Ppt1：

通过例子，复习上一次内容，引入本次内容

例1：alice语文146分，数学122分，求总分：

用计算机处理步骤如下：

将146送入贮存单元1.

将122加贮存单元1内容，结果存入贮存单元1

现在贮存单元1中的内容为268

此为10进制规则，若使用16进制：

将92h送入贮存空间1

将7Ah加上贮存单元1中的内容，结果存入贮存单元1

但是他们的编码都是：

10010010 + 01111010 = （0001）00001100（10Ch存在溢出）

**无符号数，只要最高位存在借位或者进位情况，那么就溢出了**。

Ppt2：

例2：A欠银行-110元，B存银行122元，则银行现共有多少钱：

计算机处理如下：

将-110送入贮存单元1

将122加贮存单元1中的内容，结果存入贮存单元1

贮存单元1中的内容为0Ch

上面使用十进制计数，下面用16进制：

将-6Eh送入贮存单元1

将7Ah加贮存单元1内容，结果存入贮存单元1

结果未溢出。

（即使这个例2中的计算和例1当中的完全一致）

**有符号数，1. 同号数相加，结果与加数符号不同；2. 异号数相减，结果与被减数符号不同**。

但无论是例1还是例2，他们的表示都是：

MOV [100H] 92H

ADD [100H] 7AH

Ppt3：

文字叙述的功能对应计算机指令

Ppt4：

贮存单元对应寄存器、内存单元、或者端口

MOV AX, 146

ADD AX, 122

MOV WORD PTR [100H], 146

ADD WORD PTR [100H], 122

Ppt5：

80x86微机系统的组成：

Cpu的功能: 指令

Cpu内部的寄存器——用以临时存放只领执行所产生的中间结果存储单元

内存——程序中的指令和处理的数据主要存放在这个空间

I/O端口——计算机和外设通过这个接口交换数据

汇编语言必须理解的数据结构

内存单元的类型

80x86的栈结构

Ppt6：

80x86计算机组织结构：

1. 80x86微处理器

80x86系列始于1978年intel公司生产的8086，先后有8088，8086，80286，80386，80486

包括AMD,Cyrix系列的芯片，它们在指令和体系结构方面是完全兼容的

在80386之前，是16位的cpu，以后的32位的，属于IA-32

Ppt7：

工作模式：

32位处理器提供了三种工作模式：

实模式，相当于高速运行的8086芯片

保护模式

模拟8086模式，在保护模式下，模拟多个8086处理器工作

关于64-bit处理器：

IA-64：不兼容x86

Ppt8：

80x86计算机系统结构：

80x86 《---------》 内存储器

中央处理器 《---------》 I/O接口 《-------》 I/O设备

编程相关的是：

1. CPU寄存器及使用（在中央处理器中）
2. 内存储器及使用 （在内存储器）
3. I/O端口及访问（在I/O接口里面）

Ppt9：

控制器和运算器：

1. 运算器:
   1. 主要执行算术和逻辑运算操作，来完成对数据的加工和处理
2. 控制器
   1. 负责从内存取出指令，并对指令的分析，发出相应的控制和定时信息，控制和协调计算机的各个部件的工作，以完成指令所规定的操作
   2. 指令指针（IP）指向**下一条待执行指令**的位置

Ppt10：

80x86寄存器组：

寄存器——用以临时存放指令执行产生的中间结果的存储单元

和编程密切相关的寄存器：

——16位CPU有14个基本寄存器：AX, BX, CX, DX, SP, BP, DI, SI, IP, FLAGS, CS, DS, ES, SS

——32位CPU有16个基本寄存器：

1. 通用寄存器：

用于算术运算，逻辑运算和数据的传送

虽然是通用寄存器，但是各自有各自的用途

1. 数据寄存器：EAX,EBX,ECX,EDX

低16位的是16位寄存器，记作，AX,BX,CX,DX

低16位寄存器的高8位、低8位是2个8位寄存器。

低8位叫AL，高8位叫AH。

Ppt11：

1. 于AX,作为累加器使用，是算术计算所使用的主要寄存器。

8，16，32位累加器对应AL,AX,EAX

1. EBX，基址寄存器。16，32基址对应BX, EBX
2. ECX，计数器。8，16，32计数器对应CL,CX,ECX
3. EDX，数据寄存器。8，16，32数据寄存器分别为DL,DX,EDX

Ppt12：

1. 指针寄存器：

ESP,EBP

1. ESP：堆栈指针，专门用以访问堆栈上的数据的寄存器。

32，16用ESP, SP,内容始终指向堆栈栈顶。从这一点上来看，ESP/SP是专用的

1. EBP：基址指针。可以用来存放数据，但是更经常、更重要的用途是作为堆栈区的一个基地址，以便访问堆栈中的数据。

Ppt13：

3. 变址寄存器：

1. ESI：源变址寄存器，16位对应SI，可以存放数据，也可以作为指针，存放内存单 元地址。

2. EDI：目的变址寄存器，对应16位的是DI, 可以存放数据，也可以作为指针，存放内存单元地址。

Ppt14：

4. 控制寄存器：

1. EIP：用于指示待执行指令的位置：

32，16下对应EIP IP，运行期间，cpu自动修改ip，程序不直接修改ip，随着指令的进行，ip相应的变动。

2. EFLAGS：一条指令执行后，cpu所处的状态

16位下是flags

主要有：运算结果标志，状态控制标志，和系统状态标志等寄存器。

Ppt15：

2. 控制寄存器：

主要掌握CF, PF, AF, ZF, SF, OF

分别在0，2，4，6，7，11位

1. **无符号数溢出**标志：（无符号溢出法则）
   1. CF加运算最高位有进位或者减法有借位，那么CF为1，否则为0。
   2. 可以说是无符号数运算结果溢出的标志。
2. 奇偶校验标志：
   1. PF在运算结果中有偶数个1，为1，否则为0
3. Sd
   1. AF加法或剑法运算时，最低4位向高位有借位或者进位时，AF为1，否则为0
4. 补码正负标志：
   1. 运算结果最高为为1，SF为1，否则为零，反应补码数正负性
5. 零非零标志：
   1. 运算结果为0，为1，否则为0
6. **有符号数溢出**标志：（有符号溢出法则）
   1. 结果超过补码表示范围，那么溢出，OF为1，否则为0.

Ppt16：

ZF, OF, CF, SF经常用，但是PF, AF比较少用。

CPU每执行一条算术指令，或者逻辑运算的指令，依据运算结果状态来设置各个标志位。

然后写了一波程序：

-A

13E5:0100 MOV [1000], BYTE PTR 92

13E5:0105 ADD [1000], BYTE PTR 7A

13E5:010A

-R

-T

前为1, 后为0

(CF:CY/NC;

PF:PO/PE

AF:AC/NA

ZF:ZR/NZ

SF:PL/NG

OF:OV/NV)

Ppt17：

状态控制标志:

控制cpu的操作，可通过专门的指令设置或清除

1. TF用于调试程序，若为1，那么cpu每执行完一条指令，便产生一个单步中断
2. IF允许/禁止中断响应。若IF为1，那么允许cpu相应，若IF为0，那么禁止响应。
3. DF方向控制，若DF清零，串指令按加方式处理，若DF置1，串指令减法执行。

Ppt18：

段寄存器

一个连续单元的内存区域为段，段寄存器是用于存放指示该段首地址的相关内容

16位cpu有4个段寄存器，cs，ds，ss，es，32位cpu又增加了fs和gs

Cs（code segment）指示代码段

Ds（data segment）指示数据段

Ss（stack segment）指示堆栈段

Es（extra segment）fs，gs无意义，是排下来的

Ppt19：

内存储器

用途：用于存放执行的指令，所处理的数据

基本单位：byte

地址:每个字节单元的唯一编号

寻址空间受限于地址总线宽度

20bit地址总线，可访问00000~fffff

未完待续。。。

Ppt20：

内存单元和数据存放格式：

存储单元：连续若干个字节存储空间

类型：

字节单元：

2：word

4：dword

8：qword

标识：以内存块的起始地址标识

存放格式：低位存入低单元，高位存入高单元（小端格式，intel格式）

Ppt22：

78h 100H

56h 101H

将5678h存放于100号2字节单元（word）

78h 200h

56h 201h

34h 202h

12h 203h

A2h 204h

A3h 205h

将12345678h存放于200号4字节单元（dword）

从201号字单元取，结果为3456h

从201号双字单元取，结果为a2123456h

Ppt23：

内存的分段使用:

内存段：连续的若干字节的内存区域

基地址由段寄存器和偏移来决定（没抄完）

Ppt24：

应用程序将内存分为三种区域

代码段

堆栈段

数据段

Ppt26：

20bit总线，寻址空间为00000~FFFFFh

16bit为0000H~FFFFH

解决办法就是16位短地址和16位偏移地址，合成20位物理地址

规则：20位地址 = 段地址 \* 16 + 偏移地址

如：段地址1000h 偏移地址位1234h

则 1000h \* 16d + 1234h = 11234h， 11234h为20位地址

Ppt27：

I/O地址空间：

80x86处理器通过I/O接口与外部的I/O设备交互

I/O接口在其中的接口寄存器和转换器之间编码解码

Io接口寄存器称为io端口

端口类型：8，16，32

用输入输出指令直接控制外部io设备

系统硬件制造商将常用的功能模块，固化在rom中，形成bios，basic input/output system

Ppt28：

Ss寄存器专用于确定堆栈区的基地址

Sp专门用于指示栈顶位置，即总是指向坐进进栈的数据位置

栈底固定为栈区的最高地址单元。执行push操作后，sp向低地址移动，执行pop操作之后，sp向高地址方向移动

8位的话是2个2个执行

同理可得16位

Masm T.ASM

LINK T.OBJ

DEBUG T.EXE