**《计算机组成原理与结构》重要知识点**

1. **概论**
2. 基本概念
3. 冯.诺依曼体制？存储程序方式？
4. 冯诺依曼体制：
5. 采用二进制形式表示数据和指令
6. 采用存储程序方式工作（核心）
7. 由五大部分组成计算机系统的硬件，并规定了这五部分的基本功能
8. 存储程序方式：
9. 事先编制程序
10. 事先将程序存入计算机
11. 自动连续运行程序
12. 控制流？数据流？

控制流：控制计算机工作的信息，即指令、命令

数据流：计算机加工处理的对象，即数值型数据、非数值型数据

1. 模拟信号？数字信号？数字信号有哪两种？
2. 模拟信号：随时间连续变化
3. 数字信号：在时间或空间上断续变化。E.g.电平（并行）脉冲（串行）
4. 总线及其组成？

（1）总线：能为多个部件分时共享的一组信息传送线路及相应的控制逻辑

（2）按功能（内总线、系统总线、局部总线、外总线），按信息（地址总线、数据总线、控制总线），格式（并行总线、串行总线），时序（同步总线、异步总线），方向（单向总线、双向总线）

1. 接口的概念？

部件与部件（软件或硬件）之间的交接部分称为接口

1. 基本字长？

参加一次定点运算的操作数的位数

1. CPU主频？时钟频率？

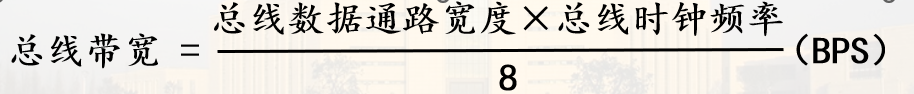
CPU主频：计算机振荡器输出的脉冲序列的频率，是计算机中一切操作的时间基准信号，决定计算机工作的快慢。

时钟频率：主频脉冲经分频后形成的时钟脉冲序列的频率，两个相邻的时钟脉冲之间的间隔时间是一个时钟周期时间，也称为节拍。

1. 数据通路宽度？数据传输率？

数据通路宽度：数据总线能并行传送的数据的位数

数据传输率：数据总线每秒传送的数据量。



**9.**

**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）**

1. **计算机中的信息表示**
2. 基本概念
3. 一个数值型数据的完整表示需三个方面？

进位计数制、编码方式（源码、补码、反码）、小数点处理（定点、浮点）

1. 权、基数？

基数：进位制中产生进位的数值，每个数位上允许的最大码值；

权：位权，例如十进制百位的位权为102

1. 真值？机器数的定义？分类？

真值：用正负符号加绝对值表示的数

机器数：连同数符一起数字化的数。原码、反码、补码、移码

1. 数的定点表示与浮点表示？
2. 定点数：
3. 定点整数：小数点隐含在最低位之后

带符号定点整数：符号在最高位

无符号定点整数：全部数位表示数值

1. 带符号定点小数：小数点在最高位之后，最高位是符号位
2. 浮点数：

N = ±RE x M

E：阶码

M：尾数，为定点小数，规格化要求（要求1/2≤|M|<1）最高有效位为1（除了符号位的第一位）

原码规格化要求：尾数第1位为1，补码规格化要求尾数第1位与符号位相反

1. 指令？指令系统？

指令：一系列按照某种规律有序排列的，能被CPU识别、执行的二进制代码。

指令系统（或集）：一台计算机所能执行的全部指令。

1. 地址码？寻址方式？地址结构？
2. 寻址方式：

大致可将众多的寻址方式归纳为以下四大类，其它的寻址方式则是它们的变型或组合。

① 立即寻址。在读取指令时也就从指令之中获得了操作数，即操作数包含在指令中。

② 直接寻址类。直接给出主存地址或寄存器编号，从CPU内或主存单元内读取操作数。

③ 间接寻址类。先从某寄存器中或主存中读取地址，再按这个地址访问主存以读取操作数。

不影响标志寄存器的值出的是形式地址（不是最终地址），经过某种变换（例如相加、相减、高低位地址拼接等），才获得有效地址，据此访问主存储器以读取操作数。

8.何谓隐式I/O指令？其主要特点是什么？

所谓隐式I/O指令，就是用传送指令实现I/O操作，即针对统一编址，用传送指令访问I/O端口

不影响标志寄存器的值

9.主机调用外围设备，外设编制可采用那几种方式？

（1）对外设单独编址：单独编址到寄存器级、单独编址到设备级

（2）外设与主存统一编址：将外设接口有关存储器视作主存单元分配一个存储单元地址

二、计算题

1.扩展操作码：根据条件计算指令条数？

主要是判断操作码位数，看操作码有多少种

2.根据寻址方式寻找操作数？

3.IEEE754短浮点数格式表示？

1. **中央处理器**
2. 基本概念
3. CPU内部各寄存器的功能？

通用寄存器：R0,R1,R2,R3（可编程）

暂存器：C,D

指令寄存器：IR

程序计数器：PC（可编程）

程序状态字寄存器：PSW（可编程）

地址寄存器：MAR

数据缓冲寄存器：MDR

堆栈指针寄存器：SP（可编程）

1. 数据通路结构？

见书133

3.何谓同步控制方式，何谓异步控制？有何主要特征？应用场合？

同步控制：各项操作与统一的时序信号同步。

异步控制：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。

同步控制有固定时钟周期，而异步控制方式没有。

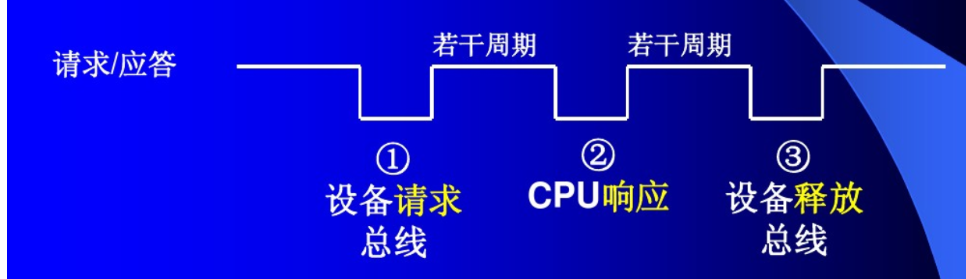
4.何谓主从设备，试举例说明。

主设备：能申请并掌握总线控制权的设备

从设备：响应主设备要求的设备，不能掌握总线控制权

DMA控制方式中，DMA是主设备，I/O是从设备

5.同步控制中如何引入异步应答的，试举例说明.  
以固定时钟周期作为时序基础，引入应答思想  
例如： 8088最大模式，用一根总线请求/应答线实现总线权的转移。



6.微程序控制思想的基本要点是什么？

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。

7.判断下面叙述是否正确，说明理由：串行加法器中的进位链是串行进位链，并行加法器中的进位链只有并行进位链。

错，并行加法器也有串行进位链

8.常用的加法器进位链结构有哪几种？

串行进位、并行进位、分组进位（组内并行组间并行）

9.组合逻辑控制器、微程序控制器有何区别？

产生微命令的方式不同。

组合逻辑控制器通过组合逻辑电路产生微命令

微程序控制器通过微指令译码产生微命令。

10.组合逻辑控制器、微程序控制器的时序系统是如何划分的？

组合逻辑控制器：工作周期（FT,ST,DT,ET,IT,DMAT）>时钟周期（节拍）>脉冲

微程序控制器：指令周期 > 微指令周期

11.微命令、微操作、微指令、微指令周期、微程序？

微命令：EMAR,R,SIR

微指令：M->MDR->C

微操作：计算机工作过程电路实现的最基本操作，一条微指令控制实现一步微操作

微指令周期：读取一条微指令并执行相应的一步操作所需的时间，为一个时钟周期

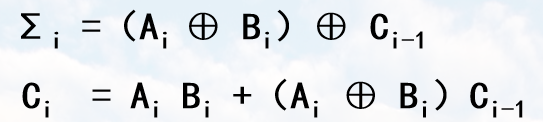
微程序：一系列微指令的有序集合称为微程序，用来解释执行一条机器指令（MOV R0,R1）

微命令 -> 微指令(执行周期为为微周期) -> 微程序(解释一条机器指令)

12.1位全加器的结构及关系表达式。

结构：三个输入两个输出

关系表达式：



13.并行加法器中的串行进位链结构：Cn = Gn + PnCn-1

并行进位链结构：Cn = Gn + PnGn-1+ … + Pn…P1C0

14.时序信号？时序系统？

（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

二、设计题：

CPU的逻辑组成及工作机制

1.CPU的逻辑组成（模型机框图）

（1）CPU的逻辑组成→模型机框图；

（2）CPU内每个寄存器的作用；

（3）总线的分类及定义；

（4）控制器的分类及区别；

2.CPU的指令流程

（1）指令类型：MOV指令、双操作数算数逻辑运算指令、单操作数算是逻辑运算指令、转移/返回指令、转子指令；

（2）核心是寻址方式：立即寻址、R、(R)、—(R)、(R)+、@(R)+、X(R)；

3.操作时间表的安排（微命令的安排）：

（1）CPU数据通路操作：按照数据的流向分成四段

ALU输入选择→AUL功能选择→移位器功能选择→分配脉冲（打入到寄存器中的脉冲）；

1. 与访问主存有关的微命令。

4.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。

1. **存储子系统**
2. 基本概念
3. 主存、缓存、外存？

主存：能由CPU直接编程访问的存储器，存放当前CPU需要执行的程序和需要处理的数据。其特点是：能随机访问、工作速度快、容量有限。

缓存：存放的是最近要使用的程序和数据，作为主存中当前活跃信息的副本。特点：速度很快、容量小。

外存：存放大量的后备程序和数据。特点：速度较慢、容量大。

三级存储体系结构：高速缓存（cache）——主存——外存（辅存）

1. 高速缓存Cache用来存放什么内容？设置它的主要目的是什么？

高速缓存存放的是最近要使用的程序和数据，作为主存中当前活跃信息的副本。

主要目的：为了缓解CPU与主存在存取速率上的不匹配

1. 存储介质？
2. 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：利用双稳态电路交叉反馈机制存储信息，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。

1. 何谓随机存取？何谓顺序存取？何谓直接存取？请各试举一例。

随机存取：可按地址随机地访问任意存储单元；访问各存储单元的读/写时间相同，与地址无关。

e.g.主存与高速缓存Cache

顺序存取：存储器的信息按记忆块组织、顺序存放；访问时间与信息存放位置有关。

e.g.磁带采用顺序存取方式

直接存取：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

e.g.磁盘

5. 静态存储器SRAM、动态存储器DRAM存储原理？

静态存储器SRAM：依靠双稳态电路内部交叉反馈机制存储信息。

动态存储器DRAM：依靠电容存储电荷的原理存储信息。

1. 动态刷新分为哪几种情况，各有什么特点？

集中刷新：2ms内集中安排所有刷新周期。用在实时要求不高的场合

分散刷新：各刷新周期分散安排在存取周期中。用在低速系统中。

异步刷新：各刷新周期分散安排在2ms内。每隔一段时间刷新一行。用在大多数计算机中。

Ps: 刷新一行所用的时间——刷新周期（存取周期）

1. 全地址译码方式？部分地址译码方式？

全地址译码：将除了与芯片连接的地址以外的所有高位地址用于译码产生片选信号

部分地址译码：将除了与芯片连接的地址外的部分高位地址用于译码产生片选信号

二、设计题：

半导体存储器逻辑设计：(地址分配、片选逻辑、逻辑框图)，片选逻辑采用全/部分译码方式。

略。

1. **I/O系统**
2. 基本概念
3. 系统总线上一般包含哪三组信息？

数据信息、地址信息、控制信息

1. 中断方式定义、实质、特点？

定义：CPU暂时中止现行程序的执行，转去执行为某个随即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

方法：保存断点，保护现场；恢复现场，返回断点；

时间：一条指令结束时切换，保证程序的完整性

1. 中断向量、中断向量表、向量地址？

中断向量：中断服务程序入口地址和程序状态字

中断向量表：存放中断向量的一段存储区

向量地址：访问中断向量表的地址码，即读取中断向量所需要的地址

1. 何谓向量中断？何谓非向量中断？各有何优点和缺点？

向量中断：将中断向量组织在中断向量表中；响应中断时由硬件产生对应于中断源的向量地址；据此访问中断向量表，从中读取中断服务程序入口地址，转向中断服务程序的执行；此过程在中断周期IT中由硬件完成。

非向量中断：CPU响应中断时产生一个固定地址，由此读取中断查询程序的入口，从而转向查询程序，通过软件查询确定优先批准的中断源，然后进入相应的中断服务程序。

1. 单级中断、多重中断？

单级中断：CPU响应后只处理一个中断源的请求，处理完成后才可能相应新的请求

保护现场——具体服务处理——恢复现场——开中断、返回

多重中断：在某次中断服务过程中，允许响应处理更高级别的中断请求。

保护现场——开中断、送新屏蔽字——具体服务处理——关中断——恢复现场及原屏蔽字——开中断、返回

1. 程序直传方式定义、实质、特点？

定义：依靠CPU直接执行相关I/O程序来实现对数据输入和输出的控制

特点：硬件开销小，并行度差，实时处理能力差

1. DMA方式定义、实质、特点

定义：依靠硬件在主存与外围设备之间进行数据传送，在传送过程中不需要CPU干预

实质：总线权切换

特点：只能处理简单的数据传送（随机性）

应用：一般用于主存与高速I/O设备的简单数据传送。

1. 程序直传方式，中断方式，DMA方式之间的区别？

中断方式和DMA方式都能响应随机请求，可并行操作；而程序直送方式则不能

中断方式在一条指令结束时响应请求，可以处理复杂事态，实现中低速I/O请求；而DMA方式在一个总线周期结束时响应请求，传送期间不需要CPU干预，但只能依靠硬件实现高速、简单I/O操作。

**《微型计算机原理与接口技术》重要知识点**

**第2章 微处理器与总线**

一、基本概念

1.微处理器主要由哪几部分构成？

运算器、控制器、内部寄存器

2.说明8088CPU中EU和BIU的主要功能。在执行指令时，EU能直接访问存储器吗？

（1）EU：指令译码、指令执行（ALU）、暂存中间运算结果（通用寄存器）、保存运算结果特征（FLAGS）。

（2）BIU：从内存中取指令到指令队列、负责CPU与内存、I/O接口之间的数据传送、在执行转移程序时，BIU清除指令队列，从指定的新地址取指令，并发送给执行单元执行。

不能，只能由BIU完成。

3.8086/8088CPU中，有哪些通用寄存器和专用寄存器？说明他们的作用。

1. 通用寄存器：
2. 数据寄存器：
3. AX：累加器，存中间结果；所有I/O指令都通过AX与接口传信息
4. BX：基址寄存器，间接寻址中存放基地址。
5. CX：计数寄存器，循环或串操作中存放计数值
6. DX：数据寄存器，在间接寻址的I/O指令中存放I/O端口地址；在32位乘除法操作中存放高16位数
7. 地址寄存器：
   1. SP：堆栈指针寄存器，其内容为栈顶的偏移地址
   2. BP：基址指针寄存器，访问内存时存放内存单元的偏移地址
8. 变址寄存器：变址寄存器在指令中常用于存放数据在内存中的地址。
   1. SI：源变址寄存器
   2. DI：目标变址寄存器
9. 控制寄存器：
10. IP：指令指针寄存器
11. 标志寄存器FLAGS:
    1. CF:进位标志位，若最高位有进/借位则1
    2. PF:奇偶标志位，运算结果低8位“1”有偶数个则1
    3. AF:辅助进位标志位，加减时，若bit3向bit4有进位或借位则1
    4. ZF:零标志位，结果为0则1
    5. SF:符号标志位，最高位为1则1
    6. OF:溢出标志位，结果超出有符号数表示范围则1

（3）段寄存器：用于存放相应逻辑段的段基地址

1. CS:代码段寄存器，存放代码段的段基地址
2. DS：数据段寄存器，存放数据段的段基地址
3. ES：附加段寄存器，存放附加段的段基地址
4. SS：堆栈段寄存器，存放堆栈段的段基地址

4.8086/8088系统中，存储器为什么要分段？一个段最大为多少字节？最小为多少字节？

（1）8088/8086有20条地址线，可寻址最大物理内存容量为1MB（220），任何一个内存单元都有1个20位的地址，而8088/8086内部寄存器都是16位，而访问内存单元多数要通过寄存器间接寻址，于是将地址空间分段，用段基址加上段内偏移地址来访问物理存储器。

（2）1个段最大为64K个字节（寄存器为16位，216=64）

（3）最小为16个字节（当两个段的偏移地址相同时，段基地址相差0001H则整体物理地址相差00010H即16个）

5.8086/8088系统中，物理地址和逻辑地址是指什么？

（1）物理地址：8088/8086有20根地址线，可寻址的最大物理内存容量为1MB，每一个内存单元的20位地址称为物理地址

（2）逻辑地址：段基地址（决定该逻辑段在内存中的位置）和段内偏移地址（决定该存储单元相对段起始单元的距离）为逻辑地址

6.8088/8086 CPU的特点？

（1）采用并行流水线工作方式（执行单元EU和总线接口单元BIU）

（2）对内存空间实行分段管理（分为4段，管理1MB空间）

（3）支持多处理器系统（最小模式：单处理模式‘最大模式：多处理器模式）

1. 计算题：

8086/8088系统中，物理地址和逻辑地址的转换？

物理地址 = 段基地址 x 16 + 段内偏移地址

1. **8086/8088指令系统**
2. 基本概念

1.8086/8088指令系统的8种寻址方式

（1）立即寻址：源操作数是一个立即数，存储在DS段中

（2）寄存器寻址：速度较快，MOV指令源操作数与目的操作数位数一致

（3）直接寻址：给出源操作数偏移地址，存储器操作数长度由另外一个操作数确定；默认在数据段，允许段重设

（4）寄存器间接寻址：寄存器内容是操作数的偏移地址，操作数在内存储器中。寄存器选择SI、DI、BX时，操作数在数据段；用BP时操作数在堆栈段。

（5）寄存器相对寻址：可使用SI、DI、BX、BP，规则与上类似。MOV AX, DATA[SI]

（6）基址——变址寻址：由基址寄存器（BX/BP）的内容和变址寄存器（SI/DI）内容相加形成操作数的偏移地址，操作数的段基地址由基址寄存器决定MOV AX, [BX][SI]

(7)基址——变址——相对寻址 MOV AX, DATA[SI][DI]

（8）隐含寻址：某些指令隐含了操作数地址MUL BL : AL x BL->AX

二、六大类指令

1.数据传送：

1）通用数据传送：MOV、堆栈操作指令push/pop、交换指令XCHG、查表转换指令XLAT、（字位扩展指令不要求）

2）输入输出IN、OUT

3）地址传送LEA、LDS、LES

4）标志位操作PUSHF、POPF、LAHF、SAHF

2.算术运算类指令：

1）加法运算指令ADD、ADC、INC

2）减法运算指令SUB、SBB、DEC、CMP（求补指令NEG不要求）

3）乘法指令（不要求）

4）除法指令（不要求）

3.逻辑运算和移位指令：AND、OR、NOT、XOR、TEST、SHL、SHR、SAL、SAR、ROL、ROR、RCR、RCL

4.串操作指令：MOVS、CMPS、SCAS、LODS、STOS

5.程序控制指令：

1）转移指令：JMP、条件转移指令

2）循环控制指令：LOOP、LOOPZ/LOOPE、LOOPNZ/LOOPNE

3）过程调用指令：CALL

4）中断控制指令：INT

6.处理器控制指令：常用的CLD 、STD、CLI、STI、HLT、CLC、STC

1. **汇编语言程序设计**
2. 基本概念
3. 指令与伪指令区别？

指令：由指令助记符组成的可被CPU执行的语句；

伪指令（指示性语句）：本身不产生对应的机器目标代码。它指示汇编程序对其后面的指令语句和伪指令语句如何处理。

1. 伪指令：段定义伪指令、数据定义伪指令：常数(字符串的使用DB、DW、DD)、变量和标号,当前位置计数器$（$!作变量使用，是16位偏移量）与定位伪指令ORG等伪指令。

(1)变量名在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，在伪指令语句中引用表示取偏移地址或偏移地址（低位）+段基地址（高位）

(2)位置计数器$（$作变量使用，是16位偏移量）

(3)ORG: 用来改变位置计数器的值。将数值表达式的值赋给当前位置计数器$ 。ORG语句为其后的变量或指令设置起始偏移量。

(4)DOS功能调用

二、汇编语言程序设计

1.汇编语言源程序结构

见笔记本

2.汇编语言程序设计四种类型：顺序程序设计、分支程序设计、循环程序设计、子程序设计

见笔记本

**第6章 输入输出和中断技术**

一、基本概念

1.输人/输出系统主要由哪几个部分组成？主要有哪些特点？

（1）输入输出设备、输入输出接口、输入输出软件

（2）复杂性、异步性、实时性、与设备无关性

2.I/0接口的主要功能有哪些？有哪两种编址方式？在8088/8086系统中采用哪一种编址方式？

（1）数据的缓冲与暂存；信号电平和类型的转换；增加信号的驱动能力；对外设进行检测、控制与管理、中断处理

（2）与内存统一编址，单独编址

（3）8088/8086采用I/O端口独立编址

3.试比较 4种基本输入/输出方法的特点。

（1）无条件传送：要求外设总是处于准备好状态。适用于简单外设

（2）查询式传送：每满足一次条件只能进行一次数据传送。适用于外设并不总是准备好，对传送速率和效率要求不高

（3）中断方式传送：外设在需要时向CPU提出请求，CPU再去为它服务。服务结束后或在外设不需要时，CPU可执行自己的程序。CPU效率高，实时性好，速度快。

以上三种：信息的传送均需通过CPU

（4）直接存储器存取：总线由DMA控制器（DMAC）进行控制（CPU要放弃总线控制权），内存/外设的地址和读写控制信号。CPU不再担当数据传输的中介者

4.主机与外部设备进行数据传送时,采用哪一种传送方式CPU的效率最高？

DMA方式。

二、逻辑电路及程序设计

1.根据状态端口地址查询相关状态位的值，符合条件通过数据端口传送一次数据。

见笔记本

2.中断向量表中中断向量设置。

8086/8088中断向量表位于内存最低1KB（00000H-003FFH），中断向量表存放中断服务程序入口地址，每个入口占用4个字节，高字节为段基地址，低字节为偏移地址。

中断向量表所在的段基址=0，存放子程序入口的单元的偏移地址=n× 4

存储器设计：

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）

图示, 示意图

描述已自动生成图示, 示意图

描述已自动生成

图示

描述已自动生成

指令流程题：

【例2】ADD (R1)+, X(R0);

FT0：M→IR EMAR、R、SIR

PC+1→PC PC→A、A+1、DM、CPPC、1→ST、CPT()、CPFT()、

CPST()、CPDT()、CPET()

ST0: PC→MAR PC→A、输出A、DM、CPMAR、T+1、CPT( )

ST1: M→MDR→C EMAR、R、SMDR、MDR→B、输出B、DM、CPC、T+1、

CPT()

ST2:C+ R0→MAR C→A、R0→B、A+B、DM、CPMAR、T+1、CPT( )

ST3:M→MDR→C EMAR、R、SMDR、MDR→B、输出B、DM、CPC、T+1、

CPT()

ST4:PC+1→PC PC→A、A+1、DM、CPPC、1→DT、CPT()、CPFT()、

CPST()、CPDT()、CPET()

DT0: R1→MAR R1→A、输出A、DM、CPMAR、T+1、CPT()

DT1: M→MDR→D EMAR、R、SMDR、MDR→B、输出B、DM、CPD、T+1、

CPT()

DT2: R1+1→R1 R1→A、A+1、DM、CPR1、1→ET、CPT()、CPFT()、

CPST()、CPDT()、CPET()

ET0: C+D→MDR C→A、D→B、A+B、DM、CPMDR、T+1、CPT()

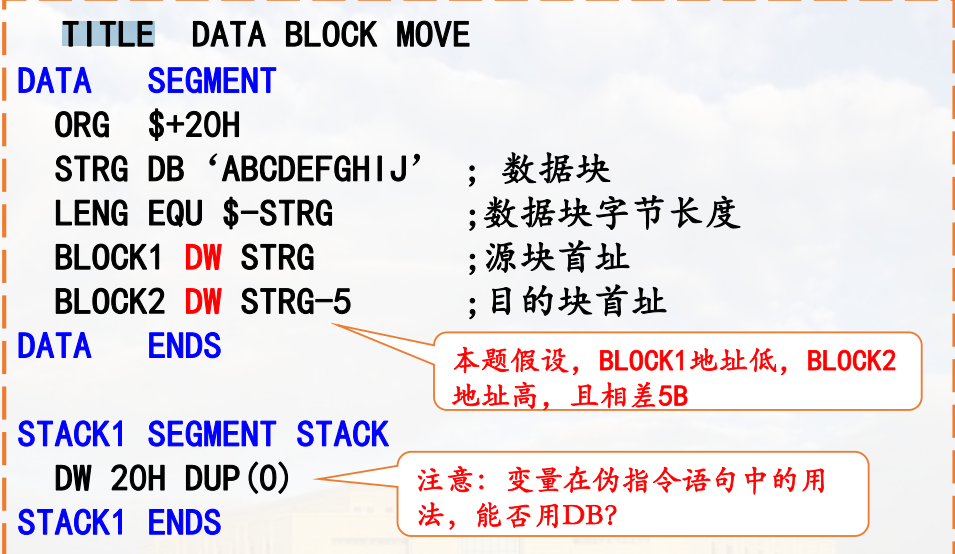
ET1: MDR→M EMAR、W、T+1、CPT()

ET2: PC→MAR PC→A、输出A、DM、CPMAR、1→FT、CPT()、CPFT()、

CPST()、CPDT()、CPET()

汇编编程题：  
1.存储单元之间不能直接传送！可以先传给AL/AX再传给另外一个存储单元

（3）PPT“3.4 汇编语言程序设计-3” P12 “例3 实现将存储器中的源数据块传送到目的数据块。”（注意串长度的统计！）

文本, 信件

描述已自动生成文本

描述已自动生成

3.PPT“3.4 汇编语言程序设计-3” P34 “例6 将字单元VARW 中含1的个数(含1的个数是指用二进制表示时,有多少个1)统计出来，存入CONT单元中”

用SHL+JS来判断二进制表示最高位数值

4.教材《接口技术》P133 【例3-38】，在ES段中从2000H单元开始存放了16个字符“*this* *is* *for* *you*”，寻找其中有无字符“o”。若有则记下搜索次数，将搜索次数写入到DATA1单元，并将存放“o”的地址写入DATA2单元。

用重复操作前缀REPNZ + SCASB来查找ES段中有无特定字符，将查找字符放入AL中

5.有10个16位的无符号（JA/JNBE,JAE/JNB,JB/JNAE,JBE/JNA），请按照从小到大的顺序进行排序。

/\*有符号数（JG/JNLE,JGE/JNL,JL/JNGE,JLE/JNG）\*/

含有双重循环的排序题，有子函数要注意保护寄存器的值，字类型的传送寄存器的值应该＋2！！！

DSEG SEGMENT

NUM1 DW 74H，89H，20H，0FDH，07FH，15A0H，0F0FAH，50H，15H，0F02H

DSEG ENDS ;定义数据段

ESEG SEGMENT

DB 0

ESEG ENDS ;定义附加段

SSEG SEGMENT PARA STACK

DW 20H DUP (0)

SSEG ENDS ;定义堆栈段

CSEG SEGMENT

ASSUME CS:CSEG,DS:DSEG, ES:ESEG,SS:SSEG

START: ;指令开始地址

MOV AX,DSEG ;

MOV DS,AX ;初始化DS

MOV AX,ESEG;

MOV ES,AX ;初始化ES

LEA DI, NUM1

MOV CX, 9 ;数据段的10个数据，比较9次

NEXTI：CALL SORTNUM

ADD DI，2

LOOP NEXTI

MOV AH, 4CH ;

INT 21H ;返回DOS操作系统

SORTNUM PROC

PUSH CX

MOV BX,2

MOV AX, [DI]

LOP1: CMP AX, [BX+DI]

JAE LOP2

XCHG AX, [BX+DI]

MOV [DI], AX

LOP2: ADD BX, 2

LOOP LOP1

POP CX

RET

SORTNUM ENDP

CSEG ENDS

END START ;汇编结束标志

I/O编程题：  
1.开关闭合状态表示输入0，使二极管亮表示输入0

2.输入输出记得用DX当端口地址

3.一次只能传输一个数据，每传输完即要返回判断输入是否满足条件

4. 判断多位同时满足情况，先用AND再用CMP！

5.用HLT结束程序

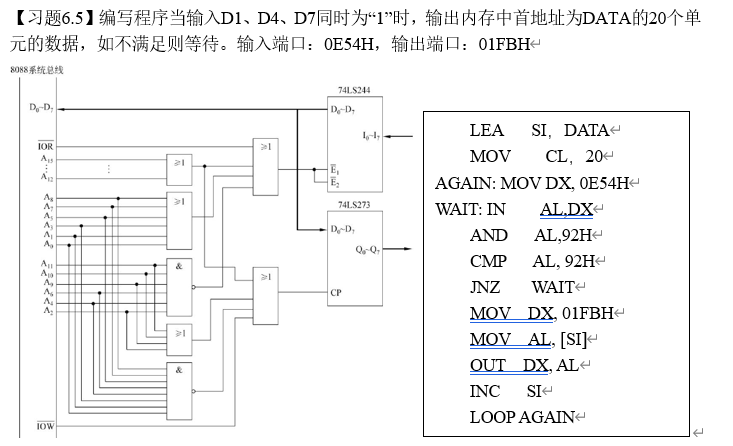
【例1】编程实现：当开关闭合时，使发光二极管亮。

图示

描述已自动生成

图表, 瀑布图

描述已自动生成



【例5】中断向量表的初始化，将用户自定义的将类型码为48H的中断服务子程序TIMER的中断向量放入向量表。

分析：向量表所在的段地址=0，存放子程序入口的单元的偏移地址=n×4。

**MOV AX，0000H**

**MOV DS，AX**

**MOV SI，0120H**

**MOV BX，OFFSET TIMER**

**MOV [SI]，BX**

**MOV BX，SEG TIMER**

**MOV [SI+2]，BX**