

# ICS Lab1 实验报告

Name: 张展翔

Student Number: PB20111669

## L程序

### 程序设计

由于L程序要求使用的代码行数尽可能的短，因此，我们可以将寄存器 $R_1$ 的值作为累加的次数， $R_0$ 的值来作为累加的量，每次累加后 $R_0$ 的大小减一，直到为0为之。

故我们可以得出以下代码：

```
0011 0000 0000 0000 ; start the program at location x3000
0001 111 000 0 00 111;R7=R0+R7
0001 001 001 1 11111;R1=R1-1
0000 101 111111101 ;if(R0!=0) goto HALT,else goto HALT
1111 0000 00100101 ; HALT
```

### 程序分析

去除开始和结尾，共使用了3行代码

由于该程序利用累加法，因此执行的指令数与 $R_1$ 的大小成正相关

## P程序

### 程序设计

由于P程序要求运行执行指令数尽可能少，故我们可以考虑利用2进制乘法的原理来实现

设这两个数分别为a(储存在 $R_0$ )和b(储存在 $R_1$ )

```
while(i<i_max){
    if (b[i]==1)
    {
        R7+=a;a=a<<1;
    }
    else
        a=a<<1;
    i++;
}
```

因此，可以写出如下代码：

```

0011 0000 0000 0000;start at x3000
0001 010 010 1 00001; R2 = 1
0000 010 000000110; if (R2==0) goto HALT
0101 011 001 0 00 010; R3 = R1 & R2
0000 010 000000001; if (R3==0) skip next
0001 111 111 0 00 000; R7 = R7 + R0
0001 000 000 0 00 000; R0 *= 2
0001 010 010 0 00 010; R2 *= 2
0000 111 111111001; goto R2?0
1111 0000 00100101; HALT

```

## 程序分析

去除开始和结尾，共使用了8行代码

最坏情况下，及每次都需要进行循环，每个循环执行7跳指令，共执行16遍，故所需执行指令数  $w = 16 * 7 + 3 = 115$

最好情况下不需要执行循环，只需进行条件判断，故所需执行指令数  $w = 16 * 6 + 3 = 99$

平均情况下，所需执行指令数  $w = 16 * 6.5 + 3 = 107$

## P程序优化

## 程序设计

由于P程序中执行了大量的判断语句，而无论如何程序都需要从R0的每一位进行判断，故可将循环展开来避免使用所需执行的判断语句

```

0011 0000 0000 0000;start at x3000
0001 010 010 1 00001; R2 = 1

0101 011 001 0 00 010; R3 = R1 & R2
0000 010 000000001; if (R3==0) skip next
0001 111 111 0 00 000; R7 = R7 + R0
0001 000 000 0 00 000; R0 *= 2
0001 010 010 0 00 010; R2 *= 2

0101 011 001 0 00 010; R3 = R1 & R2
0000 010 000000001; if (R3==0) skip next
0001 111 111 0 00 000; R7 = R7 + R0
0001 000 000 0 00 000; R0 *= 2
0001 010 010 0 00 010; R2 *= 2

0101 011 001 0 00 010; R3 = R1 & R2
0000 010 000000001; if (R3==0) skip next
0001 111 111 0 00 000; R7 = R7 + R0
0001 000 000 0 00 000; R0 *= 2
0001 010 010 0 00 010; R2 *= 2

0101 011 001 0 00 010; R3 = R1 & R2
0000 010 000000001; if (R3==0) skip next
0001 111 111 0 00 000; R7 = R7 + R0
0001 000 000 0 00 000; R0 *= 2
0001 010 010 0 00 010; R2 *= 2

0101 011 001 0 00 010; R3 = R1 & R2

```

[illegible]

```
0101 011 001 0 00 010; R3 = R1 & R2
0000 010 000000001; if (R3==0) skip next
0001 111 111 0 00 000; R7 = R7 + R0
0001 000 000 0 00 000; R0 *= 2
0001 010 010 0 00 010; R2 *= 2

0101 011 001 0 00 010; R3 = R1 & R2
0000 010 000000001; if (R3==0) skip next
0001 111 111 0 00 000; R7 = R7 + R0
0001 000 000 0 00 000; R0 *= 2
0001 010 010 0 00 010; R2 *= 2

1111 0000 00100101; HALT
```

## 程序分析

去除开始和结尾，共使用了81行代码

最坏情况下，需要执行指令数为 $w = 5 * 16 + 3 = 83$

最好情况下，需要执行指令数为 $w = 4 * 16 + 3 = 67$

平均情况下，所需执行指令数 $w = 16 * 4.5 + 3 = 75$