

Lab 5 Riddle

实验内容

- 将C语言中的质数判断程序转换为汇编语言

程序设计

```
.ORIG x3000

;initial
AND R1, R1, #0; clear R1 = 0;
ADD R2, R1, #2; int i = 2;
ADD R1, R1, #1; r1 = 1;
JSR JUDGE
HALT

JUDGE
ST R7, RETURN
;while (i * i <= r0)
LOOP
JSR POW;          i * i
NOT R3, R3
ADD R3, R3, #1
ADD R3, R0, R3; R3 = R0 - i * i
BRn OVER;         compare i * i with R0

;if(r0 % i == 0)
JSR MOD;          check R5
BRZ IsNotPrimer;break

ADD R2, R2, #1; i += 1
BR LOOP

IsNotPrimer
AND R1, R1, #0

OVER
LD R7, RETURN
RET

POW
ADD R6, R2, #0; save R2
AND R3, R3, #0; clear R3 = 0;
ADD R4, R3, #1; R4 = 1
AND R5, R4, R2; get bit of R2
BRZ #1;          if(R2[n] == 0) ignore add
ADD R3, R3, R6; temp sum
ADD R6, R6, R6; R6 << 1
ADD R4, R4, R4; R4 << 1
BRnp #-6;
RET;             return R3
```

```

MOD
NOT R3, R2
ADD R3, R3, #1; R3 = -R2
AND R4, R4, #0;

ADD R4, R4, R3; n * -R2
ADD R5, R4, R0; R0 = n * R2
BRp #-3;
RET;          return R5
RETURN .BLKW #1
.END

```

以上这段代码就是对题目中C语言代码的翻译，翻译过程如下：

- 在原代码中，不能直接实现的语句为 $i * i$ 和 $r0 \% i == 0$ ，因此我们可以将这两个语句独立为两个函数进行单独处理：
 - 乘方函数：这里采用快速移位乘法，具体实现已经在Lab1中体现过，故不再赘述。快速乘法的目的主要在于将多次循环中的乘法进行简化，从而尽可能减少执行的指令；
 - 取模函数：由于这里的取模运算不再是对2的n次方进行取模，因此不能用简单的与运算来处理。注意到这里实际上是要判断i是否能整除r0，因此我们只需要对i不断自增直至大于r0，若其中某一次出现两数相减为0的情况，则判断出i可以整除r0。
- 按照题目的要求，我们需要独立出JUDGE函数，但JUDGE函数中也调用了其他函数，这会导致R7的值的变化，从而不能顺利返回主程序，而在POW函数中我们基本使用了全部的寄存器，因此只能在调用前先将R7的值保存在内存中，返回后再读出，从而得以返回主程序；

程序分析

该程序的时间开销主要集中在判断质数的循环上，其中判断整除的MOD函数也使用了循环，当i较小时，这个循环执行的指令条数较多。但由于我没有思考出很简洁的取模快速算法，因此采取了这种比较基本的算法。

在程序正确性方面，我对诸多数据进行了测试，结果均正确。这里举出对2、997和10000的判断结果：

- 对2的判断：

Registers			
R0	x0002	2	
R1	x0001	1	
R2	x0002	2	
R3	xFFFE	-2	
R4	x0000	0	
R5	x0000	0	
R6	x0000	0	
R7	x3004	12292	
PSR	x8001	-32767	CC: P
PC	x3004	12292	
MCR	x0000	0	

- 对997的判断：

Registers			
R0	x03E5	997	
R1	x0001	1	
R2	x0020	32	
R3	xFFE5	-27	
R4	x0000	0	
R5	x0000	0	
R6	x0000	0	
R7	x3004	12292	
PSR	x8001	-32767	CC: P
PC	x3004	12292	
MCR	x0000	0	

- 对10000的判断：

Registers			
R0	x2710	10000	
R1	x0000	0	
R2	x0002	2	
R3	xFFFE	-2	
R4	xD8F0	-10000	
R5	x0000	0	
R6	x0000	0	
R7	x3004	12292	
PSR	x8001	-32767	CC: P
PC	x3004	12292	
MCR	x0000	0	

实验总结

在本次实验中，我加深了对高级语言的底层实现的理解，同时也意识到：某些高级语言的一个语句可能底层实现是较为繁琐的，这可能会降低程序运行效率，但这种通过调用功能函数来简便地实现更高级功能的编程思想是极为重要的，这能够使得程序员只需要了解函数实现功能就可以顺利在编程中完成该项功能。