x0000	NOP	
			x0000	NOP	
			x0000	NOP	
			x0000	NOP	
			x0000	NOP	
			x0000	NOP	
			x0000	NOP	
			x0000	NOP	
			x0000	NOP	
			x0000	NOP	
			x0000	NOP	
			x0000	NOP	

try1.obj 92 instructions executed Idle

- 测试3: R0<-xFFFF 结果如图, 100条指令:

LC3 Simulator - try1.obj

File Execute Simulate Help

Jump to:

R0	xFFFF	-1	R4	x8000	-32768	PC	x3010	12304
R1	x0000	0	R5	x0000	0	IR	x1003	4099
R2	x8000	-32768	R6	x0000	0	PSR	x8004	-32764
R3	xFFFF	-1	R7	x0000	0	CC	N	

```

x3000 0101001001100000 x5260 AND R1, R1, #0
x3001 0001001001100010 x1262 ADD R1, R1, #2
x3002 0101010010100000 x54A0 AND R2, R2, #0
x3003 0001010010100001 x14A1 ADD R2, R2, #1
x3004 0101011011100000 x56E0 AND R3, R3, #0
x3005 0101100001000000 x5840 AND R4, R1, R0
x3006 0000010000000001 x0401 BRZ x3008
x3007 0001011011000010 x16C2 ADD R3, R3, R2
x3008 0001010010000010 x1482 ADD R2, R2, R2
x3009 0001001001000001 x1241 ADD R1, R1, R1
x300A 0000101111111010 x0BFA BRNP x3005
x300B 0101100010000000 x5880 AND R4, R2, R0
x300C 0000010000000001 x0401 BRZ x300E
x300D 0001011011000010 x16C2 ADD R3, R3, R2
x300E 0101000000100000 x5020 AND R0, R0, #0
x300F 0001000000000011 x1003 ADD R0, R0, R3
x3010 1111000000100101 xF025 TRAP HALT
x3011 0000000000000000 x0000 NOP
x3012 0000000000000000 x0000 NOP
x3013 0000000000000000 x0000 NOP
x3014 0000000000000000 x0000 NOP
x3015 0000000000000000 x0000 NOP
x3016 0000000000000000 x0000 NOP
x3017 0000000000000000 x0000 NOP
x3018 0000000000000000 x0000 NOP
x3019 0000000000000000 x0000 NOP
x301A 0000000000000000 x0000 NOP
x301B 0000000000000000 x0000 NOP

```

try1.obj 100 instructions executed Idle

#### 4. 源代码

```

0011000000000000 ;.ORIG x3000
0101001001100000 ; initialization
0001001001100010 ; R1 start from 2
0101010010100000 ; follow
0101010010100001 ; R2 start from 1
0101011011100000 ; R3 store the result
; LOOP
0101100001000000 ; R4 check the bit
0000010000000001 ; BRZ ADDD
0001011011000010
0001010010000010 ; ADDD
0001001001000001
0000101111111010 ; BRnp LOOP

```

```
0101100010000000
0000010000000001 ; BRZ SAVE
0001011011000010

0101000000100000 ; SAVE
0001000000000011
1111000000100101 ; HALT
```