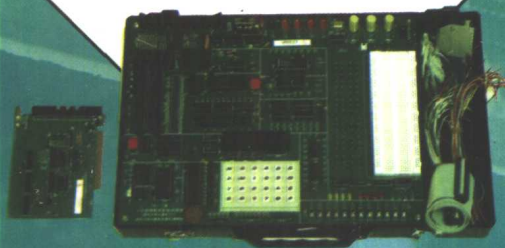


仇玉章
冯一兵
编 著

32位 微型计算机 原理 与 接口技术 实验指导



清华大学出版社

<http://www.tup.tsinghua.edu.cn>



32 位微型计算机原理与接口技术 实 验 指 导

仇玉章 冯一兵 编著

清 华 大 学 出 版 社

(京)新登字 158 号

内 容 简 介

本书是《32 位微型计算机原理与接口技术》一书(仇玉章主编,清华大学出版社出版)的配套教材,也是“汇编语言程序设计”、“微型计算机原理”、“微型计算机接口技术”等课程的辅助教材。

本书从应用的角度出发,紧密结合教学内容,针对汇编语言程序设计、接口电路实验、课程设计等环节,选编了 42 个软硬件实验课题和 11 个指导性例题。对于各类接口电路芯片,首先从应用的角度对理论知识进行小结,继而给出有指导性的例题,详细介绍设计思路,最后选编实验课题。

接口电路实验以“TPC-1 型十六位微机实验培训系统”为实验设备。书中详细介绍了各类接口的电路设计,对于从事微机系统软、硬件开发的工程技术人员,也是一本很好的参考书。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

32 位微型计算机原理与接口技术实验指导/仇玉章,冯一兵编著. —北京:清华大学出版社,2000

ISBN 7-302-03881-3

I. 32… II. ①仇…②冯… III. ①微型计算机-理论②微型计算机-接口 IV. TP36

中国版本图书馆 CIP 数据核字(2000)第 28963 号

出版者:清华大学出版社(北京清华大学学研大厦,邮编 100084)

<http://www.tup.tsinghua.edu.cn>

印刷者:北京市清华园胶印厂

发行者:新华书店总店北京发行所

开 本:787×1092 1/16 印张:4.75 字数:116 千字

版 次:2000 年 7 月第 1 版 2000 年 11 月第 2 次印刷

书 号:ISBN 7-302-03881-3/TP·2332

印 数:6001~11000

定 价:6.00 元

前 言

本书是《32 位微型计算机原理与接口技术》一书的配套教材,也是“汇编语言程序设计”、“微型计算机原理”、“微型计算机接口技术”等课程的辅助教材。

本书共分 4 章,内容如下:

第 1 章汇编语言程序设计,按照程序设计的类型分为 6 小节。

第 2 章详细介绍了实验设备上的接口电路设计。

第 3 章是接口实验。本章根据接口的类型分为 6 小节。对于每一种接口电路,首先从应用的角度对理论知识进行小结,然后给出指导性例题,最后选编实验课题。接口实验以清华同方教学仪器设备公司和清华大学计算机系微机教研组共同研制的“TPC-1 型十六位微机实验培训系统”为实验设备。

第 4 章是课程设计选题。

本书共选编了 42 个软硬件实验课题,11 个指导性例题。实验课题既有一定代表性也有一定深度,学生应在教师指导下,选做其中的一部分。

本书第 1 章、第 3 章和第 4 章由仇玉章老师编写,第 2 章由冯一兵老师编写。

由于编者水平所限,难免有错漏之处,敬请读者指正。

作 者

1999 年 12 月

目 录

第 1 章 汇编语言程序设计	1
1.1 汇编语言语法练习	1
1.2 循环与子程序调用	2
实验 1 统计负数的个数	2
实验 2 BCD 码数→二进制数显示	2
1.3 字符串操作	3
实验 1 通行字识别	3
实验 2 通行字识别	3
实验 3 显示器色彩验证	3
1.4 代码转换	4
1.4.1 代码转换举例	4
1.4.2 代码转换实验	6
实验 1 二进制数→十六进制数显示	6
实验 2 二进制数→十进制数显示	6
实验 3 十进制数→二进制数显示	7
实验 4 十进制数→十六进制数显示	7
实验 5 十六进制数→二进制数显示	8
1.5 磁盘文件操作	8
实验 1 文件拷贝	8
实验 2 文本文件的分页显示	8
1.6 图形方式程序设计	8
1.6.1 图形方式程序设计基础	8
1.6.2 图形方式程序设计举例	10
1.6.3 图形方式编程实验	18
实验 1 编程显示奥运会五环标志	18
实验 2 画圆球	18
实验 3 动态画圆球	18
实验 4 画椭圆	18
第 2 章 TPC-1 型十六位微机实验培训系统	19
2.1 概述	19
2.2 公共控制电路	19
2.3 辅助电路	22
2.4 接口实验电路	23

第3章 接口实验	30
3.1 数码管显示	30
3.1.1 数码管电路工作原理	30
3.1.2 数码管显示实验	32
实验1 2位数码管交替显示	32
实验2 6位数码管静态显示	32
实验3 6位数码管静态/动态混合显示	32
实验4 6位数码管动态显示	32
3.2 定时器与定时中断	33
3.2.1 I8253 应用小结	33
3.2.2 I8253 应用举例	34
3.2.3 I8253 定时器实验	42
实验1 中断方式的数码管交替显示	42
实验2 中断方式的数码管静态/动态混合显示	42
实验3 中断方式的数码管动态显示	42
实验4 设计数码管电子表	42
实验5 系统定时中断实验	42
实验6 设计屏显电子表	42
实验7 菜单选择的音乐程序	43
实验8 菜单选择的音乐程序	44
3.3 串行异步通信	44
3.3.1 串行通信小结	44
3.3.2 8251 串行通信举例	45
3.3.3 串行通信实验	48
实验1 8251 自发自收实验	48
实验2 测试微机系统串行口	49
3.4 并行接口	49
3.4.1 8255 应用小结	49
3.4.2 8255 双向通信举例	51
3.4.3 8255 并行口实验	54
实验1 8255 选通型输出实验	54
实验2 小键盘按键识别	55
实验3 小键盘按键识别	56
3.5 数模、模数转换	57
3.5.1 数模转换实验举例	57
3.5.2 模数转换电路的分析	58
3.5.3 数模、模数转换实验	59
实验1 数模转换实验	59
实验2 单路模数转换实验	59

实验 3 3 路模数转换实验	60
3.6 DMA 传送	60
3.6.1 8237A 应用小结	60
3.6.2 DMA 传送实验举例	61
3.6.3 DMA 传送实验	66
第 4 章 课程设计选题	67
选题 1 微机系统串行口自动测试程序	67
选题 2 设计两级下拉式菜单	67
选题 3 游戏程序——小球对撞	69
选题 4 设计指针式的屏显电子钟	69
选题 5 汉字菜单歌曲选唱程序	70
选题 6 TPC-1 型十六位微机实验培训系统综合测试软件	70

第 1 章 汇编语言程序设计

1.1 汇编语言语法练习

【实验目的】

学习汇编源程序的编辑、编译、链接、执行全过程。

学习编辑文件、TASM、TLINK 的使用方法以及汇编语言的语法规则。

【实验内容】

下面的程序采用比较法,将一个 8 位二进制数转换成十进制数显示,其设计方法拙劣,语法错误很多,希望以它为蓝本,练习源程序的编辑、编译和链接。

读者应照原样编辑源程序,汇编之后,必定给出若干错误信息,再根据错误提示,修改源程序,直到汇编后没有错误时为止,然后进行链接、执行。正确的执行结果是在屏幕上显示:

100+9=109

[程序清单]

```
                ;FILENAME:EXA111.ASM
                .486
DATA            SEGMENT USE16
N1              DW      0809H,AB
N2              DB      100
SUM             DB      ?
MSG            DB      '100+9='
                DB      30H,30H,30H                ;3个0的ASCII码
DATA            ENDS
CODE            SEGMENT USE16
                ASSUME  CS:CODE,DS:DATA
BEG:            MOV     DS,DATA                ;段基址→DS
                MOV     BX,OFFSET SUM
                MOV     AH,N1                  ;9→AH
                MOV     AL,N2                  ;100→AL
                ADD     AH,AL
                MOV     [BX],AH                ;和数→SUM单元
                CALL    N2-10
                MOV     AH,9
                MOV     DX,OFFSET MSG
                INT     21H
                MOV     AH,4CH
                INT     21H
N2-10:          PROC                                ;二进制→十进制ASCII码
```

```

LAST:   CMP      [BX],100           ;和数与 100 比较
        JC       NEXT1             ;小于 100 转
        SUB      [BX],100          ;否则,和数减 100
        INC      [BX+7]            ;百位数加 1
        JMP      LAST
NEXT1:  CMP      [BX],10            ;余数与 10 比较
        JC       NEXT2             ;小于 10 转
        SUB      [BX],10           ;否则,余数减 10
        INC      [BX+8]            ;十位数加 1
        JMP      NEXT1
NEXT2:  ADD      [BX+9],SUM          ;生成个位数
        RET
N2-10:  ENDP
CODE    ENDS
        END      BEG

```

1.2 循环与子程序调用

【实验目的】

复习有符号数的表示与鉴别。

学习循环程序与子程序调用的编程方法。

【实验 1】 统计负数的个数

设数据段有 8 个有符号数：

```
NUM    DB    -19,+28,37,-46,+55,61,-74,+255
```

设计一个程序(不是两个程序)完成两项要求：

- ① 统计并且显示负数的个数。
- ② 找出真值最大的数,并以十六进制的形式,显示在屏幕上,显示格式为：

MAX=××H

提示：

- ① 正确的答案是有 4 个负数,如果你得到 3 个负数,肯定是错误的。
- ② 最大的真值数应是 3DH。
- ③ 为了理解汇编语言是怎样对数据进行汇编的,建议在对源程序进行汇编时,键入汇编参数“/L”,即键入：

TASM 源文件名/L

键入汇编参数“/L”的目的,是在生成 OBJ 文件的同时,也生成 LST 文件。再用编辑文件显示 LST 文件的内容,就可以看到 NUM 单元的 8 个数经汇编后生成的机器数了。

【实验 2】 BCD 码数→二进制数显示

设数据段有两个 BCD 码数：

```

N1    DB    10010101
N2    DB    01110011

```

要求采用子程序调用的方法,将它们转换成二进制数,并按下列格式显示在屏幕上(这也是正确答案):

N1=01011111B

N2=01001001B

1.3 字符串操作

【实验目的】

学习串操作指令的应用、DOS 功能调用与文本方式下的彩色字符串显示以及宏指令调用。

【实验 1】 通行字识别

程序执行后,给出简单明了的操作提示,请用户给出“通行字”。只有当用户键入的字符串和程序内定的字符串相同时,程序才能返回 DOS。

实验 1 设计要求:

- ① 彩色界面,界面清晰美观,色彩协调。
- ② 使用 DOS 系统的 0AH 号子功能接收用户键入的字符串。
- ③ 建议按图 1.1(a)设计用户界面,颜色自选。

【实验 2】 通行字识别

实验 2 设计要求:

- ① 彩色界面,界面清晰美观,色彩协调。
- ② 使用 DOS 系统的单字符输入子功能,配合循环程序接收用户键入的字符串。
- ③ 为了保密,用户键入的字符串不应当如实地回显在屏幕上,为了真正起到通行字的作用,程序在接收键入的过程中,不应当响应 Ctrl_C。
- ④ 建议按图 1.1(b)设计用户界面,颜色自选。



图 1.1 通行字识别程序的用户界面

注意:

- ① 在公共实验室的计算机上,不应当把上述的实验程序写入系统的自动批处理文件中。
- ② 单线框和双线框均由若干 CRT 显示符组成,相应的 CRT 显示符(十进制数)已经标注在图 1.2 中。

【实验 3】 显示器色彩验证

设计一个文本方式下的显示器色彩验证程序。

设计要求:

- ① 分别显示 3×5 个彩色方块,方块的颜色为蓝、绿、青、红、品红、棕、灰白、灰、浅蓝、浅绿、浅青、浅红、浅品红、黄、白。

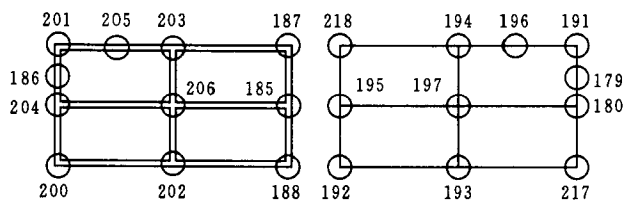


图 1.2 CRT 显示符生成的表格

- ② 每一个方块面积为 7 行×14 列。
- ③ 每隔 0.5 秒显示一个方块,并发出短暂的音响,一屏显示完毕清屏再显。
- ④ 主机键盘按下任意键时,停止演示。

编程提示:

- ① 一个方块由 7×14 个小矩形块组成。一个小矩形块的 CRT 显示码等于十进制数 219。
- ② 向屏幕输出一个“报警”字符(ASCII 码为 07),即发出短暂的音响。
- ③ 为了避免多重循环,必须减少循环层次,建议把每一个大方块左上角的行列值组成一张表,通过查表的方法确定方块的屏显位置。

1.4 代码转换

1.4.1 代码转换举例

【例 1.4.1】十六进制数→十进制数显示

将键盘输入的 4 位十六进制数转换成等值的十进制数送屏幕显示。

设计要求:

- ① 程序执行后,首先给出操作提示:

Please input 4 hexadecimal numbers

② 程序要有保护措施,对于非法键入不受理、不回显,也不显示错误信息,但要响铃以示警告。

- ③ 实时地显示合法键入的数据,当收到第 4 位合法数据后,立即显示转换结果。

- ④ 显示格式示范如下:

ABCDH=43981

[设计思路]

(1) 程序通过功能调用得到的键入数据均是键入字符的 ASCII 码。程序送往屏幕显示的数据,其每一个数都必须是该数的 ASCII 码。从这个意义上讲,键盘和显示器均是 ASCII 码外设。

(2) 根据设计要求,程序应首先判别键入的数据是否在'0'~'9'和'A'~'F'之间,不在这个范围就是非法键入。

(3) DOS 系统的 7 号和 8 号子功能,对键入的字符没有回显功能,如果键入的字符是合法数据,再用单字符输出的子功能“回显”合法数据,即可达到“实时地显示合法键入的数据”

这一设计要求。

(4) 代码转换的方法: 首先把键入的十六进制数 ASCII 码, 转换成等值的二进制数, 然后再把二进制数转换成十进制数。

[程序清单]

```
                ,FILENAME,EXA141.ASM
                .486
DISP            MACRO   NNN
                MOV     AH,0EH
                MOV     AL,NNN
                INT     10H
                ENDM

DATA            SEGMENT USE16
MSG             DB      'Please input 4 hexadecimal numbers',0DH,0AH,'$'
BINARY         DW      0
DATA            ENDS
CODE            SEGMENT USE16
                ASSUME  CS:CODE,DS:DATA
BEG:            MOV     AX,DATA
                MOV     DS,AX
                MOV     AH,9
                MOV     DX,OFFSET MSG
                INT     21H
                MOV     CH,4
WAIT_IN:        MOV     AH,08H
                INT     21H
                MOV     BL,AL
                CMP     AL,'0'
                JL      BEL                ;非法键入转
                CMP     AL,'9'
                JA      NEXT1
                JMP     NEXT2
NEXT1:          CMP     AL,'A'
                JL      BEL                ;非法键入转
                CMP     AL,'F'
                JG      BEL                ;非法键入转
                SUB     BL,7
NEXT2:          SUB     BL,30H              ;生成二进制数
                DISP    AL                ;显示位代码
                SAL     BL,4
                MOV     CL,4
LAST1:          SAL     BL,1
                RCL     BINARY,1          ;生成二进制数
```

```

DEC      CL
JNZ      LAST1
DEC      CH
JNZ      WAIT_IN
DISP     'H'                      ;显示'H'
DISP     '='                      ;显示'='
MOV      CX,0
MOV      AX,BINARY
MOV      BX,10
LAST2:   MOV      DX,0
DIV      BX
PUSH     DX                      ;余数压栈
INC      CX
CMP      AX,0
JNE      LAST2                  ;商不为0转
AGA:     POP      DX
ADD      DL,30H
DISP     DL                      ;依次显示十进制数
LOOP     AGA
MOV      AH,4CH
INT      21H
BEL:     DISP     7              ;告警
JMP      WAIT_IN
CODE     ENDS
END      BEG

```

1.4.2 代码转换实验

【实验目的】

掌握代码的输入、转换与显示的方法。

【实验 1】 二进制数→十六进制数显示

从键盘输入 16 位二进制数,然后转换成等值的十六进制数显示。

设计要求:

① 程序执行后,给出下列操作提示:

Please input 16 binary numbers

② 程序要有保护措施,对于非法键入不受理,不回显,也不显示错误信息,但要响铃以示警告。

③ 实时地显示合法键入的数据,当收到第 16 位 0/1 代码后,立即显示转换结果。

④ 显示格式示范如下:

0111100010011010=789AH

【实验 2】 二进制数→十进制数显示

从键盘输入 16 位二进制数,然后转换成等值的十进制数显示。

设计要求:

- ① 程序执行后,给出下列操作提示:

Please input 16 binary numbers

② 程序要有保护措施,对于非法键入不受理,不回显,也不显示错误信息,但要响铃以示警告。

- ③ 实时地显示合法键入的数据,当收到第 16 位 0/1 代码后,立即显示转换结果。

- ④ 显示格式示范如下:

0101010101010101=21845

【实验 3】十进制数→二进制数显示

从键盘输入 5 位十进制数,然后转换成等值的二进制数显示。

- (1) 设计要求

- ① 程序执行后,给出下列操作提示:

Please input 5 decimal numbers

② 程序要有保护措施,对于非法键入不受理,不回显,也不显示错误信息,但要响铃以示警告。

- ③ 实时地显示合法键入的数据,当收到第 5 位十进制数后,立即显示转换结果。

- ④ 显示格式示范如下:

98765=11000000111001101

- (2) 算法提示

- ① 设 5 位十进制数的数列为:

$N = D_4 D_3 D_2 D_1 D_0$

则等值的二进制数 $= D_4 \times 10000 + D_3 \times 1000 + D_2 \times 100 + D_1 \times 10 + D_0 \times 1$ (算法 1)

$= (((((0 + D_4) \times 10 + D_3) \times 10 + D_2) \times 10 + D_1) \times 10 + D_0$ (算法 2)

- ② 最大的 5 位十进制数为 99999,其等值的二进制数为 17 位。

- (3) 编程提示

- ① 按算法 1 编程。

设计一张含有 10000,1000,100,10,1 的“权值表”,依次取出权值与相应的位代码相乘(使用 32 位乘法指令),再求累加和。

- ② 按算法 2 编程。

关键是设计一个 $EBX \times 10 + EDX \rightarrow EBX$ 的子程序供调用,EBX 初值为 0。

第 1 次调用前,令 $EDX = D_4$

第 2 次调用前,令 $EDX = D_3$

.....

第 5 次调用前,令 $EDX = D_0$

【实验 4】十进制数→十六进制数显示

从键盘输入 5 位十进制数,然后转换成等值的 5 位十六进制数显示。

设计要求:

- ① 程序执行后,给出下列操作提示:

Please input 5 decimal numbers

② 程序要有保护措施,对于非法键入不受理,不回显,也不显示错误信息,但要响铃以示警告。

③ 实时地显示合法键入的数据,当收到第 5 位十进制数后,立即显示转换结果。

④ 显示格式示范如下:

98765=181CDH

【实验 5】 十六进制数→二进制数显示

从键盘输入 4 位十六进制数,然后转换成等值的二进制数显示。

设计要求:

① 程序执行后,给出下列操作提示:

Please input 4 hexadecimal numbers

② 程序要有保护措施,对于非法键入不受理,不回显,也不显示错误信息,但要响铃以示警告。

③ 实时地显示合法键入的数据,当收到第 4 位合法数据后,立即显示转换结果。

④ 显示格式示范如下(4 位二进制数为一组,用空格做间隔符):

3A4BH=0011 1010 0100 1011

1.5 磁盘文件操作

【实验目的】

学习磁盘文件操作的程序设计方法。

【实验 1】 文件拷贝

设计一个文件拷贝程序。

设计要求:

① 通过人机会话获得源文件和目标文件的文件名。

② 仿照 DOS 系统给出错误信息,例如:

当源文件不存在时,给出错误信息:

File not found ——— 源文件名

当目标文件已经存在时,给出提示信息:

Overwrite 目标文件名(Yes/No)?

【实验 2】 文本文件的分页显示

仿照 DOS 命令“TYPE 文件名 | MORE”的功能,设计一个文本文件的分页显示程序。

设计要求:

(1) 通过命令行参数给出待显示的文件名。

(2) 每屏显示 24 行,敲击任意键之后,再显下一屏。

1.6 图形方式程序设计

1.6.1 图形方式程序设计基础

图形方式下的程序设计,首先要设置屏幕工作在图形方式。在图形方式下屏幕被划分为

M 列 $\times N$ 行个“小方格”，每一个“小方格”称为一个“像点”，把不同的像素值“写”到小方格中，像点就有了不同的颜色。基本图形有点、直线、斜线、空心圆、实心圆、方块等等，这些基本图形都是由“写像点”操作完成的。

为了本小节的完整性，我们列出有关的“图形方式 BIOS 功能”，便于读者在实验中查阅。

INT 10H 中的图形方式功能调用：

【功能号 00H】 设置屏幕显示方式

模式	分辨率	工作方式	色数	显示卡
入口参数：AL=4	320 $\times$ 200	图形方式	4	CGA
AL=5	320 $\times$ 200	图形方式	2	CGA
AL=6	640 $\times$ 200	图形方式	2	CGA
AL=0DH	320 $\times$ 200	图形方式	16	EGA
AL=0EH	640 $\times$ 200	图形方式	16	EGA
AL=0FH	640 $\times$ 350	图形方式	4	EGA
AL=10H	640 $\times$ 350	图形方式	16	EGA
AL=11H	640 $\times$ 480	图形方式	2	VGA
AL=12H	640 $\times$ 480	图形方式	16	VGA
AL=13H	320 $\times$ 200	图形方式	256	VGA

出口参数：无

注：分辨率=列 $\times$ 行。

CGA (color graphics adapter)为彩色图形接口卡。EGA (enhanced graphics adapter)为增强图形显示卡。VGA (video graphics array)为视频图形阵列。EGA 兼容 CGA,VGA 兼容 EGA 和 CGA。

【功能号 0BH】 选择屏幕背景色/选择调色板

[选择屏幕背景色]

入口参数：BH=0,BL=像素值 0~15

若 BL=	0	1	2	3	4	5	6	7
则对应背景色为：	黑	蓝	绿	青	红	品红	棕	灰白
若 BL=	8	9	10	11	12	13	14	15
则对应背景色为：	灰	浅蓝	浅绿	浅青	浅红	浅品红	黄	白

[选择 CGA 调色板]

入口参数：BH=1,BL=0 选择 0 号调色板

BH=1,BL=1 选择 1 号调色板

【功能号 0CH】 写像点

入口参数：AL=像素值,BH=0,CX=像点所在的列号,DX=像点所在的行号。

说明：

① 实验表明，对于 VGA 卡模式 4、5 都是 320 $\times$ 200,4 色图形方式。

② 对于模式 4、5、6，像点的颜色由调色板和像素值共同决定，程序只能选择一个调色板。调色板、像素值和像点颜色的对应关系如下：

若选择 0 号调色板,则:像素值 = 0 1 2 3

像点颜色 = 背景色 绿 红 棕

若选择 1 号调色板,则:像素值 = 0 1 2 3

像点颜色 = 背景色 青 品红 白

③ 对于模式 0DH~13H,像点的颜色由像素值决定,以 16 色为例,像素值为 0~15。像素值 D7 位=0,写一个像点,D7 位=1,新的像素值和像点的原像素值异或之后再写入。

④ 设定模式 0DH~13H 之后,不能再选择调色板,否则无法显示图形。

⑤ 屏幕上任一像点的行列值,以屏幕左上角为参考点,左上角像点的行列坐标为 0 行 0 列。

1.6.2 图形方式程序设计举例

《32 位微型计算机原理与接口技术》一书的 6.14 节介绍了画水平线、画方块、画正弦曲线的设计方法,本小节再补充介绍画斜线和画圆的设计方法。

【例 1.6.1】 画斜线

以屏幕中心为起点,画一条与正向水平轴成 30°角的斜线。

[设计思路]

本例设置屏幕为模式 12H,即 640×480 图形方式。斜线长度 RR 取值为 200 个像点。设计的关键是找出斜线端点 A 的行列坐标表达式,该表达式必须包含线长 RR 这一因子。假设 OY 、 OX 为屏幕中心的行列坐标值,看图 1.3 可知:

A 点的行坐标 = $OY - (100 \times \sin 30^\circ \times RR) \div 100$

A 点的列坐标 = $OX + (100 \times \cos 30^\circ \times RR) \div 100$

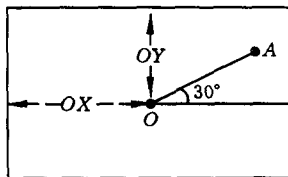


图 1.3 斜线端点的坐标图

设计一个写像点子程序,每写一个像点之后将线长 RR 减 1,再重新计算斜线端点的行列坐标,再次调用写像点子程序……,直到线长 RR 为 0 时止。

[程序清单]

```
                ;FILENAME ;EXA161.ASM
                .486
DATA            SEGMENT USE16
Y_COORD        DW      ?           ;暂存像点的行坐标
X_COORD        DW      ?           ;暂存像点的列坐标
COLOR          EQU     14          ;像素值
OY             EQU     240          ;屏幕中心行坐标
OX             EQU     320          ;屏幕中心列坐标
RR             DW      200          ;斜线长度
```