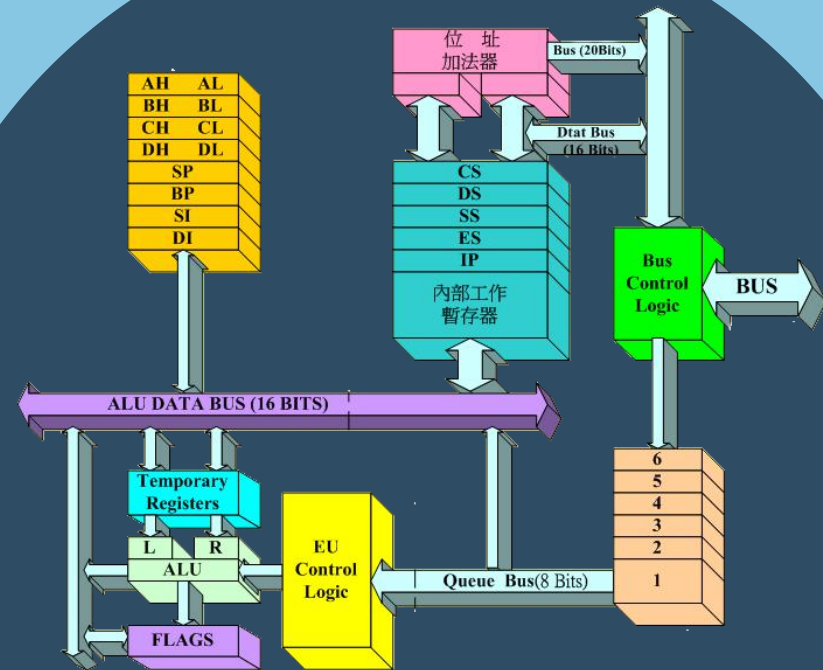


# 寻址方式的综合应用

# 贺利坚 主讲



# 汇编语言程序设计

## Assembly Language

# 应用问题

🖥️ 关于姚明2001年的一条记录：

- 📄 姓名：Yao
- 📄 生日：'19800912'
- 📄 球衣号码：15
- 📄 场均得分：32
- 📄 效力球队：SHH(上海)



🖥️ 2002年，姚明的信息有了变化：

- 1、球衣号码变换成了11号
- 2、场均得分为13
- 3、效力球队变为NBA的休斯顿火箭队（HOU）

🖥️ 任务：编程修改内存中的过时数据。

seg:60 +00	'Yao'	'Yao'
+03	'19800912'	'19800912'
+0C	15	11
+0E	32	13
+10	'SHH'	'HOU'
	2001年数据	2002年数据

# 解決方案

```
mov ax,seg
mov ds,ax
mov bx,60h
mov word ptr [bx+0ch],11
mov word ptr [bx+0eh],13
```

```
mov si,0
mov byte ptr [bx+10h+si],'H'
inc si
mov byte ptr [bx+10h+si],'O'
inc si
mov byte ptr [bx+10h+si],'U'
```

seg:60 +00	'Yao'	'Yao'
+03	'19800912'	'19800912'
+0C	15	11
+0E	32	13
+10	'SHH'	'HOU'

## 2001年数据

## 2002年数据

## 汇编指令中寻址的其他写法

 [bx+idata] [bx].idata [bx+idata+si] [bx].idata[si]

[bx+10h+si]

[**bx**].10h[**si**]

# C语言和汇编的处理方式对比

```
1  #include <stdio.h>
2  struct Player{
3      char name[3];
4      char birthday[9];
5      int num;
6      int ppg; //Points Per Game
7      char team[3];
8  };
9  struct Player yao={"Yao", "19800912", 15, 32, "SHH"};
10 int main()
11 {
12     int i;
13     yao.num = 11;
14     yao.ppg = 13;
15     i = 0;
16     yao.team[i] = 'H';
17     i++;
18     yao.team[i] = 'O';
19     i++;
20     yao.team[i] = 'U';
21     return 0;
22 }
```

yao.team[i]：yao  
是一个变量名，  
指明了**结构体变  
量的地址**；team  
是一个名称，指  
明了**数据项**team  
的地址；i用来定  
位team中的字符。

用bx定位**整个结构  
体**；用idata定位结  
构体中的某一个**数  
据项**；用si定位**数  
据项中的元素**。

seg:60 +00

+03

+0C

+0E

+10

	'Yao'
	'19800912'
	15
	32
	'SHH'

2001年数据

	'Yao'
	'19800912'
	11
	13
	'HOU'

2002年数据

```
1  mov ax,seg
2  mov ds,ax
3  mov bx,60h
4  mov word ptr [bx+0ch],11
5  mov word ptr [bx+0eh],13
6
7  mov si,0
8  mov byte ptr [bx+10h+si],'H'
9  inc si
10 mov byte ptr [bx+10h+si],'O'
11 inc si
12 mov byte ptr [bx+10h+si],'U'
```

8086CPU提供的如[bx+si+idata]的寻址方式为结构化数据的处理提供了方便，使得我们可以在编程的时候，从结构化的角度去看待所要处理的数据。