



哈爾濱工業大學（深圳）

Harbin Institute of Technology, Shenzhen

课程实验报告

课程名称：汇编语言程序设计

实验名称：实验三

学生所在院（系）：计算机科学与技术学院

学号：190110509

学生姓名：王铭

上交日期：2020/12/13

指导老师：裴文杰

目 录

| | |
|----------------------------------|---|
| 1. 问题描述（将题目转换为计算机问题进行描述） | 1 |
| 2. 解决方案（包括程序具体设计和流程图） | 1 |
| 3. 具体实现（源代码和程序调试步骤、程序运行截图） | 2 |
| 4. 总结（实验中遇到的问题和收获） | 7 |

1. 问题描述

将题目所给的用十进制表示的数字转换为二进制和十六进制，即设计一个程序，该程序能够计算每个所给数字的二进制和十六进制表示形式，并将每一位数字转换为 ASCII 码，打印到屏幕中显示。

2. 解决方案

2.1 程序具体设计

在数据段中，用 Number 数组存储所给的要转换的十进制数。设计能够将十进制数转化成二进制和十六进制数且将结果打印到屏幕上的子程序。在主程序中，每次调用子程序并输出。

子程序设计：

将要转换的进制存储到 k 所表示的存储单元当中，要转换的数的地址存储在 bx 寄存器当中。进行循环，每次循环除以 k，将 ax 寄存器进栈保存，同时判断此次除法结果的商是否为 0，为 0 则退出循环否则继续循环。当循环终止时，要转换的数的 k 进制表示自顶向下地保存在栈当中，则每次取栈顶元素转换为 ASCII 码输出到屏幕上进行循环，直至栈空。

2.2 流程图

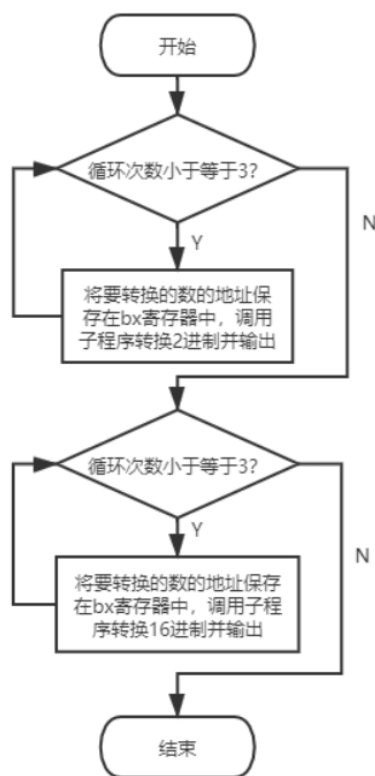


图 2.2.1 主程序流程图

子程序流程图：

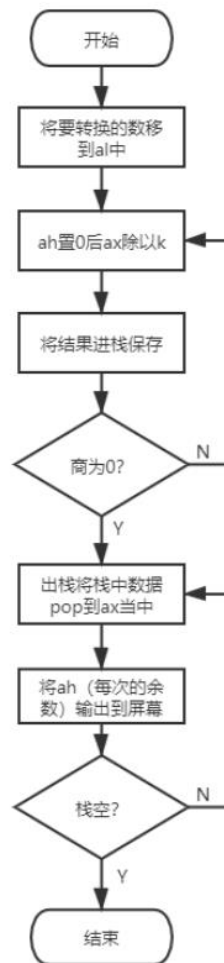


图 2.2.2 子程序流程图

3. 具体实现

3.1 源代码

OUTPUT MACRO k ;将k输出到屏幕

```

    local out1
    mov al,k
    add al,'0'
    cmp al,39h
    jle out1
    add al,07h

```

out1:

```

    mov dl,al
    mov ah,02h
    INT 21h

```

endm

CTRL MACRO ;输出回车符

```
        mov dl,0dh
        mov ah,02h
        INT 21h
        mov dl,0ah
        mov ah,02h
        INT 21H
    endm

DATA SEGMENT
Number db 117,36,77
k db 2
DATA ENDS

STACK SEGMENT
db 60 dup(?)
top db ?
STACK ENDS

CODE SEGMENT
assume ds:DATA,cs:CODE,ss:STACK
main proc
start::
    mov ax,DATA
    mov ds,ax
    mov ax,STACK
    mov ss,ax
    lea sp,top
    lea bx,Number ;将Number的首地址赋值给bx寄存器
    mov cx,3
loop1:
    call near ptr cal
    inc bx
    CTRL
    loop loop1
    mov cx,3
    mov k,16
    lea bx,Number
loop2:
    call near ptr cal
    inc bx
    CTRL
    loop loop2
    jmp last
main endp
```

```

cal proc near ;要转换的数的地址存储在bx寄存器当中，要转换的进制存储在k当中
    mov al,[bx]
beg:
    mov ah,0
    div k ;每次除以k，把余数进栈
    push ax
    cmp al,0 ;商为0，结束循环
    jnz beg
ex1:
    pop ax
    OUTPUT ah ;输出余数
    cmp sp,offset top-2 ;到栈顶停止
    jnz ex1
    ret
cal endp
last:
    mov ah,4ch
    INT 21h
CODE ENDS
end start
    
```

3. 2程序调试步骤

初始化并将Number数组的地址置于bx寄存器当中

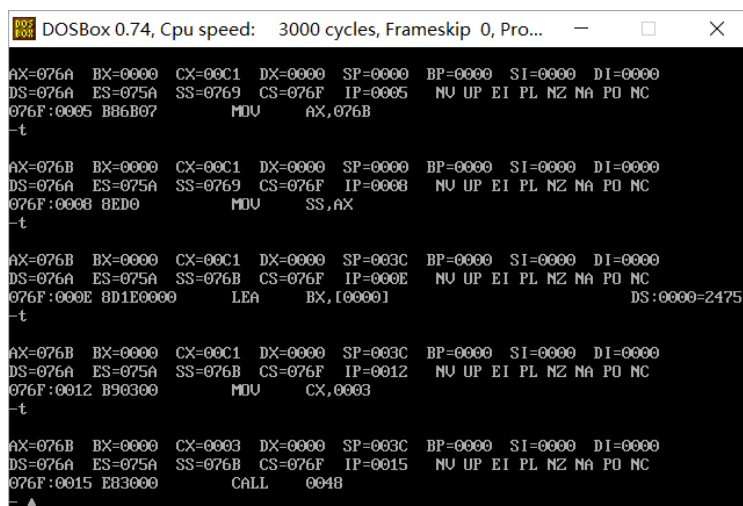


图3.2.1 初始化

调用子程序：

```
AX=076B BX=0000 CX=0003 DX=0000 SP=003A BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=076B CS=076F IP=004B NU UP EI PL NZ NA PO NC
076F:004B 8A07 MOV AL,[BX] DS:0000=75
-t
AX=0775 BX=0000 CX=0003 DX=0000 SP=003A BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=076B CS=076F IP=004A NU UP EI PL NZ NA PO NC
076F:004A B400 MOV AH,00
-t
AX=0075 BX=0000 CX=0003 DX=0000 SP=003A BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=076B CS=076F IP=004C NU UP EI PL NZ NA PO NC
076F:004C F6360300 DIV BYTE PTR [0003] DS:0003=02
-t
AX=013A BX=0000 CX=0003 DX=0000 SP=003A BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=076B CS=076F IP=0050 NU UP EI PL NZ NA PO NC
076F:0050 50 PUSH AX
-a
```

图3.2.2 调用子程序

进入循环：

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Pro...
AX=013A BX=0000 CX=0003 DX=0000 SP=003A BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=076B CS=076F IP=0050 NU UP EI PL NZ NA PO NC
076F:0050 50 PUSH AX
-t
AX=013A BX=0000 CX=0003 DX=0000 SP=0038 BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=076B CS=076F IP=0051 NU UP EI PL NZ NA PO NC
076F:0051 3C00 CMP AL,00
-t
AX=013A BX=0000 CX=0003 DX=0000 SP=0038 BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=076B CS=076F IP=0053 NU UP EI PL NZ NA PE NC
076F:0053 75F5 JNZ 004A
-t
AX=013A BX=0000 CX=0003 DX=0000 SP=0038 BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=076B CS=076F IP=004A NU UP EI PL NZ NA PE NC
076F:004A B400 MOV AH,00
-t
AX=003A BX=0000 CX=0003 DX=0000 SP=0038 BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=076B CS=076F IP=004C NU UP EI PL NZ NA PE NC
076F:004C F6360300 DIV BYTE PTR [0003] DS:0003=02
-a
```

图3.2.3 进入循环

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Pro...
AX=003A BX=0000 CX=0003 DX=0000 SP=0038 BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=076B CS=076F IP=004C NU UP EI PL NZ NA PE NC
076F:004C F6360300 DIV BYTE PTR [0003] DS:0003=02
-t
AX=001D BX=0000 CX=0003 DX=0000 SP=0038 BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=076B CS=076F IP=0050 NU UP EI PL NZ NA PE NC
076F:0050 50 PUSH AX
-t
AX=001D BX=0000 CX=0003 DX=0000 SP=0036 BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=076B CS=076F IP=0051 NU UP EI PL NZ NA PE NC
076F:0051 3C00 CMP AL,00
-t
AX=001D BX=0000 CX=0003 DX=0000 SP=0036 BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=076B CS=076F IP=0053 NU UP EI PL NZ NA PE NC
076F:0053 75F5 JNZ 004A
-t
AX=001D BX=0000 CX=0003 DX=0000 SP=0036 BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=076B CS=076F IP=004A NU UP EI PL NZ NA PE NC
076F:004A B400 MOV AH,00
-a
```

图3.2.4 进入循环₂

屏幕打印:

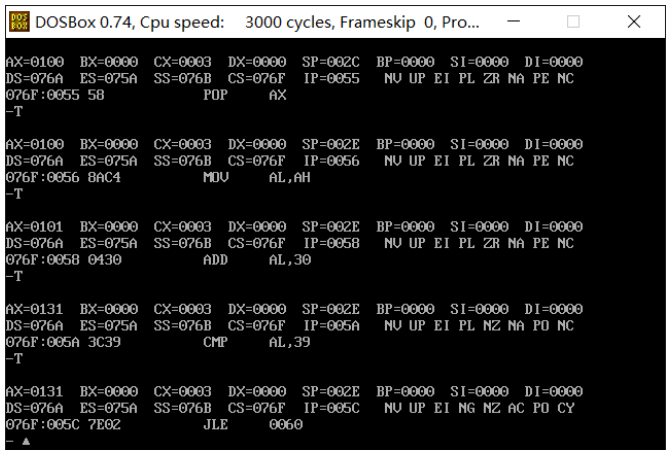


图3.2.5 转换为ASCII码

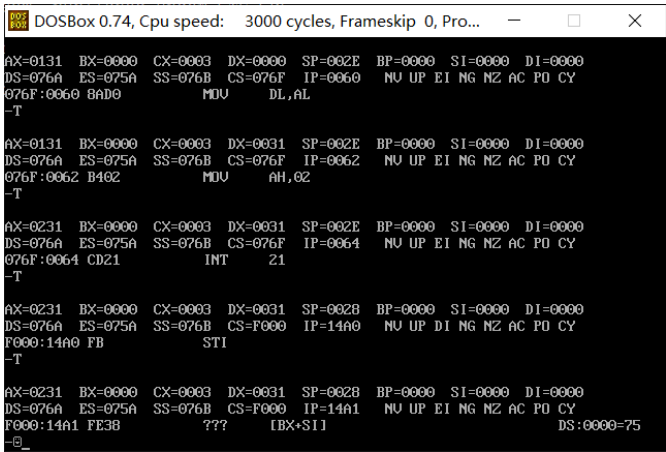


图3.2.6 屏幕打印

3. 3程序运行结果截图

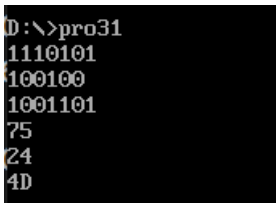


图 3.2.7 运行结果截图

4. 总结

4.1 遇到的问题

实验中，未考虑当 k 取较小数值如 2 时（不影响此次实验），调用 `div` 操作后商会超出 `ah` 所能表示的最大整数。遇到此类可能溢出的情况，可以用 `(dx,ax)` 的 32 位除以 16 位解决。在进行栈的循环出栈时，考虑栈为空的条件时忘记考虑调用子程序时，主程序的调用位置会入栈，占用一个栈空间。解决这个问题，可以将栈顶指针与栈初始值-2 做比较，从而控制循环。

4.2 收获

通过本次实验，熟练了对循环语句，调用子程序，编写宏以及调用 `DOS` 功能向屏幕输出一个字符的运用，加深了对理论的理解。同时加强了我对特殊边界情况的思考，并且学习了如何处理在实验过程中遇到的一些问题遇到的一些问题。