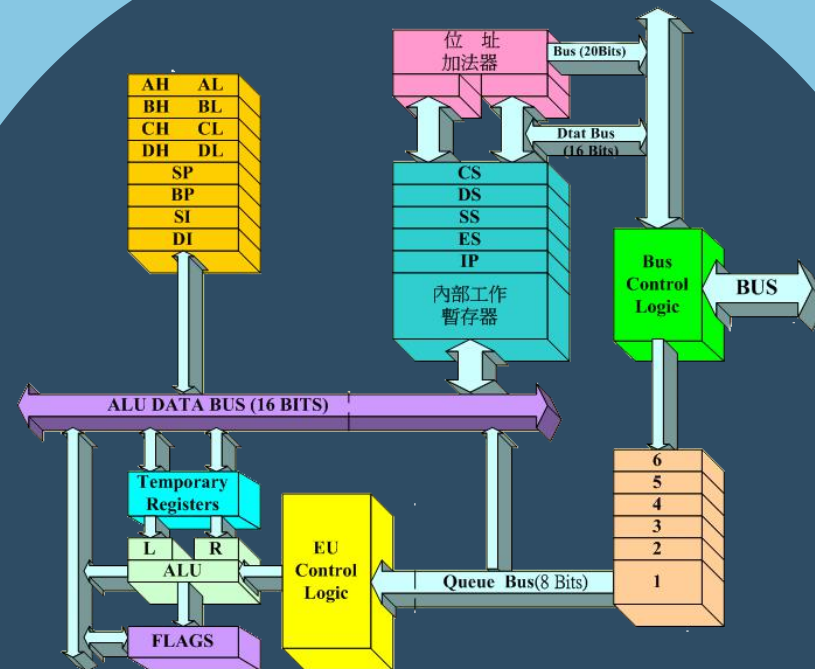


数据的直接定址表

贺利坚 主讲



汇编语言程序设计
Assembly Language

直接定址表：用查表的方法解决问题

0	0	1	0	1	0	1	1
2				B			
'2'				'B'			

问题：以十六进制的形式在屏幕中间显示给定的byte 型数据。

分析：先将一个byte的高4位和低4 位分开，显示对应的数码字符

1. 直接方案

逐个比较并显示

如果数值为 0，则显示'0';
如果数值为 1，则显示'1';
:
如果数值为 11，则显示'B';

要使用多条比较、
转移指令，程序将
长且混乱。

2. 改进方案

在数值0~15和字符'0'~'9'、'A'~'F'之间找到一种映射关系

- 数值0~9和字符 '0'~'9'之间的映射关系：数值+30h=对应字符的ASCII值：
0+30h='0'的ASCII值
1+30h='1'的ASCII值
2+30h='2'的ASCII值，
仍需要进行一些比较，
对于0~9和10~15的值，
要用不同的计算方法。
- 10~15和'A'~'F'之间的映射关系：数值+37h=对应字符的ASCII值：
0Ah+37h='A'的ASCII值
0Bh+37h='B'的ASCII值
0Ch+37h='C'的ASCII值，

3. 最简方案

建立一张表，表中依次存储字符 “0” ~ “F”，我们可以通过数值0~15直接查找到对应的字符

```
table db '0123456789ABCDEF' ;字符表
```

最简解决方案

```
assume cs:code
code segment
start: mov al,2Bh
      call showbyte
      mov ax,4c00h
      int 21h
```

;子程序
;用al传送要显示的数据

```
code ends
end start
```

```
showbyte:
    jmp short show
    table db '0123456789ABCDEF' ;字符表
show:  push bx
      push es
      push cx

      mov ah,al
      mov cl,4
      shr ah,cl ;右移4位, ah中得到高4位的值
      and al,00001111b ;al中为低4位的值

      ;用高4位的值(ah)作为相对于table的偏移,
      ;取得对应的字符并显示

      ;用低4位的值(al)作为相对于table的偏移,
      ;取得对应的字符

      pop cx
      pop es
      pop bx
      ret
```



最简方案：建立一张表，表中依次存储字符'0'~'F'，通过数值0~15直接查找到对应的字符。

```
mov bl,ah
mov bh,0
mov ah,table[bx]

mov bx,0b800h
mov es,bx
mov es:[160*12+40*2],ah
```

```
mov bl,al
mov bh,0
mov al,table[bx]

mov es:[160*12+40*2+2],al
```

直接定址表

问题求解思路

利用表，在两个数据集合之间建立一种映射关系，用查表的方法根据给出的数据得到其在另一集合中的对应数据。

优点

- 算法清晰和简洁
- 加快运算速度
- 使程序易于扩充

三角函数表

角	正弦	余弦	正切	角	正弦	余弦	正切
0°	0.0000	1.0000	0.0000	45°	0.7071	0.7071	1.0000
1°	0.0175	0.9998	0.0175	46°	0.7103	0.6947	1.0355
2°	0.0349	0.9994	0.0349	47°	0.7314	0.6820	1.0724
3°	0.0523	0.9986	0.0524	48°	0.7431	0.6694	1.1106
4°	0.0698	0.9976	0.0699	49°	0.7547	0.6551	1.1504
5°	0.0872	0.9962	0.0875	50°	0.7660	0.6428	1.1918
6°	0.1045	0.9945	0.1051	51°	0.7771	0.6293	1.2349
7°	0.1219	0.9925	0.1228	52°	0.7880	0.6157	1.2799
8°	0.1392	0.9903	0.1405	53°	0.7986	0.6018	1.3270
9°	0.1564	0.9877	0.1584	54°	0.8090	0.5878	1.3764
10°	0.1736	0.9848	0.1763	55°	0.8192	0.5736	1.4281
11°	0.1908	0.9816	0.1944	56°	0.8290	0.5592	1.4826
12°	0.2079	0.9781	0.2126	57°	0.8387	0.5446	1.5399
13°	0.2250	0.9744	0.2309	58°	0.8480	0.5299	1.6003
14°	0.2419	0.9703	0.2493	59°	0.8572	0.5150	1.6643
15°	0.2588	0.9659	0.2679	60°	0.8660	0.5000	1.7321
16°	0.2756	0.9613	0.2867	61°	0.8746	0.4848	1.8040
17°	0.2924	0.9563	0.3057	62°	0.8829	0.4695	1.8807
18°	0.3090	0.9511	0.3249	63°	0.8910	0.4540	1.9626
19°	0.3256	0.9455	0.3443	64°	0.8988	0.4384	2.0503
20°	0.3420	0.9397	0.3640	65°	0.9063	0.4226	2.1445
21°	0.3584	0.9336	0.3839	66°	0.9135	0.4067	2.2460
22°	0.3746	0.9272	0.4040	67°	0.9205	0.3907	2.3559
23°	0.3907	0.9205	0.4245	68°	0.9272	0.3746	2.4751
24°	0.4067	0.9135	0.4452	69°	0.9336	0.3584	2.6051
25°	0.4226	0.9063	0.4663	70°	0.9397	0.3420	2.7475
26°	0.4384	0.8988	0.4877	71°	0.9455	0.3256	2.9042
27°	0.4540	0.8910	0.5095	72°	0.9511	0.3090	3.0777
28°	0.4695	0.8829	0.5317	73°	0.9563	0.2924	3.2709
29°	0.4848	0.8746	0.5543	74°	0.9613	0.2756	3.4874
30°	0.5000	0.8660	0.5774	75°	0.9659	0.2588	3.7321
31°	0.5150	0.8572	0.6009	76°	0.9703	0.2419	4.0108
32°	0.5299	0.8480	0.6249	77°	0.9744	0.2250	4.3315
33°	0.5446	0.8387	0.6494	78°	0.9781	0.2079	4.7046
34°	0.5592	0.8290	0.6745	79°	0.9816	0.1908	5.1446
35°	0.5736	0.8192	0.7002	80°	0.9848	0.1736	5.6713
36°	0.5878	0.8090	0.7265	81°	0.9877	0.1564	6.3138
37°	0.6018	0.8018	0.7536	82°	0.9903	0.1392	7.1154
38°	0.6157	0.8157	0.7813	83°	0.9925	0.1219	8.1443
39°	0.6293	0.8293	0.8098	84°	0.9945	0.1045	9.5144
40°	0.6428	0.8428	0.8391	85°	0.9962	0.0872	11.430
41°	0.6561	0.8561	0.8693	86°	0.9976	0.0698	14.301
42°	0.6691	0.8691	0.9004	87°	0.9986	0.0523	19.081
43°	0.6820	0.8820	0.9325	88°	0.9994	0.0349	28.636
44°	0.6947	0.8947	0.9657	89°	0.9998	0.0175	57.290
45°	0.7071	0.7071	1.0000	90°	1.0000	0.0000	∞

应用示例：为加快运算速度而采用查表方法

问题

编写程序，计算 $\sin(x)$ ， $x \in \{0^\circ, 30^\circ, 60^\circ, 90^\circ, 120^\circ, 150^\circ, 180^\circ\}$ ，并在屏幕中间显示计算结果。

常规解法

- 利用麦克劳林公式来计算 $\sin(x)$
- 将角度 x 换为弧度： $y = x / 180 * 3.1415926$

$$\sin(x) = \sin(y) \approx \frac{1}{3!} y^3 + \frac{1}{5!} y^5$$

评价

- 计算 $\sin(x)$ 需要进行多次乘法和除法。
- 乘除是“昂贵”的操作

空间换时间方案

- 将所要计算的 $\sin(x)$ 的结果都存储到一张表中，然后用角度值来查表，找到对应的 $\sin(x)$ 的值。

具体方法

- 用 ax 向子程序传递角度
- 以角度值/30为table表中的偏移，可以找到对应的字符串的首地址。

$\sin(0)=0$
 $\sin(30)=0.5$
 $\sin(60)=0.866$
 $\sin(90)=1$
 $\sin(120)=0.866$
 $\sin(150)=0.5$
 $\sin(180)=0$

解决方案

```
push bx
push es
push si

mov bx,0b800h
mov es,bx

mov ah,0
mov bl,30
div bl
mov bl,al
mov bh,0
add bx,bx
mov bx,table[bx]

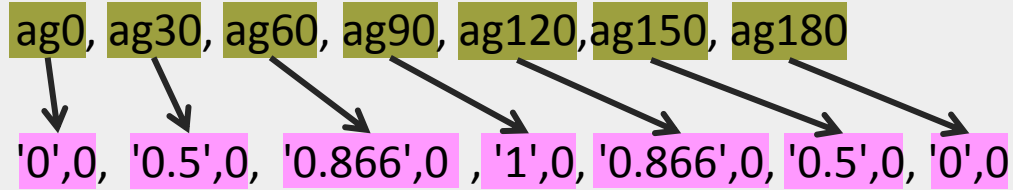
pop si
pop es
pop bx

assume cs:code
code segment
start:
    mov al,60
    call showsin
    mov ax,4c00h
    int 21h

showsin:
    jmp short show
    ;字符串偏移地址表

show:
    ;预备
    ;用角度值/30 作为相对于table的
    ;偏移量，取得对应的字符串的偏
    ;移地址，放在bx中
    ;显示sin(x)对应的字符串
    ;善后
    ret

code ends
end start
```



```
table dw ag0,ag30,ag60,ag90,ag120,ag150,ag180
ag0    db '0',0    ;sin(0)对应的字符串'0'
ag30   db '0.5',0  ;sin(30)对应的字符串'0.5'
ag60   db '0.866',0 ;sin(60)对应的字符串'0.866'
ag90   db '1',0    ;sin(90)对应的字符串'1'
ag120  db '0.866',0 ;sin(120)对应的字符串'0.866'
ag150  db '0.5',0  ;sin(150)对应的字符串'0.5'
ag180  db '0',0    ;sin(180)对应的字符串'0'
```

```
mov si,160*12+40*2
shows: mov ah,cs:[bx]
      cmp ah,0
      je showret
      mov es:[si],ah
      inc bx
      add si,2
      jmp shows

showret:
```

功能拓展：考虑程序的容错性，加上对提供的角度值是否超范围的检测，定位不到正确的字符串，将出现错误。

方法拓展：直接定址表中存储子程序的地址，从而方便地实现不同子程序的调用。