

期末考试题型：

一、选择题（10 小题，共 20 分）

二、判断题（10 小题，共 10 分）

三、计算题（3 小题，共 18 分：IEE754、寻址方式、汇编程序分析）

四、指令流程和微指令（16 分：传送指令、双操作数指令）

五、主存设计（14 分，PPT 例题、作业题）

六、汇编程序设计题（2 小题，22 分，完整程序设计【数据段、代码段、堆栈段、附加段】、部分程序设计）

复习范围：计算机组成原理第 1-5 章，微型计算机接口技术第 2, 3, 4, 6 章。

复习资料：教材、作业、期中试卷、PPT、以下资料（计算机组成原理复习.PPT+指令和伪指令复习(12.7).pdf）

《计算机组成原理》

第 1 章

- 1、冯诺依曼思想包含三个方面
 - 2、编译方式和解释方式对源程序处理有什么区别
 - 3、汇编程序、汇编语言（汇编源程序）
 - 4、控制流驱动和数据流驱动
 - 5、计算机性能评价指标
- 题 14，题 15
- 6、名词

机器字长：CPU 一次能处理数据的位数。通常与 CPU 的寄存器的位数有关，单位 b (bit)

存储字长（按字节/字/双字节编址）：存储单元的位数。MDR 的位数反映了存储字长。

存储单元的个数：存储器中存储单元的个数。MAR 的位数反映了存储单元的个数。如果 MAR 为 N 位，则存储单元的个数为 2 的 N 次幂。

地址线条数：决定了存储单元的最大寻址范围，决定了 MAR 的位数和存储单元个数。

存储容量：存储容量=存储单元的个数*存储字长。单位 B (byte)

指令字长：一条指令中包含的二进制代码的位数。指令字长一般是存储字长的整数倍，如果指令字长等于存储字长的 2 倍，需要 2 次访存来取出指令，因此，取指周期是机器周期的 2 倍

【例 1】某主存总容量为 64MB，按字编址，需要多少条地址线，MAR 位数是多少？

【例 2】若数据传输指令的元 and 目的操作数都采用变址寻址，指令字长为 48bit（包含形式地址），数据总线和地址总线为 16bit，则该指令的执行需要从存储器取指令、取源操作数，并送到目的存储单元，需要多少次访存？

第 2 章

- 1、二进制、十进制、十六进制转换；原码、补码、反码、移码，注意原码和反码的“0”的表示法。表示范围和精度
- 1) 若寄存器的内容为 111111，若它等于-1，则为 (B)
A、原码 B、补码 C、反码 D、移码
- 2) 若寄存器的内容为 11111111，若它的真值为+127，则机器数为 (D)
A. 反码 B. 补码 C. 原码 D. 移码

2、IEEE754 表示法，掌握十进制和 IEEE754（32 位）之间的相互转换。

浮点数的表示方法（数符、尾数、阶码、阶符、尾数规格化）

3、算术移位、逻辑移位规则、溢出判断规则

第 3 章 CPU 子系统

3.1 概述

1、运算部件（多路选择器、锁存器）

2、数据缓存部件（Cache L1、L2、L3）

3、寄存器

各种寄存器的作用（R0~R3、PSW、PC、SP、IR、MAR、MDR、C、D）：哪些是可编程寄存器，哪些是不能编程访问的。

4、控制部件

（1）控制部件的输入信号包括 IR、PSW、PC、时序系统、I/O 请求、复位信号

（2）按照微命令产生方式，控制器分为组合逻辑控制器和微程序控制器。

（3）一般指令的读取和执行分为取指令、取源操作数、取目的操作数、执行阶段

（4）微命令：在计算机中用来控制微操作（如逻辑门的开或关、寄存器的打入或清除等操作）的控制命令称为微命令，也称为微操作控制信号。

（5）组合逻辑控制：简单地讲，由硬连逻辑电路产生微命令的方式称为组合逻辑控制方式。它的基本思想如下：

综合、化简产生微命令的条件，形成相应逻辑式，并用组合逻辑电路实现；执行指令时，由组合逻辑电路（微命令发生器）在相应时间发出所需微命令，控制有关操作。

（6）微程序控制：简单地讲，由微指令译码产生微命令的方式称为微程序控制方式。它的基本思想如下：

将若干微命令编制成一条微指令，控制实现一步操作；将若干微指令组成一段微程序，解释执行一条机器指令；将微程序事先存放在控制存储器中，执行机器指令时再取出。

5、时序部件

（1）时序信号：周期、节拍、脉冲

（2）时序系统：产生时序信号的部件，由 1 个低频振荡器和倍频逻辑组成

（3）指令周期、工作周期、总线周期、时钟周期、时钟信号

（4）同步控制、异步控制、扩展同步控制

1、PSW 寄存器的特征位是程序员设置的（ ）

x

2、MAR 寄存器是采用双向三态门或者 OC 组件构成。（ ）

x

3、暂存器 C 用于存放源操作数或者源操作数地址，暂存器 D 存放目的操作数或者目的操作数地址（ ）

v

4、IR 寄存器适用于存放当前指令地址（ ）

x

5、现代微处理器缓存采用三级缓存 L1、L2 和 L3，三级缓存都部署与 CPU 内部。（ ）

v

6、CM 属于 CPU 的一部分（ ）

v

7、时序系统里面有一个启停控制逻辑，它的作用是

保证可靠的送出完整脉冲，并且产生一个总清信号/复位信号，对计算机相关部件进行初始化。

8、晶体振荡器产生的时钟频率低于 CPU 的主频（ ）

V

9、同步控制的同一时序信号都由 CPU 产生（ ）

X

10、同步控制的时钟周期长度固定不变（ ）

V

11、异步控制方式是指（ ）

各项操作不受统一时序信号【如时钟周期】的约束，各部件之间数据传输采用应答方式。

12、扩展同步控制根据实际时间分配时钟周期数，时钟周期长度不变（ ）

V

13、计算机工作的最小时间周期是（ ）

A、时钟周期 B、指令周期 C、总线周期 D、工作脉冲

D

14、时序控制方式中，有一种方式，是总线周期中允许插入延长周期，其含义是（ ）

A、将一个总线周期延长为多个总线周期

B、在一个时钟周期内插入新的总线周期

C、在总线周期内增加新的总线周期

D、在一个总线周期内，延长时钟周期长度

B

3.2 指令系统

1、指令格式

简化地址结构、操作码（固定操作码、扩展操作码、复合型操作码）

（1）简化地址结构的基本途径

在指令中减少显地址的数量，即使用隐地址方式给出地址，指令中的地址（段）个数就可减少。

（2）减少指令中地址信息的位数的方法

采用寄存器寻址、寄存器间址等以寄存器为基础的寻址方式可以大大减少指令中一个地址的信息位数。

（3）显地址

在指令代码中明显给出的地址，如在指令中写明操作数的主存单元地址或寄存器号，则这种地址称为显地址。

（4）隐地址

在指令中不明显给出地址码，地址以隐含方式约定，这种隐含约定的地址称为隐地址。

1、指令长度和机器字长有什么关系？

解：指令长度与机器字长没有固定的关系，指令长度可以等于机器字长，也可以大于或小于机器字长。

2、假设指令系统的指令字长 16 位，地址位 4 位，若两地址指令需要 12 条，单地址指令需要 12 条，那么零地址指令最多有（ ）条。

$((2^8-12) \cdot 2^4-12) \cdot 2^4$

2、寻址方式

指令中以什么方式提供操作数或操作数地址，称为寻址方式。

(1) 立即寻址：由指令直接给出操作数，在取出指令的同时也就取出了可以立即使用的操作数，这种寻址方式称为立即寻址。

(2) 直接寻址：由指令直接给出操作数地址，根据该地址可以从主存（或寄存器）中取出操作数，或向主存（或寄存器）写入数据，这种寻址方式称为直接寻址。

(3) 寄存器寻址：在指令中给出寄存器号，从该寄存器号所指的寄存器中取出操作数或将数据传送到该寄存器号所指的寄存器中。这种寻址方式实为寄存器直接寻址。

(4) 间接寻址：在指令中给出间址单元地址码（即操作数地址的存放单元地址），按照该地址访问主存中该间址单元，从中读取操作数地址，接着按操作数地址再次访问主存，从该单元中读取或向该单元写入操作数。

a) 寄存器间址：由指令给出寄存器号，在该寄存器号所指定的寄存器中存放着操作数地址，按此地址访问主存，读取或写入操作数。

b) 间址单元：在间接寻址方式中，存放操作数地址的主存编址单元称为间址单元。

(5) 变址寻址：在指令中的地址部分给出一个形式地址，并且指定一个寄存器作为变址寄存器，将变址寄存器的内容（称为变址量）与形式地址相加，得到操作数地址（称为有效地址）；按有效地址访问主存，从相应的主存单元中读得操作数或向该单元写入数据。

(6) 基址寻址：在指令中给出一个形式地址（作为位移量），并且指定一个寄存器作为基址寄存器（该基址寄存器内容作为基准地址）；将基址寄存器内容和形式地址相加，其和作为操作数有效地址；按有效地址访问主存，从该单元读取操作数或向该单元写入数据。

(7) 相对寻址：指用程序计数器 PC 的内容作为基准地址，指令中给出的形式地址作为位移量的基址寻址方式。

(8) 页面寻址：将程序计数器 PC 的高位段作为操作数有效地址的高位段，指令中给出的形式地址作为操作数有效地址的低位段，将这两部分拼接构成操作数有效地址，这种寻址方式称为页面寻址方式。

(9) 堆栈

一种按“后进先出”（或称“先进后出”）存取顺序进行存取的存储结构。

a) 栈顶：堆栈是一个连续的存储区，其一端固定称为栈底，存放最先压入的数；堆栈的另一端是浮动的，称为栈顶，对堆栈的读写都是对栈顶单元进行的；对堆栈的寻址也就是对栈顶单元的寻址，随着堆栈操作的进行，栈顶位置也发生变化。

b) 堆栈指针：指用于指向栈顶位置的寄存器 SP，堆栈指针 SP 的内容是栈顶单元地址。

模型机：双重间址@ (R) +、变址 X (R)

1、若某个主存储器部分的地址单元与主存内容对应关系如下：

地址码	存储内容
1000H	A307H
1001H	0B3FH
1002H	1200H
1003H	F03CH
1004H	D024H

(1) 若采用寄存器间接寻址方式读取操作数，指定寄存器 R0 的内容是 1002H，则操作数是多少？

操作数是 1200H

(2) 若采用自增型寄存器间址方式 $(R1)+$ 读取操作数，R1 内容是 1000H，则操作数是多

少? 指令执行完后 R1 内容是多少?

操作数是 A307H, 指令执行后 R0 的内容变为 1001H

(3) 若采用自减型寄存器间址-(R2)读取操作数, R2 内容为 1003H, 则操作数是多少?
指令执行完后 R2 内容是多少?

操作数是 1200H, 指令执行后 R1 的内容为 1002H

(4) 若采用变址寻址方式 X(R3)读取操作数, 指令中给出的形式地址 d=3H, 变址寄存器 R3 的内容是 1000H, 则操作数是多少?

操作数为 F03CH。

2、假设 R0 的内容为 2000H, R1 的内容为 2500H, (2000H) = 3000H,
(3000H) = 4000H, 则指令 “MOV R1, @(R0)+” 执行后, R0 的内容为 (), R1 的内容为
(), (2000H) 的内容为 (), (3000H) 的内容为 ()。

2001H, 4000H, 3000H, 4000H

3、若数据传输指令的源和目的操作数都采用变址寻址, 指令字长为 48bit (包含形式地址), 数据总线和地址总线为 16bit, 则该指令的执行需要从存储器取指令、取源操作数, 并送到目的存储单元, 需要多少次访存? ()

5

4、隐地址是指 () 的地址。

A. 用寄存器号表示 B. 存放在主存单元中
C. 事先约定, 指令中不必给出 D. 存放在寄存器中

C

5、指令 ADD (R0), @(R1)+ 的指令长度是 3 字节 (存储器按字节编址), 则指令从取指开始执行, 共需要 () 次访问存储器。

7

3、运算部件

运算器的核心部件是 (ALU), 一般应具有 (算术运算) 与 (逻辑运算) 两大类运算能力。

1、并行加法器的运算速度取决于全加器单元的速度。(x)

2、串行进位链是串行加法器中的进位链。(v)

3、用 4 片 74181 和 1 片 74182 可组成 ()

A、组内并行进位, 组间串行进位的 16 位 ALU
B、组内并行进位, 组间并行进位的 16 位 ALU
C、组内串行进位, 组间串行进位的 16 位 ALU
D、组内串行进位, 组间并行进位的 16 位 ALU

B

4、模型机 CPU 设计方法

(1) 指令系统、

(2) 寻址方式:

寄存器寻址、寄存器间址、自减型寄存器间址、自增型寄存器间址、自增型双重间址、相对寻址、跳步

(3) 指令类型

双地址指令、单地址指令、转移指令

(4) 模型机的数据通路结构 P137 图 3-43

(5) 指令流程和微命令 P140-P146

传送指令、双操作数指令

中断周期 P149

(6) 组合逻辑控制器

时序系统（工作周期【6个触发器】、时钟周期【计数器】、工作脉冲）

机器字长：CPU 一次能处理数据的位数。通常与 CPU 的寄存器的位数有关，单位 b (bit)

存储字长（按字节/字/双字节编址）：存储单元的位数。MDR 的位数反映了存储字长。

存储单元的个数：存储器中存储单元的个数。MAR 的位数反映了存储单元的个数。如果 MAR 为 N 位，则存储单元的个数为 2 的 N 次幂。

地址线条数：决定了存储单元的最大寻址范围，决定了 MAR 的位数和存储单元个数。

存储容量：存储容量=存储单元的个数*存储字长。单位 B (byte)。

【例 1】某主存总容量为 64MB，按字编址，需要多少条地址线，MAR 位数是多少？(25, 25)

【例 2】若数据传输指令的源和目的操作数都采用变址寻址，指令字长为 48bit（包含形式地址），数据总线和地址总线为 16bit，则该指令的执行需要从存储器取指令、取源操作数，并送到目的存储单元，需要多少次访存？(5)

【例 2】模型机中，MOV X(R0)， X(R1)，则该指令的执行需要从存储器取指令、取源操作数，并送到目的存储单元，需要多少次访存？(5)



【例 3】模型机中，MOV (R0)， @(R1)+，则该指令的执行需要从存储器取指令、取源操作数，并送到目的存储单元，需要多少次访存？(4)

【例 4】模型机中，执行指令 JSR @(R0)+后，PC 和 SP 的值如何改变？（SP-1→SP ,PC 值压栈后，PC 值变为子程序入口地址）

执行 RST (SP)+指令后，PC 和 SP 的值如何改变？（PC 出栈，SP+1→SP）

(7) 微程序控制器

一条机器指令的执行分为若干步，将每一步操作所需的微命令按照固定格式进行编码，并存储成一条微指令，多条微指令构成一段微程序，这段微程序对应一条机器指令。

机器在执行指令过程中，每一步（时钟周期）取出一条微指令，经过译码后，产生一组控制信号（微命令），控制各个部件的操作。

控制存储器 (Control memory, CM, 控存)：只读存储器，存储可以经过译码产生控制信号的微程序（经过按照一定规则进行编制）。

1、控制存储器 CM 是 () 的一种存储器

A、用于存储汇编指令 B、用于存储显示字符

C、位于 CPU 内部 D、用于存储指令流程

C

2、后续微地址的形成有（增量和断定）两种，其中按照目的操作数寻址方式实现分支属于（断定）方式

3、对微指令编码，采用分段编译法，下面说法正确的是 ()

A、如果 K 个字段，则可提供 K 个微命令

B、微命令的数量与段的数量和各段长度都有关系

C、微命令的数量和段的数量没有关系，只与微指令的总长度有关

D、以上三种都不正确。

第4章 存储子系统

1、存储器的层次结构

主存：CPU可直接访问，存放CPU当前使用的指令和数据。其特点是：能随机访问、工作速度快、容量有限。

辅存（外存）：存放大量的后备程序和数据。特点：速度较慢、容量大。

高速缓冲存储器(Cache)：存放CPU在当前一小段时间内多次使用的程序和数据，以缓解CPU和主存的速度差异。特点：速度很快、容量小。

2、存储器的分类

按存储介质（物理存储机制）分类：

（1）半导体存储器

静态存储器：利用双稳态触发器存储信息，功耗较大，速度快，做Cache。

动态存储器：依靠电容上的电荷暂存信息，功耗较小，容量较大，速度较快适合做主存。

（2）磁表面存储器：磁带、存盘、硬盘

利用磁层上不同方向的磁化区域表示信息。特点：存储容量大；非破坏性读出、永久保存；读写速度比半导体存储器慢；适合用于外存。

（3）光盘存储器

利用光斑的有无存储信息，特点：容量大；非破坏性读出、永久保存；速度慢；适合用于外存。

按存取方式分类

（1）随机存取存储器(RAM, ROM, Flash memory)

随机存取：可按地址访问存储器中的任一单元，与访问时间和单元地址无关。

技术指标：存取周期、存取时间

（2）顺序存取存储器(SAM)

访问时读/写部件按顺序查找目标地址，访问时间与数据位置有关。

例如，磁带机 — 与磁带录音机工作原理类似，但存储信息为数字信息，而非模拟信息。

技术指标：平均存取时间、数据传输率

（3）直接存取存储器(DAM)

访问时读/写部件先直接指向一个小区域，再在该区域内顺序查找。访问时间与数据位置有关，如磁盘、硬盘、光盘

技术指标：平均定位/寻道时间、平均等待/旋转时间、数据传输率

3、半导体存储原理

静态存储器SRAM(双极型、静态MOS型)：依靠双稳态电路内部交叉反馈机制，功耗较大，速度快，作Cache、(六管单元构成1位，有VCC供电)非破坏性读出，读出后不需重写。

动态存储器DRAM：依靠电容存储电荷的原理存储信息，功耗较小，容量大，速度较快，作主存。(4管单元、无VCC供电)四管单元是非破坏性读出，读出过程即实现刷新；需定期向电容补充电荷(动态刷新)。

(单管单元，无VCC)：单管单元是破坏性读出，读出后需重写。

4、存储器的逻辑设计

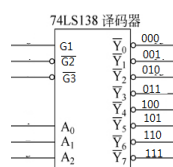
计算芯片数量(位扩展、单元数扩展)、地址分配和片选逻辑、电路连接(数据线、地址

线、片选线、读/写线)、3:8 译码器或 2:4 译码器的连线、与门、与非门、或门、或非门的画法。

参见 PPT 例题、书上作业题

步骤:

- (1) 确定芯片数量 (先位扩展, 然后扩展单元数)
- (2) 地址分配, 确定片选逻辑
- (3) 画连线图



5、动态刷新

输入							输出							
G1	G2	G3	A2	A1	A0		Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
1	0	0	0	0	0		1	1	1	1	1	1	1	0
1	0	0	0	0	1		1	1	1	1	1	1	0	1
1	0	0	0	1	0		1	1	1	1	1	0	1	1
1	0	0	0	1	1		1	1	1	1	0	1	1	1
1	0	0	1	0	0		1	1	1	0	1	1	1	1
1	0	0	1	0	1		1	1	0	1	1	1	1	1
1	0	0	1	1	0		1	1	0	1	1	1	1	1
1	0	0	1	1	1		1	0	1	1	1	1	1	1
1	0	0	1	1	1		0	1	1	1	1	1	1	1

(1) 刷新和重写的区别: 刷新是动态存储芯片, 需补充电荷以保持原信息; 重写是破坏性读出后重写, 以恢复原来的信息。

(2) 刷新原因: 动态存储器依靠电容电荷存储信息。电容电荷随时间推移会缓慢泄放, 需定期向电容补充电荷, 以保持信息不变。

(3) 最大刷新闻隔: 2ms 内, 必须对所有动态单元刷新一遍。

(4) 刷新方法: 各 DRAM 芯片同时刷新, 片内按行刷新(按行读)

刷新周期(存取周期): 刷新一行所用的时间。刷新周期数: 刷新一块芯片所需的周期数, 由芯片矩阵的行数决定。

刷新周期的安排方式:

(a) 集中刷新

(b) 分散刷新

(c) 异步刷新: 每隔一定时间 (2ms/芯片的行数) 提出一次刷新请求 (DMA 请求), 刷新一行。

6、主存储器校验

奇偶校验编码规则: 编码中 “1” 的个数为奇数或者偶数

1、某计算机主存空间是 4GB, 字长为 32 位, 按字节编制, 采用 32 位字长的指令格式, 则程序计数器 PC 和指令寄存器 IR 的位数分别是 ()

A、30, 30 B、30, 32 C、32, 30 D、32, 32

D

2、半导体 SRAM 的存储信息原理是 ()

A、依靠双稳态电路

B、依靠定时刷新

C、依靠读后再生

D、信息不再变化

A

3、下列选项不属于动态刷新的方式为 ()

A、集中刷新

B、分散刷新

C、同步刷新

D、异步刷新

C

4、主存按字节编址, 地址从 A400H 到 DFFFH, 共有容量 15KB ; 若用存储容量为 3K*4 的存储芯片, 至少需要 10 片

10

5、在计算机系统中存储器按介质分类, 它们可以分为 ()。

A、半导体存储器, 磁表面存储器, 光盘存储器; B、硬盘, 软盘, 光盘;

C、U 盘, 磁带, RAM; D、RAM, ROM, EPROM。

A

6、存储器按存取方式分为：

随机存取存储器，顺序存取存储器（SAM），直接存取存储器（DAM）。

7、地址总线 A0~A15，用 4K*4 的存储芯片组成 16KB 的存储器，则用于存储芯片内部寻址的地址线是（ ）

- A. A16~A15 B. A0~A9 C. A0~A11 D. A4~A15

C

8、某主存容量为 32KB，由 16 片 16K*1 位（内部采用 128*128 存储阵列）的 DRAM 芯片组成，若采用集中式刷新方式，且刷新周期为 2ms，那么所有存储单元刷新一遍需要至少需要安排_____个刷新周期。

128

9、16 片 2K*4 位的存储器可以设计存储容量为_____的 16 位存储器

8K

10、磁盘属于（ ）类型的存储器

- A. 随机存取存储器（RAM） B. 只读存储器（ROM） C. 顺序存取存储器（SAM） D. 直接存取存储器（DAM）

C

11、下列几种存储器中，CPU 不能直接访问的是（ ）

- A、硬盘 B、内存 C、Cache D、寄存器

A

12、DRAM 的刷新是以（ ）为单位进行的。

- A、存储单元 B、行 C、列 D、存储字

B

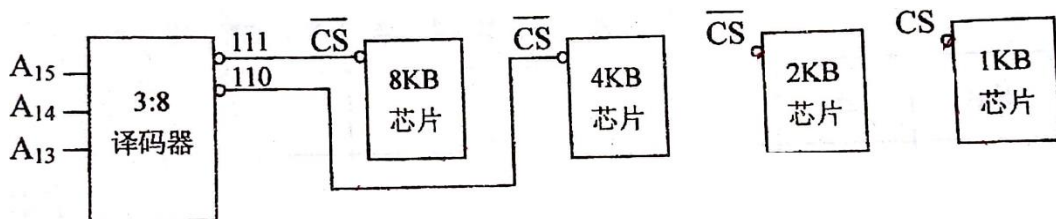
13、某存储器容量为 32K×16 位，则地址线有_____条。

15

14、某 DRAM 芯片，采用地址复用技术，其容量为 1024×8 位，除电源和接地外，该芯片的引脚最少是_____条（读写控制线为 2 条）

17 数据8+地址5+读写2+片选1+复用1（0表示行，1表示列）

16、假设地址总线 16 条，数据总线 8 条，存储器按字节编址，有以下存储器连接电路



(1) 写出 8KB 和 4KB 芯片的地址范围

(2) 分别为图中的 2KB 和 1KB 芯片分配地址范围（要求：1KB 芯片从 0000H 开始分配，2KB 芯片从 1000H 开始分配），写出其片选逻辑表达式，画出译码器电路，并与 2KB 和 1KB 芯片的片选信号相连接（要求采用全译码方式）

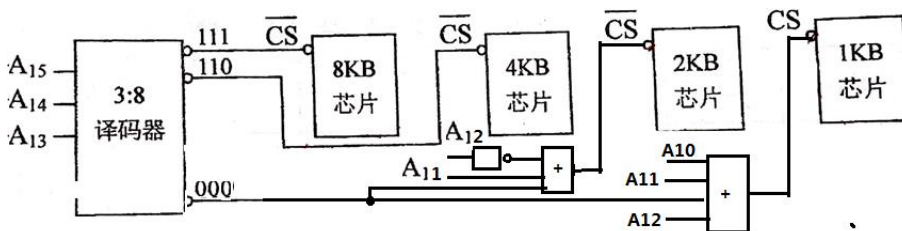
(3) 如果直接用一片 8KB 芯片替换图中的 4KB 芯片，且不修改其片选逻辑，是否可以，为什么？

(1) 8KB 芯片的地址范围：E000H~FFFFH，4KB 芯片的范围：C000H~DFFFH

	A ₁₅	A ₁₄	A ₁₃	A ₁₂	A ₁₁	A ₁₀	A ₉	A ₈	A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀
1KB	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
2KB	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1

(2) 1KB 芯片的地址范围：0000H~03FFH，2KB 芯片的范围：1000H~17FFH

低位地址分配给芯片，高位地址形成片选逻辑。			
芯片	芯片地址	片选信号	片选逻辑
1K	A ₉ ~A ₀	CS ₀	$\overline{A_{15}}\overline{A_{14}}\overline{A_{13}}\overline{A_{12}}\overline{A_{11}}\overline{A_{10}}$
2K	A ₁₀ ~A ₀	CS ₁	$\overline{A_{15}}\overline{A_{14}}\overline{A_{13}}\overline{A_{12}}\overline{A_{11}}$



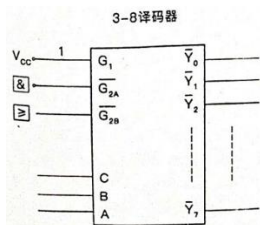
(3) 可以，4KB 的对应的地址范围：C000H~DFFFH，有 4KB 的地址重叠。

17、用 2Kx4b 的若干芯片构成一个 12KB 的存储器，其地址范围在 C000H~C2FFFH 和 C4000H~C6FFFH 之间，数据总线 D₀~D₇，地址总线是 A₀~A₁₉，芯片读写控制信号 R/W，且片选信号为 3-8 译码器输出。

(1) 需要 2Kx4b 的芯片多少片？每组芯片地址线如何分配？

(2) 哪些地址线作 3-8 译码器的使能端，哪些做 3-8 译码器输入端？

(3) 画出存错逻辑电路图（3-8 译码器使能端、输入端、输出连线，以及组成 12KB 存储芯片电路图）



(1) 需要 2Kx4b 的芯片多少片？每组芯片地址线如何分配？

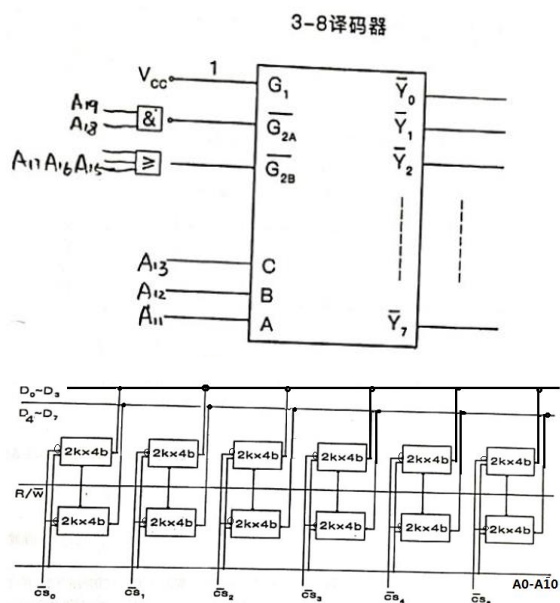
12 片，A₀~A₁₀

(2) 哪些地址线作 3-8 译码器的使能端，哪些做 3-8 译码器输入端？

使能端：A₁₉~A₁₅

片选：A₁₁~A₁₃

(3) 画出存错逻辑电路图（3-8 译码器使能端、输入端、输出连线，以及组成 12KB 存储芯片电路图）



第5章 总线与输入/输出子系统

1、I/O 系统包括：

I/O 设备：实现与主机的信息交换和人机交互

I/O 接口：主机与外设之间的连接逻辑，控制外设的 I/O 操作

系统总线：连接系统各大部件的公共信息通道

2、为什么在外设与 CPU 之间需要接口？

(1) 时间同步：不同速度外设如何与 CPU 之间进行时间同步

(2) 数据格式的转换(如果需要)：外设数据格式可能与 CPU 数据格式不同

(3) CPU 与外设之间一次数据传送量的控制

(4) 其它因素(如电平转换)

3、接口分类

(1) 按数据传送格式划分

并行接口：接口与系统总线、接口与外设均按并行方式传送数据；数据各位同时传送；适合设备本身并行工作，距主机较近的场所。

串行接口：接口与系统总线并行传送，接口与外设串行传送；数据逐位分时传送。

(2) 按时序控制方式划分

同步接口：连接同步总线，接口与系统总线的信息传送由统一时序信号控制。

异步接口：连接异步总线，接口与系统总线的信息传送采用异步应答方式。

(3) 按 I/O 传送控制方式划分

程序控制接口：通过硬件或软件方式按指定优先级查询各设备是否要进行输入/输出

中断接口：设备提出中断请求，主机响应后与设备交换信息，接口中包含中断控制逻辑

DMA 接口：支持高速外设与主机之间进行 DMA 方式交换数据

4、I/O 接口主要功能

(1) 寻址：接收 CPU 送来的地址码，选择接口中的寄存器供 CPU 访问

(2)数据缓冲：实现主机与外设的速度匹配；缓冲深度与传送的数据量有关。

(3)预处理：串-并格式转换（串口）；数据通路宽度转换（并口）；电平转换

(4)控制功能：传送控制命令与状态信息，实现 I/O 传送控制方式

5、接口编址

统一编址：将一部分总线地址（高区）分配给设备接口中的寄存器（占用了主存地址）。通用的访存指令

单独编址：为接口中的每个寄存器分配独立的端口号（与主存地址无关），有专门的 I/O 指令

6、接口传送信息包括控制命令、数据、状态

7、总线

定义：一组能为多个部件分时共享的信息传送线路。通常作法：发送部件通过 OC 组件或三态门分时发送信息，由打入脉冲将信息送入指定接收部件。

实体：一组传送线与相应控制逻辑

8、总线分类

（1）按时序控制方式划分

同步总线：由统一时序信号控制总线传送操作；在固定时钟周期内完成数据传送，由同步脉冲定时打入。

异步总线：无固定时钟周期划分，以异步应答方式控制传送

扩展同步总线：以时钟周期为时序基础，允许总线周期中的时钟数可变。

时钟周期：CPU 执行一步操作的时间

总线周期：经过总线的一次数据传送（访存）时间，通常包含若干时钟周期

工作周期：指令周期中的一个操作阶段；可包含多个总线周期

（2）计算机设计中，不同类型的总线可能采用的时序控制方式和数据传送格式：

CPU 内总线：同步、并行

局部总线：同步、异步、扩展同步、并行

系统总线：同步、异步、扩展同步、并行

外总线：异步、并行、串行

9、总线的技术指标

总线宽度、总线频率、总线数据传输率

10、总线标准

内总线(典型)：ISA、EISA、VESA、PCI、PCI-e

外总线(典型)：RS232、RS485、USB、IEEE1394

11、中断

定义：CPU 暂时中止现行程序的执行，转去执行为某个随机事态服务的中断处理程序。处理完毕后自动恢复原程序的执行。

实质：程序切换。在一条指令结束时切换，保证程序的完整性；切换前需保存断点、保护现场，中断服务程序执行结束后，需要恢复现场，返回断点。

特点：随机性

中断与转子程序：

- (1) 子程序的执行由程序员事先安排, 而中断服务程序的执行则是由随机中断事件触发。
- (2) 子程序的执行受主程序或上层程序控制, 而中断服务程序一般与被中断的现行程序无关。
- (3) 一般不存在同时调用多个子程序的情况, 但可能发生多个外设同时向 CPU 发出中断服务请求的情况。

12、中断分类

硬中断好软中断、内中断和外中断、可屏蔽中断和非屏蔽中断、向量中断和非向量中断

13、中断组成

- (1) 软件: 中断服务程序和中断向量表
- (2) 硬件: 接口 (中断请求、传递、判优逻辑)、CPU (中断响应逻辑)

14、中断优先级判断

- (1) 优先顺序: 故障、内中断、DMA、外中断; 高速操作优于低速操作, 输入优于输出。
- (2) CPU 现行程序与外设请求的判优
 - (a) CPU 设置允许中断标志
 - (b) CPU 设置程序状态字的优先级字段, 为现行程序赋予优先级

中断接口判优: 中断请求寄存器、中断屏蔽寄存器、中断服务寄存器, 判断是否发出 INT 请求。

15、中断响应

- (1) 响应条件: 外设有请求, 且未被屏蔽; CPU 开中断; 中断源优先级高于当前程序的优先级; 一条指令(非停机)结束, 即 ET 之后。
- (2) 如何获取中断程序的入口地址
 - (a) 非向量中断: 将服务程序入口组织在查询程序中, CPU 响应时, 执行查询程序, 查询中断源, 转入相应的服务程序。
 - (b) 向量中断: 将中断源的服务程序入口 (中断向量) 组织在中断向量表中, CPU 响应时, 由硬件直接产生相应的向量地址, 按地址查表, 取得服务程序入口, 转入相应的服务程序。

中断向量: 中断服务程序入口地址

中断向量表: 存放各中断向量的表 (一段存储区)

向量地址: 访问中断向量表的地址 (指向中断向量首址), 可通过中断号 (中断类型码) 计算得到

(3) 响应过程

(a) 中断周期 IT 流程 (关中断; 保存断点; 获取服务程序入口地址; 转向程序状态 1→FT)

(b) 中断服务程序流程

单级中断: 保护现场、中断服务处理、恢复现场、开中断、返回。

多重中断: 保护现场、送屏蔽字和开中断、中断服务处理、关中断、恢复现场及屏蔽字、开中断、返回。

16、DMA

定义: 直接由硬件实现主存与 I/O 间的数据传送, 传送期间不需 CPU 程序干预。传送结束以后, 通过中断的方式通知 CPU。

- 1) 传送发生在 I/O 与主存之间;
- 2) 由 DMA 控制器来控制传送, DMA 控制器接管总线权, 传送完毕再将总线权交还 CPU

- 3) 只要 CPU 不访存, DMA 操作与 CPU 可并行操作
- 4) 传送前和传送后需要 CPU 程序干预。

特点:

- (1) 响应随机请求
- (2) 一般不影响 CPU 程序的执行, 仅占用总线、无程序切换
- (3) 大批量数据的简单传送

应用场景:

- (1) 主存与高速 I/O 设备之间的简单数据传送。
- (2) 大批量数据采集系统
- (3) 动态存储器 (DRAM) 的自动刷新

17、DMA 的数据传送操作方式

- ① 单字传送: DMA 请求获得批准后, CPU 让出一个总线周期用于字或字节的传送, 再回收并重新判断下一个周期的总线控制权, 也称为周期挪用或窃取。
- ② 成组连续传送方式: DMA 被批准后, 连续占用若干个总线周期, 成组连续批量地传送, 结束后将总线的控制权交回给 CPU。

18、CPU 初始化 DMA 步骤

- ◆ 向接口送出 I/O 设备的寻址信息;
- ◆ 向 DMA 控制器送出控制字, 如传送方向;
- ◆ 向 DMA 控制器送出主存缓冲区首址;
- ◆ 向 DMA 控制器送出传送的数据量;
- ◆

19、DMA 硬件功能

DMA 控制器功能:

- ① 接收初始化信息 (传送方向、主存首址、交换量)
- ② 接收外设 DMA 请求, 判优, 向 CPU 申请总线。
- ③ 接管总线权, 发地址、读/写命令。

接口功能

- ① 接收初始化信息 (外设寻址信息)
- ② 向 DMA 控制器发请求。
- ③ 传送数据。

1、主设备发送数据, 从设备接收数据 ()

X

2、串行接口与系统总线串行传送, 接口与外设串行传送 ()

X

3、有关 I/O 接口的叙述中, 错误的是 ()

- A、状态端口和控制端口可以合并为同一寄存器
 - B、I/O 接口中 CPU 可访问的寄存器称为 I/O 端口
 - C、采用独立编址方式时, I/O 端口地址和主存地址可能相同
 - D、采用同一编址方式, CPU 不能用访存指令访问 I/O 端口
- D

4、下列关于中断 I/O 方式和 DMA 方式的叙述中, 错误的是 ()

- A、中断 I/O 方式请求的是 CPU 处理时间, DMA 方式请求的是总线使用权
- B、中断响应发生在一条指令执行结束后, DMA 响应发生在一个总线周期完成后

C、中断 I/O 方式下数据传送通过软件完成，DMA 方式下数据传送通过硬件完成
中断 I/O 方式下适用于所有外部设备，DMA 方式仅适用于快速外部设备
D

5、程序控制类指令的功能是（ ）

- A、进行算术和逻辑运算
- B、进行主存与 CPU 之间的数据传送
- C、进行 CPU 和 I/O 设备之间的数据传送
- D、改变程序的执行顺序

D

6、采用向量中断方式，是通过（ ）来识别中断来源（中断源）

- A、中断号或者中断类型号
- C、中断优先排队的方式
- B、中断源端口地址译码
- D、提出中断请求的先后顺序

A

7、中断屏蔽和允许中断的描述，以下说法不正确的是（ ）

- A、中断屏蔽是指向 CPU 向接口写入控制字，以控制是否让中断源提出中断请求
- B、允许中断是指 CPU 向接口写入控制字，以允许或不允许中断源提出中断请求
- C、允许中断是 CPU 内部状态位，用于决定是否响应中断请求
- D、中断屏蔽功能的实施一般在中断接口中。

B

8、中断响应周期结束后，CPU 应该转入（ ）

- A、执行周期
- B、取指周期
- C、取源数据周期
- D、取目的数据周期

B

9、为了保证返回主程序后的正常执行，在中断服务程序中，应进行（ ）操作

- A、保存断点地址
- B、关中断
- C、保存现场
- D、开中断

C

10、在有多个中断源的系统中，采用向量中断方式，中断屏蔽字来自于（ ）

- A、中断接口
- B、提出中断请求设备
- C、CPU
- D、向量地址

C

11、在单级中断中，中断服务程序执行顺序是（ ）

- a. 保护现场； b. 开中断； c. 关中断 d. 保存断点； e. 中断事件处理；
- f. 恢复现场； g. 中断返回

A、a→e→f→b→g

B. c→a→e→g

C. c→d→e→f→g

D. d→a→e→f→g

A

12、某机有 4 级中断，优先级从高到低是 1→2→3→4。若将优先级顺序修改，修改后 1 级中断的屏蔽字为 1011，2 级中断的屏蔽字为 1111，3 级中断的屏蔽字为 0011，4 级中断的屏蔽字为 0001，则修改后的优先顺序为_____。

2→1→3→4

13、有 8 个中断源，优先级为 I0>I1>I2>I3>I4>I5>I6>I7，采用多重中断方式，屏蔽字中 Di 用于屏蔽 Ii（为 0 则屏蔽）。CPU 在执行 I2 所对应的中断服务程序时，应写入接口的屏蔽字是_____。

11000000

14、CPU 响应中断必须满足的条件是（）

- A. CPU 接收到中断请求信号
- B. CPU 允许中断

- C. 一条指令执行完毕 D. 以上都是

D

15、某计算机有五级中断 L4-L0，中断屏蔽字为 M4M3M2M1M0， $M_i=1$ ($0 \leq i \leq 4$) 表示 L_i 级中断屏蔽，若中断响应优先级顺序是 $L4 \rightarrow L0 \rightarrow L2 \rightarrow L1 \rightarrow L3$ ，则 $L1$ 的中断处理程序中设置的中断屏蔽字是_____。

01010

16、关于 DMA 方式的说法，正确的是（ ）

- A、DMA 方式利用软件是新数据传送
B、DMA 方式能完全取代中断方式
C、DMA 方式在传送过程中需要 CPU 程序的干预
D、DMA 方式一般用于高速外设数据的简单传送

D

17、由于 DMA 传输发生在外设与内存之间，因此当 CPU 响应 DMA 请求以后，CPU 应将（ ）。

- A、数据总线交给外设； B、总线控制权交给 DMA 控制器.；
C、存储器交给外设； D、内存交给外设。

B

18、下列关于中断方式和 DMA 方式的描述，错误的是（ ）

- A、中断方式请求的是 CPU 处理时间，DMA 方式请求的是总线使用权
B、中断响应发生在一条指令执行结束，DMA 响应发生在总线空闲完成
C、中断方式下数据传送通过软件完成，DMA 方式下数据传送由硬件完成
D、中断方式适用于外部设备，DMA 方式仅适用于快速设备。

D

《微型计算机原理与接口技术》

第 1 部分 指令

1.1 数据传送

1、通用数据传送

MOV dest, src

PUSH OPRD ; 16 位寄存器或者存储器字单元

POP OPRD ; 不能从栈顶弹出一个字给 CS

XCHG REG/MEM, MEM/REG; 两操作数必须有一个是寄存器操作数, **不允许使用段寄存器。**

XLAT ; 输入: 用 BX 的内容代表表格首地址, AL 内容为表内位移量, BX+AL 得到要查找元素的偏移地址; 输出: 将 BX+AL 所指单元的内容送 AL。

2、输入输出

IN acc, PORT ; PORT 为端口地址, acc 为累加器 AL 或 AX

OUT PORT, acc

直接寻址: 8 位端口地址, 地址范围: 0-0FFH。 如 IN AL, 30H

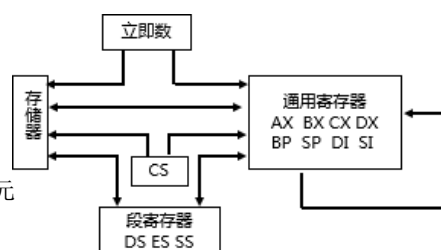
间接寻址: 16 位端口地址, 由 DX 指定端口地址。如 IN AL, DX

3、取偏移地址

LEA REG, MEM

1.2 算术运算

影响标志位 CF、AF、SF、OF、PF、ZF



1、加法运算

ADD OPRD1, OPRD2 ;段寄存器不能作为操作数
ADC OPRD1, OPRD2 ;常用于多字节数相加
INC OPRD ;不影响 CF 标志

2、减法运算

SUB OPRD1, OPRD2
SBB OPRD1, OPRD2
DEC OPRD ;不影响 CF 标志
NEG OPRD
CMP OPRD1, OPRD2

对于无符号数:

- (1) ZF=1, OPRD1=OPRD2
- (2) CF=0, OPRD1>OPRD2
- (3) CF=1, OPRD1<OPRD2

对于有符号数:

- (1) ZF=1, OPRD1=OPRD2
- (2) OF=SF, OPRD1>OPRD2
- (3) OF≠SF, OPRD1<OPRD2

3、逻辑运算

; AND、OR、XOR、TEST 使标志位 OF=CF=0;
AND OPRD1, OPRD2
OR OPRD1, OPRD2
NOT OPRD ;对标志位无影响
XOR OPRD1, OPRD2
TEST OPRD1, OPRD2

4、移位

移动次数一般用 CL 来指定, 若只移位 1 次, 也可以在指令中直接写常数“1”。

SAL OPRD, CL ;有符号, D7
SHL OPRD, CL ;无符号
SAR OPRD, CL ;有符号, D7→D7
SHR OPRD, CL ;无符号
ROL OPRD, CL
ROR OPRD, CL
RCL OPRD, CL
RCR OPRD, CL

多字节单元数据联合移位:

SAL M, 1
RCL M+2, 1
RCL M+4, 1

1.3 串操作

源串一般存放在数据段, 偏移地址由 SI 指定。允许段重设; 目标串必须在附加段, 偏移地址由 DI 指定。

指令自动修改地址指针, 修改方向由 DF 决定。DF=0, 增地址方向; DF=1, 减地址方向; 用指令 CLD 或者 STD, 设置 DF 值。

数据块长度值由 **CX** 指定。

```
REP  MOVSB      ;初始化 SI、DI、CX、DF
REP  MOVSW      ;初始化 SI、DI、CX、DF
REPE  CMPSB     ;初始化 SI、DI、CX、DF
REPNE CMPSW     ;初始化 SI、DI、CX、DF
REPNZ SCASB     ;初始化 AL、DI、CX、DF
REPNZ SCASW     ;初始化 AX、DI、CX、DF
LODSB          ; 初始化 SI，输出 AL
LODSW          ; 初始化 SI，输出 AX
REP  STOSB      ;初始化 AL、DI、CX、DF
REP  STOSW      ;初始化 AX、DI、CX、DF
```

1.4 程序控制

```
JMP      Label
JMP WORD PTR[BX]
JC/JNC  Label   ; 条件转移指令均为段内（标号）短转移，即转移范围为：-128~+127
JZ/JNZ
JO/JNO
JP/JPE
JA/JAE/JB/JBE 无符号数比较大小
JG/JGE/JL/JLE 符号数比较大小
```

```
LOOP Label ;初始化 CX
CALL OPRD
RET
```

CLI、STI、CLC、STC、CMC、CLD、STD

第二部分 伪指令

2.1 标识符

指令语句中的标号和伪指令语句中符号名统称为**标识符**。

命名规则：

- ◆ 1. 字符的个数为 1~31 个；
- ◆ 2. 第一个字符必须是字母、问号、@或下划线 “_” 这 4 种字符之一；
- ◆ 3. 从第二个字符开始，可以是字母、数字、@、“_” 或问号 “?”；
- ◆ 4. 不能使用系统专用的保留字。

保留字：

- CPU 中各寄存器名（如 AX、CS 等）
- 指令助记符（如 MOV、ADD）
- 伪指令符（如 SEGMENT、DB）
- 表达式中的运算符（如 GE、EQ）以及属性操作符（如 PTR、OFFSET 等）

2.2 汇编语言数据

2.2.1 常数

1、表示形式：

01001001B、2540、0FEH、2.134 E +10。如果常数的第一个数符为字母，为了与标识符区别，必须在其前面冠以数字 “0”。

2、字符串常数：

用引号（单引号或双引号）括起来的一个或多个字符，其值为字符的 ASCII 码值。

2.2.2 变量

给变量在内存中分配一定的存储单元。

1、一般格式

变量名 {	DB	表达式1, 表达式2.....;	; 定义字节变量	VAR_DATA	SEGMENT
	DW		; 定义字变量		DATA1 DB 12H
	DD		; 定义4字节变量		DATA2 DB 20H,30H
	DQ		; 定义8字节变量		DATA3 DW 5678H
	DT		; 定义10字节变量		VAR_DATA ENDS

2、变量赋初值

(1) 字符串表达式：

对于 DB 伪指令：字符串按从左到右，一个字符分配一个字节单元。将字符的 ASCII 码以地址递增的排列顺序依次存放。

对于 DW 伪指令：两个字符的存放顺序是前一个字符放在高地址，后一字符放低地址单元。

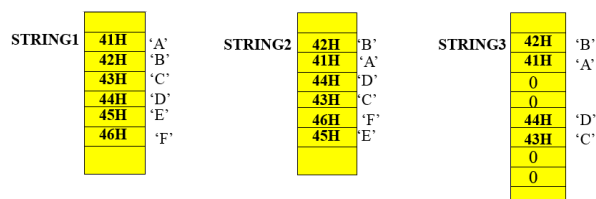
对于 DD 伪指令，只能给两个字符组成的字符串分配 4 个字节单元。两个字符存放在较低地址的两个字节单元中。

注意：DW 和 DD 伪指令不能用两个以上字符构成的字符串赋初值，否则将出错。

如 STRING1 DB 'ABCDEF'

STRING2 DW 'AB', 'CD', 'EF'

STRING3 DD 'AB', 'CD'



(2) DUP 表达式

变量名 $\left\{ \begin{array}{l} \text{DB} \\ \text{DW} \\ \text{DD} \end{array} \right\}$ 表达式1 DUP(表达式2)

其中：表达式 1 是重复的次数，表达式 2 是重复的内容。

如 DATA_B DB 20H DUP('AB')

3、变量的使用

(1) 在指令语句中引用

- ✓ 在指令语句中直接引用变量名就是对其存储单元的内容进行存取。

如 DA1 DB 0FEH

MOV AL, DA1 ;将 0FEH 传送到 AL 中

- ✓ 当变量出现在变址（基址）寻址或基址变址寻址的操作数中表示取用该变量的**偏移量**。

如 DA3 DB 10H DUP(?)

MOV DA3[SI], AL

(2) 在伪指令语句中引用

如 NUM DB 75H

ARRAY DW 20H DUP(0)

ADR1 DW NUM ;取变量地址的**偏移量**

ADR2 DD NUM ;取变量段基值和偏移量。

NUM	75H
ARRAY+0	00
+1	00
:	:
+3F	00
ADR1	04
:	:
ADR2	04
:	00
:	15H
:	09H
ADR3	07
:	00
:	:

设段基值为 0915H，NUM 的偏移量为 0004H，则存储单元的分配情况如图所示。

2.2.3 标号

标号加在一条指令的前面，它就是该指令在内存的存放地址的符号表示，也就是指令地址的别名。

距离属性（也叫类型属性）指定

(1) 隐含方式

当标号加在指令语句前面时，它隐含为 NEAR 属性。如 SUB1: MOV AX, 30H

(2) 用 LABEL 伪指令给标号指定距离属性

SUB1_FAR LABEL FAR

SUB1: MOV AX, 30H

2.3 符号定义语句

1、等值语句

(1) 常数或数值表达式，如 COUNT EQU 5

(2) 地址表达式，如 ADR1 EQU DS: [BP+14]；ADR1 为在 DS 中以 BP 作基址寻址的一个存储单元。

(3) 变量、寄存器名或指令助记符，如 CREG EQU CX

注意：在同一源程序中，同一符号不能用 EQU 定义多次。

2、等号语句

(1) 等号语句可以对一个符号进行多次定义。

如 NUM=14H

NUM=NUM+10H

(2) 等号语句不能为助记符定义别名

如 CBD=DAA；错误语句

注意：等值语句与等号语句都不会为符号分配存储单元。所定义的符号没有段、偏移量和类型等属性。

2.4 表达式与运算符

1、算术运算符

+、-、*、/、MOD、SHL、SHR、[]

- (1) 运算符“+”和“-”也可作单目运算符，表示数的正负
- (2) 使用“+”、“-”、“*”、和“/”运算符时，参加运算的数和运算结果都是整数。
- (3) “/”运算为取商的整数部分，而“MOD”运算取除法运算的余数。
- (4) “SHR”和“SHL”为逻辑移位运算符。空位补0，如：MOV AX, NUM SHL 1

注意：移位运算符的操作对象是某一具体的数（常数），在汇编时完成移位操作。而移位指令是对一个寄存器或存储单元内容在程序运行时执行移位操作。

- (5) 下标运算符“[]”具有相加的作用。

如 MOV AX, DA_WORD[20H]

MOV AX, DA_WORD+20H

下列语句错误：

MOV AX, ARRAY+DA_WORD

MOV AX, ARRAY+BX+SI

2、逻辑运算符

NOT、AND、OR、XOR，如 MOV AX, NOT 0FOH

3、关系运算符

EQ(等于)、NE(不等于)、LT(小于)、LE(小于等于)、GT(大于)、GE(大于等于)

- (1) 若是常量，则按无符号数进行比较；若是变量，则比较它们的偏移量的大小。
- (2) 关系运算的结果只能是“真”（全1）或“假”（全0）

4、取值运算符和属性运算符

SEG、OFFSET、PTR

MOV AX, WORD PTR DA_BYTE[10]

5、运算符的优先级

优先级别	运算符
(最高) 1	LENGTH, SIZE, 圆括号
2	PTR, OFFSET, SEG, TYPE, THIS
3	HIGH, LOW
4	*, /, MOD, SHR, SHL
5	+, -
6	EQ, NE, LT, LE, GT, GE
7	NOT
8	AND
(最低) 9	OR, XOR

2.5 程序的段结构

1、一般格式

段名 SEGMENT [定位类型] [组合类型] ['类别名']

.....
..... } 本段语句序列
.....

段名 ENDS

(1) 定位类型

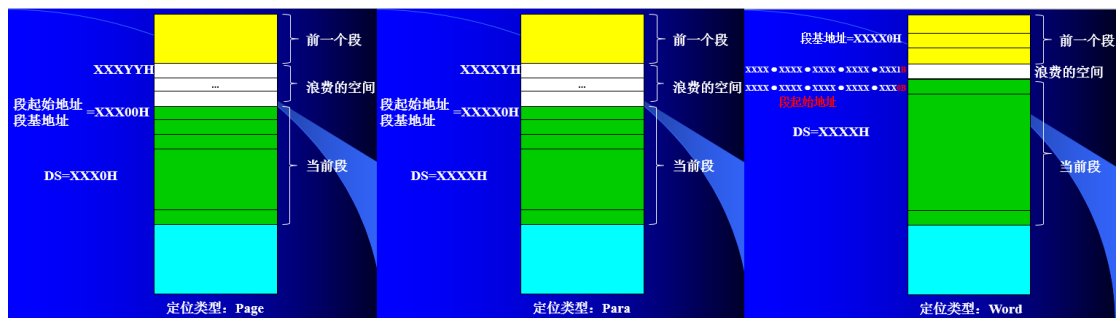
PAGE: 表示该段从一个页面的边界开始存放数据

PARA: 表示该段从一个小节的边界开始存放数据，缺省为 PARA。

WORD: 表示该段从一个偶数字节地址开始存放数据。

BYTE: 表示该段起始数据单元地址可以是任一地址值。

注意：定位类型为 PAGE 和 PARA 时，段基址直接选用段起始地址，即它们是重合的。定位类型为 WORD 和 BYTE 时，段基址与段起始地址可能不同。



(2) 组合类型

用来指定段与段之间的连接关系和定位。

NONE: 若未指定组合类型, 表示本段与其它段无连接关系。

PUBLIC: 在满足定位类型的前提下, 将与该段同名的段邻接在一起, 形成一个新的逻辑段, 共用一个**段基址**。

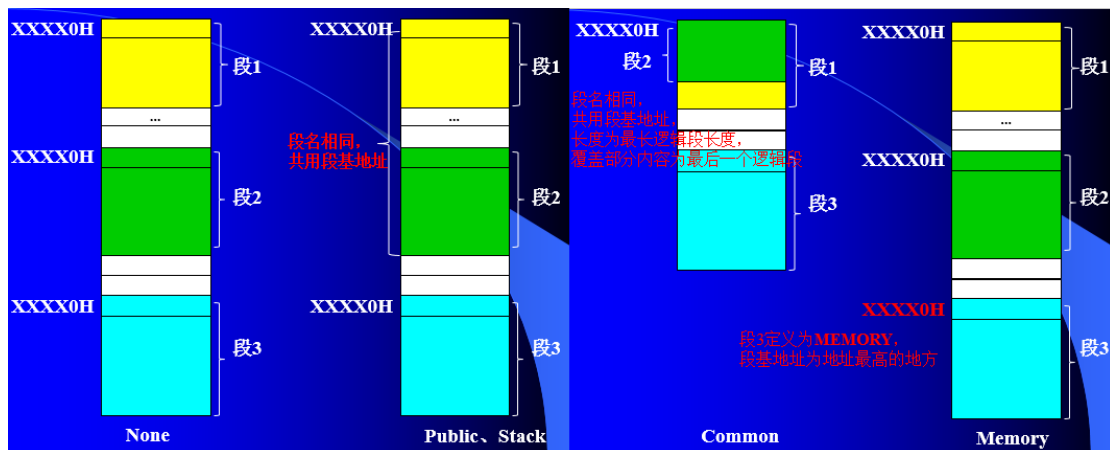
COMMON: 产生一个覆盖段。在多个模块连接时, 把该段与其它也用 COMMON 说明的同名段置成相同的段基址, 这样可达到共享同一存储区。

STACK: 把所有同名段连接成一个连续段, 且**系统自动对 SS 段寄存器初始化为该连续段的段基址**。并初始化堆栈指针 SP。

用户程序中应至少有一个段用 STACK 说明, 否则需要用户程序自己初始化 SS 和 SP。

AT 表达式: 表示本段可定位在表达式所指示的小节边界上。表达式的值也就是**段基址**。

MEMORY: 表示本段在存储器中应定位在所有其它段之后的最高地址上。如果有多个用 MEMORY 说明的段, 则只处理第一个用 MEMORY 说明的段, 其余的被视为 COMMON。



(3) 类别名

类别名为某一个段或几个相同类型段设定类型名称。系统在进行连接处理时, 把类别名相同的段存放在相邻的存储区, 但段的划分与使用仍按原来的设定。

2、段寻址伪指令

ASSUME 的作用是告诉汇编程序, 在处理源程序时, 定义的段与哪个寄存器关联。

如 ASSUME CS:CODE, DS:DATA1, ES:DATA2

3、段寄存器的装入

(1) DS 和 ES 的装入, 用传送数据指令。

如 MOV AX, DATA1

MOV DS, AX

(2) SS 装入

- ✓ 在段定义伪指令的组合类型项中，使用 STACK 参数，并在段寻址伪指令 ASSUME 语句中把该段与 SS 段寄存器关联。
- ✓ 如果在段定义伪指令的组合类型中，未使用 STACK 参数，或者是在程序中要调换到另一个堆栈，这时，可以使用传送数据指令装入方法。

如 DATA_STACK SEGMENT
DB 40H DUP(?)
TOP LABEL WORD
DATA_STACK ENDS
CODE SEGMENT
.....
MOV AX, DATA_STACK
MOV SS, AX
MOV SP, OFFSET TOP

(3) CS 的装入

- ✓ 由系统软件按照结束伪指令指定的地址装入初始的 CS 和 IP。
根据 END 伪指令的起始地址。
END 起始地址
- ✓ 段间调用 CALL 和 RET、JMP、中断服务程序。

4.6 子程序

过程名 PROC [NEAR/FAR]
.....
RET
.....
过程名 ENDP

第3部分 8086/8088 中断系统

3.1 中断类型

1、内部中断

0=除法出错中断；1=单步中断；3=断点中断；4=溢出中断；n 型中断（DOS 软中断【20H-3FH】和 BIOS【11H-1AH】软中断）：INT n。

中断响应过程：**无#INTA，中断类型码固定或者指令直接给出。**

- (1) 将类型码乘 4, 得到中断向量地址
- (2) 硬件保护现场, 将 FLAGS 压栈
- (3) 清除 TF 和 IF 标志
- (4) 保存断点, CS 和 IP 压栈
- (5) 中断服务程序入口地址送 CS 和 IP
- (6) 执行中断服务程序。

2、外部中断

- (1) 非屏蔽中断

2=NMI

- (2) 可屏蔽中断

08H~0FH: 主 8259 管理, 包括定时器、键盘、串口、并口、软盘驱动器。

70H~77H: 从 8259 管理, 包括实时时钟、协处理器、硬盘控制器。

- (3) 可屏蔽中断响应

两个中断响应总线周期。

第一个总线周期: 地址/数据总线置高阻 (ALE=0), 送出#INTA 响应信号, 禁止其他总线请求。
最大模式下, 送#LOCK 信号, 禁止其他处理器访问总线。

第二个总线周期: 送第 2 个#INTA, 通知 8259 送出中断类型码 n。

3、中断向量表

向量地址= $n \times 4$

用户中断服务程序, 需要用户主程序初始化中断向量表中, 该中断号对应的服务程序地址。如将类型码为 48H 的中断服务子程序 TIMER 的中断向量放入向量表。

```
• MOV AX, 0000H
• MOV DS, AX
• MOV SI, 0120H
• MOV BX, OFFSET TIMER
• MOV [SI], BX
• MOV BX, SEG TIMER
• MOV [SI+2], BX
```

3.2 中断控制器 8259

3.2.1 8259A 中断处理过程

- 1、中断请求输入线 (IR0-IR7) 上有中断请求, 则 IRR 相应位=1
- 2、若 IMR 该位未屏蔽, 则 8259 由 INT 引脚向 CPU 发出中断请求信号 INTR
- 3、CPU 处于开中断, CPU 执行完指令后, CPU 用#INTA 信号作为对 INTR 的响应。
- 4、8259 收到 CPU 的第 1 个#INTA 脉冲后, 将最高优先权的 ISR 位置 1, 并使 IRR 相应位复位。
- 5、8259 在第 2 个总线周期, 将中断类型号送到数据总线上。
- 6、若 8259 工作在 AE0I, 8259 在第 2 个#INTA 结束时, 复位 ISR 相应位。非 AE0I, 则需要 CPU 在中

断服务程序结束时，写入操作命令字 OCW₂ 中置位 EOI 位，8259 接收到名字字后将 ISR 的该位复位。

3.2.2 8259 工作方式

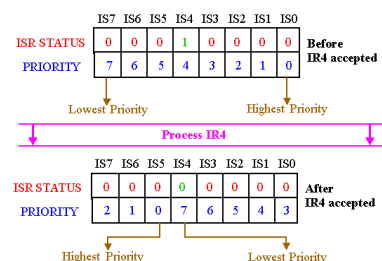
1、中断嵌套

- ✓ 不允许嵌套
- ✓ 全嵌套（普通嵌套，允许优先级高的中断请求）
- ✓ **特殊全嵌套**（允许响应同级中断请求）P270 图 6-28

在多片 8259A 级联的情况下，**主片**通常设置为特殊完全嵌套方式，**从片**设置为普通嵌套方式。当主片响应某一个从片的中断请求时，从片中的 IR₇~IR₀ 的请求都是通过主片中的某个 IR_i 请求引入的。因此从片的 IR₇~IR₀ 对于主片 IR_i 来说，它们属于同级，只有主片工作于特殊完全嵌套方式时，从片才能实现完全嵌套。

2、中断优先方式

- ✓ 固定优先级（从高到低顺序为 IR₀~IR₇）
- ✓ 优先级自动循环方式（多个中断源优先级相等的场合）
 - ◆ 普通自动循环：初始优先级为默认，从高到低顺序为 IR₀~IR₇。
 - ◆ 特殊自动循环：初始的最低优先级是由 OCW₂ 确定



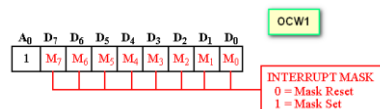
开始的时候，优先级从高到低是 **IR₀**，**IR₁**，**IR₂**，**...**，**IR₇**。某时刻，IR₄ 正在被 CPU 处理。处理完成后，IR₄ 的优先级变为最低，IR₅ 的优先级变为最高，也就是说，新的优先级序列从高到低是 **IR₅**，**IR₆**，**IR₇**，**IR₀**，**IR₁**，**IR₂**，**IR₃**，**IR₄**。

3、中断结束方式

- ✓ 自动中断结束 AEOI：**适用于没有中断嵌套的情况**，初始化命令字 ICW₄ 的 AEOI 位置 1 实现。
- ✓ 常规中断结束 EOI：**全嵌套**，操作命令字 OCW₂ 中的 EOI=1、SL=0、R=0。输出到 8259A 的偶地址端口（A₀=0）
- ✓ 特殊中断结束 SEOI：**非全嵌套** OCW₂ 的 EOI=1、SL=1、R=0，**L₂**、**L₁**、**L₀** 这三位指出了对 ISR 中的**哪一位**进行清除。

4、屏蔽中断源方式

- ✓ 常规屏蔽：IMR 某位为“1”表示屏蔽对应的中断请求。



- ✓ 特殊屏蔽 SMM：8259A 工作在特殊屏蔽方式时，所有未被屏蔽的优先级中断请求（较高的和较低的）均可在某个中断过程中被响应，即低优先级级别的中断可以打断正在服务的高优先级中断。

在特殊屏蔽方式中，可在**中断服务子程序**中用中断屏蔽命令屏蔽当前正在处理的中断级，同时可使其在 ISR 中的对应位清零，这样一来不仅屏蔽了当前正在处理的中断级，而且也真正开放了较低级别的中断请求。在这种情况下，虽然 CPU 仍然继续执行较高级别

的中断服务子程序，但由于 ISR 中对应位已经清零，就如同没有响应该中断一样。所以，此时对于较低级别的中断请求，CPU 可以响应。

特殊屏蔽方式由于打乱了正常的全嵌套结构，低优先级中断可以打断正在服务的高优先级中断，高优先级中断也可以打断正在服务的低优先级中断，此时，根据 ISR 的内容无法确定出刚刚所处理的中断。**只能用 SEOI 命令**（命令中指定要复位的中断级）来清除 ISR 中的指定位。

5、中断触发方式

8259A 中断请求输入端 $IR_7 \sim IR_0$ 的触发方式有电平触发和边沿触发两种，由初始化命令字 ICW1 中的 LTIM 位来设定。

(1) 电平触发

当 LTIM=1 时，为电平触发方式。当 8259A 检测到 IR_i ($i = 0 \sim 7$) 端有高电平时产生中断。在这种触发方式中，要求触发电平必须保持到中断响应信号 #INTA 有效为止。在 CPU 响应中断后，应及时撤销该请求信号，以防止 CPU 再次响应，出现重复中断现象。

(2) 边沿触发

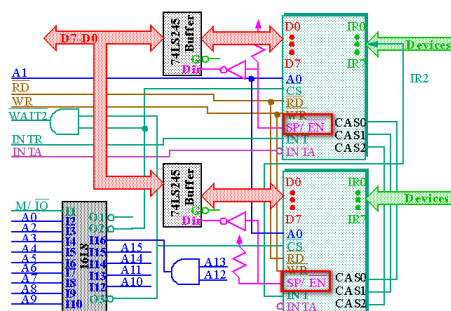
当 LTIM=0 时，为边沿触发方式。当 8259A 检测到 IR_i 端有由低到高的跳变（上升沿）信号时产生中断。

5、数据总线连接方式

8259A 的数据线与系统数据总线的连接有缓冲和非缓冲两种方式。

(1) 缓冲方式

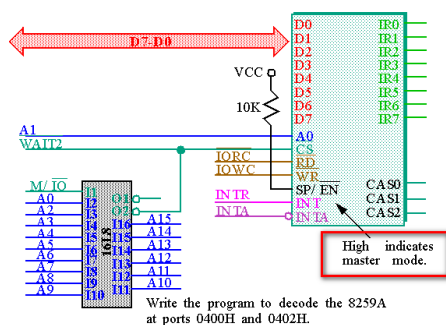
如果 8259A 通过总线驱动器和系统数据总线连接，则应选择缓冲方式。此时 EN 为输出引脚。在 8259A 输出中断类型号的时候，EN 输出一个低电平，用此信号作为总线驱动器的启动信号。在缓冲方式下，由 ICW4 的 M/S 位来标识本 8259A 是主片还是从片。



(2) 非缓冲方式

如果 8259A 的数据线与系统数据总线直接相连，那么应选择非缓冲方式。此时 SP 为输入引脚，用其电平高低来标识本 8259A 是主片 (SP=1) 还是从片 (SP=0)。在非缓冲方式下，ICW4 的 BUF=0，M/S 位无意义。

•A single 8259A connected in the 8086.



2.5.3 8259 编程

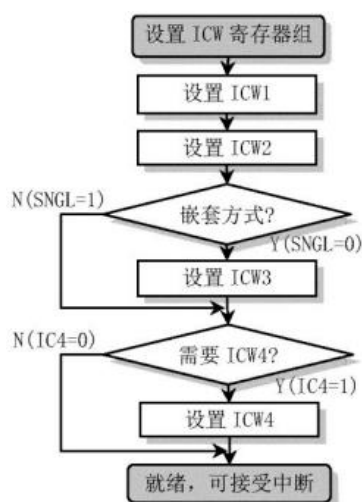
命令字端口地址：

命令字	A0	主片端口地址	从片端口地址	备注
ICW1	0	0x20	0xA0	D4 = 1
ICW2	1	0x21	0xA1	
ICW3	1	0x21	0xA1	
ICW4	1	0x21	0xA1	
OCW1	1	0x21	0xA1	
OCW2	0	0x20	0xA0	D4.D3 = 00
OCW3	0	0x20	0xA0	D4.D3 = 01

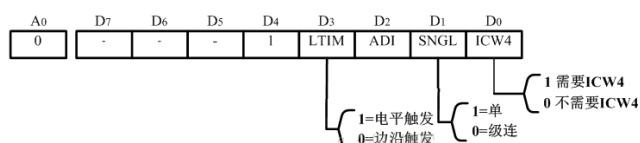
1、初始化命令字 ICW

在 8259A 可以正常工作之前,必须首先设置初始化命令字 ICW (Initialization Command Words) 寄存器组的内容。而在其工作过程中,则可以使用写入操作命令字 OCW (Operation Command Words) 寄存器组来随时设置和管理 8259A 的工作方式。

A0 线用于选择操作的寄存器。在 PC/AT 微机系统中,当 A0=0 时芯片的端口地址是 0x20(主芯片)和 0xA0(从芯片);当 A0=1 时端口就是 0x21(主芯片)和 0xA1(从芯片)。

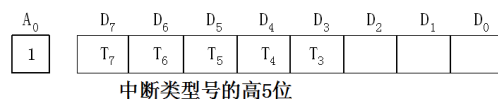


(1) 初始化字 ICW1

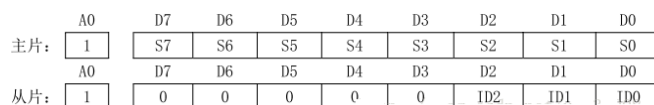


收到 ICW1 后, 8259 完成以下工作: 清除 ISR、IMR; 初始化优先级 IR0-IR7; 普通屏蔽; 非自动 EOI; 读出 IRR。

(2) 中断向量码 ICW2



(3) 级联控制字 ICW3

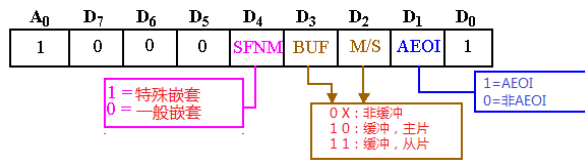


主芯片的端口地址是 0x21, 从芯片的端口地址是 0xA1。

对于主片, Si=1, 表示 IR_i 接从片的 INT 引脚。主片 S7~S0 各比特位对应级联的从片。

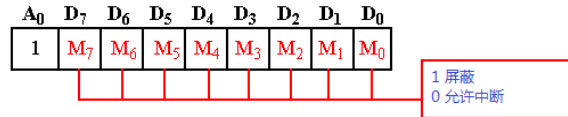
对于从片, ID2~ID0 对应各从片的标识号, 即连接到主片的中断级。

(4) ICW4 中断结束

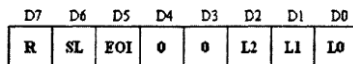


2、操作命令字 OCW

(1) 中断屏蔽字 OCW1



(2) 中断结束和优先级循环 OCW2



D₂D₁D₀ (L₂L₁L₀) : 当 D₆=1 时, 这三位才有效。此时有两个用处: 第一是在发特殊结束中断命令时, 三位指出要清除 ISR 寄存器中的哪一位; 第二是在特殊优先级循环方式中, 三位指出循环开始时初始优先级队列的最低优先级。

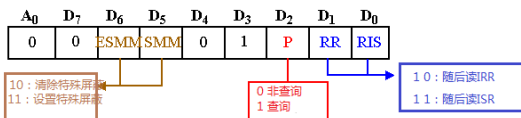
D₄D₃ : OCW2 的特征位, 恒等于 0;

D₅ (EOI) : 中断结束命令位。1=在中断结束后需发送中断结束命令, 使 ISR 相应位清 0; 0=不需要发送中断结束命令, 即为中断自动结束方式。

D₆ (SL) : 低三位有效位。1=低三位有效, 0=低三位无效。

D₇ (R) : 优先级循环方式位。1=设置优先级循环方式, 0=非循环方式。

(3) 屏蔽方式和状态读出控制字 OCW3



✓ **查询方式:** 对同一地址读, 读出本片 8259 的中断请求。



✓ 读 8259 状态

IRR 和 ISR : 由 RR、RIS 两位确定, 且和 OCW3 同一地址读取。

IMR : A₀=1 (即主片 21H 或者从片 A1H) 时, 读到的就是 IMR 的内容。