

四川 大学 计算机 学院、软件 学院
实 验 报 告

学号：____2021141460301____ 姓名：____杨怡____ 专业：__计算机科学与技术____
____ 班级：____行政六班____ 第 13 周

课程名称	汇编语言程序设计课程设计	实验课时	4
实验项目	编制完整的汇编语言源程序	实验时间	2022. 5. 18
实验目的	1. 掌握编写完整汇编语言源程序的方法，掌握汇编、连接等将源程序转换为可执行代码的必要步骤。 2. 在理解逻辑运算指令原理的基础上，按照实验要求的功能编写完整程序，生成可执行文件后，使用 DEBUG 对程序进行调试		
实验环境	文本编辑器，dosbox		
实 验 内 容 (算 法、程 序、步 骤 和 方法)	<p>一、编写一个完整源程序完成如下功能：</p> <p>1) 设定字节型变量 VAR1 的内容为 35H，编写程序将 VAR1 的高 4 位与低 4 位分离，分别作为低 4 位保存在 RES1 和 RES2 中, 这两个变量高 4 位置 0。</p> <p>2) 设定 32 位（二进制）数据 0E2597455H（注： 最高位为 E）保存在变量 VAR2 中，将该变量中的内容算术左移 2 位，所得结果仍然保存在 VAR2 中。</p> <p>二、代码如下：本实验代码如下：</p> <pre>DATAS SEGMENT var1 db 35h res1 db ? res2 db ? var2 dw 7455h dw 0e259h</pre>		

```
DATAS ENDS

CODES SEGMENT

ASSUME CS:CODES, DS:DATAS

START:

MOV AX, DATAS

MOV DS, AX

mov al, var1

mov res2, al

and res2, 0fH

mov res1, al

and res1, 0f0H

mov cl, 4

shr res1, cl

clc

shl var2, 1

rcl var2+2, 1

shl var2, 1

rcl var2+2, 1

MOV AH, 4CH

INT 21H

CODES ENDS
```

END START

- (1) 输入代码生成 test.asm 文件后，在 dosbox 中进行汇编

```
To adjust the emulated CPU speed, use ctrl-F11 and ctrl-F12.
To activate the keymapper ctrl-F1.
For more information read the README file in the DOSBox directory.

HAVE FUN!
The DOSBox Team http://www.dosbox.com

Z:\>SET BLASTER=A220 I7 D1 H5 T6

Z:\>mount c c:\dos
Drive C is mounted as local directory c:\dos\

Z:\>c:

C:\>masm test.asm;
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987. All rights reserved.

51718 + 464826 Bytes symbol space free

0 Warning Errors
0 Severe Errors

C:\>S_
```

- (2) 查看目录，生成 test.obj 文件

```
DOSBox 0.74, Cpu speed: 3000 cycles, Fra...
.. <DIR> 01-01-1980 0:00
DOSBOX-0 74 <DIR> 27-05-2021 20:44
MYDEBUG <DIR> 04-10-2018 11:04
α~1 <DIR> 04-03-2018 10:41
CREF EXE 15,830 12-05-1996 16:28
DEBUG EXE 20,634 14-04-2008 12:00
DOSBOX~1 EXE 1,448,809 23-03-2018 11:06
ERROUT EXE 9,499 12-05-1996 16:28
EXEMOD EXE 12,149 12-05-1996 16:28
EXEPACK EXE 14,803 12-05-1996 16:28
LIB EXE 32,150 12-05-1996 16:28
LINK EXE 39,100 12-05-1996 16:28
MAKE EXE 24,199 12-05-1996 16:28
MASM EXE 65,557 12-05-1996 16:28
README DOC 9,216 12-05-1996 16:28
SETENU EXE 10,601 12-05-1996 16:28
TEST ASM 463 27-05-2021 20:42
TEST EXE 579 27-05-2021 20:42
TEST LST 1,716 27-05-2021 20:38
TEST MAP 0 25-05-2021 8:47
TEST OBJ 218 27-05-2021 20:44
17 File(s) 1,705,523 Bytes.
5 Dir(s) 262,111,744 Bytes free.

C:\>S_
```

- (3) 使用 link 程序连接，生成 test.exe 文件

```
DOSBox 0.74, Cpu speed: 3000 cycles, Fra...
ERROUT  EXE      9,499 12-05-1996 16:28
EXEMOD   EXE     12,149 12-05-1996 16:28
EXEPACK  EXE     14,803 12-05-1996 16:28
LIB       EXE     32,150 12-05-1996 16:28
LINK      EXE     39,100 12-05-1996 16:28
MAKE      EXE     24,199 12-05-1996 16:28
MASM      EXE     65,557 12-05-1996 16:28
README   DOC       9,216 12-05-1996 16:28
SETENV    EXE     10,601 12-05-1996 16:28
TEST      ASM       463 27-05-2021 20:42
TEST      EXE       579 27-05-2021 20:42
TEST      LST      1,716 27-05-2021 20:38
TEST      MAP        0 25-05-2021  8:47
TEST      OBJ       218 27-05-2021 20:44
  17 File(s)      1,705,523 Bytes.
   5 Dir(s)      262,111,744 Bytes free.

C:\>link test.obj;

Microsoft (R) Overlay Linker  Version 3.60
Copyright (C) Microsoft Corp 1983-1987.  All rights reserved.

LINK : warning L4021: no stack segment

C:\>S_
```

(4) 使用 debug 程序进行调试，使用 u 命令查看程序指令

```
DOSBox 0.74, Cpu speed: 3000 cycles, Fra...
TEST      OBJ       218 27-05-2021 20:44
  17 File(s)      1,705,523 Bytes.
   5 Dir(s)      262,111,744 Bytes free.

C:\>link test.obj;

Microsoft (R) Overlay Linker  Version 3.60
Copyright (C) Microsoft Corp 1983-1987.  All rights reserved.

LINK : warning L4021: no stack segment

C:\>debug test.exe
-u
076B:0000 B86A07      MOV     AX,076A
076B:0003 8ED8          MOV     DS,AX
076B:0005 A00000      MOV     AL,[0000]
076B:0008 A20200      MOV     [0002],AL
076B:000B 80260200FF    AND     BYTE PTR [0002],0F
076B:0010 A20100      MOV     [0001],AL
076B:0013 80260100F0    AND     BYTE PTR [0001],F0
076B:0018 B104          MOV     CL,04
076B:001A D22E0100      SHR     BYTE PTR [0001],CL
076B:001E FB          CLC
076B:001F D1260300      SHL     WORD PTR [0003],1
-S_
```

数据
记录
和计
算

- 1、将数据段基值赋给 ds，mov 指令不影响标志位
- 2、执行指令 mov al,[0000] 和 mov [0002],al；将数据从 var1 传送到 res2 中，mov 指令不影响标志位。

```
DOSBox 0.74, Cpu speed: 3000 cycles, Fra...
075A:0040 05 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
075A:0050 CD 21 CB 00 00 00 00 00-00 00 00 00 00 00 00 .!.....
075A:0060 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
075A:0070 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
-t
AX=076A BX=0000 CX=0043 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=076B IP=0003 NU UP EI PL NZ NA PO NC
076B:0003 8EDB          MOV     DS,AX
-t
AX=076A BX=0000 CX=0043 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=0769 CS=076B IP=0005 NU UP EI PL NZ NA PO NC
076B:0005 A00000       MOV     AL,[0000]          DS:0000=35
-t
AX=0735 BX=0000 CX=0043 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=0769 CS=076B IP=0008 NU UP EI PL NZ NA PO NC
076B:0008 A20200       MOV     [0002],AL          DS:0002=00
-t
AX=0735 BX=0000 CX=0043 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=0769 CS=076B IP=000B NU UP EI PL NZ NA PO NC
076B:000B 802602000F    AND     BYTE PTR [0002],0F    DS:0002=35
-S_
```

- 3、执行指令 and byte ptr [0002],0F，将 res2 的高四位清零，在 res2 保留 var1 中的低四位,res2 中的值为 05H.CF、0F 被强行置为 0,AF 不确定,SF=0,ZF=0,PF=1；

```
DOSBox 0.74, Cpu speed: 3000 cycles, Fra...
AX=0735 BX=0000 CX=0043 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=0769 CS=076B IP=0008 NU UP EI PL NZ NA PO NC
076B:0008 A20200       MOV     [0002],AL          DS:0002=00
-t
AX=0735 BX=0000 CX=0043 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=0769 CS=076B IP=000B NU UP EI PL NZ NA PO NC
076B:000B 802602000F    AND     BYTE PTR [0002],0F    DS:0002=35
-t
AX=0735 BX=0000 CX=0043 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=0769 CS=076B IP=0010 NU UP EI PL NZ NA PE NC
076B:0010 A20100       MOV     [0001],AL          DS:0001=00
-t
AX=0735 BX=0000 CX=0043 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=0769 CS=076B IP=0013 NU UP EI PL NZ NA PE NC
076B:0013 80260100F0    AND     BYTE PTR [0001],F0    DS:0001=35
-t
AX=0735 BX=0000 CX=0043 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=0769 CS=076B IP=0018 NU UP EI PL NZ NA PE NC
076B:0018 B104          MOV     CL,04
-S_
```

- 4、执行指令 mov [0001], al；将数据从 var1 传送到 res1 中，mov 指令不影响标志位。
- 5、执行指令 and byte ptr [0001],F0，将 res1 的低四位清零，在 res1 中保

留 var1 中的高四位, res1 中的值为 30H。CF、OF 被强行置为 0, AF 不确定, SF=0, ZF=0, PF=1;

6、执行指令 mov cl,4 和 shr byte ptr [0001],cl 后, res1 的值变为 03h, OF=0, CF=0, ZF=0, SF=0, PF=1, AF 不确定。

```
DOSBox 0.74, Cpu speed: 3000 cycles, Fra...
AX=0735 BX=0000 CX=0043 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=0769 CS=076B IP=0010 NU UP EI PL NZ NA PE NC
076B:0010 A20100 MOV [0001],AL DS:0001=00
-t
AX=0735 BX=0000 CX=0043 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=0769 CS=076B IP=0013 NU UP EI PL NZ NA PE NC
076B:0013 80260100F0 AND BYTE PTR [0001],F0 DS:0001=35
-t
AX=0735 BX=0000 CX=0043 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=0769 CS=076B IP=0018 NU UP EI PL NZ NA PE NC
076B:0018 B104 MOV CL,04
-t
AX=0735 BX=0000 CX=0004 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=0769 CS=076B IP=001A NU UP EI PL NZ NA PE NC
076B:001A D22E0100 SHR BYTE PTR [0001],CL DS:0001=30
-t
AX=0735 BX=0000 CX=0004 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=0769 CS=076B IP=001E NU UP EI PL NZ AC PE NC
076B:001E F8 CLC
-S_
```

7、查看内存单元, [0001]的数据为 03H, [0002]的数据为 05H

```
-t
AX=0735 BX=0000 CX=0004 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=0769 CS=076B IP=0018 NU UP EI PL NZ NA PE NC
076B:0018 B104 MOV CL,04
-t
AX=0735 BX=0000 CX=0004 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=0769 CS=076B IP=001A NU UP EI PL NZ NA PE NC
076B:001A D22E0100 SHR BYTE PTR [0001],CL DS:0001=30
-t
AX=0735 BX=0000 CX=0004 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=0769 CS=076B IP=001E NU UP EI PL NZ AC PE NC
076B:001E F8 CLC
-d ds:0000
076A:0000 35 03 05 55 74 59 E2 00-00 00 00 00 00 00 00 00 5..UtY.....
076A:0010 B8 6A 07 BE D8 A0 00 00-A2 02 00 80 26 02 00 0F .j.....&...
076A:0020 A2 01 00 B0 26 01 00 F0-B1 04 D2 2E 01 00 F8 D1 ...&.....
076A:0030 26 03 00 D1 16 05 00 D1-26 03 00 D1 16 05 00 B4 &.....&.....
076A:0040 4C CD 21 E8 EA 48 83 C4-04 50 E8 7B 0E 83 C4 04 L.?.H...P.f...
076A:0050 3D FF FF 74 03 E9 ED 00-C4 5E FC 26 8A 47 0C 2A =..t.....^.&.G.*
076A:0060 E4 40 50 BB C3 BC C2 05-0C 00 52 50 E8 C1 48 83 .0P.....RP..H.
076A:0070 C4 04 50 BD 86 FA FE 50-E8 17 73 83 C4 06 8B B6 ..P....P..s....
-S_
```

8、对 CF 标志位清零, 下面进行 32 位数据 0E2597455H 的算数左移两位

9、执行完指令 shl word ptr [0003],1 后, VAR2 变为 E8AA, 发生溢出, OF=1, 左边移出最低位 0 保存到 CF 中, CF=0

10、执行完指令 `rcl word ptr [0005],1` 后, `VAR2+2` 变为 `C4B2`, 未发生溢出, `OF=0`, `CF=1`.

```

DOSBox 0.74, Cpu speed: 3000 cycles, Fra...
-t
AX=0735 BX=0000 CX=0004 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=0769 CS=076B IP=001F NU UP EI PL NZ AC PE NC
076B:001F D1260300 SHL WORD PTR [0003],1 DS:0003=7455
-t
AX=0735 BX=0000 CX=0004 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=0769 CS=076B IP=0023 OU UP EI NG NZ AC PE NC
076B:0023 D1160500 RCL WORD PTR [0005],1 DS:0005=E259
-t
AX=0735 BX=0000 CX=0004 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=0769 CS=076B IP=0027 NU UP EI NG NZ AC PE CY
076B:0027 D1260300 SHL WORD PTR [0003],1 DS:0003=E8AA
-d ds:0000
076A:0000 35 03 05 AA E8 B2 C4 00-00 00 00 00 00 00 00 00 5.....
076A:0010 B8 6A 07 8E D8 A0 00 00-A2 02 00 80 26 02 00 0F .j.....&...
076A:0020 A2 01 00 80 26 01 00 F0-B1 04 D2 2E 01 00 F8 D1 ...&.....
076A:0030 26 03 00 D1 16 05 00 D1-26 03 00 D1 16 05 00 B4 &.....&.....
076A:0040 4C CD 21 E8 EA 48 83 C4-04 50 E8 7B 0E 83 C4 04 L.!..H...P.(!...
076A:0050 3D FF FF 74 03 E9 ED 00-C4 5E FC 26 8A 47 0C 2A =.t.....^.&.G.*
076A:0060 E4 40 50 8B C3 8C C2 05-0C 00 52 50 E8 C1 48 83 .@P.....RP..H.
076A:0070 C4 04 50 8D 86 FA FE 50-E8 17 73 83 C4 06 8B B6 ..P....P..s....
-S

```

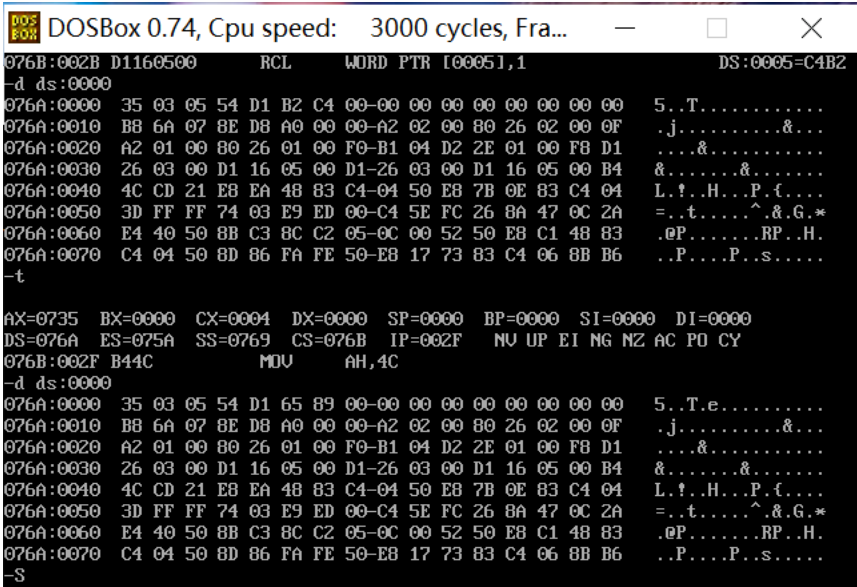
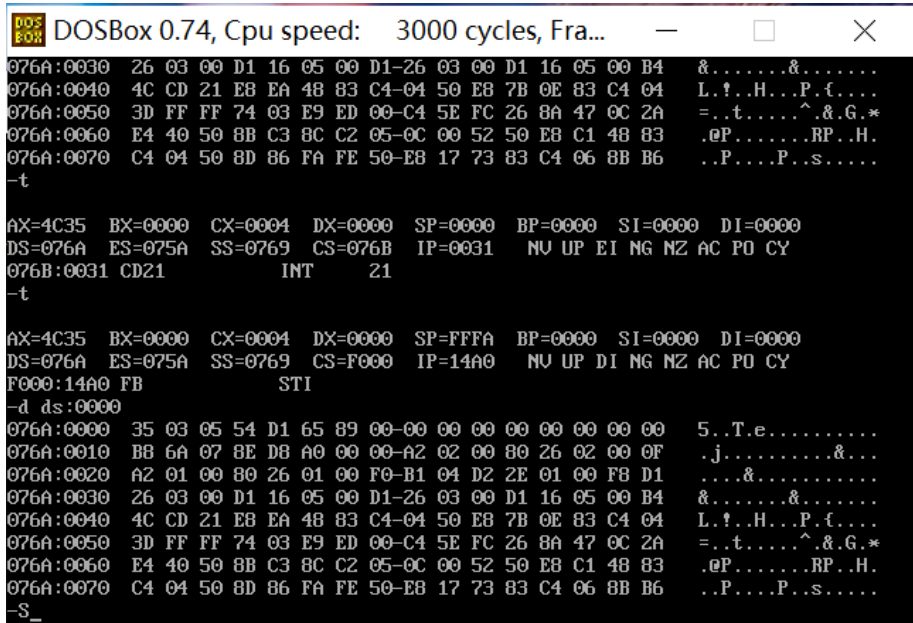
11、执行指令 `SHL WORD PTR [0003],1`, `var2` 变为 `D154`, 未发生溢出, `OF=0`, 左边移出最低位 1 保存到 `CF` 中, `CF=1`

```

DOSBox 0.74, Cpu speed: 3000 cycles, Fra...
076B:0027 D1260300 SHL WORD PTR [0003],1 DS:0003=E8AA
-d ds:0000
076A:0000 35 03 05 AA E8 B2 C4 00-00 00 00 00 00 00 00 00 5.....
076A:0010 B8 6A 07 8E D8 A0 00 00-A2 02 00 80 26 02 00 0F .j.....&...
076A:0020 A2 01 00 80 26 01 00 F0-B1 04 D2 2E 01 00 F8 D1 ...&.....
076A:0030 26 03 00 D1 16 05 00 D1-26 03 00 D1 16 05 00 B4 &.....&.....
076A:0040 4C CD 21 E8 EA 48 83 C4-04 50 E8 7B 0E 83 C4 04 L.!..H...P.(!...
076A:0050 3D FF FF 74 03 E9 ED 00-C4 5E FC 26 8A 47 0C 2A =.t.....^.&.G.*
076A:0060 E4 40 50 8B C3 8C C2 05-0C 00 52 50 E8 C1 48 83 .@P.....RP..H.
076A:0070 C4 04 50 8D 86 FA FE 50-E8 17 73 83 C4 06 8B B6 ..P....P..s....
-t
AX=0735 BX=0000 CX=0004 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=0769 CS=076B IP=002B NU UP EI NG NZ AC PD CY
076B:002B D1160500 RCL WORD PTR [0005],1 DS:0005=C4B2
-d ds:0000
076A:0000 35 03 05 54 D1 B2 C4 00-00 00 00 00 00 00 00 00 5..T.....
076A:0010 B8 6A 07 8E D8 A0 00 00-A2 02 00 80 26 02 00 0F .j.....&...
076A:0020 A2 01 00 80 26 01 00 F0-B1 04 D2 2E 01 00 F8 D1 ...&.....
076A:0030 26 03 00 D1 16 05 00 D1-26 03 00 D1 16 05 00 B4 &.....&.....
076A:0040 4C CD 21 E8 EA 48 83 C4-04 50 E8 7B 0E 83 C4 04 L.!..H...P.(!...
076A:0050 3D FF FF 74 03 E9 ED 00-C4 5E FC 26 8A 47 0C 2A =.t.....^.&.G.*
076A:0060 E4 40 50 8B C3 8C C2 05-0C 00 52 50 E8 C1 48 83 .@P.....RP..H.
076A:0070 C4 04 50 8D 86 FA FE 50-E8 17 73 83 C4 06 8B B6 ..P....P..s....
-S

```

12、执行指令 `RCL WORD PTR [0005],1`, `var2+2` 变为 `8965`, 未发生溢出, `OF=0`, 将 `CF` 看作移位数据的最高位, `CF=1`

	<div data-bbox="267 231 1129 815"></div> <p>13、输入命令 q 退出 debug</p>
结论 (结果)	<div data-bbox="267 938 1185 1558"></div> <p>该程序完成了“设定字节型变量 VAR1 的内容为 35H，编写程序将 VAR1 的高 4 位与低 4 位分离，分别作为低 4 位保存在 RES1 和 RES2 中,这两个变量高 4 位置 0。设定 32 位（二进制）数据 0E2597455H（注：最高位为 E）保存在变量 VAR2 中，将该变量中的内容算术左移 2 位，所得结果仍然保存 VAR2 中”的目的，运行结束后[0001]中的数据为 03H，[0002]中的数据为 05H；变量 var2 中的数据为</p>

	8965D154.
小 结	通过这次实验，我不仅掌握了编写完整汇编语言源程序的方法以及汇编、连接等将源程序转换为可执行代码的必要步骤，而且在理解逻辑运算指令原理的基础上，按照实验要求的功能成功编写了完整程序，生成了可执行文件并使用 DEBUG 对程序进行调试。通过这次实验，使我对汇编有了更深刻的见解，更好的理解和掌握了相关的知识。
指导 老师 评 议	<div>成绩评定：</div> <div>指导教师签名：</div>