Lab2实验: Lending Your Name

姓名: 杨涛

学号: PB20020599

L版本程序设计

循环函数为 $F(n) = (F(n-1) + 2 * F(n-3)) \mod 1024$ (1 <= n <= 16384) 其中 $mod 1024 \equiv mod 2^{10}$ 即为取寄存器16位的低10位作为数值 而溢出同样意味着取模,且不会对低位的正确性造成影响,则只需要对寄存器加法进行考虑即可。计算结束后只需要将结果与x03FF 取AND即可得到最终答案。

原始版本

算法直接选择打表,并将打表结果储存在开辟的空间中。

```
.ORIG x3000; start the program at location x3000
LD R4, NUMBER
LD R5, for AND
LEA R1,Data;R1存正在运算的n-3的地址
LDR R3,R1,#2
LOOP LDR R2,R1,#0
ADD R2,R2,R2
ADD R3,R3,R2
AND R3,R3,R5
STR R3,R1,#3
ADD R1,R1,#1
ADD R4,R4,#-1
BRp LOOP
LEA R2,Data;Data的地址
ADD R0,R0,R2;F(n)的地址
LDR R7,R0,#0
HALT
forAND .FILL x03FF
Num .FILL #99
NUMBER .FILL #16381
Data .FILL 1
.FILL 1
.FILL 2
.BLKW 16381 ;总计x4000
.FILL x03A2;F(20)
.FILL x0002;F(02)
.FILL x000A;F(05)
.FILL x0356;F(99)
.END
```

部分实验结果

从x3013起从F(0)开始存储实验结果,即在本例中,F(n)=mem[n+x3013]

•	 	x3000	x2811	10257	LD R4,NUMBER
•	 	x3001	x2A0E	10766	LD R5, forAND
0	 	x3002	xE210	-7664	LEA R1,Data
0	 	x3003	x6642	26178	LDR R3,R1,#2
0	 	x3004	x6440	25664	LOOP LDR R2,R1,#0
0	 	x3005	x1482	5250	ADD R2,R2,R2
0	 	x3006	x16C2	5826	ADD R3,R3,R2
0	 	x3007	x56C5	22213	AND R3,R3,R5
0	 	x3008	x7643	30275	STR R3,R1,#3
0	 	x3009	x1261	4705	ADD R1,R1,#1
0	 	x300A	x193F	6463	ADD R4,R4,#-1
0	 	x300B	x03F8	1016	BRp LOOP
0	 	x300C	xE406	-7162	LEA R2,Data
0	 	x300D	x1002	4098	ADD R0,R0,R2
0	 	x300E	x6E00	28160	LDR R7,R0,#0
0		x300F	xF025	-4059	HALT
0		x3010	x03FF	1023	forAND .FILL x03FF
0		x3011	x0063	99	Num .FILL #99
0		x3012	x3FFD	16381	NUMBER .FILL #16381
0		x3013	x0001	1	Data .FILL 1
0		x3014	x0001	1	.FILL 1
0		x3015	x0002	2	.FILL 2
0		x3016	x0004	4	.BLKW 16381
0		x3017	x0006	6	.BLKW 16381
0		x3018	X000X	10	.BLKW 16381
0	 	x3019	x0012	18	.BLKW 16381
0		x3019 x301A	x0012 x001E	18	.BLKW 16381
	>				
0	>	x301A	x001E	30	.BLKW 16381
0	 	ж301A ж301B	x001E x0032	30 50	.BLKW 16381 .BLKW 16381
0	 • • •	x301A x301B x301C	x001E x0032 x0056	30 50 86	.BLKW 16381 .BLKW 16381 .BLKW 16381
0		x301A x301B x301C x301D	x001E x0032 x0056 x0092	30 50 86 146	.BLKW 16381 .BLKW 16381 .BLKW 16381
0 0 0 0		x301A x301B x301C x301D x301E	x001E x0032 x0056 x0092 x00F6	30 50 86 146 246	.BLKW 16381 .BLKW 16381 .BLKW 16381 .BLKW 16381
0 0 0 0	>>>>	x301A x301B x301C x301D x301E x301F	x001E x0032 x0056 x0092 x00F6 x01A2	30 50 86 146 246 418	.BLKW 16381 .BLKW 16381 .BLKW 16381 .BLKW 16381 .BLKW 16381
0 0 0 0		x301A x301B x301C x301D x301E x301F x3020	x001E x0032 x0056 x0092 x00F6 x01A2 x02C6	30 50 86 146 246 418	.BLKW 16381
0 0 0 0 0		x301A x301B x301C x301D x301E x301F x3020 x3021	x001E x0032 x0056 x0092 x00F6 x01A2 x02C6 x00B2	30 50 86 146 246 418 710	.BLKW 16381
0 0 0 0 0	>>>>>>	x301A x301B x301C x301D x301E x301F x3020 x3021 x3022	x001E x0032 x0056 x0092 x00F6 x01A2 x02C6 x00B2 x03F6	30 50 86 146 246 418 710 178	.BLKW 16381
0 0 0 0 0 0	>>>>>>	x301A x301B x301C x301D x301E x301F x3020 x3021 x3022 x3023	x001E x0032 x0056 x0092 x00F6 x01A2 x02C6 x00B2 x03F6 x0182	30 50 86 146 246 418 710 178 1014	.BLKW 16381
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	 > > > > > > 	x301A x301B x301C x301D x301E x301F x3020 x3021 x3022 x3023	x001E x0032 x0056 x0092 x00F6 x01A2 x02C6 x00B2 x03F6 x0182 x02E6	30 50 86 146 246 418 710 178 1014 386 742	.BLKW 16381
0 0 0 0 0 0 0	 > > > > > > > 	x301A x301B x301C x301D x301E x301F x3020 x3021 x3022 x3023 x3024 x3025	x001E x0032 x0056 x0092 x00F6 x01A2 x02C6 x00B2 x03F6 x0182 x02E6 x02D2	30 50 86 146 246 418 710 178 1014 386 742 722	.BLKW 16381
0 0 0 0 0 0 0 0	 > > > > > > > 	x301A x301B x301C x301D x301E x301F x3020 x3021 x3022 x3023 x3024 x3025 x3026	x001E x0032 x0056 x0092 x00F6 x01A2 x02C6 x00B2 x03F6 x0182 x02E6 x02D2 x01D6	30 50 86 146 246 418 710 178 1014 386 742 722 470	.BLKW 16381
0 0 0 0 0 0 0 0	 > > > > > > > 	x301A x301B x301C x301D x301E x301F x3020 x3021 x3022 x3023 x3024 x3025 x3026 x3027	x001E x0032 x0056 x0092 x00F6 x01A2 x02C6 x00B2 x03F6 x0182 x02E6 x02D2 x01D6 x03A2	30 50 86 146 246 418 710 178 1014 386 742 722 470 930	.BLKW 16381
0 0 0 0 0 0 0 0 0	 > > > > > > > 	x301A x301B x301C x301D x301E x301F x3020 x3021 x3022 x3023 x3024 x3025 x3026 x3027 x3028	x001E x0032 x0056 x0092 x00F6 x01A2 x02C6 x00B2 x03F6 x0182 x02E6 x02D2 x01D6 x03A2 x0146	30 50 86 146 246 418 710 178 1014 386 742 722 470 930 326	.BLKW 16381
0 0 0 0 0 0 0 0 0	> > > <p< th=""><th>x301A x301B x301C x301D x301E x301F x3020 x3021 x3022 x3023 x3024 x3025 x3026 x3027 x3028 x3029</th><th>x001E x0032 x0056 x0092 x00F6 x01A2 x02C6 x00B2 x03F6 x0182 x02E6 x02D2 x01D6 x03A2 x0146 x00F2</th><th>30 50 86 146 246 418 710 178 1014 386 742 722 470 930 326 242</th><th>.BLKW 16381 .BLKW 16381</th></p<>	x301A x301B x301C x301D x301E x301F x3020 x3021 x3022 x3023 x3024 x3025 x3026 x3027 x3028 x3029	x001E x0032 x0056 x0092 x00F6 x01A2 x02C6 x00B2 x03F6 x0182 x02E6 x02D2 x01D6 x03A2 x0146 x00F2	30 50 86 146 246 418 710 178 1014 386 742 722 470 930 326 242	.BLKW 16381
	> <	x301A x301B x301C x301D x301E x301F x3020 x3021 x3022 x3023 x3024 x3025 x3026 x3027 x3028 x3029 x302A	x001E x0032 x0056 x0092 x00F6 x01A2 x02C6 x00B2 x03F6 x0182 x02E6 x02D2 x01D6 x03A2 x0146 x00F2 x0036	30 50 86 146 246 418 710 178 1014 386 742 722 470 930 326 242 54	.BLKW 16381
	> >	x301A x301B x301C x301D x301E x301F x3020 x3021 x3022 x3023 x3024 x3025 x3026 x3027 x3028 x3029 x302A x302B	x001E x0032 x0056 x0092 x00F6 x01A2 x02C6 x00B2 x03F6 x0182 x02E6 x02D2 x01D6 x03A2 x0146 x00F2 x0036 x02C2	30 50 86 146 246 418 710 178 1014 386 742 722 470 930 326 242 54 706	.BLKW 16381
	> >	x301A x301B x301C x301D x301E x301F x3020 x3021 x3022 x3023 x3024 x3025 x3026 x3027 x3028 x3029 x302A x302B x302C	x001E x0032 x0056 x0092 x00F6 x01A2 x02C6 x00B2 x03F6 x0182 x02E6 x02D2 x01D6 x03A2 x0146 x00F2 x0036 x02C2	30 50 86 146 246 418 710 178 1014 386 742 722 470 930 326 242 54 706 166	.BLKW 16381
	> <	x301A x301B x301C x301D x301E x301F x3020 x3021 x3022 x3023 x3024 x3025 x3026 x3027 x3028 x3029 x3028 x3029 x302B x302C x302D	x001E x0032 x0056 x0092 x00F6 x01A2 x02C6 x00B2 x03F6 x0182 x02E6 x02D2 x01D6 x03A2 x0146 x00F2 x0036 x02C2 x00A6 x0112	30 50 86 146 246 418 710 178 1014 386 742 722 470 930 326 242 54 706 166 274	.BLKW 16381
	> >	x301A x301B x301C x301D x301E x301F x3020 x3021 x3022 x3023 x3024 x3025 x3026 x3027 x3028 x3029 x302A x302B x302C x302D x302E	x001E x0032 x0056 x0092 x00F6 x01A2 x02C6 x00B2 x03F6 x0182 x02E6 x02D2 x01D6 x03A2 x0146 x00F2 x0036 x02C2 x00A6 x0112 x0296	30 50 86 146 246 418 710 178 1014 386 742 722 470 930 326 242 54 706 166 274 662	.BLKW 16381
	> >	x301A x301B x301C x301D x301E x301F x3020 x3021 x3022 x3023 x3024 x3025 x3026 x3027 x3028 x3029 x302A x302B x302D x302C x302D x302E x302F	x001E x0032 x0056 x0092 x00F6 x01A2 x02C6 x00B2 x03F6 x0182 x02E6 x02D2 x01D6 x03A2 x0146 x00F2 x0036 x02C2 x00A6 x0112 x0296 x03E2	30 50 86 146 246 418 710 178 1014 386 742 722 470 930 326 242 54 706 166 274 662 994	.BLKW 16381

改进版本1&2

改进版本1

```
充分利用寄存器,存储三个数据并计算下一位数据,计算完毕后进行位置移动。R7=F(n),R1=F(n+1),R2=F(n+2)
计算F(n+3)=R3=R2+2*R7,之后进行移位
R7←R1,R1←R2,R2←R3
由此可实现了R7从F(n)到F(n+1)的变化
根据n=n-1是否为小于则可以判断是否输出。此算法可输出n=0的情况。
```

```
.ORIG x3000
ADD R1,R1,#1
ADD R2,R2,#1
ADD R3,R3,#2
LOOP ADD R7,R1,#0
                      ;移位开始
ADD R1,R2,#0
                      ;移位结束
ADD R2,R3,#0
ADD R3,R7,R7
                      ;计算R3
ADD R3,R2,R3
ADD R0, R0, #-1
BRzp LOOP
LD R6, for AND
AND R7,R7,R6
HALT
forAND .FILL x03FF
Fa .FILL x03A2
Fb .FILL x0002
Fc .FILL x000A
Fd .FILL x0356
.END
```

改进版本2

在改进版本1的基础上作两个调整

- 去掉HALT并调整指令顺序:因为后面四个数据所对应的指令不是BR就是BRp,让中间循环块结束后的ConditionCode来进行验证则必然是NOP,最后结果一定是未运行,则可滑到.END结束程序。
- 将R7*2一步存储到R1中,减少了先移位存储后乘二的步骤,因此改变存储顺序为 R1←R7←R2←R3,其中R1=2*R7。

```
;2*R7=R1←R7←R2←R3
.ORIG x3000
ADD R7,R7,#1
ADD R2, R2, #1
ADD R3,R3,#2
LD R6, for AND
LOOP ADD R1,R7,R7
ADD R7,R2,#0
AND R7,R7,R6
ADD R2,R3,#0
ADD R3,R1,R2
ADD R0, R0, #-1
BRp LOOP
forAND .FILL x03FF
Fa .FILL x03A2
Fb .FILL x0002
Fc .FILL x000A
Fd .FILL x0356
.END
```

正确性检验

初始版和改进版采用两种不同的算法,则直接对比两者的结果输出,可知其基本正确。

输入n	初始版结果	改进版结果
20	930	930
2	2	2
5	10	10
99	854	854

行数计算

改进前的版本为27行,改进后减少至两个版本分别为18行和16行。