电子科技大学信息与软件工程学院

**实 验 报 告**

学 号 2018091602007

姓 名 王乐卿

（实验）课程名称《ARM处理器体系结构及

应用》课程实验

理论教师 兰刚

实验教师 兰刚

**电 子 科 技 大 学**

**实 验 报 告**

**学生姓名：王乐卿 学号：2018091602007 指导教师：兰刚**

**实验地点：家 实验时间：2020.6.6**

1. 实验名称：数值变换
2. 实验学时：4
3. 实验目的：
4. 通过实验让学生掌握计算机是如何输入数据、处理数据、输出数据的。
5. 通过实验让学生掌握计算机的工作原理，培养学生的全局观念。
6. 通过该编程实验，进一步巩固和强化学生ARM汇编编程的能。
7. 实验原理：

**1、十进制转换**

32位无符号二进制数转换为十进制数有多种方法，但本实验由于需要提取出十进制数中的每一位，单独存放进一个字节单元，因此本实验采用以下的方法，其基本原理如下：

1. 用10去除要转换的32位无符号二进制数，得到的余数y个就是个位数，而商为S个。
2. 再用10去除上步得到的商S十，得到的余数y十就是十位数。
3. 反复使用1010去除上步得到的商Sx，得到的余数yx就是对应位数的值。
4. 由于32位2进制数最大可以表示的十进制数为4,294,967,295，共有10位，因此进行10次除10的循环操作后，就可以得到转换后的十进制数的每一位。

**2、整除10**

实现除以10的方法有多种，比如：把被除数不断的减去10，每减一次，商就加1，只要被除数还没有减到小于10，就循环进行这个减法操作，直到被除数被减到小于10为止，这时候就可以得到商和余数。本次实验推荐的方法如下：

1. 初始时。被除数存放在R3中，R5存放的是余数（初始化为0），R6存放的是商（初始化为0）。
2. R3、R5、R6进行逻辑左移每次左移1位。其中R3的最高bit位（bit31位）将移动到R5的bit0位。
3. 然后判断，位移后的R5是否大于10，如果大于10，则从余数（R5）中减去10，并把商（R6）加1。（注：商加1，则相当于原始加10）。
4. 然后反复执行（2）、（3）步骤，一共做32次位移，每个比特位移一次，进行一次循环，32次左移完成后，则就完成了除以10的操作。对于第（2）步的减10操作，经过1次左移后，则等效于减20的操作，经过2次左移后，则等效于减40的操作，以此类推。同理，对于商来说，加1的操作，经过1次左移后，就等效于加2的操作，经过2次左移后，则等效于加4的操作，以此类推。

3、LED码转换关系

LED码是也称为七段编码，是用于LED数码管最示相关数字使用的编码，在本实验中，各数字和LED码的对应关系如下表1-1:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 十进制数 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| LED码 | 0X3F | 0X06 | 0X5B | 0X4F | 0X66 | 0X6D | 0X7D | 0X07 | 0X7F | 0X7F |

**3、程序流程图**

程序流程图Markdown代码：

|  |
| --- |
| st=>start: 开始  oper1=>operation: 初始化开始  R0：待转换的无符号32位二进制数  R1：存放十进制数的起始地址  R7：十进制数的最大位数  oper2=>operation: 除10取余数  R5：存放余数  R6：存放商  oper3=>operation: 十进制数转换为LED码  cond=>condition: 转换完否  ed=>end: 结束  st->oper1->oper2->cond(yes)->oper3->ed  cond(no)->oper2 |

程序流程如图4-1所示：

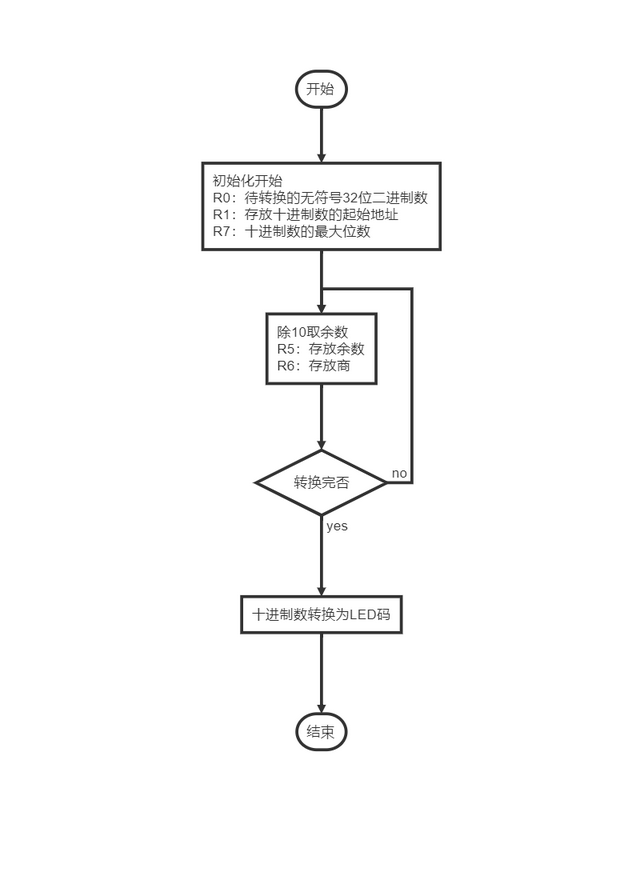


图2 二进制转换为LED码流程图

1. 实验内容：
2. 将寄存器中的一个无符号 32 位二进制数先转换为 10 进制数。

* 要转换的数在 R0 中，该数能任意设置；
* 十进制数的每一位存放在0X40000000开始的**字节单元**中，十进制数的低位存放在低地址单元中；
* 实验中先画程序流程图，再写代码。

1. 将上个实验所得到的10 进制数转换为 LED 码。

* 十进制数的每一位存放在0X40000000开始的**字节单元**中；
* LED 码数放在 0X40000010 开始的**字节单元**中；
* 实验中先画程序流程图，再写实验代码。

1. 实验器材（设备、元器件）：
2. PC机一台；
3. Keil MDK-ARM uVision4开发工具。
4. 实验步骤：
5. 打开Keil MDK-ARM uVision4开发工具；
6. 新建一个工程文件；
7. 在新建的工程文件中，添加新的源程序文件
8. 编写代码
9. 选择“Build target”菜单对编写好的工程文件进行编译链接。
10. 点击““Start/Stop Debug Section””按键，对程序进行跟踪调试，在调试界面，单步执行，对CPU各寄存器的值的变化、以及相关内存的变化进行分析比较，判断程序的执行是否符合预期要求。
11. 实验结果与分析（含重要数据结果分析或核心代码流程分析）
12. **实验代码**

代码1数值变换程序代码

|  |
| --- |
| AREA Init, CODE, READONLY  ENTRY  LDR R0,=0521632478 ;将十进制数以无符号32位二进制数输入  LDR R1,=0X40000000 ;R1为十进制数的起始地址，每字节放一位十进制数  LDR R2,=0X40000010 ;R2为LED码的起始地址，每字节放一位  MOV R7,#10 ;转换位数，最大就是10位数  MY\_M1 BL MY\_DIV10 ;调用转换为十进制数，每次转换一位  STRB R5,[R1],#1  MOV R0,R6  SUBS R7,R7,#1  BNE MY\_M1  B MY\_TEND  MY\_DIV10 MOV R3,R0 ;该子程序用于除10得余数，余数在R5中  MOV R4,#32 ;循环操作次数，总共循环32次，一次1bit。  MOV R5,#0 ;R5为余数  MOV R6,#0 ;R6为商  MY\_DIV2 MOV R6,R6,LSL #1 ;R6为商。商逻辑左移，相当于乘2  MOV R5,R5,LSL #1 ;R5为余数，余数逻辑左移，相当于乘2  MOVS R3,R3, LSL #1 ;R3被除数，逻辑左移1位，如果最高位为1，则左移后  ;进位位C为1。  ADC R5,R5,#0 ;如果有进位位，则进位位加到余数里。  SUBS R5,R5,#10 ;把左移，且加上进位位（原被除数最高位）的余数减10.  ADDCS R6,R6,#1 ;如果比10大，则把商加1  ADDCC R5,R5,#10 ;如果比10小，则把原来减去的10加回去，通过这种方式，可以始终保证余数小于除数10。而且前期减去的10，随着左移就相当于10\*2n（n是移位次数）  SUBS R4,R4,#1 ;是否32位都移位完成。  BNE MY\_DIV2  BX LR ;如果都完成32位移位，则R5为余数，R6为商  MY\_TEND LDR R3,=BCDCODE ;转换为LED码  LDR R4,=0X40000000 ;R4为十进制数的起始地址  LDR R5,=0X40000010 ;R5为LED码的起始地址  MOV R6,#10 ;一共有10位数  MY\_TEND1 LDRB R7,[R4],#1  LDRB R8,[R3,R7]  STRB R8,[R5],#1  SUBS R6,R6,#1  BNE MY\_TEND1  MOV R0,R0    BCDCODE DCB 0X3F,0X06,0X5B,0X4F,0X66,0X6D,0X7D,0X7F,0X6F,0X00,0X00,0X00  END |

1. **运行过程及结果界面截图**

下图3是程序运行结果的界面。从图中可以看出从0x40000000单元开始，每个字节单元都存放转换后的10进制数的1位，且十进制数的低位在低地址单元，高位在高地址单元。转换后的LED码是存放在从0x40000010开始的单元中，由于我们输入时，使用的是LDR R0,=0521632478，该伪指令将十进制数0521632478作为32位二进制数输入，因此经过我们三十二位准换为十进制数保存到0x40000000之后，0x40000000单元内存储的数据应该形式上与输入的十进制数相同，从图3中可以看到，结果符合预期要求。在转换LED码时，只要实现数字的一一对应即可，实验结果亦符合预期要求。

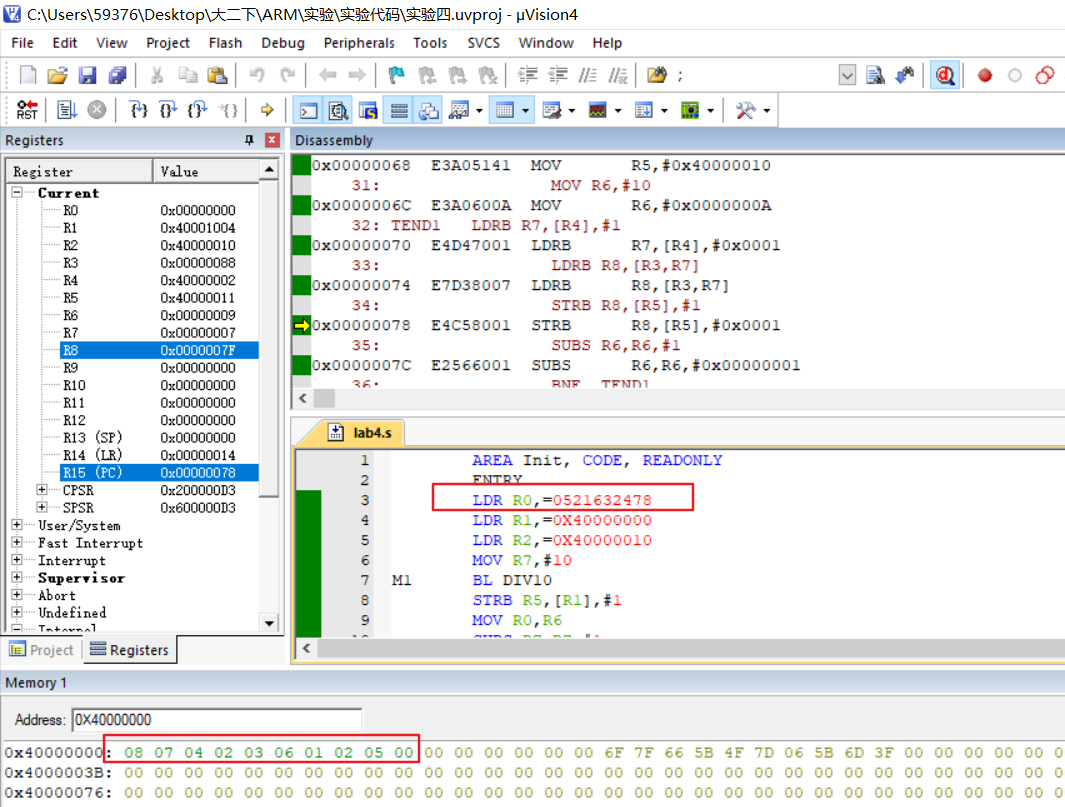


图3 数值转换程序运行结果图

1. **实验结论**

该程序实现了把一个无符号32位二进制数，转换为10位10进制数，并进一步把10进制数转换为对应的LED码。

1. 总结及心得体会：

通过本次实验，我更加深入的掌握了计算机是如何输入数据、处理数据、输出数据的。对计算机的工作原理也有了更深入的理解，同时通过该编程实验，进一步巩固和强化了我ARM汇编编程的能力。

在本次实验过程中，我重新回顾了很多之前已经遗忘的知识，比如LDR,Rn，=xxx的伪指令，同时，在进行算法设计实现的过程中，我也成功掌握了如何通过查表等方法，实现数码转换。

1. 对本实验过程及方法、手段的改进建议：

实验指导中并没有说明何为LED码，且给出的示例报告中存在不少的错误，比如把LED码错写为BCD码，造成了理解上的误导。

报告评分：

指导教师签字：