

ICS Lab2 实验报告

张艺耀 PB2011630

原初设计思路

初始化 将F(0) F(1) F(2)分别存在R1 R2 R3中。

当 $R3 = F_n$ 时 计算 $F_{n+1} = 2 * R1 + R3$ 存到R1、R7；

当 $R1 = F_n$ 时 计算 $F_{n+1} = 2 * R2 + R1$ 存到R2、R7；

当 $R2 = F_n$ 时 计算 $F_{n+1} = 2 * R3 + R2$ 存到R3、R7；以此类推。

R0用于计数，每次循环 $R0 = R0 - 1$ ，当 $R0 = 0$ 时halt。

为了实现模1024的操作，我们将R5初始化为1，R6初始化为10用于计数。将R5左移10位后取反 $R4 = R5$ 左移1位取反 将R4和R5相与的结果存入R5中 得到11位12位均为0，其余位均为1的数，这样才能保证在每次计算过程中只需将相应寄存器内值与R5内值相与即可得到模1024的结果。

最后将相应的计算结果存到R7中即可。

考虑到 $n = 1, 2$ 的情况，我们需要加个简单的特判。

原初代码实现

```
1      .ORIG x3000
2      ADD R1,R1,#1
3      ADD R2,R2,#1
4      ADD R3,R3,#2
5      ADD R5,R5,#1
6      ADD R6,R6,#10
7  AGAIN  ADD R5,R5,R5
8          ADD R6,R6,#-1
9          BRp AGAIN
10         ADD R4,R5,R5
11         NOT R4,R4
12         NOT R5,R5
13         AND R5,R4,R5
14         ADD R0,R0,#-2;if R0 = 1, 2
15         BRz TOED1
16         BRn TOED2
17 LABEL  ADD R0,R0,#-1
18         BRn TOED
```

```

19      ADD R1,R1,R1
20      ADD R1,R3,R1
21      AND R1,R1,R5
22      ADD R7,R1,#0
23      ADD R0,R0,#-1
24      BRn TOED
25      ADD R2,R2,R2
26      ADD R2,R1,R2
27      AND R2,R2,R5
28      ADD R7,R2,#0
29      ADD R0,R0,#-1
30      BRn TOED
31      ADD R3,R3,R3
32      ADD R3,R2,R3
33      AND R3,R3,R5
34      ADD R7,R3,#0
35      BRnzp LABEL
36 TOED1  ADD R7,R7,#1
37 TOED2  ADD R7,R7,#1
38 TOED   .BLKW  #1
39      .END
40 N1     .FILL  x03A2
41 N2     .FILL  x00F6
42 N3     .FILL  x0182
43 N4     .FILL  x0332

```

简单c语言验证程序：

```

1  #include<stdio.h>
2  int main(){
3      int n, count = 1, out;
4      while(1){
5          scanf("%d", &n);
6          if(!n) break;
7          int x = 1, y = 1, z = 2;
8          if(n == 1||n == 2) out = 1;
9          for(int i = 2; i < n; i++,count++){
10             if(count%3 == 1){
11                 x = (2*x + z)%1024;
12                 out = x;
13             }
14             else if(count%3 == 2){
15                 y = (2*y + x)%1024;
16                 out = y;
17             }
18             else{
19                 z = (2*z + y)%1024;
20                 out = z;
21             }
22         }
23         printf("N = %d Result: %d\n",n, out);

```

```

24     x = y = 1; z = 2; count = 1;
25     }
26     return 0;
27 }

```

将本人学号20 11 16 30代入得计算结果分别为：

930 246 386 818

对应16进制数分别为：

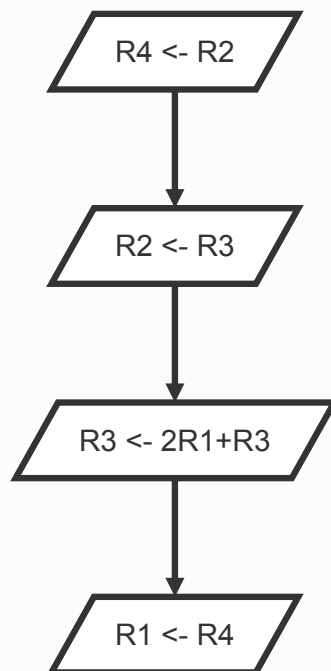
x03A2 x00F6 x0182 x0332

最终实现

可以看到这种方式实现的代码行数较多，因为R1 R2 R3轮换存 F_{n+1} 的值，很不实用。我们考虑只用R3存 F_{n+1} 的值。

我们考虑用R4暂存R2的值，顺序轮换。

flowchart如下：



代码如下：

```

1      .ORIG x3000
2      ADD R1,R1,#1
3      ADD R2,R2,#1
4      ADD R3,R3,#2
5      ADD R5,R5,#1
6      ADD R6,R6,#10
7  AGAIN  ADD R5,R5,R5

```

```

8      ADD R6,R6,#-1
9      BRp AGAIN
10     ADD R4,R5,R5
11     NOT R4,R4
12     NOT R5,R5
13     AND R5,R4,R5
14     ADD R0,R0,#-2;if R0 = 0,1
15     BRz TOED1
16     BRn TOED2
17 LABEL ADD R0,R0,#-1
18     BRn TOED
19     ADD R4,R2,#0
20     ADD R2,R3,#0
21     ADD R1,R1,R1
22     ADD R3,R3,R1
23     AND R3,R5,R3
24     ADD R1,R4,#0
25     ADD R7,R3,#0
26     BRnzp LABEL
27 TOED1 ADD R7,R7,#1;if R0 = 2
28 TOED2 ADD R7,R7,#1;if R0 = 0,1
29 TOED  HALT
30      .END
31 N1   .FILL  x03A2
32 N2   .FILL  x00F6
33 N3   .FILL  x0182
34 N4   .FILL  x0332

```

共34行 除去寄存器初始化14行。

经验证，程序正确

 home.ustc.edu.cn

测试数据 $F(1) = 1$ 你的回答正确, 指令数 43

测试数据 $F(2) = 2$ 你的回答正确, 指令数 43

测试数据 $F(3) = 4$ 你的回答正确, 指令数 54

测试数据 $F(24) = 706$ 你的回答正确, 指令数 264

测试数据 $F(144) = 642$ 你的回答正确, 指令数 1464

测试数据 $F(456) = 66$ 你的回答正确, 指令数 4584

测试数据 $F(1088) = 2$ 你的回答正确, 指令数 10904

测试数据 $F(1092) = 290$ 你的回答正确, 指令数 10944

测试数据 $F(2096) = 898$ 你的回答正确, 指令数 20984

测试数据 $F(4200) = 322$ 你的回答正确, 指令数 42024

测试数据 $F(8192) = 514$ 你的回答正确, 指令数 81944

测试数据 $F(12000) = 258$ 你的回答正确, 指令数 120024

测试数据 $F(14000) = 898$ 你的回答正确, 指令数 140024

平均指令数 33330.769230769234

确定